APPLICATION OF GASP II TO JOB-SHOP SCHEDULING

by 1264

CHANDRA KUMAR PATEL

B.E. (Hons.), Electrical Engineering, University of Jabalpur

Jabalpur, (M.P.), India, 1967

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

1970

Approved by:

L. E. Grosh

Major Professor

LD
2668
T4
1970
P368

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
II.	JOB-SHOP SCHEDULING	4
	1. CLASSIFICATION OF SCHEDULING PROBLEMS	4
	2. DEFINITION OF TERMS	6
	3. PERFORMANCE MEASURES FOR THE SHOP	10
	4. EFFECTS OF SCHEDULING	10
III.	LITERATURE REVIEW	12
	1. INTRODUCTION	12
	2. ASSUMPTIONS	13
	3. JOB-SHOP SCHEDULING RESEARCH	14
IV.	THE MODEL	21
	1. TERMINOLOGY	21
	2. ASSUMPTIONS	21
	3. MEASURES OF PERFORMANCE	21
	4. PRIORITY RULES	23
	5. LOOK AHEAD RULES	27
	6. ASSIGNMENT OF DUE-DATES	29
V.	THE EXPERIMENT	31
	1. DESCRIPTION OF THE SYSTEM	33
	2. INPUT DATA	34
VI.	THE SIMULATION	38
	1. THE SIMULATION MODEL	38
	2. CONDITIONS FOR THE EXPERIMENTAL RUNS	53

TABLE OF CONTENTS (cont.)

Chapter		Page
VII.	RESULTS	59
	1. EXPERIMENT ONE	59
	2. EXPERIMENT TWO	73
VIII.	CONCLUSIONS	91
	REFERENCES	93
	ACKNOWLEDGEMENT	96
APPENDIX A.		97
	GASP SUMMARY REPORT	98
	SAMPLE QUEUE PRINTOUT	100
APPENDIX B.		103
	DESCRIPTION OF VARIABLES FOR THE FORTRAN PROGRAM	104
	FORTRAN LISTING OF THE SIMULATION PROGRAM	105

LIST OF FIGURES

Figure		Page
1.	Frequency distributions of Interarrival times	36
2.	Frequency distributions of Processing times	37
3.	Subroutine RANJOB	42
4.	MAIN PROGRAM.	43
5.	Subroutine EVENTS and Subroutine OUTPUT	44
6.	Subroutine ARIVAL	45
7.	Subroutine ENPROS	46
8.	Subroutine SELECT	49
9.	Subroutine CHECK.	51
10.	Convergence of Interarrival and Interdeparture times.	55
11.	Frequency distribution of Interdeparture times.	56
12.	Frequency distribution of Lateness (FCFS)	63
13.	Frequency distribution of Lateness (SPT).	64
14.	Frequency distribution of Lateness (SLACK).	65
15.	Frequency distribution of Lateness (S/OPN).	66
16.	Frequency distribution of Lateness (S/RT)	67
17.	Frequency distribution of Lateness (DDATE).	68
18.	Frequency distribution of lateness with different due-dates - SPT rule	75
19.	Relationship between due-date multiplier and mean lateness for SPT rule	76
20.	Relationship between due-date multiplier and % jobs late with SPT rule.	76
21.	Lateness frequency distribution for SLACK rule - DATE = 5	80
22.	Lateness frequency distribution for SLACK rule - DATE = 7	81

Page

23. Effect of factor GSLK on SLACK rule 82

24. Lateness frequency distribution for SLACK/OPERATION rule - DATE = 5 84

25. Lateness frequency distribution for SLACK/OPERATION rule - DATE = 7 85

26. Effect of factor GSLK on SLACK/OPERATION rule 86

27. Lateness frequency distribution for SLACK RATIO rule - DATE = 5 88

28. Lateness frequency distribution for SLACK RATIO rule - DATE = 7 89

29. Effect of factor GSLK on SLACK RATIO rule 90

LIST OF TABLES

Table		Page
1.	Summary of Results for Experiment One with Constant Due-dates	69
2.	Number of Jobs Late with Constant Due-dates	70
3.	SPT Performance at Different Levels of Utilization with Constant Due-dates	71
4.	Summary of Results with Constant Due-dates	72
5.	Performance of SPT and SLACK rules under different Due-dates	74
6.	Effect of LOOK AHEAD on SLACK rule	79
7.	Effect of LOOK AHEAD on SLACK PER OPERATION rule	83
8.	Effect of LOOK AHEAD on SLACK RATIO rule	87

CHAPTER I

INTRODUCTION

The development of high speed electronic computers and sophisticated simulation languages such as GPSS/360, SIMSCRIPT, GASP II etc. has made possible the study of complex man-machine systems. Most man-machine systems are made up of components which interact with each other. The interaction between the various components of the system and the behavior of individual components is seldom deterministic and delays are inherent between decisions and resulting actions at various places in the system. These complexities have made it almost impossible to obtain solutions using present day analytical tools.

Simulation allows experimentation with a model of the system. The model is constructed by expressing the characteristics of the system components in the form of mathematical expressions and logical statements, with all components being interconnected by networks of information and material flow. The actual experimentation is performed with a simulator which when run on a computer generates performance data typical of the system under study.

The primary goal of most job-shop simulation studies has been to determine ways to improve the performance of a system from a macroscopic viewpoint. In the present work an effort to investigate the effect of various priority rules on job-shop scheduling has been made. The choice of the priority rules and other experimental conditions are based on the works of Gere (15) and Conway (9). The effect of the look ahead feature suggested by Gere (15) has also been studied.

The GASP II (General Activity Simulation Program), a FORTRAN based

simulation language developed by Pritsker and Kiviat (32) has been used for programming.

Scheduling is the final stage in production planning. The order in which units (jobs or orders) are to be processed at each of the series of machine centers is known as sequencing. A typical manufacturing plant may have a number of machines or facilities. Jobs or orders are continuously coming in to be processed on the various facilities. The purpose of scheduling is to determine what job a facility should process at any given time, so as to fully utilize the plant capacity and at the same time minimize the lateness of orders. The ideal situation would be one in which all the facilities are kept busy all the time. In order to meet due-dates, a facility should be available as soon as a job is ready for processing on that facility. These are two extremely difficult conditions to fulfill. The complexity of the scheduling problem has attracted a number of researchers, who have been trying to find ways of meeting these two conditions.

A solution to the complex scheduling problem could be based on a number of different criterion. One of these could be the total elapsed time, i.e. a method which minimizes the total time required to complete all the jobs and hence minimizes the idle time and maximizes the utilization of facilities, can be said to be optimal. Others could be, minimize total cost of the product, minimize set-up cost of machines and minimizing the in-process inventory.

One of the earliest and most widely used methods for scheduling is the use of the Gantt chart. The scheduler would chart the projected work load for each machine or work center, hour by hour, order by order. As

the complexity of the shop increases it becomes increasingly difficult to keep track of the flow of jobs in the shop. To do this would require a constant updating of the Gantt chart. Jobs enter the shop according to some stochastic process and the processing times are not fixed times but actually follow some probability distribution.

Research in the field of scheduling, using the analytical tools available has been restricted to solving very small problems. Further, these studies have been restricted by a number of simplifying assumptions. A better method would be to test these procedures in the real shop. The difficulties involved in conducting such an experiment and the prohibitive cost of such a study have been the limiting factors in the use of this method.

Faced with these problems, researchers have resorted to simulation as a means for studying the action of jobs, machines, and scheduling procedures with a computer program. A number of such studies have been made in recent times; the earliest among these are due to Jackson, Nelson, and Rowe at UCLA (19, 34), Baker and Dzielinski at IBM (2); and Conway at RAND (9).

Most simulation studies have been focused towards investigating the effect of various priority rules on job-shop scheduling. Some significant results have been obtained by the application of this technique to industry (25).

As production becomes fully automated, well formulated decision rules will be required to control the operation of the plant. Production facilities of the future would rely increasingly on the computer as a means for carrying out daily decisions in the factory.

CHAPTER II

JOB-SHOP SCHEDULING

1. CLASSIFICATION OF SCHEDULING PROBLEMS

Job-shop scheduling has been regarded as one of the most complex problems faced by industry. The difficulty arises due to a wide variation in the requirements of each job or order, such as processing requirements, routing, number of operations etc. Depending on the routing of operations, a shop may be classified as either a flow-shop or job-shop. An intermittent system which lies between these two classifications is known as the production shop.

In a job-shop equipment of the same kind is grouped together. This is done primarily for two reasons. Firstly, the job-shop produces specialized items, usually made to customer specification and as such the equipment is not fully utilized. Secondly, the sequence of operations and processing times are different for different items, and this sequence is not common for a large fraction of the products. Thus, for both economic and technological reasons, equipment of the same kind is grouped together.

The production shop has been classified as an intermittent system differing from both the job-shop and flow-shop (4). This difference is brought about by the nature of jobs processed. A production shop is geared to manufacture large volumes of a rather small number of standard items (jobs), having almost identical routings. The production shop is faced with the lot-size problem and it becomes essential to determine the

number to produce in a single run.

One of the distinguishing features of job-shops is that a job-shop produces to customer order according to customer order size, whereas a production shop maintains a finished goods inventory to meet subsequent demand. Thus in the case of a job-shop a customer has to wait for his order while in the case of a production shop, this service is provided by the manufacturer at the cost of holding finished goods inventory.

A flow-shop on the other hand can be distinguished by the order in which machine numbers appear in operations of individual jobs. In the case of a flow-shop all jobs follow essentially the same path from one machine to another. This is the most striking difference between the job-shop and flow-shop. It is this routing difference that makes the job-shop scheduling problem a complex one.

The randomly routed job-shop is one in which there is no common pattern of movement from one machine to another. Most real shops fall between the job-shop and flow-shop, but almost all the research in scheduling has assumed one or the other of these two extreme cases.

Job-shop scheduling problems can be distinguished either as static or dynamic depending on the nature of job arrival. In a static problem a certain number of jobs arrive simultaneously in the shop that is idle and immediately available for work. Since no further jobs will arrive, attention could be focused at scheduling and completing this set of jobs. In the case of the dynamic problem, jobs are continuously arriving at the shop. Due to this difference in arrival pattern, entirely different methods have to be adopted for the purpose of scheduling.

Let us now consider the methods that have been used to deal with the scheduling problem in job-shops. In most cases the foreman handles the detailed scheduling, making ad hoc decisions whenever necessary. In some shops, a job may be given top priority because of the importance of the customer or because the job has been delayed in prior operations. Thus, there is no standard set procedure for scheduling, and the performance of the shop depends on the experience of the person in charge of scheduling. A relatively new approach that is being increasingly used is to establish a priority rule, such that higher priority is assigned to a job released earlier to the shop, or one with an earlier delivery date.

One of the earliest methods used in job-shop scheduling is the use of the Gantt chart (5). This method, though useful, has been found costly to maintain since it requires constant updating.

2. DEFINITION OF TERMS

As discussed in Chapter I, a number of different criteria can be employed in the study of scheduling problems. Definitions and descriptions of the attributes of job i that have been used and the interrelations between them is given below, (12):

For a job-shop process the shop is completely defined by giving the number of machines and a set of jobs. Let us consider a "m-machine shop" and identify the individual machines by the integers 1, 2, . . ., m.

The jobs can similarly be identified by integers 1, 2, . . ., n.

Arrival-time - r_i

This is defined as the time at which the job is released to the shop by some external job-generation process. It has also been referred to as the ready-time or release-time.

Due-date - d_i

This is defined as the time specified by some external agency by which the job must leave the shop. It may also be referred to as the time by which all operations on the job must be completed.

The difference between the due-date and arrival-time is defined as the total allowance for time in the shop.

$$a_i = d_i - r_i$$

Processing-time - $P_{i,j}$

The job consists of a set of g_i operations which could be described by g_i pairs of values:

$$\begin{matrix} m_{i,1} & P_{i,1} \\ m_{i,2} & P_{i,2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ m_{i,g_i} & P_{i,g_i} \end{matrix}$$

The machine required to perform the jth operation is identified as $m_{i,j}$, $1 \leq m_{i,j} \leq m$.

$P_{i,j}$ is defined as the processing-time. This is the amount of time that machine $m_{i,j}$ requires to perform the operation.

The total processing-time for the job is given by P_i.

$$P_i = \sum_{j=1}^{g_i} P_{i,j}$$

Waiting-time - $W_{i,j}$

This is defined as the time a job must wait after the completion of the (j-1)th operation before beginning the jth operation.

The total waiting-time for a job i.e., the sum of the waiting-times for all operations of the job is given by W_i.

$$W_i = \sum_{j=1}^{g_i} W_{i,j}$$

A scheduling procedure that reduces the waiting times for individual jobs would also reduce the lateness and shop-time.

Completion time - C_i

This is defined as the time at which processing of the last operation of the job is completed.

$$C_i = r_i + \sum_{j=1}^{g_i} P_{i,j} + \sum_{j=1}^{g_i} W_{i,j} = r_i + P_i + W_i$$

Flow-time - F_i

This is defined as the total time that the job spends in the shop. It is also known as the shop-time.

$$F_i = \sum_{j=1}^{g_i} P_{i,j} + \sum_{j=1}^{g_i} W_{i,j} = P_i + W_i = C_i - r_i$$

Lateness - L_i

The lateness of job i is defined as the difference between the completion-time and due-date.

$$L_i = C_i - d_i = F_i - a_i$$

Lateness is the algebraic difference between the actual completion-time and the desired completion-time for each job. It does not consider the sign of the difference. On the other hand tardiness considers only positive differences, i.e., it only takes into account those jobs which are completed after their due-date.

$T_i = \max [o, L_i]$, the tardiness of job i.

3. PERFORMANCE MEASURES FOR THE SHOP

The four important measures of performance in a job-shop are, (12),

i) Facility utilization

ii) Work-in-process-inventory

iii) Shop time

iv) Lateness

The first two are attributes of the shop while the next two are attributes of jobs passing through the shop.

All these measures are important and need consideration when scheduling decisions are taken. For economic reasons a plant might consider it more important to keep the in-process inventory low. In most cases, however, the ability to fulfill delivery promises on time is far more important than any of the other measures.

4. EFFECTS OF SCHEDULING

A scheduling procedure may affect a production facility in many ways. The overall effect, of course, would be on the costs of inventory, lateness and utilization. It would also help in providing an increased control over the operation of the plant.

The cost of holding inventory, though difficult to measure, are real and a good scheduling procedure should consider ways and means of reducing in-process inventory.

An increase in facility utilization would mean greater output. A scheduling procedure that reduces the mean flow-time would, in essence, permit a facility to do more work.

Finally, since a job-shop manufactures to customer ordering as opposed to the production shop which holds finished goods inventory, it is essential that it be able to deliver the goods in time. A manufacturing facility that is not able to meet its customer's requirements in time will pay the penalty by losing future orders.

Thus a good scheduling procedure should be able to reduce cost of holding inventory, lateness and improve facility utilization. As stated previously it is difficult to actually measure these costs. Simulating the system, using different priority rules, is one of the methods that is increasingly being used to obtain a solution without actual experimentation.

CHAPTER III

LITERATURE REVIEW

1. INTRODUCTION

The development of several simulation languages has made possible the investigation of complex job-shop scheduling problems. Researchers have attempted to obtain a solution to this problem using a number of different techniques. Due to the complexity of the problem most researchers have resorted to simulation as a tool to study the job-shop system.

Most studies have dealt with models of the following type (15):

i) Optimizing rules for one- or two-machine situations.

ii) Mathematical models, such as integer programming models; which provide an optimum solution to the problem, but are computationally unfeasible.

iii) Iterative methods, employing a Monte Carlo device or a rule that reduces the number of schedules which are potentially optimum.

iv) The use of priority rules together with heuristics in a computer program to obtain a solution.

v) Testing the effect of various priority rules by simulating the system.

The mathematical techniques though capable of giving the optimal solution are restricted due to computer limitations. These studies are of theoretical interest only, due to the various restrictions that

have been imposed on the model. Even a small shop with ten machines and ten jobs would have $(10!)^{10}$ possible job permutations. Thus one can very well imagine the problems in trying to set up a scheduling procedure for even a ten machine problem with the aid of mathematical techniques.

It is these problems that have led investigators to simulated experimentation. Simulation, though not capable of providing optimal solutions allow the experimenter to study the behavior of the system under different conditions.

In recent times a number of simulation studies of real shops have been made. A job-shop simulator set up at General Electric to evaluate scheduling and dispatching rule has been in use since 1958 (12). Simulation has successfully been used to evaluate the effect of various management decisions on the operation of the Fabrication Shop at Hughes Aircraft Company (25). Conway (12) remarks that there is no evidence to suggest that the use of actual shop data and dimensions significantly alters the comparative performance of key procedures. The problem in studying a real system is in representing all characteristics of the system and in managing the data. With further developments in computer technology and automation of production control systems, these problems will disappear, making possible the setting up of a simulator for actual shops.

2. ASSUMPTIONS

The numerous studies made so far have been restricted by a number

of simplifying assumptions. A brief summary of some of the major assumptions imposed is given below (12, 15):

i) Each machine is continuously available for assignment, without significant division of the time scale into shifts or days, and without consideration of temporary unavailability for causes such as breakdown or maintenance.

ii) Jobs are strictly-ordered sequences of operations, without assembly or partition.

iii) Each operation can be performed by only one machine in the shop.

iv) There is only one machine of each type in the shop.

v) Preemption is not allowed - once an operation is started on a machine, it must be completed before another operation can begin on that machine.

vi) The processing-times of successive operations of a particular job may not be overlapped. A job can be in process on at most one operation at a time.

vii) Each machine can handle at most one operation at a time.

viii) No setup time for operations.

x) Instantaneous transfer to next machine (or queue) after an operation has been completed.

3. JOB-SHOP SCHEDULING RESEARCH

Jackson (20) has solved the problem of sequencing several jobs on a single machine so as to minimize maximum tardiness or to minimize the

sum of completion times. The maximum tardiness is minimized by arranging jobs according to due-dates in increasing order. The sum of completion times is minimized if jobs are arranged in increasing order of operation times.

One of the earliest papers in the field of scheduling is Johnson's solution to the two-machine flow-shop problem (22). The algorithm given can be used for sequencing n jobs, simultaneously available, so as to minimize the maximum flow-time. The general acceptance of minimizing the maximum flow-time as a criterion for the general job-shop problem is due to Johnson's result (12). Johnson has shown that the rule applies also to the special three-machine case.

The job-shop scheduling problem can be formulated as an integer linear programming problem. The first algorithm for solving integer programming problems is due to Gomory (18). The development of this new technique simulated a great deal of interest amongst researchers. Bowman (3), Wagner (39), and Manne (27) are amongst the earliest to formulate the job-shop scheduling problem using integer programming. According to Bowman, a simple problem involving three jobs and four machines would require an integer programming problem containing 300 to 600 variables and many more constrains. The formulation by Wagner of the problem would be of the same order of magnitude. Of the three, Manne's formulation is the most compact requiring 31 variables and 94 constrains, and could be solved on computers in a reasonable length of time. Further complications arise when additional constrains are added during the course of its solution. Finally it could be said that none of these

formulations are computationally practical.

Story and Wagner (37) solved a number of three-machine problems in which a schedule span is to be minimized, with up to nine jobs on the IBM 7090 computer by the integer programming method. Based on their experience with this technique, the authors conclude, "we have not yet found an integer programming method that can be relied upon to solve most machine sequencing problems rapidly." (29).

Computational difficulties and the restriction on the size of problem that can be solved using the techniques discussed so far, have led people to resort to simulation and heuristic procedures to obtain a solution to the scheduling problem. These methods do not provide an optimal solution to the problem, but have the advantage of at least allowing one to find a relatively good schedule.

Heller (19) investigated a ten-machine flow-shop and found that the schedule times are approximately normally distributed for large numbers of jobs. This knowledge of the probability distribution permits one to determine the sample size to use. The problem of sample size is a real one and one must be able to determine this beforehand. One could, of course, take a large sample size at the expense of high sampling costs.

Another approach to the scheduling problem would be to reduce the total number of possible optimal schedules through an algorithm and then take a Monte Carlo sampling of the remaining possibilities. Finally one could employ simulation. Giffler, Thompson and Van Ness (17) have devised an algorithm for complete enumeration of all 'active' schedules. An active schedule is defined to be a feasible schedule with the following

properties: (a) a machine is not idle for a length of time in which an order simultaneously idle could be processed completely, and (b) whenever an order is assigned to a machine, processing should begin as soon as both the machine and order are free. Giffler (16) has also developed a schedule algebra to solve production scheduling problems.

In the Monte Carlo version of the Giffler-Thompson program, only a small sample from the population of active possible schedules is tested. This does not guarantee the finding of an optimal solution, but it does provide a means of testing a fairly large number of active schedules and finding the shortest amongst them.

Thus, we may conclude that except for the special case considered by Johnson (22) and the simple problems solved by complete enumeration and by other techniques like integer programming, little success has been obtained in solving the scheduling problems. It is for these reasons that simulation has in recent years found limited practical application as well as eliciting a great deal of theoretical study. Baker and Dzielinski (2) set up a shop simulator on the IBM 704 and tested several rules such as first in - first out, greatest number of operations remaining, greatest remaining processing-time, and found the "shortest imminent process-time" rule was best, if average job time in the shop is used as a measure of effectiveness. Rowe (34) tested several rules by simulation. He used the first come - first served rule, minimum (or maximum) imminent processing-time and earliest start date rules. The concept of "flow allowance" was introduced in another rule. The flow allowance is in effect an estimate of total time,

including delays, that a job takes. The minimum imminent processing-time rule was again found best for minimizing the expected waiting time, and the flow allowance was shown to be significant in reducing waiting time.

Most of the recent literature on job-shop scheduling is due to Conway. Conway, Johnson, and Maxwell (11) set up a shop simulator for the IBM 650 and Burroughs 220 computers with the shop envisioned as a network of queues. This is one in a series of experiments that have been conducted so far. In another work Conway and Maxwell (10) investigated the performance of the "shortest imminent operation" rule. The model studied consisted of a simple shop with n jobs and one machine. Again the shortest operation rule was found to give the best performance while rules independent of processing-time were found to be equivalent. Essentially the same result is shown to hold for queuing systems with exponentially distributed arrival times (12). They conjectured that the above-mentioned results would continue to hold for a job-shop characterized as a network of queues. The results of the simulation runs give support to this conjecture and also show that for the shortest operation rule to be effective, an accurate time estimate for job times is not necessary. Finally, they caution against the prohibitively long flow-times that may be experienced by an occasional job even though mean flow times may be reduced by the shortest operation rule. They tried, with some success, two variations of the rule to handle this: (1) alternate the shortest-operation rule with a low variance rule, (2) truncate the shortest-operation rule by imposing, a limit on the delay that

individual jobs will tolerate.

In a more detailed study at RAND, Conway (9) investigated the effect of 92 different priority rules on scheduling on a 9 machine job-shop. The measures of performance considered, include both measures of inventory-number of jobs in the system and work-content of these jobs - and measures of individual job progress - time in the shop, and lateness against an assigned due-date.

He concludes that if the difficulty and cost of implementation, were to be considered, then the shortest processing-time rule would be the best with respect to the minimization of the mean number of jobs in the queue. This rule also minimizes the mean number of jobs in the shop, the mean time in the shop, and the mean lateness.

Finally, Conway points out that the performance of a shop with respect to meeting its due-dates is a function not only of the sequencing rule employed but also of the method used to assign the due-dates to the jobs.

In particular, for minimizing the number of jobs that are late, the shortest processing time rule was superior to the slack per operation rule, when due-dates were assigned according to constant lead time or randomly. Furthermore, Conway also observed that under heavier loads, the shortest processing time rule appeared to be better than the slack per operation rule with respect to the number of jobs late.

Fisher and Thompson (13) studied the minimum make-span problem and introduced a learning routine in their program. They used two priority rules within the same schedule - shortest imminent operation and the longest remaining time. Either of these two rules was used depending upon its

relative success in its use on previous schedules. The learning routine did the selection and some improvement in schedules was observed. It was also found that the two rules in combination, in general, did better than any given rule applied singly.

Gere (15) studied the effect of priority rules together with heuristics or rules of thumb on job-shop scheduling. He tested some slack based rules alone and in combination with heuristics. He concluded that the heuristics which anticipate the future progress of a schedule; the alternate operation and look ahead heuristics, improve schedules significantly in both a statistical and practical sense. This improvement is obtained at a considerable increase in computer cost. The heuristic program was designed for the due-date problem, but the author concludes that it may be very effective in handling the minimum make-span problem as well.

Schwarz and Schriber (36) studied the model developed by Gere (15) using the GPSS/360 simulation package. They, however, did not consider the effect of any of the heuristics.

CHAPTER IV

THE MODEL

1. TERMINOLOGY

The terminology used to describe the job-shop is discussed below (9):

The shop can be said to consist of a set of machines, divided into subsets called machine groups. A machine group consists of identical machines, each having the same processing capability and performance. The work performed in the shop consists of a sequence of jobs. The routing of a job describes the order in which the operations are to be performed. An operation is identified by specifying the machine group on which the work is to be performed and by the length of time required to do the work. Each operation is performed by a single machine. The time required to do the work may be a random variable whose actual value is not known in advance of execution. This time consists of a set-up-time - time required to prepare the machine to do the work - and processing time - time to do the actual work.

2. ASSUMPTIONS

The assumptions mentioned in Chapter III shall pertain, although, it is highly doubtful that there are any interesting real systems which are simple job shops in this strict sense.

3. MEASURES OF PERFORMANCE

As has been previously stated in Chapter II, there are four measures

of performance of interest in a job-shop. The utilization of facilities, and the amount of work-in-process inventory are essentially attributes of the shop. The total time in the shop, and the lateness with respect to an assigned due-date are attributes of the jobs passing through the shop.

Little (26) has offered formal proof of the interrelatedness of shop performance measures. The following relationship holds as a shop is unsaturated (utilization < 1):

$$\bar{L} + \bar{A} = \bar{S} = \frac{\bar{W}\bar{N}}{m\bar{U}} = \frac{\bar{W}}{m\bar{U}} (\bar{N}q + m\bar{U}).$$

where:

$\bar{L}$ is the mean lateness,

$\bar{A}$ is the mean allowable time in the shop,

$\bar{S}$ is the mean total time in the shop,

$\bar{N}$ is the mean number of jobs in the shop,

$\bar{N}q$ is the mean number of jobs in the queue (total of all queues),

$\bar{U}$ is the mean shop utilization,

$\bar{W}$ is the mean work content (sum of the processing times) of the jobs,

m is the number of machines in the shop.

It is apparent from the relationship given above that a scheduling procedure that minimizes the mean number of jobs in queue for a particular shop and load would also minimize the mean number of jobs in the shop, the mean time spent in the shop by the jobs, and the mean lateness.

As has been previously stated, the purpose of this research is to investigate the effectiveness of various priority rules in reducing job

lateness. A brief description of the priority rules considered in this research is given below.

4. PRIORITY RULES

The concept of job or operation "priority" is inherent in many schedule-generation procedures. A priority is simply a numerical attribute of a job or operation on which selection is based.

Priority rules can have several effects upon the shop. In most situations it is desirable to have a rule that is designed to minimize the mean and variance of the lateness distribution, as this has the effect of giving as short a lead time as possible consistent with reliability in meeting promised deliveries. However, there are many situations in which this is not satisfactory, such as the case when early deliveries are of no advantage and are not offset equally by late deliveries. In this case "lateness" becomes an important measure.

Three major effects of most priority rules are listed below. These benefits may be obtained in varying degrees with tradeoffs possible depending upon the rule chosen.

i) A lowering of work-in-process inventory

ii) A decreased job flow-time through the shop

iii) An increased probability of meeting due-dates.

The choice of the six rules described below was motivated by the work of Gere (15) and Conway (9).

Definition of Symbols used in Priority Rules (9, 15)

t time at which selection for machine assignment is to be made

i index over the jobs to be processed by the shop

j index over the sequence of operations for which a job is in queue

J specific value of j; the operation for which a job is in queue

M_i the total no. of operations on the i-th job

$1 \leq j \leq M_i$

$T_{i,j}$ the time at which the i-th job became ready for its j-th operation -- i.e. the time at which the (j-1)th operation was completed

D_i the due-date of job i

$P_{i,j}$ the processing time for the j-th operation of the i-th job (including set-up and tear-down time, if any, assumed to be sequence independent)

$Z_i(t)$ the priority rating of job i at time t

The reason for calculating priority ratings is to determine what job is to be scheduled next on the given machine at the given time; the job with the minimum priority rating is the one selected. Thus the calculations need be carried out only for those jobs which are waiting to be processed on the machine. The notation does not take machines into account, since the procedure is independent of the machine being considered.

