A STATE DEPENDENT HEURISTIC METHOD OF JOB SHOP SCHEDULING

Mohamad A.El-Kilani

Ph.D. Thesis
EDINBURGH UNIVERSITY
MAY - 1992

DECLARATION

I declare that this thesis has been composed by myself and that it presents my own work.

Mohamad A. El-Kilani
Mechanical Engineering Department;
Edinburgh University;
May 1992.

ACKNOWLEDGEMENTS

I would like to thank my supervisors; ***Dr. George Alder*** *and* ***Mr. Frank Mill***, for their valuable advice and help during the preparation of this thesis. Also thanks are due to **all** those who give their help in the department specially ***secretaries*** of the department. Thanks are also due to ***Mr. M. Yousif*** for the time he spent in reviewing this work.

A special thank to my ***family; parents, brothers*** and to my ***wife*** for their financial support and for being patient till this work is accomplished.

ABSTRACT

A STATE DEPENDENT HEURISTIC METHOD OF JOB SHOP SCHEDULING

By: M. El-Kilani

The main object of this work is to develop a "Fair Delivery and Shop State Dependent scheduling dispatching rule" (FDSSD) in a job shop environment. The fairness principle could be defined by saying that the customers who came first be given a higher priority than those who came afterwards. The basic principle behind the FDSSD rule is fairness towards customers. This is to some extent taken into account by the First Received First Served and Earliest Due Date scheduling rules. The FDSSD rule, however, takes into account both the order in which orders have been received and related delivery dates. Techniques which do not consider both of these criteria can produce unnecessary anomalies. These anomalies can often be overcome by human judgement in relatively simple situation where results produced by logical scheduling may be immediately seen as unfair. The FDSSD rule introduces this moral element into logical scheduling. Because of this, much more anomalies which could adversely affect customers can be thrown out. The unnecessary unfairness within the schedule may not be apparent to management until too much work has been done to change things. Owing to the moral principles introduced within the FDSSD rule, a direct comparison (one to one) with other scheduling rules does not exhibit the complete performance of the FDSSD rule. However, some comparisons based on tardiness criteria are made. Towards this end, a computer simulation model has been developed. The computer model is named herein as "Job shop Scheduling Simulation Model" (JSSM). The model has been used in improving the procedure of In-Process scheduling of the FDSSD rule.

In contrast to the currently available scheduling rules which tend to be used, the FDSSD rule achieves a balance between the three main objectives of a production system. The objectives are: (i) to meet delivery dates,: (ii) to decrease Work-In-Process (WIP), and (iii) to increase machine utilisation. This balance compromises the Fairness Principle. The FDSSD rule uses First Received First Served rule (FRFS), delivery date consideration and state in the shop. The FDSSD rule offers a very close result if not better than some other known rules such as FRFS, FCFS and EDD rules.

Scheduling problems have been classified according to their elements - job, machine, shop and evaluation criteria. Detailed classification facilitates scheduling procedures. It consists of four levels - shop input/output, machine loading, queue sequencing and job dispatching.

Scheduling rules are classified in accordance with the above elements. A scheduling rule may work well either locally or globally. The FDSSD rule developed herein is concerned with the global performance with the customer requirement a high priority.

The JSSM has been used as a tool to investigate and compare some scheduling rules with the FDSSD rule. Also, it has been demonstrated that the procedure of scheduling may be significantly improved by the proposed model.

Finally, some conclusions and suggestions for future research are mentioned.

CONTENTS

ABBREVIATION LIST

There are a number of rules which are investigated and used to compare their results with the FDSSD rule. These rules are:

- **FASFS**: First Arrived to Shop First Served
- **FRFS**: First Received First Served rule.
- **FCFS**: First Come First Served rule.
- **EDD**: Earliest Due Date (minimum due date) rule.
- **SLACK**: Minimum Slack rule.
- **S/ROP**: Slack per Remaining operations rule.
- **S/OPN**: Slack per operational time rule.
- **WINQ**: Work In Next Queue rule.
- **DCR**: Dynamic Composite Rule.
- **TSPT**: Truncated Shortest Processing Time rule.
- **FIFO**: First In First Out rule.
- **SST**: Shortest Set-up Time rule.
- **CEXSPT**: Conditional EXpected Shortest Processing Time rule.
- **MWKR**: Most Work Remaining rule.
- **SPT**: Shortest Processing Time rule.
- **FDSSD**: Fair Delivery & Shop State Dependent rule.

Also there are a number of abbreviated terms could be listed below:

- **ISIS**: Intelligent Scheduling and Interactive System
- **JSSM**: Job Shop Scheduling simulation Model.
- **WIP**: Work-In-Process.
- **WINQ**: Work In Next Queue.
- **MAINQ**: Main-Queue in material store.
- **MINPQ**: Main-in-process-queues or local buffer. It is the machine buffer
- **Q23**: Global buffer. It is the buffer where jobs are placed if the local buffer is full.
- **SARQ**: Shop-arrival queue

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

1.2 IMPETUS FOR THE PRESENT STUDY

1.3 AIMS OF THE STUDY

1.4 PLAN OF THE STUDY

1.5 SUMMARY

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

Generally speaking, the scheduling activity has to assign each operation of a job to a specified machine on a time scale. A scheduling technique may resolve many possible conflicts between the main goals of a production system during the execution of the schedule. These main goals are:

- to offer service in quoted date,
- to utilise the plant capacity efficiently, and
- to avoid unnecessary Work-In-Process (WIP).

In general, a scheduling process is a complex task, especially in a job shop, where many machines and jobs are involved, consequently, constructing a schedule is a complicated problem [1,2]. Many problems arise during the execution of the schedule, such as high level of WIP, idleness, and lateness. An efficient schedule depends on many different factors, for example, processing time of each job, due date requirement, production level, capacity and state of a job shop. Most of these factors could be classified under customer, machine and shop requirements.

Customer requirements could be defined as what a customer needs from the system; for example, he needs his order to be delivered on time. Late jobs may lead to losing the trust of a customer in the system, especially, if a later customer is completed before those delayed jobs without providing a reasonable explanation. There are two requirements regarding the use of the machine. Firstly, a machine should be efficiently loaded but not over-loaded. Secondly, the machine utilisation should be maximised. Shop requirements are concerned with global requirements in the shop; minimising WIP and monitoring the queues at different machines.

Many of current approaches to scheduling use part of the above mentioned factors separately. Other approaches may use the local available information to optimise schedule according to the current local situation. Many scheduling techniques were suggested and simulation models were used for investigation, but despite this scheduling problems are calling for more investigation and study [3-11].

