A LOADING AND BALANCING METHODOLOGY FOR JOB SHOP CONTROL

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

Joseph C. Irastorza

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

October, 1973

A LOADING AND BALANCING METHODOLOGY FOR JOB SHOP CONTROL

Approved:

Richard H. Deane, Chairman

Lynwood A. Johnson

Kong Chu

David E. Fyffe

Douglas C. Montgomery

Date approved by Chairman: 11/5/73

ACKNOWLEDGMENT

There are several persons who have given me the support and encouragement that I needed during my graduate studies.

I am particularly indebted to my advisor, Dr. Richard Deane, who gave generously of his time and provided the guidance needed at many critical points in the research; and to Dr. Lynwood Johnson, who was my advisor during most of my doctoral program. I also want to thank Dr. Robert N. Lehrer for providing the financial support and guidance which allowed me to get started in the program, as well as Drs. Kong Chu, David Fyffe, and Douglas Montgomery, members of my thesis reading committee.

I must also thank Kurt Salmon Associates for providing the arrangement under which I was able to obtain the time necessary to complete this research, and also because I did the initial part of the work reported here as company-sponsored research.

Last, but not least, I want to thank my wife Berta, without whose constant encouragement I could never have completed this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

LIST OF TABLES (Concluded)

Table | Page

LIST OF FIGURES

SUMMARY

This research presents a loading and balancing methodology for job shop control.

The importance of achieving shop balance in many types of manufacturing job shops is shown and a large number of indices for measuring balance in the job shop are developed. In addition to the balance measures, other measures of performance indicating ability to meet due dates and levels of work in process are also employed.

A method to provide good control in the operation of the job shop with respect to most measures of performance is presented. This method consists of setting up a pool of jobs prior to releasing them to the job shop and establishing a mathematical programming algorithm to select jobs to be loaded in the shop from the pool.

It is shown by this research that most of the balance measures calculated, as well as all of the work in process level measures, are significantly improved by the control methodology derived.

The job shop control methodology is also employed in conjunction with a variety of conditions such as shops with few interactions, job arrival distributions with static and dynamic means, and allowance of alternative machine operations.

A job shop simulation model is utilized to test the control methodology.

CHAPTER I

INTRODUCTION

A job shop is a collection of distinct machine groups and jobs which require processing on the machines. Each job will require processing on a certain number of the machines. Furthermore, a job could require any ordering of processing by the machines such that there is no common pattern of movement from machine to machine. The lack of a common routing for all jobs is characteristic of a job shop.

Job shop scheduling research has centered on the sequencing problem. This problem consists of determining the sequence in which units are to be processed at each of the machine centers. A solution to the job shop scheduling problem has not been developed, in fact, there is no general agreement as to what the solution should be. The most common approach has been to develop a sequencing rule and then to attempt to show that this rule performs adequately or better than other rules with respect to some measures of performance. The most commonly employed measures of performance have been concerned with the ability to meet due dates and with job flow times, but the level of work in process has also been used. A group of criteria that have been almost completely overlooked are the measures of shop workload balance.

The purpose of this research has been to develop and test a loading and balancing methodology for job shop control. Loading in this research is taken to mean the release of jobs to the shop. The definition of job shop balance is of primary concern in this research. It will be shown that different measures of shop balance may be devised for different types of shops. All of these measures will be analyzed in detail later. Briefly, however, it can be said that balancing the shop involves the scheduling of jobs in the shop so that a shop related measure (i.e. work output, queue size, etc.) is spread as evenly as possible over time or over all machines. The specific objective has been to improve the balance and work in process measures of the shop while still operating within due date constraints.

Most of the job shop scheduling research to date has attempted to optimize a measure of performance related to individual jobs such as the frequency with which assigned due dates are met, minimization of mean flow time, maximum flow time, etc. The attempts to obtain these objectives have usually consisted of the use of various dispatching rules. This research introduces a higher degree of shop control by the use of a different approach. This approach makes use of a pool of jobs in front of the shop and an algorithm to select the jobs to be released, or loaded, from the job pool to the shop.

The job pool concept is used explicitly in some industries (apparel, leather products) and implicitly in many others where a manufacturing lot is ready "on paper" to be placed in the shop long before this is actually done. This research, however, introduces additional realism in job shop research by developing a formal way of utilizing the job pool.

The algorithm employed to select jobs from the pool is primarily concerned with maintaining a balanced aggregate workload in the shop for each machine, while still allowing the jobs to meet their due dates. The utilization of this objective function is based on the fact that every job shop is physically set up to operate with a given workload mixture among the machines and at a certain overall load level and output. Any deviation from this workloading causes implicit or explicit costs, that is, load fluctuations from period to period for a given machine and/or across machines in a given period create costs of idle machinery and labor, costs of overtime premiums, or costs of performing some operations in other than their normal machines at increased costs.

The final measure of a methodology in an industrial situation lies on its ability to reduce costs and thereby increase profits. This research shows that a balanced shop workload allows the shop to operate at a lower work-in-process level. The work in process level and the improvement in shop operating conditions mentioned in the preceding paragraph result in very important cost reductions. The most important cost reduction in some industries, however, is due to the smaller risk caused by the ability to delay final production decisions while a job is in the job pool until the moment that the job is moved into the shop. A good example where this situation occurs is the fashion industry where the ability to postpone a cutting decision of a wide cuff pant for two or three weeks could mean the difference between $10,000 profit and $2,000 loss in a production lot. This is due to the fact that many fashion producers manufacture to stock in anticipation of store orders and styles in fashion ordered by stores and chains sometimes "die" in a matter of days.

The loading algorithm permits the loading frequency to be controlled directly by the production planner, thus allowing discrete releasing of the jobs at fixed time intervals (for instance once per shift). Also the weight attached in the algorithm to meeting due dates can be varied and the level of work in process in the shop as well as the degree of balance obtained can also be controlled.

The primary purpose of this research has been to develop a methodology to provide better job shop balance and control. This led to the formalization of the job pool concept, the development of the discrete releasing approach and the mathematical programming job loading algorithm. It is evident that many balance measures could be formulated with different ones being appropriate for different shops and conditions. Therefore, several balance measures and their usefulness have been identified by this research. It must also be recognized that other constraints face the job shop manufacturing facility in addition to the "balance" conditions. These are the ability to meet assigned due dates and the level of work in process inventory.

Three basic types of performance measures are employed in this research. They are:

1. Shop Balance Oriented Measures
2. Measures Related to Work in Process Levels
3. Measures Dealing with the Ability to Meet Job Due Dates

Several criteria which can be classified into the three groups above are studied throughout this research and their relationships under different loading (scheduling) and dispatching rules are analyzed. No attempt has been made to assign weights to the various measures of performance employed. The problem is approached in this way because the cost structure for the various criteria will vary from shop to shop and probably even within the same shop at different times.

The loading and balancing methodology developed by this research was tested through the use of a job shop simulation model. The reason for the use of a simulation model for this purpose is the scarcity of theoretical results in the queuing network area. In fact, Conway (1967), has stated that "a harsh critic could conclude that there are no network queuing results."

The following chapters present the results of this research. Chapter II gives a description of the job shop and the measures of performance studied and it also provides a review of relevant literature. In Chapter III a number of balance measures are presented and their potential applications are discussed.

Chapter IV provides a description of the loading and balancing methodology developed by this research. These include the maintenance of a job pool, discrete job selection, and job loading from the pool. The loading algorithm is formulated and its control aspects are explained. Also a loading heuristic as an alternative to the loading algorithm is shown.

Chapter V presents the testing vehicle. It starts with a description of the job shop parameters and the dispatching rules and then gives a brief explanation of the job shop simulator which is a GASP II simulation program. The chapter includes the description of the simulation programs with reference to the flow charts. Some variations of the simulation program are used to investigate special conditions such as a job arrival distribution with a static and a dynamic mean, shops with few interactions, and shops with a non-symmetric transition matrix.

Chapter VI is concerned with the validation of the simulation and the design of the experiment including the validation and testing of the random number generator. Chapter VII presents the results obtained when using the job pool and the loading algorithm and analyzes the comparison of these results with the output of an "uncontrolled" shop. Chapter VIII presents some additional theoretical models.

Finally, Chapter IX presents the conclusions and provides an overall interpretation of the research as well as presenting suggestions for possible extensions of the work presented here.

CHAPTER II

JOB SHOP FRAMEWORK AND LITERATURE REVIEW

2.1 Job Shop Framework

2.1.1 Brief Description of the Job Shop

A job shop is considered in this research to be a production shop with distinct machine centers which performs different types of operations. There are one or more machines in each one of these machine centers. Therefore, the machine centers may have different capacities.

A job to be performed by the shop requires several operations for completion and therefore may require time in different machine centers. Each job may also follow a different machine operation sequence before it is completed and in fact the same machine center may appear more than once in the operation sequence.

Jobs become available to the shop in a continuous stream with random interarrival times. They enter a job pool from which they are selected in groups every period (for instance, daily) to be loaded into the manufacturing shop. At this time they are allowed to enter the queue at their first respective operation and the jobs then remain in the shop until completion.

A ten-machine shop was utilized for most of the investigations in this research. Ten machines are enough to allow the interactions and complexities of a "real world shop" to develop while at the same time the shop size is small enough to maintain the computer time required in the simulation within reasonable bounds.

The job arrival process was generated by using exponential interarrival times and the processing time per operation consisted also of samples from an exponential distribution. A job due date was assigned to each job as it entered the shop with the due date being a function of the work content of the job.

The above shop conditions were deemed reasonable and are generally representative of shop conditions in several industries such as the apparel style shop. It has not been the intent of this research, however, to reproduce a particular shop, but rather to model a shop structure which is a reasonable image of existing shops in many industries in order to evaluate the effects of the proposed shop loading methodology. In fact, Conway and others (1967, p. 220) state 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 shop characteristics employed in this research have been generally accepted by previous research studies by Baker and Dzielinski (1960), Nanot (1963), Conway, et al. (1965, 1967), Gere (1966), Bulkin et al. (1966), and Deane (1972).

It is evident that in actual shops the arrival pattern does not follow exactly the exponential distribution. Job arrivals tend sometimes to be grouped together more closely than would be indicated by the exponential interarrival times. Other times the job arrivals are more widely spaced than is justified by the mean time between arrivals being used. These two conditions seem to occur in cycles and therefore in order to add realism to the job arrival pattern an option has been provided in the job shop simulator to allow the mean time between arrivals to fluctuate.

This in effect creates a mean time between arrivals that is dynamic with respect to time rather than static as has been commonly done. The time for the next arrival is obtained by sampling an exponential distribution with a dynamic mean time between arrivals.

Other assumptions used in developing the models presented here are common to most job shops. These assumptions serve to simplify the study and the most important ones are listed below:

a) Each machine is continuously available for assignment
b) Each operation can be performed by only one type of machine in the shop
c) Each job can be processed on only one machine at a time
d) Jobs are strictly-ordered sequences of operations, without assembly or partitions
e) Pre-emptions of jobs on machines is not allowed
f) There is no set-up required for operations
g) Each machine can handle at most one operation at a time
h) A job is considered immediately available for its next operation when it finishes the current one.

2.1.2 Traditional Scheduling and Job Shop Classifications

Scheduling problems can be classified in several ways. The most common classifications are the following:

a) Classifications according to the job arrival pattern
    1 - There is a fixed finite number of jobs in the shop. This is normally called the static case.
    2 - The jobs arrive to the shop in a continuous stream and at

random intervals. This case is called a dynamic job shop.

b) Classifications according to the number of machines in the shop

1 - There is only one machine in the shop.

2 - There are two or more machines.

c) Classification according to the type of set-up times considered

1 - The set-up times are independent of the job sequence and therefore can be incorporated within the production time.

2 - The set-up times are dependent on job sequence.

d) Classification according to the job routing in the shop

1 - All jobs have identical routings through the shop. The shop is then called a flow shop.

2 - The jobs have non-identical routing through the shop. This is a job shop and if the routing is completely random, the shop is called a pure job shop.

e) Classification according to the job dispatching rule used

1 - First come first served

2 - Random

3 - Earliest due date

4 - Shortest processing time

5 - Minimum slack

6 - Dynamic slack per operation

7 - Minimum slack per operation

8 - Minimum work in next queue

9 - Longest processing time, etc.

f) Classification according to measures of job or shop performance

1 - Ability to meet specified completion dates

2 - Variance of the lateness distribution

3 - Average job flow time

4 - Maximum job flow time

5 - Work in process in total hours of work in the shop

6 - Work in process in hours of work done in the shop, etc.

The studies conducted in this research fit the following classification conditions given above: a.2, b.2, c.1, d.2, e.1, e.4, e.6, e.8, f.1, f.2, f.5, and f.6.

2.1.3 Measures of Performance

Measures of performance of primary interest in this research are those dealing with workload balance. However, the discussion of these will be deferred until balance measures are defined in the next chapter.

Other measures of performance obtained by the job shop simulator are:

Measures of performance related to work in process

a) Average work in process in hours of work in the shop.

b) Average number of jobs in the shop.

c) Average number of operations performed for jobs in the shop.

d) Average hours of work done for jobs in the shop. This measure gives an indication of the investment made in work performed for work in process in the shop.

e) Average queue length.

Measures of performance related to the ability to meet due dates

f) Average job lateness.

g) Variance of the lateness distribution.

h) Average job tardiness.

i) Average tardiness variance.

No attempt has been made to develop a composite performance criterion by assigning weights to the various measures of performance since these would vary from shop to shop. Instead, a subset of the above measures, based on their importance and/or how representative of their group they are, has been selected for statistical analysis and detailed study. The measures selected were a, d, g, and h. The average work in process in hours of work in the shop was selected because it is probably the most commonly accepted measured of work in process in both industry and in the literature. The average hours of work done for jobs in the shop is of particular interest to this research because it is an objective of the work to prove that the controlled shop loading methodology keeps away from the shop jobs that would be partially completed otherwise, and not only jobs at the end of their first queues.

The variance of the lateness distribution was selected because in many job shop situations the inability to predict completion dates, that is, the variability of the completion date with respect to the due date causes more problems than missing the due date itself. Finally, the average job tardiness was selected because it is the statistic most commonly accepted to measure due date performance.

2.2 Literature Review

The classifications provided before serve to give an underlying structure to the literature. The review, however, will be presented according to methodology, shop structure, and measure of performance used.

The number of articles in the general area of shop scheduling is extremely large. This research lists close to 200, Conway (1967) gives 202, Buffa and Taubert (1972) list 61, Day and Hottenstein (1970) include 162, and although there is some duplication in these sources, there are many other sources. Most of these articles deal with queuing problems, the problem of sequencing jobs in a static flow shop or they are concerned with simulation studies of a dynamic job shop using a job related measure of performance.

Very good and comprehensive reviews in the sequencing and scheduling area are provided in the book by Conway (1967) and the paper by Day and Hottenstein (1970). Other reviews given by Elmaghraby (1968), Mellor (1966), and Sisson (1959) are also available. The literature review presented here will not attempt to duplicate these reviews. An attempt, however, is made to highlight those articles from the literature which are directly relevant to this research as well as providing a sketch of the breadth of shop scheduling literature.

2.2.1 Analytical Approaches (Flow Shops, Restricted Problems)

The analytical approaches that have been used are algebraic, integer programming, dynamic programming, enumerative, branch and bound and graph theoretic. The initial ones were primarily algebraic and the majority of the most recent ones have used branch and bound.

Johnson (1954) considered the problem of minimizing maximum flow time in a two machine flow shop. In this frequently cited paper he developed a rule to minimize the maximum flow time.

Smith (1956) did extensive work on the one machine job problem. Among other results he showed that the mean flow time is minimized by sequencing the jobs in order of non-decreasing processing time. It is also true in this case that SPT sequencing minimized mean lateness and mean number of jobs in the shop.

Smith and Dudek (1967) developed an algorithm for makespan minimization in a flow shop with no passing.

Ignall and Schrage (1965) and Lomnicki (1965) are generally credited with first using the branch and bound approach to the solution of flow shop problems. The basic idea in this approach is to partition the set of possible solutions into subsets and to use a lower bound of the schedule time in a solution subset to eliminate some of the subsets. Backtracking, of course, is required to guarantee optimality. Burton and McMahon (1967) expanded the previous work by Ignall and Schrage (1965) by introducing a job based bound in addition to the machine based bound.

Ashour (1969) applies a graph theoretic approach to the flow shop problem. This approach generates a sequence of "j" jobs in "j" iterations regardless of the number of machines involved.

In another article, Ashour (1970) presents a comparative evaluation of flow shop scheduling techniques and concludes that branch and bound techniques without backtracking give the best results at present when computer time is considered.

2.2.2 Analytical Approaches (General or Job Shops)

There have been several attempts at formulating the shop scheduling problem in terms of a mathematical programming model. Probably the most compact one is the one by Manne (1960). The formulation by Manne can handle several objective functions including minimization of mean flow time, maximum flow time, or mean tardiness. This approach, however, is of theoretical interest only since it becomes computationally prohibitive for even very small problems.

Brooks and White (1965) use the branch and bound approach to solve the M machine, N job, job shop problem. Several measures of performance are considered but the paper concentrates on minimizing the time for completion (makespan) and minimizing average lateness.

Greenberg (1968) presents an approach for minimizing makespan or idle time in the M machine job shop. The approach formulates the shop as an integer programming problem and then uses branch and bound to solve it by transforming the integer programming problem into a series of linear programs to be solved at every branch. Charlton and Death (1970) have developed a branch and bound approach that can be applied to a wide variety of machine scheduling problems and they show how the algorithm reduces to methods previously published under special conditions.

There have been some analytical papers that approach a job shop as a network of queues. The most significant result in these papers has been to develop sufficient conditions under which a network of queues can be treated as an aggregation of independent queues, Jackson (1963). Burke (1972) presents a summary of the results obtained in this area.

2.2.3 Computer Simulation Approaches

Computer simulation has been practically the only approach used to study the dynamic job shop problem.

The earlier work in this area was done by Jackson (1957), and Le Grande (1963), Baker and Dzielinski (1960), Nanot (1963), Bulkin, Colley, and Steinhoff (1966), and others. The general objective in most of these studies was to compare the effectiveness of dispatching rules with respect to job related measures of performances.

Good reviews have been provided by Sisson (1959) and Moore and Wilson (1967). Buffa and Taubert (1972) provide a good summary of several of the above articles.

Conway, Maxwell, and Miller (1967) reported a significant amount of new work done at Cornell University as well as an excellent general discussion of simulation approaches to the shop scheduling problem. The general conclusion of the simulation work done is that the shortest processing time rule minimizes the mean flow time, mean number of jobs in the shop, and the mean lateness. Most of the rules considered, however, were local dispatching rules, that is, rules that did not consider shop conditions except at the individual queue where the dispatching decision took place.

Emery (1969) introduces a job shop simulation program in which various dispatching rules are combined through the use of weights into a single job dispatching criterion and results slightly better than those produced by any of the individual rules are obtained. However, some of the more interesting rules are not included and the test results given are very limited.

2.2.4 Articles of Miscellaneous Interest

Several articles of general interest that could not be easily classified in any of the previous sections will be presented next.

Harding, Gentry, and Parker (1969) proposed a heuristic sequencing rule based on job due date, processing time, and status of the work center where the job will go next. They reported improvements in the percentage of jobs meeting their scheduled dates in an actual shop, but no controlled experiment was provided to allow an evaluation of results.

Ebert (1972) analyzes the performance of intuitive decision making when compared with a mathematical model. A controlled experiment was used and it was concluded that in this case (aggregate production scheduling) the model decisions were superior to the intuitive decisions and that furthermore the superiority increases as the time-horizon complexity increases. No rigorous study of this type comparing mathematical models to an intuitive dispatcher was found in a job shop environment.

Von Lanzenauer (1970) presents a model to attack the scheduling as well as the sequencing problem. By scheduling he means how much and when to produce. The model is a 0-1 integer programming formulation with the objective of minimizing total costs. The terms in the objective function include set up, inventory, and shortage costs. This is a welcome attempt at integrating these two problems, however, the model in its present form cannot be utilized for realistic problems as the author recognizes.

Eilon and Christofides (1971) analyze a particular type of loading problem. This problem consists of allocating n objects or items of magnitude Q_i to boxes, each box having a capacity C, in such a way that the capacity constraints are not violated and the number of boxes required is a minimum. They presented a zero-one programming solution and a heuristic algorithm and demonstrated that the algorithm obtained the optimal solution almost all the time. Greenberg (1972) uses this loading algorithm to allocate workloads over a number of identical stations or workers under static job conditions such that the resulting workloads are nearly equally balanced.

Ghare, Givens, and Torgensen (1969) presented a paper, "A Machine Release Scheme for the Job Shop," which considered several machines at each work station and the effect of operator learning on the performance of the shop. The job shop was viewed as a network of queues, with all the required restrictive assumptions. A scheme was developed to release machines to other assignments as learning takes place so as to maintain a relatively constant level of machine utilization. This is a very interesting paper, but not directly applicable to the work in this thesis. The title of the article is so closely related to this research, however, that a discussion of the article was deemed necessary.

Franklin (1969) proposes a framework for job shop research. He states that the value of a shop's output depends upon both the technique or rule employed for scheduling and the product mix or aggregation of jobs upon which the technique operates. Franklin also claims that there are four basic components in every experimental or theoretical approach to the job shop problem. These are:

a) the model of the process

b) the scheduling technique

c) the product mix, the particular problem under analysis

d) the objective function.

He further claims that the product mix is instrumental to every analysis and that the others, except for the objective function, can be expressed in terms of the variables describing this product mix. This is a good attempt at providing a framework, but should have been complemented with a presentation of several articles and problems in order to "test" the structure proposed.

Day and Hottenstein (1970) present a comprehensive review of sequencing research in which a classification scheme is provided. The primary classifications proposed are the following:

1. Numbers of component parts comprising a job
    a) Single component jobs
    b) Multi-component jobs which require assembly operations
2. Production factors possessed by the shop
    a) Machines
    b) Labors and machines
3. Jobs available for processing
    a) N jobs to be sequenced where N is finite (static problem)
    b) An undetermined number of jobs arrive continuously, but randomly at the shop for service.

Most of the articles in the literature are of the (1a-2a-3a) and (1a-2a-3b) variety and this research fits in the (1a-2a-3b) group. An additional scheme for classifying problems of these two varieties is also provided in the article. This is done by considering one machine and multimachine problems as well as the variations allowed by routing (flow shops, job shops). The article then proceeds to examine the accomplishments and the methods used to solve problems in each one of the cases citing several papers in each classification. The conditions that have been used in the literature related to the dynamic job shop problem, which is of special interest to this research, are covered in detail by Day and Hottenstein on pages 17-26. These conditions deal with job arrivals, processing times, shop size, job routing, assignment of due dates, types of priority rules used, initialization of job shop simulations, and statistical methods used in the study of job shops.

2.2.5 Articles of Direct Interest

Ackerman (1963, 1964) presented the idea that a job spends most of its time waiting in queues rather than being processed and that job flow time is therefore highly correlated with the number of operations in a job and not with the job processing time. He used this idea to develop a scheduling procedure which he called "Even Flow" based on scheduling a job by allocating one time period for each job operation starting backwards from the job due date. Ackerman also presented some simulation results which backed his claim of reduced lateness when compared to Random, FIFO, and SPT dispatching rules. However, the comparisons are not strictly valid since the Even Flow system allowed machine overtime in some cases.

Schussel (1968) presents an algorithm directed at work balance and in-process inventory minimization based on a matrix concept with machines and days. The algorithm starts from the due date of any job back, trying to fill in slots of production time. The objective function used is quite complex and could probably be simplified while retaining the main ideas in the article. No application results of this algorithm were presented.

The work by Deane (1972) and the resulting article by Deane and Moodie (1972) bear the closest relationship to this research. Deane developes a Balance Index to be used as the primary measure of performance. This machine work balance index (MWB) measures the deviation of machine utilization from its average every period. Deane then develops what he calls a "flow controlled scheduling methodology." This consists of a periodic search procedure which directly attempts to guide work to underloaded machines, that is, jobs that can make large contributions in their next operation to underloaded machines are given high priorities in their present operations. The search procedure is a dynamic one in that any job given a high priority has an effect in the selection of all future jobs.

An additional balance index was developed by Deane. This was the shop Workload Balance Index (SWB) which is based on variations in the utilization of the shop as a whole. The search dispatching rules offered significant improvements in the machine workload balance index, but not on the shop workload balance index. Deane, however, allowed all the jobs to get in the shop as soon as they became available and did not recognize the advantages of maintaining a job pool for providing additional flexibility in the operation of the shop.

CHAPTER III

WORKLOAD BALANCE MEASURES

3.1 General

The two most important considerations regarding the objective of balancing a job shop are the method selected to measure balance and the determination of the effect of "balancing" the shop, if any, on other measures of shop performance. The exploration of the first point is the primary objective of this chapter, while the second one will be briefly discussed below and in a more quantitative basis in Chapters IV and VIII.

The effect of balance on other measures of performance must be investigated because the only obvious objective functions in a shop are minimization of costs or maximization of profits. It is difficult, however, to construct models explicitly in terms of those objectives because many subjective evaluations are required. For this reason, indirect measures of performance are used.

It would be possible, of course, to assign weights and conversion factors to these indirect measures of performance so that total costs could be obtained, but the results would depend heavily on the weights and conversion factors used.

It is usually preferable, however, to establish a logical relationship between the indirect measure and the final objective (cost minimization or profit maximization) and then proceed to devise methods to improve or optimize the indirect measure of performance directly. This method has been followed in this research.

For example, the ability to meet due dates is among the most common measures of performance used in a job shop. This is justified because of the large penalty cost or opportunity cost resulting from the probable loss of business if due dates are consistently missed. This type of approach is the same one that is used in justifying the importance of shop balancing. The relationship between balancing and the final objectives have been briefly discussed in Chapter I and will be further considered in Chapter IV.

Balancing must be machine oriented or time period oriented. Machine oriented balance measures recognize the fact that shops are designed with a certain product mix in mind and operate most efficiently under those conditions. These measures do not allow an underloaded machine to cancel the effects of an overloaded machine and they also detect the changes of a machine load over time, even when these changes are due to an overall shop condition. Time period oriented balance measures place primary attention to the efficiency improvements that can be achieved when shop productivity and/or loading is predictable over time. These measures do not allow the index to be influenced by overall shop changes from period to period.

It is important to note that the concept of workload balancing necessarily implies the division of the planning horizon into scheduling periods. This, of course, is a very realistic assumption. Most shops work within a definite scheduling period such as a shift, day, week, etc. In practical situations the length of the scheduling period will correspond to the "period of accountability" imposed upon the shop by management. Management will usually require efficiency or cost statistics for this period.

The length of the time period to be utilized must be given careful thought because a very long one will hide significant fluctuations while a short time period will place too much weight on unavoidable variations.

A scheduling period of eight hours was chosen for the investigations presented in this research. This time period was selected as being reasonable with respect to the other shop parameters that were employed. A longer time period will require that more jobs be loaded in the shop every period so that the amount of work released to the shop per unit time stays fairly constant. The longer scheduling period will also afford less opportunities to correct out of balance conditions existing in the shop. A shorter scheduling period will have the opposite effect, but it will require more computer time.

3.2 Notation

The following notation will be used in developing the balance measures which are presented in the rest of this chapter.

p number of scheduling periods in the scheduling horizon

m number of machines or machine centers

u_{ij} work done by machine i in period j

$\bar{u}_i$ average work done by machine i over all periods j

$$\bar{u}_i = \sum_{j=1}^{p} \frac{u_{ij}}{p}$$

$\bar{u}_j$ average work done in period j over all machines i

$$\bar{u}_j = \sum_{i=1}^{m} \frac{u_{ij}}{m}$$

v_{ij} average queue size in number of jobs for machine i in period j

$\bar{v}_i$ average queue size in number of jobs for machine i over all periods

j $$\bar{v}_i = \sum_{j=1}^{p} \frac{v_{ij}}{p}$$

$\bar{v}_j$ average queue size in number of jobs in period j over all machines

i $$\bar{v}_j = \sum_{i=1}^{m} \frac{v_{ij}}{m}$$

H_{ij} average queue size (plus work remaining on job being processed) in number of hours of work for machine i in period j. This is the average work in process (work to be done) for machine i in period j.

$\bar{H}_i$ average queue size in hours of work for machine i over all periods j.

$\bar{H}_j$ average queue size in hours of work in period j over all machines i.

P_{ij} aggregate load for machine i, in hours, including jobs that have just been placed in the shop at the beginning of scheduling period j.

$\bar{P}_i$ average aggregate load (in all queues and machines) in hours of work for machine i over all periods j.

$\bar{P}_j$ average aggregate load or work in process in period j over all machines i.

C_i desired aggregate load for machine i. This aggregate load includes not only the load given by the queue in front of machine i, but also the future load for machine i given by jobs in other queues.

q_{ij} maximum queue size in number of jobs for machine i in period j.

r_i desired queue size in number of jobs for machine i.

O_{ij} work output by machine i, in hours, performed on jobs leaving the machine during period j. The difference between this variable and u_{ij} is due to work done on a job during a period in which the job was not completed by machine i.

$\bar{O}_i$ average output of machine i over all periods j. As the number of periods increases the percentage difference between $\bar{O}_i$ and $\bar{u}_i$ becomes very small,

$$\lim_{p \to \infty} \frac{\bar{u}_i - \bar{O}_i}{\bar{O}_i} \le \epsilon$$

where

$$\bar{O}_i = \sum_{j=1}^{p} \frac{O_{ij}}{p}$$

$\bar{O}_j$ average output in period j over all machines i

$$\bar{O}_j = \sum_{i=1}^{m} \frac{O_{ij}}{m}$$

A_{ij} amount of work in hours arriving to machine i in period j.

$\bar{A}_i$ average work in hours arriving to machine i over all periods j.

$\bar{A}_j$ average work in hours arriving in period j for all machines.

3.3 Definitions of Shop Balance Measures

Balance in a job shop can be measured in many different ways. A large number of balance measures will be defined in this section, but the ones discussed here are not the only ones.

There is not one best measure of shop balance, but rather each one of the measures given applies best to a specific type of shop, product, or management structure. Other conditions will certainly be identified in the future that can be measured best in terms of balance statistics not included in this group.

These balance measures are based on work done (or equivalently, machine or shop utilization), work output, queue size, or work arrival. The balance measures presented in paragraphs 3.3.1 and 3.3.2 are due to Deane (1972) and the rest of them are presented here for the first time. Table 1 presents a brief summary of all the balance measures considered.

3.3.1 Machine Work Balance Index (MWB)

The variance in the work done by each machine over all time periods is calculated. The word "variance" is used here to indicate the form of the formula employed and does not imply any statistical meaning.

An overall index is then obtained by averaging over all machines. Let B_i be the machine index for machine i.

$$B_i = \sum_{j=1}^{p} \frac{(u_{ij} - \bar{u}_i)^2}{p}$$

$$MWB = \frac{\sum_{i=1}^{m} B_i}{m}$$

Objective: Minimize MWB

This index can be used when it is important to consider the utilization of individual work centers without allowing a cancelling effect. That is, this index will detect the variations of a machine production over time and will not allow the over production of one machine in a time period to compensate the underutilization of another machine in the

Table 1. Measures of Shop Balance

Paragraph No.	Name	Index Symbol	Unit Symbol	Characteristics, Definition	Comments
3.3.1	Machine Work Balance Index	MWB	B_i	Variance of work done per period by each machine	It considers the work done by individual work centers.
3.3.2	Shop Work Balance Index	SWB	--	Variance of the utilization or work done by the shop as a whole	It measures variability of work done by entire shop; but does not detect variations in individual machines.
3.3.3	Period Work Balance Index	PWB	BP_j	Variance of the work done by all machines for each time period	It measures changes in work done by machines within a period, but ignores differences from period to period.
3.3.4	Machine Output Balance Index	MOB	BO_i	Variance of output per period of each machine	Similar to 3.3.1 but output rather than work done is used. Could be useful when output variability is undesirable.
3.3.5	Shop Output Balance Index	SOB	--	Variance of the output by the shop as a whole over time	Similar to 3.3.2 but output rather than work done is employed.
3.3.6	Period Output Balance Index	POB	PO_j	Variance of the output by all machines for each time period	Similar to 3.3.3 but output rather than work done is employed.
3.3.7	Machine Queue Balance Index	QWB	Q_i	Variance of queue size in number of jobs per period for each machine	Used when it is desired to keep track of WIP stability at the machine level.
3.3.8	Shop Queue Balance Index	SQB	--	Variance of the number of jobs in the shop over time	Used when it is desired to keep track of the WIP stability for the shop as a whole.
3.3.9	Period Queue Balance Index	PQWB	PQ_j	Variance of the queue size over all machines for each time period	Used in a manner similar to 3.3.7, but ignores WIP variations over time.

Table 1. (Continued)

Paragraph No.	Name	Index Symbol	Unit Symbol	Characteristics, Definition	Comments
3.3.10	Measures Related to Hours of Work in Queue	---	---	These measures are similar to 3.3.7, 3.3.8, 3.3.9, but hours of work instead of number of jobs are used	
3.3.11	Machine Work in Process	AMWP	$MWIP_i$	Variance of the aggregate WIP hours per period for each machine	The difference with measure 3.3.7 is that the total work in the shop for a machine rather than work in that machine queue only is considered.
3.3.12	Shop Work in Process Balance Index	SWIP	---	Variance of the aggregate WIP in the whole shop over time	Similar to 3.3.8.
3.3.13	Period Work in Process Balance Index	APWP	$PWIP_j$	Variance of the aggregate WIP over all machines for each time period	Consideration of variations in the aggregate load could be important as a means of looking beyond immediate period.
3.3.14	Desired Loading Measure	D	D_i	Deviation of aggregate shop load for each machine from management target	Useful as an objective function in trying to improve other balance and shop measures.
3.3.15	Maximum Queue Deviation Index	QD	QD_i	Maximum excess in queue size over the desired amount set up by management	Could be used in shops where a penalty must be paid if queue lengths exceed some amount.
3.3.16	Period Idle Time	PIT	IDT_j	Maximum idle time for any machine in a period	Useful when there is no operator flexibility and operator idleness in a period should not be excessive.

Table 1. (Concluded)

Paragraph No.	Name	Index Symbol	Unit Symbol	Characteristics, Definition	Comments
3.3.17	Machine Idle Time	MIT	IDT_i	Maximum idle time for a machine over all periods	Useful if machines require adjustments when left unused over a certain length of time.
3.3.18	Machine Arrival Balance Index	MAB	A_i	Variance of the work arrived per period to each machine	This index is a function of the work arrival pattern, loading mechanism, and dispatching mechanism.
3.3.19	Shop Arrival Balance Index	SAB	--	Variance of the work arrival to the shop as a whole over time	Similar to 3.3.2 but work arrival rather than work done is employed.
3.3.20	Period Arrival Balance Index	PAB	DA_j	Deviation between actual arrivals to a machine in a period and the machine average	Useful in a situation where relief could be provided to one overloaded machine per period.

same period. A characteristic of those shops where this measure is important is the existence of several machines in a machine group so that no labor is wasted by the partial utilization of a machine group, or the possibility of assigning several tasks or machines to an employee.

3.3.2 Shop Work Balance Index

This is the variance of the utilization of the shop as a whole taken over time.

$$SWB = \frac{1}{p} \sum_{j=1}^{p} \left(\sum_{i=1}^{m} u_{ij} - \sum_{i=1}^{m} \bar{u}_i \right)^2$$

The formula for SWB can be simplified considerably by using the definition of $\bar{u}_i$ given in the notation and by defining the average work per period for the whole shop as follows:

$$\bar{u} = \sum_{i=1}^{m} \bar{u}_i$$

Then,

$$SWB = \frac{1}{p} \sum_{j=1}^{p} (m\, \bar{u}_j - \bar{u})^2$$

The objective is: Minimize SWB

This measure is important in those shops where there is flexibility in the type of work that each worker can do. In shops with this flexibility, a given job can be moved from one machine or operator to another without incurring a significant penalty. Therefore it is of primary importance that the total work to be done by the shop be fairly constant over time so that shop expansion and contraction be reduced, but it is not so important that the work available be distributed exactly according to the nominal machine or operator's capacity.

3.3.3 Period Work Balance Index

This balance index is based on the variance of the work done over all machines for each time period. An index can be calculated then by averaging the variance obtained in each time period. Let BP_j be the index for period j.

$$BP_j = \frac{1}{m}\sum_{i=1}^{m}\left(u_{ij} - \bar{u}_j\right)^2 = \frac{1}{m}\left(\sum_{i=1}^{m} u_{ij}\right)^2 - \bar{u}_j^{\,2}$$

$$PWB = \sum_{j=1}^{P} \frac{BP_j}{P}$$

Objective: Minimize PWB

This measure is similar to #1 in that it does not recognize any cancelling effects between machines, but in this case it accepts the fact that shop workloads will vary from period to period and looks only at the work balance achieved given the existing work load.

The use of this measure instead of #1 implies great flexibility in expanding or reducing the work force since no penalties are assigned for changes in the work performed in different time periods.

3.3.4 Machine Output Balance Index

This measure and the two that follow are very similar to measures #1-3. The only difference is that work output rather than work done or utilization is used.

The difference is better illustrated by an example. If machine i in period j finished job #1 and it spent two hours during the previous period and one hour in period j working on job #1, finished job #2 which took three hours, and spent three hours on job #3 but couldn't finish it, then the utilization of machine i in period j was seven hours while its work output was six hours. The example is represented by Figure 1.

In the case of the machine measures (3.3.4 and 3.3.6), this is a reasonable approach when the job movements are heavily dependent on time periods as would be the case when the machine groups are located in different buildings and trips with vehicles of fixed limited capacity can take place only once per scheduling period.

$$BO_i = \frac{1}{p} \sum_{j=1}^{p} (O_{ij} - \bar{u}_i)^2$$

$$MOB = \frac{1}{m} \sum_{i=1}^{m} BO_i$$

Objective: Minimize MOB

3.3.5 Shop Output Balance Index

$$SOB = \frac{1}{p} \sum_{j=1}^{p} \left(\sum_{i=1}^{m} O_{ij} - \sum_{i=1}^{m} \bar{u}_i \right)^2$$

$$SOB = \frac{1}{p} \sum_{j=1}^{p} (m\bar{O}_j - \bar{u})^2$$

Objective: Minimize SOB

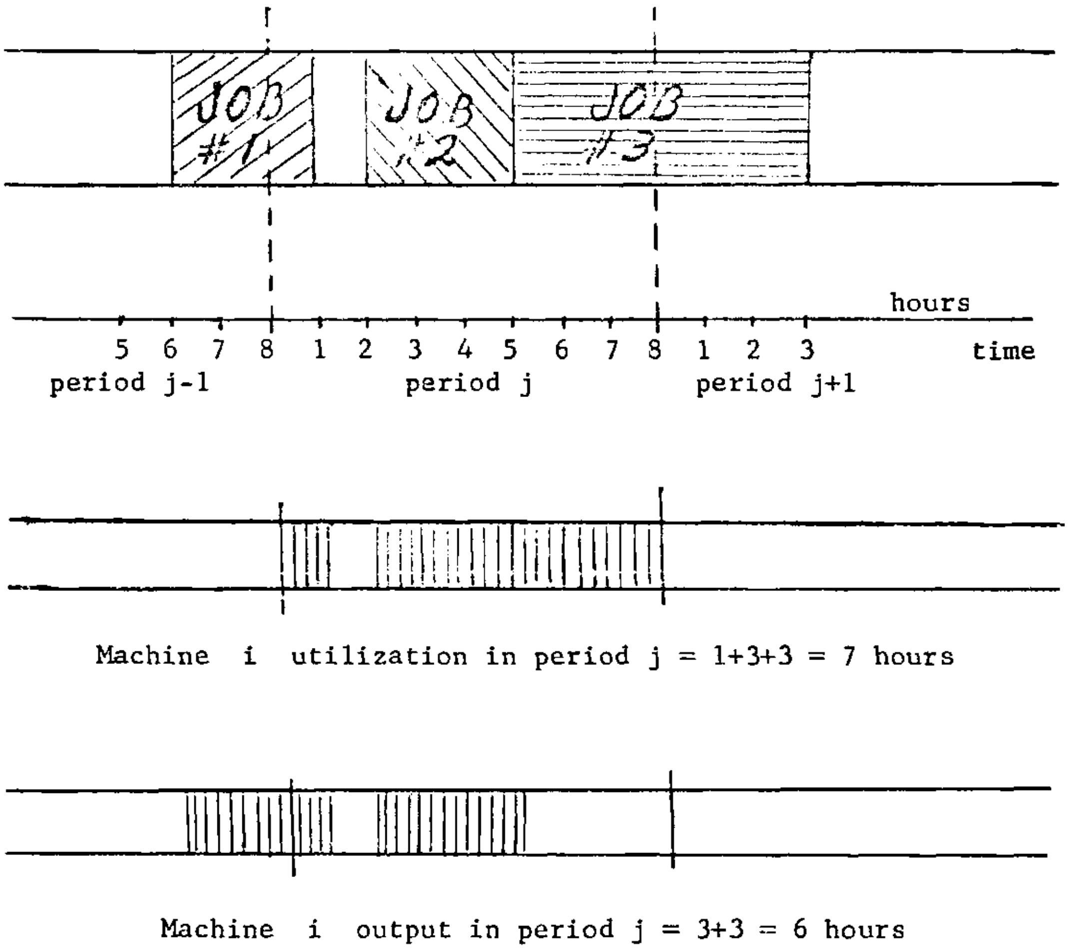

Figure 1. Illustration of Machine Output and Utilization in a Period

3.3.6 Period Output Balance Index

$$BPO_j = \frac{1}{m} \sum_{i=1}^{m} (O_{ij} - \bar{O}_j)^2$$

$$POB = \frac{1}{p} \sum_{j=1}^{p} BPO_j$$

Objective: Minimize POB

3.3.7 Machine Queue Balance Index

This is the variance of queue size in number of jobs for each machine over time. Then an overall index is obtained by averaging over all machines. Let Q_i be the machine queue index for machine i.

$$Q_i = \frac{1}{p} \sum_{j=1}^{p} (V_{ij} - \bar{V}_i)^2$$

$$QWB = \frac{1}{m} \sum_{i=1}^{m} Q_i$$

Objective: Minimize QWB

This measure is important when there is not much flexibility in the machine assignment for job operations and when furthermore it is desirable to keep the work in process as stable as possible during the scheduling horizon.

3.3.8 Shop Queue Balance Index

This is the variance of the number of jobs in the whole shop taken over time. This is a measure of work in process variability.

$$SQB = \frac{1}{p} \sum_{j=1}^{p} \left(\sum_{i=1}^{m} V_{ij} - \sum_{i=1}^{m} \bar{V}_i \right)^2$$

The equation for SQB can be simplified by using the definition of $\bar{V}_j$ given in the notation and by defining the average number of queues in the shop as follows:

$$\bar{V} = \sum_{i=1}^{m} \bar{V}_i$$

Then,

$$SQB = \frac{1}{p} \sum_{j=1}^{p} (m\bar{V}_j - \bar{V})^2$$

Objective: Minimize SQB

3.3.9 Period Queue Balance Index

This is the variance of queue size over all machines for each time period. An index is then calculated by averaging the variance obtained in each time period. Let PQ_j be the period queue index for period j.

$$PQ_i = \frac{1}{m} \sum_{i=1}^{m} (V_{ij} - \bar{V}_i)^2$$

$$PQWB = \frac{1}{p} \sum_{j=1}^{p} PQ_j$$

Objective: Minimize PQWB

This measure is similar to #7, but it recognizes that load variations for the entire shop over the scheduling horizon are unavoidable and attempts to reduce the influence of that kind of variation on the measure of performance.

3.3.10 Measures Related to Hours of Work in the Queue

Indices #7, 8, and 9 measure the variability of work in process using the number of jobs in the queue or the shop as the basis. It is obvious that similar measures can be obtained using the number of hours of work in the queue for the corresponding machines.

3.3.11 Machine Work in Process Balance Index

Measures #11-13 differ from measure group #10 in that before, the hours of work in process at a given queue to be worked by that machine were considered. This time the aggregate work in process for a machine, regardless of the queue where it presently resides, is of interest. Let $MWIP_i$ be the work in process index for machine i.

$$MWIP_i = \frac{1}{p} \sum_{j=1}^{p} (P_{ij} - \overline{P}_i)^2$$

$$AMWP = \frac{1}{m} \sum_{i=1}^{m} MWIP_i$$

Objective: Minimize AMWP

3.3.12 Shop Work in Process Balance Index

$$SWIP = \sum_{j=1}^{p} \frac{\left(\sum_{i=1}^{m} P_{ij} - \sum_{i=1}^{m} \overline{P}_i \right)^2}{p}$$

Let the average work in process in the shop be given as follows:

$$\bar{P} = \sum_{i=1}^{m} \bar{P}_i$$

Then SWIP can be simplified as shown below:

$$SWIP = \frac{1}{p} \sum_{j=1}^{p} (m\bar{P}_j - \bar{P})^2$$

Objective: Minimize SWIP

3.3.13 Period Work in Process Balance Index

Let $PWIP_j$ be the work in process index for period j

$$PWIP_j = \frac{1}{m} \sum_{i=1}^{m} (P_{ij} - \bar{P}_j)^2$$

$$APWP = \sum_{j=1}^{p} \frac{PWIP_j}{p}$$

Objective: Minimize APWP

3.3.14 Desired Loading Measure

This measure identifies the deviation of aggregate shop load for each machine (each period) from a specified target set up by management. The quantity obtained for each machine is then averaged over all periods. Let D_{ij} be the deviation of the aggregate load for machine i in period j from the desired amount.

$$D_{ij} = |P_{ij} - C_i|$$

(continued)

$$D_i = \frac{1}{p} \sum_{j=1}^{p} D_{ij}$$

$$D = \sum_{i=1}^{m} D_i$$

Objective: Minimize D

This measure is important because of its relationship to other shop oriented measures, its intuitive appeal to management and the fact that it attempts to look beyond the immediate conditions at one queue or machine (but without trying to predict or consider the interactions that occur in a job shop as one job moves from one machine to the next).

3.3.15 Maximum Queue Deviation Index

This measure gives for each machine the maximum excess in queue size over the desired queue size (or average) set up by management. The measure can also be used in terms of absolute maximum queue size without referring to any desired quantity.

$$QD_i = \max_j |q_{ij} - r_i|$$

$$QD = \max_i QD_i$$

or

$$QD = \sum_{i=1}^{m} QD_i$$

Objective: Minimize QD

A possible variation is to consider only positive deviations, that is,

$$QD_i = \max_j (q_{ij} - r_i)$$

This measure could be useful in those shops where a large penalty must be paid when queue lengths at machine i exceeding r_i cause a large penalty cost.

3.3.16 Period Idle Time

This measure obtains the maximum idle time for any machine in a period.

Then the average of all such maximum period idle times is calculated and the objective is to minimize this average.

Let PLEN be the period length.

IDT_{ij} is idle time for machine i, period j

$$IDT_{ij} = PLEN - u_{ij}$$

$$IDT_j = \max_i IDT_{ij}$$

$$PIT = \frac{1}{p} \sum_{j=1}^{p} IDT_j$$

$$= \frac{1}{p} \sum_{j=1}^{p} \max_j (PLEN - u_{ij})$$

Objective: Minimize PIT

3.3.17 Machine Idle Time

This measure looks at the idle time for a machine over all periods. Then either the maximum, average, or variance of this idle time is calculated and the index is obtained by averaging over all machines.

$$IDT_{ij} = PLEN - u_{ij}$$

$$IDT_i = \max_j IDT_{ij}$$

or

$$IDT_i = \frac{1}{p} \sum_{j=1}^{p} IDT_{ij}$$

or

$$IDT_i = \frac{1}{p} \sum_{j=1}^{p} \left(IDT_i - \frac{1}{p} \sum_{j=1}^{p} IDT_{ij}\right)^2$$

$$MIT = \frac{1}{m} \sum_{i=1}^{m} IDT_i$$

Objective: Minimize MIT

The balance index based on the maximum idle time for a machine over all periods can be used when there is a machine that could be "spoiled" or require adjustments if it is left unused over a certain length of time. The index based on the average idle time can be employed when it is necessary to measure the average machine utilization and finally the index based on the variance of the machine idle time could be useful in those cases in which a machine should be used at a steady rate from period to period.

3.3.18 Machine Arrival Balance Index

The variance in the work arrived to each machine over all time periods is calculated. Then an index is obtained by averaging over all machines. Let A_i be the machine index for machine i

$$A_i = \frac{1}{p} \sum_{j=1}^{p} (a_{ij} - \bar{a}_i)^2$$

$$MAB = \frac{1}{m} \sum_{i=1}^{m} A_i$$

Objective: Minimize MAB

This index is a function of the work arrival pattern, the loading mechanism, and the dispatching mechanism.

3.3.19 Shop Arrival Balance Index

This measure is similar to the machine arrival balance index, but the shop as a whole is considered.

$$SAB = \frac{1}{p} \sum_{j=1}^{p} \left(\sum_{i=1}^{m} a_{ij} - \sum_{i=1}^{m} \bar{a}_i \right)^2$$

$$= \frac{1}{p} \sum_{j=1}^{p} (m\bar{\bar{a}}_j - \bar{a})^2$$

3.3.20 Period Arrival Balance Index

Obtain the maximum deviation over all machines between actual arrivals to a machine in a period and the average arrivals to that machine.

The index then consists of the average over all periods

$$DA_{ij} = a_{ij} - \overline{a}_i$$

$$DA_j = \max_i DA_{ij} = \max_i (a_{ij} - \overline{a}_i)$$

$$PAB = \frac{1}{p} \sum_{j=1}^{p} DA_j$$

Objective: Minimize PAB

CHAPTER IV

METHODOLOGY FOR LOADING AND BALANCING THE SHOP

4.1 Shop Balancing, Job Pool Concept, Discrete Job Selection and Loading from the Pool

In general, industrial job shops are designed to operate optimally at certain load levels and outputs for each machine type. Load fluctuations from period to period for a given machine or for the shop, or even deviations from machine to machine within a period, result in costs of idle machinery and labor or overtime premiums.

The need for balancing workloads is beginning to be recognized by some shops and the author is personally aware of two apparel style shops where keeping the shop workload balanced is a primary objective.

The fact remains, however, that most job shop managers do not explicitly mention shop balancing as their primary goal. This has been reported by Panwalkar, Dudek, and Smith (1972). This result is not surprising because maintaining a balanced workload in a shop does not have an obvious payoff or direct penalty, as for example meeting due dates does. Besides, most shops tend to maintain an excessive amount of work in process therefore hiding the effects of poor balancing. Another way in which job shop managers hide the effect of poor balance conditions is by having some job operations performed at other than their normal machines. This, of course, results in increased costs due to poor machine use and/or expensive operator transfers.

Job shops need a certain level of work in process to operate at a given shop utilization percentage. If the work in process is evenly spread over all machines, then a smaller amount of work in process in the shop is needed to maintain the required shop utilization than if the work in process is concentrated in a few machines and not enough work exists for other machines.

The relationship between shop operating costs, work in process levels, and balance conditions in the shop floor has been discussed in the preceding paragraphs, but a direction of causality has not been definitely established. It will be shown during this research, however, that when jobs are loaded in the shop using an algorithm with an objective function which is primarily oriented towards improving one balance measure, then the other balance measures and work in process measures calculated are improved. The variance of the lateness distribution is also improved in some cases. It is therefore anticipated that when shop balance is improved, work in process levels are significantly reduced.

In traditional job shop studies, all jobs are scheduled and sent to their first operation machine as soon as they arrive in the shop. This causes long queues and high work in process as well as shop imbalance according to most balance measures.

Most of the existing dispatching rules are local rules in which only the queue information is used. Expected Work in Next Queue (EWIQ) is the only one of the more common rules mentioned by Conway and others (1967) which is not a local rule or a job dispatching rule since information that is not job related is used in determining job priorities. Deane (1972) considered the use of a shop dispatching methodology to improve machine workload balancing, but in his study all jobs were released to the shop as soon as they became available.

Analyzing the methods followed in actual shops in several industries, it can be seen that the shop is not loaded with every job that becomes available, that is, not all jobs are released immediately after it becomes theoretically possible to do so. Rather they are retained in a "suspense file"; this being nothing more than a notation in a scheduling book or at most an open purchase order or some unused raw material.

It is wise to keep backlogs off the factory floor. This reduces the work in process and allows a faster flow of jobs through the actual shop, even though the total flow time from the moment a job becomes available might and probably will be increased. Obviously, over a long period of time the total work arriving at the shop cannot be over 100% of shop capacity. In fact, as 100% utilization is approached, the queue sizes begin to move towards infinity. However, over short periods of time the work content of incoming jobs may exceed shop capacity. In these cases a temporary overload will exist in the shop. This overload will consist not only of jobs that have not been started yet, but also of an excessive quantity of partially completed jobs.

A useful tool to remedy this situation is to let the shop work behind a pool of jobs not yet released to the shop floor. Additional benefits can be obtained from the job pool if the job due dates are not critical so that there is increased flexibility in job selection.

Under the job pool concept the shop consists of a pool of jobs not yet released to the floor and distinct machine centers with a queue of jobs in front of each.

Loading consists of the release into the shop of a subset of the pool every scheduling period. The scheduling period can be a shift, a day, a week, etc. If the scheduling period is a day, then new jobs would come into the pool at various times during the day, but a subset of the jobs will be released from the pool to the shop once every day.

The key to the successful use of the job pool is the availability of a good mechanism to select those jobs from the pool that should be moved to the factory floor. This mechanism is in fact the proposed loading algorithm.

The use of the job pool and the loading algorithm provide another useful by-product. This is the concept of "discrete" decision making which is used in practice in many job shops. By this it is meant that decisions in many shops are not made in a continuous fashion, but rather they are made periodically by shop supervisors.

4.2 The Loading Algorithm

The loading algorithm, together with the job pool and the discrete decision making, is an integral part of the proposed loading and balancing methodology. As such, the objectives of the loading algorithm are the same as those of the complete methodology, that is, the improvement of shop balance and work in process measures while still operating under due date constraints. The specific objective function, however, employed by the loading algorithm is the minimization of the deviation from aggregate balance for each machine center in the shop.

Deviation from aggregate balance is interpreted as the difference between a desired total or aggregate load ahead of a machine and the actual load for each machine. The desired load is set by management and provides control over the shop operation.

It is evident that different objective functions are possible, but the one used concentrates on aggregate scheduling and releasing, as opposed to detailed dispatching, which is in line with the objectives of the research.