Priority Rule 1: First come, first served (FCFS)

$$Z_i(t) = \min\left[T_{i,j}\right]$$

Priority Rule 2: Shortest processing time (SPT)

The job which will tie up the machine for the least amount of time gets the machine next.

$$Z_i(t) = \min\left[\overline{P}_{i,j}\right]$$

Priority Rule 3: Job slack (SLACK)

Job slack is the number of "free" hours available before the due-date. If the job is necessarily going to be late, the slack is negative.

$$Z_i(t) = \min\left[D_i - t - \sum_{j=J}^{M_i} P_{i,J}\right]$$

Priority Rule 4: Job slack per operation (S/OPN)

Job slack per operation is the number of "free" hours available before the due-date divided by the number of operations remaining.

$$Z_i(t) = \min\left[\frac{D_i - t - \sum_{j=J}^{M_i} P_{i,J}}{M_i - J + 1}\right]$$

Priority Rule 5: Job slack ratio (S/RT)

Job slack ratio is the number of "free" hours available before the due-date divided by the number of hours remaining until the due-date.

$$Z_i(t) = \min\left[\frac{D_i - t - \sum_{j=J}^{M_i} P_{i,j}}{D_i - t}\right]$$

Priority Rule 6: Earliest due-date (DDATE)

$$Z_i(t) = \min\left[D_i\right]$$

The above set of rules provides varied representation both in time dependent characteristics and in the scope of information required. Hence, for rules 1, 2 and 6 the priority enjoyed by the job does not depend on the current time, whereas the other rules make use of the clock in assigning priorities. Finally rules 1 and 2 provide benchmarks of a sort against which results produced by the other four rules can be gauged.

The information necessary for processing jobs is contained in a job file. A job file is a record of the jobs on hand, the sequence in which the jobs are to visit various machines, and the operation times involved on the machines.

For example, "job slack" is a function of the job file, it equals the due-date measured in time units from the present minus the sum of operation times. On the other hand, the "first come, first served" rule is not a function of the job file. A rule which is not a function of the job file is in effect a random rule (15).

In effect, what is desired is a perfect prediction of delay time, that is, a knowledge of job completion time, together with a rule which

will minimize lateness. Conway (9) points out that little good is done by giving preference to a job that would then move on into a congested queue. He suggested that the priority assigning mechanism should look ahead to where the jobs will go on leaving this machine and give preference to those that will move to relatively empty queues. He tested several such rules and found them to be relatively better than the first come, first serve rule, though none of them could surpass the performance of the shortest processing time rule.

5. LOOK AHEAD RULES

Gere (15) and Conway (9) have investigated the behavior of several look ahead rules. The rules formulated so far are based in some way on processing times. They argue that the use of other information could be used to reduce queue lengths and inventory.

Conway (9) tested three such rules alone and in combination with other rules. The three rules used were XINQ, XWINQ and NINQ. These three rules though independent of the processing times depend on status information, such as, amount of work in queue to which the job will go for next operation. These rules performed better than the FCFS rule, but could not surpass the performance of the SPT rule. A simulation run using XWINQ took almost three times as long as a comparable run under SPT.

Gere (15) in his study, supplemented the basic priority rule with the alternate operation and look ahead heuristic and succeeded in improving the schedules significantly, although this increased the

computing cost.

The look ahead heuristic suggested by Gere is applicable only to the slack based rules. The theory behind the look ahead heuristic is as follows:

In using job slack or some adaptation of it as a priority rule, one is looking ahead to a limited extent. If the job slack is positive for a job, then that job will be completed on time unless it is delayed due to some conflict, such as, queuing at the machines. Thus, if one could look ahead and anticipate every conceivable conflict, an optimal schedule could be made. This of course would increase the computing costs considerably and the question of economics comes in.

However, one could look ahead a short ways and, before scheduling an operation check to see if a critical (i.e. late or nearly late) job is due to reach this machine at some future hour, yet before the scheduled operation is completed. If so, schedule that job. Check to see the effect of this on other jobs and depending on the resulting job lateness - either let the schedule remain, or replace the operation with the previously scheduled operation.

The look ahead rule and its variations studied in this work are described below -

Step-1: Check to see if the slack of the selected job is positive (refer to subroutine SELECT) and greater than the factor GSLK. This factor GSLK was assigned values 0, 5, 7.5, 10 for different runs and its effect on the schedule was observed.

Step-2: If the condition of Step-1 is fulfilled then,

(i) check to see if any job is due to come to the machine in question while the selected job is still there, and

(ii) determine if this job is late, i.e. it has a negative SLACK (refer to subroutine CHECK).

Step-3: If both conditions of Step-2 are fulfilled, then replace the selected job by the look ahead job (obtained by subroutine CHECK). If condition (ii) of Step-2 is not met then let the schedule remain.

6. ASSIGNMENT OF DUE-DATES

For a manufacturing unit, the ability to fulfill delivery promise is of utmost importance. Hence it is essential to determine a proper method for assigning due-dates.

In most practical situations, due-dates are assigned by some external agency and the production foreman is left with the task of meeting these due dates. In other situations the producer and consumer meet and agree upon a final due-date which depends mostly on the customer's need and competitive forces. In conclusion it could be said that no one general procedure could be set for assigning due-dates and each manufacturing unit must determine the best way for doing so.

Conway (9) proposed four different methods of assigning due-dates. Of these, two were intended to represent ways of assigning reasonable and attainable due-dates, providing a greater allowance for a job with more processing-time or a large number of operations. The other two

were representative of common practice. These four methods are summarized below:

I. Allowance proportional to the total work to be done on the job.

$$D_i = T_{i,1} + g \sum_{j=1}^{M_i} P_{i,j}$$

II. Allowance proportional to the number of operations on the job.

$$D_i = T_{i,1} + 8.883\ M_i$$

III. Constant allowance, the same for all jobs.

$$D_i = T_{i,1} + 78.7985$$

IV. Random allowance

$$D_i = T_{i,1} + 157.597\ U_i$$

where

U_i a random variable, uniformly distributed between 0 and 1, assigned to the i-th job.

CHAPTER V

THE EXPERIMENT

The experimental work can be divided into two parts:

Experiment One

This dealt with the investigation of the six priority rules under the constant due-date assigning procedure suggested by Conway (9). There are two reasons for selecting this procedure.

i) This method compares favorably with the one practiced in real life,

ii) Results of the experiment could be compared with those obtained by Conway (9) under similar experimental conditions.

The constant allowance due-date can be defined as

$$D_i = T_{i,1} + 78.8$$

Conway in his experiment gave an allowance of 78.7985 to every job.

Experiment Two

This experiment was concerned with the testing of the effect of the look ahead rule on the three slack based rules, which, by the previous experiment were found to have a much lower lateness variance - a distinct advantage over the SPT rule.

Each job is given an allowance proportional to the total work to be done on the job under the due-date assigning procedure used in this

experiment. One of the prime reasons for choosing the method was to find ways to reduce the mean lateness under the SPT rule. The mean lateness under the constant allowance method was found to have a high negative value (Table 1).

$$D_i = T_{i,1} + \text{DATE} \sum_{j=1}^{M_i} P_{i,j}$$

where

DATE - Due-date multiplier.

Different values of this due-date multiplier were tested to determine the best method of assigning due-dates. Conway (9) experimented with a multiplier of 9 and found this method to give the best results. The effect of this multiplier on the SLACK rule was investigated. These results are given in Chapter VII.

Experiments with the look ahead rule were found to show no improvement in the schedule (with factor GSLK equal to 0). Under this condition every job with a positive slack was checked and replaced by a late job due to come to the machine under consideration. It was thought that a more selective method of replacement would be more effective, i.e. provide the job under consideration sufficient positive slack so that it would not be affected by the change in schedule. Thus the factor GSLK can be defined as

GSLK - The slack allowance given to a job which is under consideration for a possible replacement by a late job.

Various values of GSLK ranging from 0-15 were tried. This method of looking ahead should certainly be more effective in improving the schedule. The experimental results are discussed in Chapter VII.

1. DESCRIPTION OF THE SYSTEM

The system studied was simply a network of queues, consisting of a set of identical, single-server queues with exponential arrival times and exponential service times. The customers were randomly routed from one server to another.

The number of machines was set at nine throughout the experiment with one machine in each machine group. Previous studies by Conway (9) and Baker and Dzielinski (2) had shown that this would be a sufficient number to exhibit full shop complexity.

The routing of the operations was strictly random. The number of operations per job was obtained from a normal distribution with a mean of 10 and a range from 6 to 15. A job was equally likely to find its first operation on any of the 'm' machines in the shop. A job was allowed to return to a machine but not to have two consecutive operations on the same machine. For reasons of computer storage capacity and calculation speed, the route was truncated to a reasonable upper limit. This upper limit is of course difficult to define and would depend on the core capacity of the computer. Thus, the in-shop inventory and computer storage capacity required for accounting purposes is reduced and a maximum number of jobs can be accomodated at any given time.

The jobs were released to the shop one at a time, in the order

generated. For Experiment One the interarrival times were obtained from an exponential distribution whose mean was set to yield a nominal utilization of 90 percent in the shop. For Experiment Two the interarrival time mean was kept constant for all the runs.

Runs differed in that different priority rules were used to select jobs from queue whenever a machine became available for assignment. The simulator always selected, from among those in the queue, the job with minimum priority value. Each run was presented with an identical set of jobs and the aggregate performance recorded. In each case, however, recording did not begin until the process had been running long enough to at least approach steady-state conditions.

2. INPUT DATA

The experiment was designed so that an identical sequence of job arrivals was available for each run. This was accomplished by means of subroutine RANJOB (Figure 3 in Chapter VI), which provided the following information, each time an arrival was scheduled.

i) The arrival time of the job;

ii) The number of operations per job;

iii) The machine group necessary for each operation;

iv) The processing time for each operation;

v) The machine group necessary for subsequent operation;

vi) The sum of processing time remaining, and

vii) The due-date for the job.

Interarrival and Processing Times

The interarrival times were obtained from an exponential distribution

whose mean was set to yield a nominal utilization of 90 percent in the shop. A minimum value of 0.5 time units between arrivals and a maximum of 10.0 time units were specified. This upper truncation did not change the intended distribution as indicated by the interarrival times for 1200 customer arrivals in Figure 1.

A Chi-square test was performed for goodness of fit for the inter-arrival distribution. The results obtained were:

For 95% confidence level and 18 d.f. the critical value of X^2 = 28.9

X^2 obtained for distribution = 24.058

Since chance alone will give a value of X^2 = 28.9, 95 times out of 100 the distribution can be safely assumed to be exponential.

The processing-time for each machine was identical; an exponential distribution with a mean of one. The minimum and maximum values were set at 0.5 and 10.0 time units respectively. The frequency distribution of processing-times for 1400 jobs is shown in Figure 2.

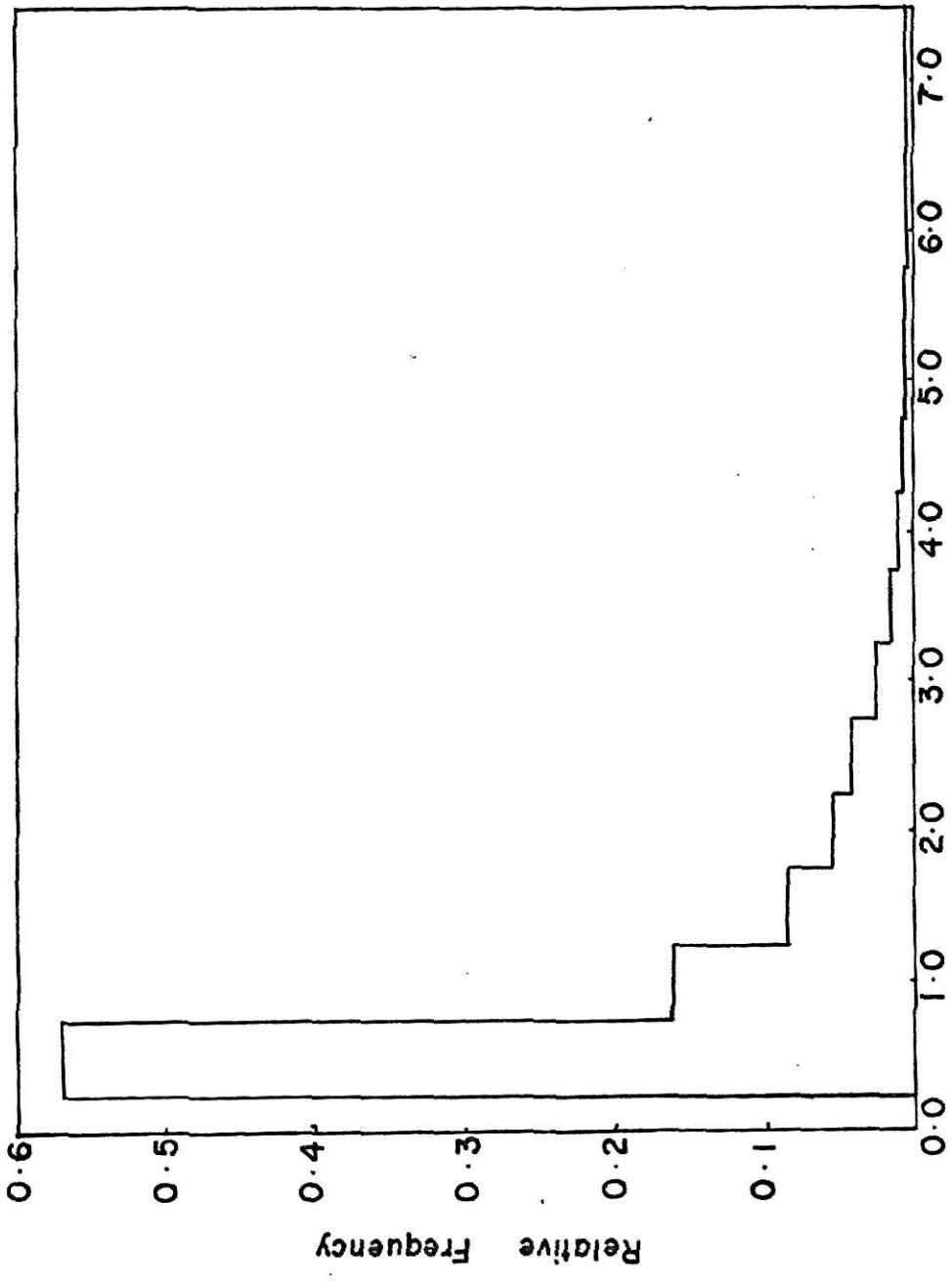


Fig. 1. Frequency Distribution of Interarrival Times.

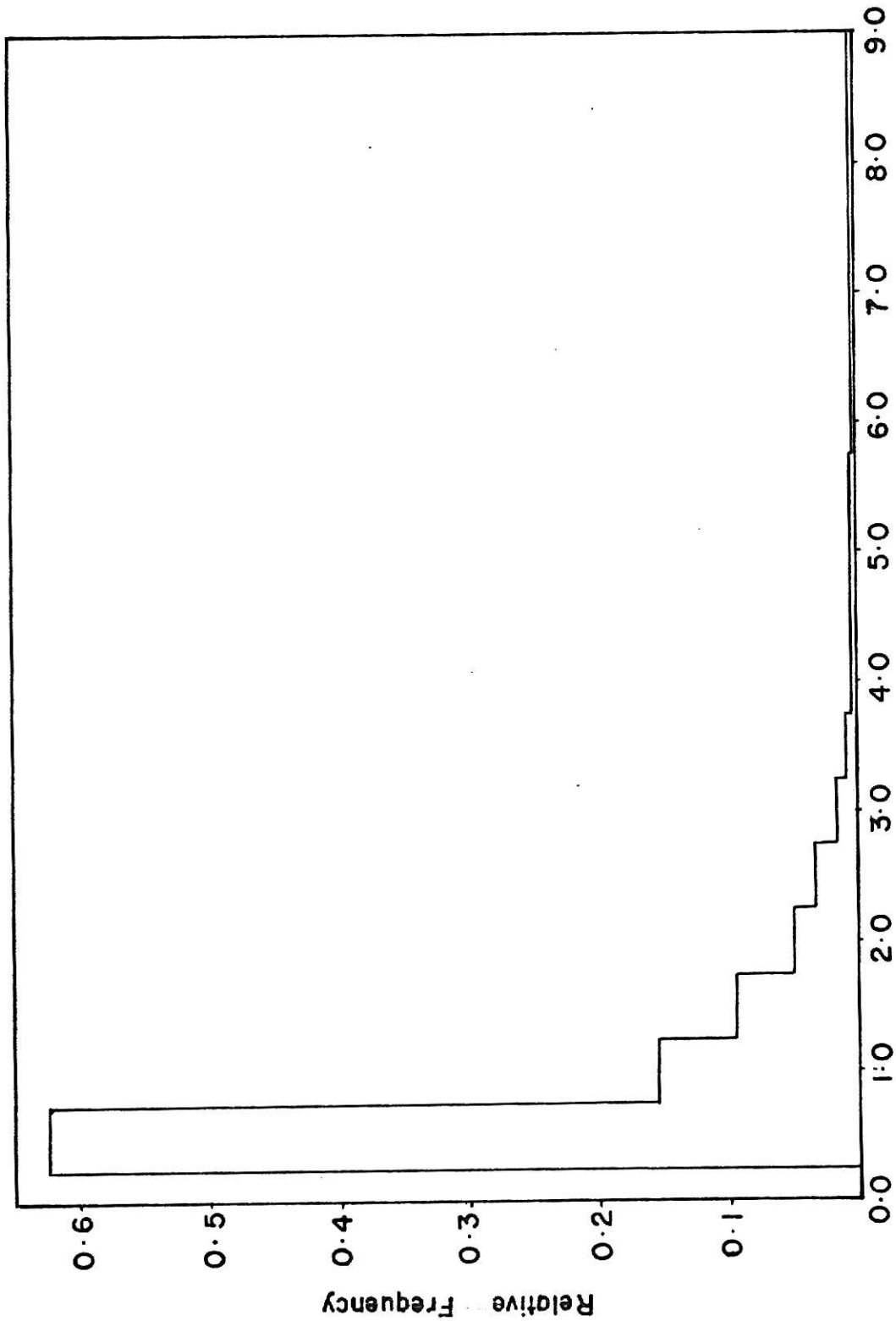


Fig.2. Frequency Distribution of Processing Times.

CHAPTER VI

THE SIMULATION

GASP II - General Activity Simulation Program was used in the analysis of the job-shop. The version used in this work was developed at Arizona State University by Pritsker and Kiviat (32).

Variations in the formulation of the packaged program were required to expand the network capacity capability of the simulator. The simulation was accomplished using an IBM 360/50 computer.

GASP II - is a collection of subroutines and functions written in FORTRAN and expressly designed for use in simulation studies of industrial systems. The principle advantages offered by GASP are its machine-independence and its modular characteristics, which make it quite easy to expand and alter simulation programs to suit the needs of any given system. GASP has been designed to facilitate 'next event' types of simulations. In such simulations, simulated time progresses from one event to another until an end of simulation event occurs, or a preplanned total simulation time is exceeded.

1. THE SIMULATION MODEL

The specific events of the job-shop simulation were,

a) Arrival of the job at the shop; and

b) End of processing for a job.

The model was designed so that a special event for job departures would be unnecessary. This was accomplished by means of the variable JOP(JOB) which stored the number of operations remaining for each job (JOB). When the

last operation of a job was completed, JOP(JOB) = 0; therefore, statistics were calculated and the job was allowed to exit the system.

The two entities in the system are machine groups and job/operations. As a permanent entity, each machine group must be considered separately and its current status available at any time. Operations are classified as temporary entities and as such must be associated with enough information to maintain their integrity. This was accomplished by defining various job/operation attributes which follow a particular job/operation until its processing has been completed and statistics gathered.

Initially eleven files were set up. The first being the event file: files 2-10 being queues for each of the nine machine groups and file eleven was set up as a storage for future operations on all jobs in the shop.

Whenever any operation on a particular machine was completed, subroutine FIND was called. This subroutine picked the next operation for the particular job. Subroutine REMOVE would then remove this operation from file eleven and, depending on the availability of a machine in the required machine group, an end of processing event would be scheduled or the job/operation was placed in the machine group queue for service.

On an average, each job has 10 operations. Of these, the first is placed in the event file and the rest in file eleven. Subroutine FIND begins the search by comparing the value of the required attribute, starting from the first entry in the file and continues down the line until it hits the required entry. This procedure is repeated each time a job/operation is completed and the next job/operation has to be removed from file eleven.

This difficulty was overcome by increasing the number of files from eleven to nineteen. The first ten performed the operations mentioned previously. Files 11-19 may be considered as schedule charts for the nine machine groups which indicated the jobs scheduled for operation on the particular machine group at some future time. This necessitated an increase in the number of attributes from ten to eleven. Attribute eleven indicated the next machine group required, thus providing the information necessary to find the next job/operation. With this new setup a 75% reduction in computing time was achieved. A sample of the queue printout is given in Appendix A.

The eleven attributes specified for each job/operation are,

Attribute (1) - Scheduled time of next event

Attribute (2) - Event code

Event Code = 1 - Arrival of job (subroutine ARIVAL)

Event Code = 2 - End of processing (subroutine ENPROS)

Attribute (3) - Arrival time of job/operation

Attribute (4) - Machine group number

Attribute (5) - Processing-time/operation

Attribute (6) - Operation number

Attribute (7) - Job number

Attribute (8) - Due-date

Attribute (9) - Sum of processing-times for remaining operations

Attribute (10) - Number of operations

Attribute (11) - Next machine group requirement

These attributes were the same for all files. File 1 was ranked on attribute 1 on the basis of FIFO (first in-first out). Files 11-19 were ranked on attribute 7 while the ranking attribute of the remaining files was dependent on the priority rule being investigated. All priority rules were ranked on attribute 8 on the basis of FIFO, except for the following,

i) FCFS was ranked on attribute 1.

ii) SPT was ranked on attribute 5.

Logical flow diagrams of the main program and seven subroutines are presented in Figures 3 - 9. A listing of these eight programs and subroutines DATAIN and SUMARY from GASP II, which were modified is given in Appendix B, in addition to a description of the non-GASP variables. Description and listing of the GASP II program can be found in Pritsker et al. (32).

The priority assignment procedure was done by means of subroutine SELECT, that assigned a numeric value to an attribute named priority, or priority index, for each operation of each job. For priority rules 3, 4, and 5, the effect of the look ahead rule was investigated. Subroutine CHECK was called whenever the slack of the selected job/operation was positive. If some late job was scheduled to arrive at the machine group in consideration, then the selected job/operation was replaced with the late job/operation.

The priority rules were identified by means of the variable KODE:

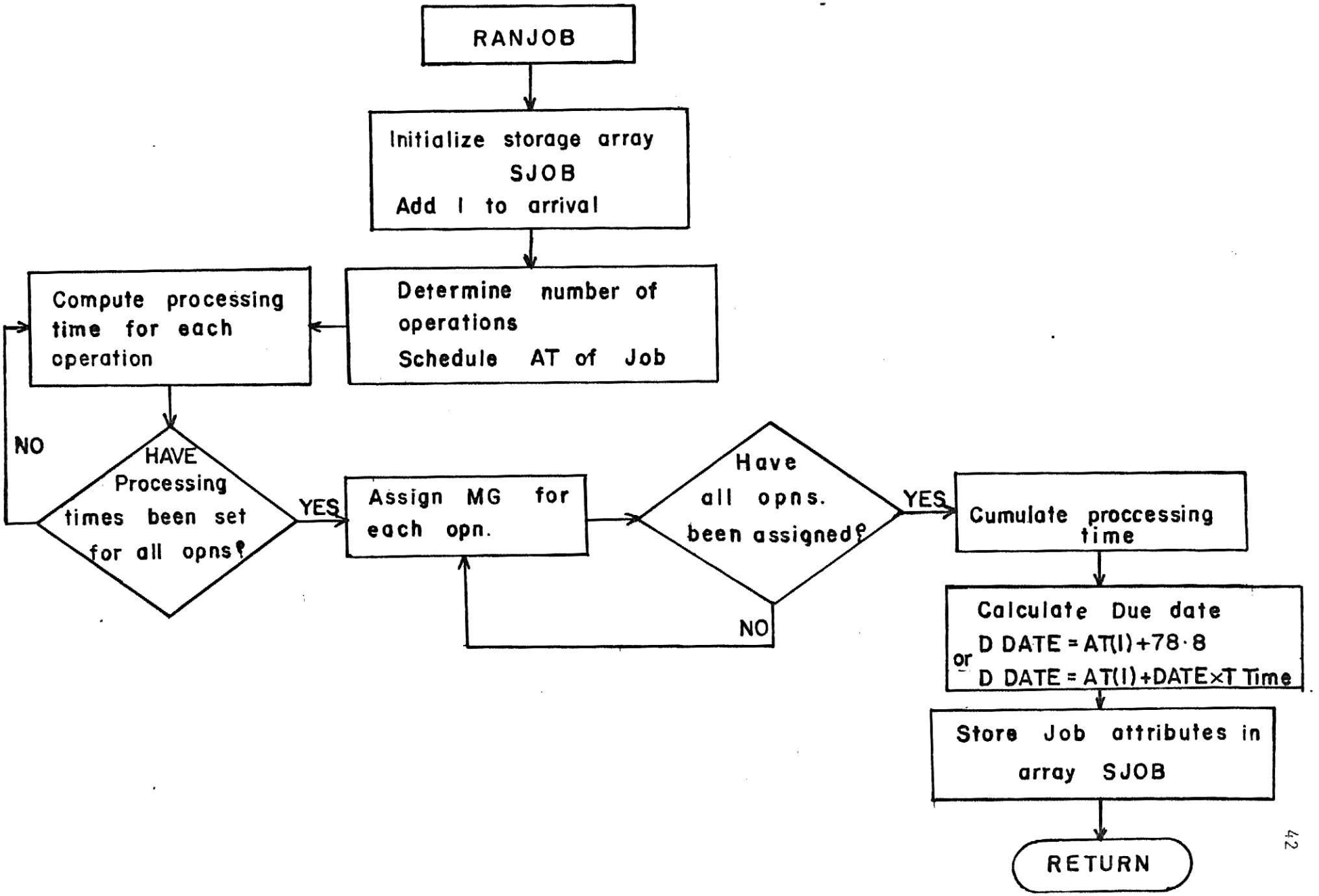


Fig.3. Subroutine RANJOB

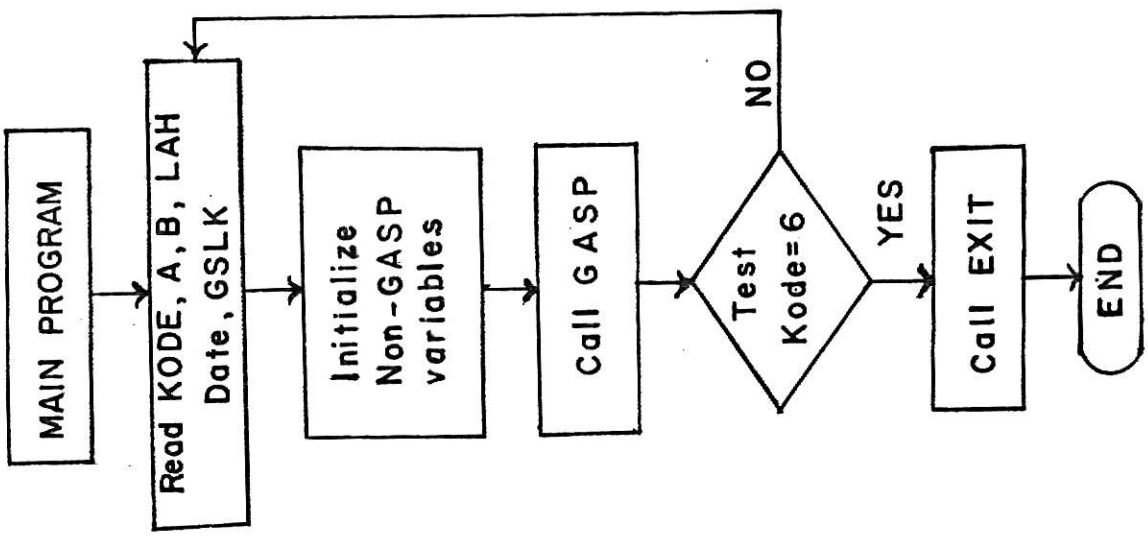


Fig.4. MAIN PROGRAM

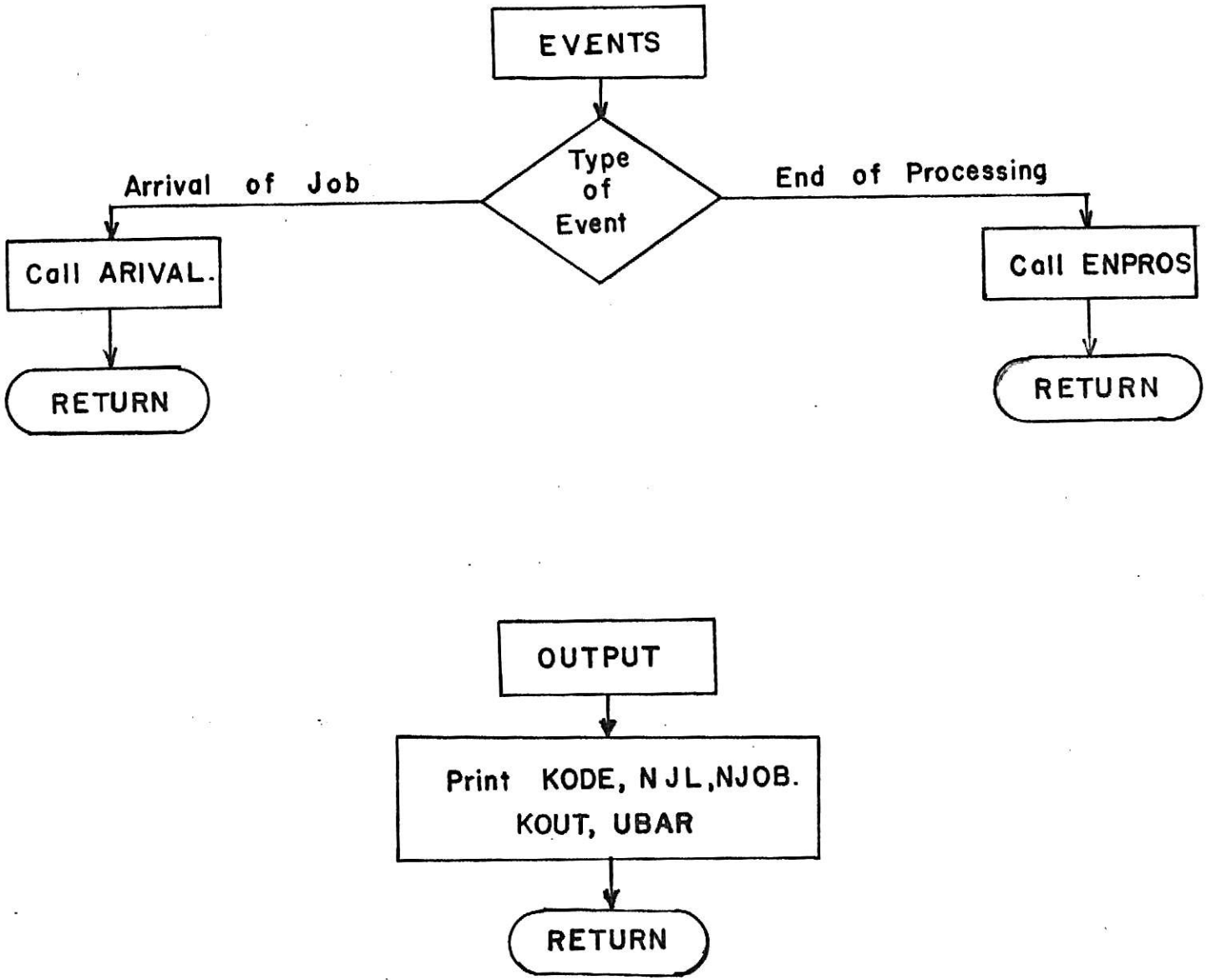


Fig.5. Subroutines EVENTS and OUTPUT.