Different methods are employed in order to obtain an optimal schedule. Some of these methods are graphical, mathematical, enumeration, iteration and simulation. A graphical solution is one of the simplest methods to present the processes in the shop on a time scale. The Gantt chart is a well known example. Mathematical approaches (e.g. linear programming) could be used [12-14]. However, there are also many problems which may arise when this solution is employed; simplifying assumptions are required and many parameters could be omitted from the model. Iteration or enumeration solutions may perform well, but it is difficult to apply them to job shop scheduling problems. For example, to process five jobs through four operations in a job shop system a huge number of possible schedules will be obtained. It may become more than $(5!)^4$ in a normal situation. Therefore, iteration or enumeration methods would be difficult to be used in this matter. Simulation technique is widely used in scheduling area.

As mentioned above, the scheduling procedures in a job shop are relatively complex, and still production managers seek a solution. Waiting times of jobs inside a job shop form one of the main problems in the production system [3-6,10]. The source of such waiting times could be a result of inadequate scheduling policy. Again, waiting times could result in a high WIP inventory. A high WIP may disturb the flow of the production or may lead to the loss of some orders due to long queues. Also, it could cause an increase in the lead time of an order. Whatever the case, the situation would affect most customer delivery dates.

Many scheduling techniques are designed to perform the optimum schedule according to some measures of performance. Most of these techniques concentrate on a jobs specification. Processing time and due date are two examples of job specification. Some other rules are concerned with the information in the shop such as waiting time. Although, some of these techniques perform relatively well, they either ignore the receiving or delivery dates for each job. Also, they are not concerned with balancing between conflicts of job, machine, and shop requirements.

This study is concerned with developing a technique called the Fair Delivery and Shop State Dependent scheduling heuristic rule, hereafter is called the FDSSD rule. The FDSSD rule concerns the relationship with attitude towards the customers as well as shop requirements. The FDSSD rule incorporates First Arrived at Shop First Served (FASFS) - hereafter is called First Received First Served (FRFS), delivery date consideration, and the situation of machines and queues in the shop. The attitude towards customers may be called the Fairness principle. It takes considerable account of the order in which jobs have been received such as that no later job is dealt with at the expense of those that were received earlier.

Tests and investigations on the FDSSD rule will be carried out on a simulation model called the Job shop Scheduling Simulation Model, hereafter is called the JSSM. The JSSM is a program which is written in Fortran77 on an Unix operating system at Sun 3/50 work-station. This model includes some other traditional scheduling rules and techniques. First come First Serve, Earliest Due Date and Shortest processing time rules are examples. The JSSM also may be used to determine due dates and optimum procedures by which an optimum schedule is obtained. Also, it supports shop monitoring. More details are presented in chapter six.

1.2 IMPETUS FOR THE PRESENT STUDY:

The motivation to develop the FDSSD rule came from the frequently repeated phrases: "the customer is king" and "delivery date is a promise".

These are widely used in Japan, which is relatively one of the leaders in the field of production. By using the above two phrases, many of Japanese firms improve the confidence of their customers. The companies' strategy is clearly understood by the customer. They give to the customer the right to know about the situation of his order [15-18]. Since many of Japanese firms rely on the above simple and clear phrases or rules, everyone in a firm is aware of what is happening. Therefore, harmony is achieved within the system [19]. Similar rules were followed in Italy (SAC company)[20]. Although SAC is a service system, it was acting as a marketing section for different production systems. SAC was applying many of the Japanese strategies mentioned above. From this observation, effort has been made to find a scheduling technique to be applied in operations scheduling area, especially in the West where many companies are looking for a solution.

The FDSSD rule considers due dates as the basis for decision making while the situation in the shop is considered as well. However, in a critical situation where a former received job is going to be late, the basis becomes the receiving order of jobs. The FDSSD rule is presented in a technical, moral and reasonable way. A computer is used to show the possibility of applying it by a computer simulation.

1.3 AIMS OF THE STUDY:

The present thesis aims to contribute an answer to a number of difficulties associated with job shop scheduling. The main purpose of this research is to investigate and develop a scheduling method, by which delivery dates are met, WIP is decreased to the lowest level, and the future state of a shop is considered. This research flows in two main directions:

A- The study is to investigate some scheduling rules and to develop a technique in order to achieve the balance between customers, machines and shop requirements.

B- A computer simulation model; JSSM, is to be built to represent the job shop production system. The main objective of JSSM is to investigate and compare the traditional rules with the FDSSD rule. By this means, an optimum schedule could be obtained. JSSM could also be used to determine the most suitable delivery date of jobs. It has been designed to provide a reasonable procedure for practical use in job shop scheduling.

1.4 PLAN OF THE STUDY:

The study comprises of nine major sections:

- **Chapter 1** consists of the current introduction.
- **Chapter 2** presents a survey of the main literature covering production operations scheduling rules and problems associated with them.
- **Chapter 3** discusses the job shop scheduling problem environment and states the nature of the problem which concerns this study.
- **Chapter 4** job shop scheduling rules are classified and discussed.
- **Chapter 5** the framework of the developed technique FDSSD is presented in detail.
- **Chapter 6** provides a description of the possibility of using computer simulation in

investigating scheduling rules. JSSM is presented. A discussion of an application on JSSM is made.

- **Chapter 7** states the experimental environment and experiments.

- **Chapter 8** consists of a discussion of the investigation.

- **Chapter 9** features the conclusions drawn from the study and outlines some directions for future extensions (further research).

1.5 SUMMARY:

This study explores attitudes towards the customers in conjunction with state dependent procedures. This could be considered as another dimension in production scheduling technique. The FDSSD rule, the proposed one, includes the above mentioned attitude towards the customers in combination with known rules such as the FRFS and Due date based rules. It aims to achieve a balance between a job, machine and shop requirements. The FDSSD rule aims to meet quoted delivery dates, decrease WIP and increase shop performance. These aims and the concern with customer could be established, in this study, under moral dimension. The FDSSD rule introduces this moral element into scheduling procedures. Investigation will be carried out on a developed, herein, simulation model called the JSSM.

CHAPTER 2
REVIEW OF LITERATURE

2.1 INTRODUCTION

2.2 LITERATURE REVIEW

2.3 RELEVENT LITERATURE REVIEW

2.3.1 SCHEDULING PROBLEMS LITERATURE

2.3.2 SCHEDULING RULES LITERATURE

2.3.3 SIMULATION LITERATURE

2.4 SUMMARY

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION:

In the past three decades there has been a sizable body of work in the area of production scheduling. Many of the previous research efforts have been concerned with developing, optimising or evaluating an effective scheduling rule to decrease or eliminate scheduling problems. Late jobs and high WIP are examples of scheduling problems. In other words, the main aim has been to obtain more efficient scheduling rule for job shop production system in order to keep commitment to delivery dates valid, to minimise WIP, and to achieve a high machine utilisation. In spite of the above mentioned studies, there still remains room for investigation to be carried out with more consideration for the real environment. Also, effort has been made to give greater consideration towards the customer's requirement in scheduling decisions. In the current work, customer requirement is taken as the moral consideration of fairness. Unfortunately, no reference whatsoever was found that dealt with the scheduling problem under such moral considerations.