The loading algorithm utilizes a mixed integer programming approach with equality constraints based on the current workload assignments at each machine center. The constraints become equalities by the use of positive and negative slack variables giving the excess or lack of work (when compared to desired load) at each machine center. The objective of the program consists then of minimizing the sum of these slack variables, that is, minimizing the absolute deviations from the desired aggregate load for each machine center. An additional term is introduced in the objective function to make jobs in the pool increasingly attractive to be loaded in the shop as their due date approaches. The weight assigned to this term can be easily controlled by the production planner.

The decision variable in the algorithm (X_i) is a "0,1" variable. There is one such variable for each job in the pool. A value of "1" for the variable X_i implies that job i will be loaded in the shop.

The notation used is explained below:

$i = 1,2,\ldots,n$	n is the number of jobs in the pool
$j = 1,2,\ldots,m$	m is the number of machine groups
$X_i = 0$	decision variable, job not loaded
$X_i = 1$	decision variable, job loaded

W_{ij} — amount of work (standard hours) contributed by job i to machine center j

P_j — present load in the shop ahead of machine j. If $P_j \geq C_j$, then $P_j = C_j$ should be used.

C_j — desired aggregate load for machine j

S_{jL} — amount of work by which the set of jobs loaded plus any existing work in the shop that needs to be done by machine j falls short of the desired load for that machine center j

S_{jH} — amount of work by which the set of jobs loaded plus existing work exceed the desired load for machine center j

a_{jL}, a_{jH} — weights used to indicate the seriousness of out of balance conditions in one machine center relative to others. Also used to indicate the different effect of having a machine center underloaded as opposed to having it overloaded.

$f(di)$ — a function which increases as the due date d_i of job "i" gets closer. This function is a constant, for jobs having the same due date, in any scheduling period. The function used was:

$$f(di) = \frac{K}{[.1 + (di-d)]} ,$$

where (di-d) is the number of periods away from the due date.

B_j — limit desired, if any, on the work loaded for machine j

$F_c \pm L$ — upper and lower limits desired on total amount loaded

The mathematical formulation of the algorithm is given as:

Minimize

$$D = \sum_{j=1}^{m} a_{jL} S_{jL} + \sum_{j=1}^{m} a_{jH} S_{jH} - \sum_{i=1}^{n} f(d_i)X_i$$

subject to:

$$X_i = 0,1$$

$$S_{jL} \geq 0$$

$$S_{jH} \geq 0$$

$$\sum_{i=1}^{n} W_{ij} X_i + P_j + S_{jL} - S_{jH} = C_j \qquad j = 1,2,\ldots,n$$

The first term in the objective function is a measure of the sum of the underload conditions in hours for each machine type. The second term represents the hours of work in excess of the amount desired for those work centers that are overloaded. The work loads being mentioned here are aggregate workloads in the shop and not loads at the individual machine center queues. The third term is the due date adjustment term as given above.

The first term in the constraint function is the work loaded in the shop for one machine by the jobs selected for release by the algorithm. The second term, P_j, is the existing aggregate shop load for machine j prior to the release of the new jobs. The total load for machine j given by these two terms falls short or exceeds the desired amount C_j by the value of the slack variable S_{jL} or S_{jH}, respectively.

Additional constraints can be imposed to allow only a range on the total work hours loaded in the shop in one scheduling period. That is, they will require that the total work hours moved to the shop every scheduling period will be over (F_c-L) and below (F_c+U). These constraints are:

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{m} W_{ij} \right) X_i \geq F_c - L$$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{m} W_{ij} \right) X_i \leq F_c + U$$

Finally, the amount of work loaded in one period for one work center or group of work centers can be restricted. This type of constraint is useful, for example, in a shop where there is a preliminary operation, such as cutting in an apparel shop, through which jobs have to pass before arriving at the true job shop. The equation that follows indicates that the aggregate work loaded in one period for machine center j is not to exceed B_j.

$$\sum_{i=1}^{n} W_{ij} X_i \leq B_j \qquad \text{for any } j \text{ or group of } j\text{'s}$$

This algorithm offers a degree of control on the operation of the shop by the use of different values for the constants K and C_j. The specific effect of changes in the C_j will be discussed in Chapter VII.

4.3 The Linear Approximation to the Mixed Integer Programming Loading Algorithm

The above mathematical formulation makes use of a mixed integer program. The simulation program for the job shop uses a linear approximation with bounded variables. The decision to do this was based on the following considerations.

a) Based on tests made with the mathematical programming package OPHELIE from Control Data Corporation, the time required to obtain an integer solution was from 5 to 50 times the linear solution time requirements. This excessive time requirement, even with a fast commercial code, eliminated the possibility of using the mixed integer model in the simulation where the loading algorithm had to be used in 500 periods for each run type and replication.

b) Bounded variable theory shows that the number of non-integer variables in the basis of the LP when a solution is obtained is limited. In a bounded variable problem with m equality constraints and n structural bounded variables the number of structural variables that is between zero and the upper bound is equal to the number of constraints. This is demonstrated by Chung (1963) using the following argument:

> If there are r structural variables reaching their respective upper bounds and s variables equal to zero, then the number of structural variables which are positive but below their upper bounds must be n-(r+s). In order to satisfy the upper-bound constraints, this means that there must be [s+n-(r+s)] slack variables in the basis. By assumption we know that there are [r+n-(r+s)] structural variables in the basis; therefore the total is [r+n-(r+s)] + [s+n-(r+s)] = 2n - (r+s). But in a bounded variable problem of this type, the number of basis vectors is m+n. Therefore, we have 2n-(r+s) = m+n, which yields

$$n-(r+s) = m$$

indicating that the number of structural variables whose values are between 0 and the upper bound is m.

The above argument needs only slight modification to fit the basic equations in the algorithm described in section 4.2. In this case the number of equations is also m, but the number of structural variables with an upper bound is k and there are 2m structural variables without an upper bound. The total number of structural variables is again n where n = k+2m. It is easy to see from the structure of the problem that at least half of the non bounded structural variables (designated as S_{jL} and S_{jH} in the algorithm formulation) will be equal to 0 because for any machine center there will be an overload or underload condition but not both.

Using the same argument employed by Chung, the number of structural bounded variables, X_i, and their slacks in the basis is 2k-(r+s). The total number of variables in the basis is m+k. Now let t be the number of non bounded structural variables in the basis.

Then

$$2k-(r+s)+t = m+k$$

$$k-(r+s) = m-t$$

but $$0 \leq t \leq m$$

therefore $$0 \leq k-(r+s) \leq m$$

This indicates that the number of structural bounded variables (decision variables for jobs to be loaded in the shop) whose values are larger than zero but less than one is less than the number of equations, that is, less than the number of machines in the shop.

In actual practical problems the number of fractional variables is considerably smaller than the theoretical limit. For example, in an actual problem with 10 machines and 29 jobs in the job pool, the number of jobs with a fractional X_i value was four while, of course, the theoretical limit was 10. Four problems with 10 machines and between 20 and 30 jobs in the job pool were observed and the number of non-integer job decision variables in the output was between three and five in each case.

c) The results obtained with the job pool and the linear version of the loading algorithm were significantly better than those obtained in an "uncontrolled" shop. The use of the mixed integer version could only improve the results further. This potential improvement, however, is fairly limited because less than a .1% difference has been observed between the values of the objective functions when fractional decision variables are allowed and when the non-integer decision variables are rounded to 0 or 1.

d) There is no guarantee that the optimum will consist of a conversion of the non-integer variables in the solution to 0-1 variables, in fact, most of the time this will not be so. However, the tests performed with the OPHELIE LP system indicated that although a few of the variables changed, the objective value of the rounded solution was only slightly worse than the one given by the mixed integer solution.

e) When the model is used to load an actual shop, the algorithm will be used only once or twice per day while on this research due to the time simulated, the number of replications and the different conditions tested, it was employed over 20,000 times. It can be seen that the additional computer time required to use the mixed integer version in an actual situation will be fairly insignificant and therefore in that case a detailed study should be made regarding the trade off involved in using the linear approximation.

4.4 Loading Heuristics

The concepts of using a job pool and loading the shop at discrete intervals with jobs in the pool have also been employed with a heuristic loading method as well as the mathematical programming loading algorithm. This does not imply that the loading algorithm is optimal in a general sense, although, of course, it is optimal with respect to its objective function.

The heuristic consisted of loading a job in the shop if the first job operation made a contribution to the queue of a machine that was underloaded. For a given machine, the jobs were selected one at a time with those having the earliest due date selected first until the desired load level for that machine queue was reached or until the job list was exhausted. This was done for every machine.

In addition, an optional feature provided for the loading of additional jobs in the shop if the management desired load for the shop in total had not been reached with jobs loaded in the first part of the algorithm. Again jobs with the earlier due dates were selected first.

In effect, there are two main "factors" in the loading and balancing methodology that may influence the performance criteria. One of these factors is the concept of the job pool itself while the other factor is the releasing (or loading) methodology employed. The results obtained with the heuristic releasing method have been employed to try to isolate the effects of the two "factors."

CHAPTER V

DESCRIPTION OF THE TESTING VEHICLE

5.1 The Job Shop

The general type of shop with which this research is concerned was presented in Chapter II and its specific characteristics are next described.

No special effort was made to model a specific shop, however, most of the parameters employed are within common ranges for job shops in the apparel "style" and other industries. An exception to this was the selection of ten machines, but this choice was previously explained as a compromise due to computer time requirements. The practice of employing reasonable parameter values, but not values from a specific shop, has been commonly employed in job shop research.

The interarrival times are samples of an exponential distribution with a mean of 1.88 hours and truncated at 40 hours (the true mean is therefore slightly less than 1.88 hours). This arrival rate, together with the other shop parameters used, resulted in a shop utilization between 81% and 83.5%. The shop utilization was determined from statistics accumulated in the simulation runs.

The jobs arriving to the shop were assigned an equal probability of having their first operation performed by any of the machines in the shop. The machine for subsequent operations was then obtained by employing a transition probability matrix. The transition probability matrix used in most of this research was such that a job was assigned an equal probability of moving to any machine in the shop for their next operation regardless of the machine in which the current operation was performed. The experimental model was thus characteristic of a pure job shop. Some experimental investigations were also performed utilizing a shop with "flow structure".

The processing time per operation was generated using an exponential distribution with a mean of 2.48 hours, but with no operations lasting less than one hour or over nine hours.

The number of operations of an incoming job was generated when the job arrived at the shop using a symmetric unimodal distribution which is shown in Table 2.

A job due date was assigned to each job as it entered the shop using one of two methods. The first method which was used almost exclusively consisted of assigning a due date equal to the current time plus the work content of the job plus a sample from the uniform distribution between 0 and 150. The second method was slightly more complex and it consisted of assigning to 10% of the jobs a due date such that the job had three times its work content in hours to go through the shop. The remaining 90% of the jobs had their work content in hours plus the product of 300 hours times a random number between .1 and 1 to complete all operations. This method is illustrated in Figure 2. Its purpose is to eliminate the existence of jobs with very tight due dates. Only one of the two methods described was used in any given run.

5.2 Dispatching Rules

Four dispatching rules were studied in detail. These rules were

Table 2. Probability Distribution for the Number of Operations of an Incoming Job

Number of Operations	Probability of Occurrence
	General Shop
4	0.15
5	0.20
6	0.30
7	0.20
8	0.15
	Shop with Few Interactions
1	0.20
2	0.60
3	0.20

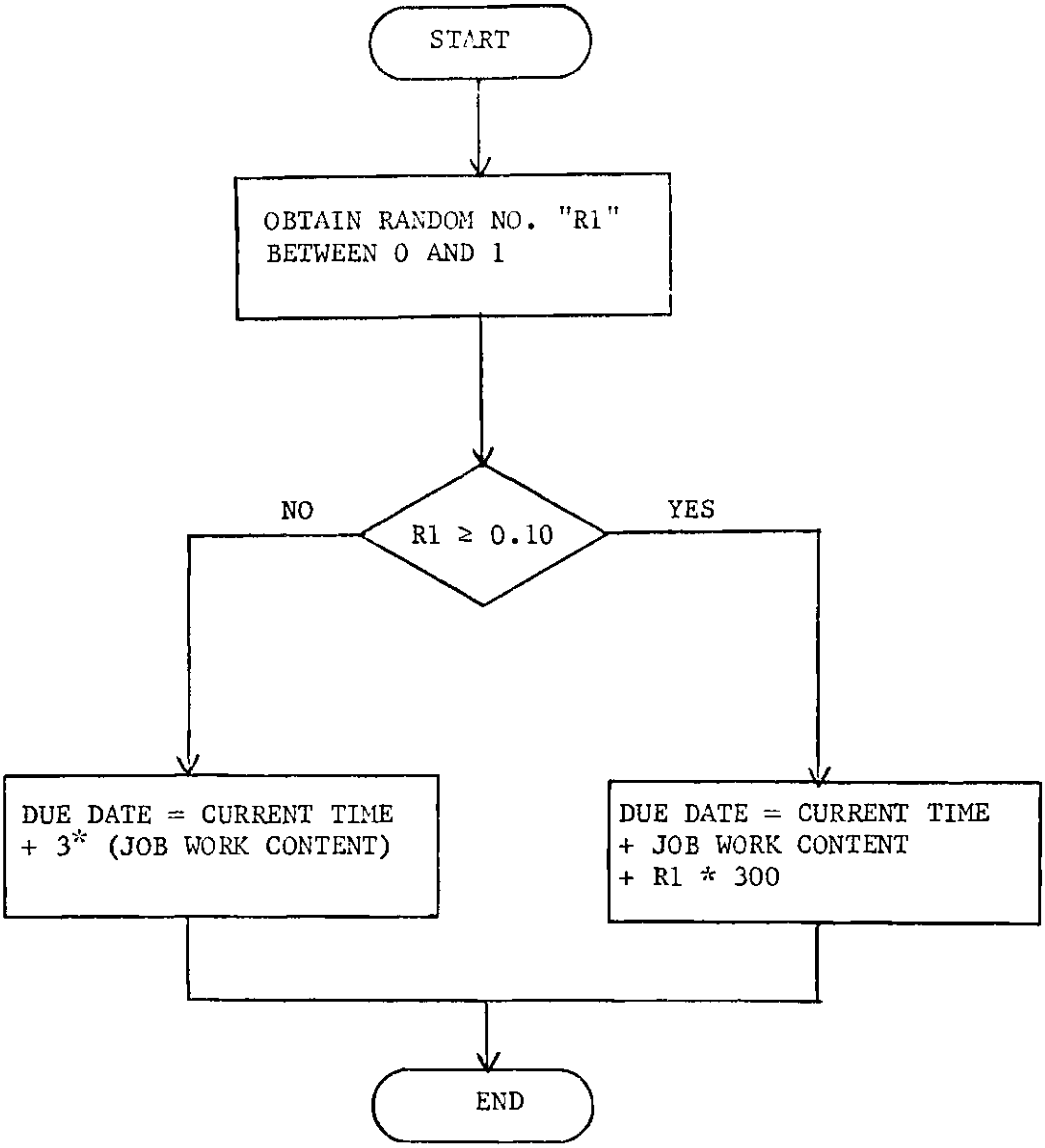

Figure 2. Method No. 2 of Assigning a Due Date to a Job

used with both an uncontrolled shop model and the controlled loading model with the job pool. The dispatching rules were:

a) DSOP, Dynamic Slack per Operation.

The job priority equals the ratio of the slack remaining to the number of operations remaining. The jobs are selected in all cases such that those with the smallest algebraic priority measure value are selected first from the queue.

DSOP was selected because prior studies (Le Grande (1963), Gere (1966), and Conway and others (1967)) have shown that this rule performs well with respect to minimizing the variance of the lateness distribution.

b) EWIQ, Expected Work in Next Queue.

The job priority equals the sum of the imminent operation processing times of other jobs in the queue to which the candidate will enter after its current operation. The queue load being added is considered to include jobs now on other machines which will arrive before the job being considered, if it is selected for immediate operation.

Thie rule was selected because of its "look ahead feature" that is, it is not a local dispatching rule.

c) SPT, Shortest Processing Time.

The job priority equals the processing time of the imminent operation. This rule had to be selected. Practically all of the simulation studies that have been mentioned show that it is a very good rule with respect to many measures of performance and at least acceptable with respect to the remaining measures. See, for example, Conway and others (1967).

d) FIFO, First In, First Out.

The job priority equals the time the job enters the particular queue. This rule was selected due to its implicit fairness and also due to the fact that it is used quite often in practice.

5.3 The Simulation Model

In order to test the effects of the loading and balancing methodology a computer simulation approach was employed. As previously discussed a simulation approach had to be selected for this purpose due to the lack of theoretical queuing results in job shop scheduling research.

The job shop simulator program in this research was written using the GASP II language described by Pritsker and Kiviat (1969). GASP II is a collection of Fortran IV subroutines organized to assist in performing simulation studies. GASP II provides subprograms for handling those simulation tasks that are independent of particular problems. The tasks handled by GASP II are the maintenance of the simulation clock, the handling of independent files and the ranking of elements in those files, the placing and removal of elements from the files, the random variable generation and the maintenance of simulation statistics as well as the production of appropriate summaries.

The user subroutines complement the GASP II program and must be tailored to the specific application. A description of these subroutines as well as flow charts are given in Appendix A.

Figure 3 depicts the operation of the job shop simulator. The Main program reads the user subroutine parameters and starts the simulation by transferring control to the GASP programs. The simulation proceeds

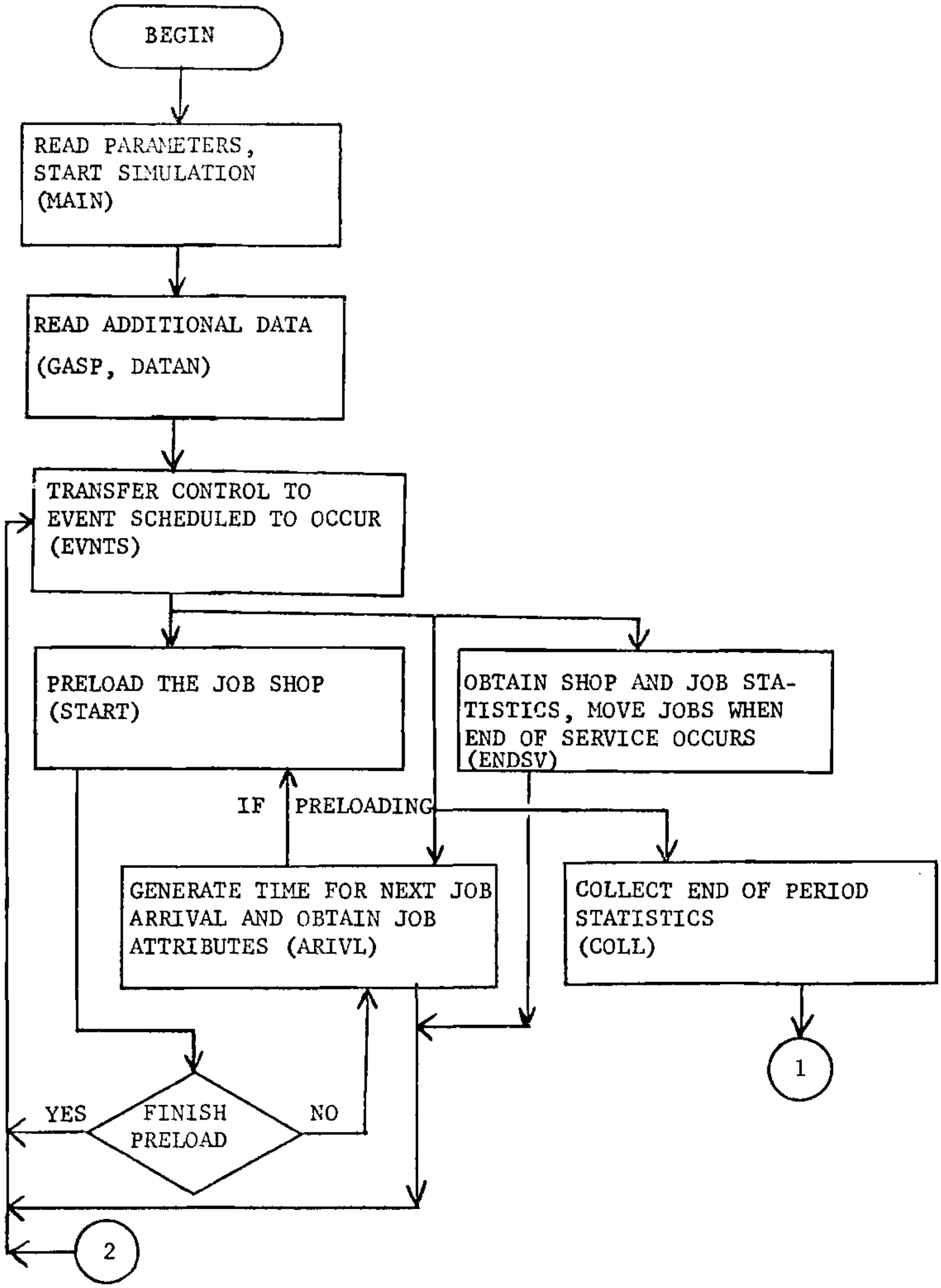

Figure 3. Job Shop Simulator

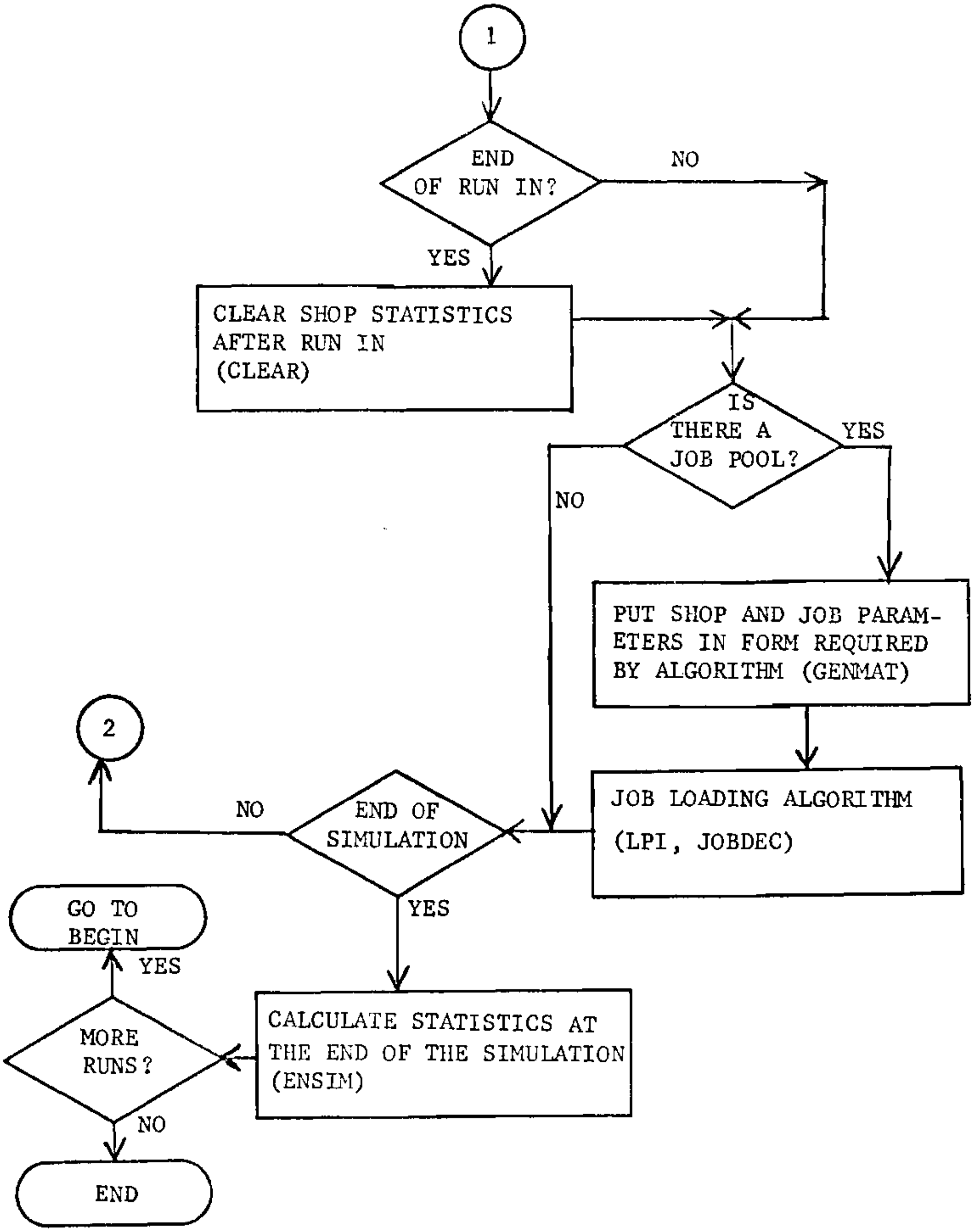

Figure 3. Concluded

by causing discrete events to occur and calling the right event at the right time is the function of EVNTS. The four events in this simulation are: preloading the shop at time zero (START), obtaining job attributes for new arrivals (ARIVL), moving jobs when a machine operation is finished (ENDSV) and collecting end of period statistics (COLL). The subroutines related to the job loading algorithm are GENMAT, LPI and JOBDEC. Finally, there is a subroutine whose function is to calculate all the statistics at the end of the simulation and print results.

The Fortran IV listing of the user programs are provided in Appendix B. Appendix D contains the description of all attributes, events and optional variables in the GASP programs. The non-GASP variables are described in Appendix E. A variation of the subroutines ENDSV, PTJOB and GENMAT needed for the alternative machine operation feature is given in Appendix K. A sample input set for the simulation program is shown in Appendix F.

The input data provides for reading some decision parameters that are used to change significantly the character of a simulation run. The most important ones are the following:

NRULE Indicates the dispatching rule to be employed in the simulation run.

NLDR Specifies whether a job pool is to be used or not. If a job pool is to be used, it determines which loading method should be employed.

IDUE Specifies which one of two methods of generating due dates is to be used.

NARR Indicates whether the arrival process is strictly Poisson, or whether the interarrival times calculated according to the exponential distribution are superimposed on a sine curve. This causes the mean interarrival time to fluctuate with respect to time.

MSW Specifies which special loading modification, if any, will be used.

DESLF Specifies the desired shop load level or management load factor when the controlled shop loading approach is utilized.

Using different values for these parameters and others, several special shop conditions were investigated.

CHAPTER VI

DESIGN OF THE EXPERIMENT AND VALIDATION

The approaches toward validation and experimental design of the simulation experiments that have been followed and which are mentioned below are based on the books by Naylor, Balintfy, Burdick and Chu (1966), Schmidt and Taylor (1970), and Tocher (1963); the dissertation by Deane (1972) and the papers by Naylor, Burdick and Sasser (1969), Van Horn (1971) and Conway (1963).

The elements in planning a simulation experiment according to Naylor, Balintfy, Burdick and Chu are the following:

1. Formulations of the problem
2. Collection and processing of real world data
3. Formulations of the mathematical model
4. Estimation of parameters of operating characteristics from real world data
5. Evaluation of the model and parameter estimates
6. Formulations of a computer program
7. Validation
8. Design of the Simulation Experiment
9. Analysis of Simulation Data

Items 1-6 have already been discussed and item 9 will be covered in a subsequent chapter. The purpose of this chapter is to discuss items 7 and 8. Items 7 as listed here includes the steps of verification and validation given by Fishman and Kiviat (1967). Verification insures that a simulation model behaves as an experimenter intends. Validation tests the agreement between the behavior of the simulation model and a real system.

6.1 The Experiment

The design of simulation experiments must include a random number generator which is truly random, and considerations of start up conditions, run lengths, replications and finally, have the results pass adequate tests of statistical significance. These items will be examined next.

6.1.1 Random Number Generator

The pseudo random number generator used employs a 17 bit multiplicative congruential method. The general formula used is:

$$N_{i+1} = AN_i \pmod{m}$$

where $A = 5^7$ and $m = 2^{17}$.

The maximum attainable period with this generator is 32,768 and the quantity of random numbers used by a run in this research is close to 30,000.

The random number was tested with a group of seeds some of which were used for the experimental runs. The tests used and the purpose were the following:

a. Goodness of fit, Chi Square test.

The numbers generated were grouped in intervals of .1 from 0 to 1 and a χ^2 test was used to check fitness to a uniform distribution.

b. Goodness of fit, Kolmogorow - Smirnov test.

Same purpose as the first test.

c. Serial test, Chi Square.

The purpose of this test was to detect any first order serial correlation. The numbers were truncated so that only the first digit was used and every number was placed in one cell of a ten by ten array as given by the first digit with the columns indicating the previous number obtained and the rows giving the current number. A Chi Square test was then used to test the uniform distribution of the random numbers over the 100 cells.

d. Total Runs, Normal Statistic.

The expected total number of runs was calculated. For samples greater than 20, the distribution of the total number of runs can be approximated by the normal distribution. This fact was used in constructing a two-tailed normal test for checking the number of runs generated.

e. Number of Runs for each Run Length, Chi Square.

A Chi Square test was used to compare the observed vs expected number of runs of run length 1, 2, 3, 4 and greater than 4.

The results of the tests and critical values at the $\alpha = .05$ level for the generator with twelve seeds that passed the test and for a sequence of 10,000 numbers are given on Table 3.

6.1.2 Starting Conditions

Starting conditions are one part of the more general question of equilibrium. According to Tocher (1963), the accepted technique has been to invent starting conditions and to allow the simulation to proceed for

Table 3. Results of Tests on Random Number Generator

		Goodness of Fit χ^2	Goodness of Fit KOLM-Smirnov	Serial Correlation χ^2	Total Number of Runs -- Normal	Number of Runs of Each Length -- χ^2
Critical Values		16.92	.0136	123.2	1.96	9.49
Test Results						
Seeds						
1.	100933	7.59	.0040	64.7	.34	5.36
2.	411719	8.12	.0050	64.7	.02	1.32
3.	297449	9.11	.0080	83.9	.02	3.86
4.	349387	6.85	.0050	61.9	.07	3.07
5.	281923	7.22	.0040	64.9	.43	1.18
6.	154231	4.41	.0060	64.3	1.11	5.30
7.	329963	7.33	.0060	80.2	.85	1.61
8.	900131	7.61	.0080	81.8	.24	4.43
9.	392819	2.55	.0060	61.1	.19	.48
10.	214753	2.47	.0050	61.2	.17	.35
11.	200933	6.15	.0050	54.4	.01	2.52
12.	117341	5.87	.0050	55.8	.00	2.86

some time and take the final conditions as the initial conditions of the genuine run. According to Conway (1963), the length of time required to render the state probability distribution independent of the starting conditions must certainly depend upon the starting conditions used. The approach selected in this research was to preload the shop with a number of jobs that would give approximately the same number of hours of work in process in the shop as the hours of work in process that were observed at the end of several trial runs.

Actually, the specific condition selected is not too important since all that must be done is to select a reasonable starting condition. "Reasonableness" according to Conway (1963) should simply be associated with conditions that possess non-zero probability in the equilibrium state probability distribution.

6.1.3 Run in Period

The run in period in a simulation study is the time during which the simulation is allowed to proceed so that operating conditions hopefully reach a "steady" or "representative" state, but not allowing shop statistics accumulated during this time to influence final results. There is not any general method that can be used to determine the length of run-in period. Tocher (1963) flatly states this and Conway (1963) says "there is no single point in the execution of a simulation experiment beyond which the system is in equilibrium."

Regardless of how "good" the initial conditions selected are, there is general acceptance of the idea that a run-in period is needed. Deane (1972) presents a very convincing argument for this. In effect, he argues that if no run-in period is considered, the first jobs leaving the system will have biased statistics for time spent in the system and due date measures. Also, the initial statistics for work in process performed in the shop will also be biased.

The run-in period selected for this research was 400 hours during which about 175 jobs left the shop and around 1200 operations were performed.

The selection of 400 hours as the run-in period was made after detailed printouts were obtained showing conditions at the end of every 8-hour period. After examining these results, it was clear that there were no easily spottable abnormal conditions in the statistics collected after 30 or 40 periods. However, 50 periods were used in order to stay in the safe side. For example, Figure 4 shows the work in process in operations done for jobs in the shop plotted against time periods for periods 1 through 50 in one of the trial runs made. It can be seen that after the first 25 or 30 periods the initial almost uninterrupted increase in the value of the variable has ceased and a more normal fluctuation is observed. The run-in period selected certainly satisfies the rule of thumb given by Tocher (1963) that the longest cycle in the simulation should have been executed three or four times before abnormal behavior caused by starting conditions can be expected to have died away. Although there are many cycles in a job shop, it was felt that the longest cycle of interest in this research is the time a job spends in the system. This time was from 70-100 hours in the experimental runs, depending on the conditions used.

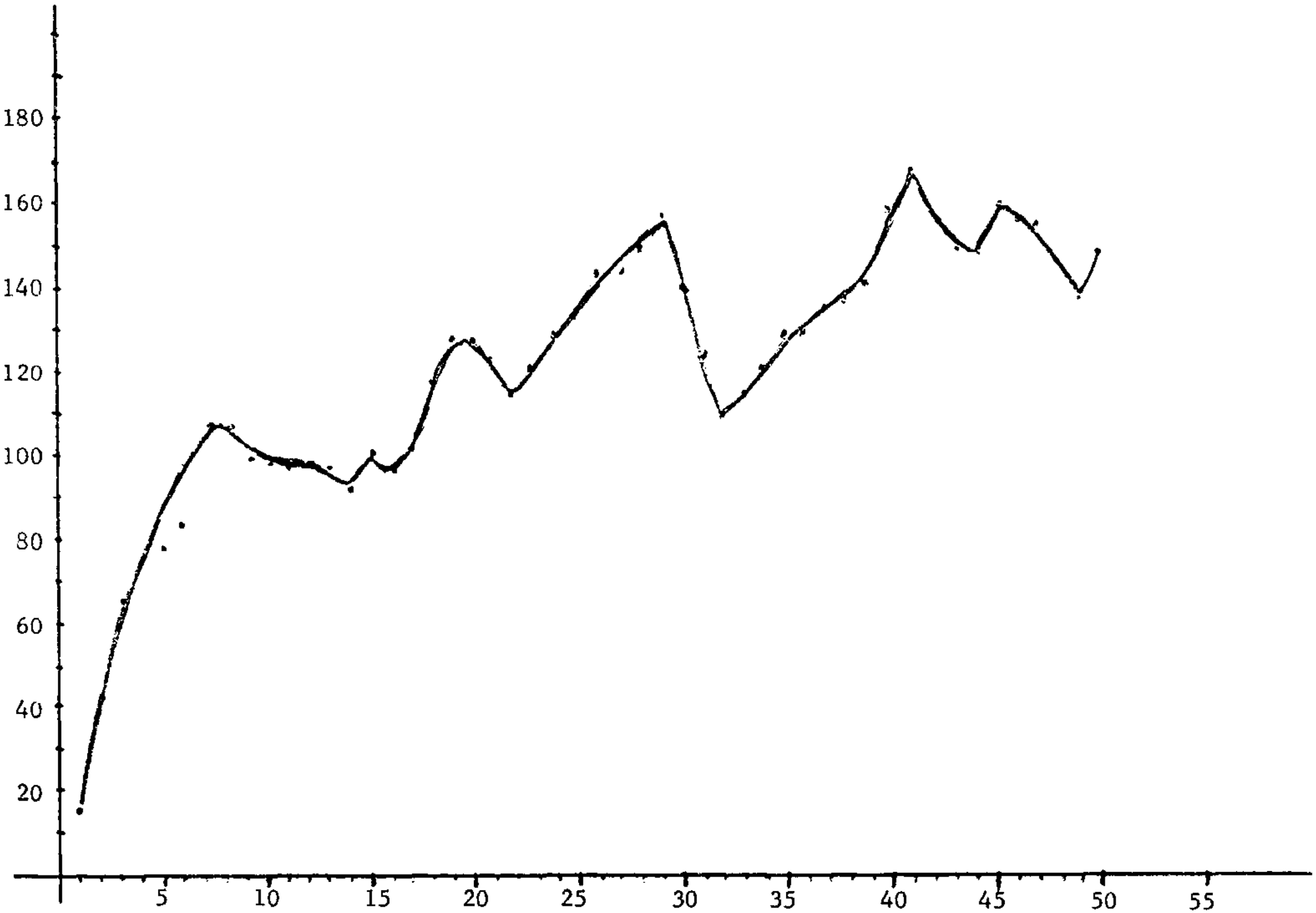

Figure 4. Work in Process in Operations Done for Jobs in the Shop During the Run In Period

Another rule of thumb that was considered is the following one proposed by Conway (1963): "From pilot runs, truncate a series of measurements until the first of the remaining series is neither the maximum nor the minimum of the remaining set." Conway cautions against examining cumulative statistics for this purpose because they may cause the discarding of too much data. However, if they are used, the error would be on the conservative side. If an average based on a cumulative statistic were used, the last measurement of the truncated series instead of the first of the remaining one must be used. The run-in period used in this research also meets this rule of thumb for the statistics that were printed in detail, whether they are presented on a cumulative basis or not.

6.1.4 Run Length

The variability associated with the measurements of even very simple simulation models is discouragingly large according to Tocher (1963). However, what is desired in most simulation experiments, including this one, is the comparison of alternatives so that relative results are more important than absolute ones.

Another property of simulation experiments that helps keep run lengths and replications to manageable levels is the use of identical event sequences. This procedure insures that any relative differences observed can be attributable to the alternatives and not to random variation.

A trade off in run length still exists since it would be desirable to have very large samples to reduce variability as much as possible, while at the same time run lengths must be kept at reasonable levels to economize computer time.

The time selected for each replication in this simulation experiment was 4000 hours (500, 8-hour periods). In this time about 2,120 jobs left the shop and about 12,720 operations were performed.

Initially, several replications were obtained by using 100 periods after run-in but the variance on the measures of performance was too high. Other run lengths (Tables 4 and 5) were tried until it was decided that 500 periods (4,000 hours) reduced the variance considerably and that additional use of computer time would not be justified. The statistical verification of this run length was made by taking six 4000 hours runs with different seeds and comparing these results with those of a second set with different seeds (Table 5). A standard t-test was then performed to test for equality of means as suggested by Deane (1972). The results on the Shop Balance Measure (SWB) and Average WIP (hours of work in the shop) for each set of runs are shown in Table 5.

The $t_{.975}$ (10) value obtained from the tables is 2.228 while the calculated values were $t_{SWB} = .38$ and $t_{WIP} = 1.39$. Therefore, there are no grounds to reject the hypothesis of equality of means in either case. It is granted that this is not a rigorous justification for the run length selected, but it provides additional assurance.

6.1.5 Replications

One of the first questions faced when deciding the number of runs to be made is whether successive runs shall consist of wholly independent runs started with new random number seeds or whether they should be started using the final calculation of one run as the beginning of the next one.

The advantage of using the first approach is that there is less

Table 4. Runs in an "Uncontrolled" Shop. DSOP Dispatching Rule. Results with Run Lengths of 800 and 3200 Hours

	100 Periods (800 Hours)		400 Periods (3200 Hours)	
	SWB	WIP (Hours)	SWB	WIP (Hours)
	.467	854	.721	670
	.867	616	.597	665
	1.443	505	1.074	642
	.973	559	.997	613
	1.290	565	.953	616
	.586	724	.675	617
Avg.	.938	637	.836	637
Var.	.1456	17013	.0388	918
Std. Dev.	.382	130.5	.197	30.3

Table 5. Two Sets of Runs in an "Uncontrolled" Shop. DSOP Dispatching Rule. Results Used to Test the Adequacy of 4000 Hours Run Length

	Set 1		Set 2	
	SWB	WIP (Hours)	SWB	WIP (Hours)
	.753	670	1.071	620
	.715	626	.966	598
	1.036	625	.992	621
	.954	619	.945	616
	.892	623	.683	625
	.646	635	.652	638
Avg.	0.833	633	0.868	620
Var.	.0229	352	.0287	170
Std. Dev.	.151	18.8	.170	13.1

$s^2_{SWB} = .0258$

$s^2_{WIP} = 261$

$t_{SWB} = .38$

$t_{WIP} = 1.39$

risk of running into autocorrelation problems. The second approach insures that satisfactory initial conditions are used in all replications after the first one and also eliminates the need for a run-in period in the second and successive replications.

The first approach has been used in this thesis.

The actual number of replications used is a function of the precision desired in the results and the computer time available. In a case like this one in which it is desired to obtain and compare the values of a group of statistics under different conditions, it is impractical to start from the precision required and arrive at the number of runs needed. Instead, the approach followed was to select the quantity of five replications as an acceptable number from both points of view.

6.1.6 Statistical Design of the Simulation Experiment

The selection of factor levels and combinations of levels and the order of experimentation is often a critical decision in simulation experiments. The number of runs, even with incomplete experimental designs, that might be needed to cover an acceptable range of the factors often gets out of hand. A factorial treatment arrangement was not employed because this arrangement was not necessary to answer the most important question being investigated.

The primary purpose of this study is to explore the effects of loading jobs into a shop from a pool, and to compare the values of some measures of performance using this approach against letting the jobs arrive to the shop directly. It was desired to do this for four different dispatching rules. A paired observation t-test was used to test for significance in the differences observed between the two loading methods. A total of 40 runs were required for the main portion of the experiment. There were two loading rules to test (an "uncontrolled" shop and shop with a job pool and the loading algorithm) and four dispatching rules to be used with each loading method. Each one of these eight conditions was replicated five times.

An ANOVA has been performed on the four runs with five replications each that do not use the job pool, that is, the conventional uncontrolled loading approach to test for any differences in the means of the measures of performance. Another ANOVA has been used in a similar way for the 20 runs (4×5) using the job pool. Statistical tests were also performed to determine the effect of the job pool and loading algorithm on the different dispatching rules.

Several additional items have been explored utilizing the runs mentioned before, but also requiring some additional runs. These runs were made under only one dispatching decision rule, DSOP.

Dynamic Slack per Operation (DSOP) was selected because this is a decision rule which has been shown to give good results with respect to due date measures without showing an extraordinary adverse effect on other measures. A t-test has been used to test for the significance of any differences observed, unless otherwise noted.

The additional shop conditions that have been tested are

a. Effect of a variable job arrival rate. The effect of a variable job arrival rate, that is, an arrival distribution with a dynamic mean which has been used throughout in this research is illustrated by comparing results previously obtained against the results of five additional replications using a fixed arrival rate. The fluctuating arrival rate was obtained by having the interarrival time generation process superimposed on a sine curve such that the mean interarrival time changed from 50% to 150% of its normal value with a period of 16 hours.

b. Effect of the job pool and the loading algorithm when used in a shop with less interactions. This is illustrated with a shop of five machines and an average of two operations/job. Ten additional runs were required here consisting of five replications for each loading condition.

c. Effect of using a heuristic to load the shop from the pool. The purpose of this test is to show the advantages of the loading algorithm over a reasonable heuristic which also utilizes the job pool concept.

d. Effects of variations in the loading algorithm. Several variations of the loading algorithm were explored for various management load factors. The variations consisted of changes in the job releasing mechanism. The results obtained, however, did not justify making the additional computer runs necessary for statistical analysis.

e. Effects of a non-symmetric transition matrix when the machine utilizations remain the same. This experiment requires additional replications (only with DSOP) under loading and no loading conditions.

It is desired to investigate the effect of a non-symmetric transition matrix under the uncontrolled loading approach. Also it is desired to check if the improvements produced by the controlled loading methodology are more significant when the imbalance condition exists. A t-test has been used.

The non-symmetric transition matrix is characteristic of a shop in which special work flow patterns can be identified, that is, when a pure job shop does not exist. The average utilization for each machine and the probability of initial job arrival at each machine was maintained equal between all machines, but the work flow structure used was such that some paths were much more likely than others.

Complete results of all simulation runs are presented in Appendix I. Results are analyzed and summarized in Chapter VII.

6.2 Program Validation

The conditions normally recommended (Naylor, Chu & others; 1966) to insure a satisfactory program validation are:

a. To verify how well the simulated values of the endogenous variables compare with known historical data.

b. To verify how accurate are the simulation models' predictions of the behavior of the real system in future time periods.

It is not possible to satisfy the above conditions in this research because there is no shop data available of the type required to make the comparisons.

Fortunately, however, there have been previous job shop simulation models reported in the literature, some of which have been verified. The verification and validation in this case will consist of comparing results in this research to results reported by Conway (1963) and Deane (1972).

The measures of performance of primary interest in this research are the Shop Balance Measure (SWB) and other measures of balance, the level of work in process calculated in two different ways and a measure of the ability of jobs to meet due dates.

Many of these measures are not available in published research that has been validated and, therefore, it is not possible to use the most interesting measures (balance measures) to validate the program in this thesis. Three measures of performance that were selected for validation are the average flow time, the level of in process inventory and the standard deviation of the lateness distribution. These measures were selected because of their relative interest to this thesis and their availability in published research. The comparisons are shown in Tables 6-8.

The results shown for this thesis are based on average values for the applicable runs (five replications) reported elsewhere on this thesis. The results reported by Deane are based on three runs of about 2100 jobs each and the results of Conway are based on one run of 8700 jobs.

The absolute value of the results reported is not very important due to the difference in parameter values used. The important consideration is the relative performance of the three dispatching rules used. Of course, differences are to be expected even in the relative values shown in the Tables. These differences are caused by "structural" variations in the shops used. For example, the shop used by Conway had nine machines while Deane's and this one had ten machines. Also, the due date generation process used in this thesis is different than the one employed by Deane and the one used by Conway is unknown.

Table 6. Comparison with Conway's Results on Mean Flow Time

Rule	Conway's Results (p 232)		Thesis Results	
	Actual	Percent	Actual	Percent
DSOP	74.0	218.	74.6	154.
SPT	34.0	100.	48.4	100.

Table 7. Comparison with Conway's and Deane's Results on Standard Deviation of the Lateness Distribution

Rule	Conway's Results (p 232)		Deane's Results (p 41)		Thesis Results	
	Actual	Percent	Actual	Percent	Actual	Percent
DSOP	4.15	100.	26.9	100.	28.	100.
SPT	66.5	160.	53.3	199.	59.5	213.

Table 8. Comparison with Conway's and Deane's Results on Work-in-Process Levels (Hours)

Rule	Conway's Results (p 224)		Deane's Results (p 42)		Thesis Results	
	Actual	Percent	Actual	Percent	Actual	Percent
SPT	545	100.	661	100.	472	100.
EWIQ	709	130.	720	109.	553	117.
FCFS	1078	198.	815	123.	657	139.

The results shown in Tables 6-8 indicate noticeable differences in the absolute value of the measures of performance for the various shops. The relative differences are smaller, but still significant. The directions of movement for all the measures shown, however, from one dispatching rule to another is the same for Conway's, Deane's and this thesis. It is felt that these results indicate the reasonableness of the shop model used and, therefore, the program can be considered validated.

CHAPTER VII

ANALYSIS OF RESULTS OF THE SIMULATION RUNS

The results of the computer simulation runs are discussed in this chapter. The results of all runs are given in Detail in Appendices I and H. In this chapter, the more significant results have been summarized and presented in tabular form. The chapter is divided in three sections. The first section concentrates on the effects of the management or desired load factor (DESLF) of the job loading algorithm for the control of various measures of performance.

The second section analyzes the effects of the job pool and the loading algorithm when various dispatching rules are used. The improvements obtained in this area were the main objective of the research. The third section analyzes the results obtained under various special shop conditions. The results on these last two sections are based on t tests (Table 53), ANOVA (Table 54) and Duncan Ranking Tests (Table 55).

7.1 Effect of Changes in the Management Load Factor (DESLF)

The effect of changing "DESLF" is equivalent to changing the value of the C_j's in the mathematical formulation of Chapter IV. The results are shown in Table 43 for the DSOP dispatching rule and Table 44 for the SPT dispatching rule. Table 44 illustrates the effects obtained. A reduction in the DESLF parameter causes a reduction in the desired load used in the algorithm since the relation between the two is the following:

Desired Load = (DESLF) × (Scheduling Period)

In this research the scheduling period is eight hours, therefore, for a DESLF of 4.25, the management desired load used is 34 hours. The management desired load is the aggregate shop load that the algorithm attempts to maintain in the shop every scheduling period. As the DESLF decreases, the algorithm attempts to maintain a lower amount of work in the shop, but at the same time attempts to minimize the absolute deviation from desired balance while loading jobs with close due dates. Starting from a relatively high management load factor (DESLF), the following basic effects are observed as the DESLF value is reduced (up to a point):

- Average time spent in the system by a job increases.
- Average time spent in the shop by a job decreases.
- All balance measures improve.
- Average number of jobs in the pool before and after loading increases.
- Average hours of work in process in the shop decrease.
- Average hours of work done for jobs in the shop decrease.
- Variance of the Lateness Distribution decrease.
- There is a very small reducing trend in job tardiness.

The net effect of reducing the management load factor (DESLF) is to keep off the factory floor extra jobs that couldn't be worked on anyway. This condition is illustrated by the progressively shorter shop flow time shown by the jobs as the management load factor (DESLF) decreases. It is fairly obvious that the total hours of work in process in the shop should be reduced as the DESLF is reduced and more jobs are kept in the job pool. It is more interesting, however, to note that the hours of work done for jobs in the shop also goes down. When the shop is overloaded with jobs, there are many jobs for which one or two operations have been performed, but the jobs still stay in the shop waiting to have the final operations done.

The number of jobs in the pool increases because the algorithm has a smaller requirement from the shop and, therefore, tends to be more selective in loading jobs from the pool. The balance measures improve for the same reason, that is, there are more jobs in the job pool to choose from.

The variance of the lateness distribution is decreased, when the dispatching rule is Shortest Processing Time (SPT), as the Management Load Factor (DESLF) is reduced because this dispatching rule does not explicitly consider due date. Under this condition, the improved shop balance obtained with the smaller DESLF parameter and the due date term in the objective function of the loading algorithm produce a smoother job flow through the shop. This more than offsets the fact that jobs are placed in the shop at a later time. A due date oriented dispatching rule such as Dynamic Slack per Operation (DSOP) does not cause the conditions described in the paragraph above to occur.

The results obtained when the management load factor (DESLF) is reduced do not of course, continue indefinitely. There is a range of values for DESLF where many of the measures of performance start to move in an opposite direction or where the values stay basically constant.

7.2 The Job Pool and the Loading Algorithm with Various Dispatching Rules

The program employing the shop control methodology, that is, with the job pool and the loading algorithm was replicated five times with different random number seeds. It was desirable to test the effects of the loading algorithm when compared to an "uncontrolled" loading scheme whereby all jobs are released to the shop floor as they are consigned. Thus, five runs were made with the same random seeds for the uncontrolled shop model. The results for the DSOP dispatching rule are shown in Table 9. These results show the average value obtained in the five replications. Similar results for other dispatching rules are shown in Tables 10 to 12.

The detailed discussion and interpretation of results that follow will generally show that very significant improvements were obtained in most balance indices, except the Machine Work Balance Index (MWB) where the results obtained varied depending on the dispatching rule. The Work in Process measures showed consistent improvements and the results with respect to due date measures were mixed.

It can be seen in Table 9 that, when the dispatching rule is DSOP, there is virtually no difference in MWB but there is a 36% improvement in the SWB index. Other balance measure indices showing very significant improvements are QWB with a 39% reduction (improvement) and PQB with a 65% improvement.

The work in process measures were also significantly improved with the job pool and the loading algorithm. The total hours of work in the shop were reduced by 16.5% and the hours of work done for jobs in the shop were reduced by 31.6%. The variance of the lateness distributions on the other hand, was increased by 62% and the average tardiness changed from 2.01 hours to 24.7 hours. A reduction in these last two measures,

Table 9. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Ten Machines. Dispatching Rule is DSOP.

Measures of Performance	Uncontrolled Shop (From Table 35)	Job Pool, Loading Algorithm (From Table 47)
Aggregate Deviations from Balance	157.	120.
Balance Index, MWB	4.97	5.11
Balance Index, SWB	.893	.571
Balance Index, QWB	14.4	8.84
Balance Index, PWB	4.12	4.58
Balance Index, PQB	73.1	25.6
Work in Process, hours	634	529
Work done for jobs in shop	231	158
Variance of the lateness dist.	784	1276
Average Tardiness	2.01 hours	24.7 hours

Table 10. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against One "Uncontrolled" Shop. Ten Machines. Dispatching Rule is EWIQ.

Measures of Performance	Uncontrolled Shop (From Table 36)	Job Pool, Loading Algorithm (From Table 40)
Aggregate Deviation from Balance	153	131.
Balance Index, MWB	5.19	5.24
Balance Index, SWB	1.291	1.220
Balance Index, QWB	9.03	6.47
Balance Index, PWB	3.94	4.06
Balance Index, PQB	61.9	47.6
Work in Process, hours	553	495
Work done for jobs in shop	258	228
Variance of the lateness dist.	4296	3629
Average Tardiness	13.4	12.1

Table 11. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Ten Machines. Dispatching Rule is SPT.

Measures of Performance	Uncontrolled Shop (From Table 45)	Job Pool, Loading Algorithm (From Table 46)
Aggregate Deviation from Balance	156.	84.4
Balance Index, MWB	5.15	4.69
Balance Index, SWB	1.16	.442
Balance Index, QWB	3.65	2.07
Balance Index, PWB	4.04	4.28
Balance Index, PQB	17.3	4.85
Work in Process, hours	471	366
Work done for jobs in shop	150	120
Variance of the lateness dist.	3218	2217
Average Tardiness	6.89	13.1

Table 12. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Ten Machines. Dispatching Rule is FCFS.

Measure of Performance	Uncontrolled Shop (From Table 38)	Job Pool, Loading Algorithm (From Table 42)
Aggregate Deviation from Balance	153.	94.8
Balance Index, MWB	5.00	4.99
Balance Index, SWB	.781	.684
Balance Index, QWB	15.3	8.67
Balance Index, PWB	4.26	4.34
Balance Index, PQB	130.	40.
Work in Process, hours	657	540
Work done for jobs in shop	249	201
Variance of the lateness dist.	2967	2670
Average Tardiness	14.9	16.4

usually involving some tradeoff with the balance and WIP measures could be accomplished by changing the weighting factor in the due date term of the loading algorithm or by forcing the jobs from the pool to the shop sooner. This second approach is illustrated in Table 50. Also Tables 51 and 52 illustrate the insignificance of the tardiness measure in either case when the jobs have looser due dates.

A t test for paired observations (Ostle, 1963) has been used to test for the equality of pairs of means. Results of these tests are given in Table 53.