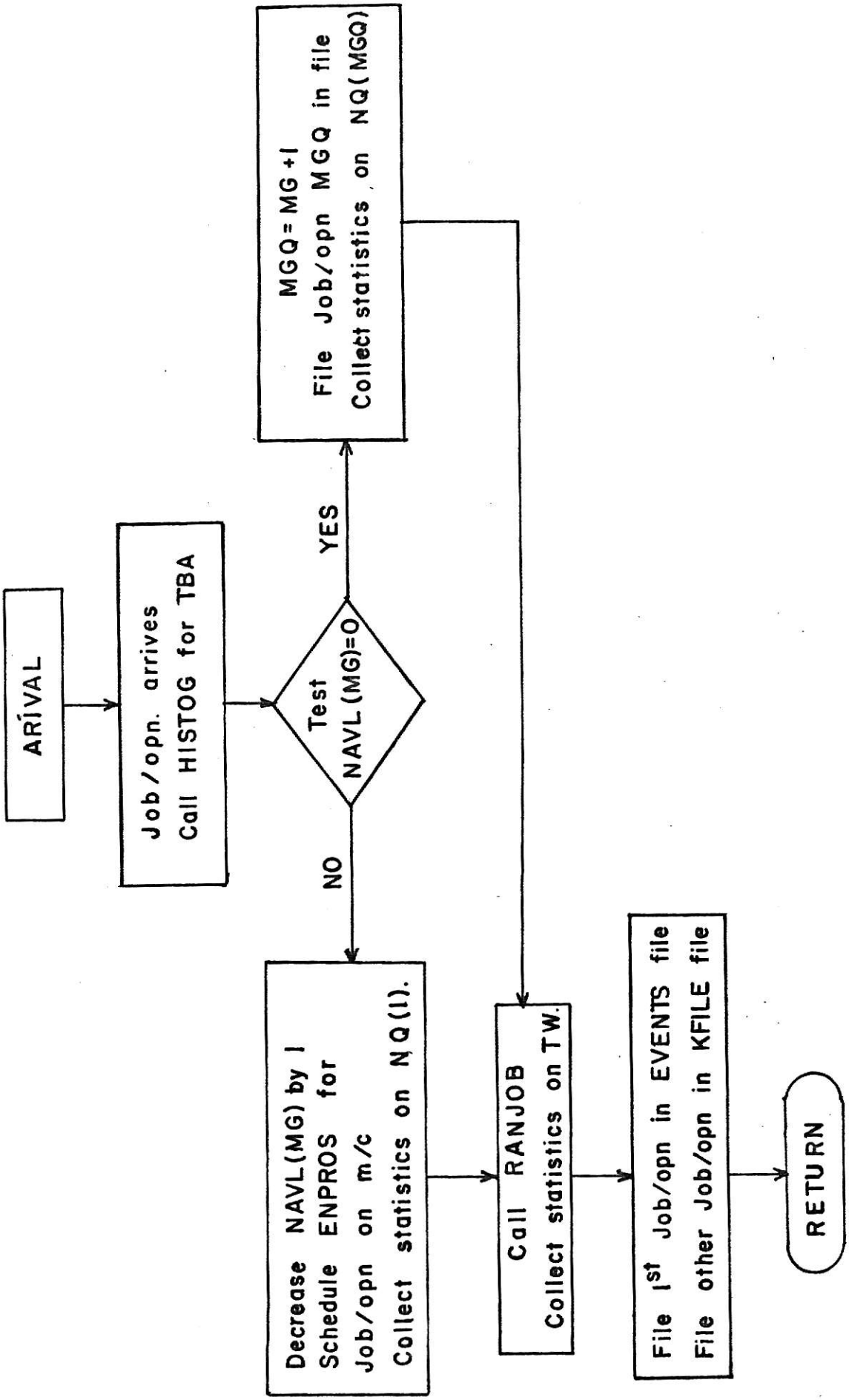


Fig.6 . Subroutine ARIVAL.

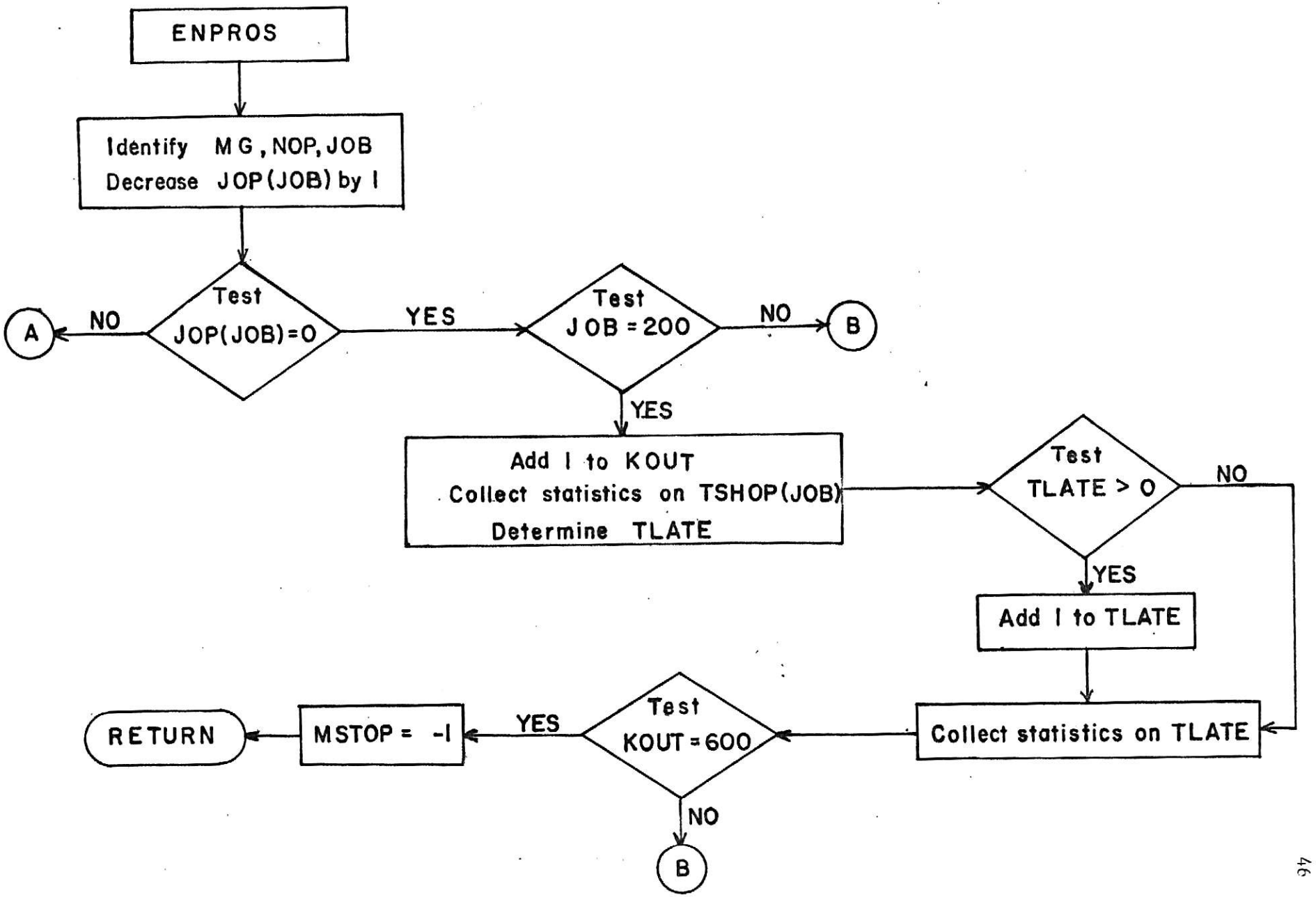


Fig. 7. Subroutine EPROS

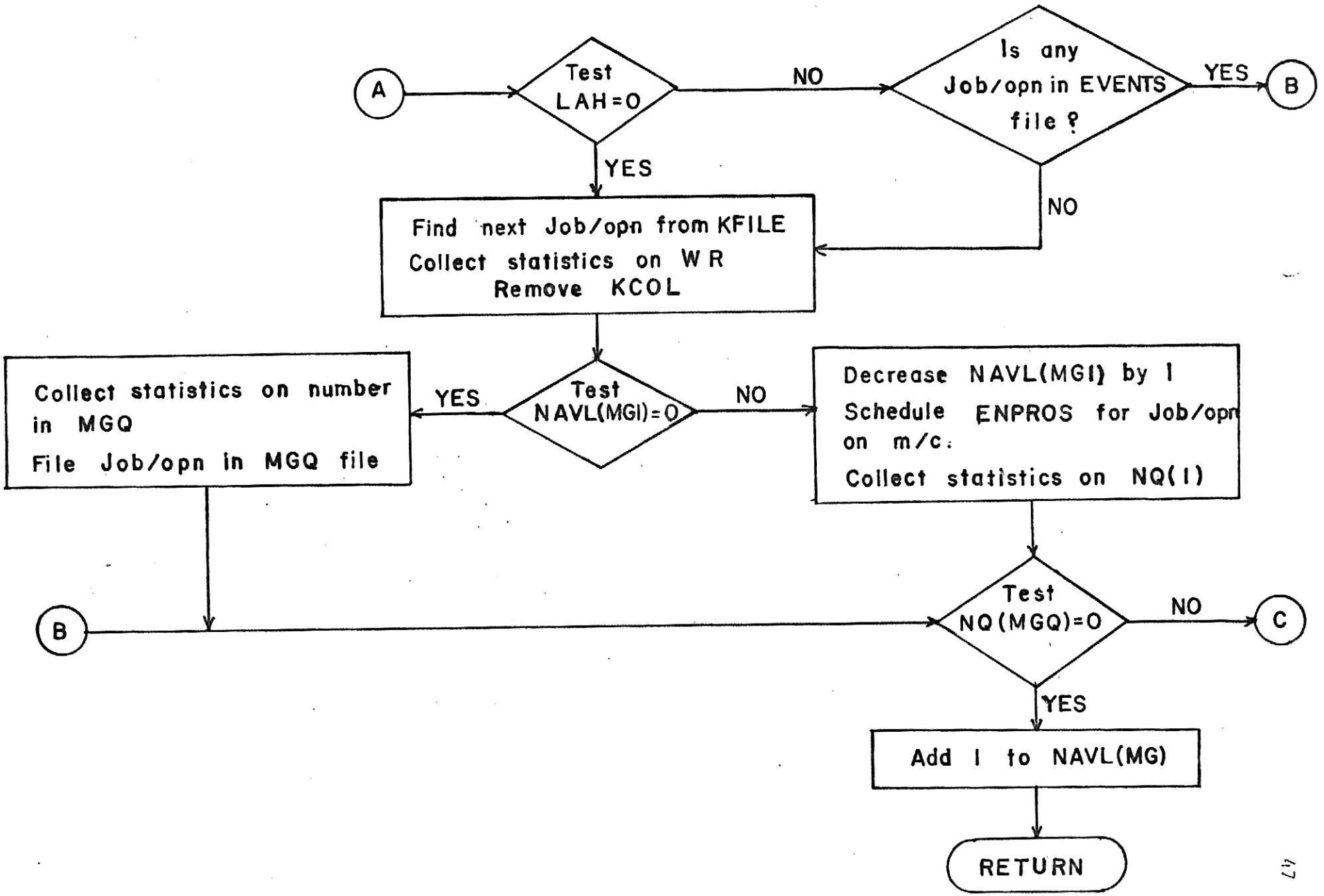


Fig. 7. Subroutine ENPROS (contd.)

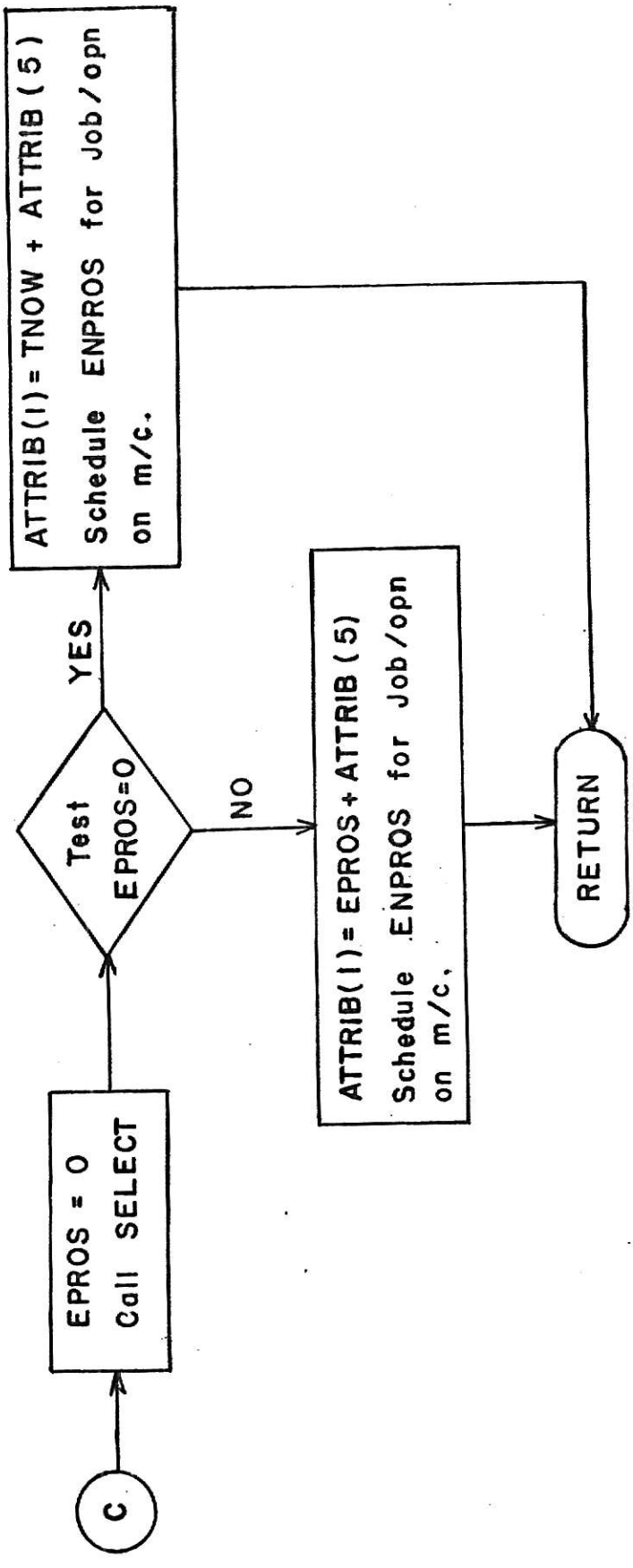


Fig.7. Subroutine ENPROS (contd.)

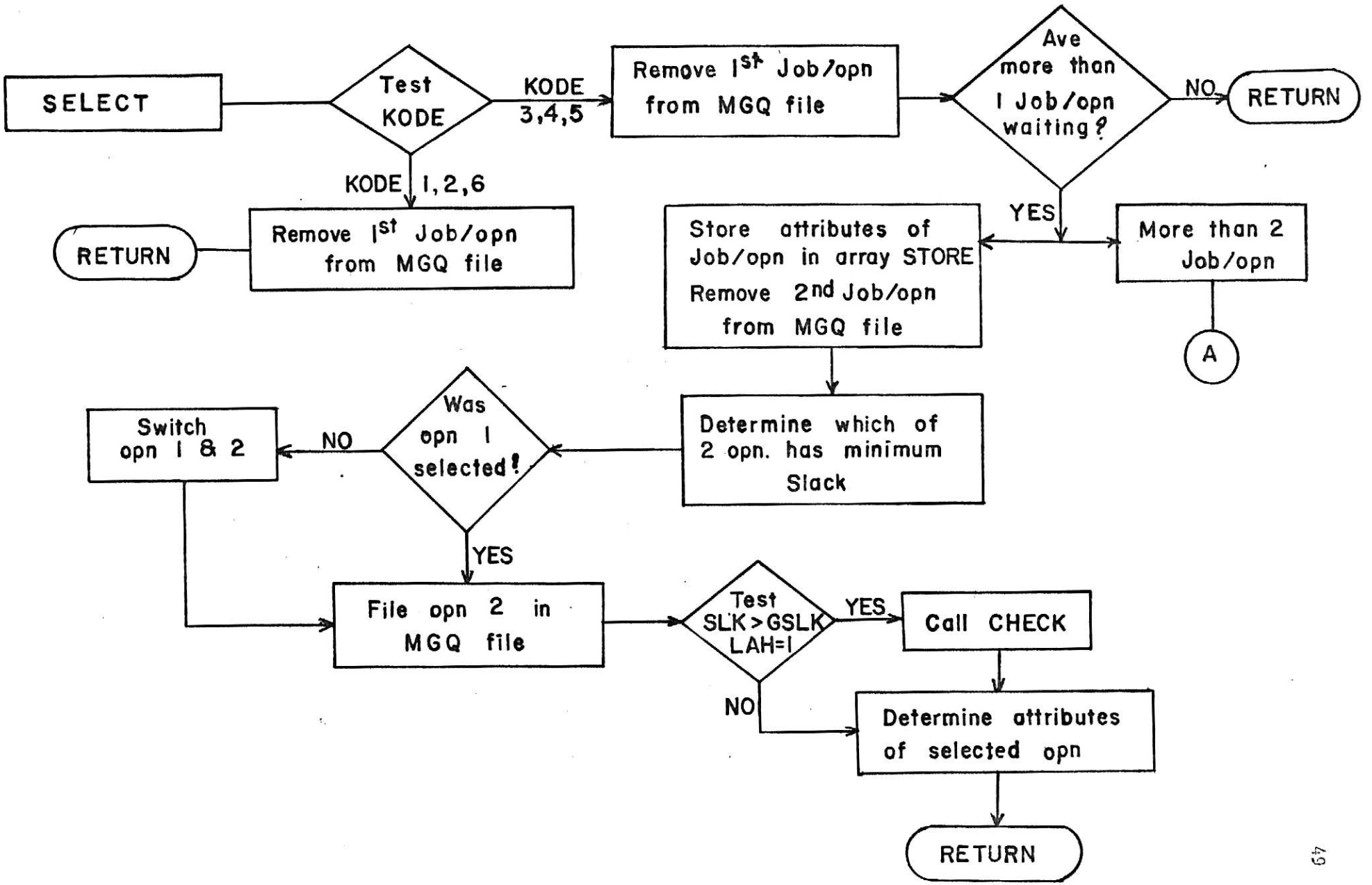


Fig.8. Subroutine SELECT

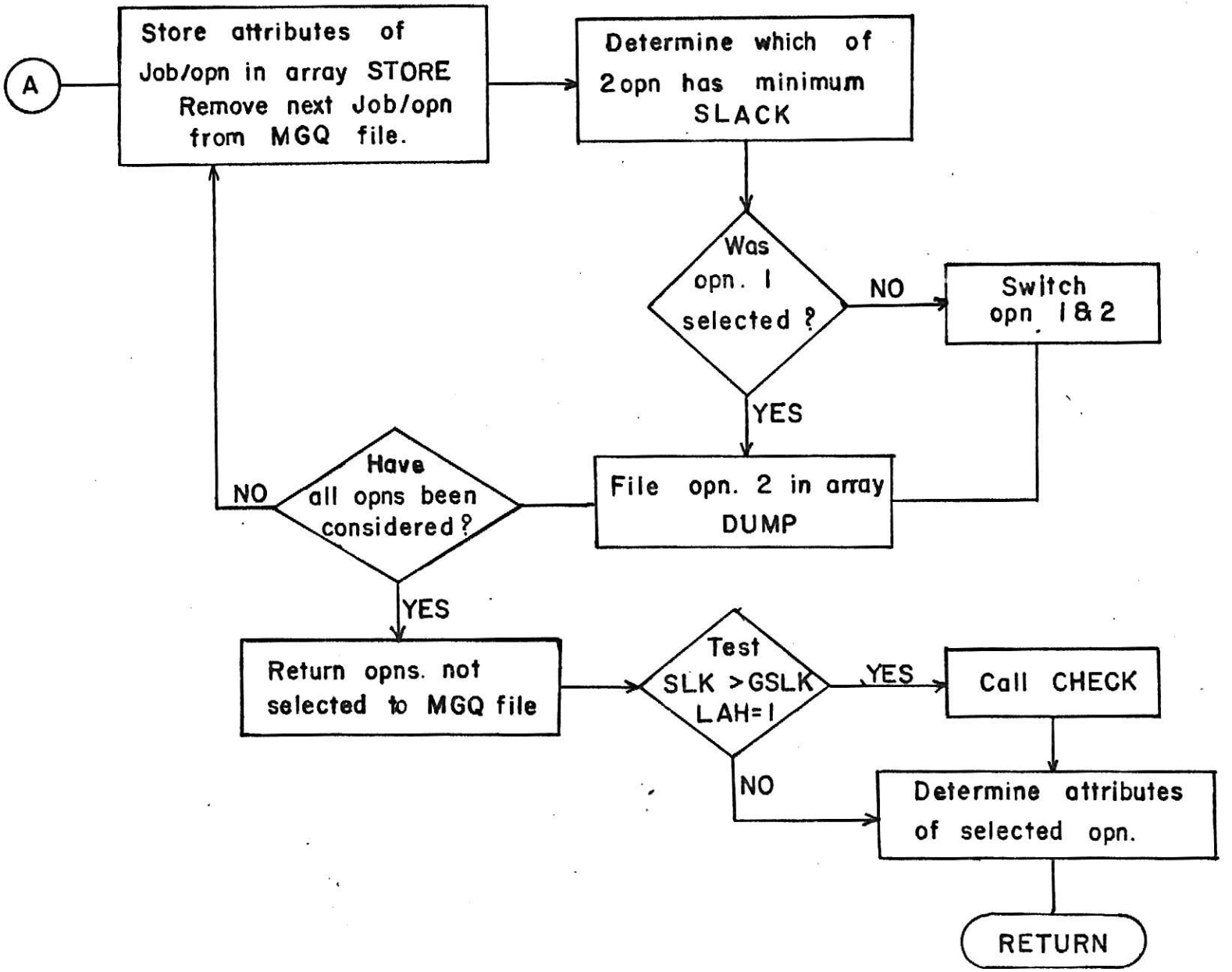


Fig. 8. Subroutine SELECT (contd)

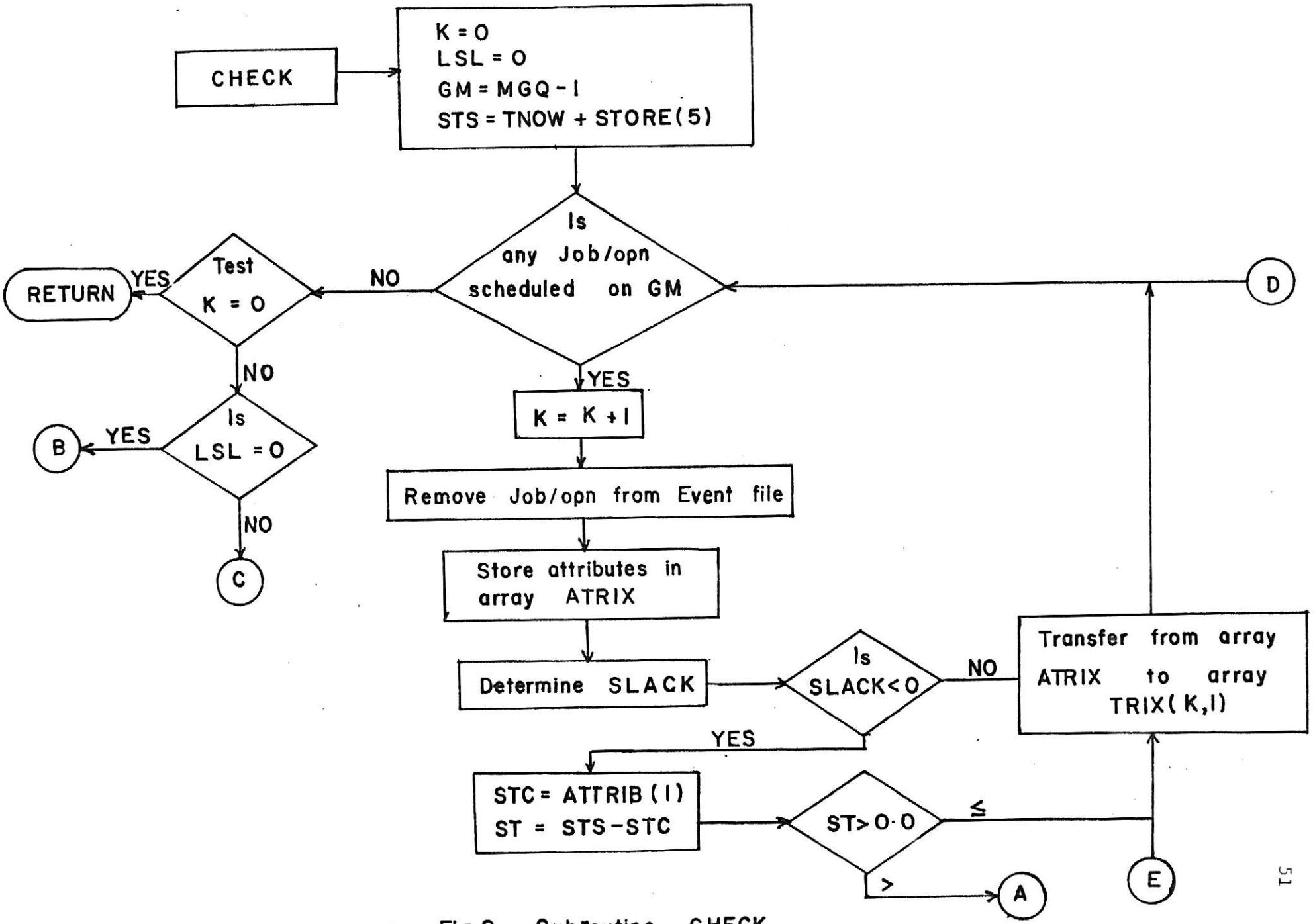


Fig.9. Subroutine CHECK

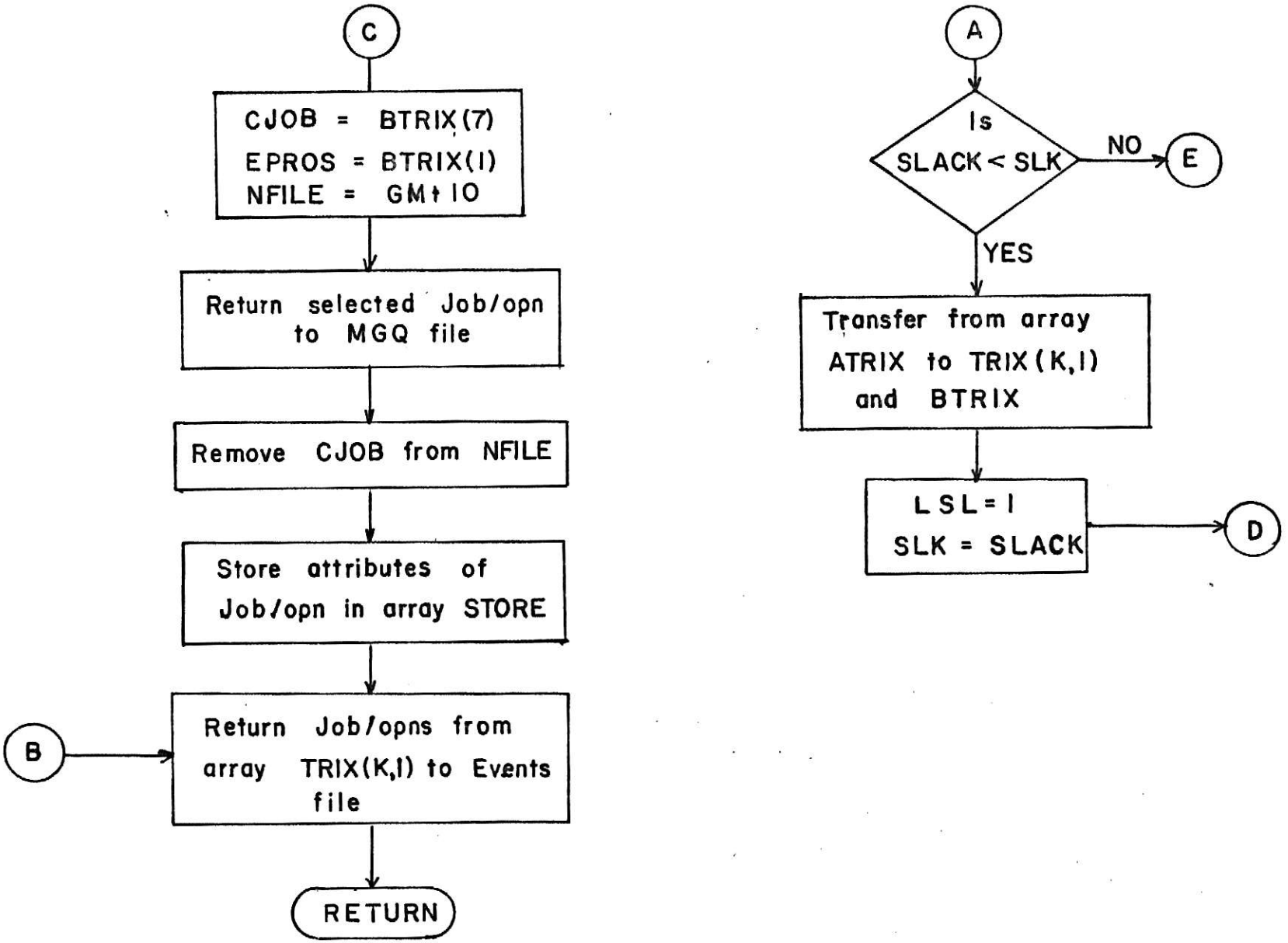


Fig.9. Subroutine CHECK (contd.)