Since scheduling rules have a large influence on different measures of performance in a production system, previous studies have drawn attention to the need for an effective production scheduling rule

[8,21,22]. In their totality they contributed towards establishing a rule to obtain optimum scheduling procedures. Generally speaking, each study has different standards and measures to achieve their goals.

Many studies investigated scheduling rules required to build an optimum schedule, while others developed scheduling rules. Scheduling rules are varied in their complexity, applicability, and efficiency. Chapter 4 will discuss scheduling rules in more detail.

In this chapter efforts will be made to present some of the major scheduling literature. The more general literature is reviewed first of all, followed by a review of more specialised literature more closely related to the subject of this thesis. Some problems are mentioned such as long waiting times which may be a result from a high WIP problem. Finally, some of the simulation literature is reviewed. Most of the studies in the scheduling area, including this one, use the simulation technique (see Fig 1).

2.2 LITERATURE REVIEW:

The problem of scheduling has been treated differently by different researchers. Mathematical procedures are one of many methods used to sort out scheduling problems - linear programming is an exam-

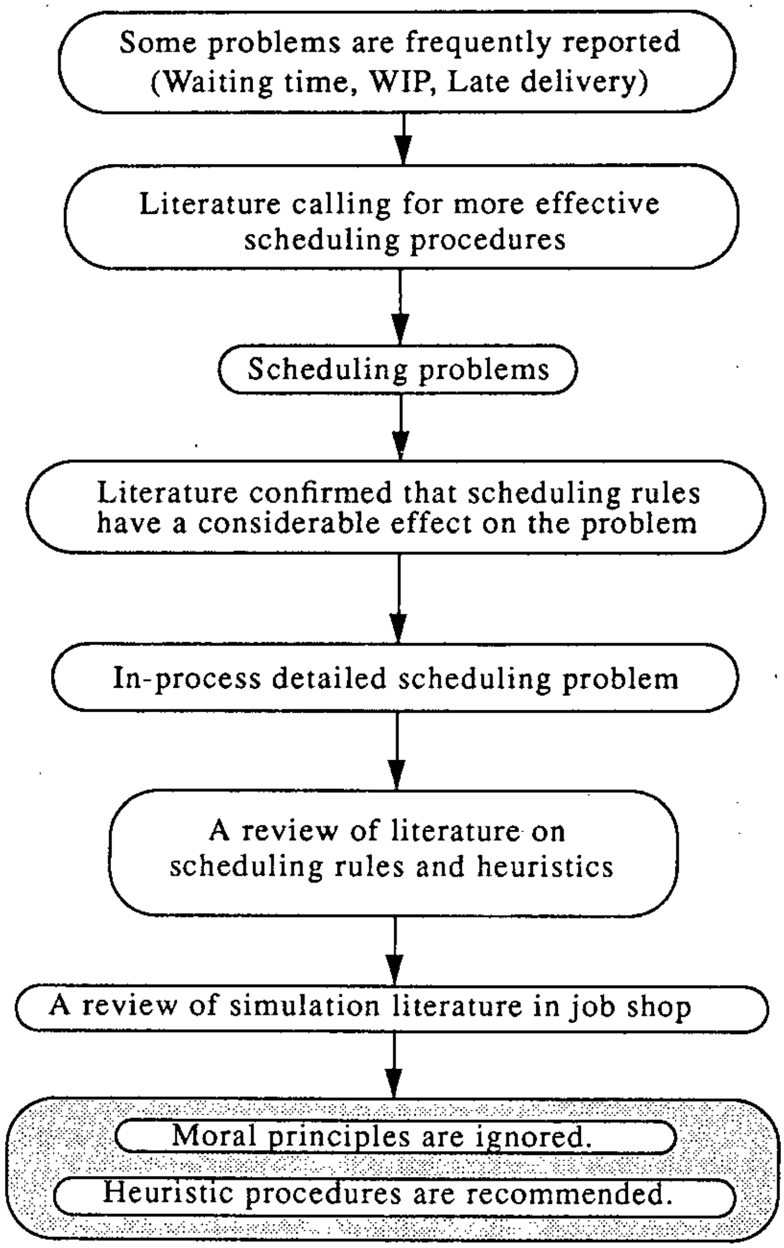


Fig 1 The flow of literature presentation in chapter 2

ple [21,23-25]. Dudek et. al. [26] used mathematical methods to evaluate each sequence of jobs in order to minimise the idle time on the latest machine. He ignored due date completely. Although the scheduling problem could be solved mathematically, this type of approach is not recommended, especially for large size of scheduling problems. Most references explain the pitfalls of using the mathematical approach - too many parameters, complex relationships and the omission of some important parameters [21,27,28].

Enumeration procedure is another approach. However, it is not recommended for large scale problems because of the huge number of iterations required. This can also be true even for smaller problems [29-31].

Simulation technique is another well known method. Computer simulation has been highly considered, as a tool to illustrate and evaluate different scheduling rules, in a number of previous studies [7,11,22,32-36]. Many researchers in the scheduling area have recognised the simulation technique as an essential and effective method to study and represent real job shop scheduling [21,35-39]. Moore et. al. [32] and Kiran et. al. [33] have made surveys of simulation studies in the job shop environment. By using simulation techniques, many scheduling rules can be investigated, improved, experienced or developed.

Many studies have pointed to the large impact of scheduling rules on the schedule performance

[8,21,22]. The effectiveness of scheduling rules depends considerably on chosen measures of performance criteria [22]. The results of different studies were presented by Blackstone et. al. [21] in terms of measurement criteria which was used in making a comparison between results. Authors have found that the cost based criteria have been most highly considered, followed by tardiness, lateness and then inventory measures.

Most of the previous studies have concentrated on either cost based, time based or/and inventory based. Most of the literature has ignored the moral aspect, i.e. giving fair consideration to the customer with more respect towards the date when the order was placed and the due date, even if the order may be nominal. The FRFS rule for example, serves orders blindly without any consideration to either delivery dates or machine and shop requirements.

Scheduling problems could seem to have been excessively researched since the 1960's, but despite this, many researchers are still calling for an effective solution to scheduling problems [3-8,10, 11].

2.3 RELEVENT LITERATURE REVIEW:

In the following section, efforts are directed towards discussing the related review to the scheduling problem (e.g. Late delivery, long waiting time in the shop, high WIP). Some literature refers to the Japanese policy in order to describe how the Japanese treat their customers as regards moral respect for their requirements, confidence and promise. Some others highlighted the importance of customer satisfaction. Then a review of scheduling rules literature is provided followed by a general review of simulation literature. The sequence of review flows according to the structure of this thesis - scheduling problem followed by scheduling rules and heuristics then finally, it ends with simulation literature.