A computed t value of 10.9 for SWB is greater than the $t_{.99}$ (4) table value of 3.747. The job pool and loading algorithm have thus improved significantly the SWB index. Other statistically significant improvements in measures of performance consisted of the QWB balance index and the hours of work done for jobs in the shop.

Even more dramatic improvements are observed when the SPT dispatching rule is employed.

In this case the improvement in the MWB index is a respectable 9% while the SWB index is reduced by 62%. The Period Queue Balance Index is reduced from 17.3 to 4.85, an improvement of 72%.

The work in process measures, total hours of work in the shop and hours of work done for jobs in the shop, are reduced by 22.3% and 20%, respectively.

Finally, the average tardiness is increased from 6.9 hours to 13.1 hours, but there is a reduction in this case (as opposed to the increase with DSOP) on the variance of the lateness distribution from 3218 to 2217 for a 34.1% improvement. The results obtained in the SWB and QWB indices and in the Work in Process measures are significant at the 99% level, while the improvements in the MWB index and the variance of the lateness distribution are significant at the 99% level. The increase in average tardiness is also significant at the 99% level.

The results obtained with dispatching rules EWIQ and FCFS are of a similar nature to the ones already described, although in these cases no experimental search was made for the best range for DESLF. These results are given in Tables 10 and 12.

An analysis of variance was performed utilizing the results of Tables 35 to 38 to test the differences in the effects of the four dispatching rules. The calculated F values are given on Table 54. It can be seen that the MWB index does not change significantly for the various dispatching rules. However, SWB, QWB, PWB, the work in process measures and the timeliness measures show significant differences when the four dispatching rules are used. For example, the calculated F value for average hours of work in process is 75.15 which greatly exceeds the $F_{.99}$ (3, 16) value of 5.29. This is not a new result since it has been reported before by Conway and others (1967) and also in many other works.

Another analysis of variance was performed using the results of Tables 40, 42, 46 and 47 to determine the effect, if any, of various dispatching rules on the measures of performance studied when a job pool and the loading algorithm were used. The calculated ANOVA values are given on Table 54. The results are basically the same as in the ANOVA described in the preceding paragraph, except that the conclusion that there is some difference in the four dispatching rules with respect to the average tardiness can not be reached this time. In addition to the ANOVA, Duncan

Ranking Tests as described by Hicks (1964) were also performed to identify the dispatching rules with significant differences in the measures of performance. These results are presented in Table 55.

7.3 Other Results Obtained

7.3.1 Variations in Shop Arrival Patterns

One of the arrival patterns used assumed Poisson arrivals, while the other arrival pattern as was explained in Chapter VI was obtained by superimposing the exponential interarrival times on a sine curve. This created a fluctuating or dynamic mean interarrival time. The results for each condition are shown in Table 13. The Shop Balance Index increases by 50% when fluctuating arrivals are introduced. The reason for this is that some of the variability of the arrival rate filters through the shop and is seen also in the departure rate. The other measures where significant differences at the 99% level are detected are WIP (hours), PWB and PQB. The difference for the WIP (hours) was only a 4.1% increase in the hours for the case with fluctuating arrivals, but this became significant due to the small variability observed over the various replications. The conclusion in this case is that a fluctuating arrival rate of the magnitude used here causes most measures of performance to have a less favorable value than when a flat arrival rate (pure Poisson arrivals) is employed. The variance of the lateness distribution is the notable exception, but the results in this case are not significant at the 99% level.

Table 13. Comparison of Results in an "Uncontrolled" Shop Obtained when Arrivals Are Generated by a Distribution with a Static Mean vs Results when a Dynamic Mean Was Employed. Dispatching Rule is DSOP.

Measures of Performance	Flat Arrivals (From Table 29)	Fluctuating Arrivals (From Table 35)
Aggregate Deviation from Balance	138	157
Machine Balance, MWB	4.83	4.97
Shop Balance, SWB	.595	.893
Queve Workload Balance, QWB	13.5	14.4
Period Workload Balance, PWB	4.27	4.12
Period Queue Balance, PQB	109	73.1
Work in Process, hours	609	634
Work done for jobs in shop	227	231
Variance of Lateness Dist.	894	784
Average Tardiness	2.05	2.01

Note: The results shown are the average of 5 runs.

7.3.2 Shop with Few Interactions

The effects produced by the job shop control methodology in a shop with few interactions are illustrated by simulations performed in a shop with five machines and where the average number of operations per job is only two. The comparison of these results with results obtained in the same shop while operating under "controlled" conditions are shown in Table 14. They are of the same type as those obtained for the larger shop when the same dispatching rule (DSOP) was used, except that the percentage improvements obtained by the job pool and the loading algorithm are even more dramatic here. The balance indices are reduced as follows:

MWB - 16%

SWB - 46%

OWB - 42%

PWB - .5%

POB - 83%

The work in process measures are reduced by 43% (total hours in the shop) and 65% for hours of work done for jobs in the shop. The variance of the lateness distribution shows a 12.5% increase and the average tardiness increased from .2 to 6.5 hours. The results where the improvement was statistically significant at the .99 level were the deviation from balance, the MWB, SWB, QWB indices, and the work in process measures.

The other measures were not shown to be statistically significant using the paired observation t test due to the large variance in the observed samples. There is no question, however, that the percentage improvements obtained do have practical significance.

Table 14. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Small Shop (5 Machines) with Few Interactions. Dispatching Rule is DSOP.

Measures of Performance	Uncontrolled Shop (From Table 31)	Job Pool, Loading Algorithm (From Table 32)
Aggregate Deviation from Balance	41.6	26.5
Machine Workload Balance, MWB	5.67	4.76
Shop Workload Balance, SWB	1.54	.831
Queue Workload Balance, QWB	28.5	16.3
Period Workload Balance, PWB	4.19	3.99
Period Queue Balance, PQB	166	27.5
Work in Process, hours	161	91.7
Work done for jobs in shop	38.5	13.6
Variance of the Lateness Dist.	1158	1303
Average Tardiness	.21	6.50

The reason why the job pool and the loading algorithm cause an even greater improvement in this case is because the balancing features of the model have a greater effect on the jobs since they remain in the shop for an average of two operations only. In this form, the shop interactions have a much smaller chance of disrupting the work done at loading time.

7.3.3 Special Loading Modifications

It was desired to investigate the effect of several variations of the releasing of jobs provided by the loading algorithm.

The loading provided by the algorithm is done normally once every scheduling period. It is apparent that in practical situations a shop should be flexible enough to expedite jobs to idle machines if the need arises. It was desirable to test this feature as a modification or extension of the basic loading algorithm.

Modification 1

This condition consists of putting a job directly in the shop, without passing through the pool, if the machine which is to perform the job's first operation is idle at the time the job arrives in the shop.

Modification 2

This option is put into use when a job is finished by a machine and that machine queue is empty. Under this condition, the pool is then searched to see if any job from the pool uses the machine in question for its first operation.

Modification 3

This option provides a modification of conditions 1 and 2. It allows conditions 1 and 2 to take place only if the machine in question has not yet performed its average amount of work in the scheduling period.

Modification 4

This option operates as follows: after loading from the pool using the loading algorithm, a check is made to see if a match is found between an idle machine and the first operation number of a job in the pool. If this match is found, the job is loaded immediately. Condition 4 can be used by itself or with options 1, 2, and 3. It should not be used with options 1 and 2 alone because it would be redundant in that case.

The results obtained with these special loading conditions are shown in Tables 26 to 28. These special loading modifications were investigated to test the shop control methodology under various shop conditions. It was observed that the improvements obtained for most balance measures and for work in process levels were maintained. No significant improvement was obtained in the MWB index, however, and it must, therefore, be concluded that to obtain changes in this index, it is necessary to get into the shop and "direct traffic" from machine to machine.

7.3.4 Results Obtained with a Loading Heuristic

The loading heuristic utilized was explained in Chapter IV and briefly consisted of loading a job in the shop if the first job operation made a contribution to the queue of a machine that was underloaded at the time. The complete list of jobs in the pool was examined every period but no attention was paid to the contribution of the second and succeeding operations. The results obtained with this heuristic method are compared to those obtained with the loading algorithm (both using the job pool) and they are shown in Table 15.

Table 15. Comparison of Results Obtained by Using the Job Loading Algorithm and a Loading Heuristic. Job Pool is Used. Dispatching Rule is DSOP.

Measures of Performance	Job Loading Algorithm (From Table 39)	Loading Heuristic (From Table 30)
Aggregate Deviation from Balance	95.7	130 hours
Machine Workload Balance, MWB	5.04	5.09
Shop Workload Balance, SWB	.674	.830
Queue Workload Balance, QWB	9.51	11.4
Period Workload Balance, PWB	4.40	4.30
Period Queue Balance, PQB	32.8	43.6
Work in Process, hours	550 hours	588 hours
Work done for jobs in shop	181 hours	210 hours
Variance of Lateness Dist.	1029	854
Average Tardiness	14.6 hours	5.44 hours

The results obtained with the loading algorithm are better in most areas except in the variance of the lateness distribution and the average tardiness. The improvements are significant at the 99% level in the cases of the SWB and QWB indices as well as with the work in process measures. Some improvement is obtained, however, by the loading heuristic in some of the measures when results are compared to those obtained when the shop operates under "uncontrolled" loading conditions. It must be concluded, therefore, that the improvements reported elsewhere in this research have been produced jointly by the use of the job pool concept and the loading algorithm.

7.3.5 Shop with a Non-Symmetric Transition Matrix

A shop with a non-symmetric transition matrix is one in which special work flow patterns can be identified. The matrix used is shown in Figure 5. The comparison of results obtained in a shop with specific job flow structure when the shop control methodology is used and those obtained for the same shop under "uncontrolled" shop loading conditions are illustrated in Table 16.

It can be seen that sizable improvements were obtained for SWB, QWB, and PQB as well as the work in process measures. These are the same type of results obtained for the pure job shop when the same dispatching rule used here, Dynamic Slack per Operation, is employed.

7.3.6 Shops with Alternate Selections of Machines in a Machine Pair

This feature allows for some alternative routing characteristics in the shop. Specifically, the shop is treated as if it consisted of pairs of machine groups with the odd numbered machine group and the even numbered group immediately following it making up a pair. Thus each machine

from machine \ to machine	1	2	3	4	5	6	7	8	9	10
1	0	.16	.01	.20	.01	.01	.30	.01	.10	.20
2	.26	0	.01	.01	.30	.10	.20	.10	.01	.01
3	.20	.01	0	.01	.09	.26	.01	.40	.01	.01
4	.01	.10	.36	0	.01	.01	.01	.01	.20	.29
5	.01	.20	.30	.01	0	.10	.07	.10	.20	.01
6	.10	.01	.01	.35	.01	0	.11	.01	.20	.20
7	.20	.01	.01	.20	.27	.10	0	.10	.01	.10
8	.02	.30	.10	.20	.01	.10	.10	0	.10	.07
9	.10	.20	.10	.01	.10	.02	.10	.26	0	.11
10	.10	.01	.10	.01	.20	.30	.10	.01	.17	0

Example of a likely 6 operation path - 1,7,5,3,8,2

Example of an unlikely 6 operation path - 1,3,2,4,7,9

Figure 5. Non-Symmetric Transition Matrix

Table 16. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Shop with a Non-Symmetric Transition Matrix. Dispatching Rule is DSOP.

Measures of Performance	Uncontrolled Shop (From Table 33)	Job Pool, Loading Algorithm (From Table 34)
Aggregate Deviation from Balance	164	86.2
Machine Workload Balance, MWB	5.03	5.11
Shop Workload Balance, SWB	.806	.546
Queue Workload Balance, QWB	14.0	9.09
Period Workload Balance, PWB	4.25	4.59
Period Queue Balance, PQB	40.9	18.5
Work in Process, hours	647	545
Work done for jobs in shop	230	185
Variance of the Lateness Dist.	727	934
Average Tardiness	2.06	11.3

in the shop has a companion machine. Both machines do the same type of work such that jobs can be interchanged on the two machines. The shop operates in its normal way except that when a given machine becomes idle, the queue of its companion machine is checked to see if there are any jobs in it so that it can be transferred to the idle machine. Also when a job is first placed in a machine queue, the status of the companion machine is checked to verify that it is not idle. The purposes in using this feature were to investigate the effect on shop balance measures in general of having a shop with this additional flexibility and also to check on the usefulness of the pool concept and the loading algorithm under these conditions.

A group of simulation runs for shops in which the alternate selection of machines in a machine pair was allowed were performed. The DSOP dispatching rule was used in all cases.

Table 17 compares the results in a traditional shop with those in which the alternate routing feature was allowed. These runs did not utilize the loading algorithm. The hours of work in process were reduced by the use of the alternate machine feature by 32.3% and the hours of work done for jobs in the shop were reduced by 25.6%. The calculated t statistic for these measures were 50.4 and 21.7respectively, while the tabulated value for $t_{.99}$ (4) is 3.747. It can, therefore, be said that the work in process measures are improved by the use of the alternate machine feature. This result is not surprising since improvements in work in process measures and mean flow times (the improvement in average time in the shop in this research was close to 40%) when some sort of alternate machine

Table 17. Comparison of Results in a Traditional Shop with a Shop Where Alternate Routing is Allowed. Shop is "Uncontrolled". Dispatching Rule is DSOP.

Measures of Performance	Traditional Shop (From Table 35)	Alternate Routing (From Table 48)
Aggregate Deviation from Balance	157	280.
Machine Workload Balance, MWB	4.97	4.61
Shop Workload Balance, SWB	.893	1.58
Queue Workload Balance, QWB	14.4	7.04
Period Workload Balance, PWB	4.12	3.05
Period Queue Balance, PQB	73.1	49.7
Work in Process, hours	634 hours	429 hours
Work done for jobs in shop	231	172
Variance of the Lateness Dist.	784	1288
Average Tardiness	2.01	.46

scheme is used has already been reported by Conway and others (1967).

The main purpose, however, of investigating the alternate machine feature in this research was in relation to its effect on shop balance measures. The MWB, QWB, PWB and PQB indices show improvements of 7.2%, 51%, 26%, and 32% respectively with the improvements in the MWB, and PWB shown to be significant at the 99% level by the paired observation t test.

The surprising result is that the SWB index and the variance of the lateness distribution show a significant increase when the alternate machine feature is used. The calculated t values are 9.07 and 16.4 while $t_{.99}$ (4) is 3.747.

A possible explanation for this shop behavior is that the alternate machine feature causes greater fluctuation in shop output by pushing out a lot of work in some periods which can not be maintained over the long run. Table 18 illustrates the same type of comparison as Table 17, but in this case the job pool and loading algorithm are used.

The direction of the improvements observed in this case are similar to the ones observed when the job pool was not used except that the magnitude of the improvements obtained by the use of the alternate machine feature are somewhat larger this time. The SWB index shows again an increase, but the variance of the lateness distribution does not experience a significant change this time.

Finally Table 19 deals with a shop in which the alternate routing feature is used and the dispatching rule is DSOP. The Table shows a comparison of a shop with a job pool and the loading algorithm against one operating under "uncontrolled" loading conditions. Very significant

Table 18. Comparison of Results in a Traditional Shop with a Shop Where Alternate Routing is Allowed. Dispatching Rule is DSOP; a Job Pool and the Loading Algorithm Are Used.

Measures of Performance	Traditional Shop (From Table 47)	Alternate Routing (From Table 49)
Aggregate Deviation from Balance	120.	260.
Machine Workload Balance, MWB	5.11	4.24
Shop Workload Balance, SWB	.571	.767
Queue Workload Balance, QWB	8.84	2.81
Period Workload Balance, PWB	4.58	3.48
Period Queue Balance, PQB	25.6	3.84
Work in Process, hours	529	311
Work done for jobs in shop	158	97.8
Variance of the Lateness Dist.	1276	1199
Average Tardiness	24.7	9.36

Table 19. Comparison of Results Obtained by Using a Job Pool and the Loading Algorithm Against an "Uncontrolled" Shop. Shop with 10 Machines and Where Alternate Routing is Allowed. Dispatching Rule is DSOP.

Measures of Performance	Uncontrolled Shop (From Table 48)	Job Pool, Loading Algorithm (From Table 49)
Aggregate Deviation from Balance	280.	260.
Machine Workload Balance, MWB	4.61	4.24
Shop Workload Balance, SWB	1.58	.767
Queue Workload Balance, QWB	7.04	2.81
Period Workload Balance, PWB	3.05	3.48
Period Queue Balance, PQB	49.7	3.84
Work in Process, hours	429	311
Work done for jobs in shop	172	97.8
Variance of the Lateness Dist.	1288	1199
Average Tardiness	.46	9.36

improvements are also obtained this time. For example, the SWB and QWB indices are reduced by 51% and 60%. The hours of work in process are reduced by 27.5% and the hours of work done for jobs in the shop by 43%. All four of these measures showed a significant improvement at the 99% level. The average tardiness had an increase from .46 to 9.36 hours and the variance of the lateness distribution showed a small, but non-significant improvement when the job pool was used. This is somewhat surprising since the variance of the lateness distribution increased when the shop control methodology was employed under the DSOP dispatching rule and the alternate machine feature was not used (see Table 9).

CHAPTER VIII

EXTENSIONS OF THE BASIC LOADING METHODOLOGY

The extensions that follow were developed during the course of the research for this dissertation. They are not a necessary part of the central research theme and therefore have not been used. They are presented here, however, so that they can serve as possible starting points for future research.

The basic idea of loading and of the loading algorithm are extended below to the area of dispatching and then a model is proposed to use the operations to be performed in a given period while jobs are selected from the pool. Finally the desirability of combining the research presented here and the work of Deane (1972) into a single methodology is discussed.

8.1 A Dispatching Model Using the Same Concepts Employed by the Loading Model

The concept presented here consists of treating each machine queue as a "job pool" and giving priority at that machine to that subset of jobs which minimizes the deviation from balance for the rest of the shop as a whole. Conventional dispatching rules (SPT, DSOP, FCFS, etc.) can then be used to rank the subset selected.

This extension, while considering the loading or releasing problem, looks at the dispatching problem in a way similar to that used by Deane (1972).

The differences are that Deane used an elimination scheme to arrive at the subset of eligible jobs and then a repeated search to obtain the actual jobs to be worked on. Here the loading idea is added and a mathematical programming approach is used to select the subset of eligible jobs.

The objective function minimizes the deviation between actual and desired (management goals) aggregate loads for each machine. Deviation values are obtained both from the pool loading constraints and the queue loading constraints. The due date term employed in the loading algorithm in Chapter IV is also used here.

A modification of the objective function is also needed to assign some weight to those jobs with large in process inventory value. It is assumed that this can be determined from the number of work hours already spent on the job. The formulation requires m^2 constraints where m is the number of machines.

Notation

i(o)	job index for jobs in the pool
j	machine index (m machines)
No	number of jobs in the pool
i(j)	job index for jobs in the queue at machine j (also including the job being worked on)
$X_{i(o)} = 0$	job pool decision variable -- job not selected
$X_{i(o)} = 1$	job pool decision variable -- job selected
$X_{i(j)} = 0$	job queue decision variable -- job not selected
$X_{i(j)} = 1$	job queue decision variable -- job selected
W_{ij}	amount of work contributed by job i to machine center j

(work not yet performed on job i)

V_{ij} same as W_{ij}, but this time referring to work already performed on job i

P_j present load in the shop (not in the pool) ahead of machine j

$$P_j = \sum_{k=1}^{m} \sum_{i(k)=0}^{n_k} W_{i(k)j}$$

$B_{j\ell}$ present load in the shop (not in the pool) ahead of machine j, but not including the work in the queue of machine ℓ

$$B_j = \sum_{\substack{k=1 \\ k \neq \ell}}^{m} \sum_{i(k)=0}^{n_k} W_{i(k)j}$$

Q_j present load in the shop for machine j loaded in the queue at machine j

$$Q_j = \sum_{i(j)=1}^{n_j} W_{i(j)j}$$

P_{oj} present load in the pool for machine j

$$P_{oj} = \sum_{i(o)=1}^{n_o}$$

C_j desired load in the shop for machine j

F_j desired load in the queue for machine j

S_{jL}, S_{jH} deviation from desired aggregate load in the shop (except queue of machine k) for machine j after loading jobs for machine k

The formulation then consists of the following:

Pool Loading Constraint for Machine j:

$$\sum_{i(o)=1}^{n_o} W_{i(o)j} X_{i(o)} + P_j + S_{jL} - S_{jH} = C_j \qquad j=1,2,...,n$$

These constraints indicate that the contribution of jobs selected from the pool to the aggregate shop load for machine j plus the existing aggregate load in the shop for machine j plus (minus) any shortage load (any excessive load) released must equal the total desired aggregate load in the shop established by management.

Queue Loading Constraint for Machine j when Loading Jobs at Machine k:

$$\sum_{i=1}^{n_k} W_{i(k)j} X_{i(k)} + B_{jk} + q_{jkL} - q_{jkH} = \frac{m-1}{m} C_j$$

$$\text{for } j=1,2,...,k-1,k+1,...,m$$
$$k=1,2,...,m$$

The queue loading constraint for machine j when loading jobs at machine k considers the shop as if the jobs at machine k were in a job pool outside the shop. This constraint then indicates that the aggregate workload for machine j contributed by those jobs given priority at machine k plus the present load in the shop (not in the pool) ahead of machine j without including the jobs in the queue of machine k plus or minus any deficiency or excess of work equals the amount desired by management. It must be noted that, due to the constraint structure, at least one of the two slack variables in the equation will be equal to zero.

Non Negativity and Integer Constraints:

Objective Function All X's = 0,1

$$D = \sum_{j=1}^{m} a_{jL} S_{jL} + \sum_{j=1}^{m} a_{jH} S_{jH} + \sum_{\substack{j=1 \\ j \neq k}}^{m} \sum_{k=1}^{m} (q_{jkL} + q_{jkH})$$

$$- \sum_{j=o}^{m} \sum_{i=1}^{n_j} f(d_i) X_{i(j)} - K_2 \sum_{j=1}^{m} \sum_{i=1}^{n_j} V_{i(j)} X_{i(j)}$$

The objective function minimizes the sum of the deviations from desired loading from the job pool and the individual machine centers. It also includes a term to make jobs increasingly attractive loading candidates as their due date approaches and as the investment on a job, given by the work already performed on it, increases.

8.2 The Aggregate Loading Problem Using Multiple Operations in the Horizon

In aggregate scheduling problems, the number of time periods to be planned is called the planning horizon. Generally, the length of the planning horizon should be such that the addition of one more period to the planning horizon would have little effect on the production rate decisions in the early periods.

For example, according to Holt, Modigliani, Muth, and Simon (1960), since each period's decision has cost implications that extend over an appreciable length of time, this cost function must span sufficient time to include virtually all of the cost implications of the decision.

The formulation that follows does not employ time periods or scheduling periods (although a loading decision is made every scheduling period). The formulation uses "operation in a machine" as the planning period. This is based on the argument that the critical time element in a job shop is the number of operations to be accomplished, Ackerman (1963). Along the line of the HMS argument, the penalty function will have to span sufficient operations into the future to include virtually all of the shop balancing implications of the decision.

<u>Notation (n jobs, m machines)</u>

W_{ijt}	amount of work contributed by job i to machine center j on their immediately next plus t^{th} operation
$\sum W_{ijt} = W_{ij.}$	amount of work contributed by job i to machine center j
P_{jt}	present load in the shop for machine j, t operations away from machine j
P_{jo}	load in the queue for machine j
$\sum_t P_{jt} = P_{j.}$	present load in the shop ahead of machine j
C_{jt}	desired load in the shop for machine j, t operations away from machine j
C_{jo}	desired load in the queue for machine j
$\sum_t C_{jt} = C_{j.}$	desired load for machine j in the shop

The following should hold among the C_j's and make the job of developing them easier

If $\quad k_1C_{10} = k_2C_{20} = k_3C_{30} = \ldots = k_mC_{mo}$

Then $\quad k_1C_{1t} = k_2C_{2t} = k_3C_{3t} = \ldots = k_mC_{mt} \qquad$ for $t = 1,2,3,\ldots$

Of course this in no way implies that

$$C_{jo} = C_{jt} \qquad \text{for any } t$$

since the absolute values of the C_j's with respect to the C_{jo} must recognize the additional loading that the pool will effect during future scheduling periods.

a_t — weighting factors to be used in the objective function to attach different penalties to the deviations from balance right now at the queue, in the entire shop, 1 operation away, etc.

<u>Formulation</u>

$$\text{Min } D = a_{\cdot} \sum_{j=1}^{m} (S_{jL\cdot} + S_{jH\cdot}) + \sum_{t=1}^{t^*} a_t \left[\sum_{j=1}^{m} (S_{jLt} + S_{jHt}) \right]$$

subject to:

$X_i = 0,1$

$S_{jL\cdot} \geq 0, \; S_{jH\cdot} \geq 0$

$S_{jLt} \geq 0, \; S_{jHt} \geq 0$

$$\sum_{i=1}^{n} W_{ij\cdot} X_i + P_{j\cdot} + S_{jL\cdot} - S_{jH\cdot} = C_{j\cdot} \qquad j = 1,2,\ldots,m$$

$$\sum_{i=1}^{n} W_{ijt} X_i + P_{jt} + S_{jLt} - S_{jHt} = C_{jt} \qquad j = 1,2,\ldots,m; \quad t = 0,1,2,\ldots,t^*$$

where t^* is the operations horizon

The objective function is the deviation in work hours from a desirable and pre-established shop condition. The constraints contain positive and negative slack variables to indicate the excessive or deficient work loaded for a machine group. One of the two slack variables will be zero in each constraint, and the other is used in the objective function. There is a constraint for the aggregate shop load for each machine center and also constraints for the work 1,2,3, etc. operations removed from each machine center.

If the future periods are handled on a "time" basis rather than an "operation" basis, then it is not possible to present a "loading only" model since the loading decisions required for balancing "t" periods into the future are going to depend also on the dispatching decisions made during that time. Such a loading and dispatching model with a planning horizon should not be too difficult although the notation required will be cumbersome.

8.3 Combination of Dispatching and Loading Algorithms

The model in the first section of this chapter attacked the problems of dispatching and loading on an integral basis by looking at the dispatching problem as if it were a loading problem. However, it was observed that the number of resulting constraints is large and the model is rather awkward.

On the other hand, it has been shown by this research that the loading methodology presented here improves the shop workload balance measure (SWB) and other balance measures as well as work in process measures considerably. However, the results obtained with the machine workload balance measure (MWB) have been mixed. Modest improvements were obtained with some dispatching rules (SPT) and no improvement at all with some others (DSOP). The reason for this is that the job goes through too many operations in the shop without any "balance" control after being loaded from the pool. The results obtained by Deane (1972) with his dispatching method give practically the opposite results and job control is maintained at every operation in the shop. The combination of the loading methodology presented in this research and the dispatching approach introduced by Deane is, therefore, a logical step which should be investigated by future researchers.

CHAPTER IX

CONCLUSIONS AND RECOMMENDATIONS

The objective of this research has been to develop a loading and balancing methodology for job shop control. This objective has been accomplished by the introduction of the job pool concept and the development of the job loading algorithm to select jobs from the pool.

In addition, a large number of shop balance measures have been identified and their applicability to different kinds of shops has been discussed. The validity and relevance of the shop balance concept as a measure of performance has been presented.

Significant improvements were obtained through the use of the job pool and the loading algorithm in most balance measures tested as well as in work in process measures related to both total work in the shop and to work performed for jobs in the shop. The results dealing with balance measures and work in process levels are closely related.

In fact, it can be argued that the reduction of work in process levels is a consequence of the better "balanced" shop because under the improved balance condition there is less interference in the shop and, therefore, a lower in process level is needed to maintain a certain work throughput level.

These results were not achieved without paying a price, however. The use of the loading and balancing methodology resulted in increases in the average job tardiness and the variance of the lateness distribution the dispatching rule was Dynamic Slack per Operation. On the other hand, the variance of the lateness distribution was reduced by the use of the loading algorithm when the dispatching rule was shortest processing time. It has also been observed that the results obtained with the job pool and the loading algorithm are highly dependent on the desired load level (management load factor) used in the algorithm. This parameter greatly influences the average number of jobs in the job pool and the loading methodology needs a reasonable number of jobs in the pool so that it can have flexibility in selecting the jobs to be loaded in the shop.

The work performed in this dissertation can result in significant practical applications and it also provides a good starting point for additional research. Among the important areas where additional research could be done are:

a) Testing the results obtained by incorporating into a single model the dispatching approach introduced by Deane (1972) and the loading ideas developed in this research. The combination of these two approaches was discussed in Chapter VIII. This combined model should offer the benefits of a shop with better overall balance provided by the shop control methodology and the ability to react to specific out-of-balance conditions that develop on the shop floor as provided by the "search" dispatching approach.

b) Extending and testing the algorithms presented in Chapter VIII. The first one consists of a dispatching model which utilizes the same concepts employed by the loading model. The second one is a loading model which considers the shop load not only in an aggregate basis but also takes into account the "timing" of work availability to the various machines. The testing required will consist of programming the algorithms and employing them in a job shop simulator to investigate their effect on various measures of shop performance.

c) Performing sensitivity analysis on the loading algorithm with respect to both the management load factor and the due date function. The performance of the algorithm is dependent on the desired aggregate load in the shop and also on the weight assigned to the due date term in the objective function. As the management load factor is increased, the aggregate load in the shop increases and the average job pool size decreases. This condition hurts the balance and work in process measures but improves the average tardiness.

An increase in the weight assigned to the due date term forces jobs into the shop earlier, at the expense of balance and work in process measures. The effect of changes in values of these two parameters is highly interrelated and the performance of detailed sensitivity analysis on them will add new understanding to the job shop behavior.

d) Investigating the sensitivity of the results obtained with respect to the scheduling period. The value of the management load factor that should be used is closely related to the scheduling period employed because as the scheduling period gets longer, more work hours should be loaded in the shop every period. The reason for this is that the times between job releases to the shop will be longer.

e) Investigating "loading" algorithms that control the job pool size in a more direct way than the algorithm presented in this research.

The loading algorithm exercises an indirect effect in the size of the job pool through the management load factor and the weight of the due date term. The results of the simulation have shown a high correlation between the size of the job pool and the value of most of the measures of performance related to shop balance. A new algorithm that recognizes this fact and makes use of it explicitly could possibly result in additional significant improvements for several shop measures of performance.

APPENDIX A

DESCRIPTION AND FLOW DIAGRAM

OF SIMULATION PROGRAMS

Table 20. Description of Simulation Subroutines

MAIN. This is the main program and its functions are to read in the parameters describing the simulation, initialize the non-GASP variables, and to call subroutine GASP which turns over control to the GASP II language.

EVNTS. This subroutine calls one of the four event subroutines (ENDSV, ARIVL, COLL, START).

START. It sets the simulation clock to zero and generates new arrivals to preload the shop. The new arrivals are placed in the job pool if a pool is being used and if the initial number of jobs desired in the pool has not been reached yet. Otherwise the new jobs are placed directly in the proper machine queues. New arrivals continue to be generated until the total number of jobs to be preloaded is reached.

ARIVL. The subroutine ARIVL generates the simulation clock time for the next job arrival to the shop and this time is set up as an arrival event in the GASP event file. It then generates the job attributes for the job that just came in, starting with the number of operations, and then the machine number and time for the first operation, other machine numbers from the job transition matrix and their times, and finally the job due date. The subroutine then assigns a file location to the job and moves it to the job pool or shop. It also contains options to handle the special loading conditions #1,3 described in Chapter VII.

Table 20. (Continued)

ENDSV. This subroutine is used every time a job finishes an operation at a machine. It must then collect shop statistics and depending on whether the job is leaving the shop at this time or not it must collect the terminal job statistic or update the job attributes and place it in the next queue.

The next task for this subroutine is to select from the queue of the machine that just finished an operation the next job to be processed. If the queue contains one or more jobs, statistics on job waiting times as well as shop workload must be calculated. On the other hand if the queue is empty, machine utilization statistics must be updated. This subroutine also contains instructions to handle the special loading conditions #2,3 given in Chapter VII.

COLL. COLL is a subroutine called only at the end of every scheduling period. Its main functions are to calculate and update statistics which are kept on a scheduling period basis and, if a pool is used, to call the matrix generator subprogram. In addition, this subroutine tests for the end of run-in period and end of simulation conditions and takes appropriate action if these conditions have occurred.

CLEAR. This subroutine is used only at the end of the run in period to clear and reset the arrays which keep the accumulated statistics. The shop status, of course, is left undisturbed.

Table 20. (Continued)

PTJOB. Subroutine PTJOB is responsible for placing a job in the job pool or in a machine queue or in the machine itself. Which one of these actions is taken depends on whether the shop is still being preloaded, on whether the job is a new arrival or not, whether a job pool is being used, and on the status of the machine itself. In addition to the above, statistics are collected on interarrival times to the pool and each machine. Once it has been decided to put a job in a machine, the workload in the machine status is changed and the time for the completion event is set if the machine was idle.

GENMAT. Subroutine GENMAT generates the matrix required by the loading algorithm to select those jobs that will be moved from the job pool to the shop. The matrix is generated by using job attributes contained in the job pool file.

This subroutine is by-passed if the job pool is not being used and it calls the proper loading routine, either the loading heuristic or the mathematical programming loading algorithm if the job pool is being used.

LPI. LPI is basically a simple linear programming code with the bounded variable feature. It then calls JOBDEC and transmits the values of the job decision variables to it.

JOBDEC. The function of JOBDEC is to decide which jobs will be moved from the job pool to the machine queues based on the value of the job decision variables given by LPI.

Table 20. (Concluded)

ENSIM.	This subroutine is called at the end of the simulation to print simulation results. In addition it has an option to start other simulation runs with a different dispatching rule and if this option is used, ENSIM must reinitialize the non-GASP variables and call GASP to begin the new run.
POOLHE.	This is the loading heuristic subroutine and was already explained in Chapter IV.