KODE	PRIORITY RULE
1	FCFS
2	SPT
3	SLACK
4	S/OPN
5	S/RT
6	DDATE

The mean shop utilization was calculated using the formula given in Chapter IV.

$$\bar{U} = \frac{\bar{W}\bar{N}}{mS}$$

2. CONDITIONS FOR THE EXPERIMENTAL RUNS

A simulation run represents the operation of a system from a given starting point for a period of time. Starting conditions, therefore, are relevant to results obtained from the model. To a certain extent, the problem of initial conditions can be viewed as a problem of sample size in that a long enough run, which is equivalent in simulation studies to taking a large sample, can usually obliterate the effects of starting conditions.

After a few preliminary runs it was determined that a run-in of 100 jobs was sufficient to set the initial shop conditions. There is no

set procedure to determine when a simulation process has reached steady state. One way is to determine when the absolute difference in mean interarrival and interdeparture times have decreased sufficiently (14). Figure 10 for the SPT rule shows that the difference stabilizes with the arrival of the 150th job. The collection of statistics began after the steady state had been reached, i.e., with the arrival of the 101st job Experiment One and the 201st job for Experiment Two.

The rapid convergence of the inter-departure curve is due to the pre-loading. For both the experiments, the shop was pre-loaded by having the first 30 jobs arrive simultaneously at the beginning of each run. Two separate methods were used to determine the inter-arrival times - one method taking into account the pre-load jobs, the other based on arrivals after time zero.

As seen from Figure 10, the inter-arrival curve stabilizes much faster when the pre-load jobs are not considered. The inter-arrival curve with and without the pre-load jobs stabilizes after the arrival of 250 and 150 jobs respectively. Frequency distribution of interdeparture times is shown in Figure 11.

Every experimental run consisted of the release of 750 jobs for Experiment One and 600 jobs for Experiment Two, with interarrival intervals obtained from an exponential distribution. The same conditions and procedures were used throughout a run. The only difference between runs was in the choice of priority rule used.

Experiment One consisted of 6 runs. A run required anywhere from 25 to 50 minutes of computer (IBM 360/50) time, depending primarily on the selection of priority assignment procedure.

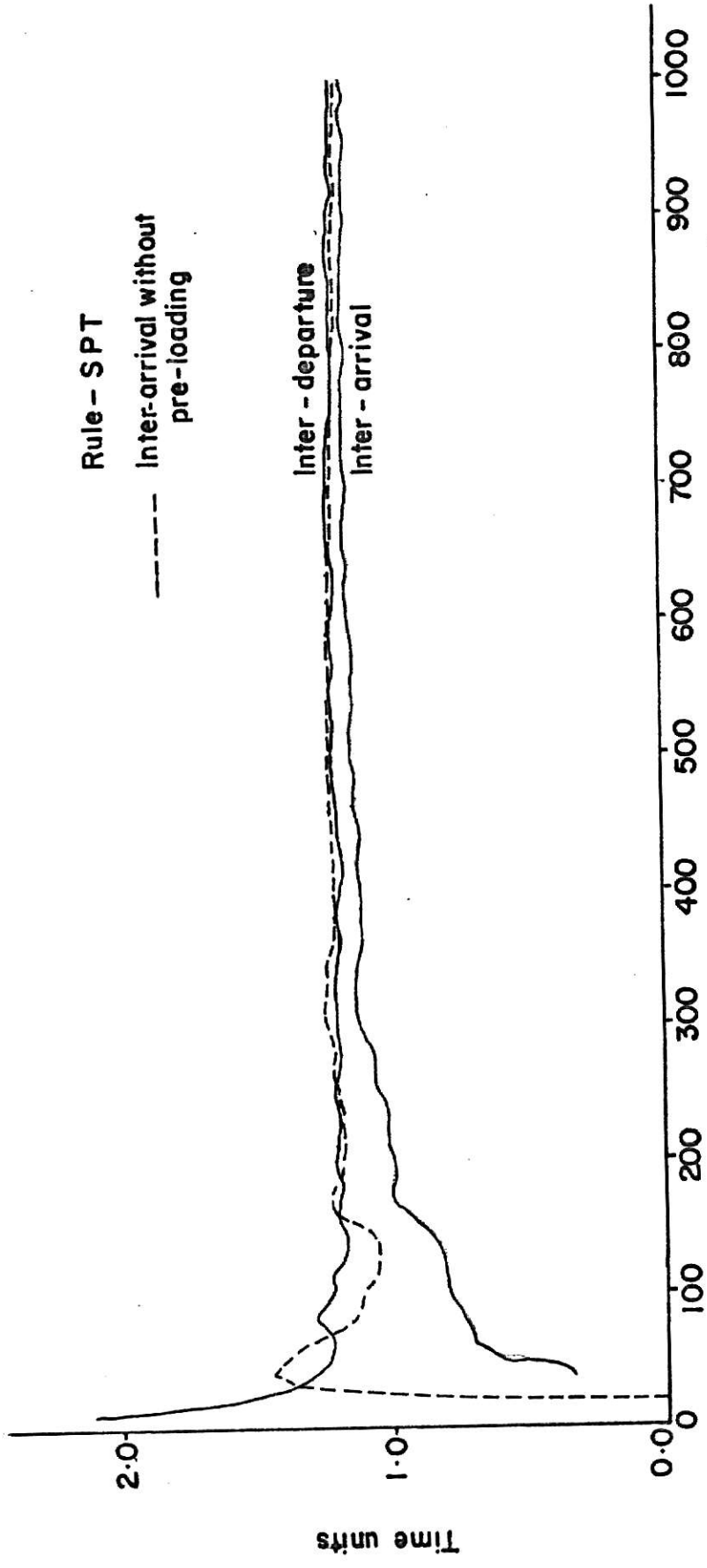


Fig.10. Convergence of Inter-arrival and Inter-departure times.

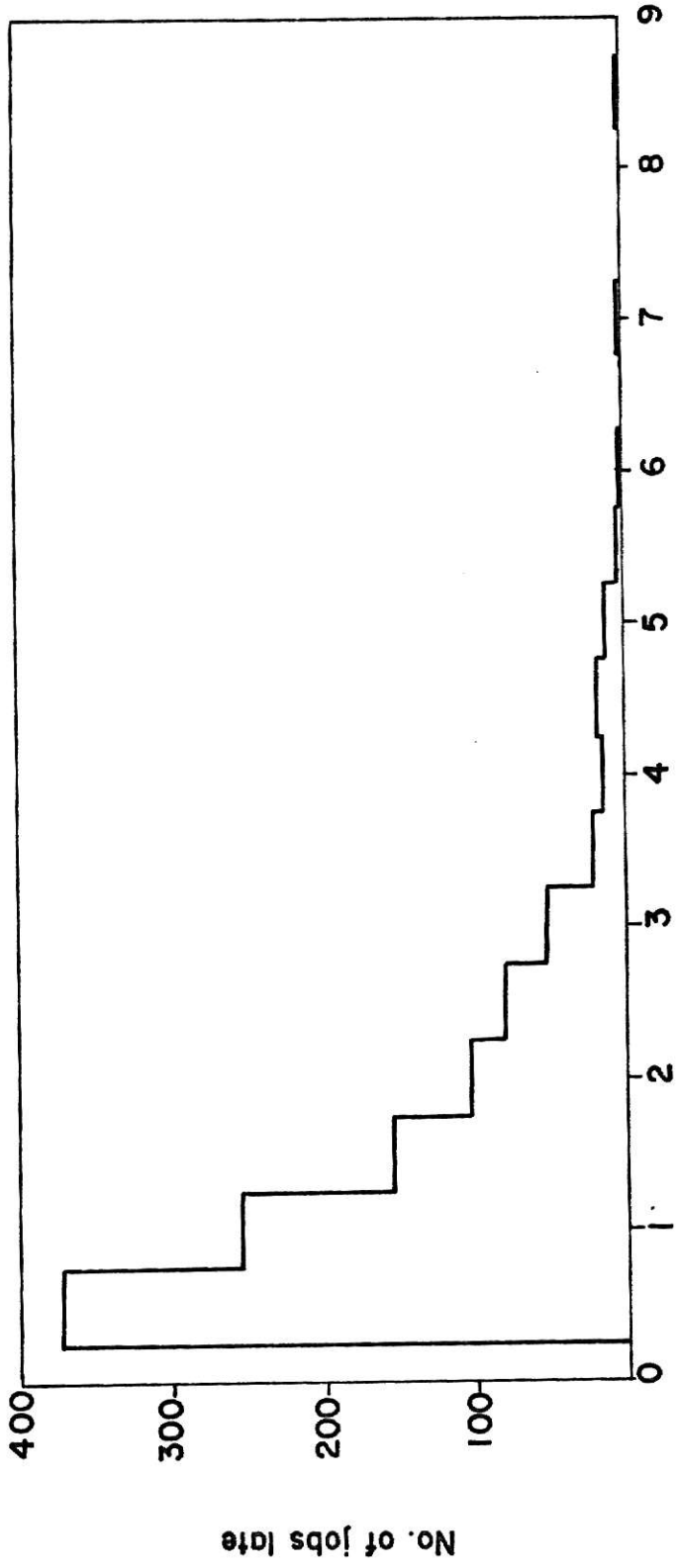


Fig. 11. Freqency distribution of inter–departure times.

A 20% reduction in computer time was achieved in Experiment Two by running the program in Fortran H-Level. Initial difficulties encountered in switching to H-Level were due to compiler problems.

One of the more important problems in simulation is concerned with determining what measurements to make and when. This depends to a large extent on the type of system being studied. A study may concern a phenomenon of definite and limited duration, such as the peak-traffic period of a passenger terminal. In most cases, the study is concerned with a process which operates more or less continuously and has no natural duration, for e.g., the study of a job-shop process. In such a case a comparison of two alternatives should be based upon measurements obtained from steady-state operation of each alternative (6).

Job statistics - total time in the shop, and lateness; and shop statistics - number of jobs in queue and other measures of work-in-process did not begin until the arrival of the 101st job. These statistics were collected for sets of 100 jobs. The sets were defined based on the arrival order of the jobs. When a job was completed and left the shop, its attributes of time-in-shop and lateness were posted to accumulators for its particular 100 job set. The jobs always arrived in the same order, but their order of completion depending on the priority assignment procedure used, so it was never the same as the arrival order and not the same for any two runs.

The following list refers to a code number under which the specific statistic is identified in the Gasp Summary Report of Appendix A.

Generated Data

1 Lateness for all jobs completed

2-15 Lateness for individual sets of 100 jobs completed

16 Total time in shop for all jobs completed

17-30 Total time in shop for individual sets of 100 jobs completed

Time Generated Data

1-10 Status of queues 1-10 with respect to time

11 Mean work content (sum of processing-times) of the jobs.

Frequency distributions were recorded, using the subroutine HISTOGRAM.

Generated Frequency Distributions

1. Interarrival times
2. Processing-times
3. Lateness
4. Shop flow times
5. Interdeparture times

The following queue statistics are an automatic output of the simulation package.

i) Average number of entries in the queue with respect to time

ii) Maximum number of entries in the queue; and

iii) Status of the queue at the end of the simulation.

Samples of the various statistics, histograms and queue printouts are given in Appendix A.

CHAPTER VII

RESULTS

The results for both Experiment One and Experiment Two will be discussed in this Chapter. One of the prime purposes of this research is to study the effectiveness of GASP II in dealing with large, complex systems and, as such, the job-shop model, priority rules, and other experimental conditions used were based on the works of Gere (15) and Conway (9). This was done to facilitate a comparison of the results of the simulation.

1. EXPERIMENT ONE

The two important measures of performance used were the minimization of work-in-process inventory and job lateness.

A discussion of the performance of the various priority rules is given below:

1. Minimization of Work-In-Process Inventory.

As stated previously in Chapter II, there are various ways of measuring work-in-process inventory. In this work, only one of these measures i.e., the number of jobs was considered. It is apparent from the relationship given in Chapter IV that a scheduling procedure that minimizes the mean number of jobs in queue would also minimize the mean number of jobs in the shop, the mean time spent in the shop by the jobs, and the mean lateness (9).

The experimental results are given in Table 1. The superiority of the shortest processing-time rule as demonstrated in previous works (2, 9, 10, 11) is obvious from the results. The mean number of jobs in

the shop under the SPT rule was 25.51. This was almost one-third of that under the FCFS rule, which had a mean number of jobs in queue of 72.37. The mean shop-time and mean lateness for the SPT rule too were reduced by the same ratio as seen from Table 1. Of the slack rules only S/RT rule was successful in reducing the mean shop-time and mean lateness although the mean number of jobs in queue was still very high - 52.47 as compared to 25.51 for the SPT rule.

The performance of the SPT rule could further be improved if some way could be devised by which jobs with longer processing-times could be given preference. The other alternative is to devise some means for reducing the queue lengths with the slack based rules and thus take advantage of their ability to reduce both mean shop-time and lateness. An effort to do this was made using the look ahead rule. These results are discussed under Experiment Two.

All the rules tested surpassed the performance of the FCFS rule. The SPT rule required the minimum amount of computer time. Further tests were made with the SPT rule in order to observe its performance at different levels of utilization. Results of these runs are given in Table 3.

Finally, the results, in general, seem to be in agreement with those obtained by Conway (9). A summary of Conway's results is given in Table 4.

2. Job Lateness

The two measures of principle interest were the total time the job spent in the shop and the lateness. Lateness is defined as the difference between the actual and allowable shop-time and can take on negative values, indicating that the job was completed in less than the allowed time.

Four due-date based rules were used. The due-dates were assigned the job at the time of their arrival in the shop. The constant method of assigning due-date was used.

The shop-time and lateness results for the six rules tested are given in Table 1. The shortest processing-time (SPT) rule, which reduced the mean number of jobs in the shop also reduces the mean shop-time. A reduction of almost two-thirds in mean shop-time over the FCFS rule was achieved using the SPT rule. The Std. dev. of shop-time and lateness was found to be less than that for FCFS rule and was only surpassed by the slack based and DDATE rules.

One of the important considerations of any shop is to get the job out at the earliest or by its due-date. To achieve this, it is necessary to find means for setting reasonable due-dates. A job is usually delayed in between operations. The only means of estimating the time spent by the job waiting in queues is the number of operations. Conway studied four rules which used the information of either the number of operations remaining or the amount of work remaining. These were:

FOPNR - Fewest operations remaining

MOPNR - Most operations remaining

LWKR - Least work remaining

MWKR - Most work remaining

FOPNR and LWKR did perform better than the FCFS rule, but, at the cost of a very high lateness and shop time variance.

The SPT rule, on the other hand, uses the information of the number

of operations and their processing-times, and, as such has a high predictive value. This is obvious from the results of Table 1.

Rules using due-date information were not able to perform as well as the SPT rule with the exception of the slack ratio (S/RT) rule, under which the lateness is only 12.6%. The S/RT rule performed far better than the FCFS rule and reduced the lateness by almost 80%. It also has a higher predictive value as seen from the low standard deviation in shop time and lateness.

The type of due-date assigning procedure used has a considerable impact on the performance of these rules, as is seen from results of Experiment Two. Thus the performance of these rules with more realistic due-date assigning procedures should be investigated. The number of jobs late under the six rules tested are given in Table 2, and the frequency distributions of lateness are shown in Figures 12-17. SPT rule, which does not consider the due-date in its operation, gives the best performance.

In spite of the poor performance of rules using due-date information, it should be noted that the lateness variance was still lower than either under FCFS or SPT rules. This is an important characteristic, and further investigation should be carried out to find ways of reducing mean lateness.

The SPT rule, using only the information of processing-time, was found to make only 10% of the jobs late. One drawback of this rule, however, is that jobs with high processing-times are found to suffer. A great majority of jobs were completed well in advance of the due-date while a small number of jobs had a very high lateness (Figure 13). It

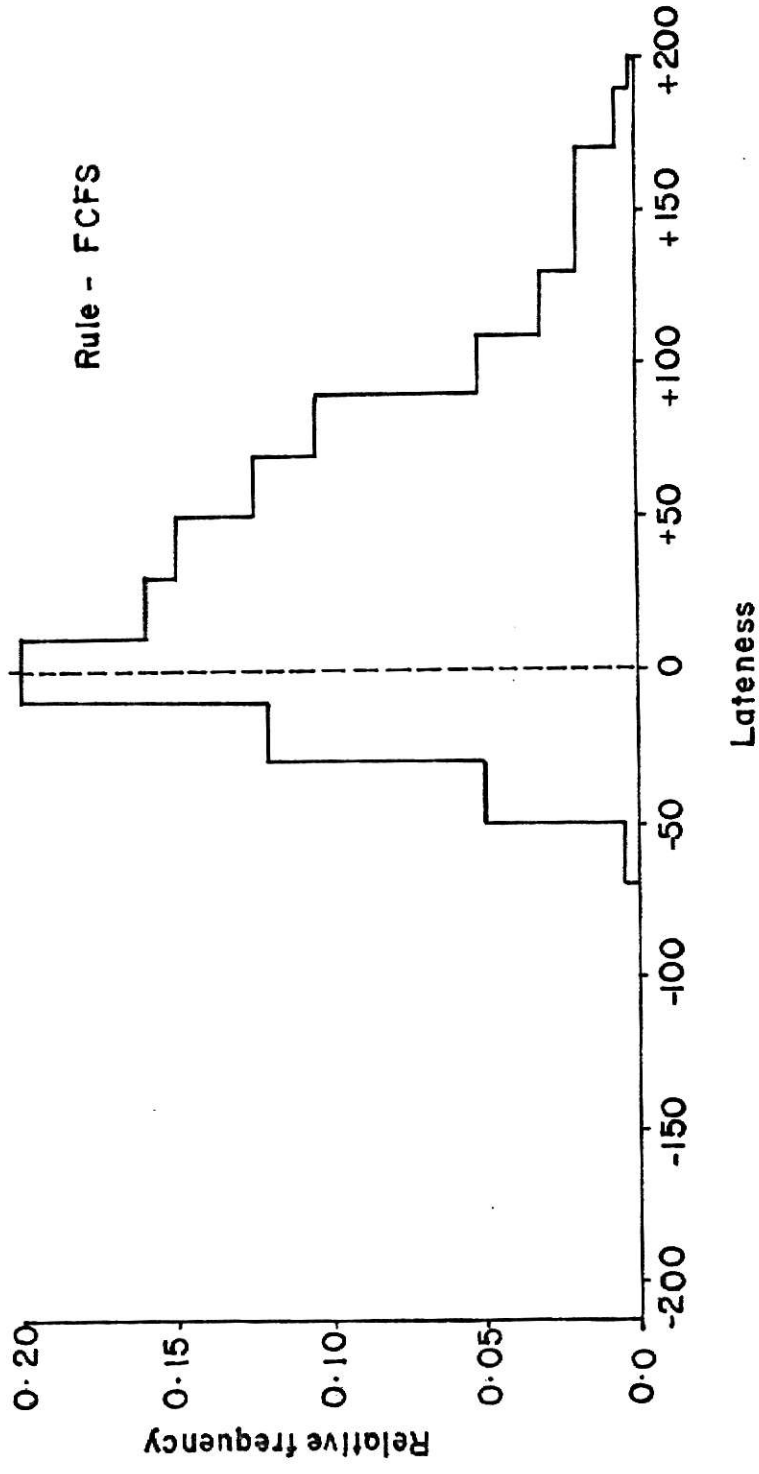


Fig.12. Frequency disribution of lateness.

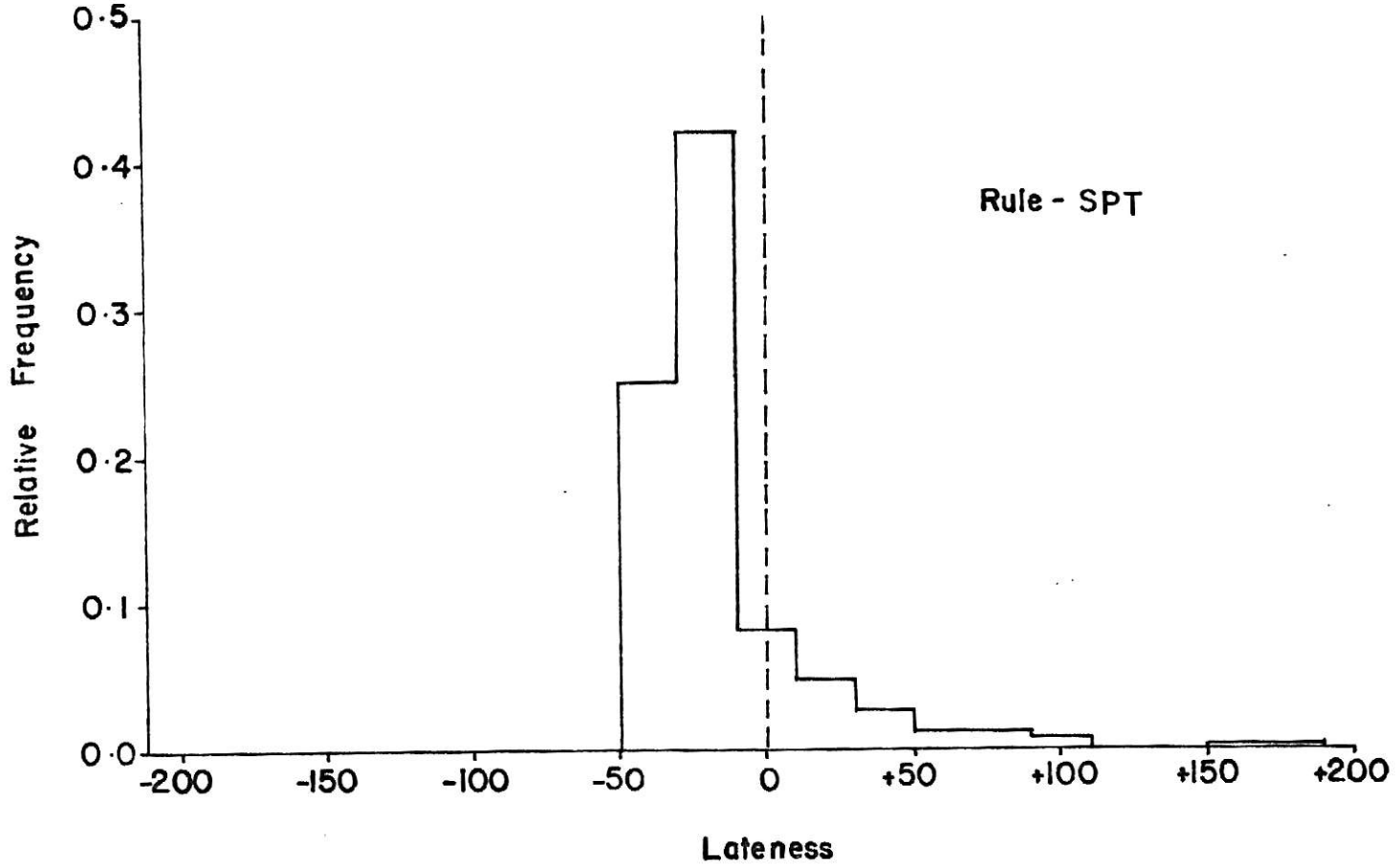


Fig. 13. Frequency distribution of lateness.

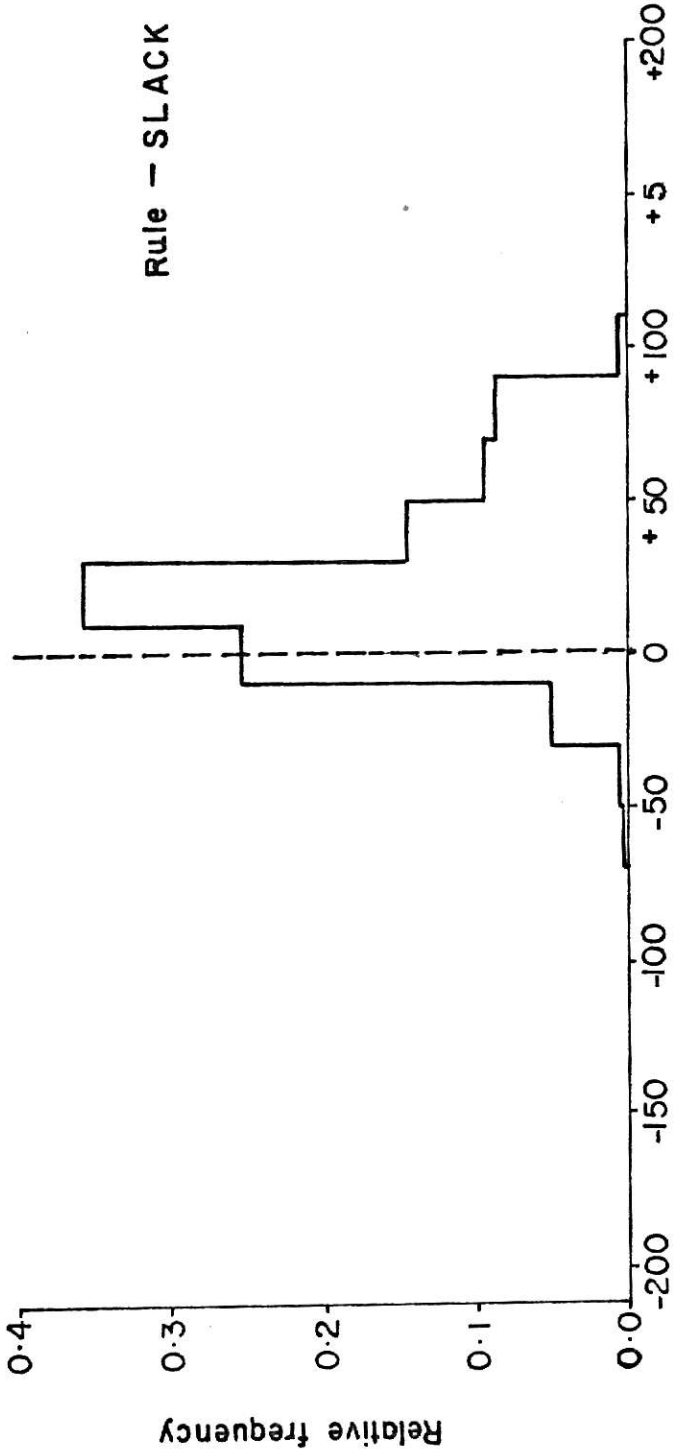


Fig.14. Frequency distribution of lateness.