2.3.1 SCHEDULING PROBLEMS LITERATURE:

In-process waiting time problem is repeatedly mentioned in recent literature [3-5]. The productive time i.e - actual time spent in processing operations, is less than 10% [3-5]. In job shops, jobs wait in queues as each job is moved from one machine to another. WIP consequently increases. As a result, scheduling jobs, which form queues, become complex. As WIP increases in the shop, more time spent waiting is anticipated. Schroeder [40] pointed out that the waiting time spent in queues for one job may become as much as 95% of its total production time.

Hon [5,18], Stommel [6] and others [3,4], between them have concluded that the productive time in a system is usually only about 5% to 10% of total time. The remaining time consists of queuing, waiting, and non-productive events. High waiting times in the system may lead to increased WIP in the system. Therefore, studying waiting time and WIP problems may require a direct effort to investigate scheduling problems because of the close relation between timing and the scheduling procedures.

Late delivery is another problem which may result from a high WIP [38,40]. Schroeder [40] has discussed decision making in operation management. A simulation supplement was provided. He highlighted several points. Some of these points emphasize the importance of the delivery dates:- "due dates seem to have more importance than efficiency and flow time".[p 371]. He added that poor delivery performance could occur if there is lack of cooperation between the marketing and operation people [p 365]. Consequently, a strategy of scheduling should be designed with the customer in mind.[p 135]

Cantellow et. al. [38] have drawn attention to the importance of promised delivery dates. Consequently, tardiness based criteria are recommended to be used as a measure of performance. A general concept of model building is introduced. Computer simulation is discussed systematically to represent a real model. An actual production environment is presented.

The points highlighted by the above mentioned literature lead to the following conclusion. High WIP may lead to a considerable amount of waiting time and a noticeably late delivery may be expected as consequence. Drawing from this summary and from some other literature [41], scheduling procedures are seen to have a great effect on scheduling problems [21,22,42,43]. Efforts are directed to investigate scheduling rules in order to find an effective procedure to manage the flow and the sequence of jobs.

An attempt to review production scheduling has been made by Graves [44]. A broad classification for different scheduling problems has been presented. Three classes of production scheduling problems have been proposed:

1- Requirement generation - open shop or close shop.

2- Complexity of shops - one machine, parallel machines and flow or job shop.

3- Scheduling evaluation criteria - schedule cost or performance.

Furthermore, the job shop environment is considered as the most general production scheduling problem which still requires more effective study [44].

Conway et.al. [37] have provided a discussion of scheduling problems and their classification followed by a presentation of measures for schedule evaluation and some of the solution methods to the scheduling problems.

2.3.2 SCHEDULING RULES LITERATURE:

Three major surveys related to scheduling rules have been made:- Moore et. al. [32], Panwalkar et. al. [23] and Montazeri et. al. [45]. Further to this, a wide body of research exists for job shop scheduling and dispatching rules [8,21,33,37,39,44,46].

Conway, Maxwell and Miller [37] developed a form of state dependent rule and Work In Next Queue (WINQ) rule in such a way to be aware of other machines in the shop. They discussed the Dynamic Composite Rule which represents a more involved form of the state dependent rule. It combines operation due date, operation processing time, work in the current queue and work in next queue, relative to total load in the shop. In determining the best parameter values for the priority index function, they have used an experimental search. In their book they summarise basic scheduling rules. It is organised according to the type of scheduling problems rather than the techniques of solution. An introductory chapter to job shop simulation is provided.

A combination between the SPT and the FCFS rules is discussed by Conway et. al. [37]. This combination can be made in order to consider the SPT rule as the basis in which case then it uses the FCFS rule if a specified limit of waiting time is exceeded by a job. Alternating the FCFS rule can be considered as the basis, and then the SPT rule is used when the number of jobs waiting in a queue reaches a certain limit. Using this combination with the FCFS rule may decrease the number of very late jobs, but this modification does not give the procedure the consideration of the moral factor - i.e. no later job is served at the expense of another earlier job. Furthermore, the above combination ignores delivery dates.

A review of sequencing research has been provided by Day et. al. [39]. They have provided a classification of sequencing literature in terms of the number of job components, production facilities and job availability.

Panwalkar et. al. [23] have categorised and described a summary of 113 priority rules which are used in more than 30 studies and a list of many references that deals with these rules. These rules have been classified and presented clearly under three categories:

- Simple priority rules: they are based on job specification such as due date and receiving time. They could be combined with another simple rule or with different weight values for each job.

- Heuristic rules: they are more complex than the simple rules because they involve complex considerations such as machine loading anticipation. They also may involve human decision.

- Other specific rules which may be designed for a special purpose or a combination of previous categories could be considered under this third category.

Blackstone, Phillips, and Heisterberg [21] presented a literature review of dispatching rules in their study. This study included two main sections. The first section discussed the methodology of dispatching rule development. The second section discussed the relative performance of some dispatching rules in order to identify the best rule. This identification is carried out through simulation. Their study presented results of previous studies in terms of the measurement criteria used. In comparing these results, the heaviest consideration is given to cost based criteria, then tardiness, lateness and flow time. They themselves also concluded that the elements of simulation process in a job shop - distribution of job receiving rate, due date assignment method [41,47], and shop size - may not have a significant influence in evaluating the effectiveness of the dispatching rules relative performance. Some of recommended rules are the SPT, EDD and FRFO rules.

Moore and Wilson [32] reported a number of simulation studies made between 1961 and 1967 in job shop scheduling. They also presented some results of simulation for different dispatching rules with respect to the consideration given to various measures of performance. A general dispatching rule classification is made according to the time dependency of a rule - static and dynamic, and according to the type of information: local and global. They stated that the effect of switching between rules had received little attention. Also, they declared that more work is required in the dispatching area.

Montazeri and Van Wassenhove [45] present a wide review of previous literature. A list of many scheduling rules is provided. Performance measures and environments which were used in the earlier research were discussed.

Neelamkavi, Rao, and Thomson [8] reported a practical approach for the selection of dispatching rules by shop management. Four rules were selected and combined using weight factors which may cause the decision to be switched between the four selected rules. These four rules are the Shortest Processing Time (SPT), Slack per Operational time (S/OPN), Shortest Set-up Time (SST) and Most Work Remaining (MWKR). Buckley et.al [48] have been drawing on the finding of Ref. [8]. The decision is taken by weighing a number of cost factors associated with manufacturing activity.

A considerable amount of research on the scheduling of job shops draws one's attention to the need for effective scheduling procedures. Rowe [34] was concerned with evaluating the applicability of sequential scheduling rules. This rule depends on partitioning scheduling into three phases: loading, scheduling, and dispatching. This rule incorporates flow allowance, processing time and due date, to calculate start machining time of an order. Flow allowance depends on the value of an order.