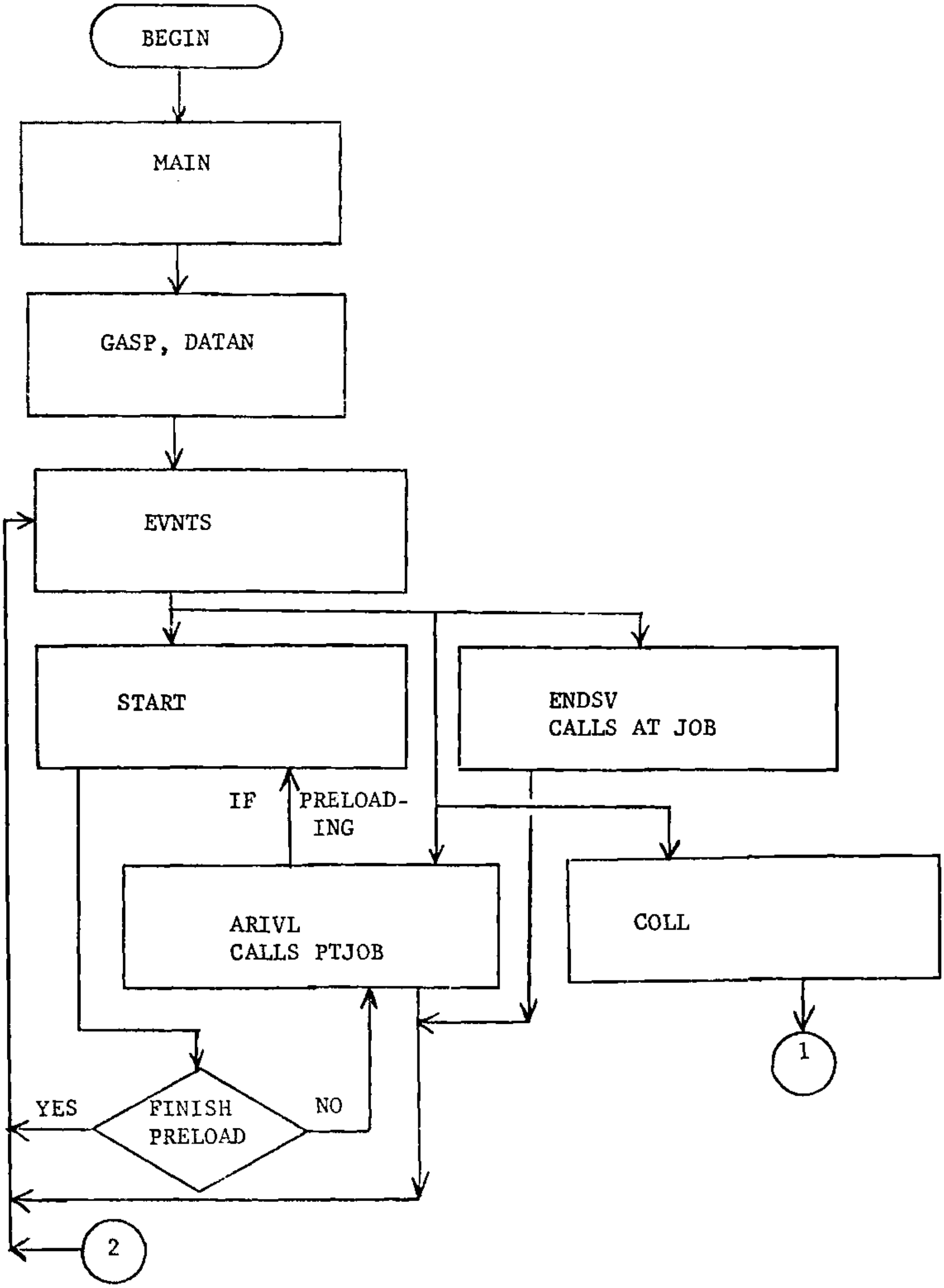

Figure 6. Job Shop Simulator

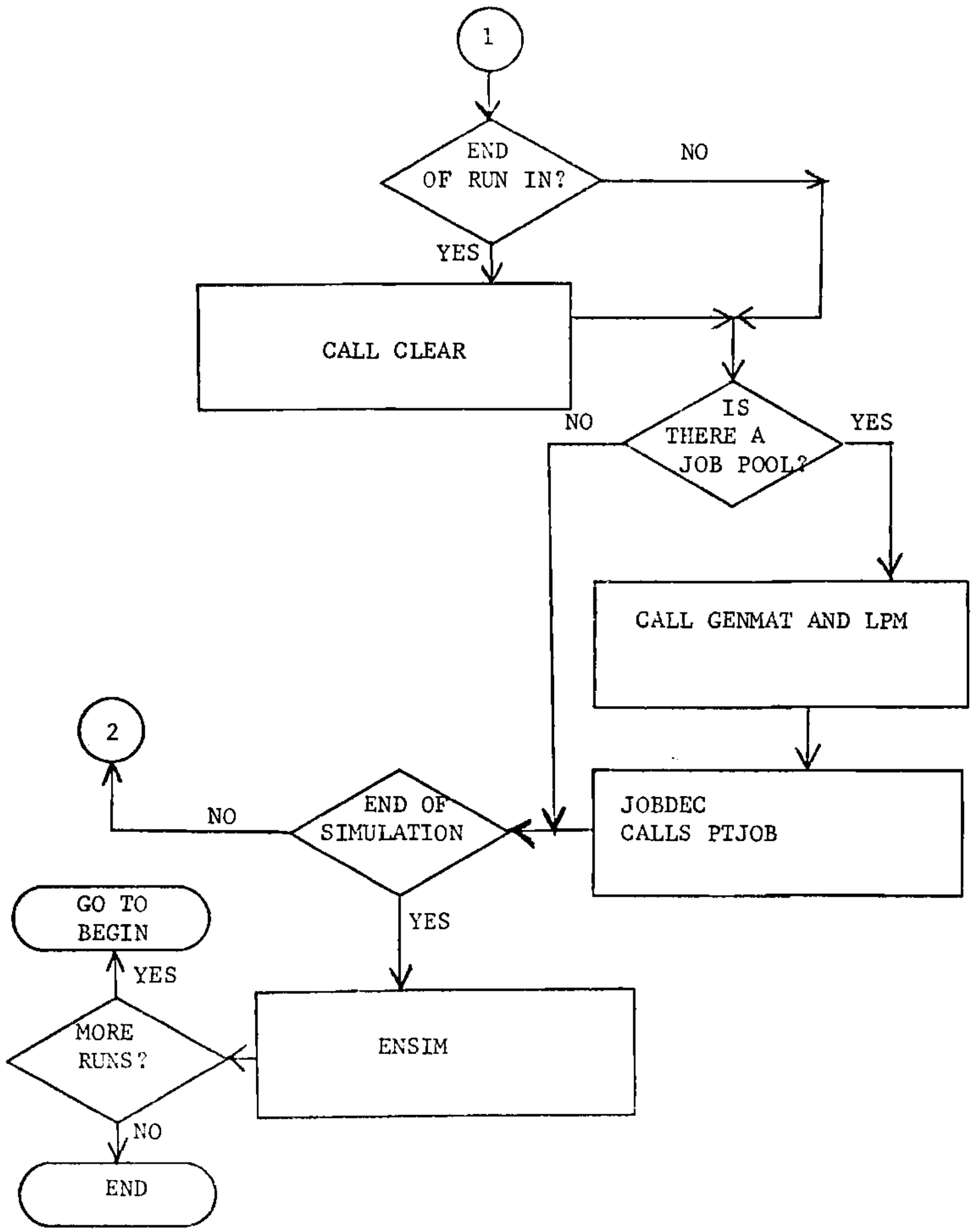

Figure 6. Concluded

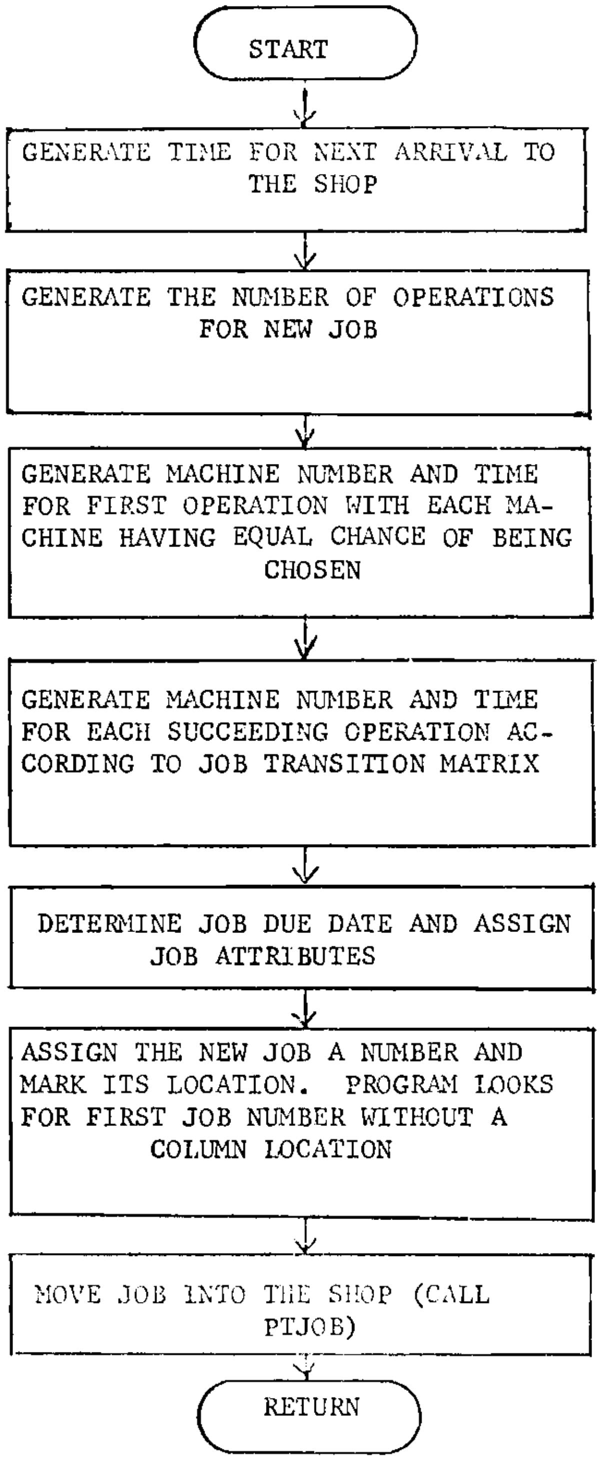

Figure 7. Subroutine ARIVL

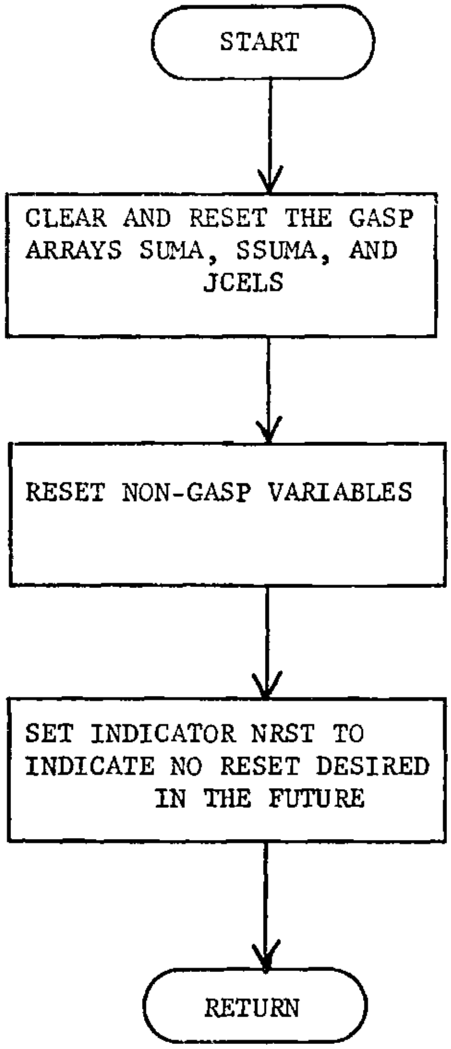

Figure 8. Subroutine CLEAR

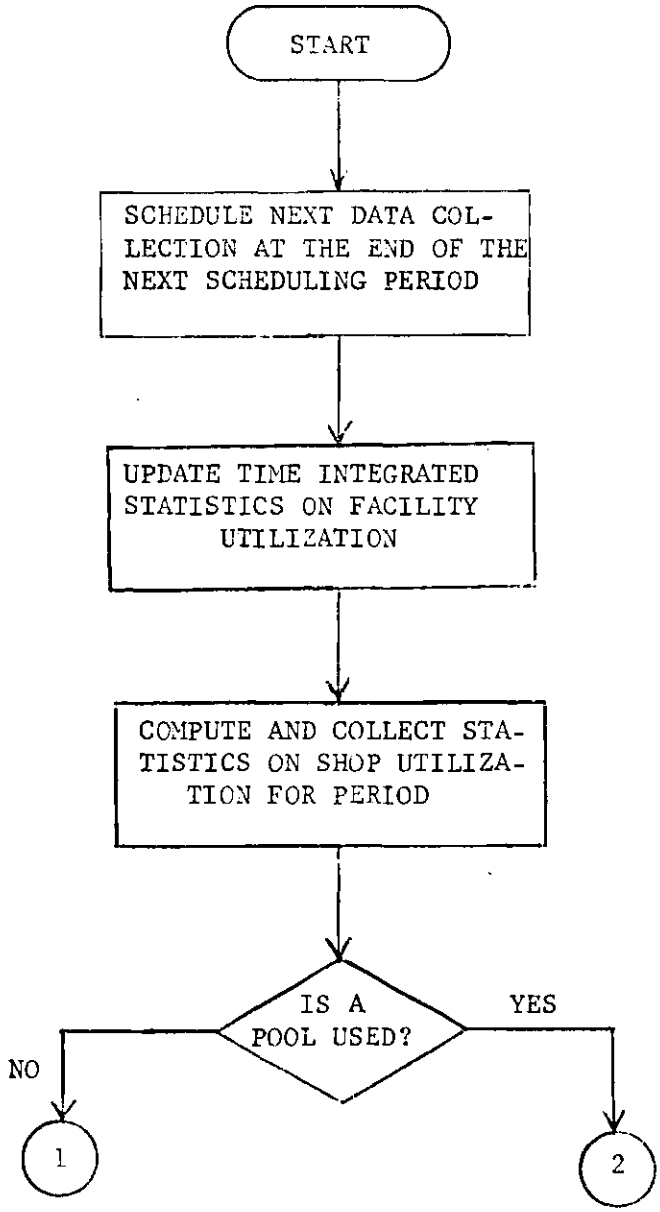

Figure 9. Subroutine COLL

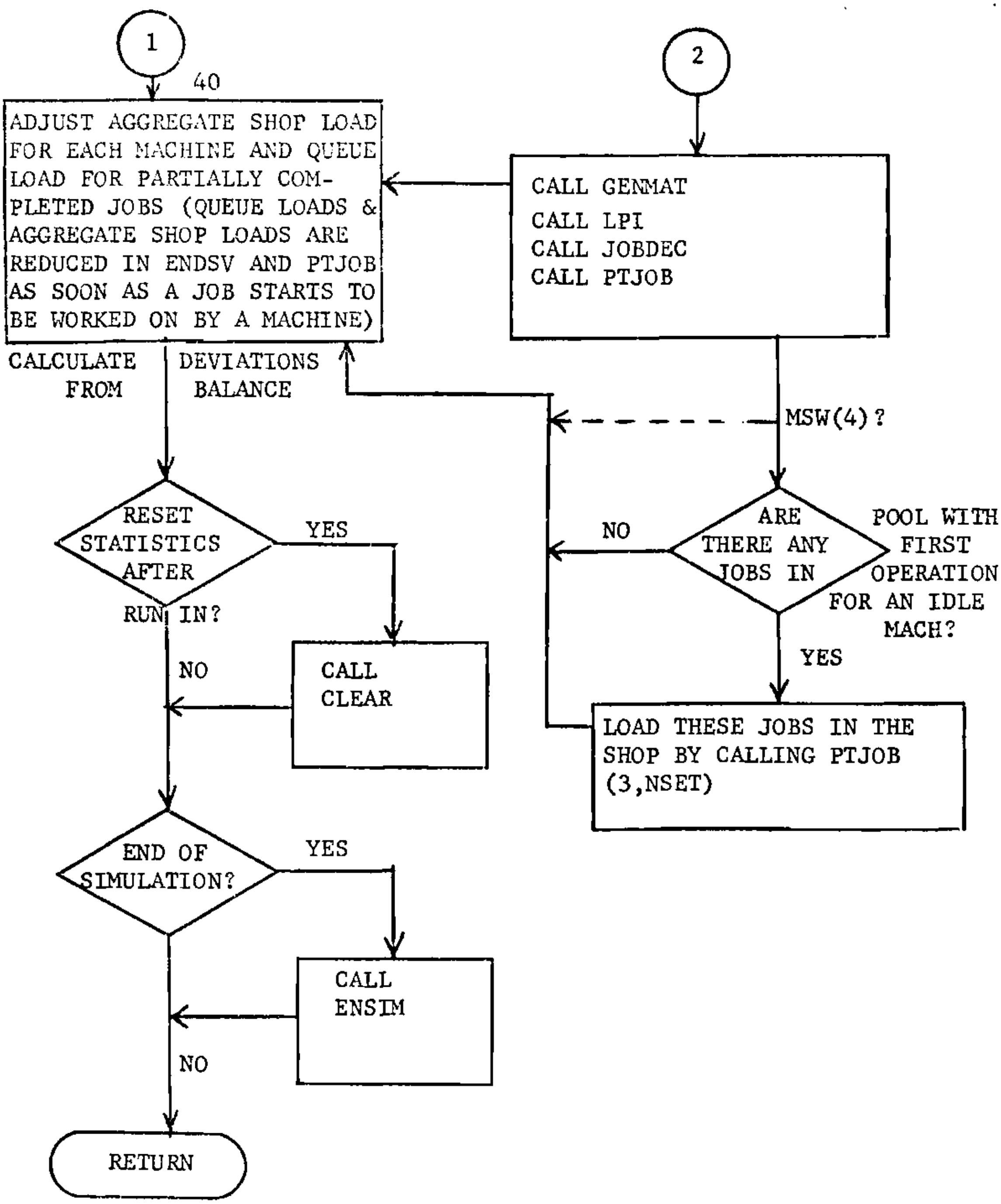

Figure 9. Concluded

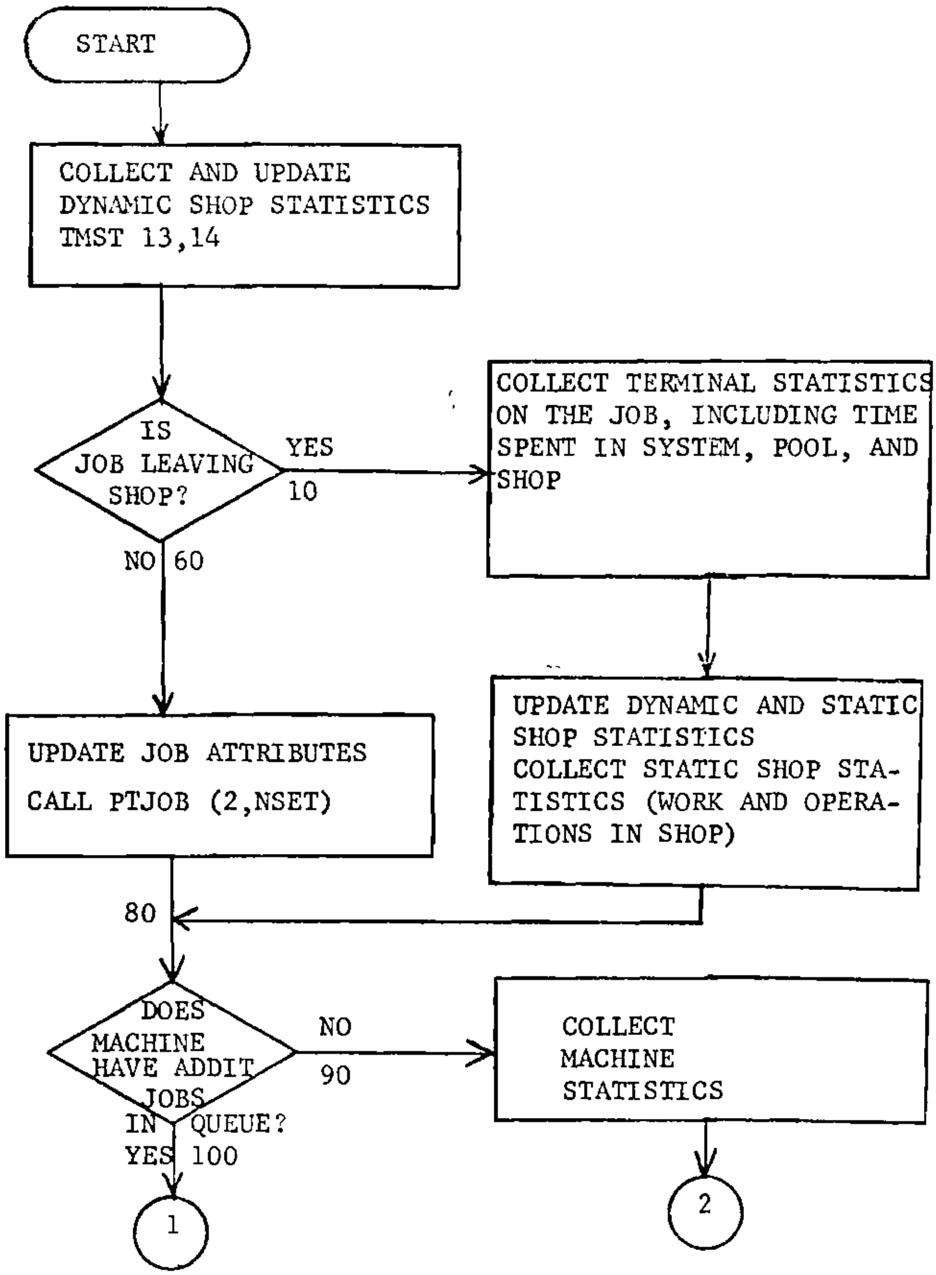

Figure 10. Subroutine ENDSV

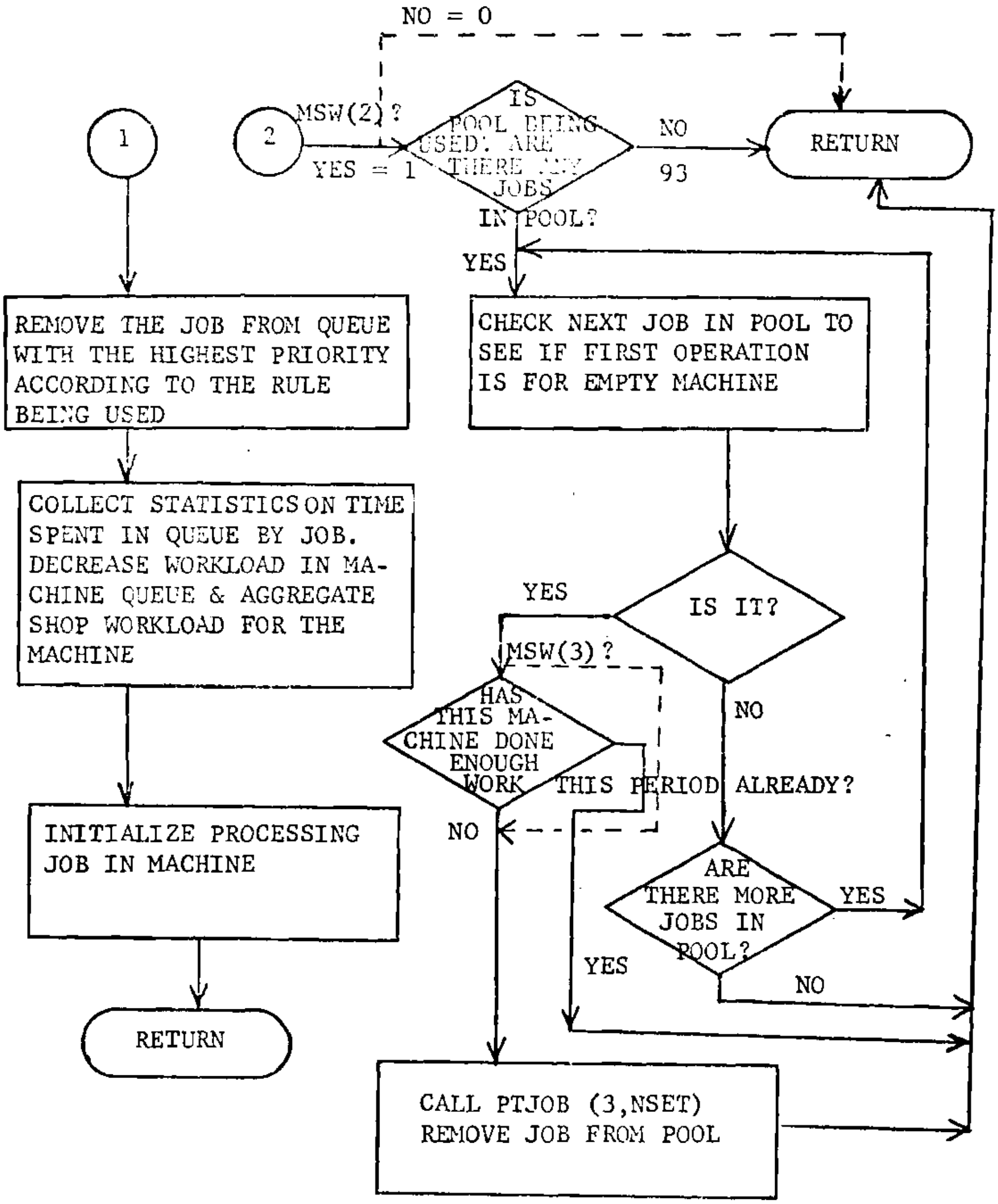

Figure 10. Continued

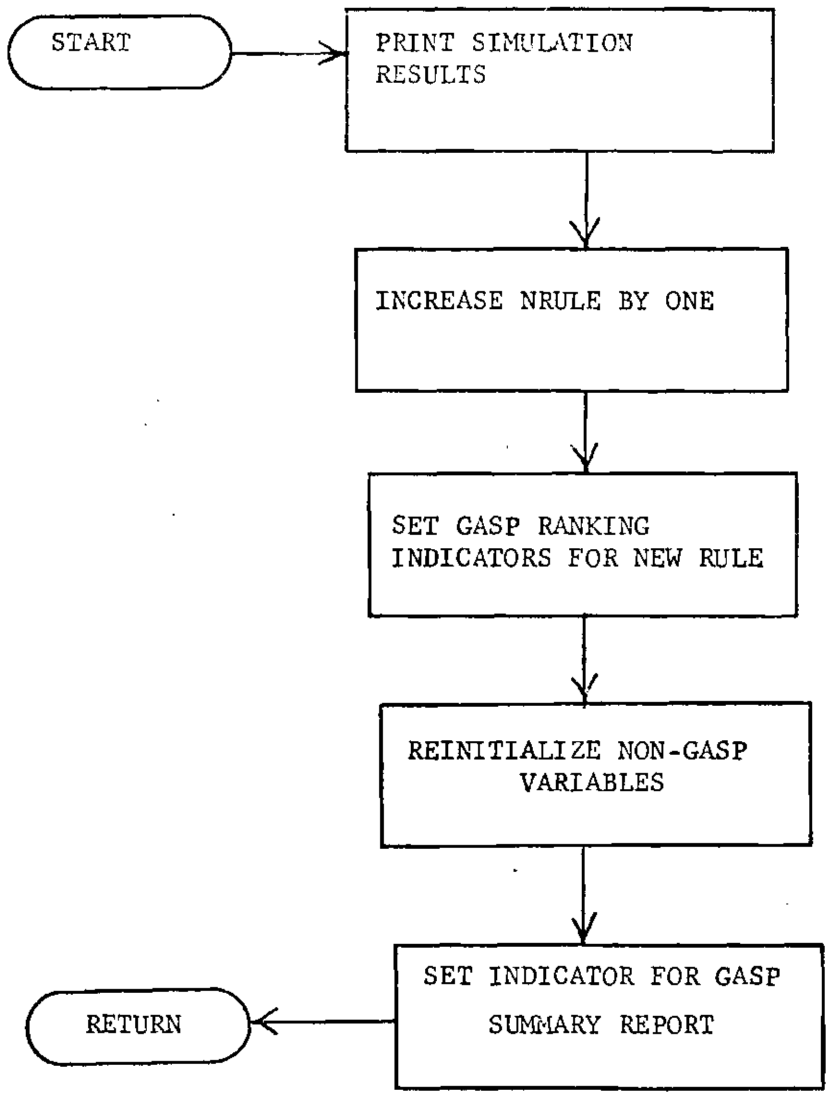

Figure 11. Subroutine ENSIM

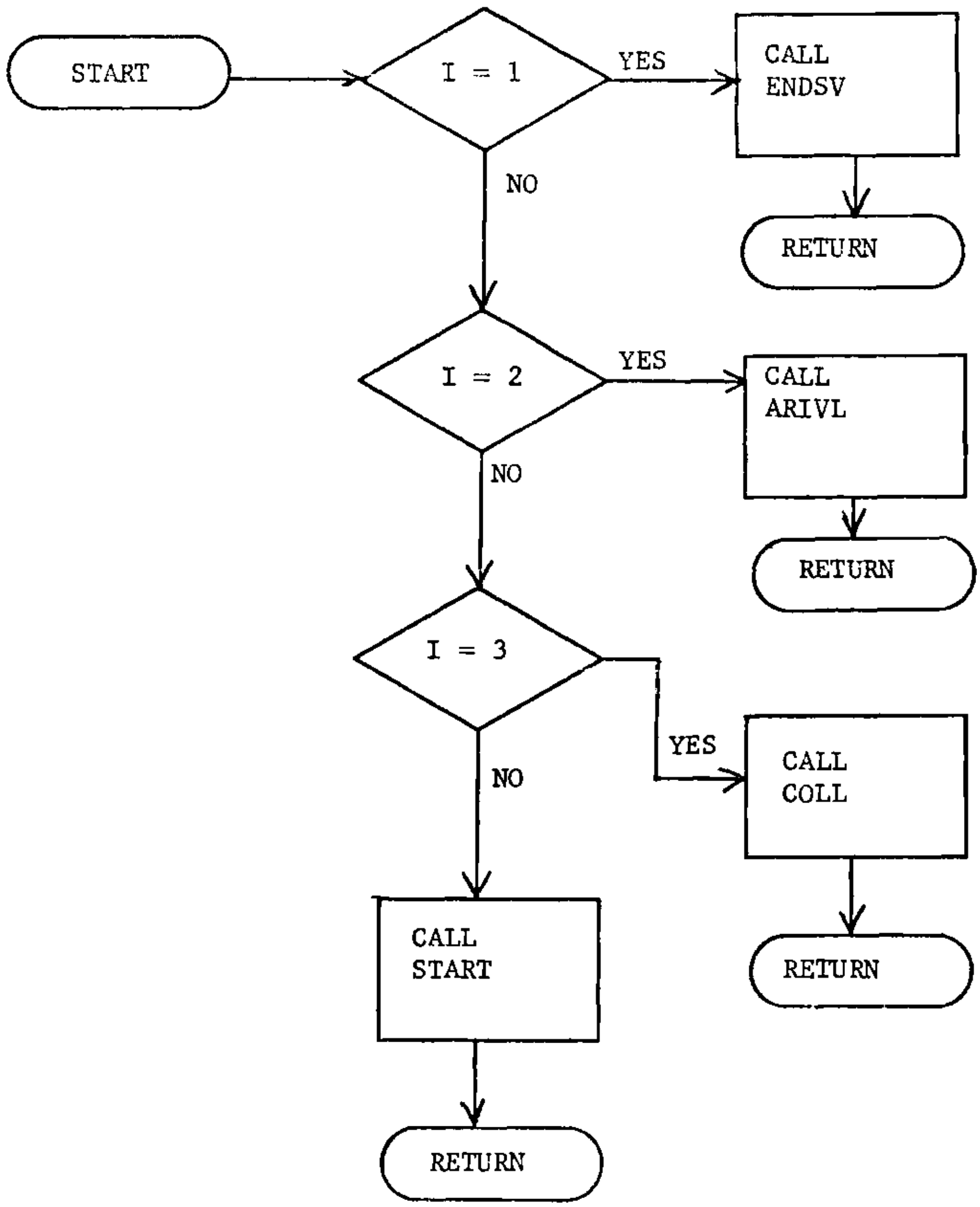

Figure 12. Subroutine EVNTS

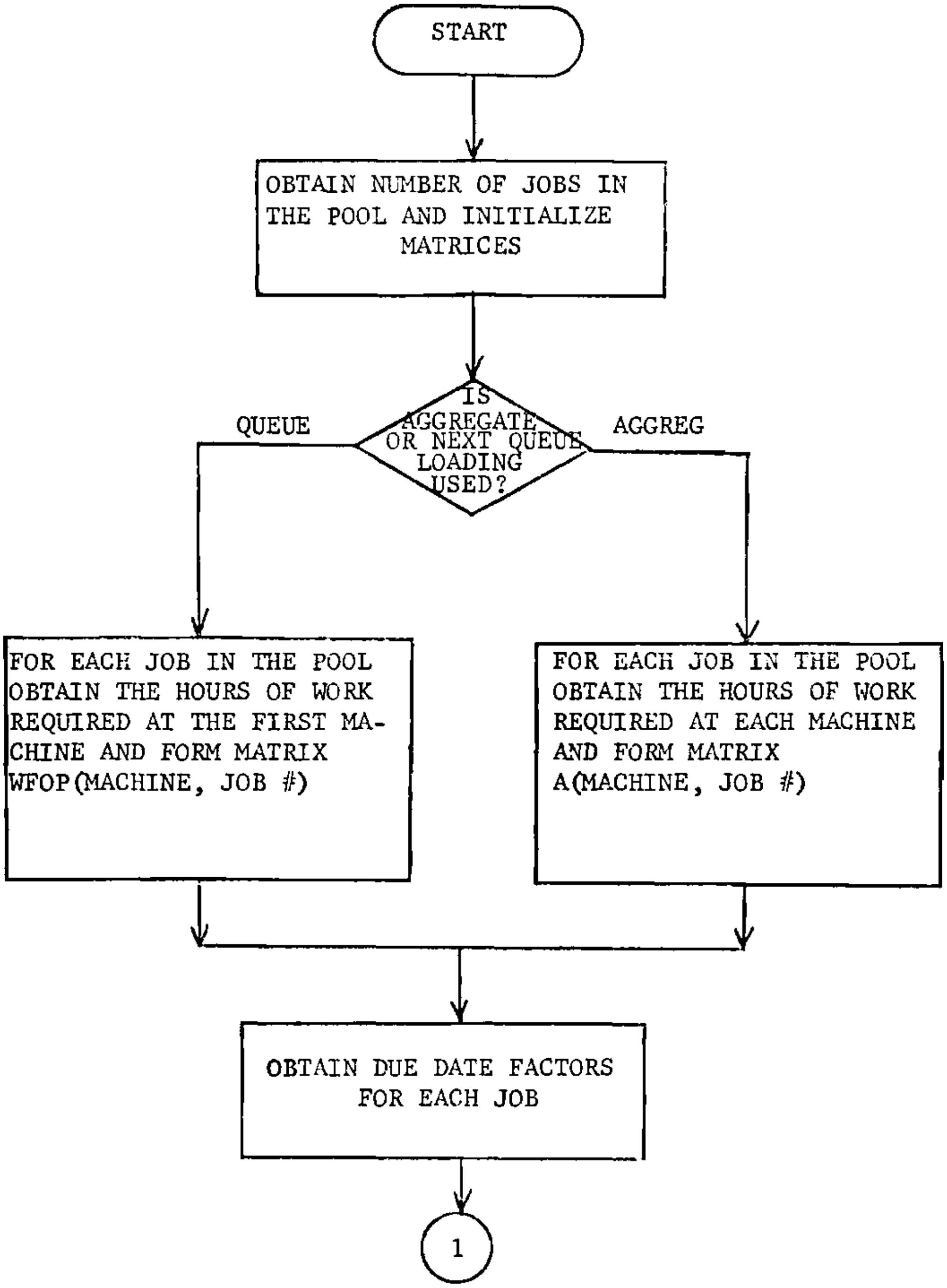

Figure 13. Subroutine GENMAT

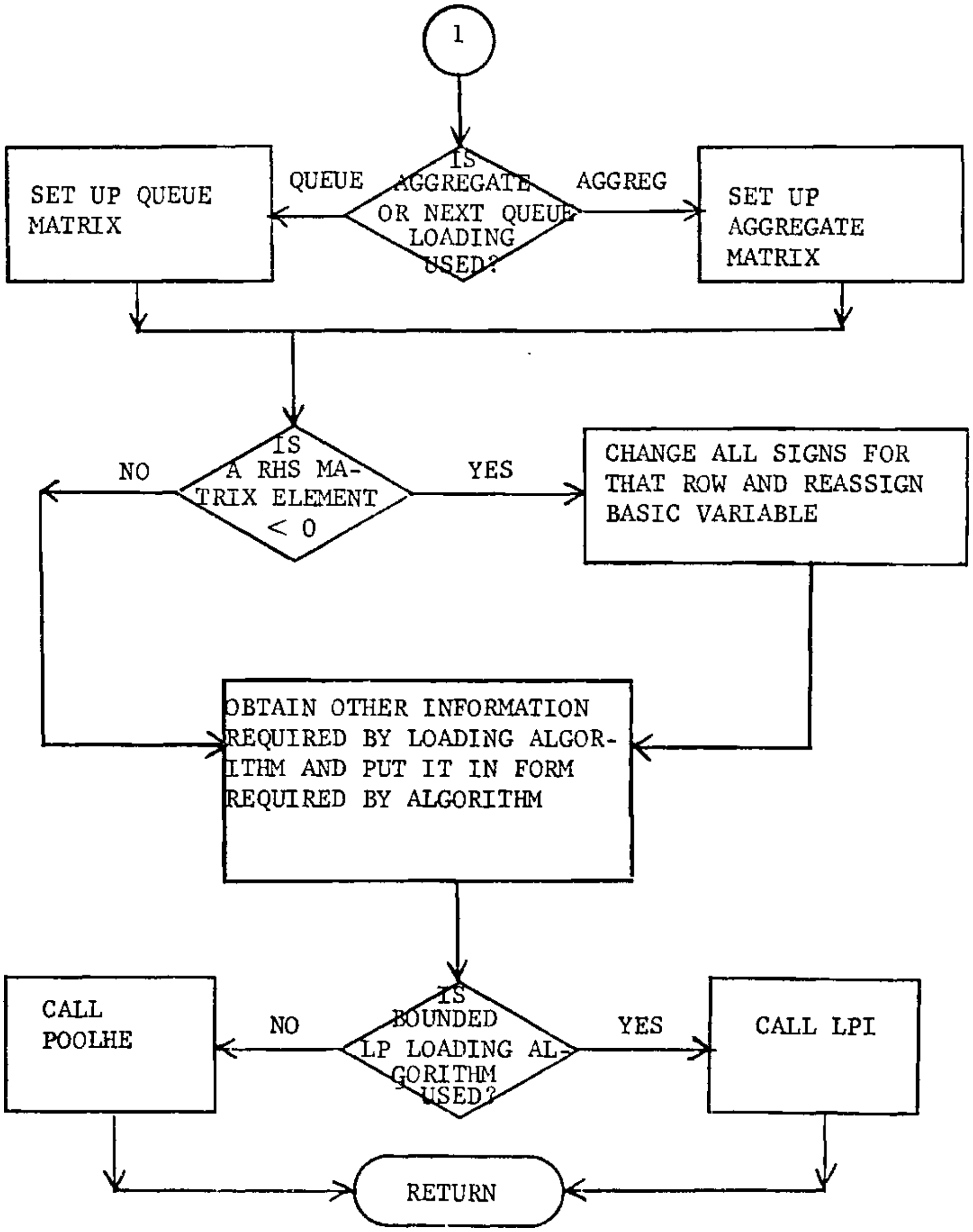

Figure 13. Concluded

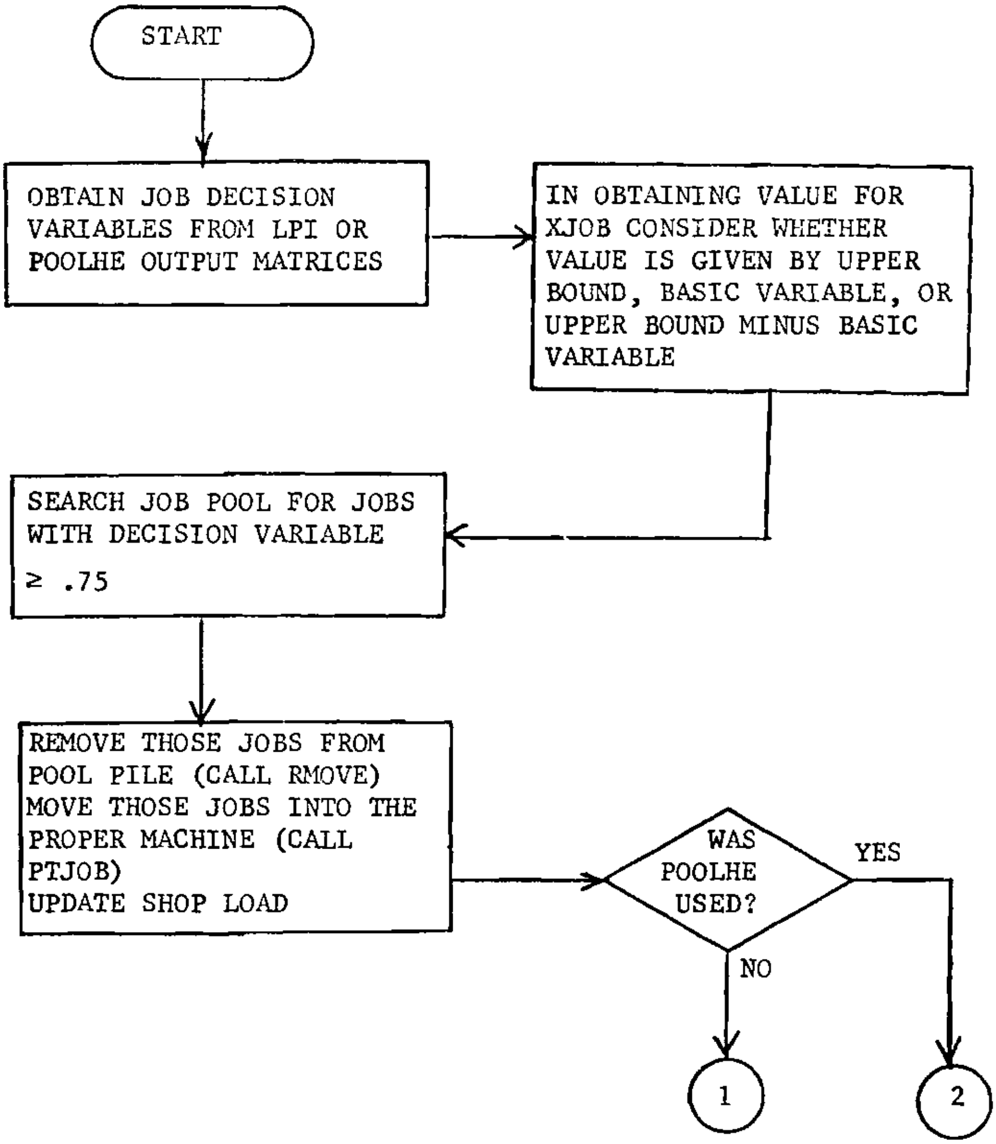

Figure 14. Subroutine JOBDEC

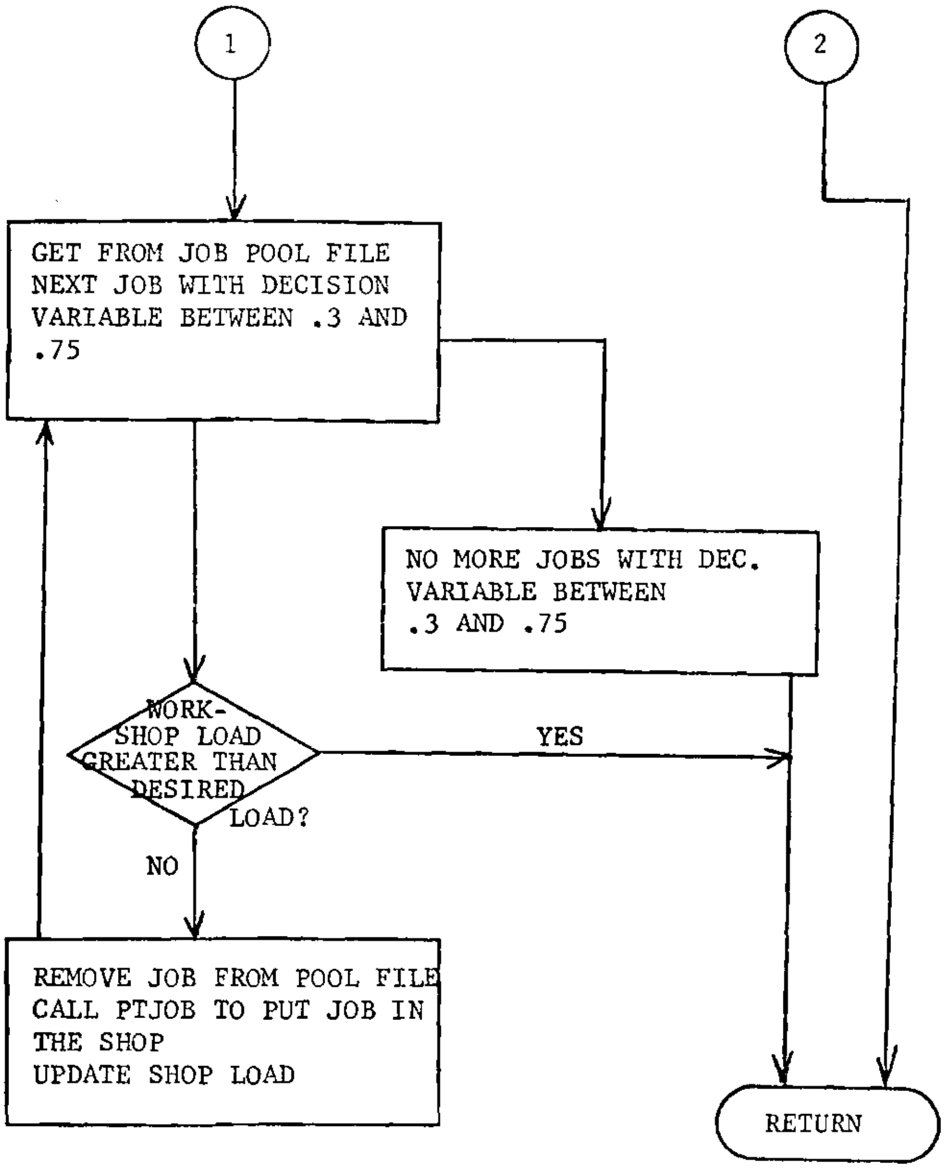

Figure 14. Continued

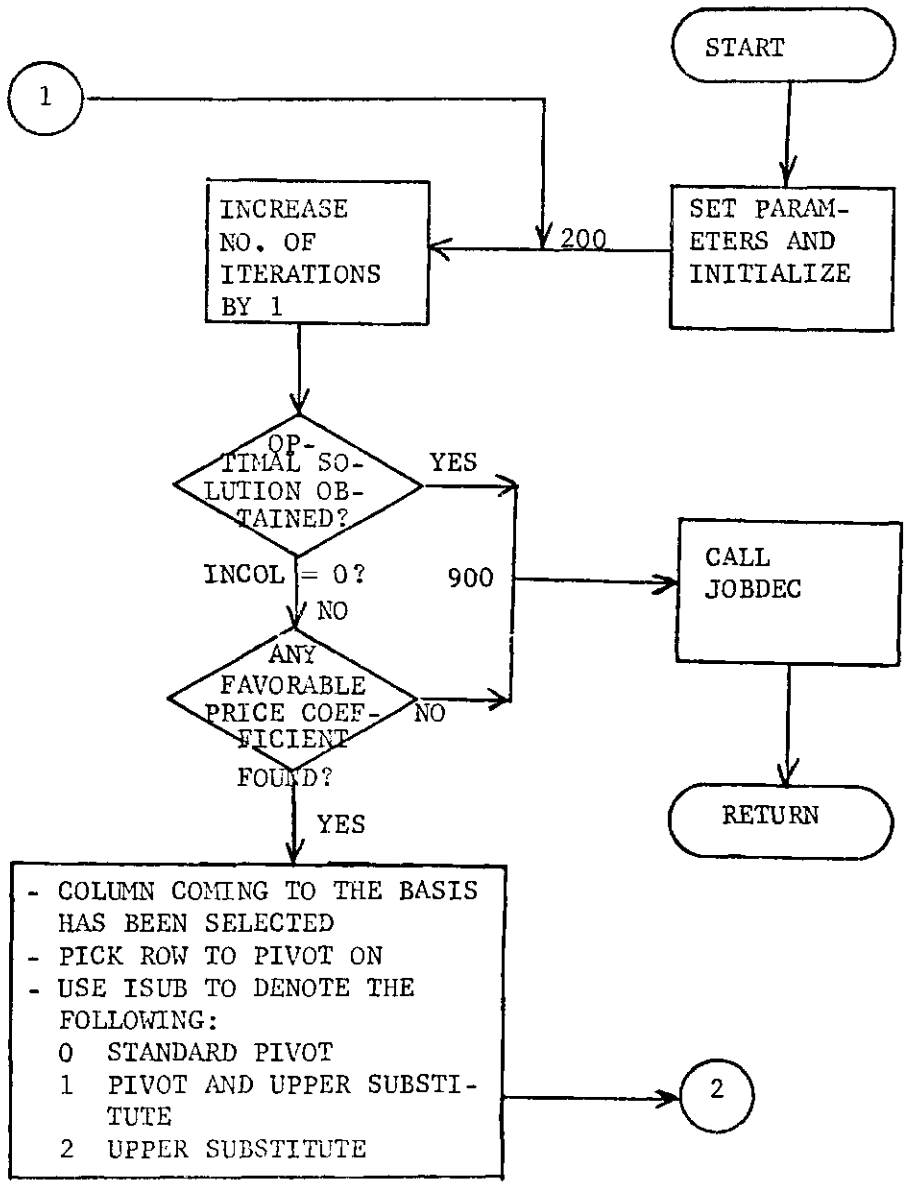

Figure 15. Subroutine LPI

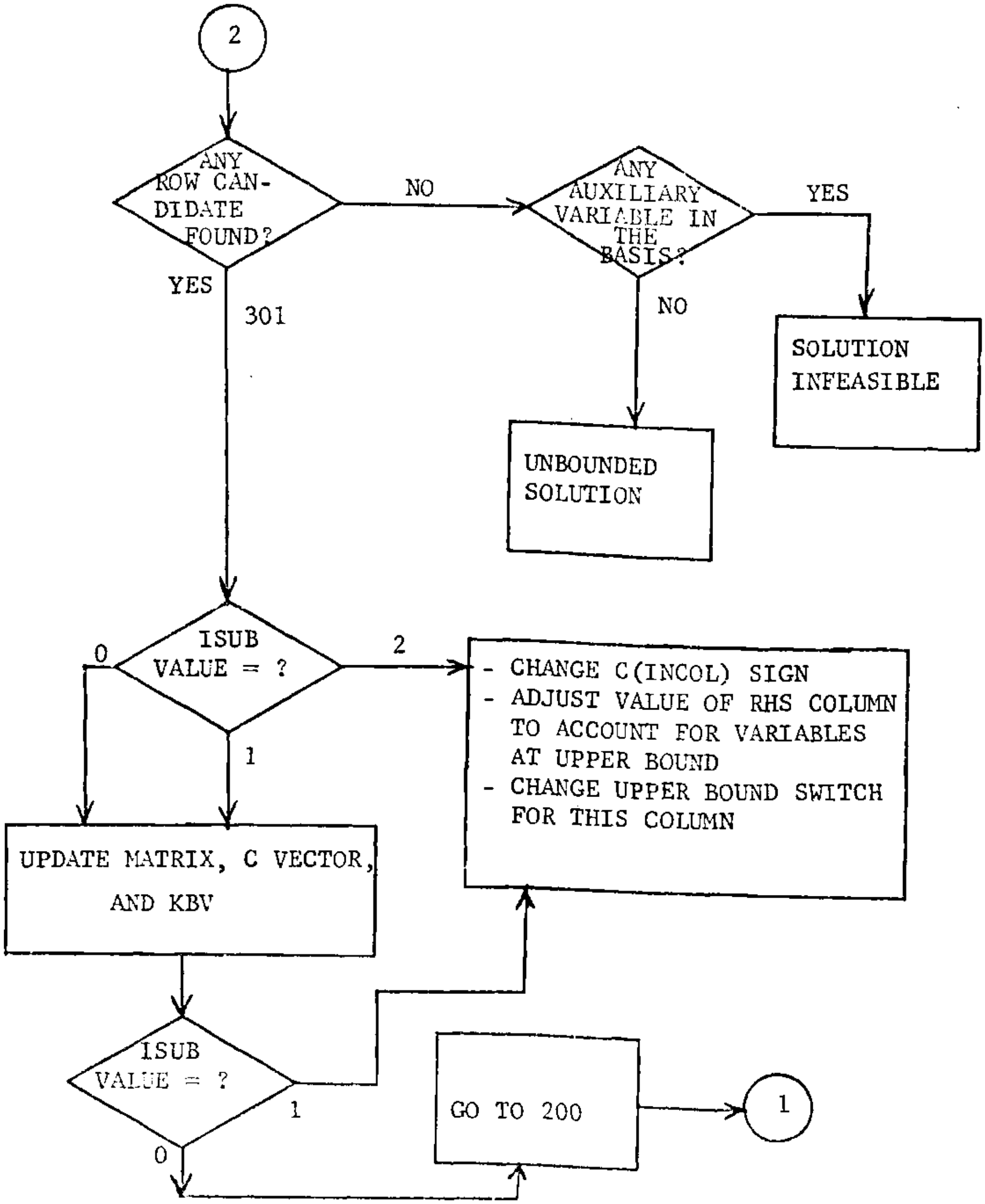

Figure 15. Concluded

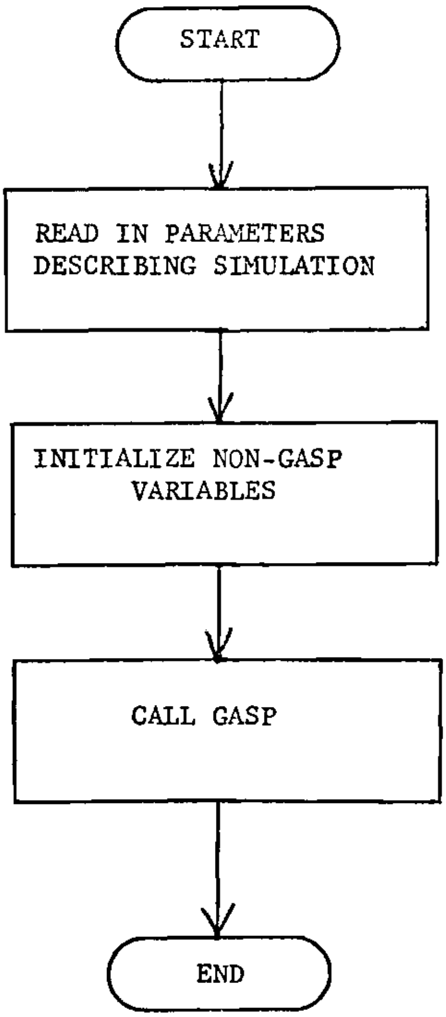

Figure 16. MAIN Program

PUT JOBS IN QUEUE UNTIL QUEUE QUANTITIES ADD UP TO DESIRED AMOUNTS. THEN DEPENDING ON FEATURE SELECTED, PUT MORE JOBS IN THE SHOP UNTIL THE TOTAL DESIRED LOAD FOR THE SHOP IS REACHED.