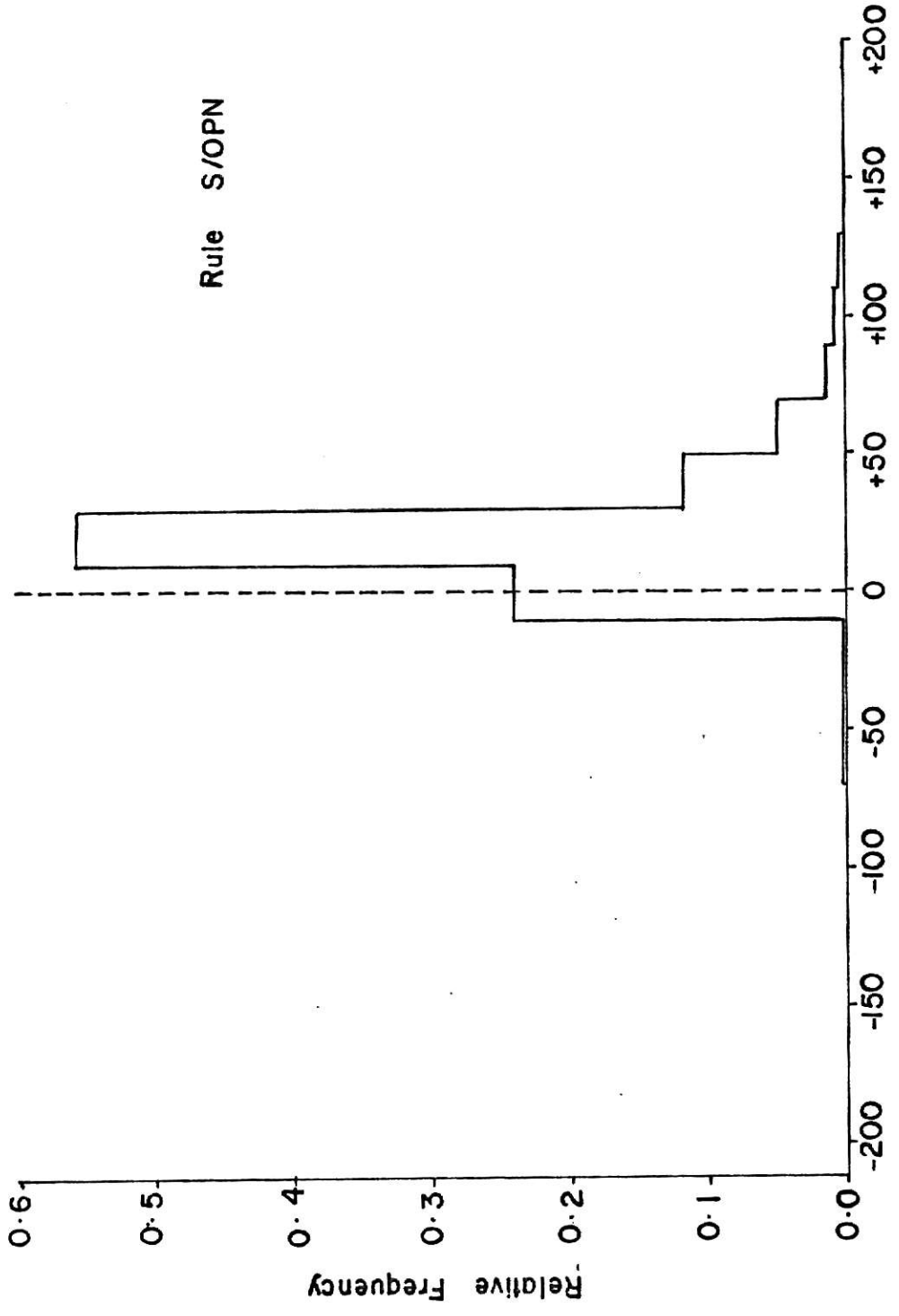


Fig. 15. Frequency distribution of lateness.

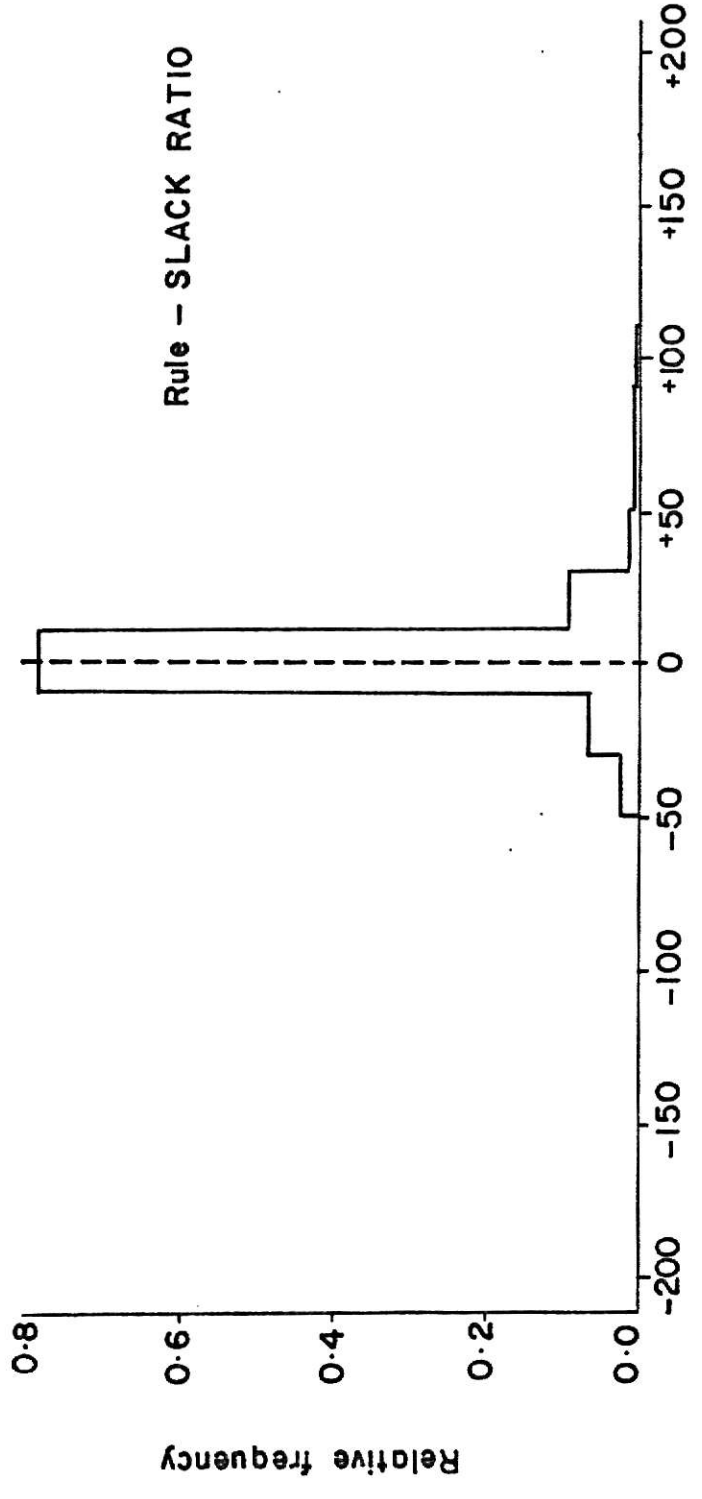


Fig. 16. Frequency distribution of Lateness.

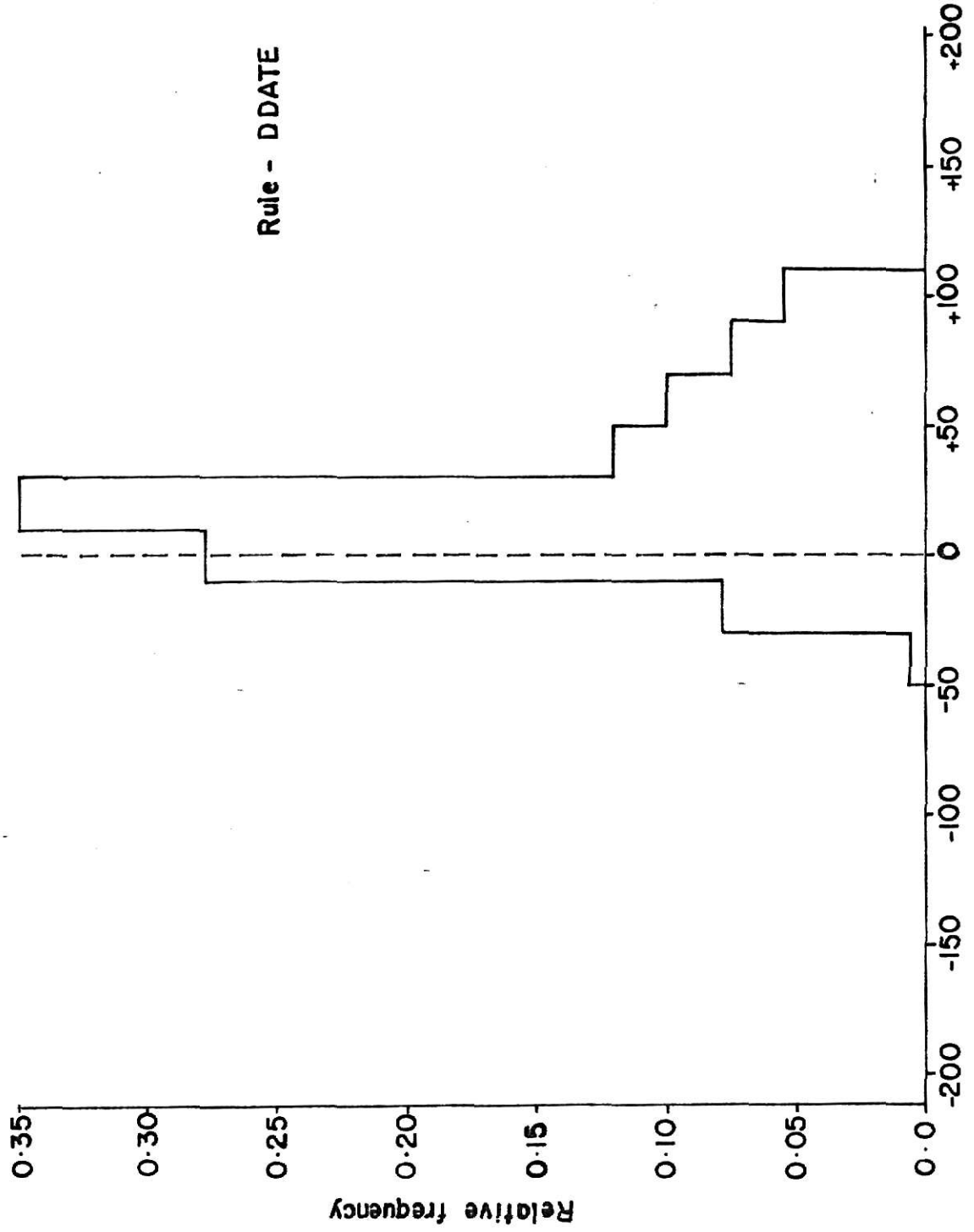


Fig.17. Frequency distribution of lateness.

TABLE 1

Summary of Results for Experiment One with Constant Due-dates

Sample size = 650

	Shop Time			Lateness						
Rule	Mean	Var.	Std. Dev.	Mean	Var.	Std. Dev.	Mean Jobs in Q	NJL	% Late	Shop Util.
1. FCFS	106.31	2722.94	52.1	27.52	2722.66	52.1	72.37	414	63.5	0.91
2. SPT	38.50	896.76	29.9	-40.29	896.75	29.9	25.51	65	10.0	0.86
3. SLACK	96.63	720.43	26.8	14.83	720.47	26.8	67.40	448	68.8	0.89
4. S/OPN	91.53	399.15	20.0	12.73	388.23	20.0	68.32	497	76.5	0.90
5. S/RT	74.18	301.48	17.4	- 4.62	301.48	17.4	52.47	82	12.6	0.88
6. DDATE	90.54	744.01	27.2	11.74	744.02	27.2	64.82	419	64.5	0.89

TABLE 2

Experiment One

Number of Jobs Late with Constant Due-date

Job Number	FCFS	SPT	SLACK	S/OPN	S/RT	DDATE
101-200	41	14	15	27	9	20
201-300	47	6	35	60	5	37
301-400	67	11	74	76	8	55
401-500	69	9	79	89	5	73
501-600	72	10	98	97	6	93
601-700	82	15	97	98	25	94
701-750	36	0	50	50	24	47
Total NJL	414	65	448	497	82	419
Percentage	63.5	10.0	68.8	76.5	12.6	64.5

TABLE 3

Experiment One

SPT Performance at Different Levels of Utilization with Constant Due-date

Job Numbers	Utilization 0.72	0.79	0.89
	Number of jobs late		
101-200	3	9	14
201-300	1	4	6
301-400	7	10	11
401-500	4	8	9
501-600	1	4	10
601-700	14	13	21
701-800	16	20	20
801-900	14	27	30
901-1000	16	20	21
Total NJL	76	115	132
Percentage	8.6	12.9	14.7

TABLE 4

Summary of Results with Constant Due-dates*

Shop Utilization = 0.90

	Shop Time			Lateness						
Rule	Mean	Var.	Std. Dev.	Mean	Var.	Std. Dev.	Mean Jobs in Q	NJL	% Late	Sample Size
FCFS	74.43	5739.2	70.6	- 4.37	5739.2	70.6	58.87	2937	33.8	10,000
SPT	35.02	2318.0	48.2	-44.78	2318.0	48.2	23.25	957	11.0	10,000
DDATE	72.52	1565.2	39.3	- 6.28	1565.2	39.3	57.51	3819	43.9	10,000
P+S/OPN (0,1) (S/OPN)	73.66	1118.1	33.5	- 5.14	1118.1	33.5	58.44	4183	48.1	10,000

*Conway, R. W., "An Experimental Investigation of Priority Assignment in a Job-Shop." Rand Corporation Memorandum, RM-3789-PR, February 1964.

could thus be concluded that if an exact method of assigning due-dates were known, the mean lateness and the variance would be zero, and no jobs would be late. This, of course, is not possible due to the presence of other jobs in the shop during the time that the given job is present.

Finally, the performance of the slack ratio (S/RT) rule was quite encouraging. If some method for reducing the queue length, and hence the shop time, could be devised, this rule would surpass the performance of the SPT rule.

2. EXPERIMENT TWO

In this experiment the input was kept constant i.e. the interarrival mean was set at 1.14, giving a shop utilization of 86.7% with the SPT rule. Also the minimum values for interarrival and processing time distributions were reduced to 0.05.

The type of due-date used has a marked effect on the mean lateness. Thus an initial investigation was made to determine the best due-date multiplier (discussed in Chapter V). Frequency distribution of lateness for the SPT rule are shown in Figure 18. The effect of this multiplier on both the SPT and SLACK rules was studied (Table 5). Figure 19 shows the variation in mean lateness for SPT rule with different due-date multipliers. From this figure, a factor of 3 would seem to be the best. However, if one were to consider the percentage of jobs late, any multiplier between 5 and 7 would seem adequate. Figure 20 shows that only 11% jobs were late with a factor of 5, and this seems quite reasonable. The look ahead rule was found to have no effect with very tight due-dates and hence

TABLE 5

Experiment Two

Performance of SPT and SLACK rules under different Due-dates

SPT RULE

Due-date Multiplier (DATE)	Shop Time Mean	Shop Time Var.	Shop Time Std. Dev.	Lateness Mean	Lateness Var.	Lateness Std. Dev.	Mean Jobs in Q	NJL	Shop Utilization
9	30.89	461.30	21.5	-55.53	919.79	30.3	21.16	2	0.867
7	30.89	461.30	21.5	-36.32	533.42	23.2	21.16	14	0.867
5	30.89	461.30	21.5	-17.12	309.46	17.6	21.16	44	0.867
3	30.89	461.30	21.5	2.08	248.23	15.8	21.16	173	0.867

SLACK RULE

9	47.29	1399.92	37.4	-37.36	97.65	9.9	32.11	0	0.825
7	49.56	1043.52	32.4	-16.37	85.88	9.3	33.47	9	0.815
5	51.01	634.06	25.2	3.51	73.66	8.6	31.90	287	0.816
3	52.65	325.08	18.1	23.90	70.34	8.4	34.74	397	0.803

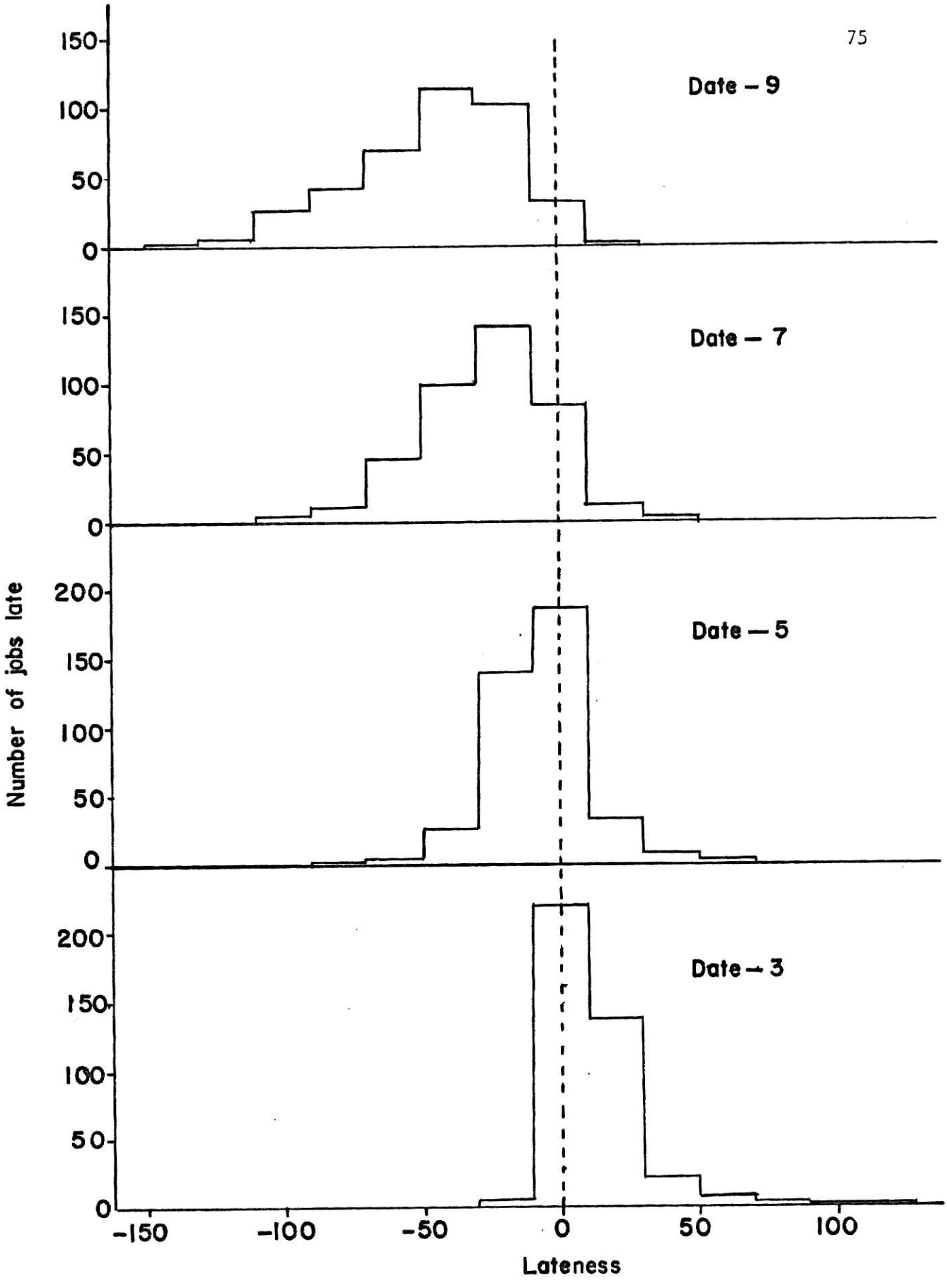


Fig.18. Frequency distribution of lateness with different due dates – SPT rule.

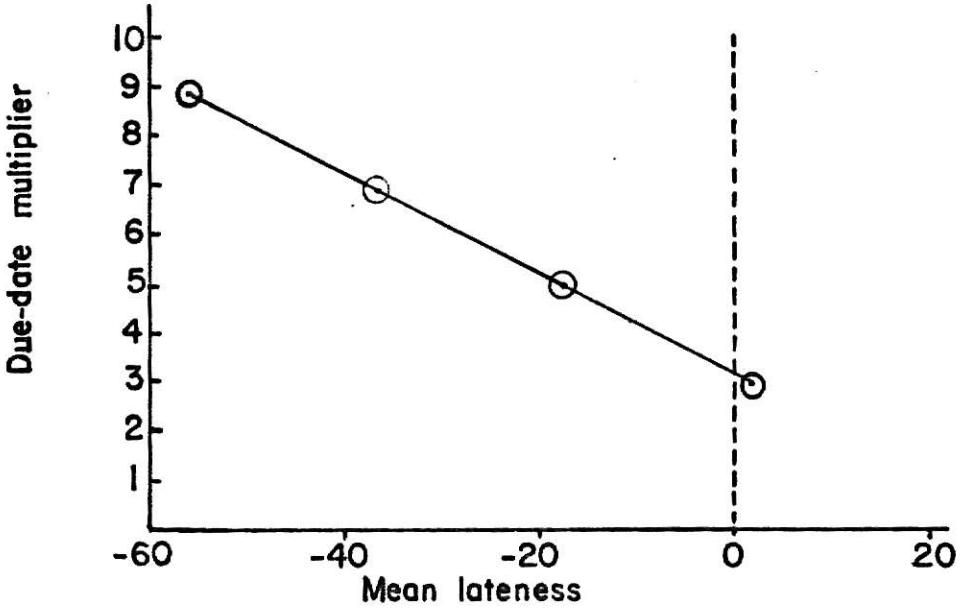


Fig.19. Relationship between due-date multiplier and mean lateness for SPT rule.

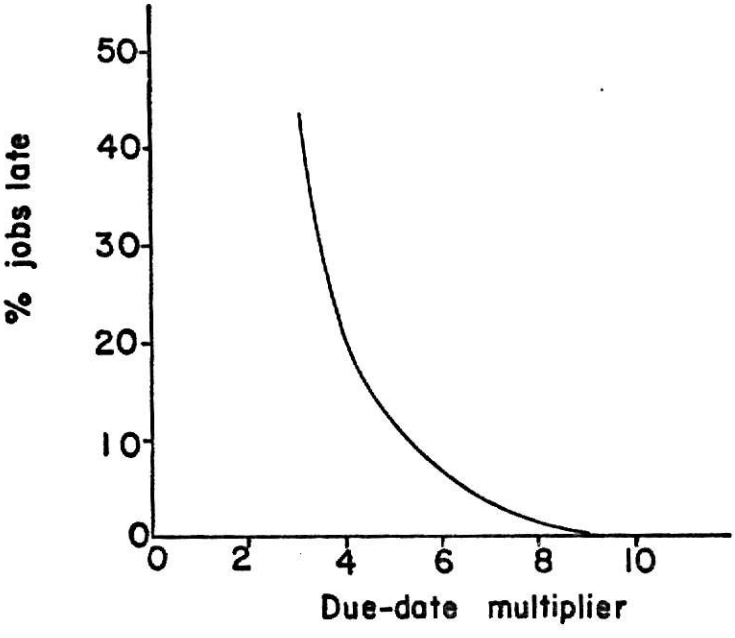


Fig.20. Relationship between due-date multiplier and % jobs late for SPT rule.

multipliers 5 and 7 were used for the rest of the experiment.

Effect of the Look Ahead Rule

Results of the experiment for the three slack rules are given in Tables 6-8. As seen from the tables no significant improvement in the schedule was achieved. The only significant improvement worth the effort was attained with the SLACK and SLACK RATIO rules with a due-date of 7. On the other hand the performance of the SLACK PER OPERATION rule was worse.

The effect of the look ahead rule varied depending on the rule and due-date multiplier used, and no particular trend was observed, thus making it difficult to generalize this rule. The look ahead rule, however, did reduce the mean lateness in the cases where the schedule was improved. Figures 21-29 show the frequency distribution for various rules and the effect of look ahead on lateness.

Effect of the factor GSLK

It is quite evident from Tables 7-12 that the mean lateness and the number of jobs late increases with a factor of 0 as compared to results obtained without the look ahead rule. With GSLK set at 0 every job with positive slack is replaced in case a critical or late job is likely to come to the machine under consideration.

A more reasonable method of looking ahead would be to replace only those jobs which have a slack that is positive and greater than zero. This way, the replaced job would have sufficient slack time to be finished on time.

The performance of the look ahead under different rules and for varying values of GSLK is shown in Figures 23, 26, 29. The improvement in most cases was obtained with GSLK values of 7.5 and 10.0. No significant changes in the other measures of performance i.e., mean jobs in queue and shop utilization were observed.

The look ahead rule did not perform as well as expected but it did point out some new directions in which future work could be directed. The most obvious of these was the value of GSLK to be used while considering the question of job replacement.