Conway [49] was among the first researchers to analyse dispatching rules. An experimental investigation of priority assignment in a job shop is made by Conway. The investigation employs computer simulation to compare and evaluate a number of priority rules. The criteria of comparison were various measures of WIP inventory and job lateness. In 1965 Conway [11] presented part of an investigation of some dispatching rules results. He concluded that the Slack per Remaining OPeration (S/ROP) rule appears to be the best due date based rule.

In 1967 Conway et. al. [37] developed a simple form of DCR state dependent rule and WINQ rule. Dar-El et. al. [7] tested scheduling rules versus job shop performance using a computer simulation model while tardiness was set as performance measure. This study aimed to guide plant management in the selection of the appropriate priority scheduling rule. They concluded that the WINQ rule gives relatively good results in a job shop.

Wacker and Lucht [43] presented a list of evaluation criteria and performance measures which could be used to evaluate the effectiveness of dispatching rules. These criteria should be simple, understandable, and usable. For performance evaluation, ten measures are listed. Number of orders completed, percentage of orders completed on time, average waiting time of orders and average number of orders waiting in the shop are some examples. Also, they pointed out that machining start times could be used to set job priorities. As a result, a dispatching rule is suggested. This rule sets the operations' start dates by subtracting processing times from the due dates. They highlighted the importance of meeting of delivery dates. A case study has been discussed. They concluded that although high quantity and productivity are given high priority, timely delivery is the most important objective. This is because of the losses to the customers that may be caused because of late delivery.

Many studies have been directed towards combining scheduling rules in order to combine the relative advantages of each one [8,36,37,48,50,51]. Recent research carried out by Schultz [36] could be taken as an example. Schultz presents a new rule called Conditional EXpected Shortest Processing Time (CEXSPT). It employs the SPT and EDD rules in order to control late or behind schedule jobs. The SPT rule is employed in a controlled manner using the EDD rule. The proposed rule partitions the waiting jobs for an operation into three queues according to whether a job is late, behind schedule or ahead of schedule. The SPT is used in the late jobs queue unless it may lead to further late jobs. Stoeva [51] extends the CEXSPT rule, which is proposed by Schultz. The extension of the CEXSPT rule incorporates shortest starting time at machine and shortest processing time.

Another modification to the SPT rule is proposed by Eilon et. al. [50]. The modified rule is called the SI^x rule. It forms two separate queues at each machine. One queue is higher priority than the other and both of them employ the SPT rule. Slack is used to decide which job is going to take the higher priority in the queue.

Efforts have been directed towards making rules more dynamic and more aware of the status of the shop and machines. A state dependent scheduling procedure is proposed [37]. Conway et. al. [37] developed a form of state dependent rule called Dynamic Composite Rule (DCR). It incorporates due date, processing time, work in the current queue and work in next queue relative to the load in the shop. Vepsalainen [52] drawing on Conway's finding, extended a new state dependent priority setting procedure for a job shop scheduling. He studied scheduling with due dates with jobs that had a different tardiness penalty. A Slack evaluation method was developed for the "Apparent Urgency" rule and "Modified CoverT" rule. Both the Apparent Urgency and Modified CoverT rules incorporate the weight value of jobs, waiting time, due date and processing time. The main objective in a weighted tardiness problem is to minimise the total tardiness cost.

A number of heuristics or rules of thumb are also incorporated. Heuristics are a complex mixture of procedures which usually depend on previous experience. Gere [2] studies the use of heuristic scheduling procedures in job shop scheduling. He tested several Heuristics (using simulation) in order to minimise the cost of tardiness (penalty cost). Some heuristics were provided. These include alternate operation heuristics, i.e no new late jobs, insert and manipulation heuristics to fit a job on an idle machine and look-ahead heuristic to anticipate the loads in the shop. Heuristic procedures can be expected to perform effectively [53].

Fox et. al. [9] provided a study which describes ISIS (intelligent scheduling and interactive system). The ISIS is a job shop scheduling system which is capable of incorporating many constraints in the construction of job shop schedules, and employs a heuristic approach for schedule generation. The system mainly uses constraints and previous experience. They examined how constraints may be represented, and the way in which they can be used to obtain an acceptable schedule. In the ISIS, interactive scheduling facilities are considered. The work was classified under five categories: organisational goal constraints (profit), physical, causal, availability of resources and preference. Their work claims to achieve a cooperative balance between the following constraints:- due date, capacity analysis and general facilities utilisation. Relaxation of constraints could be used to resolve the conflicts between them. They focused on global optimisation of the system by finding the best schedule according to due date and profit. However, in their study they have not considered dispatching rules as satisfactory for dealing with practical scheduling problems. Relaxation of constraints could be used to resolve the conflicts between them.

Hasting et. al. [53] describe a scheduling system which uses job oriented heuristics in which all the operations of one job are scheduled before considering next job. It has been concluded that job oriented heuristic is computationally efficient for large scale problems.

2.3.3 SIMULATION LITERATURE:

Many of the previous studies which used simulation technique to investigate, evaluate or develop scheduling procedures, have been reported [2,11,34]. Hollier [54] is concerned with evaluating scheduling rules and parameter values on performance measure for a hypothetical batch production shop.

Hon [5,18] described a new sequence planning system which is known as Stabilised Sequence Planning system. The system is designed to monitor WIP and to achieve a balance in manufacturing lead times. The system (SSP) is a computerised model, i.e. computer simulation is employed. The loading procedures used depend on forward loading. The author pointed to further development that could be carried out such as backward loading. The procedure of loading uses order-oriented heuristics.

Emery [55] designed a simulation model in order to minimise earliness and tardiness costs in a static job shop scheduling problem using optimum-seeking procedures. These procedures were used to search for the optimum values of parameters. These parameters are used to switch between a variety of priority rules. The involved rules vary from simple priority rules to composite rules which may consider many criteria together.

Worrall, Bancroft and Sivanesan [1] are concerned with developing dispatching rules based on the industrial environment of the job shop. These rules incorporate customer importance and rush orders. The authors depend in their study on rules which involve processing time. An external priority is created then combined with the dispatching rule to form the final parameter for decision. A simulation model is developed to compare between suggested rules and the developed rule. "First In First Out" (FIFO) rule and "Shortest Processing Time" (SPT) rule are among these suggested rules. They pointed out the difficulty involved in handling job shop scheduling problems.

Emery [24] confirmed that the problem of scheduling is complex and massive. He pointed out the effect of this problem on: poor delivery performance, increased capital requirement due to high work in process, and lowered shop morale. The use of a computer was proposed in order to improve scheduling procedures. The paper concludes that priority scheduling through job shop simulation appears to contribute a promising approach towards solving such a problem.