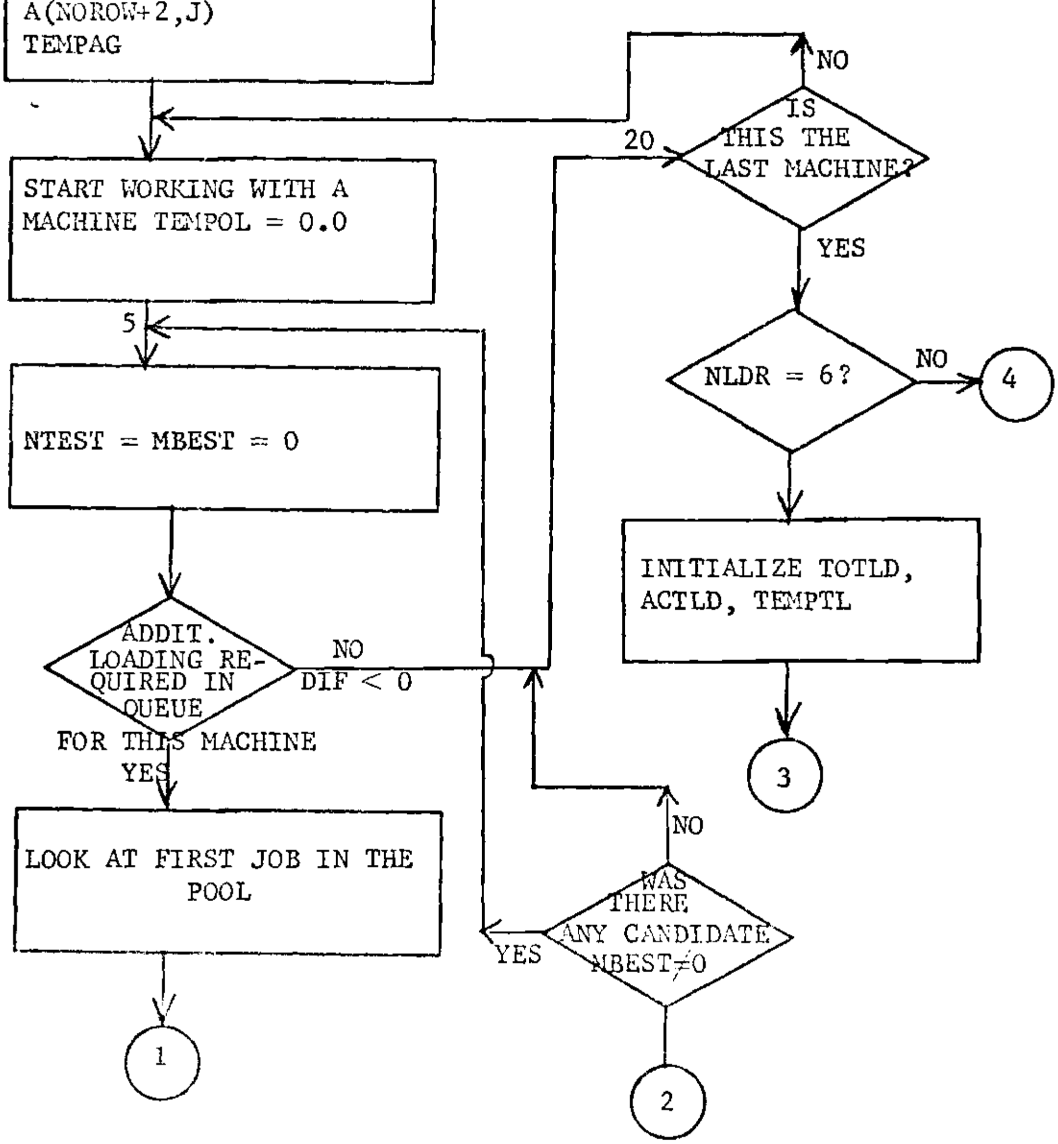

Figure 17. Subroutine POOLHE

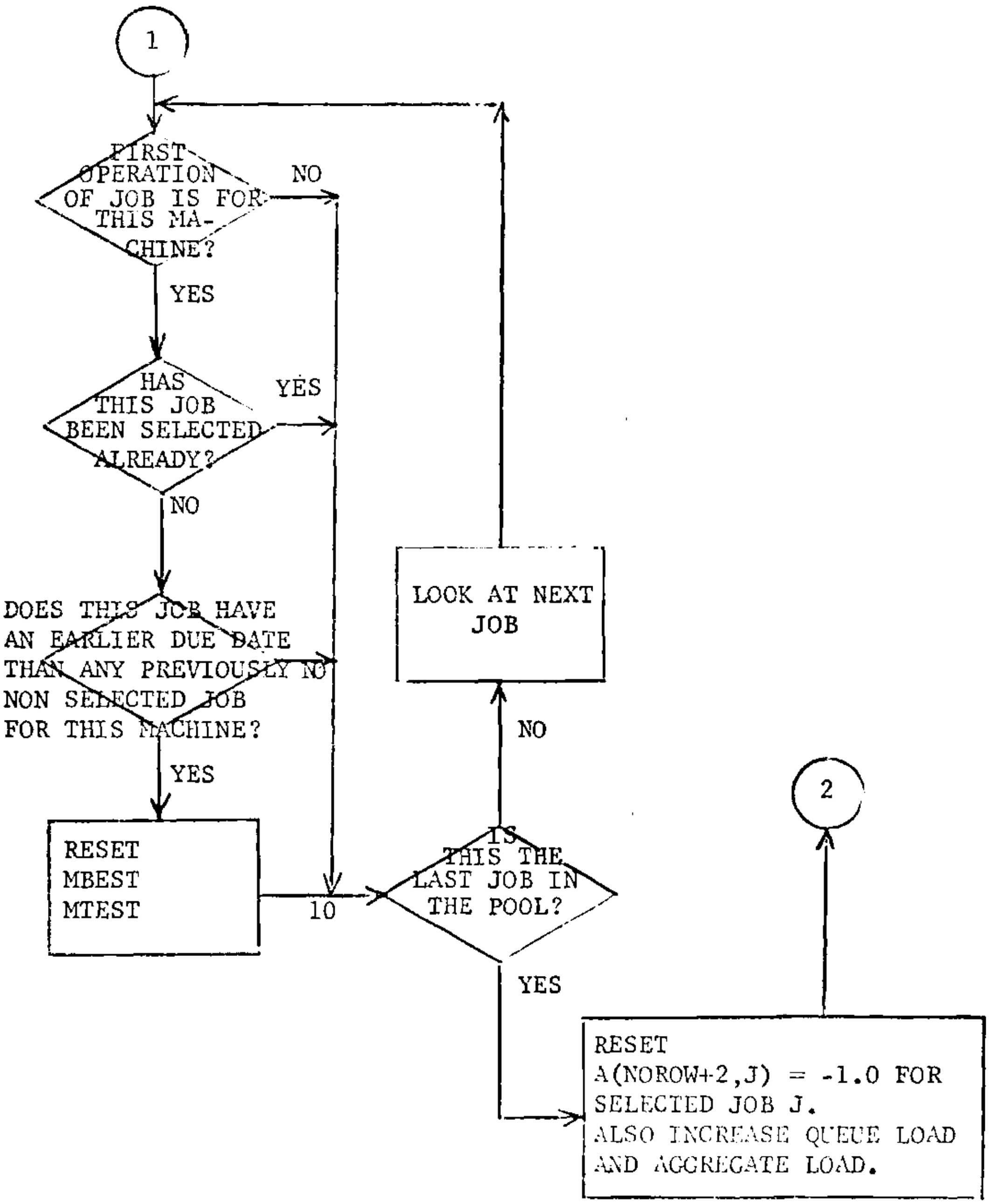

Figure 17. Continued

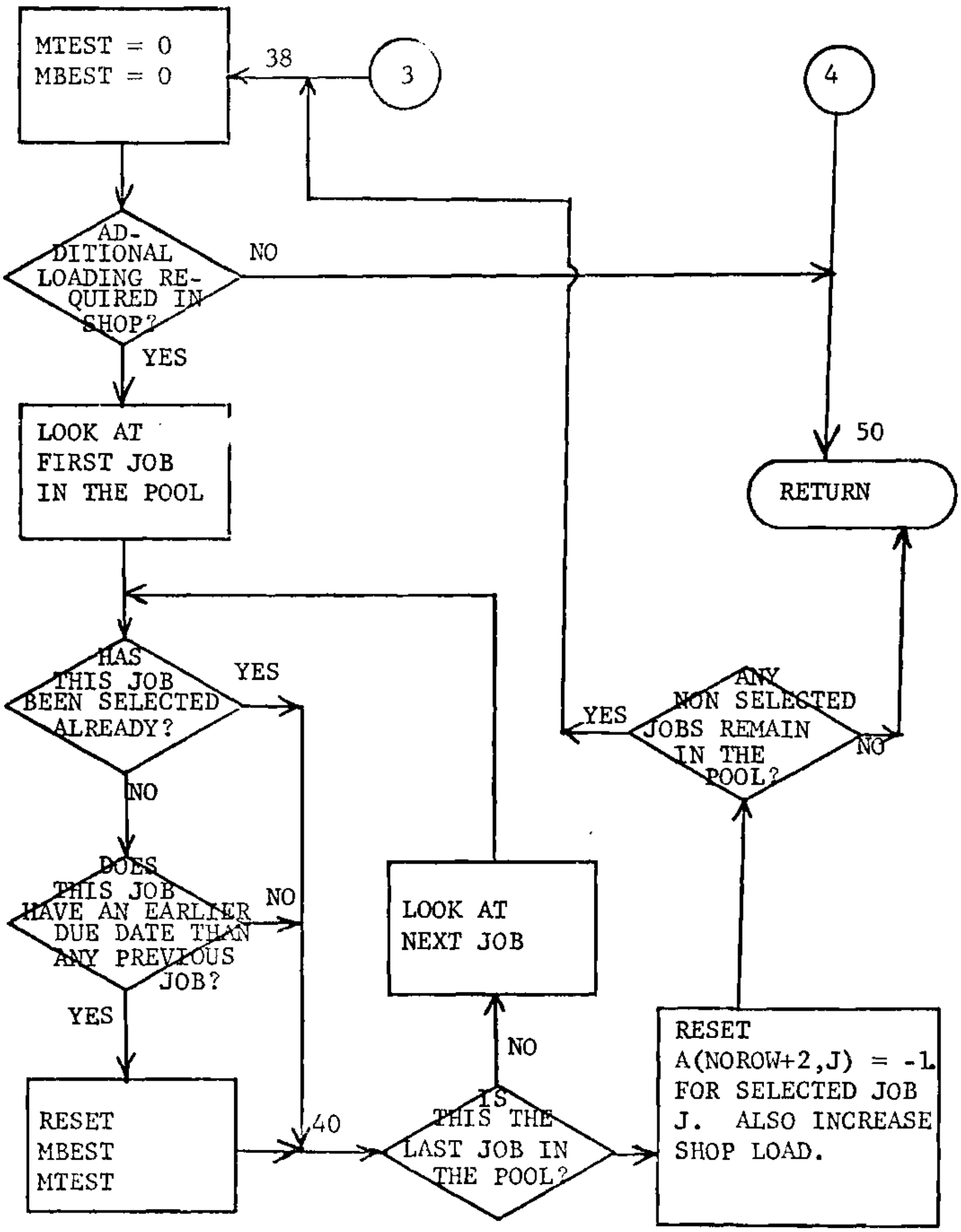

Figure 17. Concluded

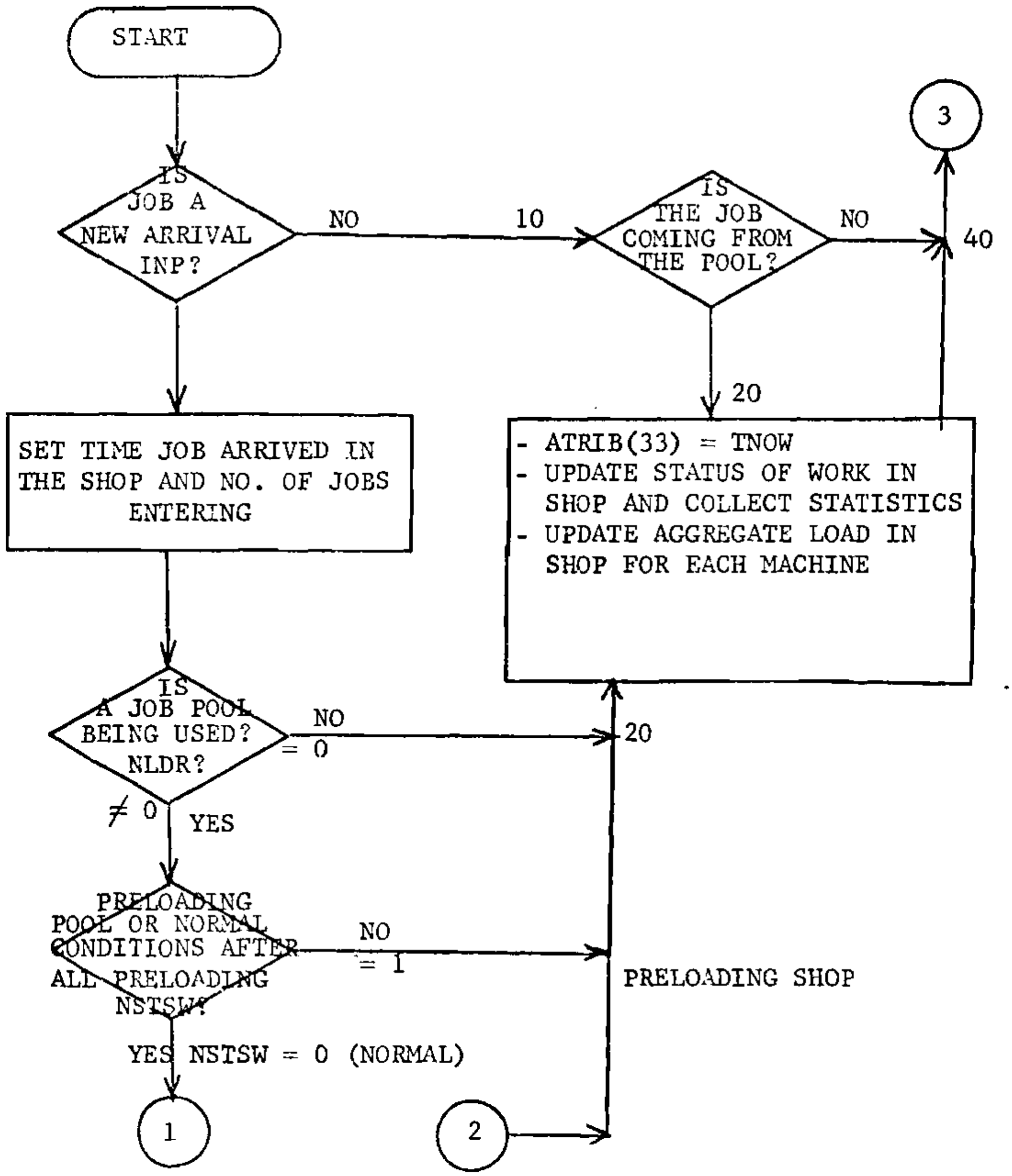

Figure 18. Subroutine PTJOB

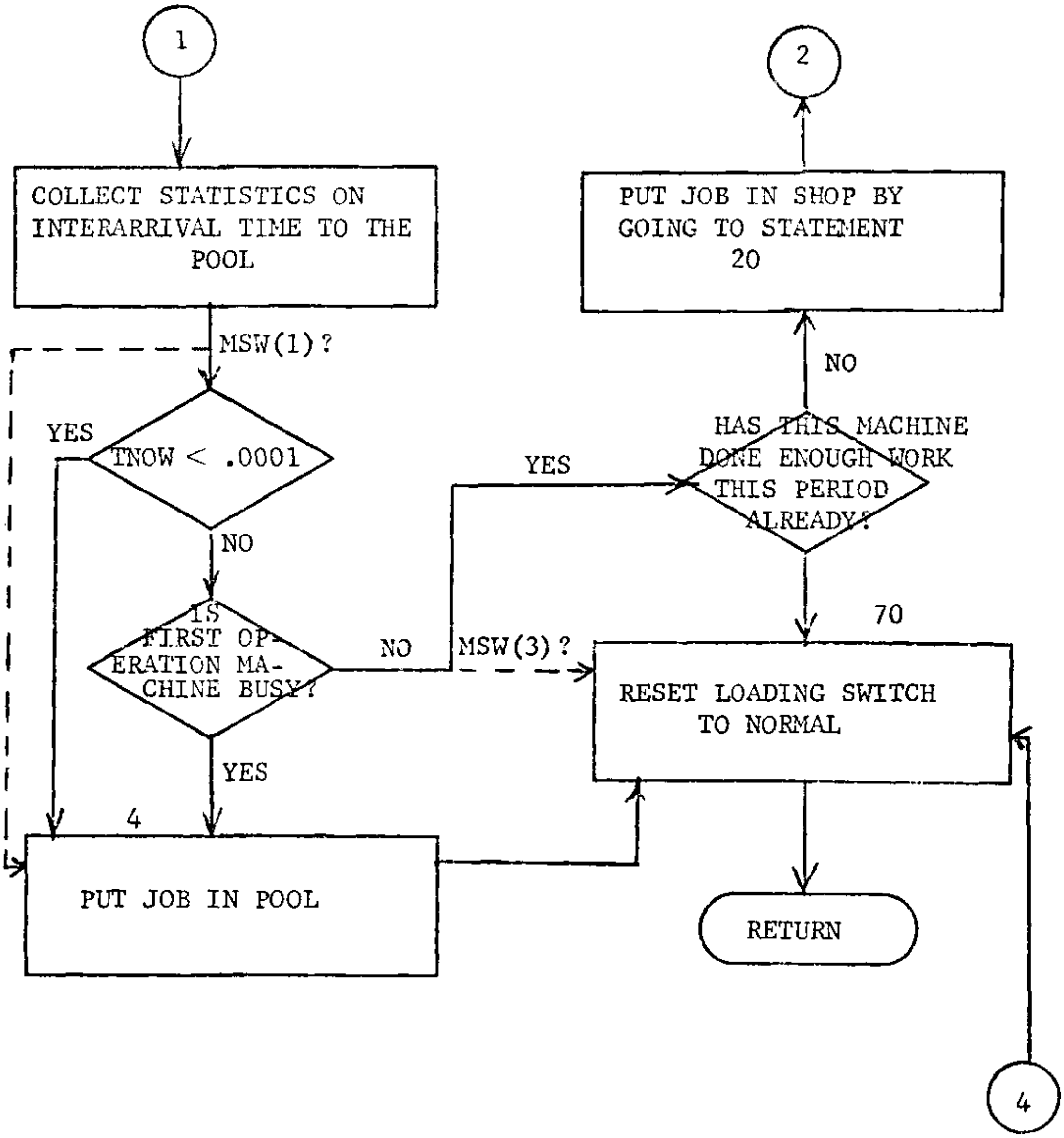

Figure 13. Continued

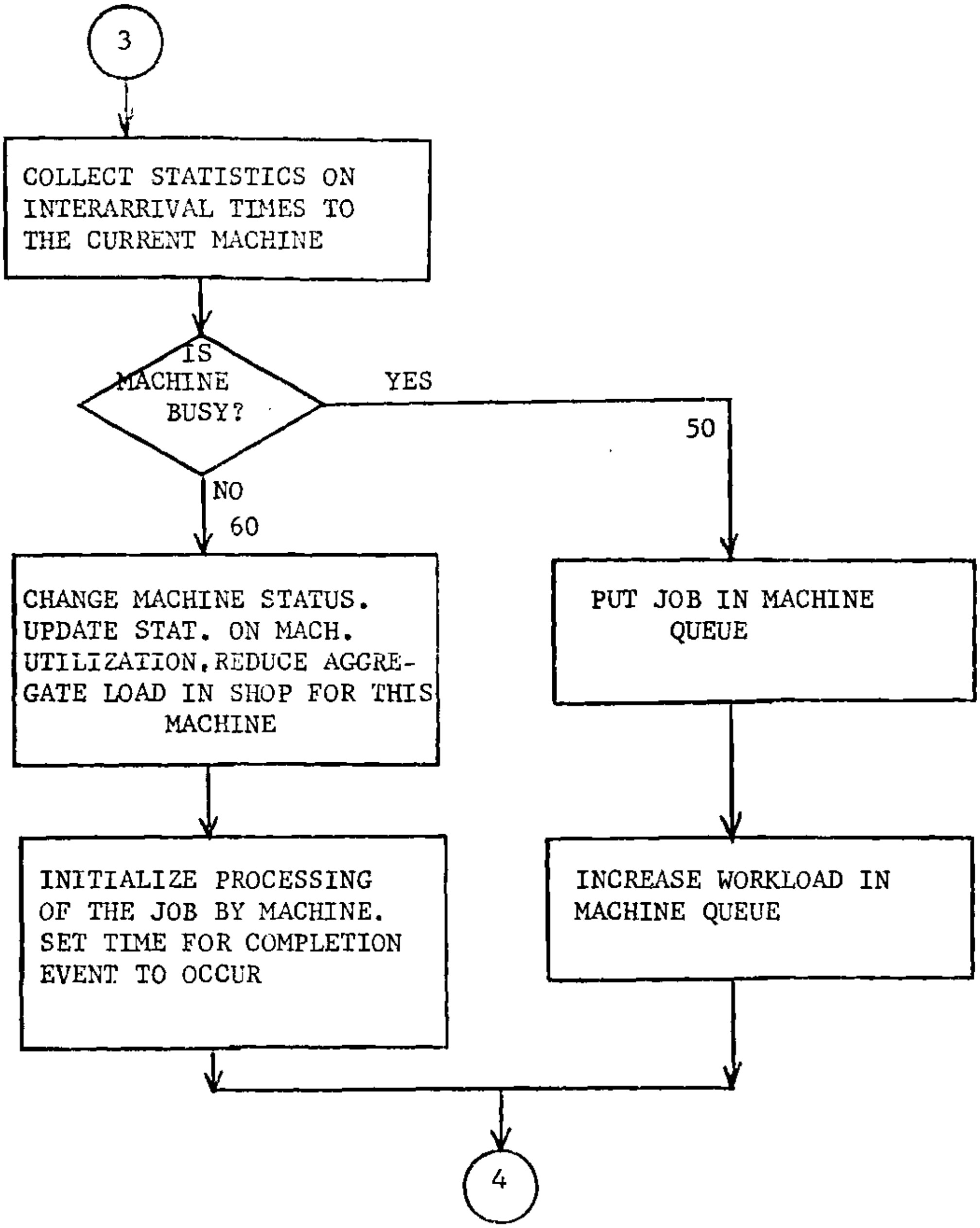

Figure 18. Concluded

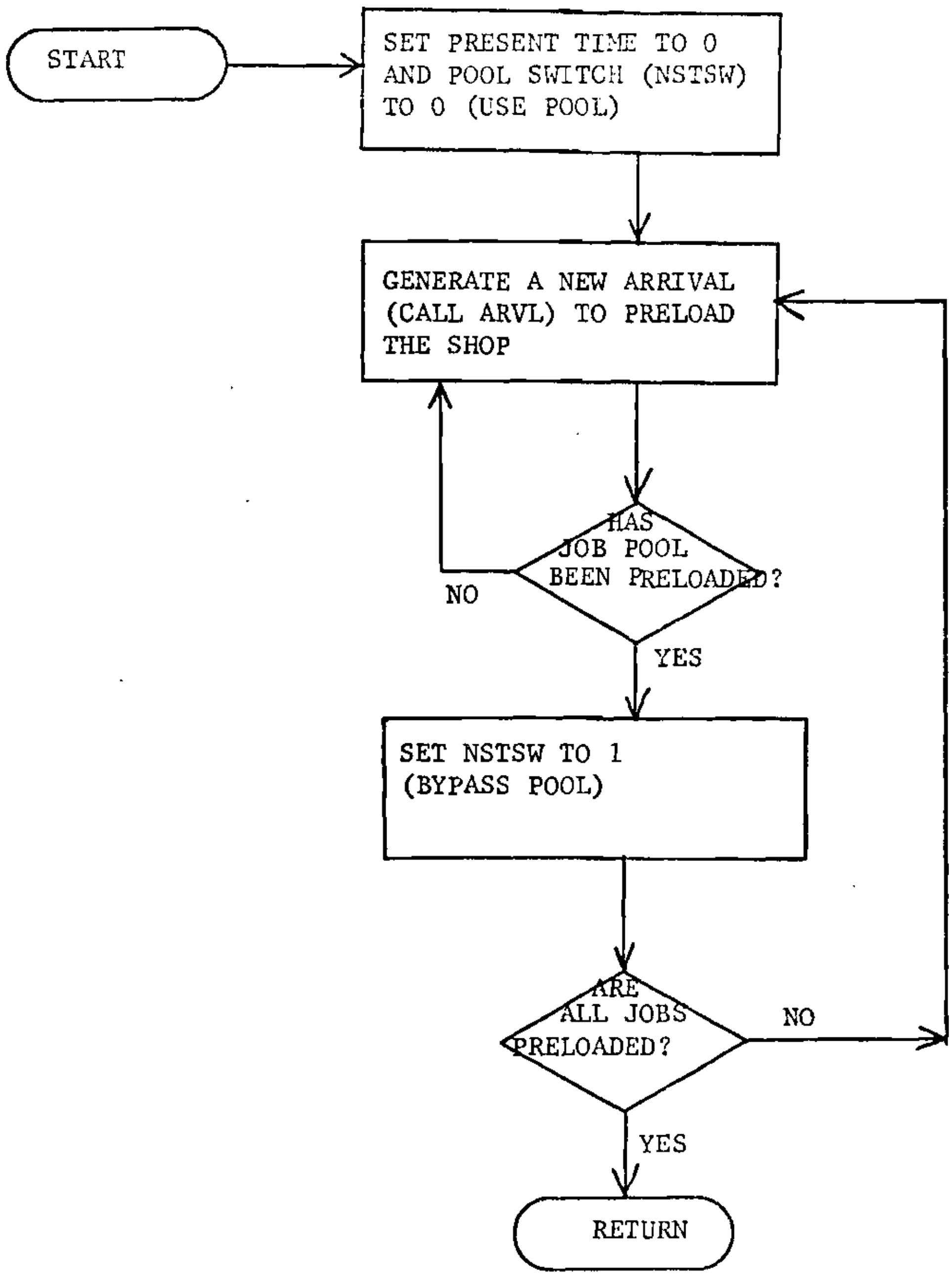

Figure 19. Subroutine START

APPENDIX B

FORTRAN IV LISTING OF SIMULATION PROGRAMS

```
-FOR,IS SHOPLOADING.ARIVL
      SUBROUTINE ARIVL (NSET)
C
C     ***  SUBROUTINE CALLED WHEN A NEW ARRIVAL COMES
C     ***  INTO THE SHOP
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,NOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON MLEN,NTPDS,MTOTPD,NN,XISYS,XMISY,INITF,
     1ITYPE,NNEXT,NLN,NLV,NHELD,NB(10),NC(10),X(10,10),
     2          BUS(10),NRSET,NRULE,NLOW,NRST,NENDS,MOL,NAL,
     3...(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDUE,CAP(10),DESL(10),
     1OGL(10),DESLF,OMLF,NLOAD(10),XOPS,XMNS,TIMEF(10),
     2NSTOR,NLDR,NARR,SHOPLD(10)
C
C     ***  GENERATE THE TIME FOR THE NEXT ARRIVAL
C     ***  TO OCCUR
C
      R=DRAND(0.0)
      ATRIB(1)=DRAWV(R) +TNOW
      ATRIB(2)=2.
      CALL FILEM (1,NSET)
C
C     ***  SPECIAL ENTRY POINT FOR JOBS TO PRELOAD
C
C     ***  THE SHOP
C
      ENTRY ARVL (NSET)
C
C     ***  GENERATE THE NUMBER OF OF OPERATIONS FOR THE
C     ***  JOB THAT JUST CAME IN
C
      R=DRAND(0.0)
      NOP=NOP(R)
      ATRIB(3)=NOP
      ATRIB(10)=NOP
      R=DRAND(0.0)
      ATRIB(31)=0.
      ATRIB(32)=0.0
C
C     ***  CHOOSE THE FIRST OPERATION OF THE JOB ROUTING
C
```

```
      DO 10 J=1,NM
      XJ=J
      IF (R.GT.XJ/FLOAT(NM)) GO TO 10
      R=DRAND(0.0)
      M=J
      ATRIB(11)=M
      A1=TIME(R)
      CALL COLCT (A1,65,NSET)
      ATRIB(12)=A1*TIMEF(M)
      WKTIM=ATRIB(12)*(8.0/CAPM(M))
      WB(M)=WB(M)+ATRIB(12)
      GO TO 20
   10 CONTINUE
C
C     ***  PICK EACH SUCCEEDING OPERATION ON THE ROUTE
C     ***  ACCORDING TO THE JOB TRANSITION MATRIX
C
   20 DO 60 I=2,NOP
   30 R=DRAND(0.0)
      DO 40 II=1,NM
      IF (R.GT.X(M,II)) GO TO 40
      MAC=II
      IF (MAC.EQ.M) GO TO 30
      GO TO 50
   40 CONTINUE
   50 M=MAC
      R=DRAND(0.0)
      ATRIB(2*I+9)=MAC
      A1=TIME(R)
      CALL COLCT (A1,65,NSET)
      ATRIB(2*I+10)=A1*TIMEF(MAC)
      WKTIM=WKTIM+A1*TIMEF(MAC)*(8.0/CAPM(MAC))
   60 WB(MAC)=WB(MAC)+ATRIB(2*I+10)
C
C     ***  SET UP OTHER ATTRIBUTE VALUES
C
      ATRIB(32)=0.0
      MNEXT=ATRIB(11)+.000001
      ATRIB(9)=0.0
      NNN=10+2*NOP
      DO 70 I=12,NNN,2
   70 ATRIB(9)=ATRIB(9)+ATRIB(I)
      ATRIB(3)=TNOW
      WORK=ATRIB(9)
      ATRIB(4)=DUED(WKTIM,IDUE)+TNOW
      ATRIB(6)=ATRIB(4)-WKTIM
      IF (NRULE.LE.3) ATRIB(6)=ATRIB(9)
      ATRIB(7)=ATRIB(6)/ATRIB(5)
C
```

```
C      ***  GIVE THE NEW JOB A NUMBER AND MARK ITS LOACTION
C      ***  PROG LOOKS FOR FIRST JOB NO. WITH NO COLUMN LOC.
C
       DO 80 I=1,200
       L=LOC(I)
       IF (L.EQ.0) GO TO 90
   80  CONTINUE
       CALL ERROR (201,NSET)
   90  ATRIB(30)=I
       IF(I.GT.MAX) MAX=I
C
C      ***  MOVE THE JOB INTO THE SHOP
C
       CALL PTJOB (1,NSET)
       RETURN
       END
```

```
-FOR,IS SHOPLOADING.CLEAR
       SUBROUTINE CLEAR (NSET)
C
C      ***  CLEAR THE STATISTICAL STORAGE AREAS
C      ***  AFTER NRSET PERIODS
C
       DIMENSION NSET(35,1)
       COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,NX,MXC,NCLCT,
      1NHIST,NOQ,NCRPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
      2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
      3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
      4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
      5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
      6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
       COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,X..KSY,IDUE,
      1ITYPE,NNEXT,NEN,NLV,MHELD,NB(10),NBM(10),X(10,10),
      2          BUS(10),NRSET,MRULE,MNOM,MSST,NENDS,NHCL,[illegible].
      3A..X(10),SEED,ARATE,LOC(200),MAX,AP(11)
       COMMON NPREL,MPREP,NDESL,NDML,CAPM(10),DESL(10),
      1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XMKS,TIMEF(10),
      2NSTS.,NLDR,NARR,SHOPLD(10)
       IF (NCLCT) 40,40,10
   10  DO 30 I=1,NCLCT
       DO 20 J=1,3
   20  SUMA(I,J)=0.0
       SUMA(I,4)=1.0E20
   30  SUMA(I,5)=-1.0E20
   40  IF (NSTAT) 80,80,50
   50  DO 70 I=1,NSTAT
```

```
      SSUMA(I,1)=TNOW
      DO 60 J=2,3
   60 SSUMA(I,J)=0.0
      SSUMA(I,4)=1.0E20
   70 SSUMA(I,5)=-1.0E20
   80 IF (NHIST) 110,110,90
   90 DO 100 K=1,NHIST
      DO 100 L=1,MXC
  100 JCELS(K,L)=0
  110 DO 120 I=1,NM
      WB(I)=0.0
      WWW(I)=0.0
  120 ABM(I)=0.0
      DO 130 K=1,NOQ
      VNQ(K)=0.0
      ENQ(K)=0.0
  130 MAXNQ(K)=NQ(K)
      NRST=9999999
      TBEG=TNOW
      NEN=0
      NLV=0
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.COLL
      SUBROUTINE COLL (NSET)
C
C     ***  EVENT SUBROUTINE TO COLLECT STATISTICS AT
C     ***  THE END OF EVERY SCHEDULING PERIOD AND TO
C     ***  CALL THE LOADING ROUTINE
C
      DIMENSION NSET(35,1),SCPLD2(10),CLOAD2(10)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPOS,NTOTPO,NM,XICYS,YMNSY,IDUE,
     1ITYPE,NNEXT,NEN,NLV,NHELD,WB(10),ABM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNO,NRST,NENDS,NHOL,NL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AM(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DGL(10),DESLF,DMLF,CLOAD(10),XCPS,XMKS,TIMEF(10),
     2NSTSW,NLDP,NARR,SHOPLD(10)
```

```
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
C
C     ***  SCHEDULE THE NEXT DATA COLLECTION POINT
C
      ATRIB(2)=3.0
      ATRIB(1)=TNOW+PLEN
      CALL FILEM (1,NSET)
      NTPDS=NTPDS+1
      ISCALE=SCALE+.000001
      NTP=NTPDS-1
      TS=0.0
      TOT=0.0
C
C     ***  UPDATE TIME INTEGRATED STATISTICS ON MACHINES
C     ***  AND COMPUTE STATISTICS ON FACILITY UTILIZATION
C     ***  DURIN THE PERIOD
C
      AP=0.0
      BP=0.0
      DO 10 I=1,NM
      CALL TMST (BUS(I),TNOW,I,NSET)
      UT=SSUMA(I,3)/PLEN*100.0
      WB(I)=WB(I)/PLEN*100.0
      TS=TS+SSUMA(I,3)
      WBM(I)=WBM(I)+SSUMA(I,3)*SSUMA(I,3)
      TOT=TOT+WB(I)
      WWW(I)=WWW(I)+WB(I)
      CALL COLCT (UT,I,NSET)
      AP=AP+SSUMA(I,3)
      BP=BP+SSUMA(I,3)*SSUMA(I,3)
      WB(I)=0.0
   10 SSUMA(I,3)=0.0
      AP=AP/FLOAT(NM)
      BP=BP/FLOAT(NM)
      BP=BP-AP**2
      CALL COLCT(BP,70,NSET)
      CP=0
      DP=0
      NM1=NM+1
      DO 12 I=2,NM1
      I1=I-1
      XNQ=NQ(I)
      XC=(ENQ(I)+XNQ*(TNOW-QTIME(I)))
      IF (TNOW.LE.0.001) XAVL(I1)=0.0
      AVG=(XC-XAVL(I1))/PLEN
      CP=CP+AVG
      DP=DP+AVG**2
   12 XAVL(I1)=XC
```

```
      CP=CP/FLOAT(NM)
      DP=DP/FLOAT(NM)
      DP=DP-CP**2
      CALL COLCT(DP,71,NSET)
      ATS=TS/FLOAT(NM)
      ATOT=TOT/FLOAT(NM)
      CALL COLCT (ATS,14,NSET)
      CALL HISTO (ATS,0.5,0.5,3,NSET)
      AT=ATOT
      CALL HISTO (AT,6.0,6.0,4,NSET)
C
C     ***  CHECK IF A JOB POOL IS BEING USED
C
      R66=NQ(12)
      CALL COLCT (R66,66,NSET)
      IF (NLDR.EQ.0) GO TO 39
C
C     ***  POOL IS BEING USED. CALL SUBROUTINES TO LOAD THE
C     ***  SHOP
C
      IF (NQ(12).EQ.0) GO TO 39
      CALL GENMAT (NSET)
C
C     ***  ADJUST AGGREGATE SHOP LOAD FOR EACH MACHINE AND
C     ***  QUEUE LOAD FOR PARTIALLY COMPLETED JOBS
C
   39 R67=NQ(12)
      CALL COLCT (R67,67,NSET)
      IF (MSW(4).EQ.0) GO TO 40
      IF (NQ(12).LT.1) GO TO 40
      J=0
      N1=MFE(12)
   15 J=J+1
      NFIRST=FLOAT(NSET(11,N1))/SCALE+.0001
      IF (BUS(NFIRST)) 25,25,20
   20 N1=NSET(MX,N1)
      IF (N1.NE.7777) GO TO 15
      GO TO 40
   25 N2=NSET(MX,N1)
      CALL RMOVE(N1,12,NSET)
      CALL COLCT(1.0,69,NSET)
      MNEXT=ATRIB(11)+0.0001
      CALL PTJOB(3,NSET)
      N1=N2
      IF (N1.NE.7777) GO TO 15
   40 N1=MFE(1)
   45 IF (FLOAT(NSET(2,N1))/SCALE.GT.1.0) GO TO 60
      TILEFT=(NSET(1,N1))/SCALE-TNOW
      M1=FLOAT(NSET(11,N1))/SCALE+.0001
```

```
      SOPLD2(M1)=SHOPLD(M1)+ (TILEFT*CAPM(M1))/8.0
      QLOAD2(M1)= QLOAD(M1)+ (TILEFT*CAPM(M1))/8.0
   60 M1=NSET(MX,M1)
      IF (M1.NE.7777) GO TO 45
C
C     ***  CALCULATE DEVIATIONS FROM BALANCE
C
      DBALT=0.0
      DO 70 J=1,NM
      DBAL    =DESL(J)-SOPLD2(J)
      N30=J+30
      D30=DBAL
      CALL COLCT (D30,N30,NSET)
      D30AB=ABS(D30)
   70 DBALT=DBALT+D30AB
      CALL COLCT (DBALT,41,NSET)
C
C     ***  CALCULATE ADDITIONAL DEVIATIONS FROM BALANCE,IF
C     ***  REQUIRED,DEPENDING ON LOADING RULE USED.
C
      DBALQT=0.0
      DO 75 J=1,NM
      DBALQ    =DQL(J)-QLOAD2(J)
      N53=J+53
      D53=DBALQ
      CALL COLCT (D53,N53,NSET)
      D53AB=ABS(D53)
   75 DBALQT=DBALQT+D53AB
      CALL COLCT (DBALQT,64,NSET)
   80 IF (NTPDS.GE.NRST) CALL CLEAR (NSET)
      IF (NTPDS.LT.NTOTPD) RETURN
      CALL ENSIM (NSET)
      RETURN
      END




-FOR,IS SHOPLOADING.DUED
      FUNCTION DUED(WORK,IDUE)
C
C     ***  ASSIGNS A DUE DATE FOR AN INCOMING JOB
C
      IF(IDUE.EQ.2) GO TO 10
      DUED=WORK+DRAND(0.0)*150.0
      GO TO 20
   10 R1=DRAND(0.0)
      IF (R1.GT. .10) GO TO 15
      DUED=3.0*WORK
```

```
      GO TO 20
   15 DUED=WORK+R1*300.0
   20 RETURN
      END
```

```
-FOR,IS SHOPLOADING.DYNAM
      SUBROUTINE DYNAM (MBEST,NSET)
C
C     ***  SUBROUTINE USED WITH DYNAMIC SLACK
C     ***  DISPATCHING RULES
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEM,NLV,MHELD,NB(10),NBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,NNOW,NRST,NENDS,NHOL,NAL,
     3NNN(10),SELD,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,OLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSA,NLDR,NAPR,SHOPLD(10)
      XX=1.0E+20
      MN1=NNOW+1
      MBEST=0
      MNXT=MFE(MN1)
      IF (MNXT) 10,10,20
   10 CALL ERROR (201,NSET)
   20 DIF=NSET(4,MNXT)-NSET(6,MNXT)
      DS=DIF/SCALE-TNOW
      DSOP=DS/(FLOAT(NSET(5,MNXT))/SCALE)
      NSET(7,MNXT)=DSOP*SCALE+.000001
      IF (NRULE.EQ.2) GO TO 30
      IF (DS.LT.XX) MBEST=MNXT
      IF (DS.LT.XX) XX=DS
      GO TO 40
   30 IF (DSOP.LT.XX) MBEST=MNXT
      IF (DSOP.LT.XX) XX=DSOP
   40 MNXT=NSET(MX,MNXT)
      IF (MNXT-7777) 20,50,50
   50 RETURN
      END
```

```
-FOR,IS SHOPLOADING.ENDSV
      SUBROUTINE ENDSV (NSET)
C
C     ***  EVENT SUBROUTINE CALLED WHEN AN END OF SERVICE
C     ***  HAS OCCURRED FOR A JOB OPERATION
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,MB(10),MBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SELD,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NOVL,CAPY(10),DESL(10),
     1DQL(10),DESLF,DHLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
      MNOW=ATRIB(11)+0.00001
      MNEXT=ATRIB(13)+0.00001
      CALL TMST (XOPS,TNOW,13,NSET)
      XOPS=XOPS+1.0
      CALL TMST (XWKS,TNOW,14,NSET)
      XWKS=XWKS+ATRIB(12)
      ATRIB(32)=ATRIB(32)+ATRIB(12)
      ATRIB(5)=ATRIB(5)-1.0
      IF (ATRIB(5)) 10,10,60
C
C     ***  COLLECT STATISTICS ON THE JOB LEAVING THE SYSTEM
C
   10 TISYS=TNOW-ATRIB(3)
      CALL COLCT (TISYS,11,NSET)
      NOP=ATRIB(10)+0.00001
      NP23=NOP+22
      CALL COLCT (TISYS,NP23,NSET)
      CALL TMST (XISYS,TNOW,12,NSET)
      XISYS=XISYS-1.0
      CALL TMST (XWKSY,TNOW,11,NSET)
      XWKSY=XWKSY-ATRIB(9)
      DDD=ABS(TNOW-ATRIB(4))
      CALL COLCT (DDD,15,NSET)
      TLATE=TNOW-ATRIB(4)
      CALL COLCT (TLATE,12,NSET)
      CALL HISTO (TLATE,-10.0,2.0,1,NSET)
```

```
      TARDY=TLATE
      IF (TLATE.LT.0.0) TARDY=0.0
      CALL COLCT (TARDY,13,NSET)
      TSYNPL=TNOW-ATRIB(33)
      CALL COLCT (TSYNPL,42,NSET)
      NP40=NOP+39
      CALL COLCT (TSYNPL,NP40,NSET)
      TIPOOL=ATRIB(33)-ATRIB(3)
      CALL COLCT (TIPOOL,48,NSET)
      PERPOL=TIPOOL/PLEN+0.5
      NPEPOL=PERPOL
      CALL HISTO (NPEPOL,1.0,1.0,16,NSET)
      NP46=NOP+45
      CALL COLCT (TIPOOL,NP46,NSET)
      B=FLOAT(NTPDS-1)*PLEN
      BDUE=ATRIB(4)
      IF (BDUE.LT.B) GO TO 30
      IF (BDUE.LT.TNOW) GO TO 20
      LP= (TNOW-BDUE/PLEN)-.9999999
      GO TO 40
   20 LP=0
      GO TO 40
   30 LP=(B-BDUE)/PLEN+.999999
   40 XP=LP
      CALL HISTO (XP,-10.5,1.0,2,NSET)
      XOPS=XOPS-ATRIB(10)
      XWKS=XWKS-ATRIB(9)
      NLV=NLV+1
      JOB=ATRIB(30)+.001
      LOC(JOB)=0
      IF (JOB.NE.MAX) GO TO 80
   50 MAX=MAX-1
      JOB=JOB-1
      IF (LOC(JOB).LE.0) GO TO 50
      GO TO 80
C
C     *** THE JOB IS NOT LEAVING THE SYSTEM
C     *** UPDATE THE JOB ATTRIBUTES
C
   60 IF (NRULL.LE.3) ATRIB(6)=ATRIB(6)-ATRIB(12)
      LRM=ATRIB(5)+.001
      LR=2*LRM+9
      DO 70 I=11,LR,2
      ATRIB(I)=ATRIB(I+2)
   70 ATRIB(I+1)=ATRIB(I+3)
      ATRIB(LR+2)=0.0
      ATRIB(LR+3)=0.0
      CALL PTJOB (2,NSET)
C
```

```
C      ***  CHECK MACHINE QUEUE FOR ANY JOBS
C      ***  AVAIABLE FOR PROCESSING
C
   80 IF (NQ(MNOW+1)) 90,90,100
C
C      ***  THERE ARE NO JOBS IN THE QUEUE
C
   90 CALL TMST (BUS(MNOW),TNOW,MNOW,NSET)
      BUS(MNOW)=0.0
      IF (MSW(2).EQ.0) GO TO 93
      IF (NLDR.EQ.0) GO TO 93
      CALL COLCT(1.0,68,NSET)
      IF (NQ(12).LT.1) GO TO 93
      IF (MSW(3).EQ.0) GO TO 88
      IF (SSUMA(MNOW,3).GE.AVGLD9) GO TO 93
C
C      ***  TRY TO MOVE JOB FROM POOL TO EMPTY MACHINE
C
   88 J=0
      N1=MFE(12)
   91 J=J+1
      NFIRST=FLOAT(NSET(11,N1))/SCALE+.0001
      IF (NFIRST.EQ.MNOW) GO TO 92
      N1=NSET(MX,N1)
      IF (N1.NE.7777) GO TO 91
C
C      ***  NO JOB WAS FOUND THAT COULD HELP IDLE MACHINE
C
      GO TO 93
C
C      ***  PUT JOB FROM POOL IN IDLE MACHINE
C
   92 CALL RMOVE(N1,12,NSET)
      CALL COLCT (1.0,69,NSET)
      MNEXT=ATRIB(11)+.00001
      CALL PTJOB(3,NSET)
   93 RETURN
C
C      ***  MORE THAN ONE JOB IS AVAIABLE. COMPUTE
C      ***  PRIORITIES AND BRING IN THE JOB WITH THE
C      ***  HIGHEST PRIORITY FROM THE QUEUE.
C
  100 MN1=MNOW+1
      IF (NQ(MN1).EQ.1) GO TO 120
      IF (NRULE.EQ.0.OR.NRULE.GT.3) GO TO 120
      IF (NRULE.GT.2) GO TO 110
      CALL DYNAM (MBEST,NSET)
      CALL RMOVE (MBEST,MN1,NSET)
      GO TO 130
```

```
  110 CALL WKINQ (MBEST,NSET)
      IF (MBEST.EQ.0) GO TO 120
      CALL RMOVE (MBEST,MN1,NSET)
      GO TO 130
  120 CALL RMOVE (MFE(MN1),MN1,NSET)
C
C     ***  COMPUTE THE WAITING TIME FOR THE JOB AND
C     ***  DECREASE THE WORKLOAD IN THE MACHINE QUEUE.
C
  130 WT=TNOW-ATRIB(8)
      MN15=MNOW+15
      CALL COLCT (WT,MN15,NSET)
      QLOAD(MNOW)=QLOAD(MNOW)-ATRIB(12)
      SHOPLD(MNOW)=SHOPLD(MNOW)-ATRIB(12)
      TIMEVT=ATRIB(12) *(8.0/CAPM(MNOW ))
      ATRIB(1)=TNOW+TIMEVT
      ATRIB(2)=1.0
      JOB=ATRIB(30)+.001
      LOC(JOB)=MFA
      CALL FILEM(1,NSET)
      RETURN
      END


-FOR,IS SHOPLOADING.ENSIM
      SUBROUTINE ENSIM (NSET)
C
C     ***  SUBROUTINE USED TO PRINT SIMULATION RESULTS
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHLLD,WB(10),WBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NAL,
     3WAW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON MPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DGL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
      PRINT 160,NLDR,NRULE
```

```
      CALL TMST (XWKSY,TNOW,11,NSET)
      CALL TMST (XISYS,TNOW,12,NSET)
      CALL TMST (XOPS,TNOW,13,NSET)
      CALL TMST (XWKS,TNOW,14,NSET)
      DO 10 I=1,NM
   10 CALL TMST (BUS(I),TNOW,I,NSET)
      NTPDS=NTPDS-NRSET
      NNTP=NTOTPD-NRSET
      WRITE (6,170) NM,NRSET,NNTP,PLEN
      WRITE (6,171) (MSW(J),J=1,10)
      IF MSW(5).GT.0) GO TO 13
      WRITE (6,172)
      XN=NTPDS
      DO 12 I=1,NM
      J30=30+I
      XM1=SUMA(J30,1)/FLOAT(NTPDS)
      J53=53+I
      XM2=SUMA(J53,1)/FLOAT(NTPDS)
   12 WRITE (6,173) I,XM1,XM2
   13 CONTINUE
      XM3=SUMA(41,1)
      XM4=SUMA(64,1)
      XN3=SUMA(41,3)
      XN4=SUMA(64,3)
      XS3=SUMA(41,2)
      XS4=SUMA(64,2)
      AVG3=XM3/XN3
      AVG4=XM4/XN4
      VAR3=(((XN3*XS3)-(XM3*XM3))/(XN3*(XN3-1.0)))
      VAR4=(((XN4*XS4)-(XM4*XM4))/(XN4*(XN4-1-0)))
      WRITE (6,174)
      WRITE (6,175) AVG3,VAR3
      WRITE (6,176)
      WRITE (6,175) AVG4,VAR4
      J=11
      DO 16 I=1,3
      XM5=SUMA(J,1)
      XS5=SUMA(J,2)
      XN5=SUMA(J,3)
      AVG5=XM5/XN5
      VAR5=(((XN5*XS5)-(XM5*XM5))/(XN5*(XN5-1.0)))
      IF (I.GT.1) GO TO 14
      WRITE (6,177)
      J=48
      GO TO 16
   14 IF (I.GT.2) GO TO 15
      WRITE (6,178)
      J=42
      GO TO 16
```

```
15 WRITE (6,179)
16 WRITE (6,175) AVG5,VAR5
   XM7=SUMA(66,1)
   XN7=SUMA(66,3)
   XS7=SUMA(66,2)
   AVG7=XM7/XN7
   VAR7=(((XN7*XS7)-(XM7*XM7))/(XN7*(XN7 1.0)))
   STD=SQRT(ABS(VAR7))
   WRITE (6,185) AVG7,STD
   XM7=SUMA(67,1)
   XN7=SUMA(67,3)
   XS7=SUMA(67,2)
   AVG7=XM7/XN7
   VAR7=(((XN7*XS7)-(XM7*XM7))/(XN7*(XN7 1.0)))
   STD=SQRT(ABS(VAR7))
   WRITE (6,186) AVG7,STD
   IF MSW(5).GT.0) GO TO 17
   WRITE (6,180)
17 CONTINUE
   XN=NTPDS
   DO 20 I=1,NM
   AB= SUMA(I,1)/FLOAT(NTPDS)
   WWW(I)=WWW(I)/FLOAT(NTPDS)
   WBM(I)=(WBM(I)*XN-SSUMA(I,2)**2)/(XN*(XN-1.0))
   IF MSW(5).GT.0) GO TO 20
   WRITE (6,190) I,AB,WBM(I)
20 CONTINUE
   TWB=0.0
   DO 30 I=1,NM
30 TWB=TWB+WBM(I)/FLOAT(NM)
   SBM=(SUMA(14,2)*XN-SUMA(14,1)**2)/(XN*(XN-1.0))
   WRITE (6,200) TWB,SBM,MEN,MLV
   DO 40 I=12,14
   XS=SUMA(I,1)
   XSS=SUMA(I,2)
   XN=SUMA(I,3)
   AVGG=XS/XN
   VAR=(((XN*XSS)-(XS*XS))/(X.. (X . 1.0)))
   IF (I.EQ.12) PRINT 210, AVGG,VAR
   IF (I.EQ.13) PRINT 220, AVGG,VAR
   IF (I.EQ.14) AVGG=AVGG/PLEN*1 0.
   IF (I.EQ.14) PRINT 230, AVGG
40 CONTINUE
   DO 50 I=11,14
   XT=SSUMA(I,1)-TDEG
   XS=SSUMA(I,2)
   XSS=SSUMA(I,3)
   AVGG=XS/XT
   STD=(XSS/XT-AVGG*AVGG)
```

```
      STD=SIGN(SQRT(ABS(STD)),STD)
      IF (I.EQ.11) PRINT 240, AVGG,STD
      IF (I.EQ.12) PRINT 250, AVGG,STD
      IF (I.EQ.13) PRINT 252, AVGG,STD
      IF (I.EQ.14) PRINT 254, AVGG,STD
   50 CONTINUE
      TIME=FLOAT(NTPDS)*PLEN
      PRINT 260, NTPDS,TIME
      IF MSW(5).GT.0) GO TO 51
      PRINT 270, (I,WWW(I),I=1,NM)
   51 CONTINUE
      WRITE (6,361)
      WRITE (6,362)
      WRITE (6,363)
      QWB=0
      MAXQ=0
      XX=0
      NM1=NM+1
      DO 53 I=2,NM1
      XNQ=NQ(I)
      XE=(ENQ(I)+XNQ*(TNOW-QTIME(I)))/(TNOW-TBEG)
      VARE=((VNQ(I)+XNQ*XNQ*(TNOW-QTIME(I)))/(TNOW-TBEG)-XE*XE)
      IF (MAXNQ(I).GT.MAXQ) MAXQ=MAXNQ(I)
      IF MSW(5).GT.0) GO TO 52
      I1=I-1
      WRITE (6,364) I1,XE,VARE,MAXNQ(I)
   52 CONTINUE
      XX=XX+XE
   53 QWB=QWB+VARE
      XX=XX/FLOAT(NM)
      QWB=QWB/FLOAT(NM)
      WRITE (6,365) XX,QWB,MAXQ
      PWB=SUMA(70,1)/SUMA(70,3)
      WRITE (6,367) PWB
      PQB=SUMA(71,1)/SUMA(71,3)
      WRITE (6,368) PQB
      IF ( NRUN .GT. 1) GO TO 69
      IF MSW(5).GT.0) GO TO 69
      PRINT 280
      DO 60 I=1,NM
   60 WRITE (6,290) (X(I,J),J=1,NM)
      WRITE (6,300)
      WRITE (6,310) IDUL,     ITYPE,SEED,ISEED,NLDR
      RRATE=1.0/ARATE
      WRITE (6,320) ARATE
      WRITE (6,321) RRATE
      XM6=SUMA(65,1)
      XN6=SUMA(65,3)
      AVG6=XM6/XN6
```

```
      WRITE (6,322) AVG6
      WRITE (6,330) NPREL,NPREP,NDESL,DESLF
      WRITE (6,335) NDML,DMLF,NARR
      WRITE (6,336) FACDUD,SINPER
      WRITE (6,340)
      WRITE (6,290) (TIMEF(J),J=1,NM)
      WRITE (6,345)
      WRITE (6,290) (CAPM(J),J=1,NM)
      WRITE (6,350)
      WRITE (6,290) (DESL(J),J=1,NM)
      WRITE (6,355)
      WRITE (6,290) (DQL(J),J=1,NM)
C
C     ***  SET UP FOR NEXT RUN. CHANGE DISPATCHING RULE.
C     ***  INITIALIZE STATUS VARIABLES.
C
   69 NRULE=NRULE+1
C     ***  IT IS DESIRED TO SKIP RULE 5 (DUE DATE)
      IF (NRULE.EQ.5) NRULE=6
      IF (NRULE.LE.4) GO TO 120
      IF (NRULE.GT.5) GO TO 80
      DO 70 I=2,11
   70 KRANK(I)=4
      GO TO 120
   80 IF (NRULE.GT.6) GO TO 150
      DO 90 I=2,11
   90 KRANK(I)=8
      GO TO 120
  120 CONTINUE
      DO 130 I=1,NM
      AR(I)=0.0
      AB(I)=0.0
      ABM(I)=0.0
      AAW(I)=0.0
      BUS(I)=0.0
      SHOPLD(I)=0.0
  130 QLOAD(I)=0.0
      MAX=0
      DO 140 I=1,200
  140 LOC(I)=0
      AR(11)=0.0
      XOPS=0.0
      XWKS=0.0
      XTSYS=0.0
      XWKSY=0.0
      NEN=0
      NLV=0
      NPOOLB=0
      NPOOLA=0
```

```
      NHELD=0
      NTPOS=0
      XXSD=URAND(ISEED)
      NRST=NRSET
  150 NSTOP=-1
      RETURN
  160 FORMAT (1H1, 36HVARIOUS APPROACHES FOR JOB SHOP LOAD
     1,37HING USING DIFFERENT DISPATCHING RULES,/20X,3H LO
     2,21HADING APPROACH NUMBER,I5,/20X,15HDISPATCHING RUL
     3,9HE NUMBER ,I4///)
  172 FORMAT (1H ////5X,7HMACHINE, 13X,16HDEVIATION FROM
     1,6HBALANCE,/21X,14HAGGREGATE LOAD,14X,10HQUEUE LOAD)
  173 FORMAT (5X,I6,10X,F12.3,4X,F12.3)
  174 FORMAT (/5X,37HDEVIATION FROM BALANCE,AGGREGATE LOAD)
  175 FORMAT (1H ,5X,7HAVERAGE,2X,F10.3,8HVARIANCE,5X,F10.3)
  176 FORMAT (/5X,33HDEVIATION FROM BALANCE,QUEUE LOAD)
  177 FORMAT (///5X,24HTIME SPENT IN THE SYSTEM)
  178 FORMAT (/5X,26HTIME SPENT IN THE JOB POOL)
  179 FORMAT (/5X,38HTIME SPENT IN THE SYSTEM W/O POOL TIME)
  252 FORMAT (10X, 37HW.I.P.(AVERAGE OPERATIONS PERFORMED P
     120HER JOB IN THE SHOP)./15X,5HAVG= ,F10.3,6H  STD=,F10.3)
  254 FORMAT(10X,39HW.I.P.(AVERAGE HOURS OF WORK DONE FOR J
     117HOBS IN THE SHOP)./15X,5HAVG= ,F10.3,6H  STD=,F10.3)
  170 FORMAT (5X, 37H NUMBER OF MACHINES IN THE SIMULATED
     15HSHOP ,I6/5X,26H NUMBER OF RUN IN PERIODS ,I8/5X,
     246H NUMBER OF TIME PERIODS SIMULATED AFTER RUN IN,
     3I6/5X, 28H LENGTH OF EACH TIME PERIOD ,F8.2)
  171 FORMAT (1H ,5X,16HSPECIAL FEATURES,4X,10I1)
  180 FORMAT (1H1////5X,  7HMACHINE, 18H  UTILIZATION BAL
     1,12HANCE MEASURE)
  185 FORMAT (//1H ,37HJOBS IN THE POOL BEFORE LOADING  AVG ,
     1F7.2,6H  STD ,F7.2)
  186 FORMAT (//1H ,37HJOBS IN THE POOL AFTER  LOADING  AVG ,
     1F7.2,6H  STD ,F7.2)
  190 FORMAT (5X,I6,F12.3,F14.3)
  200 FORMAT (///10X, 25HMACHINE BALANCE MEASURE =,F12.3/
     110X, 22HSHOP BALANCE MEASURE =,F12.3/10X,
     230HNUMBER OF JOBS ENTERING SHOP =,I7/10X,
     332HNUMBER OF JOBS LEAVING SHOP = ,I7)
  210 FORMAT (10X, 23HAVERAGE JOB LATENESS = ,F10.2/10X,
     128HAVERAGE LATENESS VARIANCE = ,F10.2)
  220 FORMAT (10X, 23HAVERAGE JOB TARDINESS =,F12.3/10X,
     1 28HAVERAGE TARDINESS VARIANCE =,F12.3)
  230 FORMAT (10X,26HAVERAGE SHOP UTILIZATION =,F12.3)
  240 FORMAT (1 X, 34HAVERAGE W.I.P.(IN HOURS OF WORK) =,
     1F12.3,1X,F14.3)
  250 FORMAT (10X, 34HAVERAGE NUMBER OF JOBS IN THE SHOP,
     12H =,F12.3,F12.3)
  260 FORMAT (10X, 29HLENGTH OF SIMULATION RUN WAS ,/10X,
```

```
   1I5, 15H  TIME PERIODS , 1H(,F10.1,  9H HOURS ) )
 270 FORMAT (/////15X, 7HMACHINE,3X, 13HAVG INPUT/PD./
   1(15X,I4,8X,F7.2))
 280 FORMAT (1H1//9X, 32HTHE JOB SHOP PROBABILITY TRANSIT
   1,10HION MATRIX,////)
 290 FORMAT (3X,10F6.3)
 300 FORMAT (/////, 34H     IDUE        ODD      ITYPE
   1,23H  SEED      ISEED   NLDR)
 310 FORMAT (/5X,I4,4X,7X,5X,I5,2X,F10.4,3X,I8,I4)
 320 FORMAT (/5X, 26HMEAN INPUT ARRIVAL RATE = ,F7.4,
   1 16H ARRIVALS/HOUR   )
 321 FORMAT (5X,29HMEAN TIME BETWEEN ARRIVALS = ,F7.4,
   18H   HOURS)
 322 FORMAT (5X,36HACTUAL MEAN TIME PER OPERATION    =
   1F7.4,7H  HOURS)
 330 FORMAT (1H //3X,7HNPREL= ,I5,3X,7HNPREP= ,I5,3X,3HNDE
   14HSL= ,I5,3X,7HDESLF= ,F10.3)
 335 FORMAT (1H ,3X,6HNDML= ,I5,3X,6HDMLF= ,F10.3,3X,3HNAP
   13HR= ,I5)
 336 FORMAT (1H ,3X,8HFACDUD= ,F8.2,3X,8HSINPER= ,F8.2)
 340 FORMAT (1H ///5X,33HJOB OPERATION TIME FACTORS FOR EA
   110HCH MACHINE)
 345 FORMAT (1H ///10X,32HMACHINE CENTER CAPACITIES PER PE
   14HRIOD)
 350 FORMAT (1H ///10X,34HDESIRED AGGREGATE LOAD PER MACHINE)
 355 FORMAT (1H //10X,30HDESIRED QUEUE LOAD PER MACHINE)
 361 FORMAT (///5X,22HOTHER BALANCE MEASURES/)
 362 FORMAT (/5X,27HMACHINE QUEUE BALANCE INDEX)
 363 FORMAT (3X,7HMACH NO,7X,7HAVERAGE,12X,3HQWB,8X,7HMAXIMUM)
 364 FORMAT (5X,I4,2F15.3,10X,I5)
 365 FORMAT (/5X,4H ALL,2F15.3,10X,I5)
 367 FORMAT (/5X,25HPERIOD WORK BALANCE INDEX,4X,6HPWB = ,F9.3)
 368 FORMAT (/5X,26HPERIOD QUEUE BALANCE INDEX,4X,6HPQB = ,F9.2)
     END
```

```
-FOR,IS SHOPLOADING.EVNTS
      SUBROUTINE EVNTS (IX,NSET)
C
C      *** DIRECTS THE PROGRAM TO THE
C      *** PROPER EVENT SUBROUTINE
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
```

```
      4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
      5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
      6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
       COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDME,
      1ITYPE,MNEXT,MEN,NLV,NHELD,WB(10),BM(10),X(10,10),
      2         BUS(10),NRSET,NRULE,NNOW,NRST,NENDS,NHOL,NPL,
      3WAW(10),SELD,ARATE,LOC(200),MAX,AR(11)
       COMMON NPREL,NPREP,NDESL,NDML,CAPH(10),DESL(10),
      1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
      2NSTSW,NLDR,NARR,SHOPLD(10)
       GO TO (10,20,30,40), IX
   10 CALL ENDSV (NSET)
      RETURN
   20 CALL ARIVL (NSET)
      RETURN
   30 CALL COLL (NSET)
      RETURN
   40 CALL START (NSET)
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.GENMAT
      SUBROUTINE GENMAT (NSET)
C
C     ***  THIS SUBROUTINE PLACES THE PARAMETERS FOR THE JOBS
C     ***  IN THE JOB POOL IN THE FORM REQUIRED BY LP
C
      DIMENSION NSET(35,1),DUDFT(50), MFOP(10,70) ,KAUX(5)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDME,
     1ITYPE,MNEXT,MEN,NLV,NHELD,WB(10),BM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,NNOW,NRST,NENDS,NHOL,NPL,
     3WAW(10),SELD,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPH(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
C
C     ***  OBTAIN NO.OF JOBS IN POOL AND INITIALIZE MATRICES
C
```

```
      NPOOL=NQ(12)
      MROW=25
      MCOL=80
      NOROW=NM
      NOCOL=NPOOL+2*NM
      INDEX=0
      DO 3 I=1,5
    3 KAUX(I)=0
      OBJIN=0.0
      DO 1 I=1,NM
      DO 1 J=1,NPOOL
      WFOP(I,J)=0.0
    1 CONTINUE
      DO 2 I=1,NM
      DO 2 J=1,NOCOL
    2 A(I,J)=0.0
      J=0
      N1=MFE(12)
C
C     ***  OBTAIN LP MATRIX ENTRIES FOR EACH JOB
C
   30 J=J+1
      WKTIM=0.0
      NO1=FLOAT(NSET(10,N1))/SCALE+.000001
      DO 35 I=1,NO1
      NON1=9+2*I
      NON2=FLOAT(NSET(NON1,N1))/SCALE+.000001
      NON3=NON1+1
      WOL=FLOAT(NSET(NON3,N1))/SCALE
      A(NON2,J)=WOL
      WKTIM=WKTIM+WOL*(8.00/CAPM(NON2))
      IF (NLDR.NE.2.OR.NLDR.NE.3) GO TO 35
      IF (NON1.NE.11) GO TO 35
      WFOP(NON2,J)=WOL
   35 CONTINUE
      TIMDUE =FLOAT(NSET(4,N1))/SCALE+.000001
      WRKDUE =FLOAT(NSET(9,N1))/SCALE+.000001
      DUDSLK=TIMDUE-TNOW-WKTIM
      IF (DUDSLK    .LE.0.0) DUDSLK    =0.0
      DUDFT(J)=FACDUD/(DUDSLK+.01)
C
C     ***  OBTAIN NEXT JOB IN THE POOL,IF THERE IS ANY
C
      N1=NSET(MX,N1)
      IF (N1.NE.7777) GO TO 30
C
C     ***  SET UP MATRICES REQUIRED BY LPI
C
      IF (NLDR.EQ.2.OR.NLDR.EQ.3) GO TO 60
```

```
      DO 51  I=1,NOROW
      DO 52  J=1,NOCOL
      IF (J.EQ.(NPOOL+I)) A(I,J)=1.
      IF (J.EQ.(NPOOL+NM+I)) A(I,J)=-1.
   52 CONTINUE
      A(I,NOCOL+1)=DESL(I)-SHOPLD(I)
      AA=A(I,NOCOL+1)
      OBJIN=OBJIN+ABS(AA)
      KBV(I)=NPOOL+I
      IF (AA.GE.0.0) GO TO 51
      A(I,NOCOL+1)= -AA
      KBV(I)= NPOOL+NM+I
      DO 54  J=1,NOCOL
   54 A(I,J)= -A(I,J)
   51 CONTINUE
      GO TO 71
C
C     ***  MATRIX PREPARATION WHEN NEXT QUEUE RULE IS USED
C
   60 DO 61  I=1,NOROW
      DO 62  J=1,NOCOL
      IF (J.LE.NPOOL) A(I,J)=WHOP(I,J)
      IF (J.EQ.(NPOOL+I)) A(I,J)=1.
      IF (J.EQ.(NPOOL+NM+I)) A(I,J)=-1.
   62 CONTINUE
      A(I,NOCOL+1)= DQL(I)-QLOAD(I)
      AA=A(I,NOCOL+1)
      OBJIN=OBJIN+ABS(AA)
      KBV(I)=NPOOL+I
      IF (AA.GE.0.0) GO TO 61
      A(I,NOCOL+1)= -AA
      KBV(I)= NPOOL+NM+I
      DO 64  J=1,NOCOL
   64 A(I,J)= -A(I,J)
   61 CONTINUE
   71 CONTINUE
      DO 76  J=1,NOCOL
      C(J)=0.0
      IF (J.GT.NPOOL) GO TO 77
      C(J)=-DUDFT(J)
      A(NOROW+1,J)=1.0
      A(NOROW+2,J)=1.0
      GO TO 76
   77 C(J)=1.0
      A(NOROW+1,J)=-1.0
      A(NOROW+2,J)=1.0
   76 CONTINUE
      C(NOCOL+1)= -OBJIN
      A(NOROW+1,NOCOL+1)= 0.0
```

```
      A(NOROW+2,NOCOL+1)= 0.0
      DO 88 I=1,NOROW
      DO 88 J=1,NOCOL
      C(J)=C(J)-A(I,J)
   88 CONTINUE
      IF (NLDR.GE.4) GO TO 91
      CALL LPI (NSET,NOROW,NOCOL,MROW,MCOL,INDEX,KAUX)
      GO TO 92
   91 CALL POOLHE (NSET,NOROW,NOCOL)
   92 RETURN
      END
```

```
-FOR,IS SHOPLOADING.GNARV
      FUNCTION GNARV(RNUM)
C     ***  COMPUTES TIME FOR THE NEXT JOB ARRIVAL
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,WB(10),MBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DGL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER
      IF (NARR.GT.1) GO TO 10
      GNARV=-1.0/ARATE*ALOG(RNUM)
      IF(GNARV.GT.40.) GNARV=40.0
      GO TO 20
   10 IF (ICOUNT.LE.5) GO TO 15
      ICOUNT=0
      NCOUNT=NCOUNT+1
   15 ICOUNT=ICOUNT+1
      A1=-1.0/ARATE*ALOG(RNUM)
      A2=(6.28*NCOUNT)/SINPER
      IF(A1.GT.40.) A1=40.0
      S=SIN(A2)
      GNARV=A1*(1.0+0.5*S)
   20 RETURN
      END
```

```
-FOR,IS SHOPLOADING.JOBDEC
      SUBROUTINE JOBDEC (NSET,NOROW,NOCOL)
C
C     ***  THIS SUBROUTINE USES THE LP RESULTS TO MAKE THE
C     ***  FINAL SELECTION REGRADING THE JOBS THAT SHOULD
C     ***  BE LOADED IN THE SHOP
C
      DIMENSION  NSET(35,1),XJOB(100)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT.
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT.OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,MB(10),MBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XCPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      NPOOL=NQ(12)
      DO 1 J=1,NPOOL
      XJOB(J)=0.0
      AA1=A(NOROW+2,J) -.0001
      IA=IFIX(AA1)
    1 IF (IA.EQ.-1) XJOB(J)=1.0
      IF (NLDR.GE.4) GO TO 20
      DO 2 I=1,NM
      JJ=KBV(I)
      XJOB(JJ)=A(I,NOCOL+1)
C
C     ***  VARIABLES IN BASIS AND WITH UPPER BOUND INDICATOR
C     ***  ON, NEED TO BE CALCULATED DIFFERENTLY
C
      AA1=A(NOROW+2,JJ)-.001
      IA=IFIX(AA1)
      IF (IA.NE.-1) GO TO 2
      XJOB(JJ)=A(NOROW+1,JJ)-A(I,NOCOL+1)
    2 CONTINUE
C
C     ***  SEARCH JOB POOL FILE AND LOAD IN THE SHOP THOSE
C     ***  JOBS WITH DECISION VARIABLE .GE. .75
C
   20 J=0
      N1=MFE(12)
      WKSHP1=0.0
```

```
      TDES1=0.0
      DO 25 I=1,NM
      TDES1=TDES1+DESL(I)
   25 WKSHP1=WKSHP1+SHOPLD(I)
   30 J=J+1
      XJBN=XJOB(J)
      IF (XJBN.LT.0.75) GO TO 40
      N2=NSET(MX,N1)
      CALL RMOVE (N1,12,NSET)
      WKSHP1=WKSHP1+ATRIB(9)
      MNEXT=ATRIB(11) +.00001
      CALL PTJOB (3,NSET)
      N1=N2
      GO TO 41
   40 N1=NSET(MX,N1)
   41 CONTINUE
      IF (N1.NE.7777) GO TO 30
C
C     ***  SEARCH JOB POOL FILE AND LOAD JOBS WITH DECISION
C     ***  VARIABLES BETWEEN 0.3 AND 0.75 IF TOTAL SHOP LOAD
C     ***  IS LESS THAN DESIRED
C
      J=0
      IF (NQ(12).EQ.0) GO TO 70
      N1=MFE(12)
   50 IF (WKSHP1.GE.TDES1) GO TO 70
   55 J=J+1
      IF (J.GT.NPOOL) GO TO 70
      XJBN=XJOB(J)
      IF (XJBN.GE.0.75) GO TO 55
      IF (XJBN.LT.0.3) GO TO 65
      N2=NSET(MX,N1)
      CALL RMOVE (N1,12,NSET)
      WKSHP1=WKSHP1+ATRIB(9)
      MNEXT=ATRIB(11) +.00001
      CALL PTJOB (3,NSET)
      N1=N2
      GO TO 66
   65 N1=NSET(MX,N1)
   66 CONTINUE
      IF (N1.NE.7777) GO TO 50
   70 CONTINUE
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.LPI
      SUBROUTINE LPI (NSET,NOROW,NOCOL,MROW,MCOL,INDEX,KAUX)
C
C     ***  THIS SUBROUTINE CALCULATES DECISION VARIABLES USED
C     ***  FOR LOADING JOBS INTO THE SHOP
C
      DIMENSION NSET(35,1),KAUX(5),COLIN(10),IDONE(10)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NLN,NLV,NHELD,WB(10),WBM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      DO 51 I=1,10
   51 COLIN(I)=0.
      EPS=.000001
      MAXCOL=MCOL-1
      MAXROW=MROW
      NORHS=NOCOL+1
      NUPPER=NOROW+1
      NUPSW=NOROW+2
      IF(NOROW.GT.MAXROW.OR.NOCOL.GT.MAXCOL) GO TO 910
      NOITER=-1
      INCOL=-1
C
C     *** BEGIN MAIN ITERATION LOOP
C
  200 NOITER=NOITER+1
  625 CONTINUE
C
C     ***  CHECK OPTIMALITY AND-OR FIND INCOMING COLUMN
C
  250 IF(INCOL.EQ.0) GO TO 900
      INCOL=0
      CMIN=-EPS
      DO 260 NC=1,NOCOL
      IF(C(NC).GE.CMIN) GO TO 260
      CMIN=C(NC)
      INCOL=NC
  260 CONTINUE
```

```
      IF(INCOL.EQ.0) GO TO 900
C
C     ***  PICK ROW TO PIVOT ON
C     ***  ISUB=0 IMPLIES STANDARD PIVOT
C     ***  ISUB=1 IMPLIES PIVOT AND UPPER SUBSTITUTE
C     ***  ISUB=2 IMPLIES UPPER SUBSTITUTE
C
      INROW=0
      RATMIN=999999.
      DO 282 NR=1,NOROW
      IF(A(NR,INCOL).LE.EPS) GO TO 280
      RATIO=A(NR,NORHS)/A(NR,INCOL)
      IF(RATIO.GE.RATMIN) GO TO 280
      RATMIN=RATIO
      INROW=NR
      ISUB=0
  280 IF(A(NR,INCOL).GE.-EPS) GO TO 282
      NDEX=KBV(NR)
      IF(A(NUPPER,NDEX) .LE. -EPS) GO TO 282
      RATIO=(A(NR,NORHS)-A(NUPPER,NDEX))/A(NR,INCOL)
      IF(RATIO.GE.RATMIN) GO TO 282
      RATMIN=RATIO
      INROW=NR
      ISUB=1
  282 CONTINUE
      IF(A(NUPPER,INCOL) .LE. -EPS) GO TO 281
      IF(A(NUPPER,INCOL) .GE. RATMIN) GO TO 281
      RATMIN=A(NUPPER,INCOL)
      INROW=NUPPER
      ISUB=2
  281 IF(INROW.NE.0) GO TO 301
C
C     ***  CHECK FOR AUXILIARY VARIABLES IN BASIS
C
      IF(INDEX.EQ.0) GO TO 420
      IT=1
      DO 421 I=1,NOROW
      DO 422 J=1,INDEX
      IF(KBV(I).NE.KAUX(J)) GO TO 422
      IDONE(IT)=KBV(I)
      IT=IT+1
      GO TO 421
  422 CONTINUE
  421 CONTINUE
      IF(IT.GT.1) GO TO 423
  420 CONTINUE
C
C     ***  UNBOUNDED SOLUTION
C
```

```
      WRITE(6,285) INCOL
  285 FORMAT(1H ,30HSOLUTION UNBOUNDED--ADDING COL,I5)
      GO TO 625
C
C     ***  PIVOT
C
  301 IF(ISUB .LT. 2) GO TO 304
  302 C(NORHS)=C(NORHS)-C(INCOL)*A(NUPPER,INCOL)
      C(INCOL)=-C(INCOL)
      RATMIN=A(NUPPER,INCOL)
      DO 303 NC=1,NOROW
      A(NC,NORHS)=A(NC,NORHS)-A(NC,INCOL)*A(NUPPER,INCOL)
      A(NC,INCOL)=-A(NC,INCOL)
  303 CONTINUE
      A(NUPSW,INCOL)=-A(NUPSW,INCOL)
      GO TO 200
  304 NCROW=KBV(INROW)
      KBV(INROW)=INCOL
      DO 305 NR=1,NOROW
  305 COLIN(NR)=A(NR,INCOL)
      CSTIN=C(INCOL)
      COEF= A(INROW,INCOL)
      DO 330 NC=1,NORHS
      A(INROW,NC)=A(INROW,NC)/COEF
      CORR=A(INROW,NC)
      DO 310 NR=1,NOROW
      IF(NR.EQ.INROW) GO TO 310
      A(NR,NC)=A(NR,NC)-COLIN(NR)*CORR
  310 CONTINUE
      C(NC)=C(NC)-CSTIN*CORR
  330 CONTINUE
      IF(ISUB.LT.1) GO TO 200
      INCOL=NCROW
      GO TO 302
C
C     ***  END MAIN ITERATION LOOP
C     ***  OPTIMAL SOLUTION
C
  900 CONTINUE
      GO TO 999
  423 WRITE(6,193)
  193 FORMAT(1H ,40HSOLUTION INFEASIBLE,AUXILITARY VARIABLES
     14HAVE,/1H ,23HVALUE GREATER THAN ZERO)
      GO TO 998
  910 WRITE (6,911)
  911 FORMAT(1H ,24HTOO MANY ROWS OR COLUMNS)
  995 WRITE (6,996)
  996 FORMAT(1H ,23H***SHOP LOADING ABORTED)
      GO TO 998
```

```
  999 CALL JOBDEC(NSET,NOROW,NOCOL)
  998 CONTINUE
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.MAIN
C     PROGRAM MAIN
      DIMENSION NSET(35,200)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,WB(10),WDM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DGL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
      NCRDR=5
      NPRNT=6
C
C     ***  READ IN SIMULATION PARAMETERS
C
      READ(5,40) NM,NTOTPD,NRSET,PLEN,ISEED
      XXSD=DRAND(ISEED)
      READ(5,50)       ITYPE,NRULE,IDUE,NLDR
      READ (5,55) (MSW(J),J=1,10)
      READ(5,60) ARATE,NARR,FACDUD,SINPER
      AVGLD9=(ARATE*8.0*6.0*2.48)/FLOAT(NM)
      AVGLD9=AVGLD9-1.24
      IF (NRSET.LE.0) NRSET=0
      NTOTPD=NTOTPD+NRSET
      NRST=NRSET
      IF (NRSET.EQ.0) NRST=999999
C
C     ***  READ IN TRANSITION MATRIX
C
      DO 10 I=1,NM
   10 READ (5,70) (X(I,J),J=1,NM)
      READ (5,80) NPREL,NPREP,NDESL,DESLF,NDML,DMLF
```

```
      READ (5,70) (TIMEF(J),J=1,NM)
      READ (5,90) (CAPM(J),J=1,NM)
      IF (NDESL.NE.0) GO TO 12
      READ (5,90) (DESL(J),J=1,NM)
      GO TO 15
   12 DO 13 J=1,NM
   13 DESL(J)=DESLF*CAPM(J)
   15 IF (NDML.NE.0) GO TO 16
      READ (5,90) (DQL(J),J=1,NM)
      GO TO 18
   16 DO 17 J=1,NM
   17 DQL(J)=DMLF*CAPM(J)
   18 CONTINUE
C
C     ***  INITIALIZE THE STATUS VARIABLES
C
      DO 20 I=1,NM
      WB(I)=0.0
      WBM(I)=0.0
      AR(I)=0.0
      WWW(I)=0.0
      BUS(I)=0.0
      SHOPLD(I)=0.0
   20 QLOAD(I)=0.0
      AR(11)=0.0
      XCPS=0.0
      XWKS=0.0
      XISYS=0.0
      XWKSY=0.0
      NEN=0
      MAX=0
      NLV=0
      NHELD=0
      NTPOS=0
      ICOUNT=0
      NCOUNT=0
      DO 30 I=1,200
   30 LOC(I)=0
      CALL GASP(NSET)
   40 FORMAT (3I5,F10.3,I10)
   50 FORMAT (4I5)
   55 FORMAT(10I1)
   60 FORMAT (F10.5,I5,F10.5,F10.5)
   70 FORMAT (10F7.4)
   80 FORMAT (3I5,F10.4,I5,F10.4)
   90 FORMAT (10F7.2)
      END
```

```
-FOR,IS SHOPLOADING.MOP
      FUNCTION MOP(R)
C     ***  COMPUTES NUMBER OF OPERATIONS FOR AN INCOMING JOB
      MOP=4
      IF(R.GT. 0.15) MOP=5
      IF(R.GT. 0.35) MOP=6
      IF(R.GT. 0.65) MOP=7
      IF(R.GT. 0.85) MOP=8
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.OTPUT
      SUBROUTINE OTPUT (NSET)
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.POOLHE
      SUBROUTINE POOLHE(NSET,NOROW,NOCOL)
C
C     ***  THIS SR LOADS THE JOBS FROM THE POOL BY ATTEMPTING
C     ***  TO KEEP THE QUEUES AT A CERTAIN LEVEL
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XMKSY,IDUE,
     1ITYPE,NNEXT,NEN,NLV,NHELD,WB(10),MDM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NSL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDNL,CAPM(10),DESL(10),
     1DGL(10),DESLF,DNLF,CLOAD(10),XCPS,XMKS,TIMFF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
      JJ1=NQ(12)
      DO 1 J=1,JJ1
    1 A(NOROW+2,J)=0.0
      TEMPAG=0.0
      DO 20 I=1,NM
```

```
      TEMPQL=0.0
    5 MTEST=0
      MBEST=0
      DIF=DQL(I)-QLOAD(I)-TEMPQL
      IF (DIF.LE.0.0) GO TO 20
      N2=MFE(12)
      DO 10 J=1,JJ1
      N1=N2
      IF(A(NOROW+2,J).LT.-.0001) GO TO 10
      IF (A(I,J).LT.0.0) A(I,J)=-A(I,J)
      IF (A(I,J).LT.0.0001) GO TO 10
      IF (C(J).GE.MTEST) GO TO 10
      MBEST=J
      MTEST=C(J)
   10 N2=NSET(MX,N1)
      IF (MBEST.NE.0) A(NOROW+2,MBEST)=-1.0
      TEMPQL=A(I,MBEST)+TEMPQL
      TEMPAG=TEMPAG+NSET(9,N1)/SCALE
      IF (MBEST.EQ.0) GO TO 20
      GO TO 5
   20 CONTINUE
      IF (NLDR.NE.6) GO TO 50
      TOTLD=0.0
      ACTLD=0.0
      DO 30 I=1,NM
      TOTLD=TOTLD+DESL(I)
   30 ACTLD=ACTLD+SHOPLD(I)
      ACTLD=ACTLD+TEMPAG
      TEMPTL=0.0
   36 MTEST=0
      MBEST=0
      DIF=TOTLD-ACTLD-TEMPTL
      IF (DIF.LE.0.0) GO TO 50
      N2=MFE(12)
      DO 40 J=1,JJ1
      N1=N2
      IF(A(NOROW+2),J).LT.-.0001) GO TO 40
      IF (C(J).GE.MTEST) GO TO 40
      MBEST=J
      MTEST=C(J)
   40 N2=NSET(MX,N1)
      IF (MBEST.NE.0) A(NOROW+2,MBEST)=-1.0
      TEMPTL=TEMPTL+NSET(9,N1)/SCALE
      IF (MBEST) 50,50,38
   50 CONTINUE
      CALL JOBDEC(NSET,NOROW,NOCOL)
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.PTJOB
      SUBROUTINE PTJOB (INP,NSET)
C
C     ***  SUBROUTINE WHICH MOVES JOB TO NEXT MACHINE
C     ***  CENTER
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,N4,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,WB(10),WBM(10),X(10,10),
     2        BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPY(10),DESL(10),
     1DGL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
C
C     ***  CHECK IF JOB IS A NEW ARRIVAL
C
      IF (INP.NE.1) GO TO 10
      ATRIB(3)=TNOW
      NEN=NEN+1
C
C     ***  NEW ARRIVAL. CHECK IF A JOB POOL IS BEING USED
C
      IF (NLDR.EQ.0) GO TO 20
C
C     ***  CHECK IF SHOP IS BEING PRELOADED AND JOB POOL
C     ***  HAS BEEN COMPLETED
C
      IF (NSTSW.EQ.1) GO TO 20
C
C     ***  PUT ARRIVING JOB IN THE POOL IF OP. 1 MACH IS NOT IDLE
C
      ATRIB(8)=TNOW
      JOB=ATRIB(30)+0.001
      LOC(JOB)=MFA
C
C     ***  COLLECT STATISTICS ON INTERARRIVAL TIMES TO
C     ***  THE JOB POOL
C
      D=TNOW-AR(11)
```

```
      CALL HISTO (D,0.5,0.5,15,NSET)
      AR(11)=TNOW
      NFIRST=ATRIB(11)+0.00001
      IF (MSW(1).EQ.0) GO TO 4
      IF (TNOW.LE.0.0001) GO TO 4
      IF (BJS(NFIRST)) 5,5,4
    4 CALL FILEM(12,NSET)
      GO TO 70
C
C     ***  IF FIRST OPERATION MACHINE IS IDLE,CONSIDER THE
C     ***  JOB AS COMING FROM POOL AND PUT IN THE SHOP
C
    5 CONTINUE
      IF (MSW(3).EQ.0) GO TO 6
      IF(SSUMA(NFIRST,3).GE.AVGLD9) GO TO 4
    6 MNEXT=NFIRST
      CALL COLCT (1.0,69,NSET)
      GO TO 20
C
C     ***  JOB IS NOT A NEW ARRIVAL. CHECK IF IT IS COMING
C     ***  FROM THE POOL
C
   10 IF (INP.EQ.2) GO TO 40
C
C     ***  JOB IS COMING FROM THE POOL.
C     ***  ALSO NEW JOBS WHEN A POOL IS NOT USED ARRIVE
C     ***  AT THIS POINT
C     ***  UPDATE STATUS OF WORK IN SHOP AND ALSO UPDATE
C     ***  AGGREGATE LOAD IN SHOP QUEUES FOR EACH MACHINE.
C
   20 CALL TMST (XISYS,TNOW,12,NSET)
      CALL TMST (XWKSY,TNOW,11,NSET)
      XISYS=XISYS+1.0
      XWKSY=XWKSY+ATRIB(9)
      ATRIB(33)=TNOW
      NNN=9.0+2.*ATRIB(10)+.00001
      DO 37 I=11,NNN,2
      J=ATRIB(I)
   37 SHOPLD(J)=SHOPLD(J)+ATRIB(I+1)
C
C     ***  JOB IS NOT GOING INTO THE POOL. COLLECT STATISTICS
C     ***  ON INTERARRIVAL TIMES TO THE CURRENT MACHINE
C
   40 D=TNOW-AR(MNEXT)
      MN4=MNEXT+4
      CALL HISTO (D,0.5,0.5,MN4,NSET)
      AR(MNEXT)=TNOW
C
C     ***  CHECK ON THE STATUS OF MACHINE FOR NEXT
```

```
C      ***  JOB OPERATION
C
       IF (BUS(MNEXT)) 60,60,50
C
C      ***  NEXT MACHINE IS BUSY. JOB CAN NOT BE PUT ON
C           MACHINE
C
    50 ATRIB(8)=TNOW
       MX1=MNEXT+1
       JOB=ATRIB(30)+0.001
       LOC(JOB)=MFA
       QLOAD(MNEXT)=QLOAD(MNEXT)+ATRIB(12)
       CALL FILEM (MX1,NSET)
       GO TO 70
C
C      ***  NEXT MACHINE IS NOT BUSY.
C      ***  JOB MAY BE PUT ON MACHINE
C
    60 CALL TMST (BUS(MNEXT),TNOW,MNEXT,NSET)
       BUS(MNEXT)=1.0
       WT=0.0
       MX15=MNEXT+15
       CALL COLCT (WT,MX15,NSET)
       TIMEVT=ATRIB(12) *(8.0/CAPM(MNEXT))
       ATRIB(1)=TNOW+TIMEVT
       ATRIB(2)=1.0
       J=ATRIB(11)
       SHOPLD(J)=SHOPLD(J)-ATRIB(12)
       JOB=ATRIB(30)+0.001
       LOC(JOB)=MFA
       CALLFILEM  (1,NSET)
    70 NSTSW=0
       RETURN
       END
```

```
-FOR,IS SHOPLOADING.START
      SUBROUTINE START (NSET)
C
C     *** GENERATES INITIAL JOBSET IN THE SHOP AT TIME ZERO
C     ***  AND ESTABLISHES THE JOB POOL IF REQUIRED
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,NNEXT,NEM,NLV,NHELD,WB(10),WBM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,VNOW,NFST,NENDS,NHCL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPY(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,MARR,SHOPLD(10)
      TNOW=0.0
      TBEG=TNOW
      NSTSW=0
      DO 10 I=1,NPREL
C
C     ***  DO NOT SET SWITCH IF POOL IS STILL BEING LOADED
C
      IF(I.LE.NPREP) GO TO 7
      NSTSW=1
    7 CALL ARVL(NSET)
   10 CONTINUE
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.TIME
      FUNCTION TIME(RNUM)
C     ***  COMPUTES TIME FOR A JOB OPERATION
      TIME=-2.48*ALOG(RNUM)
      IF(TIME.LT.1.00) TIME=1.00
      IF(TIME.GT.9.00) TIME=9.00
      RETURN
      END
```

```
-FOR,IS SHOPLOADING.WKINQ
      SUBROUTINE WKINQ (MBEST,NSET)
C
C     ***  SUBROUTINE USED WITH WORK IN NEXT QUEUE
C     ***  DISPATCHING RULE
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,MEN,MLV,NHELD,WB(10),WBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
C
C   - ***  CHECK JOBS IN QUEUE FOR THIS MACHINE
C
      XX=1.0E+20
      MBEST=0
      MN1=MNOW+1
      MXT=MFE(MN1)
   10 NMM=FLOAT(NSET(13,MXT))/SCALE+.000001
C
C     ***  CHECK IF THIS JOB HAS A NEXT OPERATION
C
      IF (NMM.EQ.0) GO TO 50
      TWK=QLOAD(NMM)
      TM=TNOW+FLOAT(NSET(12,MXT))/SCALE
      N1=MFE(1)
C
C     ***  CHECK IF NEXT EVENT IS AN END OF SERVICE
C
   20 IF (FLOAT(NSET(2,N1))/SCALE.GT.1.0) GO TO 40
C
C     ***  FIND MACHINE CENTER WHERE END OF SERVICE EVENT
C     ***  IS GOING NEXT
C
      MM=FLOAT(NSET(13,N1))/SCALE+.000001
      IF (MM.NE.NMM) GO TO 40
C
C     ***  CHECK IF EVENT IS GOING TO HAVE AN EFFECT ON WORK
C     ***  AT NEXT QUEUE WITH RESPECT TO THE JOB WE ARE
```

```
C      ***  CONSIDERING
C
       IF (FLOAT(NSET(1,N1))/SCALE-(TM)) 30,40,40
   30  TWK=TWK+FLOAT(NSET(14,N1))/SCALE
   40  N1=NSET(MX,N1)
       IF (N1.NE.7777) GO TO 20
C
C      ***  CHECK IF MEASURE IS OPTIMAL SO FAR
C
       IF (TWK.LT.XX) MBEST=MXT
       IF (TWK.LT.XX) XX=TWK
C
C      ***  GET NEXT JOB IN QUEUE FOR THIS MACHINE
C
   50  MXT=NSET(MX,MXT)
       IF (MXT.NE.7777) GO TO 10
       RETURN
       END
```