TABLE 6

Experiment Two

Effect of LOOK AHEAD on SLACK rule

Sample size = 400

Due-date multiplier - 5

Factor GSLK	Shop Time Mean	Shop Time Var.	Shop Time Std. Dev.	Lateness Mean	Lateness Var.	Lateness Std. Dev.	Mean Jobs in Q	NJL	Total Jobs in Shop	Shop Utilization
0.0	58.84	652.24	25.5	11.46	82.75	9.1	36.60	358	660	0.798
5.0	54.15	625.22	25.0	6.73	63.70	7.9	36.00	320	652	0.809
7.5	52.40	644.01	25.4	4.92	76.47	8.8	34.90	297	650	0.808
10.0	50.66	645.39	25.4	3.20	62.34	7.8	33.80	279	651	0.816
12.5	51.67	626.77	25.0	4.42	70.31	8.4	34.54	302	650	0.814
**	51.01	634.06	25.2	3.51	73.66	8.6	31.90	287	650	0.814
Due-date multiplier - 7										
0.0	48.89	1005.09	31.7	-17.10	74.37	8.6	32.92	3	648	0.820
5.0	49.08	1020.71	31.9	-16.92	66.82	8.2	33.05	2	650	0.821
7.5	50.08	1033.74	32.1	-15.80	71.23	8.4	33.67	8	650	0.814
10.0	48.12	971.09	31.1	-17.83	72.58	8.5	32.51	1	650	0.821
**	49.56	1043.52	32.4	-16.37	85.88	9.3	33.47	9	646	0.815

**Without LOOK AHEAD

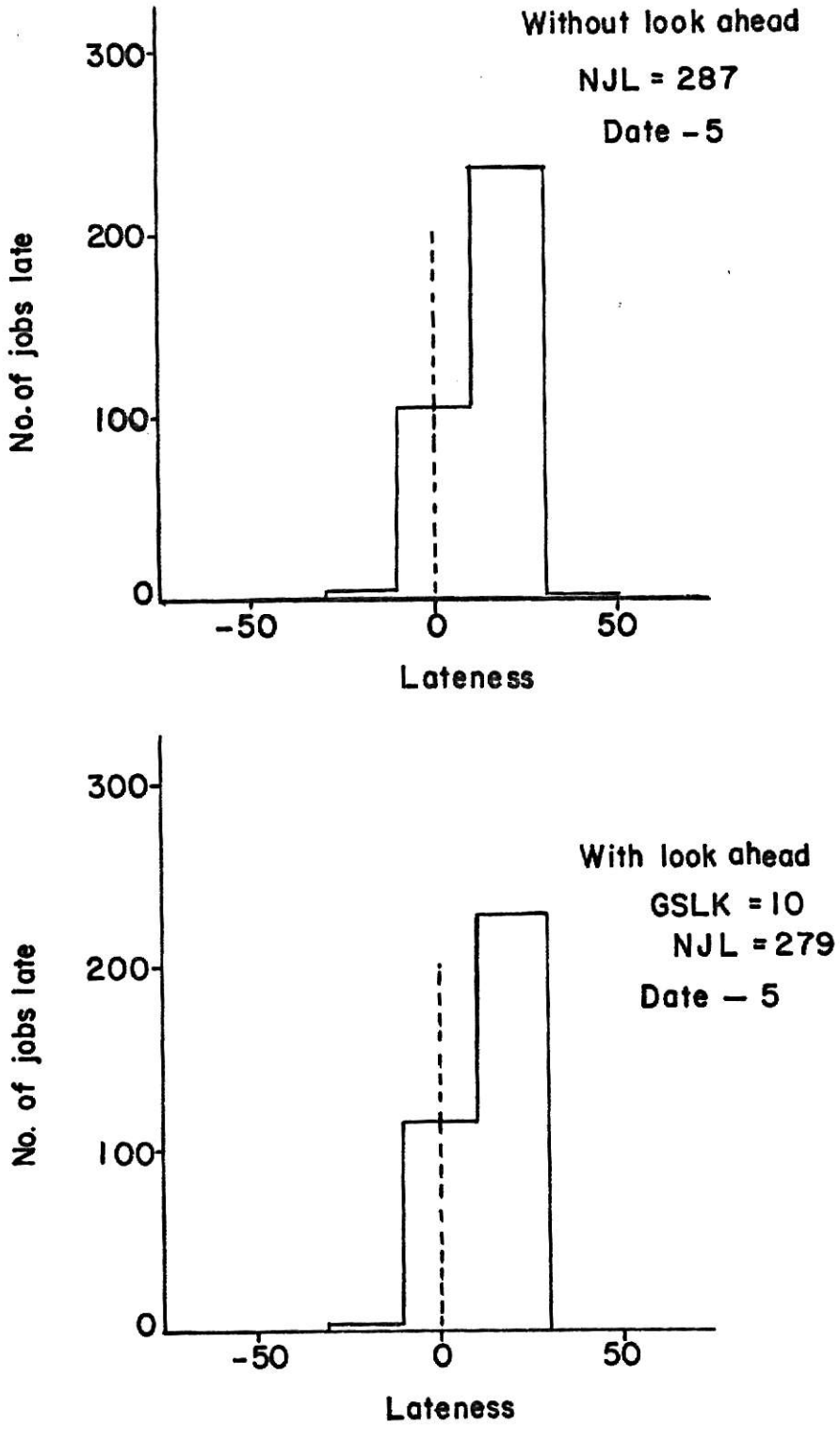


Fig.21. Lateness frequency distribution for SLACK rule.

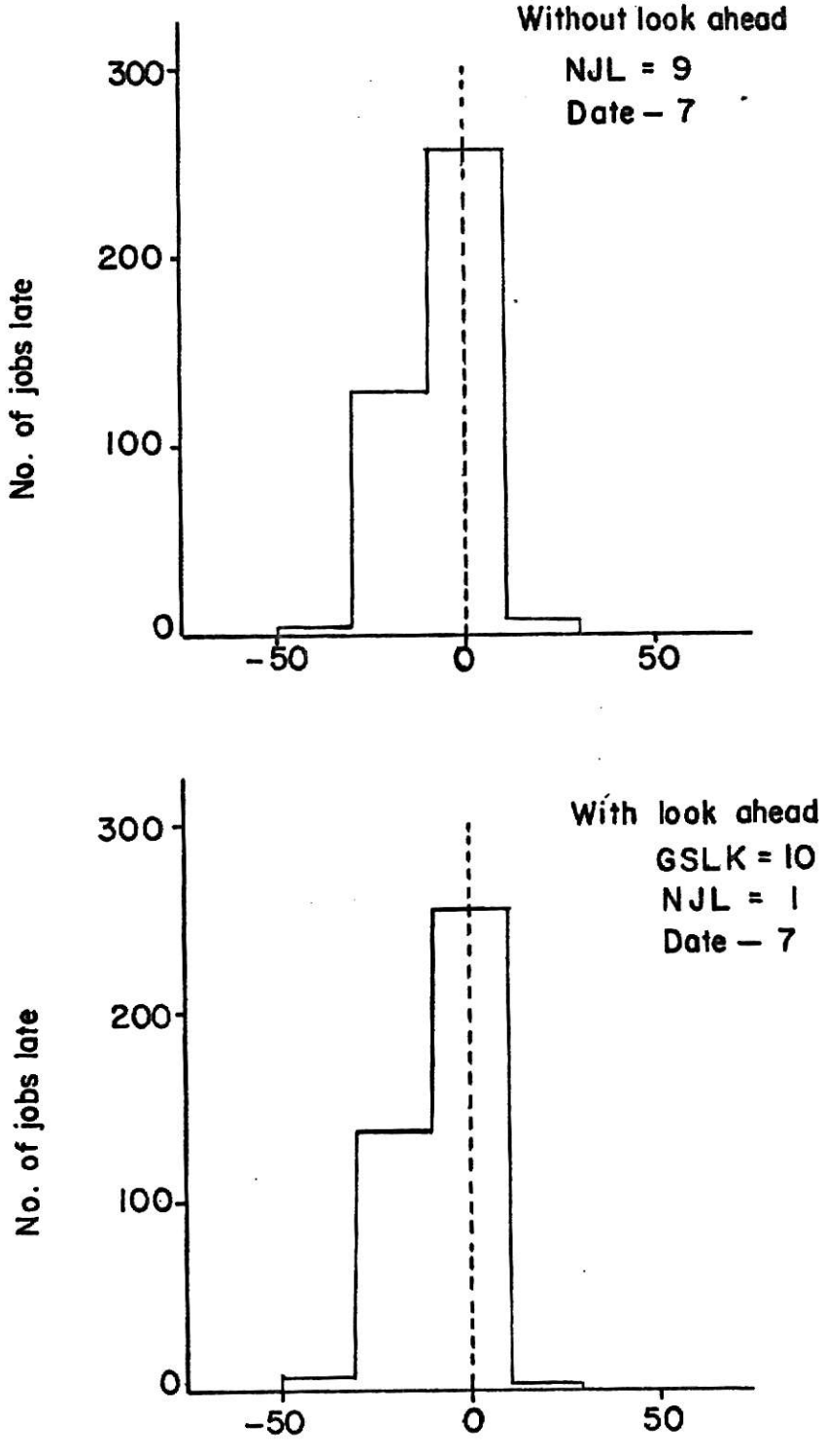


Fig. 22. Lateness frequency distribution for SLACK rule.

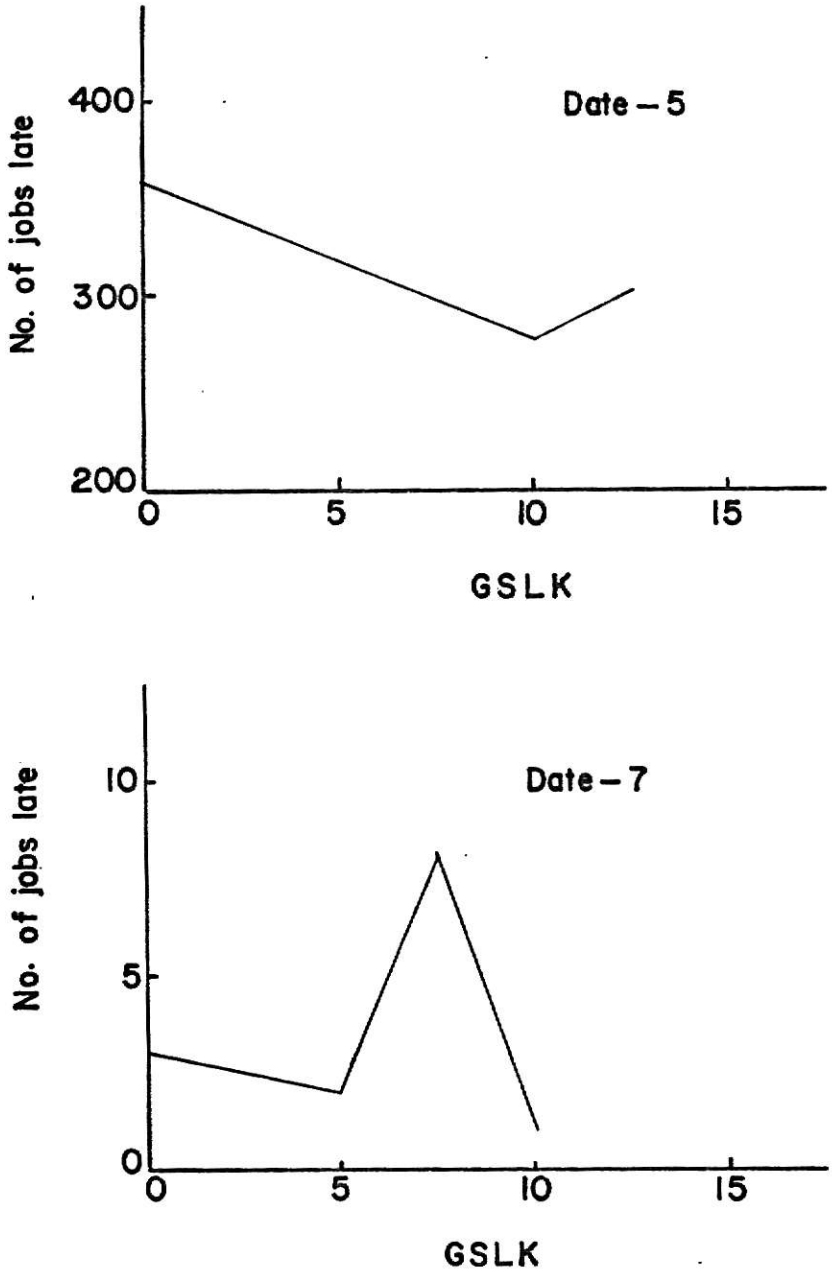


Fig.23. Effect of factor GSLK on SLACK rule.

TABLE 7

Experiment Two

Effect of LOOK AHEAD on SLACK PER OPERATION rule

Sample Size = 400

Due-date multiplier - 5

Factor GSLK	Shop Time Mean	Shop Time Var.	Shop Time Std. Dev.	Lateness Mean	Lateness Var.	Lateness Std. Dev.	Mean Jobs in Q	NJL	Total Jobs in Shop	Shop Utilization
0.0	61.00	794.62	28.2	13.77	137.82	11.7	40.64	364	660	0.810
5.0	57.98	695.75	26.5	10.66	81.54	9.0	38.86	364	659	0.813
7.5	55.93	621.95	24.9	8.59	62.84	7.9	38.85	361	653	0.816
10.0	56.95	657.77	25.5	9.63	74.35	8.6	38.21	360	656	0.813
12.5	56.84	670.25	25.8	9.37	71.51	8.4	37.98	362	651	0.810
**	51.34	570.53	23.9	3.98	43.55	6.5	34.80	279	650	0.825
					Due-date multiplier - 7					
0.0	55.41	1000.23	31.7	-10.57	40.57	6.8	39.74	1	650	0.866
5.0	55.41	1000.32	31.7	-10.57	40.56	6.8	36.63	1	650	0.866
10.0	55.81	998.55	31.5	-10.16	39.67	6.3	39.94	1	655	0.872
**	56.25	973.25	31.2	- 9.72	36.52	6.0	37.30	0	652	0.866

**Without LOOK AHEAD

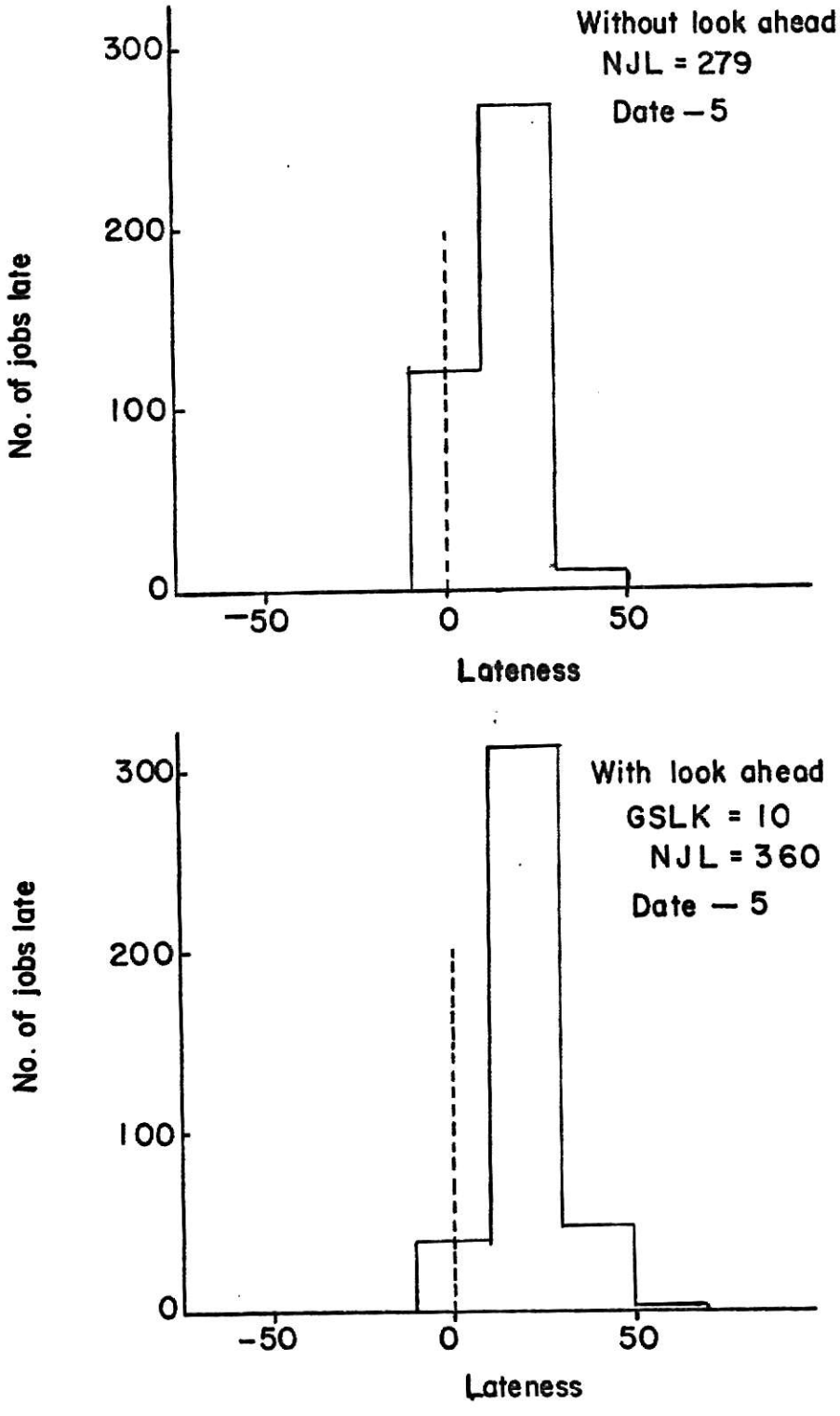


Fig.24. Lateness frequency distribution for SLACK/OPERATION rule.

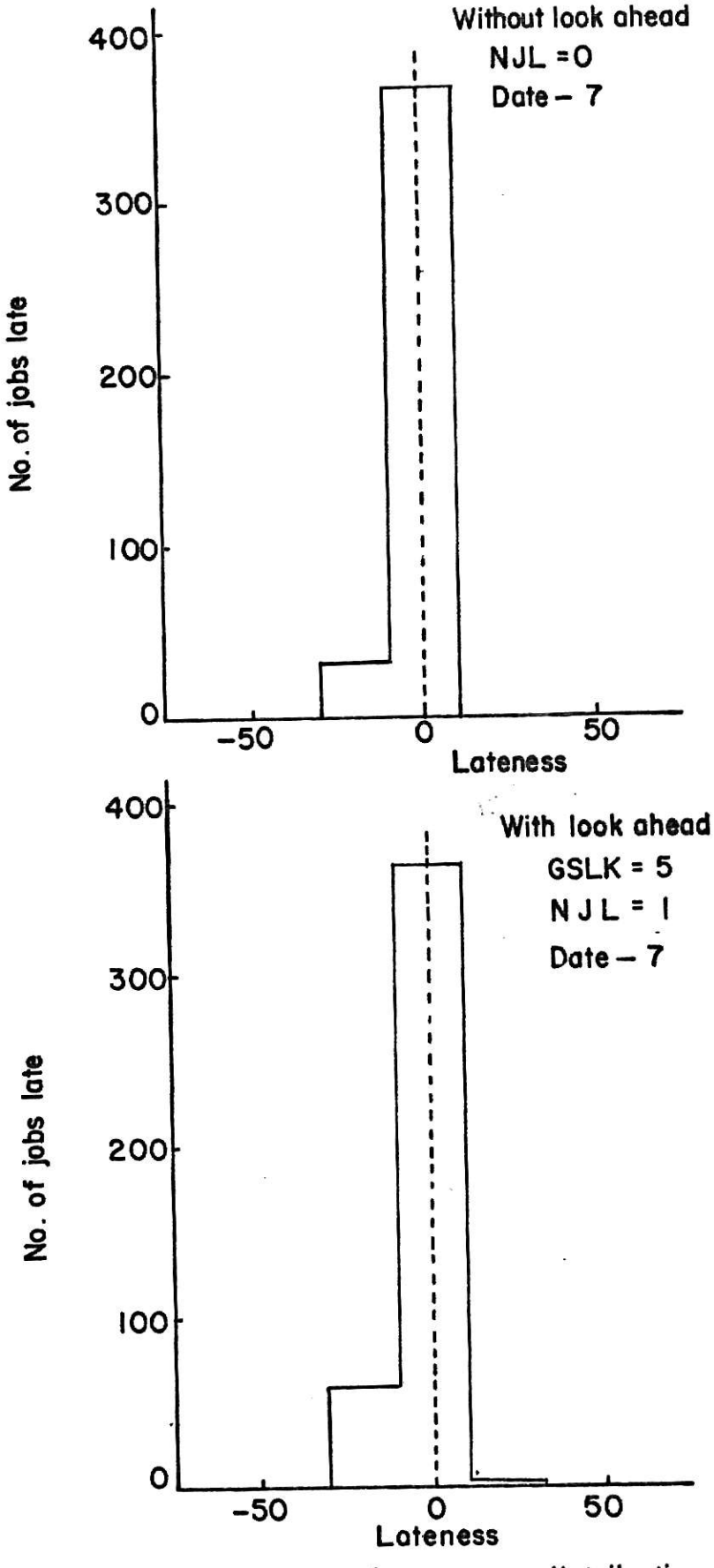


Fig.25. Lateness frequency distribution for SLACK/OPERATION rule.

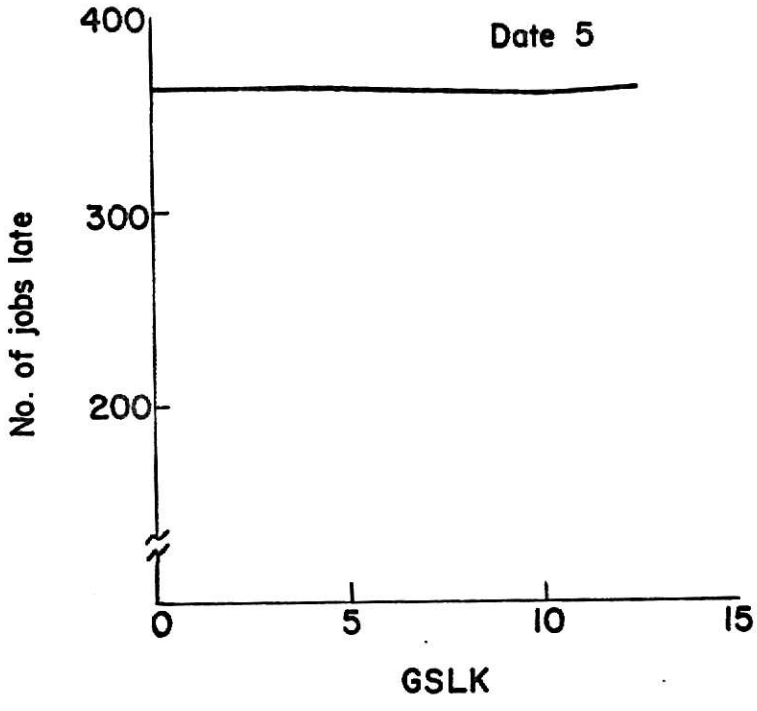


Fig. 26. Effect of factor GSLK on SLACK/OPERATION rule.

TABLE 8

Experiment Two

Effect of LOOK AHEAD on SLACK RATIO rule

Sample size = 400
Due-Date multiplier - 5

Factor GSLK	Shop Time Mean	Shop Time Var.	Shop Time Std. Dev.	Lateness Mean	Lateness Var.	Lateness Std. Dev.	Mean Jobs in Q	NJL	Total Jobs in Shop	Shop Utilization
0.0	66.53	2269.44	48.6	18.58	2066.99	45.4	50.84	211	675	0.924
5.0	67.86	2094.74	45.6	19.37	1877.12	43.3	56.55	197	698	1.000
7.5	65.26	19.2.93	44.6	17.02	1672.66	40.9	49.46	177	663	0.914
10.0	66.92	1841.06	42.9	18.42	1563.51	39.6	50.15	211	675	0.905
**	61.85	1401.25	37.4	13.57	1237.43	35.2	47.00	176	670	0.918
Due-date multiplier - 7										
0.0	59.17	893.30	29.5	-7.19	331.32	18.2	44.44	37	660	0.906
5.0	59.15	902.19	30.0	-7.25	329.37	18.1	44.26	36	660	0.904
7.5	60.52	1075.35	32.8	-6.17	537.51	23.1	44.91	31	660	0.897
10.0	61.56	1218.73	34.3	-4.98	625.76	25.0	44.80	37	660	0.878
**	63.56	1445.40	38.1	-2.68	942.01	30.7	48.26	39	661	0.878

**Without LOOK AHEAD

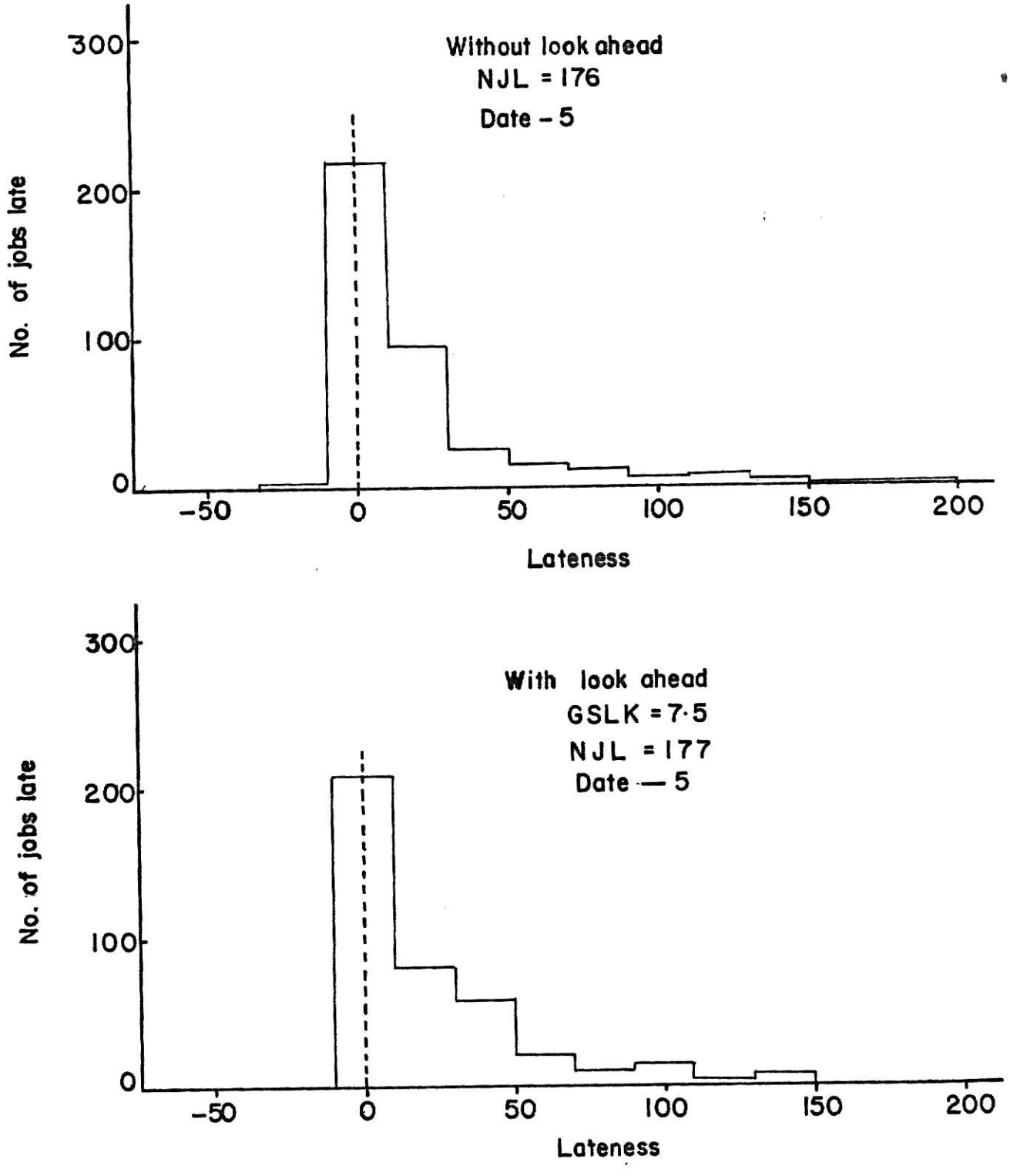


Fig.27. Lateness frequency distribution for SLACK RATIO rule.

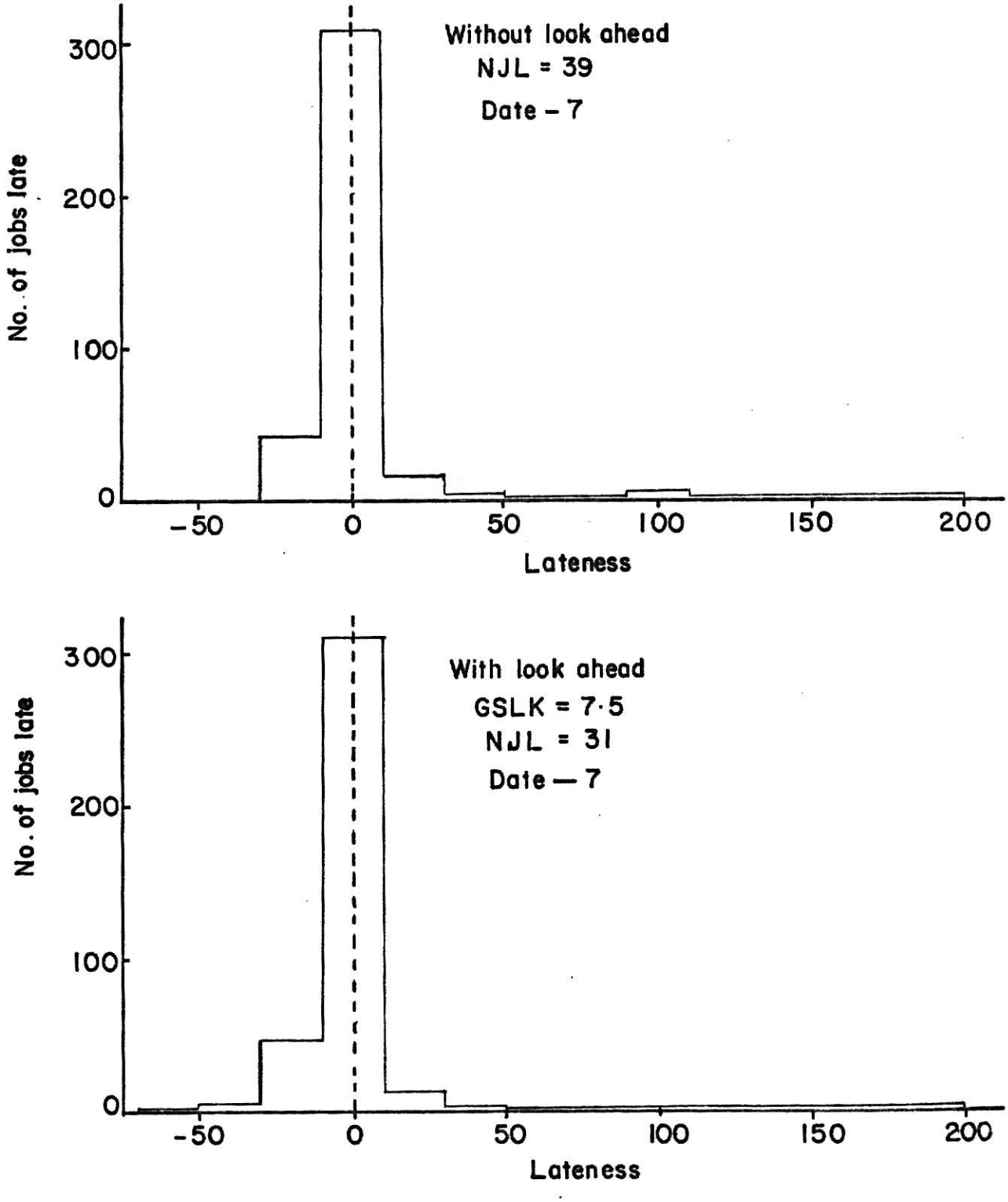


Fig.28. Lateness frequency distribution for SLACK RATIO rule.