Most researchers used simulation in their studies [5,18,24,34,38,54,56-59]. Conway et. al. [37] provided an introduction on simulation. Moore et. al. [32] provided a simulation research survey in a job shop. Wight [60] and others [61] recommend using a computer in scheduling procedures. Kiran et. al. [33] made a survey of simulation studies in job shop scheduling problems according to shop performance evaluation based on: job completion times, due dates or costs. Also, the methods that are considered within those simulation studies, such as criteria and priority rules, have been discussed.

2.4 SUMMARY:

The review of literature has shown that job shop scheduling rules have received a fair amount of attention from researchers. Unfortunately, most of this previous research has ignored the moral factor except for the blind approach of the FCFS or FRFS rules. The main concern of these studies is directed towards either the shop, the machines and/or (sometimes) customers - in order to keep the cost of production as low as possible. Consequently, commitment to meet due dates is mainly related to the status of the customer.

Wight [60] pointed out that the simple decisions which support human behaviour should be recommended - "and support man rather than supplant him". The scheduling rule should be built according to the customers' requirements. Due dates could be of more importance than other measures of performance [40].

Good and long term relationships with customers are mentioned widely in the following literature. They attempt to find an explanation of the Japanese success [16,61]. Cowhig [19], Oliver [62] and Ment [17] have pointed the harmony in the system itself. This harmony is a result of the simplicity and understandably of the rule applied, especially if the rule is near to human principles [61]. Interest to achieve customer satisfaction are increasing recently [77].

Trevor and Christie [16] made a comparative study between manufactures in Britain and Japan. It highlighted the importance of achieving good relationships, and building trust and confidence with customers. Also, the study emphasised the importance of accomplishing orders in time, and how it related to long term relations with customers. The authors presented many British and Japanese firms as case studies. A wide range of literature about Japanese successes is provided.

Generally speaking, in the 1990s, one may expect that effort will be made to standardise some decision rules between the decision maker and labour force who are responsible for executing the schedule [40]. Therefore, these rules should be simple, easily to apply internally in most levels of the system, and relatively near to human principles and thinking [60,61].

Although decreasing production cost is an important target, delivery dates should be considered as a commitment to be met. Furthermore, the order which was received earlier should be considered first in terms of meeting delivery dates.

The current work introduces a moral factor to be considered in conjunction with scheduling rules. Drawing from previous work, state dependent procedures [37,52] and heuristic procedures [2,5,18,53] are used in building the FDSSD rule - the proposed rule. Simulation technique is used as the tool to evaluate and compare scheduling procedures. Furthermore, the developed simulation model - JSSM - could be used to realise the scheduling problem in order to further develop the FDSSD rule.

CHAPTER 3

JOB SHOP SCHEDULING PROBLEMS AND THE STATEMENT OF THE PROBLEM

CHAPTER 3

JOB SHOP SCHEDULING PROBLEMS AND STATEMAENT OF PROBLEM

3.1 INTRODUCTION:

Scheduling problems differ greatly from one system to another. They tend to occur more frequently in a job shop production system because of its complexity when compared with other systems [1,44,63]. A job shop scheduling problem may be defined as assignment of time and machinery in a shop environment. The shop may contains several different machines, but in some instance they may be identical. A number of jobs have to be processed by a number of machines. Each job has its own route through machines. This routing is a sequence of operations through different machines, which is already known. The main concern in performing these operations is that the scheduling is done in such a way that following requirements are considered:

1- Job requirements: to achieve on-time delivery dates, and still to respect the sequence in which orders are received.

2- Machine requirements: to maximise the utilisation of machinery.

3- Shop requirements: to decrease in-process problems such as WIP.

A number of problems are well-known. Some of these problems are tardiness, late orders, low machine utilisation, high Work-In-Process and so forth. Many of these problems exist because of wasted time in queues and inadequate scheduling. It is confirmed that 5%-10% of total flow time is in actual production processing time [3-5,40,64].

The following sections will discuss environment of scheduling problems in a job shop. Further ahead, the statement of the problem which is the subject this research will be presented. A general description of a solution for the problem is proposed.

3.2 JOB SHOP SCHEDULING PROBLEM ENVIRONMENT:

Under this title many items could be discussed. The first item presents the assumptions which are commonly considered. The second item covers the main elements of a scheduling problem. These elements may be classified into two main categories: physical (job, machine, shop), and evaluation elements [45,64,65]. The third item discusses the measures of performance by which a problem could be evaluated. The forth item presents the classification of the scheduling problems. The final item highlights some known problems.

3.2.1 DEFINITION AND ASSUMPTIONS:

A. JOB SHOP DEFINITION:

A job shop may be defined as a manufacturing system which processes jobs in small batches by a series of operations. Each operation must be performed on the entire batch before any subsequent operation is started. Job shop production systems fall between pure jobbing and mass production systems. Job shop systems have a higher variety of jobs than mass production systems, and deal with a higher quantity of jobs than jobbing systems. (See Fig 2)

In a job shop, jobs have several set of routines. The machines' layout is organised according to type of operation which is available on a machine. This is called "Process Layout". Jobs flow through departments in batches (See Fig 3).

Because of the flexibility required to produce different types of products in relatively low quantity, low utilisation, high WIP, and long waiting times are expected. The job shop is a relatively complex system. Therefore, it requires more attention and management than other production systems [63].

B. ASSUMPTIONS:

A number of assumptions could be made. They are listed below for easy reference (see also Ref.[37]):