APPENDIX C

FORTRAN IV LISTING OF THE RANDOM NUMBER GENERATOR TEST PROGRAM

```
@ED,U RANDT.MAIN
ED 13.00-05/05-16:15-(13,14)
EDIT
0:P 1 124
C      *** PROGRAM MAIN
       DIMENSION NMAT(11,11),NRUN(10),NLIN(10),ER(10)
       READ (5,10) ISEED,INUM
   10  FORMAT (I6,2X,I5)
       DO 20 I=1,10
       NRUN(I)=0
       NLIN(I)=0
       DO 20 J=1,10
   20  NMAT(I,J)=0
       NRUNTO=0
       NRUNLE=0
       KA=ISEED
       KB=5**7
       KD=KA
       DO 100 I=1, INUM
       KA=KD
       KC=KA*KB
       KD=MOD(KC,2**17)
       D=KD
       X=D/(2.0**17)
       XX=X*10.
       IX=XX
       IX=IX+1
       IF (I.EQ.1) GO TO 99
       NMAT(IX1,IX)=NMAT(IX1,IX) +1

       IIX=X*1000000.
       IIX1=X1*1000000.
       IF (IIX.EQ.IIX1) GO TO 900
       IF (IDIP.EQ.1.AND.X.GT.X1) GO TO 30
       IF (IDIR.EQ.2.AND.X.LT.X1) GO TO 30
       NRUNTO=NRUNTO+1
       IF (NRUNLE.GE.5) NRUNLE=5
       NRUN(NRUNLE)=NRUN(NRUNLE)+1
       NRUNLE=0
   30  IF (X.GT.X1) IDIR=1
       IF (X.LT.X1) IDIR=2
       NRUNLE=NRUNLE+1
   99  X1=X
       IX1=IX
  100  CONTINUE
       NRUNTO=NRUNTO+1
       IF (NRUNLE.GE.5) NRUNLE=5
       NRUN(NRUNLE)=NRUN(NRUNLE) +1
```

```
      WRITE(6,201)
  201 FORMAT (IH1,24HRANDOM NO GENERATOR TEST///)
      WRITE (6,203)
  203 FORMAT (IH ,32HFREQUENCY COUNTS AT .1 INTERVALS)
      DO 207 J=1,10
      DO 205 I=1,10
  205 NLIN(J)=NLIN(J)+NMAT(I,J)
      WRITE(6,209) J,NLIN(J)
  207 CONTINUE
  209 FORMAT (1H ,3X,I2,3X,I6)
      WRITE (6,221)
  221 FORMAT (//1H , 13HMATRIX COUNTS)
      DO 223 I=1,10
  223 WRITE (6,225) (NMAT(I,J),J=1,10)
  225 FORMAT (1H ,10I6)
      WRITE (6,231)
  231 FORMAT (//1H ,10HRUN COUNTS)
      WRITE (6,233) (NRUN(I),I=1,5)
  233 FORMAT (10I6)
      WRITE (6,235) NRUNTO
  235 FORMAT (/1H ,20HTOTAL NUMBER OF RUNS,5X,I6)
C     *** CHI-SQUARE GOODNESS OF FIT
      CHISQ=0.0
      EXP=FLOAT(INUM-1)/10.
      DO 301 I=1,10
      Y=(NLIN(I)-EXP)**2
      Y=Y/EXP
  301 CHISQ=CHISQ+Y
      WRITE (6,303) CHISQ
  303 FORMAT(//1H ,33HCHISQUARE GOODNESS OF FIT(9DOF)  ,F9.3)
C     *** KOLMO GOROV-SMIRNOV GOODNESS OF FIT
      D=0.0
      Y1=0.0
      EXP1=0.0
      EXP=EXP/FLOAT(INUM-1)


      DO 321 I=1,10
      Y=FLOAT(NLIN(I))/FLOAT(INUM-1)
      Y1=Y1+Y
      EXP1=EXP1+EXP
      DIF=ABS(Y1-EXP1)
      IF (DIF.GT.D) D=DIF
  321 CONTINUE
      WRITE(6,323) D
  323 FORMAT (//IH ,34HKOLM-SMIRNOV GOODNESS OF FIT(9DOF),F9.3)
C     *** SERIAL TEST
```

```
      CHISQ=0.0
      EXP=FLOAT(INUM-1)/100.0
      DO 331 I=1,10
      DO 331 J=1,10
      Y=(NMAT(I,J)-EXP)**2
      Y=Y/EXP
  331 CHISQ=CHISQ+Y
      WRITE (6,333) CHISQ
  333 FORMAT (///1H ,27HCHI-SQ SERIAL TEST(99DOF)    ,F9.3)
C     *** RUN TESTS(TOTAL RUNS)

      RNUM=NRUNTO-(2.0*FLOAT(INUM)-1.0)/3.0
      RDEN=(16.0*FLOAT(INUM)-29.0/90.0
      RDEN=RDEN**0.5
      Z=RNUM/RDEN
      WRITE (6,341) Z
  341 FORMAT(//1H ,28HTOTAL RUN NORMAL STATISTIC    ,F9.4)
C     *** RUN TESTS(RUN LENGTHS)
      FINUM=FLOAT(INUM)
      ERT=((2.*FINUM)-1.)/3.
      ER(1)=((5.*FINUM)+1.)/12.
      ER(2)=((11.*FINUM)-14.)/60.


      ER(3)=((19.*FINUM)-47.)/360.
      ER(4)=((29.*FINUM)-105.)/2520.
      ER(5)=ERT-ER(1)-ER(2)-ER(3)-ER(4)


      CHISQ=0.0
      DO 351 I=1,5
      FNRUN=FLOAT(NRUN(I))
      Y=(FNRUN-ER(I))**2
      Y=Y/ER(I)
  351 CHISQ=CHISQ+Y
      WRITE (6,353) CHISQ
  353 FORMAT (//1H ,26HCHISQ   RUN LENGTHS(4DOF)    ,F9.3)
      GO TO 951
  900 WRITE (6,901) I
  901 FORMAT(1H ,34HERROR CONDITION, 2 EQUAL NO. ITER  ,I5)
  951 CONTINUE
      END
```

RANDOM NO GENERATOR TEST
FREQUENCY COUNTS AT .1 INTERVALS

1	983
2	1011
3	978
4	994
5	1049
6	1032
7	976
8	973
9	1017
10	986

MATRIX COUNTS

87	100	89	102	110	107	104	98	102	83
96	102	111	102	89	100	100	98	110	103
90	109	95	85	107	104	89	108	89	102
94	96	106	101	106	111	95	89	94	102
114	113	104	102	109	95	101	105	109	98
104	109	95	109	111	101	103	101	99	100
92	87	102	87	102	96	102	90	103	115
98	91	85	99	106	113	95	87	106	93
101	101	94	106	102	105	98	105	104	101
107	103	97	101	107	100	89	92	101	89

RUN COUNTS

4147 1872 503 121 22

TOTAL NUMBER OF RUNS 6666

CHISQUARE GOODNESS OF FIT(9DOF) 6.146

KOLM-SMIRNOV GOODNESS OF FIT (9DOF) .005

CHI-SQ SERIAL TEST(99DOF) 54.435

TOTAL RUN NORMAL STATISTIC -.0079

CHISQ RUN LENGTHS(4DOF) 2.515
END 1469 MLSEC

APPENDIX D

DESCRIPTION OF ATTRIBUTES, EVENTS, AND OPTIONAL VARIABLES IN THE GASP PROGRAM

Events

1- End of service (ENDSV)

2- Arrival of a job to the system (ARIVL)

3- Completion of a scheduling period (COLL)

4- Beginning of the simulation (START)

Other events which are not called by EVNTS are:

End of run in period (CLEAR)

End of simulation (ENSIM)

User Subroutines

MAIN

EVNTS

ENDSV This is used when a job operation has been completed on a machine.

ARIVL Called when a new arrival comes into the system.

COLL Collects statistics on machine and shop utilization at the end of every scheduling period.

START Called at the beginning of the simulation to preload jobs in the shop.

CLEAR Used to clear statistical areas after the run in period.

DYNAM This is used to calculate priorities for dynamic rules (DSOP).

WKINQ Used with the expected work in next queue rule.

PTJOB Takes an available job from ARIVL, ENDSV, or JOBDEC and moves it to the machine center required.

GENMAT Called by COLL to put the shop loading information in the mathematical programming model when this form of loading is being used.

LPM The linear program model used to decide which jobs should be moved from the job pool into the shop.

JOBDEC Program used to interpret the results of LPM and to call PTJOB as required.

ENSIM End of simulation.

POOLHE Program used to load jobs in the shop with a heuristic algorithm instead of the linear program algorithm.

Function Subprograms

DUED Computes a due date for each incoming job.

MOP Computes the number of job operations for each incoming job.

TIME Computes a processing time for each of the job operations on the routing.

GNARV Computes the time before the next arrival is due.

Files

#1 Events

#2-11 Machine queues (jobs in the queue) for machines #1-10

#12 Jobs in the job pool

Attributes

1- Time the event is going to take place

2- Event code

3- Time at which the job came into the system

4- Due date for the job (including TNOW)

5- Number of operations left

6- Slack time (including TNOW) (for static rules); work remaining (for dynamic rules)

7- Slack or work remaining/operation

8- Time at which job arrived at its current queue

9- Total work time

10- Total number of operations

11- First or actual operation (machine) number

12- Time required for the operation in attribute 11, that is, for the operation in the machine where the job is presently located.

13-26 Similar to #11,12

27-29 Not used

30- Job number

31- Not used

32- Amount of work already performed on this job (hours)

33- Time at which the job came out of the pool

Statistics Collected

COLCT (SUMA array) : 64 statistics

Nº		Var.
1-10	UT	Percent of time busy in a period for machine I
11	TISYS	Time spent in the system
12	TLATE	Time value of job lateness
13	TARDY	Time value of job tardiness
14	ATS	Average time busy in a period per machine
15	DDD	Time value (absolute) of job lateness
16-25	WT	Waiting time for jobs at queue of machine "I"
26-30	TYSYS	Time spent in the system for jobs with 4-8 operations
31-40	DBAL	Deviation from balance for machine J

41	DBALT	Deviation from balance for entire shop
42		Time spent in system w/o counting pool time
43-47		Time spent in the system w/o counting the pool time for jobs with 4-8 operations
48		Time spent in the pool
49-53		Time spent in the pool for jobs with 4-8 operations
54-63	DBALQ	Deviation from balance in queue for machine "J"
64	DBALQT	Deviation from queue balance (all machines)
65		Operation run time
66,67		N° in pool before/after loading
68,69		N° of jobs loaded in shop thru special features on PTJOB, ENDSV
70,71		PWB, PQB

TMST (SSUMA array) : 14 statistics

1-10	BUS(I)	Amount of time machine "I" has been busy
11	XWKSY	Amount of work in hours in the shop
12	XYSY	Amount of work in number of jobs in the shop
13	XOPS	Number of operations performed for jobs in the shop
14	XWKS	Amount of work already done for jobs in the shop

HISTO : 16 statistics

1	TLATE	Time value of job lateness
2	XP	Number of periods late
3	ATS	Average time busy in a period per machine
4	AT	Average percent of load arrived/machine
5-14	D	Interarrival times for machine I
15		Interarrival times to the job pool
16		Time jobs spend in the pool

APPENDIX E

DESCRIPTION OF NON-GASP VARIABLES

Description of Non-GASP Variables

ARATE	The rate of job arrivals
AVG(I)	Average percent utilization for machine i
BUS(I)	Status variable for each machine: 0 = idle, 1 = busy
LOC(I)	Column location in NSET of job "I"
MNEXT	The machine to which a current job will next proceed
NEN	Number of jobs entering the system
NLV	Number of jobs leaving the system
NM	Number of machines in the shop
NRSET	Number of runs in periods for the simulation
NTOTPD	Number of periods the simulation is to run
NTPDS	Number of elapsed periods in the simulation
PLEN	Length of one scheduling period
SEED	Random number seed to be used
TISYS	Time spent in the system by an existing job
WBM(I)	Variable used to maintain MWB (machine work balance) statistics for machine "I"
WWW(I)	Variable used to maintain utilization statistics for machine "I"
X(I,J)	Input transition matrix for job routing
XYSYS	Number of jobs in the shop
XWKSY	Amount of work (in hours) in the shop
I Type	Not used
NHELD	Not used

WB(I)	Hours of work that has arrived to the shop for each machine.
NRULE	Code which indicates queue discipline (dispatching rule) to be used
1-	(Dynamic), Dynamic Slack Rule, DS
2-	(Dynamic), Dynamic Slack per Operation Rule, DSOP
3-	(Dynamic), Expected Work in Next Queue, EWIQ
4-	(Not Dynamic), Shortest Processing Time, SPT
5-	(Not Dynamic), Due Date, DD
6-	(Not Dynamic), First in First Out, FIFO
	(Normal GASP procedure for ranking entries in the file is used to maintain ranking of jobs in machine queues for rules 4-7. Rules 1-3 utilize separate subroutines for computing priorities.) Only Rules #2,3,4,6 were used to obtain detailed simulation results in order to save computer time.
MNOW	The machine number where the current job has just finished
NRST	Used by main to indicate the number of runs in periods. Set by main NRST = NRSET except that if NRSET = 0, then NRST = 9999999. The effect of this is to eliminate the run in period if NRSET = 0.
MAX	Equals the largest job number presently in the system (not the number of jobs, but the job number)
AR(I)	Last time machine "I" had an arrival
NPREL	Number of jobs to be preloaded in the shop
NPREP	Number of jobs to be preloaded in the pool, if using a pool, out of the total in NPREL

NDESL	Switch to indicate whether desired aggregate load per machine is to be read individually or calculated using a factor: 0 = read, 1 = calculated
DESLF	Factor to be used in calculating the desired aggregate load
CAPM(J)	Machine capacity for machine "J" (per scheduling period)
DESL(J)	Desired aggregate load for machine "J" after loading.
NDML	Switch to indicate if desired queue load at each machine is to be read individually or calculated: 0 = read, 1 = calculated
DMLF	Factor to be used in calculating desired queue load
DQL(J)	Desired queue load for machine "J"
QLDAD(J)	Variable used to keep track of work in queue for machine J (to be used by work in next queue dispatching rule)
TIMEF(J)	Factor used to extend the time generated for a machine operation properly
NSTSW	Switch to determine if the job being handled by ARIVL is to be preloaded directly into the shop regardless of any pool arrangements: 0 = handle normally, 1 = preload directly in shop
XOPS	Number of operations performed already for jobs in shop (see TMST)
XWKS	Amount of work performed already for jobs in the shop (see TMST)
DBAL(J)	Deviation from aggregate balance for machine "J" (that is, deviation from desired aggregate load in the shop)
DBALT	Deviation from balance for entire shop

NLDR	Loading rule to be used:
	0 = shop with uncontrolled loading or releasing
	1 = pool with desired aggregate load, LP;
	5 = pool, desired queue load, heuristic
IDUE	method of job due date generation (1,2)
NARR	Code to indicate type of arrival rate
	1 = Poisson arrivals
	2 = Poisson arrivals with mean interarrival times superimposed on a sine curve
SHOPLD(J)	Variable used to keep track of the aggregate work in the shop for machine "J"
DBALQ(J)	Deviation from desired level of work for machine "J" in queue of machine "J"
DBALQT	Sum of $\lvert DBALQ(J) \rvert$ over all machines
A(I,J)	Matrix used by the bounded LP including RHS and two extra rows, one for upper bounds and one for switches
KBV(I)	Variables which give column number of vector in the basis in the LP programs
C(J)	Objective row, including objective value in the RHS column
FACDUD	A factor used to assign different weights to the job due dates. A factor of 0 ignores due dates.
NOROW	Number of rows in the LP program, not counting the boundary and switch rows
NOCOL	Number of columns in the LP program, not counting the RHS
INDEX	Number of artificial variables in the final basis
KAUX(J)	Column number of the artificial variables in the final basis,

if any.

<u>Variables Used Internally in Some Subroutines</u>

<u>MAIN</u>

XXSD A random number used to prime the random number generator

<u>ENDSV</u>

NOP Total number of operations for the job being handled (current job that is just leaving the system)

NP23 NOP+22. Index used to collect statistics on jobs (time spent on system) depending on their number of operations

DDD Absolute value of job lateness

TLATE Job lateness

TARDY Tardiness = 0 if TLATE is ≤ 0, = TLATE if TLATE > 0

XP Integer number of periods late for a job (could be negative if job is early)

JOB Job number for job leaving the system or job number for job entering service

LR,LRM Variables used to set indices for rolling job attributes when an operation has been finished

MBEST Value returned to ENDSV by DYNAM and WKINQ giving the column number of the job with top priority (according to the rule in use) in the machine queue where an end of service just occurred

WT Waiting time for job being placed on machine

<u>COLL</u>

UT Percent of time busy for a machine

TS Time busy for all machines this period

TOT Percent of capacity load arrived for all machines

ATS Average time busy this period per machine

ATOT Average percent of capacity load arrived per machine

<u>PTJOB</u>

INP New arrival indicator: 2 from ENDSV, 1 from ARIVL, 3 from JOBDEC

D Interarrival time

<u>DYNAM</u>

DIF Due date--work remaining

DS Dynamic slack

DSOP Dynamic slack per operation

XX Best dynamic slack value so far

MBEST Column number of best job so far

MNXT Column number of job being considered

<u>WKINQ</u>

MXT Column number of job being considered

TWK Total work content at the machine queue where the job being considered would go next

APPENDIX F

SAMPLE INPUT DATA FOR SIMULATION PROGRAM

****** DATA02 ******

```
@ELT,L SHOPLOADING.DATA02
ELT006-RLIB67-10 05/19-03:05:10
CYCLE (47)
          10   500    50      8.000      900131
           1    02     1      1
        00001
       .5000         2        80.0        16.0
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
        1000  2000  3000  4000  5000  6000  7000  8000  9000  10000
          45    25     1    4.2500     1    0.4000
     1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
        8.00  8.00  8.00  8.00  8.00  8.00  8.00  8.00  8.00  8.00
        IRASTORZA   101  1973   1  01111111101
           6    16    71    14   200    33    12    32   100.00
          20    20    30    30    30    30    30    30    30    30
          30    30    30    30    30    30
           1    12    12    12    12    12    12    12    12    12
          12    12
           1     1     1     1     1     1     1     1     1     1
           1     1
           0     1     0     7     0.000      10.0005909
                 1
```

```
1   0.0    4.0
1   8.0    3.0
1   0.0    2.0
0
0   1   0   7   0.000   10.0005909
-1
1   0.0    4.0
1   8.0    3.0
```

```
1    0.0      2.0

0
```

APPENDIX G

SAMPLE OUTPUT FROM SIMULATION PROGRAM

VARIOUS APPROACHES FOR JOB SHOP LOADING USING DIFFERENT DISPATCHING RULES

LOADING APPROACH NUMBER 1

DISPATCHING RULE NUMBER 4

NUMBER OF MACHINES IN THE SIMULATED SHOP 10

NUMBER OF RUN IN PERIODS 50

NUMBER OF TIME PERIODS SIMULATED AFTER RUN IN 500

LENGTH OF EACH TIME PERIOD 8.00

SPECIAL FEATURES 0000000000

MACHINE	DEVIATION FROM BALANCE AGGREGATE LOAD	QUEUE LOAD
1	6.324	-6.172
2	2.981	-9.382
3	9.486	-5.245
4	5.308	-7.583
5	7.366	-5.661
6	5.994	-6.226
7	3.464	-9.105
8	10.028	-3.961
9	8.159	-4.885
10	9.006	-4.680

DEVIATION FROM BALANCE,AGGREGATE LOAD
AVERAGE 123.554 VARIANCE 2210.457

DEVIATION FROM BALANCE,QUEUE LOAD
AVERAGE 72.148 VARIANCE 693.733

TIME SPENT IN THE SYSTEM
AVERAGE 51.952 VARIANCE 1270.387

TIME SPENT IN THE JOB POOL
AVERAGE 8.970 VARIANCE 201.022

TIME SPENT IN THE SYSTEM W/O POOL TIME
AVERAGE 42.981 VARIANCE 983.242

JOBS IN THE POOL BEFORE LOADING AVG 6.69 STD 4.13

JOBS IN THE POOL AFTER LOADING AVG 2.45 STD 3.26

MACHINE	UTILIZATION	BAL-ANCE MEASURE
1	84.486	4.831
2	84.822	4.756
3	77.819	6.369
4	84.111	4.930
5	80.935	5.296
6	83.869	4.573
7	85.575	4.446
8	77.805	5.949
9	77.869	6.114
10	79.646	5.743

```
MACHINE BALANCE MEASURE =          5.303
SHOP BALANCE MEASURE =          .870
NUMBER OF JOBS ENTERING SHOP =    2118
NUMBER OF JOBS LEAVING SHOP =    2128
AVERAGE JOB LATENESS =        -40.09
AVERAGE LATENESS VARIANCE =     2754.45
AVERAGE JOB TARDINESS =         6.504
AVERAGE TARDINESS VARIANCE =      424.258
AVERAGE SHOP UTILIZATION =        81.694
AVERAGE W.I.P.(IN HOURS OF WORK) =      411.768          120.893
AVERAGE NUMBER OF JOBS IN THE SHOP =        22.744          6.751
W.I.P.(AVERAGE OPERATIONS PERFORMED PER JOB IN THE SHOP).
    AVG=      58.809   STD=      19.300
W.I.P.(AVERAGE HOURS OF WORK DONE FOR JOBS IN THE SHOP).
    AVG=      135.748   STD=      49.945
LENGTH OF SIMULATION RUN WAS
   500   TIME PERIODS (      4000.0 HOURS )
```

MACHINE	AVG INPUT/PD.
1	84.30
2	84.60
3	77.60
4	84.22
5	81.16
6	84.36
7	85.95
8	77.87
9	78.18
10	79.50

OTHER BALANCE MEASURES

MACHINE QUEUE BALANCE INDEX

MACH NO	AVERAGE	QWB	MAXIMUM
1	1.512	2.154	8
2	1.850	3.284	10
3	1.289	2.456	9
4	1.661	3.753	12
5	1.377	2.938	10
6	1.466	2.402	9
7	1.898	3.984	11
8	1.195	1.855	8
9	1.257	2.127	8
10	1.167	1.891	9
ALL	1.458	2.681	12

PERIOD WORK BALANCE INDEX PWB = 4.484

PERIOD QUEUE BALANCE INDEX PQB = 11.59

GASP SUMMARY REPORT

SIMULATION PROJECT NO. 1 BY IRASTORZA

DATE 1/ 0/ 1973 RUN NUMBER 3

GENERATED DATA

CODE	MEAN	STD.DEV.	MIN.	MAX.	OBS.
1	64.4865	27.4736	.0000	100.0000	500
2	64.8222	27.2618	.0000	100.0000	500
3	77.6187	31.5957	.0000	100.0000	500
4	64.1109	27.7559	.0000	100.0000	500
5	80.9352	28.7667	.0000	100.0000	500
6	63.0690	26.7299	.0000	100.0000	500
7	85.5755	26.3558	.0000	100.0000	500
8	77.8052	30.4875	.0000	100.0000	500
9	77.8690	30.9089	.0000	100.0000	500
10	79.6457	29.9557	.0000	100.0000	500
11	51.9517	35.6425	7.6400	410.7900	2128
12	-40.0948	52.4829	-141.6700	324.2900	2128
13	6.5044	20.5975	.0000	324.2900	2128
14	6.5355	.9326	3.2960	8.0000	500
15	53.1035	39.2618	.0200	324.2900	2128
16	4.7093	8.6204	.0000	129.3200	1296
17	5.8249	16.4565	.0000	335.7900	1278
18	4.3525	9.7138	.0000	123.6900	1205
19	5.1160	13.2996	.0000	222.0500	1300
20	4.3587	16.6104	.0000	195.8700	1255
21	4.4137	8.7560	.0000	99.7700	1327
22	5.6964	16.4594	.0000	275.4100	1334
23	3.5789	6.3218	.0000	88.7600	1238
24	4.0018	8.9688	.0000	150.7900	1257
25	3.8839	7.5595	.0000	116.4900	1210
26	38.4792	30.0286	7.6400	210.9200	320
27	44.2673	30.8127	12.4200	338.9700	441
28	51.3814	35.3627	11.8000	314.4700	632
29	60.7213	37.2280	16.0800	318.7100	424
30	65.9152	37.8368	20.1800	410.7900	311
31	6.3244	11.2274	-20.2188	31.6312	500
32	2.9805	15.0919	-38.4141	33.2459	500
33	9.4859	12.4480	-29.2938	30.2462	500
34	5.3082	15.3112	-53.2075	30.9825	500
35	7.3661	13.5930	-30.6267	31.0432	500
36	5.9944	12.3046	-26.8008	30.5592	500
37	3.4640	16.5782	-64.2508	32.0392	500
38	10.0277	11.1717	-18.2562	32.3739	500
39	8.1595	12.9796	-24.6361	32.9739	500
40	9.0062	11.8658	-29.5625	30.9675	500
41	123.5543	47.0155	43.8721	268.0026	500
42	42.9613	31.3567	6.5200	398.5400	2128
43	29.0655	23.6424	6.5200	183.2700	320
44	35.2964	24.9207	8.0900	261.2200	441
45	42.7538	31.5323	10.4300	307.8000	632
46	51.0234	32.7331	9.2300	309.7100	424

47	57.6965	34.8429	18.6300	399.5400	311
48	8.9704	14.1782	.0800	131.9400	2128
49	9.4137	15.1579	.0800	116.4300	320
50	8.9709	14.0805	.1400	110.9700	441
51	8.6276	14.2230	.1200	131.9400	632
52	9.6979	14.6323	.2200	110.7500	424
53	8.2187	12.1553	.1700	89.2500	311
54	-6.1717	7.2350	-38.1669	3.1931	500
55	-9.3825	11.1457	-50.3328	3.1672	500
56	-5.2451	7.8251	-46.4900	3.1900	500
57	-7.5835	11.7105	-65.4000	3.0900	500
58	-5.6611	8.6307	-38.3200	3.1600	500
59	-6.2261	7.8923	-32.2929	3.1471	500
60	-9.1054	11.9281	-62.8400	3.1600	500
61	-3.9605	6.2822	-27.0942	3.1158	500
62	-4.8851	7.3637	-38.1018	3.1883	500
63	-4.6796	7.1631	-3[illegible].3594	3.1906	500
64	72.1481	26.3388	11.6720	136.4938	500
65	2.5832	2.0659	1.0000	9.0000	12662
66	6.6920	4.1322	.0000	18.0000	500
67	2.4540	3.2553	.0000	13.0000	500
68	NO VALUES RECORDED				
69	NO VALUES RECORDED				
70	4.4836	2.7735	.0000	12.2429	500
71	11.5922	215.2726	.0885	4815.5137	500

TIME GENERATED DATA

CODE	MEAN	STD.DEV.	MIN.	MAX.	TOTAL TIME
1	.8449	-.8449	.0000	1.0000	4000.0000
2	.8482	-.8482	.0000	1.0000	4000.0000
3	.7782	-.7782	.0000	1.0000	4000.0000
4	.8411	-.8411	.0000	1.0000	4000.0000
5	.8094	-.8094	.0000	1.0000	4000.0000
6	.8387	-.8387	.0000	1.0000	4000.0000
7	.8558	-.8558	.0000	1.0000	4000.0000
8	.7781	-.7781	.0000	1.0000	4000.0090
9	.7787	-.7787	.0000	1.0000	4000.0000
10	.7965	-.7965	.0000	1.0000	4000.0000
11	411.7676	120.8925	48.5250	696.0349	4000.0000
12	22.7440	6.7513	3.0000	44.0000	4000.0000
13	58.8087	19.2998	7.0000	133.0000	4000.0000
14	135.7477	49.9454	-14.1644	327.5462	4000.0000

GENERATED FREQUENCY DISTRIBUTIONS

CODE	HISTOGRAMS									
1	1491	34	30	38	29	34	22	29	26	30	26
	42	22	15	17	16	11	21	17	14	18	146
2	0	0	0	0	0	0	0	0	0	0	0
	164	82	59	54	22	16	15	11	15	71683	
3	0	0	0	0	0	0	1	5	10	17	45
	48	80	107	118	63	6	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	

APPENDIX H

RESULTS FROM RANDOM NUMBER GENERATOR TEST SEEDS

Table 21. Results of Tests on Random Number Generator

	Goodness of fit (X^2)	Goodness of fit Kolm-Smirnov	Serial correlation X^2	Total number of runs (normal)	Number of runs of each length (X^2)
CRITICAL VALUES	16.92	.014	123.2	1.96	9.49
TEST RESULTS					
SEEDS					
1) 100933	7.59	.004	64.7	.34	5.36
2) 411719	8.12	.005	64.7	.02	1.32
3) 297449	9.11	.008	83.9	.02	3.86
4) 349387	6.85	.005	61.9	.07	3.07
5) 281923	7.22	.004	64.9	.43	1.18
6) 154231	4.41	.006	64.3	1.11	5.30
7) 329963	7.33	.006	80.2	.85	1.61
8) 900131	7.61	.008	81.8	.24	4.43
9) 392819	2.55	.006	61.1	.19	.48
10) 214753	2.47	.005	61.2	.17	.35
11) 200933	6.15	.005	54.4	.01	2.52
12) 117341	5.87	.005	55.8	.00	2.86

APPENDIX I

SUMMARY RESULTS FROM SIMULATION RUNS

Table 22. Simulation Results

Conditions: No Pool, Results after 100 periods, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	94.6	70.4	65.8	66.1	66.2	83.0	74.4
2. Time spent in the shop	94.6	70.4	65.8	66.1	66.2	83.0	74.4
3. Aggregate deviation from Des. Bal.	234.	151.	149.	150.	153.	162	167.
4. Machine balance measure, MWB	3.81	5.00	6.62	5.19	5.51	4.14	5.05
5. Shop balance measure, SWB	.467	.867	1.443	.973	1.29	.586	.938
6. Queue workload balance, QWB	21.7	14.2	10.6	9.03	8.86	12.04	12.74
7. Period workload balance, PWB	3.62	4.31	5.44	4.32	4.28	3.69	4.28
8. Period queue balance, PQB	903.	198.	56.4	141.	110.	157.	261.
9. Average queue size	4.30	2.99	2.42	2.50	2.69	3.63	3.09
10. Average work in process in hours	854	616	505.	559	565	724	637.
11. Average number of jobs in the shop	51.6	37.9	31.5	32.9	34.9	44.7	38.9
12. Average operation done for jobs in shop	106	92.6	86.3	85.8	92.6	104.	94.6
13. Average work hours done for jobs in shop	275	230	215	222	237	261	240.
14. Average lateness	3.98	-17.2	-25.8	-23.8	-24.8	-8.0	-15.9
15. Variance of lateness distribution	773	772	786	934	790	391	741.
16. Average job tardiness	11.2	2.5	.86	1.26	.473	3.00	3.23
17. Average tardiness variance	366	36.	10.4	13.4	4.7	50.6	80.2
18. Number of jobs entering shop	422	416	379	410	405	430	410.
19. Average shop utilization	86.4	80.8	74.3	80.1	80.4	85.3	81.2
20. Average number jobs in pool before loading	0	0	0	0	0	0	0
21. Average number jobs in pool after loading	0	0	0	0	0	0	0

Table 23. Simulation Results

Conditions: No Pool, Results after 400 periods, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	77.2	74.6	75.1	72.5	71.4	71.3	73.7
2. Time spent in the shop	77.2	74.6	75.1	72.5	71.4	71.3	73.7
3. Aggregate deviation from Des. Bal.	170.	148.	164.	162.	153	144	157.
4. Machine balance measure, MWB	4.65	4.34	5.20	5.38	5.22	4.81	4.93
5. Shop balance measure, SWB	.721	.597	1.074	.997	.953	.675	.836
6. Queue workload balance, QWB	15.4	14.8	14.8	14.8	14.1	12.2	14.4
7. Period workload balance, PWB	3.94	3.76	4.19	4.45	4.31	4.14	4.13
8. Period queue balance, PQB	233	58.6	25.1	46.3	37.9	46.3	74.5
9. Average queue size	3.24	3.24	3.17	2.95	2.96	2.94	3.08
10. Average work in process in hours	670.	665	642.	613	616	617	637
11. Average number of jobs in the shop	40.8	40.9	39.8	37.6	37.6	37.6	39.1
12. Average operation done for jobs in shop	96.5	99.9	97.7	91.9	95.2	95.2	96.1
13. Average work hours done for jobs in shop	234	241.	240.	224	234	231	234.
14. Average lateness	-13.4	-15.3	-18.0	-19.9	-20.7	-19.4	-17.8
15. Variance of lateness distribution	891	613	762	843	830	749	781
16. Average job tardiness	4.03	1.99	1.82	1.64	1.36	1.50	2.06
17. Average tardiness variance	124.	25.1	23.6	21.7	17.1	21.5	38.8
18. Number of jobs entering shop	1704	1745	1691	1672	1667	1695	1696
19. Average shop utilization	83.9	84.3	81.7	80.4	80.9	82.6	82.3
20. Average number jobs in pool before loading	0	0	0	0	0	0	0
21. Average number jobs in pool after loading	0	0	0	0	0	0	0

Table 24. Simulation Results

Conditions: No Pool, Results after 500 periods, DSOP; Set 1

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	76.	71.	72.	72.	72.	72.	72.5
2. Time spent in the shop	76.	71.	72.	72.	72.	72.	72.5
3. Aggregate deviation from Des. Bal.	168.	145.	159.	151.	149.	156.	155.
4. Machine balance measure, MWB	4.73	4.59	5.24	5.11	4.89	5.18	4.96
5. Shop balance measure, SWB	.753	.715	1.036	.892	.646	.954	.833
6. Queue workload balance, QWB	16.2	13.3	14.1	14.6	----	14.0	14.4
7. Period workload balance, PWB	4.00	3.89	4.26	4.28	----	4.28	4.14
8. Period queue balance, PQB	190.	47.8	21.7	33.5	----	38.8	66.4
9. Average queue size	3.25	3.0	3.0	3.0	----	3.0	3.1
10. Average work in process in hours	671.	627	625	623	635	619	633
11. Average number of jobs in the shop	40.	38.2	38.5	38.2	38.	37.9	38.5
12. Average operation done for jobs in shop	96.8	95.1	95.6	96.5	95.9	94.3	95.7
13. Average work hours done for jobs in shop	232.	226	231	233	230.	228	230.
14. Average lateness	-13.	-18.	-19	-19	-17.	-20.	-17.7
15. Variance of lateness distribution	876.	735	784	790	815	761	794.
16. Average job tardiness	4.	1.7	1.6	1.6	2.6	1.4	2.2
17. Average tardiness variance	116.	21	21	21	59.	19.	42.8
18. Number of jobs entering shop	2132	2132	2113	2110	2114	2109	2118
19. Average shop utilization	83.4	83.2	81.5	81.2	82.4	81.4	82.2
20. Average number jobs in pool before loading	0	0	0	0	0	0	0
21. Average number jobs in pool after loading	0	0	0	0	0	0	0