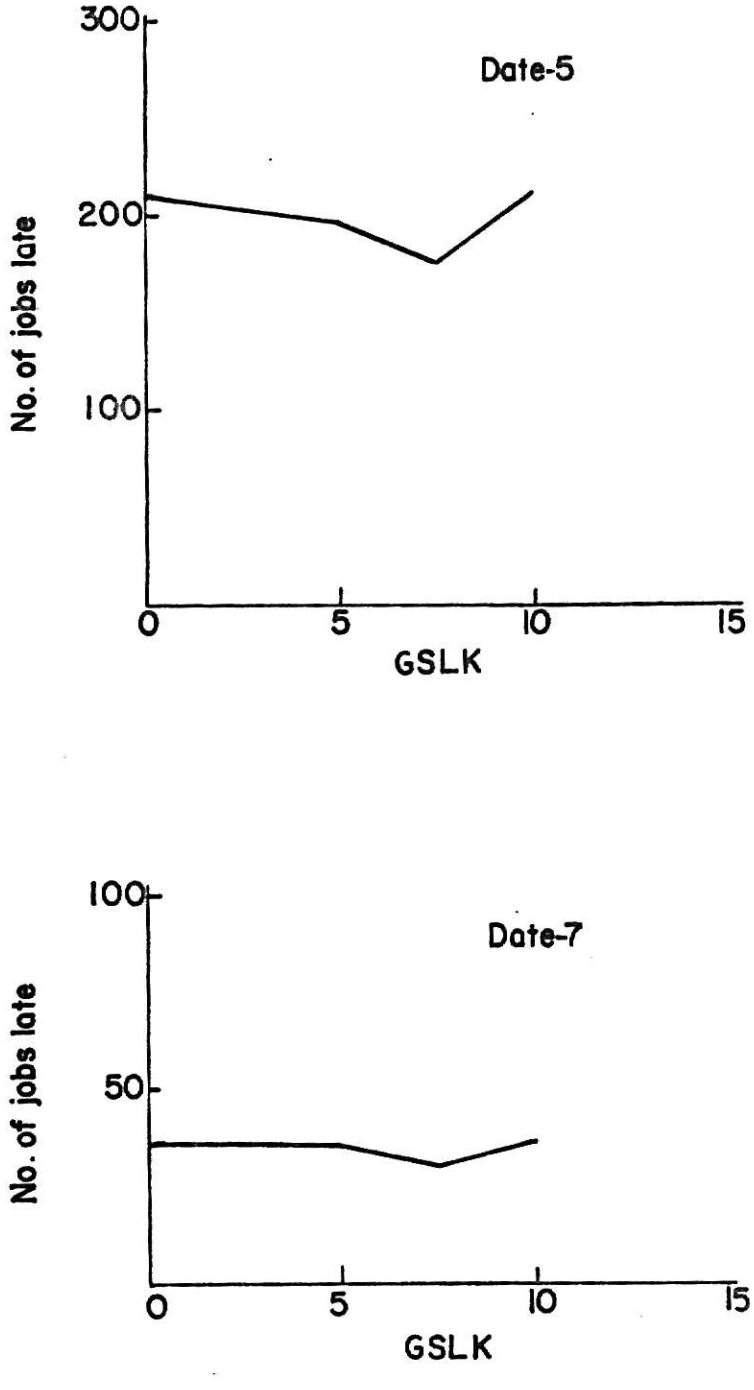


Fig.29. Effect of GSLK on SLACK RATIO rule.

CHAPTER VIII

CONCLUSION

Simulation techniques have been employed to study nearly all phases of the manufacturing process including forecasting, long-range planning, order release, and the flow of jobs on the production floor.

Simulation has been found to be a reasonably effective method of testing and comparing scheduling procedures. In this study, the shortest processing-time (SPT) rule was found to give the best performance. With this rule, the mean number of jobs in the shop, the mean shop time and the mean lateness were reduced.

From this study it was concluded that a reasonable due-date multiplier (DATE) to be used in assigning due-dates lies between 5 and 7. With the multiplier lying in this range, jobs late with the SPT rule could be kept below 10%.

The performance of the slack based rules has been shown to depend on the type of due-date assigning procedure used, and this has been substantiated here. With these rules, the lateness variance reduced considerably. To improve their performance in terms of the number of jobs late, the look ahead rule was implemented. Several variations of this rule were tested by changing the value of the slack factor GSLK. No significant improvement in the schedule was observed. The slight improvement observed was for values of GSLK above 5. This indicates that it is not worthwhile to replace every job with positive slack by a late job.

In this work, any job having a slack greater than the factor GSLK was replaced by a late job. No consideration was given to the number of operations remaining. One modification could be to consider the remaining processing-time or number of operations remaining on the job. The value of GSLK could then be some multiple of either of the above mentioned factors. This way the possibility of replacing the wrong job could be eliminated.

The other aspect of this study was to test the effectiveness of GASP II in dealing with large problems. Although simple to use, it was found to be slow in execution, a run of 650 jobs requiring anywhere from 25 to 55 minutes of computer (IBM 360/50) time, depending primarily on the selection of priority assignment procedure. A run of 10,000 jobs, using SIMSCRIPT was found to require 15 to 35 minutes on IBM 7090 computer (9).

If any consideration at all is given to the difficulty and cost of implementation, however, then SPT rule is certainly the best procedure of those tested in this study. This rule can be employed directly with very little difficulty in implementation, and with tremendous improvement in system performance.

REFERENCES

1. Ackerman, S. S., "Even-flow, A scheduling method for Reducing Lateness in Job Shops," Management Technology, Vol. 3, No. 1, May 1963.

2. Baker, C. T. and Dzielinski, B. P., "Simulation of a Simplified Job Shop," Management Science, Vol. 6, No. 3, April 1960.

3. Bowman, E. H., "The Schedule sequencing Problem," Operations Research, Vol. 7, No. 5, September 1959.

4. Buffa, E. S., "Production-Inventory Systems: Planning and Control," Richard D. Irwin, Inc., Homewood, Illinois, 1968.

5. Clark, W., "The Gantt Chart", Pitman & Sons, London, 3rd Edition, 1952.

6. Conway, R. W., "Some Tactical Problems in Digital Simulation," Management Science, Vol. 10, No. 1, October 1963.

7. _____, "Priority Dispatching and work-in-Process Inventory in a Job Shop," The Journal of Industrial Engineering, Vol. 16, No. 2, March 1965.

8. _____, "Priority Dispatching and Job Lateness in a Job Shop," The Journal of Industrial Engineering, Vol. 16, No. 4, July 1965.

9. _____, "An Experimental Investigation of Priority Assignment in a Job Shop," RAND Corporation Memorandum, RM-3789-PR, February 1964.

10. Conway, R. W. and Maxwell, W. L., "Network Dispatching by the Shortest Operation Discipline," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 17.

11. Conway, R. W., Johnson, B. M. and Maxwell, W. L., "An Experimental Investigation of Priority Dispatching," The Journal of Industrial Engineering, Vol. 11, No. 3, May 1960.

12. Conway, R. W., Maxwell, W. L. and Miller, L. W., "Theory of Scheduling", Addison-Wesley, Reading, Massachusetts, 1967.

13. Fisher, H. and Thompson, G. L., "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 15.

14. George, P. B., "An Investigation of the Effects of Idle Time Coalescense in a Network of Independent Single Server Queues," Master's Thesis, Air Force Institute of Technology, Ohio, 1968.

15. Gere, W., "A Heuristic Approach to Job-Shop Scheduling," Ph.D. Thesis, Carnegie Institute of Technology, 1962.

16. Giffler, B., "Schedule Algebras and their use in Formulating General Systems Simulations," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 4.

17. Giffler, B., Thompson, G. L. and Van Ness, V., "Numerical Experience with the Linear and Monte Carlo Algorithms for Solving Production Scheduling Problems," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 3.

18. Gomory, R. E., "An All-Integer Integer Programming Algorithm," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 13.

19. Heller, J., "Some Numerical Experiments for an M X J Flow Shop and its Decision - Theoretical Aspects," Operations Research, Vol. 8, No. 2, March 1960.

20. Jackson, J. R., "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report 43, Management Science Research Project, UCLA, January 1955.

21. Jackson, J. R., "Queues with Dynamic Priority Discipline," Management Science, Vol. 8, No. 1, October 1961.

22. Johnson, S. M., "Optimal Two- and Three Stage Production Schedules with Set-up Times Included," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963, Chapter 2.

23. Krasnow, H. S., and Merikallio, R. A., "The Past, Present, and Future of General Simulation Languages," Management Science, Vol. 11, No. 2, November 1964.

24. Lave, R. E., "Timekeeping for Simulation," The Journal of Industrial Engineering, Vol. 18, No. 7, July 1967.

25. LeGrande, E., "The Development of a Factory Simulation Using Actual Operating Data," Management Technology, Vol. 3, No. 1, May 1963.

26. Little, J. D. C., "A Proof for the Queuing Formula L = λW," Operations Research, Vol. 9, No. 3, May 1961.

27. Manne, A. S., "On the Job-Shop Scheduling Problem," Operations Research, Vol. 8, No. 2, March 1960.

28. Mize, J. H., and Cox, J. G., "Essentials of Simulation", Prentice Hall, Englewood Cliffs, New Jersey, 1968.

29. Muth, J. F. and Thompson, G. L., "Industrial Scheduling", Prentice Hall, Englewood Cliffs, New Jersey, 1963.

30. Naylor, T. H., Balintfy, J. L., Burdick, D. S. and Chu, K., "Computer Simulation Techniques", John Wiley & Sons, Inc., New York, 1966.

31. Petitt, R. G., "Job Shop Type Scheduling by Simulation," Digest of the Second Conference on Applications of Simulation, December 1968.

32. Pritsker, A. A. B. and Kiviat, P. J., "GASP II - A Fortran Based Simulation Language", Department of Industrial Engineering, Arizona State University, September 1967.

33. Reinitz, R. C., "On the Job-Shop Scheduling Problem," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall Englewood Cliffs, New Jersey, 1963.

34. Rowe, A. J., "Sequential Decision Rules in Production Scheduling," Ph.D. Thesis, UCLA, August 1958.

35. ____, "Towards a Theory of Scheduling," The Journal of Industrial Engineering, Vol. 11, No. 2, March 1960.

36. Schwarz, R. A. and Schriber, T. J., "Application of GPSS/360 to Job Shop Scheduling," Digest of the Second Conference on Applications of Simulation, December 1968.

37. Story, A. E. and Wagner, H. M., "Computational Experience with Integer Programming for Job-Shop Scheduling," Industrial Scheduling, J. F. Muth and G. L. Thompson, Prentice Hall, Englewood Cliffs, New Jersey, 1963.

38. Thompson, W. J., "The Effect of Priority Dispatching Rules on an Assembly Job Shop," Master's Report, Arizona State University, Tempe, 1968.

39. Wagner, H. M., "An Integer Linear-Programming Model for Machine Scheduling," Nav. Res. Log. Quart., Vol. 6, No. 2, June 1959.

ACKNOWLEDGEMENT

The author wishes to express his deep sense of appreciation to his major professor, Dr. L. E. Grosh for his guidance, constructive criticism, helpful suggestions and personal interest taken in the preparation of this master's thesis.

The author expresses his gratitude to Dr. R. F. Kruh, Dean of Graduate School and Dr. J. P. Noonan, Associate Dean of Graduate School, for their constant encouragement and financial assistance.

Thanks are also due to Dr. R. D. Daftary, Mr. Norman Pereira and Mr. Piyush Shah for their timely suggestions.

The author is grateful to Professor A. H. Duncan, Department of Mechanical Engineering for consenting to be a member of the Examining Committee.

Finally the author is indebted to his many friends for their help at one time or another.

APPENDIX A

SAMPLE PRINTOUT FOR SPT RULE - DATE = 5

GASP SUMMARY REPORT

** GENERATED DATA **

CODE	MEAN	VAR.	MIN.	MAX.	OBS.	NJL.
1	-17.12	309.46	-93.57	55.84	400	44.00
2	-17.47	297.91	-93.57	45.11	100	9.00
3	-16.32	250.34	-59.83	38.64	100	11.00
4	-17.34	419.29	-60.68	55.84	100	14.00
5	-17.59	294.47	-70.13	38.94	91	10.00
6	-15.03	130.78	-43.03	-3.86	9	0.0
7	NO VALUES RECORDED					
8	NO VALUES RECORDED					
9	NO VALUES RECORDED					
10	NO VALUES RECORDED					
11	NO VALUES RECORDED					
12	NO VALUES RECORDED					
13	NO VALUES RECORDED					
14	NO VALUES RECORDED					
15	NO VALUES RECORDED					
16	30.89	461.30	4.74	179.57	400	0.0
17	30.41	330.03	4.74	81.75	100	0.0
18	31.55	352.95	6.32	119.35	100	0.0
19	34.29	810.42	8.22	179.57	100	0.0
20	28.26	357.63	5.67	105.75	91	0.0
21	17.55	68.45	8.95	34.15	9	0.0
22	NO VALUES RECORDED					
23	NO VALUES RECORDED					
24	NO VALUES RECORDED					
25	NO VALUES RECORDED					
26	NO VALUES RECORDED					
27	NO VALUES RECORDED					
28	NO VALUES RECORDED					
29	NO VALUES RECORDED					
30	NO VALUES RECORDED					

GASP SUMMARY REPORT

** TIME GENERATED DATA **

CODE	MEAN	VAR.	MIN.	MAX.	TOTAL TIME
1	7.28	0.58	0.0	8.00	737.280
2	2.60	3.85	0.0	10.00	739.320
3	1.77	4.69	0.0	10.00	738.480
4	1.80	3.40	0.0	9.00	738.800
5	1.05	2.30	0.0	7.00	739.110
6	0.99	2.35	0.0	8.00	738.210
7	0.95	1.68	0.0	6.00	736.380
8	1.56	2.45	0.0	8.00	738.760
9	2.92	9.28	0.0	15.00	737.630
10	1.63	3.05	0.0	8.00	738.850
11	10.69	23.53	2.21	31.19	737.520

GASP SUMMARY REPORT

** GENERATED FREQUENCY DISTRIBUTIONS **

CODE	HISTOGRAMS																				
1	0	244	128	101	50	31	26	16	12	5	3	4	7	1	3	3	0	0	0	0	0	0
2	0	2461	1451	931	560	333	199	118	67	44	28	20	10	7	2	2	1	0	1	0	0	0
3	0	0	0	0	0	0	1	2	26	140	187	32	8	4	0	0	0	0	0	0	0	0
4	1	135	147	64	25	14	8	1	3	0	0	1	1	0	0	0	0	0	0	0	0	0
5	0	188	127	96	68	39	27	25	15	5	2	4	1	1	1	1	0	0	0	0	0	0

*** QUEUE PRINTOUT ..... QUEUE NO... 1 ***

AVERAGE NO. OF ITEMS IN QUEUE WAS 7.970

MAXIMUM 10

*** QUEUE CONTENTS ***

739.62	2.00	726.32	3.00	0.78	5.00	621.00	784.12	8.65	13.00	2.00
739.65	2.00	673.30	5.00	1.31	15.00	579.00	750.02	1.31	15.00	0.0
739.77	2.00	735.66	7.00	0.67	3.00	631.00	764.65	3.81	6.00	3.00
739.81	2.00	726.63	1.00	1.56	6.00	622.00	802.06	11.18	15.00	4.00
740.43	2.00	725.39	4.00	3.16	6.00	620.00	774.83	5.18	8.00	8.00
740.66	2.00	722.85	2.00	1.87	9.00	618.00	790.76	4.66	13.00	4.00
741.17	1.00	741.17	6.00	1.93	1.00	635.00	805.17	12.80	9.00	2.00
742.07	2.00	733.25	8.00	2.76	2.00	626.00	793.17	11.72	14.00	4.00

*** QUEUE PRINTOUT ..... QUEUE NO... 3 ***

AVERAGE NO. OF ITEMS IN QUEUE WAS 2.476

MAXIMUM 11

*** QUEUE CONTENTS ***

735.04	1.00	720.26	2.00	3.46	10.00	616.00	769.78	3.46	10.00	0.0

*** QUEUE PRINTOUT ..... QUEUE NO... 4 ***

AVERAGE NO. OF ITEMS IN QUEUE WAS 2.469

MAXIMUM 10

*** QUEUE CONTENTS ***

736.70	1.00	735.27	3.00	2.10	2.00	629.00	768.04	5.65	6.00	9.00
732.13	1.00	706.76	3.00	2.19	14.00	608.00	777.33	5.27	15.00	5.00
721.02	1.00	713.68	3.00	2.34	4.00	614.00	735.57	2.67	6.00	8.00
728.10	1.00	706.71	3.00	3.01	14.00	607.00	767.35	4.22	15.00	2.00
734.87	1.00	734.81	3.00	3.57	2.00	628.00	813.15	15.59	15.00	2.00
711.12	1.00	702.65	3.00	4.44	3.00	602.00	759.61	8.72	10.00	7.00
659.46	1.00	658.40	3.00	5.23	2.00	564.00	720.86	11.54	8.00	6.00

*** QUEUE PRINTOUT ..... QUEUE NO... 5 ***

AVERAGE NO. OF ITEMS IN QUEUE WAS 1.594

*** QUEUE CONTENTS ***

639.10	1.00	639.10	7.00	0.72	15.00	552.00	751.71	0.72	15.00	0.0
658.41	1.00	658.41	7.00	0.14	5.00	564.00	720.87	3.33	8.00	5.00
685.35	1.00	685.35	7.00	0.92	4.00	590.00	765.86	8.63	10.00	4.00
696.14	1.00	696.14	7.00	0.68	12.00	597.00	783.65	3.28	14.00	4.00
702.66	1.00	702.66	7.00	1.84	4.00	602.00	759.62	4.27	10.00	4.00
709.55	1.00	709.55	7.00	0.93	13.00	610.00	768.92	2.12	15.00	6.00
724.70	1.00	724.70	7.00	3.68	9.00	619.00	809.59	3.75	10.00	9.00
735.33	1.00	735.33	7.00	0.28	8.00	630.00	769.50	1.68	9.00	2.00
736.61	1.00	736.61	7.00	2.10	3.00	633.00	802.42	12.95	13.00	5.00
736.61	1.00	736.61	7.00	0.13	8.00	633.00	802.42	5.12	13.00	1.00

*** QUEUE PRINTOUT ..... QUEUE NO... 18 ***

AVERAGE NO. OF ITEMS IN QUEUE WAS 18.655

MAXIMUM 36

*** QUEUE CONTENTS ***

696.14	1.00	696.14	8.00	0.47	11.00	597.00	783.65	3.76	14.00	7.00
705.13	1.00	705.13	8.00	0.66	10.00	603.00	798.32	2.22	13.00	3.00
705.13	1.00	705.13	8.00	0.43	12.00	603.00	798.32	1.22	13.00	4.00
709.55	1.00	709.55	8.00	0.24	15.00	610.00	768.92	0.24	15.00	0.0
713.69	1.00	713.69	8.00	0.06	5.00	614.00	735.58	0.33	6.00	4.00
722.52	1.00	722.52	8.00	1.76	10.00	617.00	797.61	6.48	13.00	1.00
722.87	1.00	722.87	8.00	0.97	11.00	618.00	790.78	2.74	13.00	9.00
725.40	1.00	725.40	8.00	0.08	7.00	620.00	774.84	2.02	8.00	5.00
726.65	1.00	726.65	8.00	0.06	8.00	622.00	802.08	7.63	15.00	6.00
733.26	1.00	733.26	8.00	0.05	12.00	626.00	793.19	0.17	14.00	1.00
734.82	1.00	734.82	8.00	3.06	9.00	628.00	813.16	9.82	15.00	4.00
736.08	1.00	736.08	8.00	0.43	7.00	632.00	856.35	13.71	15.00	5.00
736.08	1.00	736.08	8.00	3.84	9.00	632.00	856.35	12.08	15.00	3.00
736.08	1.00	736.08	8.00	2.74	14.00	632.00	856.35	2.84	15.00	2.00
736.61	1.00	736.61	8.00	0.38	10.00	633.00	802.42	4.11	13.00	1.00
737.52	1.00	737.52	8.00	5.86	6.00	634.00	794.09	8.87	10.00	6.00
741.17	1.00	741.17	8.00	0.29	7.00	635.00	805.17	3.86	9.00	1.00