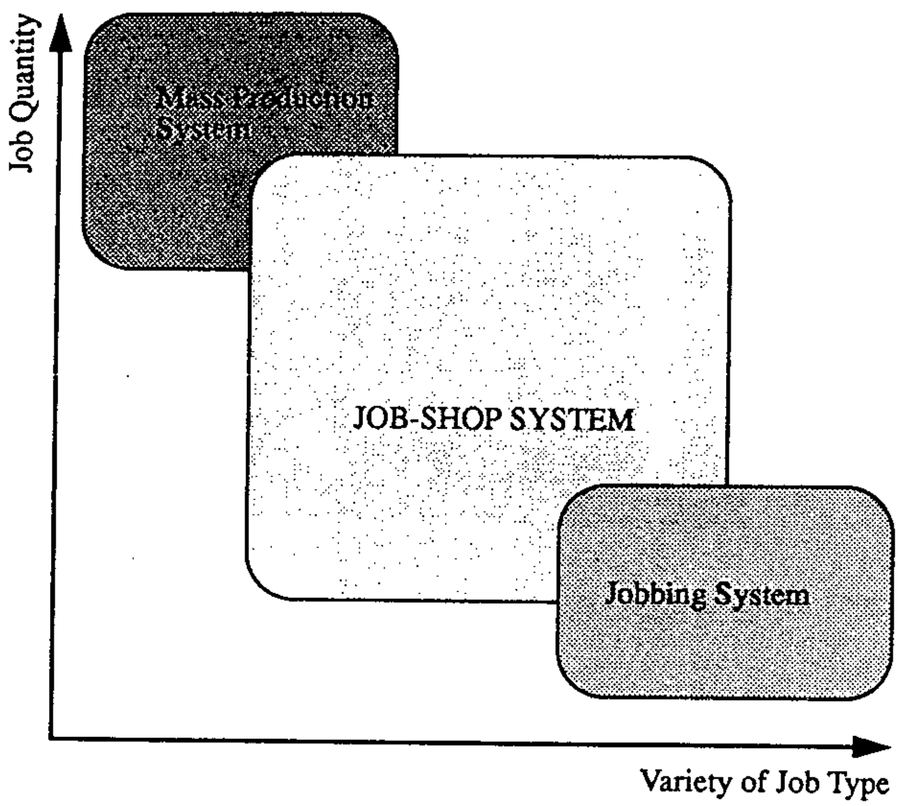


Fig.2 Production systems according to type and quantity of jobs

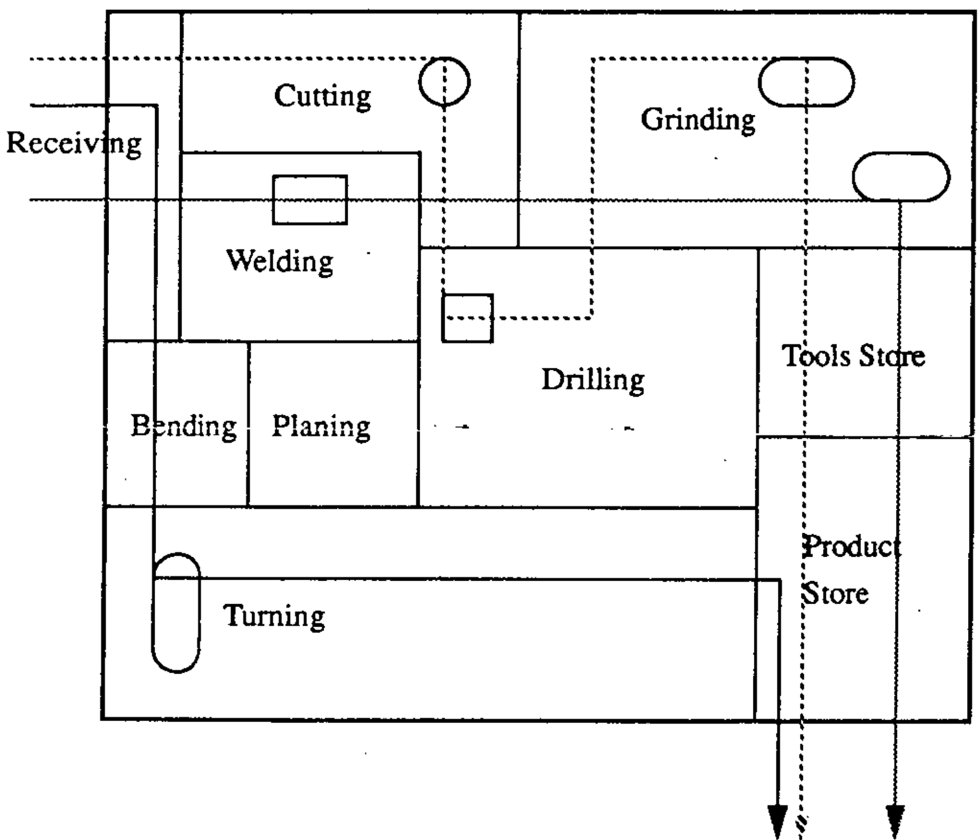


Fig .3 Process Layout of a Job-Shop

1 - Jobs are independent.

2 - No machine may process more than one job at a time.

3 - No job is processed by more than one machine at a time.

4 - Each operation processing time and routing sequence are known.

5 - Each process is independent of the sequencing order.

6 - An operation, once started, may not be interrupted.

7 - The movement between queues of jobs and setup times within the shop may be considered as a part of processing time on a machine.

8 - Each job should follow its specified route.

9 - All dates (i.e. processing times and due dates) are integers.

10- Each job can represent a lot of individual parts.

11- In-process queues are allowed.

12- All jobs are financially of equal importance.

3.2.2 SCHEDULING PROBLEM ELEMENTS:

A scheduling problem may be defined as the exercise which controls the timing of events happening on the shop floor. General scheduling problems are concerned with processing a number of jobs through a number of machines in order to achieve the main goals of the production system. The major elements of scheduling problem are:

A- PHYSICAL ELEMENTS:

1- Job condition,
2- Machine state, and
3- Shop situation.

B- EVALUATION ELEMENTS:

1- Measures of performance, and
2- Scheduling rules.

Physical elements, they may affect the other element (evaluation elements), i.e. a selected method may be more effective if it considers the physical elements in order to achieve their requirements.

A- PHYSICAL ELEMENTS:

A.1 JOB CONDITION:

Each individual job can affect the problem in different ways. Each job has a variety of parameters to be considered as an input to the scheduling procedure. This information describes the state of a job. In this study, the information primarily considered is listed below:

- Job number,
- Delivery date,
- Receiving order,
- Receiving time,
- Route on machines, and
- Operation processing times on each machine.

These items of information may collectively be called 'the job specification.'

The job specification has a considerable role to play in increasing or decreasing the complexity of the problem. For example, a job which has a long processing time may lead to a bottleneck problem. Furthermore, processing an urgent job could delay other jobs, especially in the case of a congested shop.

Generally speaking, some of the most critical states which may lead to a serious problem are:

- Dependency on a single item from the job specification,
- Having a job with a very long processing time,
- Late arrival of a very urgent job,
- By-passing of some jobs and delaying of some others,

- Prioritising orders purely from the point of view of immediate financial gain (some orders being more profitable than others),
- Having a job with a tight delivery date,
- Receiving a number of jobs at once, and
- A change in the job specification of a job after it has been scheduled or after processing has commenced.

Also, the condition of a job may be affected with any new event which occurs in the system. This new situation should therefore have an influence on the decision rule.

A.2 MACHINE STATE:

A machine or a facility is the second physical element in a scheduling problem. Again, a machine has its own specifications, e.g.:

- Code number,
- Importance of the machine,
- Capacity, and
- Available processes on each machine.

A machine could cause a notable problem because of its role in performing and producing jobs. Many problems could happen while a schedule is being carried out. Four examples are:

- Machine breakdown,
- Limited space on a particular machine.

- A bottleneck at a machine,
- Machine idleness.

The second example could cause excessive queues at machines. Take the case where three machines feed into one particular machine. If the latter machine is running to full capacity (including queues allowed) it can not accept any new orders. A backlog then occurs, and the first three machines can not operate until the problem on the later machine is cleared.