Table 25. Simulation Results

Conditions: No Pool, Results after 500 periods, DSOP; Set 2

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	72.	69.	72.	72.	72	73.	71.7
2. Time spent in the shop	72.	69.	72.	72.	72.	73.	71.7
3. Aggregate deviation from Des. Bal.	158.	145.	152.	153.	143.	148.	150.
4. Machine balance measure, MWB	5.17	5.20	5.15	5.27	4.97	4.93	5.12
5. Shop balance measure, SWB	1.071	.866	.992	.945	.683	.652	.868
6. Queue workload balance, QWB	13.6	13.6	13.5	----	----	----	13.6
7. Period workload balance, PWB	4.15	4.38	4.21	----	----	----	4.25
8. Period queue balance, PQB	72.6	----	74.0	----	----	----	----
9. Average queue size	3.00	2.8	3.0	----	----	----	2.9
10. Average work in process in hours	621	598	621	616	625	638	620.
11. Average number of jobs in the shop	38.	36.5	38.2	37.8	37.9	38.7	37.9
12. Average operation done for jobs in shop	95.5	94.2	95.7	94.1	94.6	94.1	94.7
13. Average work hours done for jobs in shop	232	288	232	226	227	225	238
14. Average lateness	-20.	-23.	-19.	-20.	-18.	-16.	-19.3
15. Variance of lateness distribution	734	810	729	798	768	768	768
16. Average job tardiness	1.4	1.1	1.4	1.4	2.0	2.4	1.62
17. Average tardiness variance	18.	13.	17.	17.	32	42	23.2
18. Number of jobs entering shop	2113	2111	2113	2100	2117	2113	2111
19. Average shop utilization	81.5	81.2	81.7	81.1	82.3	82.4	81.7
20. Average number jobs in pool before loading	0	0	0	0	0	0	0
21. Average number jobs in pool after loading	0	0	0	0	0	0	0

Table 26. Simulation Results with One Seed

Conditions: Pool, Special loading approach 10101, DSOP, Seed 411719, Various DESLF Values

	DESLF	4.25	3.50	3.00	2.50	2.25		
	Run Number	1	2	3	4	5	6	Avg.
1.	Time spent in the system	84.9	93.5	96.7	102.	109.		
2.	Time spent in the shop	62.7	59.8	57.6	56.1	57.5		
3.	Aggregate deviation from Des. Bal.	97.8	108	112	140.	169.		
4.	Machine balance measure, MWB	4.94	5.02	4.84	4.83	4.90		
5.	Shop balance measure, SWB	.721	.704	.571	.499	.551		
6.	Queue workload balance, QWB	9.4	8.36	7.30	7.36	8.36		
7.	Period workload balance, PWB	4.38	4.37	4.32	4.38	4.40		
8.	Period queue balance, PQB	41.6	34.3	24.3	23.4	23.7		
9.	Average queue size	2.50	2.34	2.22	2.14	2.21		
10.	Average work in process in hours	535	512	493	485	498		
11.	Average number of jobs in the shop	33.2	31.5	30.3	29.5	30.2		
12.	Average operation done for jobs in shop	81.3	73.5	71.4	66.9	65.5		
13.	Average work hours done for jobs in shop	194	173	166	155	152		
14.	Average lateness	-7.21	1.37	4.44	9.53	16.7		
15.	Variance of lateness distribution	1045.	1444	1311	1480	1373		
16.	Average job tardiness	8.14	15.1	16.2	20.0	24.3		
17.	Average tardiness variance	171	406	382	510	537		
18.	Number of jobs entering shop	2113	2113	2113	2113	2113		
19.	Average shop utilization	81.4	81.2	81.2	81.1	80.8		
20.	Average number jobs in pool before loading	13.54	19.7	22.6	25.9	29.0		
21.	Average number jobs in pool after loading	10.00	16.1	19.1	22.5	25.5		

Table 27. Simulation Results with One Seed

Conditions: Pool, Special loading approach 01101, DSOP, Seed 411719, Various DESLF Values

DESLF	4.25	2.50	2.00	1.50			
Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	75.0	74.5	76.9	79.8			
2. Time spent in the shop	63.0	54.1	53.9	52.8			
3. Aggregate deviation from Des. Bal.	111.	97.4	117.	139.			
4. Machine balance measure, MWB	5.28	5.12	5.09	4.97			
5. Shop balance measure, SWB	.972	.849	.721	.755			
6. Queue workload balance, QWB	10.4	7.50	8.21	7.26			
7. Period workload balance, PWB	4.36	4.33	4.42	4.27			
8. Period queue balance, PQB	49.8	35.8	27.7	22.0			
9. Average queue size	2.52	2.05	2.04	1.98			
10. Average work in process in hours	540.	467	462	453			
11. Average number of jobs in the shop	33.3	28.6	28.5	27.9			
12. Average operation done for jobs in shop	87.5	79.4	77.4	76.1			
13. Average work hours done for jobs in shop	211.	187	180	172			
14. Average lateness	-17.2	-17.6	-15.2	-12.4			
15. Variance of lateness distribution	879	1135	1226	1198			
16. Average job tardiness	3.01	4.38	6.13	6.81			
17. Average tardiness variance	46.9	69.4	102	103.			
18. Number of jobs entering shop	2113	2113	2113	2113			
19. Average shop utilization	81.6	81.6	81.4	81.7			
20. Average number jobs in pool before loading	7.98	12.2	13.4	15.4			
21. Average number jobs in pool after loading	4.84	9.8	11.4	13.7			

Table 28. Simulation Results with One Seed

Conditions: Pool, Special loading approach 00011, DSOP, Seed 411719, Various DESLF Values

DESLF		4.25	1.50 (11101)	2.50 (11101)			
Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system		78.8	74.7	71.8			
2. Time spent in the shop		63.4	51.2	54.2			
3. Aggregate deviation from Des. Bal.		103.	126	97.4			
4. Machine balance measure, MWB		5.11	4.76	4.88			
5. Shop balance measure, SWB		.796	.666	.821			
6. Queue workload balance, QWB		9.32	6.59	7.72			
7. Period workload balance, PWB		4.37	4.14	4.11			
8. Period queue balance, PQB		38.1	29.1	33.2			
9. Average queue size		2.54	1.90	2.05			
10. Average work in process in hours		539	446	466			
11. Average number of jobs in the shop		33.5	27.1	28.7			
12. Average operation done for jobs in shop		86.2	76.8	80.8			
13. Average work hours done for jobs in shop		205	182	190			
14. Average lateness		-13.3	-17.4	-20.3			
15. Variance of lateness distribution		810	1235	1124			
16. Average job tardiness		3.95	5.22	3.54			
17. Average tardiness variance		57.9	77.1	52.0			
18. Number of jobs entering shop		2113	2113	2113			
19. Average shop utilization		81.6	81.5	81.6			
20. Average number jobs in pool before loading		10.3	13.5	10.5			
21. Average number jobs in pool after loading		6.5	12.0	8.5			

Table 29. Simulation Results

Conditions: Job arrival distribution with static mean, No pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	74.9	68.9	69.0	69.1	70.1		70.4
2. Time spent in the shop	74.9	68.9	69.0	69.1	70.1		70.4
3. Aggregate deviation from Des. Bal.	158.	136.	122.	136.	140.		138.
4. Machine balance measure, MWB	4.55	4.99	4.62	5.03	4.94		4.83
5. Shop balance measure, SWB	.477	.689	.507	.672	.631		.595
6. Queue workload balance, QWB	17.1	12.5	12.6	12.3	13.1		13.5
7. Period workload balance, PWB	4.09	4.35	4.13	4.41	4.36		4.27
8. Period queue balance, PQB	255.	130.	80.5	32.3	49.1		109.
9. Average queue size	3.12	2.81	2.83	2.83	2.88		2.89
10. Average work in process in hours	652	592	602	595	602.		609.
11. Average number of jobs in the shop	39.5	36.3	36.6	36.4	36.9		37.1
12. Average operation done for jobs in shop	95.1	94.4	94.9	94.5	94.3		94.6
13. Average work hours done for jobs in shop	227	228	227	228	227		227.
14. Average lateness	-15.3	-23.2	-21.3	-23.0	-22.1		-21.0
15. Variance of lateness distribution	1170	834	804	842	822		894.
16. Average job tardiness	5.05	1.18	1.46	1.26	1.29		2.05
17. Average tardiness variance	202	1.18	1.46	1.26	1.29		54.6
18. Number of jobs entering shop	2126	2115	2127	2112	2112		2118
19. Average shop utilization	83.2	81.6	83.0	81.5	81.8		82.2
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 30. Simulation Results

Conditions: Pool, Loading heuristics, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	88.7	78.2	80.6	80.1	77.0		80.9
2. Time spent in the shop	70.6	66.9	68.0	67.2	66.6		67.9
3. Aggregate deviation from Des. Bal.	138	129	125	133.	125.		130.
4. Machine balance measure, MWB	4.77	5.32	4.80	5.28	5.28		5.09
5. Shop balance measure, SWB	.729	.925	.678	.941	.874		.830
6. Queue workload balance, QWB	12.9	10.8	11.0	10.9	11.4		11.4
7. Period workload balance, PWB	4.07	4.45	4.14	4.39	4.46		4.30
8. Period queue balance, PQB	106	44.2	28.6	16.9	22.1		43.6
9. Average queue size	2.96	2.72	2.81	2.75	2.70		2.79
10. Average work in process in hours	623	575	595	577	570		588.
11. Average number of jobs in the shop	37.9	35.4	36.4	35.6	35.1		36.1
12. Average operation done for jobs in shop	86.5	87.4	88.0	86.6	88.5		87.4
13. Average work hours done for jobs in shop	208	212	210	207	213		210.
14. Average lateness	-1.43	-13.9	-9.3	-11.9	-15.3		-10.4
15. Variance of lateness distribution	1053	821	777	828	791.		854.
16. Average job tardiness	11.1	3.6	5.0	4.5	3.0		5.44
17. Average tardiness variance	254.	58.0	79.8	73.0	47.4		102.
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.5	81.6	83.2	81.5	81.2		82.2
20. Average number jobs in pool before loading	11.7	8.2	8.8	8.9	7.6		9.04
21. Average number jobs in pool after loading	7.4	3.9	4.6	4.7	3.3		4.78

Table 31. Simulation Results

Conditions: Few interactions, No pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	34.9	34.4	35.8	36.3	34.5		35.2
2. Time spent in the shop	34.9	34.4	35.8	36.3	34.5		35.2
3. Aggregate deviation from Des. Bal.	40.8	40.8	42.3	43.4	40.6		41.6
4. Machine balance measure, MWB	5.89	5.42	5.91	5.41	5.71		5.67
5. Shop balance measure, SWB	1.720	1.453	1.700	1.271	1.558		1.540
6. Queue workload balance, QWB	28.6	25.6	31.9	27.6	28.9		28.5
7. Period workload balance, PWB	4.24	4.07	4.23	4.20	4.20		4.19
8. Period queue balance, PQB	340.	106.	66.1	289	27.5		166.
9. Average queue size	4.68	4.76	4.90	5.07	4.77		4.84
10. Average work in process in hours	154	160.	161.	170.	159		161.
11. Average number of jobs in the shop	27.5	28.0	28.6	29.5	28.0		28.3
12. Average operation done for jobs in shop	16.0	16.4	16.1	17.1	16.7		16.5
13. Average work hours done for jobs in shop	36.3	38.9	37.7	40.3	39.2		38.5
14. Average lateness	-46.3	-45.6	-44.4	-43.6	-46.0		-45.2
15. Variance of lateness distribution	1246	1105	1231	1090	1118		1158.
16. Average job tardiness	.35	.11	.30	.14	.13		.21
17. Average tardiness variance	3.19	.72	3.6	.86	.77		1.83
18. Number of jobs entering shop	3182	3257	3211	3268	3255		3235
19. Average shop utilization	81.0	83.3	81.8	83.9	82.6		82.5
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 32. Simulation Results

Conditions: Few interactions, Pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	64.1	62.5	64.3	76.2	66.3		66.7
2. Time spent in the shop	22.1	20.3	24.0	26.3	21.4		22.8
3. Aggregate deviation from Des. Bal.	25.3	23.2	28.5	31.2	24.4		26.5
4. Machine balance measure, MWB	4.72	4.51	5.02	4.81	4.74		4.76
5. Shop balance measure, SWB	.743	.694	.867	.875	.976		.831
6. Queue workload balance, QWB	16.2	11.0	21.0	20.7	12.4		16.3
7. Period workload balance, PWB	4.06	3.92	4.17	4.01	3.81		3.99
8. Period queue balance, PQB	51.8	17.0	15.1	43.3	10.1		27.5
9. Average queue size	2.72	2.44	3.05	3.48	2.63		2.86
10. Average work in process in hours	86.6	82.7	96.9	107.	85.3		91.7
11. Average number of jobs in the shop	17.6	16.3	19.3	21.5	17.3		18.4
12. Average operation done for jobs in shop	7.25	7.10	7.05	8.14	7.32		7.37
13. Average work hours done for jobs in shop	12.8	13.2	12.8	15.8	13.6		13.6
14. Average lateness	-17.0	-17.5	-15.9	-3.9	-14.1		-13.7
15. Variance of lateness distribution	1462	1193	1807	939	1116		1303
16. Average job tardiness	6.29	4.35	8.46	8.55	4.87		6.50
17. Average tardiness variance	156.	80.3	273	207	108		165.
18. Number of jobs entering shop	3182	3257	3211	3268	3255		3235
19. Average shop utilization	81.2	82.5	81.5	83.3	82.3		82.2
20. Average number jobs in pool before loading	36.4	37.7	35.4	44.2	39.8		38.7
21. Average number jobs in pool after loading	30.0	31.2	29.0	37.7	33.3		32.2

Table 33. Simulation Results

Conditions: Asymmetric trans. matrix, No pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	74.3	72.2	73.3	77.2	75.6		74.5
2. Time spent in the shop	74.3	72.2	73.3	77.2	75.6		74.5
3. Aggregate deviation from Des. Bal.	157.	168.	158.	174.	163.		164.
4. Machine balance measure, MWB	4.91	5.37	4.63	5.18	5.06		5.03
5. Shop balance measure, SWB	.755	.987	.565	.887	.838		.806
6. Queue workload balance, QWB	12.8	13.7	13.6	15.4	14.4		14.0
7. Period workload balance, PWB	4.19	4.40	4.09	4.31	4.25		4.25
8. Period queue balance, PQB	52.6	38.7	45.2	47.4	20.7		40.9
9. Average queue size	3.15	3.04	3.11	3.29	3.23		3.16
10. Average work in process in hours	647	626	644	664	655		647.
11. Average number of jobs in the shop	39.8	38.5	39.4	41.1	40.5		39.9
12. Average operation done for jobs in shop	97.2	94.9	97.9	97.4	98.7		97.2
13. Average work hours done for jobs in shop	232	224	235	228	233		230.
14. Average lateness	-16.0	-18.3	-16.9	-13.9	-15.6		-16.1
15. Variance of lateness distribution	672	793	754	713	703		727
16. Average job tardiness	2.1	1.51	2.27	2.51	1.92		2.06
17. Average tardiness variance	33.0	16.3	41.0	56.2	22.6		33.8
18. Number of jobs entering shop	2136	2135	2147	2124	2150		2138
19. Average shop utilization	83.2	81.2	83.3	82.4	82.8		82.6
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 34. Simulation Results

Conditions: Asymmetric transition matrix, Pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	93.9	85.0	84.8	95.9	95.4		91.0
2. Time spent in the shop	64.6	62.2	63.9	64.2	65.1		64.0
3. Aggregate deviation from Des. Bal.	78.3	93.2	85.4	86.1	87.9		86.2
4. Machine balance measure, MWB	4.78	5.44	4.91	5.20	5.24		5.11
5. Shop balance measure, SWB	.384	.771	.489	.566	.519		.546
6. Queue workload balance, QWB	8.40	9.05	9.20	9.11	9.71		9.09
7. Period workload balance, PWB	4.43	4.69	4.45	4.65	4.74		4.59
8. Period queue balance, PQB	27.2	15.1	14.1	19.6	16.4		18.5
9. Average queue size	2.62	2.48	2.58	2.59	2.66		2.59
10. Average work in process in hours	552	524	549	543	556		545
11. Average number of jobs in the shop	34.5	32.9	34.1	34.1	34.8		34.1
12. Average operation done for jobs in shop	77.8	80.1	83.6	77.7	49.7		49.8
13. Average work hours done for jobs in shop	182.	184.	198	178	183		185
14. Average lateness	3.6	-5.6	-5.5	5.04	4.43		.39
15. Variance of lateness distribution	760	1001.	900.	915.	1094.		934
16. Average job tardiness	11.8	8.49	7.92	13.9	14.4		11.3
17. Average tardiness variance	222	197	149	258	403		246
18. Number of jobs entering shop	2136	2135	2147	2124	2150		2138
19. Average shop utilization	83.1	80.8	82.8	81.9	82.0		82.1
20. Average number jobs in pool before loading	17.8	14.3	13.6	19.2	18.6		16.7
21. Average number jobs in pool after loading	13.5	10.1	9.4	14.9	14.3		12.4

Table 35. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, No pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	76.5	72.2	79.1	72.7	72.5		74.6
2. Time spent in the shop	76.5	72.2	79.1	72.7	72.5		74.6
3. Aggregate deviation from Des. Bal.	168.	158.	146.	159.	152.		157.
4. Machine balance measure, MWB	4.73	5.17	4.59	5.24	5.12		4.97
5. Shop balance measure, SWB	.753	1.071	.715	1.036	.892		.893
6. Queue workload balance, QWB	16.2	13.6	13.3	14.1	14.6		14.4
7. Period workload balance, PWB	4.00	4.15	3.89	4.26	4.28		4.12
8. Period queue balance, PQB	190.	72.6	47.8	21.7	33.5		73.1
9. Average queue size	3.25	3.00	2.99	3.03	3.01		3.06
10. Average work in process in hours	671.	621	627	625	624		634.
11. Average number of jobs in the shop	40.8	38.2	38.2	38.5	38.2		38.8
12. Average operation done for jobs in shop	96.8	95.5	95.1	95.7	96.5		95.9
13. Average work hours done for jobs in shop	232	232	226	232	234		231.
14. Average lateness	-13.6	-19.9	-18.4	-19.3	-19.6		-18.2
15. Variance of lateness distribution	876	734	735	784	791		784.
16. Average job tardiness	4.06	1.39	1.71	1.64	1.63		2.01
17. Average tardiness variance	116.	18.4	21.6	21.0	21.6		39.7
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.4	81.5	83.2	81.5	81.2		82.2
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 36. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, No pool, EWIQ

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	65.7	62.2	61.3	61.6	62.8		62.7
2. Time spent in the shop	65.7	62.2	61.3	61.6	62.8		62.7
3. Aggregate deviation from Des. Bal.	155.	155.	148.	155.	151.		153.
4. Machine balance measure, MWB	4.81	5.28	5.00	5.33	5.52		5.19
5. Shop balance measure, SWB	.993	1.392	1.067	1.413	1.588		1.291
6. Queue workload balance, QWB	11.4	9.13	7.86	8.93	8.81		9.03
7. Period workload balance, PWB	3.84	3.94	3.95	3.98	3.99		3.94
8. Period queue balance, PQB	142.	70.1	64.7	12.4	20.1		61.9
9. Average queue size	2.58	2.47	2.39	2.45	2.47		2.47
10. Average work in process in hours	573	549	547	546	552		553.
11. Average number of jobs in the shop	34.1	32.9	32.2	32.6	32.8		32.9
12. Average operation done for jobs in shop	112.	107.	104.	106.	107		107.
13. Average work hours done for jobs in shop	269	256.	251.	255.	260.		258.
14. Average lateness	-24.6	-30.0	-28.8	-30.4	-29.5		-28.6
15. Variance of lateness distribution	5753	3911	4208	3749	3859		4296
16. Average job tardiness	16.7	12.4	13.1	12.1	12.6		13.4
17. Average tardiness variance	2687	1158	1444.	1013.	1108		1482
18. Number of jobs entering shop	2124	2117	2126	2118	2109		2119
19. Average shop utilization	82.8	81.6	82.9	81.4	81.0		81.9
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 37. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, No pool, SPT

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	51.2	47.5	48.8	46.7	48.0		48.4
2. Time spent in the shop	51.2	47.5	48.8	46.7	48.0		48.4
3. Aggregate deviation from Des. Bal.	147.	148.	143.	149.	146.		147.
4. Machine balance measure, MWB	5.05	5.16	4.86	5.17	5.46		5.14
5. Shop balance measure, SWB	.923	.871	.877	.966	1.302		.988
6. Queue workload balance, QWB	4.35	3.82	3.61	3.66	3.61		3.81
7. Period workload balance, PWB	4.15	4.34	4.00	4.26	4.22		4.19
8. Period queue balance, PQB	78.8	27.4	26.5	6.16	12.3		30.2
9. Average queue size	1.82	1.68	1.70	1.65	1.70		1.71
10. Average work in process in hours	497	465	471.	458.	468.		472
11. Average number of jobs in the shop	26.4	25.0	25.3	24.7	25.1		25.3
12. Average operation done for jobs in shop	69.5	64.1	66.0	62.7	64.3		65.3
13. Average work hours done for jobs in shop	161	148.	150.	144.	147.		150.
14. Average lateness	-39.1	-44.6	-41.5	-45.4	-44.1		-42.9
15. Variance of lateness distribution	4783	3189	3434	3086	3158		3530
16. Average job tardiness	9.78	7.07	7.80	6.52	7.10		7.65
17. Average tardiness variance	2052	596.	891	547	585		934
18. Number of jobs entering shop	2118	2118	2124	2118	2104		2116.
19. Average shop utilization	82.8	81.6	82.8	81.5	81.2		82.0
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 38. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, No pool, FCFS

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	74.9	73.7	77.4	69.5	77.6		74.6
2. Time spent in the shop	74.9	73.7	77.4	69.5	77.6		74.6
3. Aggregate deviation from Des. Bal.	152.	148.	152	142.	172.		153.
4. Machine balance measure, MWB	4.71	5.19	4.91	5.01	5.19		5.00
5. Shop balance measure, SWB	.671	.921	.687	.685	.942		.781
6. Queue workload balance, QWB	15.9	14.4	17.0	12.8	16.4		15.3
7. Period workload balance, PWB	4.06	4.31	4.24	4.38	4.31		4.26
8. Period queue balance, PQB	171.	201.	131.	45.1	100.		130.
9. Average queue size	3.16	3.05	3.26	2.86	3.25		3.12
10. Average work in process in hours	668	642	686	611.	676.		657.
11. Average number of jobs in the shop	39.9	38.7	40.9	36.7	40.7		39.4
12. Average operation done for jobs in shop	105	102	108.	95.6	107.		104.
13. Average work hours done for jobs in shop	252	246	259.	231.	258.		249.
14. Average lateness	-15.3	-18.4	-12.7	-22.6	-14.7		-16.7
15. Variance of lateness distribution	3092	2834	3046	2713	3148		2967
16. Average job tardiness	15.7	13.9	16.8	12.0	16.2		14.9
17. Average tardiness variance	785	624.	800.	498	801		702.
18. Number of jobs entering shop	2129	2107	2115	2112	2108		2114.
19. Average shop utilization	83.3	81.6	82.8	81.5	81.6		82.2
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 39. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, Controlled shop with pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	116.	88.4	100.2	89.3	79.2		94.6
2. Time spent in the shop	69.2	62.8	64.7	62.5	61.7		64.2
3. Aggregate deviation from Des. Bal.	126.	85.2	89.1	88.0	90.3		95.7
4. Machine balance measure, MWB	4.68	5.24	4.70	5.34	5.26		5.04
5. Shop balance measure, SWB	.488	.762	.562	.778	.782		.674
6. Queue workload balance, QWB	11.6	8.99	9.54	8.48	8.96		9.51
7. Period workload balance, PWB	4.21	4.52	4.15	4.61	4.53		4.40
8. Period queue balance, PQB	82.5	31.3	21.5	12.9	15.9		32.8
9. Average queue size	2.89	2.51	2.63	2.49	2.44		2.59
10. Average work in process in hours	602.	533	559	529	525		550.
11. Average number of jobs in the shop	37.2	33.2	34.6	33.1	32.6		34.1
12. Average operation done for jobs in shop	71.7	79.0	74.1	77.5	82.5		80.0
13. Average work hours done for jobs in shop	167.	188.	173.	183	196.		181.
14. Average lateness	25.6	-3.7	10.3	-2.8	-12.9		-4.1
15. Variance of lateness distribution	1374	920	956	1026	867		1029
16. Average job tardiness	31.3	9.1	17.8	10.5	4.3		14.6
17. Average tardiness variance	762	188	355	223	68.		319.
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.7	81.5	83.1	81.3	81.2		82.2
20. Average number jobs in pool before loading	27.0	15.7	21.0	16.3	11.3		18.3
21. Average number jobs in pool after loading	22.7	11.4	16.7	12.0	7.1		14.0

Table 40. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, Pool, EWIQ

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	67.3	63.6	63.0	63.2	64.3		64.3
2. Time spent in the shop	59.4	55.2	55.5	55.4	56.3		56.4
3. Aggregate deviation from Des. Bal.	131.	130.	130.	132.	131.		131.
4. Machine balance measure, MWB	4.97	5.32	4.86	5.48	5.56		5.24
5. Shop balance measure, SWB	.998	1.199	.946	1.357	1.601		1.220
6. Queue workload balance, QWB	8.22	6.24	5.93	5.88	6.08		6.47
7. Period workload balance, PWB	4.00	4.17	3.93	4.18	4.01		4.06
8. Period queue balance, PQB	144.	36.0	34.5	8.6	14.7		47.6
9. Average queue size	2.24	2.11	2.13	2.12	2.15		2.15
10. Average work in process in hours	515.	487.	496	486	493		495.
11. Average number of jobs in the shop	30.6	29.3	29.6	29.3	29.6		29.7
12. Average operation done for jobs in shop	100.	93.0	94.9	94.2	95.6		95.5
13. Average work hours done for jobs in shop	240.	222	226	225	229		228.
14. Average lateness	-22.8	-28.5	-27.2	-28.8	-27.9		-27.0
15. Variance of lateness distribution	5188	3194	3514	3002	3246		3629
16. Average job tardiness	15.7	11.2	11.8	10.4	11.6		12.1
17. Average tardiness variance	2409	707	1040	634	742		1106
18. Number of jobs entering shop	2124	2117	2126	2118	2109		2119
19. Average shop utilization	82.8	81.6	83.0	81.5	81.1		82.0
20. Average number jobs in pool before loading	6.08	6.50	5.91	6.18	6.24		6.18
21. Average number jobs in pool after loading	1.84	2.27	1.66	1.95	2.03		1.95

Table 41. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, Pool, SPT

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	54.3	52.0	53.3	52.0	52.5		52.8
2. Time spent in the shop	45.3	43.0	44.5	43.1	43.2		43.8
3. Aggregate deviation from Des. Bal.	123.	124.	122.	124.	119.		122.
4. Machine balance measure, MWB	5.09	5.30	5.03	5.21	5.64		5.25
5. Shop balance measure, SWB	.936	.870	.844	.874	1.336		.972
6. Queue workload balance, QWB	3.26	2.68	2.84	2.63	2.46		2.77
7. Period workload balance, PWB	4.18	4.48	4.20	4.39	4.36		4.32
8. Period queue balance, PQB	34.5	11.6	10.8	3.5	5.3		13.1
9. Average queue size	1.57	1.46	1.54	1.46	1.46		1.50
10. Average work in process in hours	438.	412	427	413	413		421.
11. Average number of jobs in the shop	23.9	22.7	23.6	22.7	22.8		23.1
12. Average operation done for jobs in shop	62.9	58.8	63.0	58.5	58.8		60.4
13. Average work hours done for jobs in shop	144.	136.	143	135.	134.		138.
14. Average lateness	-36.0	-40.1	-37.1	-40.2	-39.6		-38.6
15. Variance of lateness distribution	3941	2754	2973	2810	2790		3054.
16. Average job tardiness	8.75	6.50	7.48	6.69	6.78		7.24
17. Average tardiness variance	1517.	424.	671	446	433		698.
18. Number of jobs entering shop	2118	2118	2124	2118	2104		2116.
19. Average shop utilization	82.8	81.7	82.8	81.5	81.3		82.0
20. Average number jobs in pool before loading	6.64	6.69	6.48	6.71	6.82		6.67
21. Average number jobs in pool after loading	2.39	2.45	2.23	2.48	2.61		2.43

Table 42. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 4.25, SINPER 16, FACDUD 80, DUD generation 1, Pool, FCFS

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	87.1	77.8	84.3	73.7	79.7		80.5
2. Time spent in the shop	63.9	60.2	63.2	59.3	60.1		61.3
3. Aggregate deviation from Des. Bal.	90.3	96.2	93.1	95.5	99.1		94.8
4. Machine balance measure, MWB	4.68	5.17	4.82	5.07	5.20		4.99
5. Shop balance measure, SWB	.588	.745	.544	.636	.906		.684
6. Queue workload balance, QWB	9.55	8.00	10.2	7.94	7.67		8.67
7. Period workload balance, PWB	4.11	4.47	4.29	4.49	4.34		4.34
8. Period queue balance, PQB	79.5	46.6	39.3	18.4	16.1		40.0
9. Average queue size	2.58	2.38	2.53	2.32	2.36		2.43
10. Average work in process in hours	567.	528	561	518	524		540.
11. Average number of jobs in the shop	34.2	32.0	33.6	31.4	31.8		32.6
12. Average operation done for jobs in shop	88.7	82.8	87.7	81.5	82.3		84.6
13. Average work hours done for jobs in shop	210.	197.	208.	194.	196.		201.
14. Average lateness	-3.1	-14.4	-6.0	-18.4	-12.4		-10.9
15. Variance of lateness distribution	2785	2527	2812	2518	2707		2670
16. Average job tardiness	20.2	14.3	18.9	12.7	15.8		16.4
17. Average tardiness variance	747.	483	722	430	529		582.
18. Number of jobs entering shop	2129	2107	2115	2112	2108		2114
19. Average shop utilization	83.4	81.7	82.8	81.5	81.6		82.2
20. Average number jobs in pool before loading	14.4	11.0	13.1	9.6	12.3		12.1
21. Average number jobs in pool after loading	10.2	6.8	8.9	5.4	8.1		7.9

Table 43. Simulation Results with One Seed

Conditions: Showing the effects of changes in DESLF with DSOP, Tight due dates, Sine arrivals, SINPER 16, DUDFCT 80, Seed 100933, Pool, DSOP

DESLF Run Number	5.5 1	5.0 2	4.25 3	3.75 4	3.50 5	6	Avg.
1. Time spent in the system	92.1	101.	116.	124.	126.		
2. Time spent in the shop	72.	69.9	69.2	67.5	66.3		
3. Aggregate deviation from Des. Bal.	106.	100.	126.	146.	157.		
4. Machine balance measure, MWB	4.89	4.71	4.68	4.71	4.86		
5. Shop balance measure, SWB	.698	.604	.488	.475	.486		
6. Queue workload balance, QWB	----	----	----	----	10.6		
7. Period workload balance, PWB	----	----	----	----	4.40		
8. Period queue balance, PQB	----	----	----	----	58.6		
9. Average queue size	----	----	----	----	2.72		
10. Average work in process in hours	627	607	601	591	577		
11. Average number of jobs in the shop	38.6	37.5	37.2	36.2	35.4		
12. Average operation done for jobs in shop	84.3	78.4	71.7	66.8	63.6		
13. Average work hours done for jobs in shop	202	185	167	155	146.		
14. Average lateness	2.1	11.3	25.6	34.0	35.6		
15. Variance of lateness distribution	1002	1233	1374	1158	1044		
16. Average job tardiness	12.6	20.	31.3	37.2	38.0		
17. Average tardiness variance	285	474	762	751	725		
18. Number of jobs entering shop	2132	2132	2132	2132	2132		
19. Average shop utilization	83.3	83.5	83.7	83.3	83.0		
20. Average number jobs in pool before loading	12.8	18.9	27.0	32.4	34.0		
21. Average number jobs in pool after loading	8.5	14.6	22.7	28.1	29.7		

Table 44. Simulation Results with One Seed

Conditions: Showing the effects of changes in DESLF with SPT, Tight due dates, Sine arrivals, SINPER 16, DUDFCT 80, Seed 411719M, Pool, SPT

DESLF	4.25	3.75	3.25	2.75	2.25	2.00	
Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	52.4	53.5	56.0	62.2	78.9	92.5	
2. Time spent in the shop	43.2	41.9	39.7	38.0	38.3	38.8	
3. Aggregate deviation from Des. Bal.	121.	101.	83.8	72.1	84.1	101.	
4. Machine balance measure, MWB	5.52	5.41	5.25	5.09	4.93	4.79	
5. Shop balance measure, SWB	1.19	1.12	.944	.697	.523	.515	
6. Queue workload balance, QWB	2.53	1.35	2.07	1.83	1.98	2.18	
7. Period workload balance, PWB	4.38	4.34	4.36	4.45	4.46	4.33	
8. Period queue balance, PQB	9.40	8.01	6.33	4.19	3.85	3.06	
9. Average queue size	1.47	1.40	1.29	1.19	1.21	1.24	
10. Average work in process in hours	417.	400.	379.	359.	363	371	
11. Average number of jobs in the shop	22.9	22.2	21.0	20.1	20.3	20.5	
12. Average operation done for jobs in shop	59.7	57.8	54.2	52.2	52.9	53.4	
13. Average work hours done for jobs in shop	137.	132	124.	119.	120.	123.	
14. Average lateness	-39.7	-38.6	-36.1	-29.8	-13.3	.52	
15. Variance of lateness distribution	2726	2610	2516	2474	2469	2048	
16. Average job tardiness	6.68	6.57	6.50	7.28	12.0	15.6	
17. Average tardiness variance	374.	330	322	340	564	712	
18. Number of jobs entering shop	2113	2113	2113	2113	2113	2113	
19. Average shop utilization	81.5	81.5	81.5	81.5	81.6	81.5	
20. Average number jobs in pool before loading	6.95	8.24	10.7	14.9	23.6	30.6	
21. Average number jobs in pool after loading	2.72	4.00	6.5	10.7	19.4	26.3	

Table 45. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 2.25, SINPER 16, FACDUD 80, DUD generation 1, No pool, SPT

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	49.2	47.2	47.5	47.6	47.2		47.7
2. Time spent in the shop	49.2	47.2	47.5	47.6	47.2		47.7
3. Aggregate deviation from Des. Bal.	162	156.	149.	159.	154.		156.
4. Machine balance measure, MWB	4.80	5.28	4.84	5.47	5.37		5.15
5. Shop balance measure, SWB	.990	1.279	.958	1.304	1.266		1.16
6. Queue workload balance, QWB	3.89	3.69	3.48	3.62	3.58		3.65
7. Period workload balance, PWB	3.84	4.06	3.90	4.22	4.16		4.04
8. Period queue balance, PQB	42.1	17.0	14.7	4.99	7.93		17.3
9. Average queue size	1.78	1.68	1.70	1.70	1.67		1.71
10. Average work in process in hours	489	466	468	471	462		471
11. Average number of jobs in the shop	26.1	25.0	25.3	25.2	24.8		25.3
12. Average operation done for jobs in shop	68.1	64.6	66.3	65.1	64.5		65.7
13. Average work hours done for jobs in shop	155.	149	150.	149.	148		150
14. Average lateness	-40.8	-44.8	-42.8	-44.5	-44.9		-43.6
15. Variance of lateness distribution	3744	3096	3117	3146	2986		3218
16. Average job tardiness	7.89	6.60	6.68	6.74	6.53		6.89
17. Average tardiness variance	1173	537	644	582	474		682
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.4	81.5	83.1	81.4	81.2		82.1
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 46. Simulation Results - Basic Runs

Conditions: Symmetric transition matrix, Tight due date generation--DESLF 2.25, SINPER 16, FACDUD 80, DUD generation 1, Pool, SPT

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	91.2	78.9	89.5	77.8	79.0		83.3
2. Time spent in the shop	40.8	38.3	39.6	37.1	36.8		38.5
3. Aggregate deviation from Des. Bal.	97.0	84.1	89.5	76.8	74.6		84.4
4. Machine balance measure, MWB	4.44	4.93	4.46	4.91	4.70		4.69
5. Shop balance measure, SWB	.377	.523	.403	.491	.416		.442
6. Queue workload balance, QWB	2.46	1.98	2.21	1.80	1.89		2.07
7. Period workload balance, PWB	4.08	4.46	4.07	4.47	4.33		4.28
8. Period queue balance, PQB	12.2	3.85	4.18	1.88	2.15		4.85
9. Average queue size	1.36	1.21	1.29	1.15	1.13		1.23
10. Average work in process in hours	394	363	380	348	344		366
11. Average number of jobs in the shop	21.9	20.3	21.2	19.6	19.4		20.5
12. Average operation done for jobs in shop	56.9	52.9	55.0	51.0	50.2		53.2
13. Average work hours done for jobs in shop	128.	120.	123	114	113		120
14. Average lateness	1.03	-13.3	-.7	-14.3	-13.1		-6.22
15. Variance of lateness distribution	2159	2469	1981	2235	2239		2217
16. Average job tardiness	16.3	12.0	15.3	10.9	10.9		13.1
17. Average tardiness variance	857	564	649	424	468		592
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.2	81.6	83.2	81.5	81.1		82.1
20. Average number jobs in pool before loading	29.1	23.6	28.7	23.6	24.4		25.9
21. Average number jobs in pool after loading	24.8	19.4	24.4	19.4	20.1		21.6

Table 47. Simulation Results - Basic Runs

Conditions: Pool, DSOP, DESLF 3.5

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	126.	105.	119.	97.7	90.9		108.
2. Time spent in the shop	66.3	61.2	65.0	58.8	57.2		61.7
3. Aggregate deviation from Des. Bal.	157.	121.	145.	104.	73.4		120.
4. Machine balance measure, MWB	4.86	5.35	4.77	5.30	5.25		5.11
5. Shop balance measure, SWB	.486	.665	.472	.693	.540		.571
6. Queue workload balance, QWB	10.6	8.60	9.69	7.51	7.82		8.84
7. Period workload balance, PWB	4.40	4.74	4.32	4.66	4.77		4.58
8. Period queue balance, PQB	58.6	25.9	18.7	11.5	13.1		25.6
9. Average queue size	2.72	2.41	2.64	2.29	2.20		2.45
10. Average work in process in hours	577	522	563	502	483		529.
11. Average number of jobs in the shop	35.4	32.2	34.7	31.0	30.1		32.7
12. Average operation done for jobs in shop	63.6	68.9	65.8	70.6	71.5		68.1
13. Average work hours done for jobs in shop	146	162	150.	165.	167.		158.
14. Average lateness	35.6	13.1	29.3	5.7	-1.4		16.5
15. Variance of lateness distribution	1044	1644	1210	1491	990		1276.
16. Average job tardiness	38.0	23.2	33.7	17.8	10.7		24.7
17. Average tardiness variance	725	685	705	519	209		569
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.0	80.8	82.9	81.2	80.7		81.7
20. Average number jobs in pool before loading	34.0	25.5	31.2	22.7	20.0		26.7
21. Average number jobs in pool after loading	29.7	21.3	27.0	18.5	15.8		22.5

Table 48. Simulation Results

Conditions: Normal conditions, DSOP, No pool, Alternative machine pairs

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	50.9	46.8	47.7	47.5	48.9		48.4
2. Time spent in the shop	50.9	46.8	47.7	47.5	48.9		48.4
3. Aggregate deviation from Des. Bal.	207	370.	294.	265.	264.		280.
4. Machine balance measure, MWB	4.12	4.83	4.28	4.95	4.86		4.61
5. Shop balance measure, SWB	1.234	1.741	1.281	1.857	1.777		1.58
6. Queue workload balance, QWB	7.95	6.49	6.69	6.64	7.44		7.04
7. Period workload balance, PWB	2.90	3.10	3.01	3.12	3.10		3.05
8. Period queue balance, PQB	187.	16.0	26.2	8.23	10.9		49.7
9. Average queue size	1.87	1.66	1.72	1.70	1.76		1.74
10. Average work in process in hours	454	413	429	417.	430.		429.
11. Average number of jobs in the shop	27.0	24.8	25.5	25.1	25.7		25.6
12. Average operation done for jobs in shop	77.0	70.4	72.8	71.4	74.1		73.1
13. Average work hours done for jobs in shop	181	166	170	168.	176.		172.
14. Average lateness	-39.3	-45.1	-42.3	-44.6	-43.3		-42.9
15. Variance of lateness distribution	1277	1325	1253	1305	1280.		1288.
16. Average job tardiness	.741	.424	.333	.376	.409		.457
17. Average tardiness variance	14.0	4.74	2.58	4.12	4.22		5.93
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.7	81.5	83.1	81.4	81.1		82.2
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 49. Simulation Results

Conditions: Normal conditions, DSOP, Pool, Alternative machine pairs

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	98.0	68.8	97.8	79.1	74.2		83.6
2. Time spent in the shop	37.8	32.9	38.2	25.0	33.7		35.5
3. Aggregate deviation from Des. Bal.	260.	377.	176.	269.	220.		260.
4. Machine balance measure, MWB	4.11	4.33	4.00	4.40	4.36		4.24
5. Shop balance measure, SWB	.708	.799	.620	.879	.829		.767
6. Queue workload balance, QWB	3.27	2.16	3.48	2.69	2.45		2.81
7. Period workload balance, PWB	3.41	3.54	3.38	3.54	3.55		3.48
8. Period queue balance, PQB	6.55	2.82	3.74	3.26	2.82		3.84
9. Average queue size	1.19	.931	1.20	1.04	.963		1.06
10. Average work in process in hours	336	289	336	305	291		311.
11. Average number of jobs in the shop	20.2	17.5	20.3	18.6	17.7		18.9
12. Average operation done for jobs in shop	45.3	45.2	45.7	45.1	44.8		45.2
13. Average work hours done for jobs in shop	97.0	98.8	97.9	97.8	97.6		97.8
14. Average lateness	7.83	-23.2	7.56	-13.0	-17.9		-7.76
15. Variance of lateness distribution	847	1471	842	1479	1354		1199.
16. Average job tardiness	15.0	4.13	14.8	7.79	5.07		9.36
17. Average tardiness variance	206	63.7	215	146	88.4		143.8
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.1	81.7	83.0	81.5	80.8		82.0
20. Average number jobs in pool before loading	34.3	21.1	34.0	25.4	23.4		27.6
21. Average number jobs in pool after loading	30.0	16.8	29.7	21.2	19.2		23.4

Table 50. Simulation Results

Conditions: Symmetric transition matrix, Tight due date generation (method 1), DESLF 4.25, SINPER 16, FACDUD 80, Pool loading, Modified GENMAT to force jobs in the shop 16 hours before required by job content, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	108	82.2	99.1	83.2	80.9		90.7
2. Time spent in the shop	70.3	62.9	66.4	62.1	63.7		65.2
3. Aggregate deviation from Des. Bal.	126	93.6	104.	95.7	93.6		102.6
4. Machine balance measure, MWB	4.63	5.24	4.75	5.21	5.15		5.00
5. Shop balance measure, SWB	.490	.777	.531	.771	.702		.654
6. Queue workload balance, QWB	12.3	8.53	10.6	8.51	10.2		9.99
7. Period workload balance, PWB	4.15	4.51	4.2	4.49	4.50		4.31
8. Period queue balance, PQB	91.4	35.7	28.3	11.9	17.1		36.9
9. Average queue size	2.93	2.52	2.72	2.50	2.52		2.64
10. Average work in process in hours	607	534	574	532	539		557
11. Average number of jobs in the shop	37.7	33.3	35.5	33.1	33.3		34.6
12. Average operation done for jobs in shop	73.6	81.9	73.2	80.8	82.5		78.4
13. Average work hours done for jobs in shop	171	196	170	191	197		185
14. Average lateness	18.4	-9.95	-9.08	-9.01	-11.4		-4.21
15. Variance of lateness distribution	1020	774	897	803	840		867
16. Average job tardiness	23.9	4.76	16.4	5.76	4.51		10.9
17. Average tardiness variance	513	81.9	296	117	73.2		216
18. Number of jobs entering shop	2132	2113	2132	2113	2110		2120
19. Average shop utilization	83.4	81.6	83.2	81.5	81.3		82.2
20. Average number jobs in pool before loading	22.5	12.3	19.5	13.0	11.3		15.7
21. Average number jobs in pool after loading	18.2	8.1	15.3	8.7	7.1		11.5

Table 51. Simulation Results

Conditions: Symmetric transition matrix, Loose due date generation (method 2), SINPER 16, No pool, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	74.1	74.9	75.4	75.4	76.0		75.2
2. Time spent in the shop	74.1	74.9	75.4	75.4	76.0		75.2
3. Aggregate deviation from Des. Bal.	148	138.	144	143	142		143
4. Machine balance measure, MWB	5.30	4.81	5.39	5.33	5.36		5.24
5. Shop balance measure, SWB	1.17	.929	1.22	1.23	1.21		1.15
6. Queue workload balance, QWB	14.7	15.7	14.5	15.0	14.8		14.9
7. Period workload balance, PWB	4.18	3.89	4.23	4.15	4.20		4.13
8. Period queue balance, PQB	89.5	78.9	23.4	43.6	88.4		64.8
9. Average queue size	3.1	3.17	3.17	3.15	3.17		3.15
10. Average work in process in hours	635	659	646	645	643		646
11. Average number of jobs in the shop	39.2	40.0	39.9	39.7	39.8		39.7
12. Average operation done for jobs in shop	124	127	126	126	125		126
13. Average work hours done for jobs in shop	305	313	311	309	308		309
14. Average lateness	-96.2	-91.2	-94.9	-95.1	-95.1		-94.5
15. Variance of lateness distribution	4161	3887	4136	4095	4111		4078
16. Average job tardiness	.012	.004	.002	.006	.001		.005
17. Average tardiness variance	.104	.008	.003	.037	.004		.031
18. Number of jobs entering shop	2113	2132	2113	2110	2113		2116
19. Average shop utilization	81.6	83.1	81.4	81.2	81.5		81.8
20. Average number jobs in pool before loading	0	0	0	0	0		0
21. Average number jobs in pool after loading	0	0	0	0	0		0

Table 52. Simulation Results

Conditions: Symmetric transition matrix, Loose due date generation (method 2), DESLF 3.50, SINPER 16, FACDUD 80, Pool loading, Modified GENMAT to force jobs in the shop 24 hours before required by job content, DSOP

Run Number	1	2	3	4	5	6	Avg.
1. Time spent in the system	76.0	78.4	72.4	77.9	77.1		77.4
2. Time spent in the shop	59.0	60.7	58.7	59.6	59.2		59.4
3. Aggregate deviation from Des. Bal.	84.2	77.3	82.6	84.4	83.7		82.4
4. Machine balance measure, MWB	5.30	4.76	5.31	5.40	5.18		5.19
5. Shop balance measure, SWB	.988	.667	.891	1.04	1.02		.921
6. Queue workload balance, QWB	6.79	8.38	6.59	7.33	6.65		7.15
7. Period workload balance, PWB	4.37	4.11	4.48	4.42	4.21		4.32
8. Period queue balance, PQB	33.0	24.5	10.5	16.7	48.0		26.5
9. Average queue size	2.31	2.42	2.29	2.33	2.30		2.33
10. Average work in process in hours	508	527	503	509	504		511
11. Average number of jobs in the shop	31.2	32.5	31.1	31.4	31.2		31.4
12. Average operation done for jobs in shop	98.5	103	97.8	98.8	97.5		99.1
13. Average work hours done for jobs in shop	241	248	238	240	238		241
14. Average lateness	-94.4	-87.5	-92.9	-92.6	-94.0		-92.3
15. Variance of lateness distribution	4294	4179	4385	4210	4245		4263
16. Average job tardiness	.016	.032	.017	.019	.015		.020
17. Average tardiness variance	.098	.140	.060	.091	.060		.090
18. Number of jobs entering shop	2113	2132	2113	2110	2113		2116
19. Average shop utilization	81.6	83.3	81.5	81.3	81.9		81.9
20. Average number jobs in pool before loading	11.1	11.5	12.0	11.7	11.3		11.5
21. Average number jobs in pool after loading	6.8	7.2	7.7	7.4	7.0		7.2

APPENDIX J

STATISTICAL TEST RESULTS

Table 53. T Tests, Paired Observations

CRITICAL VALUES Nonpaired test t.95(8) = 1.860 t.99(8) = 2.896 Paired tests t.95(4) = 2.132 t.99(4) = 3.747	Tables 29,35, Flat vs fluctuating Arriv. No pool, DSOP	Tables 30,39, Heuristic vs algorithm Pool, DSOP	Tables 31,32, No pool vs pool Few interactions shop, DSOP	Tables 33,34, No pool vs pool Asymmetric trans matrix, DSOP	Tables 35,39, No pool vs pool Basic runs, DSOP	Tables 36,40, No pool vs pool Basic runs, EWIQ	Tables 37,41, No pool vs pool Basic runs, SPT	Tables 38,42, No pool vs pool Basic runs, FCFS
R 3, Dev. from BAL	- 6.14	5.78	16.04	28.83	10.98	16.91	25.00	12.97
4, MWB(t)	- 3.29	1.55	9.95	-1.21	-2.24	- .92	-3.70	.56
5, SWB	- 9.14	6.12	7.03	4.97	5.87	1.81	.72	3.61
6, QWB	- 1.77	7.97	7.48	13.73	13.66	12.57	14.87	10.71
7, PWB	+ 4.93	-2.84	3.58	-8.20	-9.39	-2.37	-4.68	-3.34
8, PQB	+ 3.34	3.07	2.55	4.77	2.28	1.93	- .18	4.44
10, Avg WIP hrs	-12.07	8.04	24.55	21.21	12.79	8.62	17.65	11.36
13, Avg hrs work done	- 2.89	6.38	61.82	15.90	10.95	20.41	6.76	11.33
14, Var of lateness	+ 2.35	-4.05	-1.19	-4.12	-3.52	19.73	4.93	7.08
15, Avg tardiness	- .16	-2.76	-7.35	-7.17	-2.93	9.63	2.13	-1.70

Table 53. (Concluded)

	Tables 45,46, No pool vs pool Basic runs, SPT with DESLF = /2.25	Tables 35,47, No pool vs pool Basic runs, DSOP with DESLF = /3.50	Tables 41,46, DESLF 4.25 vs 2.25 Pool, SPT	Tables 39,47, DESLF 4.25 vs 3.50 Pool, DSOP	Tables 48,49, Using alternative machine pairs No pool vs pool, DSOP	Tables 35,48, No pool, DSOP 10 machines vs 5 alt mach pairs	Tables 47,49, Pool, DSOP 10 machines vs 5 alt mach pairs
R 3, Dev. from BAL	16.82	2.58	9.77	-2.02	.67	4.37	3.81
4, MWB(t)	7.24	-6.18	5.02	-1.55	3.65	5.73	17.70
5, SWB	12.57	10.86	5.11	2.67	8.87	-9.07	-7.03
6, QWB	15.68	9.61	14.36	2.80	13.73	27.15	13.89
7, PWB	-5.87	-12.88	1.10	-5.24	-19.35	20.56	19.64
8, PQB	2.67	2.20	1.39	1.72	1.37	2.62	2.71
10, Avg WIP hrs	15.88	7.95	10.80	2.60	15.59	50.40	22.11
13, Avg hrs work done	16.73	20.38	16.26	12.25	24.64	21.73	14.76
14, Var of lateness	5.94	-3.43	3.20	-1.41	.65	-16.42	.63
15, Avg tardiness	-6.38	-4.82	-7.37	-4.95	-3.85	3.81	4.77

Table 54. ANOVA F Tests (F values)

$F_{.99(3.16)} = 5.29$ $F_{.95(3.16)} = 3.24$	Tables 35 to 38 No Pool Four Dispatching Rules	Tables 39 to 42 Pool, DESLF = 4.25 Four Dispatching Rules	Tables 47,40,46,42 Pool, DESLF for DSOP = 3.50, SPT = 2.25, Others = 4.25
R 3, Dev from BAL	1.648	22.14	7.37
4, MWB	.881	1.17	3.99
5, SWB	6.815	8.80	21.75
6, QWB	93.598	47.13	49.87
7, PWB	6.030	4.86	7.81
8, PQB	2.904	.134	1.72
10, Avg WIP hrs	75.155	37.90	47.72
13, Avg hrs work done	212.97	110.37	198.
14, Var of lateness	36.78	22.50	20.70
15, Avg tardiness	66.77	2.51	4.44

Table 55. Duncan Ranking Tests

I -- Tables 35-38

No Pool, four dispatching rules

Row 3, Deviation from Bal No difference in the means	SPT EWIQ FCFS DSOP
Row 4, MWB No difference in the means	
Row 5, SWB EWIQ is different	FCFS DSOP SPT EWIQ
Row 6, QWB No difference between DSOP and FCFS Other groupings are different	SPT EWIQ DSOP FCFS
Row 7, PWB EWIQ is different	EWIQ DSOP SPT FCFS
Row 8, PQB SPT is different from FCFS	SPT EWIQ DSOP FCFS
Row 10, Avg WIP (hrs) DSOP and FCFS show no difference	SPT EWIQ DSOP FCFS
Row 13, Avg hours work done No difference between FCFS and EWIQ	SPT DSOP FCFS EWIQ
Row 14, Variance of lateness dest No difference between FCFS and SPT	DSOP FCFS SPT EWIQ
Row 15, Average tardiness No difference between EWIQ, FCFS	DSOP SPT EWIQ FCFS

II -- Tables 39-42

Pool, four dispatching rules

Row 3, Deviation from Balance FCFS and DSOP show no difference SPT and EWIQ show no difference, but the two groups are different from each other	FCFS DSOP SPT EWIQ

Table 55. (Continued)

Row 4, MWB
No difference shown

Row 5, SWB — DSOP FCFS SPT EWIQ
No difference between DSOP and FCFS

Row 6, QWB — SPT EWIQ FCFS DSOP
No difference between FCFS and DSOP

Row 7, PWB — EWIQ SPT FCFS DSOP
EWIQ is different

Row 8, PQB
No difference

Row 10, Avg WIP hrs — SPT EWIQ FCFS DSOP
No difference between FCFS and DSOP

Row 13, Avg hours of work done — SPT DSOP FCFS EWIQ
All means are different

Row 14, Var. of lateness — DSOP FCFS SPT EWIQ
DSOP is different from all others
Also FCFS and EWIQ are different

Row 15, Avg tardiness — SPT EWIQ DSOP FCFS
SPT and FCFS are different

<u>III -- Tables 47,40,46,42</u>

Pool, four dispatching rules with the following DESLF values:
DSOP (3.5), EWIQ (4.25), SPT (2.25), FCFS (4.25)

Row 3, Deviation from balance — SPT FCFS DSOP EWIQ
No difference between SPT and FCFS or between DSOP and EWIQ. Other comparisons show differences.

