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

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

LIST OF FIGURES

Page

LIST OF TABLES

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:

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

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.

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

<u>Flow-time - F_i</u>

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

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

<u>Lateness - L_i</u>

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 <u>tardiness</u> 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):

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

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