APPENDIX B

DESCRIPTION OF VARIABLES

FOR THE SIMULATION PROGRAM

JOB	Job number
NOP	Operation number
NJOB	Job arrival counter
KOUT	Job departure counter
KODE	Code for priority rules
MG	Machine group number
MGQ	Machine group queue number
NJL	Number of jobs late
DDATE	Due-date
KFILE	Storage file for future operations KFILE = 11 to 19
UBAR	Mean shop utilization
SLK	Slack of selected operation
LAH	Code for look-ahead feature
SUMPT	Total processing time remaining for job
M	Number of operations per job
TLATE	Lateness of job
MACH(MG)	Number of machines in machine group MG
NAVL(MG)	Number of available machines in machine group MG
SJOB(I,11)	Storage array for next job arrival. Each job/operation consists of 11 attributes.
STORE(11)	Storage array for selected job/operation.
JOP(JOB)	Number of incomplete operations on job
TSHOP(JOB)	Total time in shop for job

```
C      THIS IS THE MAIN PROGRAM.IT INITIALIZES NON-GASP VARIABLES
       COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,NXC,NCOLCT,NHISTO,NCQ
      1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
      2OP,MXX
       COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
      1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
      25),SUMA(30,6),MLC(19),IX(8)
       COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(1500),JOP(1500)
       COMMON JOB,NOP,MGQ,MG,KODE,DT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,NJL,
      1TBA,UBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
       DIMENSION NSET(13,3500)
C
C      *** APPLICATION OF GASP II TO JOB SHOP SCHEDULING ***
C
C      JOB          = JOB NUMBER
C      NOP          = OPERATION NUMBER
C      MG           = MACHINE GROUP NUMBER
C      MGQ          = MACHINE GROUP QUEUE
C      DDATE        = DUE-DATE
C      JOP(JOB)     = NO. OF INCOMPLETE OPERATIONS ON JOB
C      TSHOP(JOB)   = TOTAL TIME IN SHOP FOR JOB
C      MACH(MG)     = NO. OF M/C IN M/C-GROUP MG
C      NAVL(MG)     = NO. OF AVAILABLE M/C IN M/C-GROUP MG
C      LAH          = LOOK AHEAD FEATURE
C      SLK          = SLACK OF SELECTED OPERATION
C      M            = NO. OF OPERATIONS PER JOB
C      NJOB         = JOB ARRIVAL COUNTER
C      KOUT         = JOB DEPARTURE COUNTER
C      NJL          = NO. OF JOBS LATE
C      KFILE        = STORAGE FILES FOR FUTURE OPERATIONS
C      KFILE        = 11 TO 19
C
C         *** PRIORITY RULES ***
C
C      KODE = 1 - FIRST COME FIRST SERVED
C      KODE = 2 - SHORTEST PROCESSING TIME
C      KODE = 3 - JOB SLACK
C      KODE = 4 - JOB SLACK PER OPERATION
C      KODE = 5 - JOB SLACK RATIO
C      KODE = 6 - EARLIEST DUE DATE
C
       KKK=0
  100  READ 10,KODE,LAH,KPRB,A,B,DATE,GSLK
   10  FORMAT(3I5,4F6.1)
       PRINT 10,KODE,LAH,KPRB,A,B,DATE,GSLK
       KKK=KKK+1
       DO 15 I=1,1500
       JOP(I)=0
   15  TSHOP(I)=0.
       DO 25 J=1,9
```

```
      MACH(J)=1
   25 NAVL(J)=MACH(J)
      TBA=0.
      TBD=0.
      KTA=10
      KTD=10
      KTBA=0
      KTBD=0
      NJOB=0
      KOUT=0
      NJL=0
      BT=0.
      CALL GASP(NSET)
      IF(KKK.EQ.KPRB) GO TO 20
      GO TO 100
   20 STOP
      END
```

```
      SUBROUTINE RANJOB(NSET)
C     THIS SUBROUTINE DETERMINES THE VARIOUS ATTRIBUTES FOR THE JOB
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOO
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TAOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENC(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),WLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NCP,MGC,MG,KODE,BT,TLATE,CDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,LBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION PTIME(20),NG(20),STIME(20)
      DIMENSION NSET(13,1)
C
C     ATTRIB(1) = SCHEDULED TIME OF NEXT EVENT
C     ATTRIB(2) = EVENT CODE(ARRIVAL-1; END OF SERVICE-2)
C     ATTRIB(3) = ARRIVAL TIME OF PRODUCT(JOB)
C     ATTRIB(4) = M/C GROUP REQUIREMENT
C     ATTRIB(5) = PROCESSING TIME/OPERATION
C     ATTRIB(6) = OPERATION NUMBER
C     ATTRIB(7) = JOB NUMBER
C     ATTRIB(8) = DUE DATE
C     ATTRIB(9) = SUM OF PROCESSING TIMES FOR REMAINING OPERATIONS
C     ATTRIB(10) = NUMBER OF OPERATIONS
C     ATTRIB(11) = NEXT MG REQUIREMENT
C
      ATTRIB(1)=0.
      ATTRIB(2)=1.
C
C     DETERMINE NUMBER OF OPERATIONS
C
      M=RNORML(4,3)
      DO 15 IL=1,M
      DO 15 IK=1,11
   15 SJOB(IL,IK)=0.
      TTIME=0.
C
C     ADD ONE TO ARRIVAL
C
      NJOB=NJOP+1
      JOP(NJOB)=M
      IF(NJOB.GT.30) GO TO 20
      BT=0.
      AT=0.05
      GO TO 22
C
C     SCHEDULE  AT  FOR JOB
C
   20 AT=ERLANG(1,1)
      ATTRIB(1)=BT+AT
```

```
   22 ATTRIB(3)=ATTRIB(1)
      BT=ATTRIB(1)
C
C     DETERMINE PROCESSING TIME FOR EACH OPERATION
C
      DO 25 K=1,M
      ATTRIB(6)=K
      PTIME(K)=ERLANG(2,2)
      PT=PTIME(K)
      CALL HISTOG(PT,0.,0.5,2)
   30 NG(K)=UNIFRM(A,B,3)
      IF(K.EQ.1) GO TO 40
      IF(NG(K)-NG(K-1)) 40,30,40
   40 ATTRIB(4)=NG(K)
      ATTRIB(5)=PTIME(K)
      ATTRIB(7)=NJOB
      TTIME=TTIME+PTIME(K)
      STIME(K)=TTIME
      DO 25 JJ=1,7
   25 SJOB(K,JJ)=ATTRIB(JJ)
C
C     DETERMINE DUE-DATE FOR JOB
C
      CDATE=ATTRIB(1)+DATE*TTIME
      ATTRIB(8)=CDATE
      ATTRIB(10)=M
      SJOB(1,8)=ATTRIB(8)
      SJOB(1,9)=STIME(M)
      SJOB(1,10)=M
      SJOB(1,11)=SJOB(2,4)
      L=M+1
      SJOB(L,4)=0.
      DO 35 I=2,M
      SJOB(I,8)=ATTRIB(8)
      ATTRIB(9)=STIME(M)-STIME(I-1)
      SJOB(I,9)=ATTRIB(9)
      SJOB(I,10)=M
      K=I+1
   35 SJOB(I,11)=SJOB(K,4)
      SJOB(M,9)=PTIME(M)
      RETURN
      END
```

```
      SUBROUTINE EVENTS(I,NSET)
C     THIS SUBROUTINE CONTROLS THE EVENTS
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NPISTC,NCC
     1,NCRPT,NCT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(2C,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJCB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOP,NCP,MGQ,MG,KODE,BT,TLATE,CDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,LBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION NSET(13,1)
C
C     EVENT 1 - ARRIVAL OF JOB
C     EVENT 2 - END OF PROCESSING
C
      GO TO(1,2),I
    1 CALL ARIVAL(NSET)
      RETURN
    2 CALL ENPROS(NSET)
      RETURN
      END
```

```
      SUBROUTINE ARIVAL(NSET)
C     THIS SUBROUTINE SCHEDULES AN OPERATION ON REQUIRED MACHINE AND
C     ALSO THE ARRIVAL OF THE NEXT JOB
      COMMON ID,IN,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOC
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),CTIME(19),SSUMA(15,
     25),SUMA(30,6),WLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJCB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NCP,MGQ,MG,KODE,BT,TLATE,CDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,LBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,CSLK
      DIMENSION NSET(13,1)
      MG=ATTRIB(4)
      JOB=ATTRIB(7)
      TBA=TNOW-TBA
      KTBA=KTBA+1
      ATBA=KTBA
      AIA=TNOW/ATBA
      IF(KTBA.EQ.KTA) GO TO 50
      GO TO 60
   50 PRINT 70,KTA,AIA
      KTA=KTA+10
   70 FORMAT(1H ,10X,I5,5X,F6.2)
   60 CALL HISTOG(TBA,0.,0.5,1)
      TBA=TNOW
      IF(NAVL(MG)) 100,2,3
C
C     SCHEDULE JOB ON AVAILABLE MACHINE
C
    3 NAVL(MG)=NAVL(MG)-1
      CALL TMSTAT(FLOAT(NQ(1)),TNOW,1,NSET)
      ATTRIB(1)=TNOW+ATTRIB(5)
      ATTRIB(2)=2.
      CALL FILEM(1,NSET)
      GO TO 30
  100 CALL ERROR(1,NSET)
C
C     JOB MUST WAIT FOR MACHINE
C
    2 ATTRIB(1)=TNOW
      ATTRIB(2)=1.
      MGQ=MG+1
      CALL TMSTAT(FLOAT(NQ(MGQ)),TNOW,MGQ,NSET)
      CALL FILEM(MGQ,NSET)
C
C     READ IN DATA FOR ARRIVAL OF NEXT JOB
C
   30 CALL RANJOB(NSET)
      M=ATTRIB(10)
```

```
      DO 40 IL=1,11
      ATTRIB(IL)=SJOB(1,IL)
   40 CONTINUE
      TW=SJOB(1,9)+SJOB(1,5)
      CALL TMSTAT(TW,TNOW,11,NSET)
      CALL FILEM(1,NSET)
C
C     STORE FUTURE OPERATIONS IN KFILE
C
      DO 20 IJ=2,M
      DO 10 IK=1,11
   10 ATTRIB(IK)=SJOB(IJ,IK)
      MG=ATTRIB(4)
      KFILE=MG+10
      CALL FILEM(KFILE,NSET)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ENPROS(NSET)
C     THIS SUBROUTINE SCHEDULES THE END OF PROCESSING EVENT AND COLLECTS
C     STATISTICS
      COMMON ID,IM,INIT,JEVENT,JMCNIT,MFA,MSTOP,MX,MXC,NCOLCT,NHIST,NQQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJCB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NOP,MGQ,MG,KODE,DT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,UBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION NSET(13,1)
      MG=ATTRIB(4)
      NOP=ATTRIB(6)
      JOB=ATTRIB(7)
      DDATE=ATTRIB(8)
      MGQ=MG+1
   75 FORMAT(1H ,40X,I5,5X,F6.2)
      JOP(JOB)=JOP(JOB)-1
C
C     TEST TO SEE IF ALL OPERATIONS ON JOB HAVE BEEN COMPLETED
C
      IF(JOP(JOB)) 400,10,20
   10 IF(JOB.LT.201) GO TO 31
C
C     ADD ONE TO DEPARTURE COUNTER
C
      IF(JOB.EQ.201) GO TO 11
      GO TO 12
   11 PRINT 21,TNOW
   21 FORMAT(1H ,10X,'START STATISTICS AT',F6.2/)
   12 KOUT=KOUT+1
      TBD=TNOW-TBD
      CALL HISTOG(TBD,0.,0.5,5)
      KTBD=KTBD+1
      ATBD=KTBD
      AID=TNOW/ATBD
      IF(KTBD.EQ.KTD) GO TO 15
      GO TO 45
   15 PRINT 75,KTD,AID
      KTD=KTD+10
   45 TBD=TNOW
      TSHOP(JOP)=ATTRIB(1)-ATTRIB(3)
C
C     COLLECT STATISTICS ON TLATE AND TSHOP(JOB)
C
      CALL COLECT(TSHOP(JOB),16,NSET)
      CALL HISTOG(TSHOP(JOB),5.,15.,4)
      TLATE=ATTRIB(1)-DDATE
```

```
      IF(TLATE.GT.0) SUMA(1,6)=SUMA(1,6)+1.
      NJL=SUMA(1,6)
      CALL COLECT(TLATE,1,NSET)
      CALL HISTOC(TLATE,-200.,20.,3)
      MJ=1
      DO 80 I=301,1501,100
      MJ=MJ+1
      IF(JOB.LT.I) GO TO 90
   80 CONTINUE
   90 MS=MJ+15
      CALL COLECT(TSHOP(JOB),MS,NSET)
      IF(TLATE.GT.0) SUMA(MJ,6)=SUMA(MJ,6)+1.
      CALL COLECT(TLATE,MJ,NSET)
      IF(KOLT.EQ.400) GO TO 500
      GO TO 30
  500 MSTOP=-1
      RETURN
C
C     REMOVE NEXT OPERATION FROM KFILE
C
   20 TJOB=JOB
      IF(LAF.EQ.0) GO TO 55
      CALL FIND(TJOB,7,1,5,KCOL,NSET)
      IF(KOOL.EQ.0) GO TO 55
      GO TO 30
   55 KFILE=ATTRIB(11)+10
      CALL FIND(TJOB,7,KFILE,5,KCOL,NSET)
      IF(KOOL.EQ.0) GO TO 100
      CALL REMOVE(KCOL,KFILE,NSET)
      MG1=ATTRIB(4)
      MGF=MG1+1
C
C     CHECK IF MACHINE IN MG AVAILABLE
C
      IF(NAVL(MG1)) 200,25,35
   25 CALL TMSTAT(FLOAT(NQ(MGF)),TNOW,MGF,NSET)
C
C     JOB MUST WAIT FOR MACHINE
C
      ATTRIB(1)=TNOW
      ATTRIB(2)=1.
      CALL FILEM(MGF,NSET)
      GO TO 30
C
C     SCHEDULE JOB ON AVAILABLE MACHINE
C
   35 NAVL(MG1)=NAVL(MG1)-1
      CALL TMSTAT(FLOAT(NQ(1)),TNOW,1,NSET)
      ATTRIB(1)=TNOW+ATTRIB(5)
      ATTRIB(2)=2.
```

```
      CALL FILEM(1,NSET)
      GO TO 30
C
C     CHECK FOR JOBS WAITING FOR SERVICE IN MG QUEUE
C
   31 TBD=TNOW-TBD
      CALL HISTOG(TBD,0.,0.5,5)
      KTBD=KTBD+1
      ATBD=KTBD
      AID=TNOW/ATBD
      IF(KTBD.EQ.KTD) GO TO 16
      GO TO 46
   16 PRINT 75,KTD,AID
      KTD=KTD+10
   46 TBD=TNOW
   30 IF(NQ(MGQ)) 300,60,40
   60 NAVL(MG)=NAVL(MG)+1
      RETURN
  100 CALL ERROR(2,NSET)
  200 CALL ERROR(20,NSET)
  300 CALL ERROR(30,NSET)
  400 CALL ERROR(40,NSET)
C
C     SELECT JOB/OPERATION FROM MGQ
C
   40 EPROS=0.
      CALL SELECT(NSET)
C
C     SCHEDULE NEXT JOB
C
      IF(EPROS.EQ.0.0) GO TO 95
      GO TO 65
   95 ATTRIB(1)=TNOW+ATTRIB(5)
      ATTRIB(2)=2.
      CALL FILEM(1,NSET)
      GO TO 85
   65 ATTRIB(1)=EPROS+ATTRIB(5)
      ATTRIB(2)=2.
      CALL FILEM(1,NSET)
   85 RETURN
      END
```

```
      SUBROUTINE SELECT(NSET)
C     THIS SUBROUTINE SELECTS THE NEXT OPERATION BASED ON PRIORITY RULE
      COMMON IB,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NOU
     1,NCRPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NCP,MGQ,MG,KODE,BT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,UBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION SAVE(11),DUMP(70,11)
      DIMENSION NSET(13,1)
      GO TO(1,1,3,3,3,1),KODE
C
C     KODE = 1 - FIRST COME FIRST SERVED
C     KODE = 2 - SHORTEST PROCESSING TIME
C     KODE = 6 - EARLIEST DUE DATE
C
C     REMOVE FIRST OPERATION FROM MGQ
C
    1 MFES=MFE(MGQ)
      CALL REMOVE(MFES,MGQ,NSET)
      RETURN
C
C     KODE = 3 - JOB SLACK
C     KODE = 4 - JOB SLACK PER OPERATION
C     KODE = 5 - JOB SLACK RATIO
C
    3 MFE1=MFE(MGQ)
      TSLK=GSLK
      CALL REMOVE(MFE1,MGQ,NSET)
      IF(NQ(MGQ)-1) 110,20,30
C
C     TWO JOBS WAITING IN QUEUE
C
   20 DO 15 I=1,11
   15 STORE(I)=ATTRIB(I)
      NOP=ATTRIB(6)
      DDATE=ATTRIB(8)
      SUMPT=ATTRIB(9)
      M=ATTRIB(10)
C
C     DETERMINE SLACK BASED ON PRIORITY RULE
C
      SLACK1=DDATE-TNOW-SUMPT
      IF(KODE - 4) 260,245,250
  245 SLACK1=SLACK1/(M-NOP+1)
      GO TO 260
  250 SLACK1=SLACK1/(DDATE-TNOW)
```

```
C
C       REMOVE SECOND OPERATION FROM MGQ
C
  260 MFE2=MFE(MGQ)
      CALL REMOVE(MFE2,MGQ,NSET)
      NOP=ATTRIB(6)
      DDATE=ATTRIB(8)
      SUMPT=ATTRIB(9)
      M=ATTRIB(10)
      SLACK2=DDATE-TNOW-SUMPT
      IF(KODE - 4) 280,270,275
  270 SLACK2=SLACK2/(M-NOP+1)
      GO TO 280
  275 SLACK2=SLACK2/(DDATE-TNOW)
C
C       DETERMINE JOB WITH MINIMUM SLACK
C
  280 IF(SLACK1-SLACK2) 35,35,40
C
C       RETURN OPERATION NOT SELECTED TO MGQ
C
   40 DO 45 I=1,11
      SAVE(I)=ATTRIB(I)
      ATTRIB(I)=STORE(I)
   45 STORE(I)=SAVE(I)
      CALL FILEM(MGQ,NSET)
      SLK=SLACK2
C
C       IS LOOK AHEAD FEATURE SCHEDULED
C
      IF(LAH.EQ.1) GO TO 150
      GO TO 400
  150 IF(KODE - 4) 450,120,130
  120 TSLK=TSLK/(M-NOP+1)
      GO TO 450
  130 TSLK=TSLK/(DDATE-TNOW)
C
C       CHECK TO SEE IF SLACK IS POSITIVE
C
  450 IF(SLK.GT.TSLK) CALL CHECK(NSET)
  400 DO 50 I=1,11
   50 ATTRIB(I)=STORE(I)
      RETURN
   35 CALL FILEM(MGQ,NSET)
      SLK=SLACK1
      IF(LAH.EQ.1) GO TO 140
      GO TO 410
  140 IF(KODE - 4) 460,160,170
  160 TSLK=TSLK/(M-NOP+1)
      GO TO 460
```

```
  170 TSLK=TSLK/(DDATE-TNOW)
  460 IF(SLK.GT.TSLK) CALL CHECK(NSET)
  410 DO 55 I=1,11
   55 ATTRIB(I)=STORE(I)
  110 RETURN
C
C     MORE THAN TWO JOBS WAITING IN QUEUE
C
   30 DO 60 I=1,11
   60 STORE(I)=ATTRIB(I)
      NOP=ATTRIB(6)
      DDATE=ATTRIB(8)
      SUMPT=ATTRIB(9)
      M=ATTRIB(10)
C
C     DETERMINE SLACK BASED ON PRIORITY RULE
C
      SLACK1=DDATE-TNOW-SUMPT
      IF(KODE - 4) 310,295,300
  295 SLACK1=SLACK1/(M-NOP+1)
      GO TO 310
  300 SLACK1=SLACK1/(DDATE-TNOW)
  310 K=NQ(MGQ)
C
C     REMOVE NEXT OPERATION FROM MGQ
C
      DO 70 J=1,K
      MFE2=MFE(MGQ)
      CALL REMOVE(MFE2,MGQ,NSET)
      NOP=ATTRIB(6)
      DDATE=ATTRIB(8)
      SUMPT=ATTRIB(9)
      M=ATTRIB(10)
      SLACK2=DDATE-TNOW-SUMPT
      IF(KODE - 4) 360,335,340
  335 SLACK2=SLACK2/(M-NOP+1)
      GO TO 360
  340 SLACK2=SLACK2/(DDATE-TNOW)
C
C     DETERMINE JOB WITH MINIMUM SLACK
C
  360 IF(SLACK1-SLACK2) 75,75,80
   75 DO 85 L=1,11
   85 DUMP(J,L)=ATTRIB(L)
      GO TO 70
   80 DO 90 JK=1,11
      DUMP(J,JK)=STORE(JK)
   90 STORE(JK)=ATTRIB(JK)
      SLACK1=SLACK2
   70 CONTINUE
```

```
C
C       PUT REMAINING JOBS IN PROPER M/C GROUP QUEUE
C
        SLK=SLACK1
        DO 210 J=1,K
        DO 200 IJ=1,11
  200   ATTRIB(IJ)=DUMP(J,IJ)
  210   CALL FILEM(MGQ,NSET)
        IF(LAB.EQ.1) GO TO 570
        GO TO 420
  570   IF(KODE - 4) 470,180,190
  180   TSLK=TSLK/(M-NOP+1)
        GO TO 470
  190   TSLK=TSLK/(ODATE-TNOW)
  470   IF(SLK.GT.TSLK) CALL CHECK(NSET)
  420   DO 220 I=1,11
  220   ATTRIB(I)=STORE(I)
        RETURN
        END
```

```
      SUBROUTINE CHECK(NSET)
C     THIS IS THE LOOK AHEAD SUBROUTINE . IF SOME JOB IS LATE,THEN IT
C     REPLACES THE PREVIOUSLY SELECTED JOB/OPERATION WITH THE LATE JOB
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHIST,NOC
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXQC(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NOP,MGO,MG,KODE,BT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,UBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION TRIX(10,11),ATRIX(11),BTRIX(11)
      DIMENSION NSET(13,1)
      K=0
      LSL=0
      GM=MGO-1
      STS=TNOW+STORE(5)
      DO 110 IK=1,10
      DO 110 IJ=1,11
  110 TRIX(IK,IJ)=0.
C
C     DETERMINE IF ANY JOB SCHEDULED FOR OPERATION ON MG
C
  100 CALL FIND(GM,11,1,5,KCOL,NSET)
      IF(KCOL.EQ.0) GO TO 200
      GO TO 300
  200 IF(K.EQ.0) RETURN
      GO TO 800
  300 K=K+1
C
C     REMOVE OPERATION FROM EVENTS FILE
C
      CALL REMOVE(KCOL,1,NSET)
      DO 10 I=1,11
   10 ATRIX(I)=ATTRIB(I)
      NOP=ATTRIB(6)
      DDATE=ATTRIB(8)
      SUMPT=ATTRIB(9)
      M=ATTRIB(10)
C
C     DETERMINE SLACK BASED ON PRIORITY RULE
C
      SLACK=DDATE-ATTRIB(1)-SUMPT
      IF(KODE - 4) 35,45,55
   45 SLACK=SLACK/(M-NOP+1)
      GO TO 35
   55 SLACK=SLACK/(DDATE-ATTRIB(1))
C
C     CHECK TO SEE IF SLACK IS NEGATIVE
```

```
C
   35 IF(SLACK.LT.0.0) GO TO 500
      GO TO 400
  500 STC=ATTRIB(1)
      ST=STS-STC
      IF(ST.GT.0.0) GO TO 600
      GO TO 400
  150 CALL ERROR(50,NSET)
  600 IF(SLACK.LT.SLK) GO TO 15
  400 DO 25 I=1,11
   25 TRIX(K,I)=ATRIX(I)
      GO TO 100
   15 DO 40 I=1,11
      TRIX(K,I)=ATRIX(I)
   40 BTRIX(I)=ATRIX(I)
      LSL=1
      SLK=SLACK
      GO TO 100
  800 IF(LSL.LE.0) GO TO 900
      CJOB=BTRIX(7)
      EPRCS=BTRIX(1)
      NFILE=GM+10
      DO 95 I=1,11
   95 ATTRIB(I)=STORE(I)
      CALL FILEM(MGQ,NSET)
C
C     FIND JOB SCHEDULED ON MG FOR NEXT OPERATION
C
      CALL FIND(CJOB,7,NFILE,5,KCOL,NSET)
      IF(KCOL.EQ.0) GO TO 150
C
C     REMOVE OPERATION FROM NFILE
C
      CALL REMOVE(KCOL,NFILE,NSET)
      DO 220 I=1,11
  220 STORE(I)=ATTRIB(I)
C
C     RETURN ALL OPERATIONS TO EVENTS FILE
C
  900 DO 75 IK=1,K
      DO 65 I=1,11
   65 ATTRIB(I)=TRIX(IK,I)
   75 CALL FILEM(1,NSET)
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(NSET)
C     THIS SUBROUTINE PRINTS OUTPUT REPORT
      COMMON ID,IM,IWIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHIST0,NOQ
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNQ(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(1500),JOP(1500)
      COMMON JOB,NCP,MGQ,MG,KODE,BT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,NJL,
     1TBA,UBAR,SLK,LAF,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION NSET(13,1)
   10 FORMAT(1H1,10X,'*** OUTPUT REPORT ***'//)
   20 FORMAT(1H ,10X,'NO. OF JOBS LATE = ',I5/)
   30 FORMAT(1H ,10X,'NO. OF JOBS IN SHOP = ',I5/)
   40 FORMAT(1H ,10X,'PRIORITY RULE = ',I5/)
   50 FORMAT(1H ,10X,'NO. OF JOBS COMPLETE =',I5/)
   60 FORMAT(1H ,10X,'MEAN SHOP UTILIZATION =',F8.3/)
   70 FORMAT(1H ,10X,'END OF SIMULATION AT',F10.2//)
      PRINT 10
      PRINT 40,KODE
      PRINT 20,NJL
      PRINT 30,NJOB
      PRINT 50,KOUT
      PRINT 60,UBAR
      PRINT 70,TNOW
      RETURN
      END
```

```
      SUBROUTINE DATAIN(NSET)
C     THIS IS A GASP SUBROUTINE WHICH READS IN DATA FOR GASP CONSTANTS
C     AND INITIALIZES ALL FILES ETC.
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NCC
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNC(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(2C,4),QTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSHOP(150C),JOP(150C)
      COMMON JOB,NOP,MGQ,MG,KODE,BT,TLATE,DDATE,EPROS,NJOB,KOUT,A,F,NJL,
     1TBA,UBAR,SLK,LAH,TBD,KTA,KTC,KTBA,KTBD,DATE,GSLK
      DIMENSION NSET(13,1)
      IF (NCT) 23,1,2
   23 CALL ERROR(95,NSET)
    1 NOT=1
      NRUN=1
      READ 101,NRUNS
      IF (NRUNS) 30,30,31
   30 STOP
  101 FORMAT(I5)
   31 PRINT 102
  102 FORMAT(1H1,25X,'*** SIMULATION OF A JOB-SHOP ***'//)
    5 READ 803,NPRAMS,NHISTO,NCOLCT,NSTAT,ID,IM,NOQ,MXC,SCALE
  803 FORMAT (8I5,F10.2)
      READ 104,MSTOP,JCLEAR,NORPT,NEP,TSTART,TSTOP,NSEED
  104 FORMAT(4I5,2F10.3,I5)
  105 FORMAT(8I5)
      READ 105,(IX(I),I=1,NSEED)
  103 FORMAT(19I3)
      JMONIT=0
      INIT=1
C
C     SPECIFY INN=1 FOR FIFO, INN=2 FOR LIFO
C
      READ 103,(INN(I),I=1,NOQ)
C
C     SPECIFY KRANK=RANKING ROW
C
      READ 103,(KRANK(I),I=1,NOQ)
C
C     SPECIFY NUMBER OF CELLS IN HISTOGRAMS NOT INCLUDING END CELLS
C
      IF(NHISTO) 23,24,25
   25 READ 105,(NCELLS(I),I=1,NHISTO)
   24 IF(NPRAMS) 23,10,8
    8 DO 9 I=1,NPRAMS
      READ 106,(PARAMS(I,J),J=1,4)
    9 PRINT 107,I,(PARAMS(I,J),J=1,4)
  106 FORMAT (4F10.4)
```

```
  107 FORMAT(1H-,10X,'PARAMETER NO.',I5,4F12.4)
   10 TNOW=TSTART
C
C      INITIALIZE NSET
C
       CALL SET(1,NSET)
       PRINT 1107,SCALE
 1107 FORMAT(1H-,10X,'SCALE =',F10.0)
       GO TO 299
C
C      SPECIFY INPUTS FOR NEXT RUN
C
    2 IF(NEP) 23,298,5
  298 INIT=1
       GO TO 10
C
C      READ IN INITIAL EVENTS
C
  299 CALL RANJOB(NSET)
       DO 600 IL=1,11
       ATTRIB(IL)=SJOB(1,IL)
  600 CONTINUE
       M=ATTRIB(10)
       CALL FILEM(1,NSET)
C
C      STORE FUTURE OPERATIONS IN KFILE
C
       DO 500 IJ=2,M
       DO 400 IK=1,11
  400 ATTRIB(IK)=SJOB(IJ,IK)
       MG=ATTRIB(4)
       KFILE=MG+10
       CALL FILEM(KFILE,NSET)
  500 CONTINUE
   15 IF (JCLEAR) 20,20,16
   16 IF (NCOLCT) 23,110,116
  116 DO 18 I=1,NCOLCT
       DO 17 J=1,3
   17 SUMA(I,J)=0.
       SUMA(I,4)=1.0E20
       SUMA(I,6)=0.
   18 SUMA(I,5)=-1.0E20
  110 IF (NSTAT) 23,111,117
  117 DO 360 I=1,NSTAT
       DO 370 J=1,3
  370 SSUMA(I,J)=0.
       SSUMA(I,4)=1.0E20
  360 SSUMA(I,5)=-1.0E20
  111 IF(NHISTO) 23,20,118
  118 DO 380 K=1,NHISTO
```

```
      DO 380 L=1,MXC
  380 JCELLS(K,L)=0
C
C     PRINT OUT FILING ARRAY
C
   20 JEVENT=100
      CALL MONTR(NSET)
      PRINT 403
  403 FORMAT (1H1,38X,'**INTERMEDIATE RESULTS**'//)
      RETURN
      END
```

```
      SUBROUTINE SUMARY(NSET)
C     THIS SUBROUTINE PRINTS OUT THE STATISTICS,HISTOGRAMS AND QUEUES
      COMMON ID,IM,INIT,JEVENT,JMCNIT,MFA,MSTOP,MX,MXC,NCOLCT,NHISTO,NQC
     1,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,OUT,SCALE,NSEED,TNOW,TSTART,TST
     2OP,MXX
      COMMON ATTRIB(11),ENQ(19),INN(19),JCELLS(5,22),KRANK(19),MAXNO(19)
     1,MFE(19),MLE(19),NCELLS(5),NQ(19),PARAMS(20,4),CTIME(19),SSUMA(15,
     25),SUMA(30,6),MLC(19),IX(8)
      COMMON MACH(9),NAVL(9),SJOB(16,11),STORE(11),TSNOP(1500),JOP(1500)
      COMMON JOB,NOP,MGQ,MG,KODE,BT,TLATE,DDATE,EPROS,NJOB,KOUT,A,B,MJL,
     1TBA,UBAR,SLK,LAH,TBD,KTA,KTD,KTBA,KTBD,DATE,GSLK
      DIMENSION NSET(13,1)
      PRINT 100
  100 FORMAT (1H1)
      PRINT 21
      IF (NCOLCT) 5,60,66
    5 PRINT 199
  199 FORMAT (///1H1,36X,'ERROR EXIT,TYPE 98 ERROR')
      STOP
   66 PRINT 23
C
C     COMPUTE AND PRINT STATISTICS GATHERED BY COLECT
C
    1 DO 2 I=1,NCOLCT
      IF(SUMA(I,3)) 5,62,61
   62 PRINT 63,I
   63 FORMAT(1H ,27X,I3,10X,'NO VALUES RECORDED'/)
      GO TO 2
   61 XS=SUMA(I,1)
      XSS=SUMA(I,2)
      XN=SUMA(I,3)
      AVG=XS/XN
      VAR=((XN*XSS)-(XS*XS))/(XN*(XN-1.))
      N=XN
      PRINT 24,I,AVG,VAR,SUMA(I,4),SUMA(I,5),N,SUMA(I,6)
    2 CONTINUE
   60 IF (NSTAT) 5,67,4
    4 PRINT 100
      PRINT 21
      PRINT 29
C
C     COMPUTE AND PRINT STATISTICS GATHERED BY TMSTAT
C
    8 DO 6 I=1,NSTAT
      IF(SSUMA(I,1)) 5,71,72
   71 PRINT 63,I
      GO TO 6
   72 XT=SSUMA(I,1)
      XS=SSUMA(I,2)
      XSS=SSUMA(I,3)
```

```
      AVG=XS/XT
      VAR=XSS/XT-AVG*AVG
      PRINT 30,I,AVG,VAR,SSUMA(I,4),SSUMA(I,5),XT
    6 CONTINUE
C
C     CALCULATE MEAN SHOP UTILIZATION
C
      BJOB=0.
      DO 50 I=1,10
      BJOB=BJOB+(SSUMA(I,2)/SSUMA(I,1))
   50 CONTINUE
      UBAR=SSUMA(11,2)*BJOB*SUMA(16,3)/(9.0*SSUMA(11,1)*SUMA(16,1))
   67 IF (NHISTO) 5,75,9
    9 PRINT 100
      PRINT 21
      PRINT 25
C
C     PRINT HISTOGRAMS
C
   11 DO 12 I=1,NHISTO
      NCL=NCELLS(I)+2
   12 PRINT 26,I,(JCELLS(I,J),J=1,NCL)
C
C     PRINT FILES AND FILE STATISTICS
C
   75 DO 15 I=1,NOC
   15 CALL PRINTC(I,NSET)
      RETURN
   21 FORMAT (1H-,39X,'**GASP SUMMARY REPORT**'//)
   23 FORMAT(1H-,41X,'** GENERATED DATA **'//1H ,27X,'CODE      MEAN
     1VAR.      MIN.      MAX.    OBS.     NJL.'//)
   24 FORMAT(1H ,27X,I3,4F9.2,I7,F10.2)
   25 FORMAT(/1H-,41X,'** GENERATED FREQUENCY DISTRIBUTIONS **'//1H ,17X
     1,'CODE',35X,'HISTOGRAMS'//)
   26 FORMAT(1H ,17X,I3,5X,23I4/)
   29 FORMAT(1H-,39X,'** TIME GENERATED DATA **'//27X,'CODE      MEAN
     1  VAR.     MIN.      MAX.     TOTAL TIME'//)
   30 FORMAT(1H ,27X,I3,4F9.2,F13.3)
      END
```

APPLICATION OF GASP II TO JOB-SHOP SCHEDULING

by

CHANDRA KUMAR PATEL

B.E. (Hons.), Electrical Engineering, University of Jabalpur

Jabalpur, (M.P.), India, 1967

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

1970

ABSTRACT

The advent of digital computers and in recent times, the development of general simulation languages such as SIMSCRIPT, GPSS and GASP II has made possible the study of the complex job-shop scheduling problem.

The effect of six priority rules on job-shop scheduling was investigated. The measures of performance considered include measures of inventory - number of jobs in the system - and measures of individual job progress - total time in shop and lateness.

Regardless of the measure of performance, the shortest processing time (SPT) rule was found to be the best. The slack based rules, though capable of better performance than the first come, first served (FCFS) and earliest due-date (DDATE) rules, are much more complicated and would require elaborate computing facilities for implementation.

The behavior of the SPT rule was studied to determine the best due-date (DATE) multiplier. With a value of 5-7 assigned to this multiplier, the percentage of jobs late was reduced to about 10% or less.

The slack based rules were found to reduce the lateness variance considerably. An effort to reduce the number of jobs late was made by implementing the look ahead and its variations. No significant improvement in schedule was observed. Future research on the look ahead should be directed along the lines suggested.

Finally, GASP II, although simple to use - being a FORTRAN based language, is slow in execution. With some modifications, it would compare more favorably with a general simulation language such as SIMSCRIPT which is more economical in terms of computing time.