Row 4, MWB — SPT FCFS DSOP EWIQ
SPT differs from DSOP and EWIQ

Row 5, SWB — SPT DSOP FCFS EWIQ
EWIQ is different from all others
Also SPT is different from FCFS

Table 55. (Concluded)

Row 6, QWB No difference between FCFS and DSOP	SPT	EWIQ	FCFS	DSOP
Row 7, PWB No difference between SPT and FCFS	EWIQ	SPT	FCFS	DSOP
Row 8, PQB No difference				
Row 10, Avg WIP hrs SPT is different from all others Also EWIQ and FCFS are different	SPT	EWIQ	DSOP	FCFS
Row 13, Avg hours of work done All means are different	SPT	DSOP	FCFS	EWIQ
Row 14, Variance of lateness No difference between SPT and FCFS	DSOP	SPT	FCFS	EWIQ
Row 15, Average tardiness DSOP is different	EWIQ	SPT	FCFS	DSOP

APPENDIX K

FORTRAN IV LISTING OF SUBROUTINES CHANGED FOR THE ALTERNATIVE MACHINE OPTION IN THE SIMULATION PROGRAM

```
-FOR,IS SHOPALT2.ENDSV
      SUBROUTINE ENDSV (NSET)
C
C     ***  EVENT SUBROUTINE CALLED WHEN AN END OF SERVICE
C     ***  HAS OCCURRED FOR A JOB OPERATION
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,WB(10),WBM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,MRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,QLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
      MNOW=ATRIB(11)+0.00001
      MNEXT=ATRIB(13)+0.00001
      CALL TMST (XOPS,TNOW,13,NSET)
      XOPS=XOPS+1.0
      CALL TMST (XWKS,TNOW,14,NSET)
      XWKS=XWKS+ATRIB(12)
      ATRIB(32)=ATRIB(32)+ATRIB(12)
      ATRIB(5)=ATRIB(5)-1.0
      IF (ATRIB(5)) 10,10,60
C
C     ***  COLLECT STATISTICS ON THE JOB LEAVING THE SYSTEM
C
   10 TISYS=TNOW-ATRIB(3)
      CALL COLCT (TISYS,11,NSET)
      NOP=ATRIB(10)+0.00001
      NP23=NOP+22
      CALL COLCT (TISYS,NP23,NSET)
      CALL TMST (XISYS,TNOW,12,NSET)
      XISYS=XISYS-1.0
      CALL TMST (XWKSY,TNOW,11,NSET)
      XWKSY=XWKSY-ATRIB(9)
      DDD=ABS(TNOW-ATRIB(4))
      CALL COLCT (DDD,15,NSET)
      TLATE=TNOW-ATRIB(4)
      CALL COLCT (TLATE,12,NSET)
      CALL HISTO (TLATE,-10.0,2.0,1,NSET)
```

```
      TARDY=TLATE
      IF (TLATE.LT.0.0) TARDY=0.0
      CALL COLCT (TARDY,13,NSET)
      TSYNPL=TNOW-ATRIB(33)
      CALL COLCT (TSYNPL,42,NSET)
      NP40=NOP+39
      CALL COLCT (TSYNPL,NP40,NSET)
      TIPOOL=ATRIB(33)-ATRIB(3)
      CALL COLCT (TIPOOL,48,NSET)
      PERPOL=TIPOOL/PLEN+0.5
      NPEPOL=PERPOL
      CALL HISTO (NPEPOL,1.0,1.0,16,NSET)
      NP46=NOP+45
      CALL COLCT (TIPOOL,NP46,NSET)
      B=FLOAT(NTPDS-1)*PLEN
      BDUE=ATRIB(4)
      IF (BDUE.LT.B) GO TO 30
      IF (BDUE.LT.TNOW) GO TO 20
      LP= (TNOW-BDUE/PLEN)-.9999999
      GO TO 40
   20 LP=0
      GO TO 40
   30 LP=(B-BDUE)/PLEN+.999999
   40 XP=LP
      CALL HISTO (XP,-10.5,1.0,2,NSET)
      XOPS=XOPS-ATRIB(10)
      XWKS=XWKS-ATRIB(9)
      NLV=NLV+1
      JOB=ATRIB(30)+.001
      LOC(JOB)=0
      IF (JOB.NE.MAX) GO TO 80
   50 MAX=MAX-1
      JOB=JOB-1
      IF (LOC(JOB).LE.0) GO TO 50
      GO TO 80
C
C     *** THE JOB IS NOT LEAVING THE SYSTEM
C     *** UPDATE THE JOB ATTRIBUTES
C
   60 IF (NRULE.LE.3) ATRIB(6)=ATRIB(6)-ATRIB(12)
      LRM=ATRIB(5)+.001
      LR=2*LRM+9
      DO 70 I=11,LR,2
      ATRIB(I)=ATRIB(I+2)
   70 ATRIB(I+1)=ATRIB(I+3)
      ATRIB(LR+2)=0.0
      ATRIB(LR+3)=0.0
      CALL PTJOB (2,NSET)
C
```

```
C      ***  CHECK MACHINE QUEUE FOR ANY JOBS
C      ***  AVAIABLE FOR PROCESSING
C
   80 IF (NQ(MNOW+1)) 81,81,100
C
C      ***  THERE ARE NO JOBS IN THE QUEUE
C      ***  CHECK QUEUE FOR COMPANION MACHINE
C
   81 CONTINUE
      CAL1=(MNOW+.01)/2.0
      MCAL1=CAL1
      MCAL1=2*MCAL1
      MCAL2=MNOW-MCAL1
      IF (MCAL2.LE.0) MNO1=MNOW-1
      IF (MCAL2.GT.0) MNO1=MNOW+1
      IF (NQ(MNO1+1))90,90,82
   82 CONTINUE
C
C      ***  MORE THAN ONE JOB IS AVAILABLE IN COMPANION
C      ***  MACHINE(MNO1). COMPUTE PRIORITIES AND BRING IN
C      ***  THE JOB WITH THE HIGHEST PRIORITY FROM THE QUEUF.
C
  200 MN1=MNO1+1
      IF (NQ(MN1).EQ.1) GO TO 220
      IF (NRULE.EQ.0.OR.NRULE.GT.3) GO TO 220
      IF (NRULE.GT.2) GO TO 210
      MNO2=MNOW
      MNOW=MNO1
      CALL DYNAM (MBEST,NSET)
      CALL RMOVE (MBEST,MN1,NSET)
      GO TO 230
  210 CALL WKINQ (MBEST,NSET)
      MNOW=MNO2
      IF (MBEST.EQ.0) GO TO 220
      CALL RMOVE (MBEST,MN1,NSET)
      MNOW=MNO2
      GO TO 230
  220 CALL RMOVE (MFE(MN1),MN1,NSET)
C
C      ***  COMPUTE THE WAITING TIME FOR THE JOB AND
C      ***  DECREASE THE WORKLOAD IN THE MACHINE QUEUE.
C
  230 WT=TNOW-ATRIB(8)
      MN15=MNO1+15
      CALL COLCT (WT,MN15,NSET)
      QLOAD(MNO1)=QLOAD(MNO1)-ATRIB(12)
      SHOPLD(MNO1)=SHOPLD(MNO1)-ATRIB(12)
      TIMEVT=ATRIB(12) *(8.0/CAPM(MNOW ))
      ATRIB(1)=TNOW+TIMEVT
```

```
      ATRIB(2)=1.0
      JOB=ATRIB(30)+.001
      LOC(JOB)=MFA
      ATRIB(11)=MNOW
      CALL FILEM(1,NSET)
      RETURN
   90 CALL TMST (BUS(MNOW),TNOW,MNOW,NSET)
      BUS(MNOW)=0.0
      IF (MSW(2).EQ.0) GO TO 93
      IF (NLDR.EQ.0) GO TO 93
      CALL COLCT(1.0,68,NSET)
      IF (NQ(12).LT.1) GO TO 93
      IF (MSW(3).EQ.0) GO TO 88
      IF (SSUMA(MNOW,3).GE.AVGLD9) GO TO 93
C
C     ***  TRY TO MOVE JOB FROM POOL TO EMPTY MACHINE
C
   88 J=0
      N1=MFE(12)
   91 J=J+1
      NFIRST=FLOAT(NSET(11,N1))/SCALE+.0001
      IF (NFIRST.EQ.MNOW) GO TO 92
      N1=NSET(MX,N1)
      IF (N1.NE.7777) GO TO 91
C
C     ***  NO JOB WAS FOUND THAT COULD HELP IDLE MACHINE
C
      GO TO 93
C
C     ***  PUT JOB FROM POOL IN IDLE MACHINE
C
   92 CALL RMOVE(N1,12,NSET)
      CALL COLCT (1.0,69,NSET)
      MNEXT=ATRIB(11)+.00001
      CALL PTJOB(3,NSET)
   93 RETURN
C
C     ***  MORE THAN ONE JOB IS AVAIABLE. COMPUTE
C     ***  PRIORITIES AND BRING IN THE JOB WITH THE
C     ***  HIGHEST PRIORITY FROM THE QUEUE.
C
  100 MN1=MNOW+1
      IF (NQ(MN1).EQ.1) GO TO 120
      IF (NRULE.EQ.0.OR.NRULE.GT.3) GO TO 120
      IF (NRULE.GT.2) GO TO 110
      CALL DYNAM (MBEST,NSET)
      CALL RMOVE (MBEST,MN1,NSET)
      GO TO 130
  110 CALL WKINQ (MBEST,NSET)
```

```
      IF (MBEST.EQ.0) GO TO 120
      CALL RMOVE (MBEST,MN1,NSET)
      GO TO 130
  120 CALL RMOVE (MFE(MN1),MN1,NSET)
C
C     ***  COMPUTE THE WAITING TIME FOR THE JOB AND
C     ***  DECREASE THE WORKLOAD IN THE MACHINE QUEUE.
C
  130 WT=TNOW-ATRIB(8)
      MN15=MNOW+15
      CALL COLCT (WT,MN15,NSET)
      QLOAD(MNOW)=QLOAD(MNOW)-ATRIB(12)
      SHOPLD(MNOW)=SHOPLD(MNOW)-ATRIB(12)
      TIMEVT=ATRIB(12) *(8.0/CAPM(MNOW ))
      ATRIB(1)=TNOW+TIMEVT
      ATRIB(2)=1.0
      JOB=ATRIB(30)+.001
      LOC(JOB)=MFA
      CALL FILEM(1,NSET)
      RETURN
      END
```

```
-FOR,IS SHOPALT2.GENMAT
      SUBROUTINE GENMAT (NSET)
C
C     ***  THIS SUBROUTINE PLACES THE PARAMETERS FOR THE JOBS
C     ***  IN THE JOB POOL IN THE FORM REQUIRED BY LP
C
      DIMENSION NSET(35,1),DUDFT(50), MFOP(10,70) ,KAUX(5)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XMKSY,IDLE,
     1ITYPE,NNEXT,MEN,NLV,NHELD,MB(10),MBM(10),Y(10,10),
     2          BUS(1  ),NRSET,NRULE,MMCM,MRST,MENDS,MHOL,MRL,
     3MAW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON MPREL,MPREP,NDESL,NDML,CAPM(10),DESL(10),
     1DOL(10),DESLF,DMLF,OLOAD(10),XOPS,XMKS,TIMEF(10),
     2NSTSV,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
C
C     ***  REMEMBER TO USE ONLY AN EVEN NUMBER OF  MACHINES
C     ***  WITH THIS SPECIALLY MODIFIED PROGRAM.
C     ***  OBTAIN NO.OF JOBS IN POOL AND INITIALIZE MATRICES
C
      NPOOL=NQ(12)
      MROW=25
      MCOL=80
      NOROW=NM
      NOCOL=NPOOL+2*NM
      INDEX=0
      DO 3  I=1,5
    3 KAUX(I)=0
      OBJIN=0.0
      DO 1  I=1,NM
      DO 1  J=1,NPOOL
      MFOP(I,J)=0.
    1 CONTINUE
      DO 2  I=1,NM
      DO 2  J=1,NOCOL
    2 A(I,J)=0.0
      J=0
      M1=MFE(12)
C
C     ***  OBTAIN LP MATRIX ENTRIES FOR EACH JOB
C
   50 J=J+1
```

```
      WKTIM=0.0
      NO1=FLOAT(NSET(10,N1))/SCALE+.000001
      DO 35 I=1,NO1
      NON1=9+2*I
      NON2=FLOAT(NSET(NON1,N1))/SCALE+.000001
      NON3=NON1+1
      WOL=FLOAT(NSET(NON3,N1))/SCALE
      A(NON2,J)=WOL
      WKTIM=WKTIM+WOL*(8.0 /CAPM(NON2))
      IF (NLDR.NE.2.OR.NLDR.NE.3) GO TO 35
      IF (NON1.NE.11) GO TO 35
      WFOP(NON2,J)=WOL
   35 CONTINUE
      TIMDUE =FLOAT(NSET(4,N1))/SCALE+.000001
      WRKDUE =FLOAT(NSET(9,N1))/SCALE+.000001
      DUDSLK=TIMDUE-TNOW-WKTIM
      IF (DUDSLK    .LE.0.0) DUDSLK    =0.0
      DUDFT(J)=FACDUD/(DUDSLK+.01)
C
C     ***  OBTAIN NEXT JOB IN THE POOL,IF THERE IS ANY
C
      N1=NSET(MX,N1)
      IF (N1.NE.7777) GO TO 30
C
C     ***  SET UP MATRICES REQUIRED BY LPI
C
      NM1=NM/2
      NOROW=NM1
      NOCOL=NPOOL+2*NM1
      DO 45 I=1,NM1
      DO 45 J=1,NPOOL
      II1=(2*I)-1
      II2= 2*I
   45 A(I,J)=A(II1,J)+A(II2,J)
      IF (NLDR.EQ.2.OR.NLDR.EQ.3) GO TO 60
      DO 51 I=1,NOROW
      DO 52 J=1,NOCOL
      IF (J.EQ.(NPOOL+I)) A(I,J)=1.
      IF (J.EQ.(NPOOL+NM1+I)) A(I,J)=-1.0
   52 CONTINUE
      DESL2=DESL(II1)+DESL(II2)
      SHOPL2=SHOPLD(II1)+SHOPLD(II2)
      A(I,NOCOL+1)=DESL2 -SHOPL2
      AA=A(I,NOCOL+1)
      OBJIN=OBJIN+ABS(AA)
      KBV(I)=NPOOL+I
      IF (AA.GE.0.0) GO TO 51
      A(I,NOCOL+1)= -AA
      KBV(I)=NPOOL+NM1+I
```

```
      DO 54 J=1,NOCOL
   54 A(I,J)= -A(I,J)
   51 CONTINUE
      GO TO 71
C
C     ***  MATRIX PREPARATION WHEN NEXT QUEUE RULE IS USED
C
   60 DO 61 I=1,NOROW
      DO 62 J=1,NOCOL
      II1=(2*I)-1
      II2= 2*I
      IF (J.LE.NPOOL) A(I,J)=WFOP(II1,J)+WFOP(II2,J)
      IF (J.EQ.(NPOOL+I)) A(I,J)=1.
      IF (J.EQ.(NPOOL+NM1+I)) A(I,J)=-1.0
   62 CONTINUE
      DQL2=DQL(II1)+DQL(II2)
      QLOAD2=QLOAD(II1)+QLOAD(II2)
      A(I,NOCOL+1)= DQL2 -QLOAD2
      AA=A(I,NOCOL+1)
      OBJIN=OBJIN+ABS(AA)
      KBV(I)=NPOOL+I
      IF (AA.GE.0.0) GO TO 61
      A(I,NOCOL+1)= -AA
      KBV(I)= NPOOL+NM1+I
      DO 64 J=1,NOCOL
   64 A(I,J)= -A(I,J)
   61 CONTINUE
   71 CONTINUE
      NRT1=NOROW+1
      NCT2=NOCOL+2
      NPT2=NPOOL+(2*NM)
      DO 72 I=NRT1,NM
      DO 72 J=NCT2,NPT2
   72 A(I,J)=0.0
      DO 76 J=1,NOCOL
      C(J)=0.0
      IF (J.GT.NPOOL) GO TO 77
      C(J)=-DUDFT(J)
      A(NOROW+1,J)=1.0
      A(NOROW+2,J)=1.0
      GO TO 76
   77 C(J)=1.0
      A(NOROW+1,J)=-1.0
      A(NOROW+2,J)=1.0
   76 CONTINUE
      C(NOCOL+1)= -OBJIN
      A(NOROW+1,NOCOL+1)= 0.0
      A(NOROW+2,NOCOL+1)= 0.0
      DO 88 I=1,NOROW
```

```
      DO 88 J=1,NOCOL
      C(J)=C(J)-A(I,J)
   88 CONTINUE
      IF (NLDR.GE.4) GO TO 91
      CALL LPI (NSET,NOROW,NOCOL,MROW,MCOL,INDEX,KAUX)
      GO TO 92
   91 CALL POOLHE (NSET,NOROW,NOCOL)
   92 RETURN
      END



-FOR,IS SHOPALT2.JOBDEC
      SUBROUTINE JOBDEC (NSET,NOROW,NOCOL)
C
C     ***  THIS SUBROUTINE USES THE LP RESULTS TO MAKE THE
C     ***  FINAL SELECTION REGRADING THE JOBS THAT SHOULD
C     ***  BE LOADED IN THE SHOP
C
      DIMENSION  NSET(35,1),XJOB(100)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPOS,NTOTPD,NM,XISYS,XMKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,NB(10),NBM(10),X(10,10),
     2         BUS(10),NRSET,NRULE,MNOW,MRST,NENDS,NHOL,NRL,
     3WWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPREP,NDESL,NDYL,CAPM(10),DESL(10),
     1DGL(10),DESLF,DMLF,QLOAD(10),XOPS,XMKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      NM1=NM/2
      NPOOL=NQ(12)
      DO 1 J=1,NPOOL
      XJOB(J)=0.0
      AA1=A(NOROW+2,J) -.0001
      IA=IFIX(AA1)
    1 IF (IA.EQ.-1) XJOB(J)=1.0
      IF (NLDR.GE.4) GO TO 20
      DO 2 I=1,NM1
      JJ=KBV(I)
      XJOB(JJ)=A(I,NOCOL+1)
C
C     ***  VARIABLES IN BASIS AND WITH UPPER BOUND INDICATOR
```

```
C       ***  ON, NEED TO BE CALCULATED DIFFERENTLY
C
        AA1=A(NOROW+2,JJ)-.001
        IA=IFIX(AA1)
        IF (IA.NE.-1) GO TO 2
        XJOB(JJ)=A(NOROW+1,JJ)-A(I,NOCOL+1)
      2 CONTINUE
C
C       ***  SEARCH JOB POOL FILE AND LOAD IN THE SHOP THOSE
C       ***  JOBS WITH DECISION VARIABLE .GE. .75
C
     20 J=0
        N1=MFE(12)
        WKSHP1=0.0
        TDES1=0.0
        DO 25 I=1,NM
        TDES1=TDES1+DESL(I)
     25 WKSHP1=WKSHP1+SHOPLD(I)
     30 J=J+1
        XJBN=XJOB(J)
        IF (XJBN.LT.0.75) GO TO 40
        N2=NSET(MX,N1)
        CALL RMOVE (N1,12,NSET)
        WKSHP1=WKSHP1+ATRIB(9)
        MNEXT=ATRIB(11) +.00001
        CALL PTJOB (3,NSET)
        N1=N2
        GO TO 41
     40 N1=NSET(MX,N1)
     41 CONTINUE
        IF (N1.NE.7777) GO TO 30
C
C       ***  SEARCH JOB POOL FILE AND LOAD JOBS WITH DECISION
C       ***  VARIABLES BETWEEN 0.3 AND 0.75 IF TOTAL SHOP LOAD
C       ***  IS LESS THAN DESIRED
C
        J=0
        IF (NQ(12).EQ.0) GO TO 70
        N1=MFE(12)
     50 IF (WKSHP1.GE.TDES1) GO TO 70
     55 J=J+1
        IF (J.GT.NPOOL) GO TO 70
        XJBN=XJOB(J)
        IF (XJBN.GE.0.75) GO TO 55
        IF (XJBN.LT.0.3) GO TO 65
        N2=NSET(MX,N1)
        CALL RMOVE (N1,12,NSET)
        WKSHP1=WKSHP1+ATRIB(9)
        MNEXT=ATRIB(11) +.00001
```

```
      CALL PTJOB (3,NSET)
      N1=N2
      GO TO 66
   65 N1=NSET(MX,N1)
   66 CONTINUE
      IF (N1.NE.7777) GO TO 50
   70 CONTINUE
      RETURN
      END
```

```
-FOR,IS SHOPALT2.PTJOB
      SUBROUTINE PTJOB (INP,NSET)
C
C     ***  SUBROUTINE WHICH MOVES JOB TO NEXT MACHINE
C     ***  CENTER
C
      DIMENSION NSET(35,1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,
     1NHIST,NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,SCALE,
     2ISEED,TNOW,TBEG,TFIN,MXX,NPRNT,KCRDR,NEP,VNQ(25),
     3KOF,KLE,KOL,ATRIB(33),ENQ(25),INN(25),JCELS(20,32),
     4KRANK(25),JCLR,MAXNQ(25),MFE(25),MLC(25),MLE(25),
     5 NCELS(20),NQ(25),PARAM(40,4),QTIME(25),SSUMA(20,5)
     6,SUMA(75,5),NAME(6),NPROJ,MON,NDAY,NYR
      COMMON PLEN,NTPDS,NTOTPD,NM,XISYS,XWKSY,IDUE,
     1ITYPE,MNEXT,NEN,NLV,NHELD,MB(10),MBM(10),X(10,10),
     2          BUS(10),NRSET,NRULE,MNOW,NRST,NENDS,NHOL,MRL,
     3MWW(10),SEED,ARATE,LOC(200),MAX,AR(11)
      COMMON NPREL,NPRLP,NDESL,NDML,CAPM(10),DESL(10),
     1DQL(10),DESLF,DMLF,DLOAD(10),XOPS,XWKS,TIMEF(10),
     2NSTSW,NLDR,NARR,SHOPLD(10)
      COMMON A(25,100),KBV(15),C(100),FACDUD
      COMMON ICOUNT,NCOUNT,SINPER,MSW(10),AVGLD9
C
C     ***  CHECK IF JOB IS A NEW ARRIVAL
C
      IF (INP.NE.1) GO TO 10
      ATRIB(3)=TNOW
      NEN=NEN+1
C
C     ***  NEW ARRIVAL. CHECK IF A JOB POOL IS BEING USED
C
      IF (NLDR.EQ.0) GO TO 20
C
C     ***  CHECK IF SHOP IS BEING PRELOADED AND JOB POOL
C     ***  HAS BEEN COMPLETED
```

```
C
      IF (NSTSW.EQ.1) GO TO 20
C
C     ***  PUT ARRIVING JOB IN THE POOL IF OP. 1 MACH IS NOT IDLE
C
      ATRIB(8)=TNOW
      JOB=ATRIB(30)+0.001
      LOC(JOB)=MFA
C
C     ***  COLLECT STATISTICS ON INTERARRIVAL TIMES TO
C     ***  THE JOB POOL
C
      D=TNOW-AR(11)
      CALL HISTO (D,0.5,0.5,15,NSET)
      AR(11)=TNOW
      NFIRST=ATRIB(11)+0.00001
      IF (MSW(1).EQ.0) GO TO 4
      IF (TNOW.LE.0.0001) GO TO 4
      IF (BUS(NFIRST)) 5,5,4
    4 CALL FILEM(12,NSET)
      GO TO 70
C
C     ***  IF FIRST OPERATION MACHINE IS IDLE,CONSIDER THE
C     ***  JOB AS COMING FROM POOL AND PUT IN THE SHOP
C
    5 CONTINUE
      IF (MSW(3).EQ.0) GO TO 6
      IF(SSUMA(NFIRST,3).GE.AVGLD9) GO TO 4
    6 MNEXT=NFIRST
      CALL COLCT (1.0,69,NSET)
      GO TO 20
C
C     ***  JOB IS NOT A NEW ARRIVAL. CHECK IF IT IS COMING
C     ***  FROM THE POOL
C
   10 IF (INP.EQ.2) GO TO 40
C
C     ***  JOB IS COMING FROM THE POOL.
C     ***  ALSO NEW JOBS WHEN A POOL IS NOT USED ARRIVE
C     ***  AT THIS POINT
C     ***  UPDATE STATUS OF WORK IN SHOP AND ALSO UPDATE
C     ***  AGGREGATE LOAD IN SHOP QUEUES FOR EACH MACHINE.
C
   20 CALL TMST (XISYS,TNOW,12,NSET)
      CALL TMST (XWKSY,TNOW,11,NSET)
      XISYS=XISYS+1.0
      XWKSY=XWKSY+ATRIB(9)
      ATRIB(33)=TNOW
      [illegible]=9.0+2.*ATRIB(10)+0.0001
```

```
      DO 37 I=11,NNN,2
      J=ATRIB(I)
   37 SHOPLD(J)=SHOPLD(J)+ATRIB(I+1)
C
C     ***  JOB IS NOT GOING INTO THE POOL. COLLECT STATISTICS
C     ***  ON INTERARRIVAL TIMES TO THE CURRENT MACHINE
C
   40 D=TNOW-AR(MNEXT)
      MN4=MNEXT+4
      CALL HISTO (D,0.5,0.5,MN4,NSET)
      AR(MNEXT)=TNOW
C
C     ***  CHECK ON THE STATUS OF MACHINE FOR NEXT
C     ***  JOB OPERATION
C
      IF (BUS(MNEXT)) 60,60,41
C
C     ***  NEXT MACHINE IS BUSY. JOB CAN NOT BE PUT ON
C     ***  MACHINE. CHECK COMPANION MACHINE.
C
   41 CONTINUE
      CAL1=(MNEXT+.01)/2.0
      MCAL1=CAL1
      MCAL1=2*MCAL1
      MCAL2=MNEXT-MCAL1
      IF (MCAL2.LE.0) MNEX1=MNEXT-1
      IF (MCAL2.GT.0) MNEX1=MNEXT+1
      IF (BUS(MNEX1))42,42,50
C
C     ***  COMPANION MACHINE IS NOT BUSY.
C     ***  PUT JOB IN COMPANION MACHINE.
C
   42 CONTINUE
      CALL TMST (BUS(MNEX1),TNOW,MNEX1,NSET)
      BUS(MNEX1)=1.0
      WT=0.0
      MX15=MNEXT+15
      CALL COLCT (WT,MX15,NSET)
      TIMEVT=ATRIB(12) *(8. /CAPM(MNEXT))
      ATRIB(1)=TNOW+TIMEVT
      ATRIB(2)=1.0
      ATRIB(11)=MNEX1
      J=ATRIB(11)
      SHOPLD(J)=SHOPLD(J)-ATRIB(12)
      JOB=ATRIB(30)+0.001
      LOC(JOB)=MFA
      CALLFILEM  (1,NSET)
      GO TO 70
C
```

```
C      ***  NEXT MACHINE AND ITS COMPANION ARE BUSY.
C      ***  JOB CAN NOT BE PUT ON EITHER MACHINE.
C
   50 ATRIB(8)=TNOW
      MX1=MNEXT+1
      JOB=ATRIB(30)+0.001
      LOC(JOB)=MFA
      QLOAD(MNEXT)=QLOAD(MNEXT)+ATRIB(12)
      CALL FILEM (MX1,NSET)
      GO TO 70
C
C      ***  NEXT MACHINE IS NOT BUSY.
C      ***  JOB MAY BE PUT ON MACHINE
C
   60 CALL TMST (BUS(MNEXT),TNOW,MNEXT,NSET)
      BUS(MNEXT)=1.0
      WT=0.0
      MX15=MNEXT+15
      CALL COLCT (WT,MX15,NSET)
      TIMEVT=ATRIB(12) *(8.0/CAPM(MNEXT))
      ATRIB(1)=TNOW+TIMEVT
      ATRIB(2)=1.0
      J=ATRIB(11)
      SHOPLD(J)=SHOPLD(J)-ATRIB(12)
      JOB=ATRIB(30)+0.001
      LOC(JOB)=MFA
      CALLFILEM  (1,NSET)
   70 NSTSW=0
      RETURN
      END
```

BIBLIOGRAPHY

1. Ackerman, Sanford S., "Even-Flow; A Scheduling Method for Reducing Lateness in Job Shops," Management Technology, Vol. 3, No. 1, May 1963, pp. 20-32.

2. Ashour, S., "A Branch and Bound Algorithm for Flow Shop Scheduling Problems," AIIE Transactions, Vol. 2, No. 2, June 1970.

3. Ashour, S., "Graph-Theoretic Approach to Flow Shop Scheduling Problems," Production and Inventory Management, 4th quarter, 1969.

4. Ashour, S., "An Experimental Investigation and Comparative Evaluation of Flow-Shop Scheduling Techniques," Operations Research, Vol. 18, No. 3, May-June 1970, pp 541-548.

5. Ashour, S., " A Decomposition Approach for the Machine Scheduling Problem," International Journal of Production Research, Vol. 6, No. 2, 1967.

6. Ashour, S. and Quraishi, "Investigation of Various Bounding Procedures for Production Scheduling Problems," International Journal of Production Research, Vol. 7, N9. 3, 1969.

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

8. Baker, K. R., "Priority Dispatching in the Single Channel Queue with Sequence Dependent Set Ups," Journal of Industrial Engineering, Vol. 19, No. 4, April 1968.

9. Bakshi, M. S. and Arora, S. R., "The Sequencing Problem," Management Science, Vol. 16, No. 4, December 1969.

10. Balas, E., "Machine Sequencing via Disjunctive Graphs. An Implicit Enumeration," Operations Research, Vol. 17, No. 6, November-December 1969, pp 941-957.

11. Beaird, Samuel D., "Computerized Production Control for Job Lot Manufacturing," Automation, February 1966.

12. Beaird, Samuel D., "Job Shop Planning and Control - Part I," Automation, Vol. 15, No. 3, March 1968, pp 79-88.

BIBLIOGRAPHY (Continued)

13. Beaird, Samuel D., "Job Shop Planning and Control - Part II," Automation, Vol. 15, No. 5, May 1968, pp 92-98.

14. Beenhakker, H. L., "Development of Alternate Criteria for Optimality in the Machine Sequencing Problem," Unpublished Ph.D. Thesis, Purdue University, 1963.

15. Bellman, R., "Mathematical Aspects of Scheduling Theory," Rand Report, April 11, 1955, pp 651.

16. Benedict, A. G., "Time Sharing Keeps Machine Shop Production on Time," Computer Decisions, Vol. 2, No. 2, February 1970.

17. Berry, W. L., "Priority Scheduling and Inventory Control in Job Lot Manufacturing Systems," AIIE Transactions, Vol. 4, No. 1, December 1972, pp 267-276.

18. Blick, R. G., "Heuristics for Scheduling the General n/m Job-Shop Problem," General Electric Report No. 69-C-162, April 1969.

19. Bowman, E. H., "The Schedule-Sequencing Problem," Operations Research, Vol. 7, No. 5, September 1959, pp 621-624.

20. Brooks, George H. and White, Charles R., "An Algorithm for Finding Optimal or Near Optimal Solutions to the Scheduling Problem," Journal of Industrial Engineering, Vol. 16, No. 1, January-February 1965, pp 34-40.

21. Brown, A. P. G. and Lomnicki, Z. A., "Some Applications of the Branch and Bound Algorithm to the Machine Scheduling Problem," Operations Research Quarterly, Vol. 17, 1966, pp 173-186.

22. Bozoki, G. and Richard, J., "A Branch and Bound Algorithm for the Continuous-Process Job Shop Scheduling Problem," AIIE Transactions, Vol. 2, No. 3, September 1970.

23. Buffa, E. S. and Taubert, W. H., "Production-Invensory Systems, Planning and Control," Part V, Richard D. Irwin, Inc., Homewood, Illinois, 1972.

24. Bulkin, M. H., Colley, J. L., and Steinhoff, H. W., "Load Forecasting, Priority Sequencing, and Simulation in a Job Shop Control System," Management Science, Vol. 13, No. 2, October 1966, pp 29-51.

25. Bureau of Business Practice, "Small Backlogs-Big Control," Executive Bulletin No. 223, April 30, 1965.

BIBLIOGRAPHY (Continued)

26. Burke, P. J., "Output Processes and Tandem Queues," Paper presented at the Symposium on Computer-Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, April 4-6, 1972.

27. Campbell, H. G., Dudek, R. A., and Smith, M. L., "A Heuristic Algorithm for the N Jobs, M Machine Sequencing Problems," Management Science, Vol. 16, No. 10, June 1970.

28. Charlton, J. M. and Death, C. C., "A Method of Solution for General Machine Scheduling Problems," Operations Research, Vol. 18, No. 4, July-August 1970, pp 689-707.

29. Chung, A. M., Linear Programming, Chas. E. Merrill Books, Columbus, Ohio, 1963.

30. Collins, Robert E., "Routing, Scheduling, and Dispatching the Job--A Case," American Production and Inventory Control Quarterly Bulletin, Vol. 6, No. 4, October 1965, pp 25-42.

31. Conway, Richard W., Johnson, Bruce M., and Maxwell, William L., "An Experimental Investigation of Priority Dispatching," Journal of Industrial Engineering, May-June 1960, pp 221-229.

32. Conway, R. W. and Maxwell, W. L., "Network Dispatching by the Shortest Operation Discipline," Operations Research, Vol. 10, No. 1, February 1962.

33. Conway, Richard W. and Maxwell, W. L., "Network Dispatching by the Shortest Operation Discipline," Chapter 17 of Industrial Scheduling, Prentice-Hall, Inc., 1963.

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

35. Conway, R. W., "Priority Dispatching and Work-in-Process Inventory in a Job Shop," Journal of Industrial Engineering, Vol. 16, No. 2, March-April 1965, pp 123-130.

36. Conway, R. W., Maxwell, William L., and Miller, Louis W., Theory of Scheduling, Addison-Wesley Publishing Company, Reading Massachusetts, 1967.

37. Conway, R. W., "Priority Dispatching and Job Lateness in a Job Shop," Journal of Industrial Engineering, Vol. 16, No. 4, August 1965.

38. Dantzig, G. B., "A Machine Job Scheduling Model," Management Science, Vol. 6, p 191.

BIBLIOGRAPHY (Continued)

39. Davis, John D., "Production Control from Order to Shipment - For Computers, By Computers," Computers and Automation, Vol. 16, No. 7, July 1967, pp 14-18.

40. Day, J. E. and Hottenstein, M. P., "Review of Sequencing Research," Naval Research Logistics Quarterly, Vol. 17, No. 1, March 1970, pp 11-39.

41. Deane, R. H. and Moodie, C. L., "A Dispatching Methodology for Balancing Workload Assignments in a Job Shop Production Facility," Research Memo No. 71-13, December 1971, Purdue University.

42. Deane, R. H., "Scheduling Methodologies for Balancing Workload Assignments in the Job Shop Manufacturing Environment," Ph.D. Dissertation, Purdue University, January 1972.

43. Deane, R. H. and Moodie, C. L., "A Dispatching Methodology for Balancing Workload Assignments in a Job Shop Production Facility," AIIE Transaction, Vol. 4, No. 4, December 1972, pp 277-283.

44. Demezynski, S., "Production Control and Machine Loading in a Jobbing Shop," Control, October-December 1961, pp 94-96, 102, 109-110.

45. Dessouky, M. I. and Morgenthaler, C. R., "The One Machine Sequencing Problem with Early Starts and Due Dates," AIIE Transactions, Vol. 4, No. 3, September 1972.

46. Dudek, Richard A. and Teuton, O. F., Jr., "Development of M-Stage Decision Rule for Scheduling n Jobs through M-Machines," Operations Research, Vol. 12, No. 3, 1964, pp 471-497.

47. Eastman, W. L., Even, S., Isaacs, I. M., "Bounds for the Optimal Scheduling of N Jobs on M Processors," Management Science, Vol. 11, Series A, No. 2, November 1964, p 268.

48. Ebert, R. J., "Time Horizon: Implications for Aggregate Scheduling Effectiveness," AIIE Transactions, Vol. 4, No. 4, December 1972, pp 298-307.

49. Eilon, S. and Christofides, N., "The Loading Problem," Management Science, Vol. 17, No. 5, January 1971.

50. Eilon, S. and Christofides, N., "On the Loading Problem, A Rejoinder," Management Science, Vol. 11., No. 7, March 1972.

51. Elmaghraby, S. E. and Cole, R. T., "On the Control of Production in Small Job Shops," Journal of Industrial Engineering, Vol. 14, No. 4, July-August 1963.

BIBLIOGRAPHY (Continued)

52. Elmaghraby, S. E., The Design of Production Systems, Chapter 5, Reinhold Publishing Corp., New York, 1966.

53. Elmaghraby, S. E., "The Machine Sequencing Problem, Review and Extension," Naval Research Logistics Quarterly, Vol. 15, No. 2, pp 205-232.

54. Emlaghraby, S. E., "A Loading Problem in Process Type Productions," Operations Research, Vol. 16, No. 5, September-October 1968, pp 902-915.

55. Elmaghraby, S. E., "The One Machine Sequencing Problem with Delay Costs," Journal of Industrial Engineering, Vol. 19, No. 2, February 1968, pp 105-108.

56. Emery, James C., Wright, O. W., and Brown, R. G., "Job Lot Scheduling-Handling's Partner for Profit," Modern Materials Handling, Vol. 21, No. 10, October 1966, pp 41-56.

57. Emery, James C., "Job Shop Scheduling by Means of Simulation and an Optimum-Seeking Search," Proceedings of the Third Conference on Applications of Simulations, December 1969.

58. Emmons, H., "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness," Operations Research, Vol. 17, No. 4, July-August 1969, pp 701-715.

59. Fabricky, W. J. and Shamblin, J. E., "A Probability Based Sequencing Algorithm," Journal of Industrial Engineering, Vol. XVII, No. 6, June 1966, pp 308-312.

60. Fishman, G. S. and Kiviat, P. J., "The Analysis of Simulation-Generated Time Series," Management Science, Vol. 13, No. 7, March 1967.

61. Florian, M., Trepant, P., and McMahon, G., "An Implicit Enumeration Algorithm for the Machine Sequencing Problem," Management Science, Vol. 17, No. 12, August 1971.

62. Fox, P. D. and Kriebel, C. H., "An Empirical Study of Scheduling Decision Behavior," Journal of Industrial Engineering, Vol. 18, No. 6, June 1967.

63. Franklin, Carter L., III, "The Current State of Research in Job Shop Scheduling," Proceedings of the Third Conference on Applications of Simulation, December 1969.

BIBLIOGRAPHY (Continued)

64. Gapp, E. W., Markekar, P. S., and Mitten, L. G., "Sequencing Operations to Minimize In Process Inventory Costs," Management Science, Vol. 11, No. 3, January 1965.

65. Gere, W. S., Jr., "Heuristics in Job Shop Scheduling," Management Science, Vol. 13, No. 3, November 1966, pp 167-190.

66. Thare, P. M., Givens, J. M., and Torgensen, P. E., "A Machine Release Scheme for the Job Shop," AIIE Transactions, Vol. 1, No. 3, 1969.

67. Giffler, B. and Thompson, G. L., "Algorithms for Solving Production Scheduling Problems," Operations Research, Vol. 8, No. 4, July-August 1960, pp 487-503.

68. Giglio, R. J. and Wagner, H. M., "Approximate Solutions to the 3 Machine Scheduling Problem," Operations Research, Vol. 12, No. 2, 1964, pp 305-324.

69. Glassey, C. R., "Dynamic Linear Programs for Production Scheduling," Operations Research, Vol. 19, No. 1, January-February 1971, pp 45-56.

70. Greenberg, H. H., "A Branch and Bound Solution to the General Scheduling Problem," Operations Research, Vol. 16, No. 2, March-April 1968, pp 353.

71. Greenberg, I., "Application of the Loading Algorithm to Balance Workloads," AIIE Transactions, Vol. 4, No. 4, December 1972, pp 337-339.

72. Grossman, L., "Using A Computer to Control Production," Automation, Vol. 15, No. 10, October 1968, pp 60-65.

73. Gupta, J. N. D. and Dudek, R. A., "Optimality Criteria for Flowshop Schedules," AIIE Transaction, Vol. III, No. 3, September 1971.

74. Gupta, J. N. D., "An Improved Combinatonial Algorithm for the Flow Shop Scheduling Problem," Operations Research, Vol. 19, No. 6, November-December 1971, pp 1753-1758.

75. Gupta, J. N. D., "Optimal Scheduling in a Multistage Flowshop," AIIE Transaction, Vol. 4, No. 3, September 1972.

76. Hardgrave, W. W. and Nemhauser, G., "A Geometric Model and Graphical Algorithm for a Sequencing Problem," Operations Research, Vol. 11, No. 6, November 1963.

BIBLIOGRAPHY (Continued)

77. Harding, John, Gentry, David, and Parker, Jerry, "Job Shop Scheduling Against Due Dates," Journal of Industrial Engineering, Vol. 20, No. 6, June 1969.

78. Harris, Thomas J., "Shop Scheduling and Dispatching Concepts," American Production and Inventory Control Society Conference Proceedings, 1964, pp 186-203.

79. Harty, James D., "Problems and Principles of Controlling Work-in-Process," American Production and Inventory Control Society Conference Proceedings, 1966, pp 125-135.

80. Held, M. and Karp, R. H., "A Dynamic Programming Approach to Sequencing Problems," Journal of Industrial and Applied Mathematics, Vol. 10, No. 1, March 1962, pp 196-210.

81. Heller, J., "Some Numerical Experiments for an M x J Flow Shop and its Decision Theoretical Aspects," Operations Research, Vol. 8, No. 2, March-April 1960, pp 178-184.

82. Heller, J. and Logemann, G., "An Algorithm for the Construction and Evaluation of Possible Schedules," Management Science, Vol. 8, No. 2, 1962.

83. Heuser, W. A., Jr. and Wynne, B. E., Jr., "An Application of the Critical Path Method to Job Shop Scheduling - A Case Study," Management Technology, Vol. 3, No. 2, December 1963, pp 129-135.

84. Holstein, William K., "Production Planning and Control Integrated," Harvard Business Review, May-June 1968, pp 121-140.

85. Holstein, William K. and Berry, William L., "Work Flow Structure: An Analysis for Planning and Control," Management Science, Vol. 16, No. 6, February 1970, pp 324-330.

86. Holt, C. C., Modigliani, F., Muth, J. F., and Simon, H. H., Planning Production, Inventories and Work Force, Prentice Hall, 1960.

87. Hottenstein, M. P., Models and Analysis for Production Management, Chapter 8, International Textbook Company, Scranton, Penn., 1968.

88. Hottenstein, M. P., "Expediting in Job-Order Control Systems: A Simulation Study," AIIE Transactions, Vol. 2, No. 1, March 1970.

89. Ignall, G. and Schrage, L., "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, Vol. 13, No. 3, 1965, pp 400-412.

BIBLIOGRAPHY (Continued)

90. Jackson, J. R., "An Extension of Johnson's Results on Job-Lot Scheduling," Naval Research Logistics Quarterly, Vol. 3, No. 3, September 1956.

91. Jackson, J. R., "Networks of Waiting Lines," Operations Research, Vol. 5, No. 4, August 1957, pp 518-521.

92. Jackson, J. R., "Job-Shop Like Queuing Systems," Management Science, Vol. 10, No. 1, October 1963, pp 131-142.

93. Jackson, J. R., "Queues with Dynamic Priority Disciplines," Management Science, Vol. 7, No. 1, pp 18-34.

94. Johnson, S. M., "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol. 5, No. 3, April 1959.

95. Jones, C. H., "Parametric Production Planning," Management Science, Vol. 13, No. 11, July 1967.

96. Karush, W. and Moody, L. A., "Determination of Feasible Shipping Schedules for a Job Shop," Operations Research, Vol. 6, No. 1, January-February 1958, pp 35-55.

97. Karush, W., "A Counterexample to a Proposed Algorithm for Optimal Sequencing of Jobs," Operations Research, Vol. 13, No. 2, 1965, p 323.

98. Lawler, E. L. and Wood, D. E., "Branch and Bound Methods: A Survey," Operations Research, Vol. 14, No. 4, July-August 1966, pp 699-719.

99. Lawler, E. L. and Moore, S. M., "A Functional Equation and its Application to Resource Allocated and Sequencing Problems," Management Science, Vol. 1y, No. 1, September 1969.

100. Le Grande, Carl, "The Development of a Factory Simulation System Using Actual Operating Data," Management Technology, Vol. 3, No. 1, May 1963, pp 1-19.

101. Lev, Benjamin, "On the Loading Problem, A Comment," Management Science, Vol. 18, No. 7, March 1972, pp 428-431.

102. Link, Philip A., "Successful Control Over Parts Manufacturing," American Production and Inventory Control Society Journal, Vol. 2, No. 4, October 1961, pp 27-41.

BIBLIOGRAPHY (Continued)

103. Lomnicky, Z. A., "A Branch and Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," Operations Research Quarterly, Vol. 16, No. 1, 1965, pp 89-100.

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

105. McMahan, C. B. and Burton, B. G., "Flow Shop Scheduling with the Branch and Bound Method," Operations Research, Vol. 15, No. 3, May-June 1967, pp 473-481.

106. McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 6, p 1.

107. Marchban, J. L., "Daily Automatic Rescheduling Technique," Journal of Industrial Engineering, Vol. 17, No. 3, March 1966.

108. Maxwell, W. L., "The Scheduling of Economic Lot Sizes," Naval Research Logistics Quarterly, Vol. 11, No. 2, June 1964.

109. Maxwell, W. L., "On Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs," Management Science, Vol. 16, No. 5, January 1970.

110. Mayhugh, J. O., "On the Mathematical Theory of Scheduling," Management Science, Vol. 11, No. 2, November 1964, pp 280-288.

111. Mellor, P., "A Review of Job Shop Scheduling," Operational Research Quarterly, Vol. 17, No. 2, June 1966, pp 161-172.

112. Miller, L. W. and Schrage, L., "The Queue M/G/1 with the Shortest Remaining Processing Time Discipline," Operations Research, Vol. 14, 1966, pp 670-683.

113. Mitten, L. G., "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol.

114. Montagnon, C. E., "Simulating Scheduling Plans," Proceedings of the Third Conference on Applications of Simulation, December, 1969.

115. Moodie, C. L. and Novotny, D. J., "Computerized Scheduling and Control Systems for Discrete Part Production," Journal of Industrial Engineering, July 1968, pp 336-341.

116. Moore, J. C. and Wilson, R. C., "A Review of Simulation Research in Job Shop Scheduling," Production and Inventory Management, Vol. 8, No. 1, January 1967, pp 1-10.

BIBLIOGRAPHY (Continued)

117. Moore, J. M., "An N Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science, Vol. 15, No. 1, September 1968, pp 102-109.

118. Muth, J. F. and Thompson, G. L., Industrial Scheduling, Prentice Hall, Englewood Cliffs, N. J., 1963.

119. Naylor, T. H., Burdick, D. S., and Sasser, W. E., "The Design of Computer Simulation Experiments," editor Thomas H. Naylor, Duke University Press, Durham, N. C., 1969.

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

121. Naylor, T. H. and Finger, J. M., "Verification of Computer Simulation Models," Management Science, Vol. 14, No. 2, October 1967, pp 92-106.

122. Nelson, Rosser T., "Labor and Machine Limited Production Systems," Management Science, Vol. 13, No. 9, May 1967, pp 648-671.

123. Nelson, Rosser T., "A Research Methodology for Studying Complex Service Systems," AIIE Transactions, Vol. 1, No. 2, June 1969, pp 97-105.

124. Page, E. S., "An Approach to the Scheduling of Jobs on Machines," Journal of the Royal Statistical Society, Vol. 23, No. 2, Series B, 1961.

125. Panwalkar, S. S., Dudek, R. A., and Smith, M. L., "Sequencing Research and the Industrial Scheduling Problem," Symposium on Scheduling, Raleigh, North Carolina, May 1972.

126. Parkinson, G., "Simplified Computer Control - The Mechanics of the New Production Control System," Factory, October 1966, p 98.

127. Pegels, C. C., "Work Force Planning for the Job Shop," The Logistics Review, Vol. 5, No. 21, January-February 1969.

128. Pierce, J. F. and Hatfield, D. J., "Production Sequencing by Combinatorial Programming," IBM Cambridge Scientific Center Report, International Business Machines, Cambridge, Massachusetts, 1966.

129. Presby, J. T. and Wolfson, M. L., "An Algorithm for Solving Job Sequencing Problems," Management Science, Vol. 13, No. 8, 1967.

BIBLIOGRAPHY (Continued)

130. Pritsker, Alan B. and Kiviat, Philip J., Simulation with GASP II, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

131. Pritsker, A. B., Miller, L. W., and Zinkl, R. J., "Sequencing n Products Involving M Independent Jobs on M Machines," AIIE Transactions, Vol. III, No. 1, March 1971.

132. Randolph, P. H., "Constructing a Scheduling Linear Program Model," Journal of Industrial Engineering, March-April 1959.

133. Reed, Ruddell, Jr., "Plan Handling Systems with Network Diagrams," Journal of Industrial Engineering, January 1969, pp 27-31.

134. Reinitz, R. C., "On the Job Shop Scheduling Problem," Industrial Scheduling, J. F. Muth and G. L. Thompson, Eds., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963, Chapter 5.

135. Reiter, Stanley, "A System for Managing Job Shop Production," Journal of Business, University of Chicago, July 1966, pp 371-393.

136. Rowe, Alan J., "Toward A Theory of Scheduling," Journal of Industrial Engineering, Vol. 11, No. 2, March-April 1960, pp 125-136.

137. Salveson, M. E., "A Problem in Optimal Machine Loading," Management Science, Vol. 2, April 1956, p 232.

138. Sandeman, P., "Empirical Design of Priority Waiting Times for Job Shop Control," Operations Research, Vol. 9, No. 4, July-August 1961, pp 446-455.

139. Sasser, W. E., Graham, D. A., Burdick, D. S., and Naylor, T. H., "The Use of Sequential Sampling in Simulation Experiments," Presented at the Summer Simulation Conference, June 10-12, 1970 in Denver, Colorado.

140. Schmidt, J. W. and Taylor, R. E., Simulation and Analysis of Industrial Systems, R. D. Irwin, Inc., Homewood, Illinois, 1970.

141. Schmit, H. E., "Shop Flow Control," American Production and Inventory Control Society Proceedings, 1964, pp 204-211.

142. Schrage, L., "A Proof of the Optimality of the Shortest Remaining Processing Time Discipline," Operations Research, Vol. 16, No. 3, May-June 1968.

143. Schremer, M., "Computer Control of Discrete Parts Manufacturing," Automation, June 1967, pp 62-67.

BIBLIOGRAPHY (Continued)

144. Schussel, G., "Workload Balancing and Inventory Minimization for Job Shops," Journal of Industrial Engineering, Vol. 19, No. 4, April 1968, pp 194-202.

145. Schussel, G., "Job Shop Lot Release Sizes," Management Science, Vol. 14, No. 8, April 1968, pp 449-472.

146. Shwimer, J., "On the N-Job, One Machine, Sequence-Independent Scheduling Problem with Tardiness Penalties: A B and B Solution," Management Science, Vol. 18, No. 6, February 1972.

147. Sisson, Roger L., "Methods of Sequencing in Job Shops - A Review," Operations Research, Vol. VII, No. 1, January-February 1959, pp 161-171.

148. Sisson, R. L., "Sequencing Theory," Operations Research, Vol. 7, No. 1, January-February 1959.

149. Smith, Richard D. and Dudek, Richard A., "A General Algorithm for Solution of the N-Job M-Machine Sequencing Problem of the Flow Shop," Operations Research, Vol. 15, No. 1, January-February 1967, pp 71-82.

150. Teuton, O. F., Jr., "Optimal M-Stage Production Schedules when No Passing is Permitted," Operations Research, May-June 1964.

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

152. Tocher, K. D., The Art of Simulation, The English University Press LTD, London, 1963.

153. Tongue, Fred M., "Assembly Line Balancing Using Probabilistic Combinations of Heuristics," Management Science, Vol. 11, No. 7, May 1965, pp 727-735.

154. Trilling, Donald R., "Job Shop Simulation of Orders that are Networks," Journal of Industrial Engineering, Vol. 17, No. 2, February 1966, pp 59-71.

155. Van Horn, R. L., "Validation of Simulation Results," Management Science, Vol. 17, No. 5, January 1971, pp 247-258.

156. Von Lanzenauer, H., "A Production Scheduling Model by Bivalent Linear Programming," Management Science, Vol. 17, No. 1, September 1970.

BIBLIOGRAPHY (Concluded)

157. Wagner, H. M., "An Integer Linear Programming Model for Machine Scheduling," Naval Research Logistics Quarterly, Vol. 6, No. 2, June 1959.

158. Wilbrecht, J. K. and Prescott, W. B., "The Influence of Set Up Time on Job Shop Performance," Management Science, Vol. 16, No. 4, December 1969.

159. Wright, Oliver G., "Input-Output Control: New Way to Cut Work-In-Process Queues," Modern Materials Handling, Vol. 25, No. 8, September 1970, pp 38-45.

VITA

Joseph C. Irastorza was born in Matanzas, Cuba and is now a citizen of the United States. He is married and has three children. His education includes a B.S.E.E. with highest honors from the Georgia Institute of Technology, M.S.E.E. (Control Systems) also from Georgia Tech, M.S.I.E. (Operation Research) from Lehigh University, and doctoral studies in I.E. at Georgia Tech. The honors obtained by Mr. Irastorza include the following:

Selected as the outstanding Electrical Engineering Senior in 1960.

Recipient of the Tau Beta Pi scholarship cup for the outstanding Georgia Tech senior in 1960.

Membership in the following Honor Societies:

Phi Eta Sigma, Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Pi Tau Sigma.

Mr. Irastorza is a senior member of the American Institute of Industrial Engineering, and a member of the Institute of Electrical and Electronic Engineers and Institute of Management Sciences.

He has worked with the Ministry of Industries in Cuba, Western Electric Company and as a Lecturer in the Industrial Engineering School at Georgia Tech. Since December 1968, he has been with Kurt Salmon Associates and is currently a Principal in its Management Systems Division.