These sort of things may cause a disturbance in the shop and could lead to a sizable problem that would affect the schedule entirely. Therefore, the scheduling method should also give much consideration to this possibility.

A.3 SHOP SITUATION:

A shop usually consists of a number of machines and a number of queues in between [39,42]. These machines are ready to perform a number of jobs. A shop, as a physical element of the scheduling problem, has a great role in solving or eliminating the scheduling problem because of its general view over all other physical elements. Regarding this element, it is possible to monitor the machine acceptance and the job movement in a shop. This monitoring may take the queue length as a measure by which to adjust the situation. Also, WIP is another measure which could be considered for such monitoring. The shop situation can be divided into three areas:

- Input to the shop,
- Work-In-Process in the shop, and
- Result output.

Shop input may be divided into two types according to the pattern that how the jobs input are being received: Static or Dynamic job shop. In Static job shop, jobs are considered immediately and no more jobs are accepted until the schedule of received jobs is completed. Dynamic job shop, however, allows a continuous stream of jobs to arrive at the shop.

Shop In-process is derived from the accumulated work inside the shop which forms Work-In-Process. This WIP may cause long lead times for future jobs. Also delay may be incurred by this to some other jobs.

Finally, the output result is used to adjust the input variable, such as machining allowance time. It could also be used to adjust delivery dates in order to avoid more late jobs.

B- EVALUATION ELEMENTS:

The general form of the evaluation elements may be derived from the main goals of a production system. These goals, which may be considered to evaluate the scheduling process, are:

- Meeting delivery dates,
- Decreasing WIP, and
- Improving machine utilisation and decreasing the idleness within the system.

The evaluation element covers two main topics:

- the measure of performance and
- scheduling rules.

These two topics are inter-related. Usually scheduling rules are chosen according to their measure of performance. Section 3.2.3 will discuss measure of performance.

A scheduling rule may be called a 'priority rule'. It decides the order in which jobs are to be processed. Priority rules vary from simple rules to complex ones. They may be classified in many different ways. One classification concerns itself with the state of various criteria with regarding to time. This consists of two classes of rules: Static and Dynamic. Static rules do not change the priority of each job through time. Dynamic rules may change a job's priority as time progresses. Another classification is made according to how much information is available about the shop for decision making. Again, there are two types of rules: Local and Global. Local rules are concerned with local information on a current machine. Global rules are concerned with information regarding most machines within the shop [37,44]. More details of this will be discussed in chapter 4.

3.2.3 MEASURE OF PERFORMANCE:

Measures of performance may have different basis. Tardiness based criterion is an example. Tardiness involves due dates and completion dates. It measures those jobs which have a "positive lateness", (i.e late jobs). Another basis by which performance is measured is the cost based criteria as the expected profit. Further to this, the attitude towards customers may be considered as being relatively ignored criteria. Table 3-1 presents the mathematical formulation of some used measures in this thesis.

Scheduling is a complex process, because of the conflict between the main goals of the production system: timely delivery, low WIP inventory, and high system efficiency. One example of this conflict is that in order to achieve a high machine utilisation, stand-by jobs may be required at each machine. These stand-by jobs may create queues, i.e an increase in the WIP inventory is expected. When WIP increases, the delay of some jobs would be the result. An other example achieving a timely delivery may affect machine utilisation, i.e idle time is expected. In other words, the requirement of each element of the scheduling problem are in conflict. For example, achieving the jobs' requirement may have an impact on a machine and/or an overall shop requirement. Therefore, to build a valid and an acceptable sched-

Performance measure	formulation
Lateness (L)	Complition date - Due date
Tardiness	max(0,L)
Total tardiness	$\sum_{j=1}^{jobs} (\max(0, L_j))$
Mean tardiness	$\sum_{j=1}^{jobs} \max(0, L_j) / N$
Mean conditional tardiness	$\sum_{j=1}^{jobs} \max(0, L) / N_{late}$
Root mean square of tardiness	$\left(\sum_{j=1}^{jobs} \max(0, L_j^2) / N\right)^{1/2}$
Root mean square of conditional tardiness	$\left(\sum_{j=1}^{jobs} \max(0, L_j^2) / N_{late}\right)^{1/2}$

N= Number of jobs AND N (late)= Number of late jobs.

Table 3-1 Measures of performance

ule, performance measures should be specified clearly in order to achieve a balance between the above requirements.

Measures of performance are the tools needed to evaluate scheduling procedures. The technique of scheduling will be determined according to the chosen measure of performance. Performance criteria can be divided into three areas according to the main production system objectives.

Performance measures can be classified according to the related information and targets of the production system. In general, the classification is based on the physical elements of a scheduling problem - jobs, machines and shop - and/or based on the cost. For more detail see e.g. Ref. [8,23,45,46]. A brief classification could be presented below:

- **Performance measures related to jobs:**

  Some of these measures are average tardiness - average value of tardiness of all processed jobs, maximum tardiness among the processed jobs and total number of late jobs.

- **Performance measures related to machines:**

  The most used measures are the utilisation based measure (i.e average system utilisation and maximum utilisation) and machine idleness.

- **Performance measures related to shop:**

  Some of the measures which are commonly used are, average completion time, maximum and average WIP, maximum and average queue length.

- **Performance measures related to cost:**

  Cost penalty of an early or late completion is an example of cost based criteria. The cost of machine idleness and WIP holding are another two examples.

Generally speaking, meeting delivery dates is one of the main target of most production systems [40,41]. In the case of considering delivery dates as a basis, the job shop literature suggests several performance measures by which to evaluate the schedule. In addition to the measures described above, mean tardiness and percentage of late jobs could be considered.

3.2.4 SCHEDULING PROBLEM CLASSIFICATION:

A Job shop scheduling problem may be defined as a situation where a number of operations and jobs are required to be processed through a number of machines within a given time scale, in such a way as to optimise specific criteria or a certain goal. The optimum situation is that the delivery dates for all jobs are met and that machines' utilisation is increased at minimum production cost and low WIP.

According to this definition, a scheduling problem could be classified in the light of the primary elements which are the physical and evaluation elements (see Fig 4). This scheduling problem in a job shop has been recognised by many researchers [37,44,63-65].

Another classification could be made according to the detailed scheduling problems. This could be termed 'detailed classification' [40,66,67]. Detailed classification is concerned mainly with In-process problems such as:

- Shop input/output,
- Loading of machines,
- Sequencing in queues, and
- Dispatching of a job.

A- GENERAL CLASSIFICATION:

As mentioned above, a general classification could be divided mainly according to the elements of a scheduling problem; which are (i) physical and (ii) evaluation elements.

(i) PHYSICAL ELEMENTS:

A.1 JOB:

A job is one of the main physical elements of the problem. In a job shop each job has its own technological route, processing time and due date. In other words, each job has its own specification.

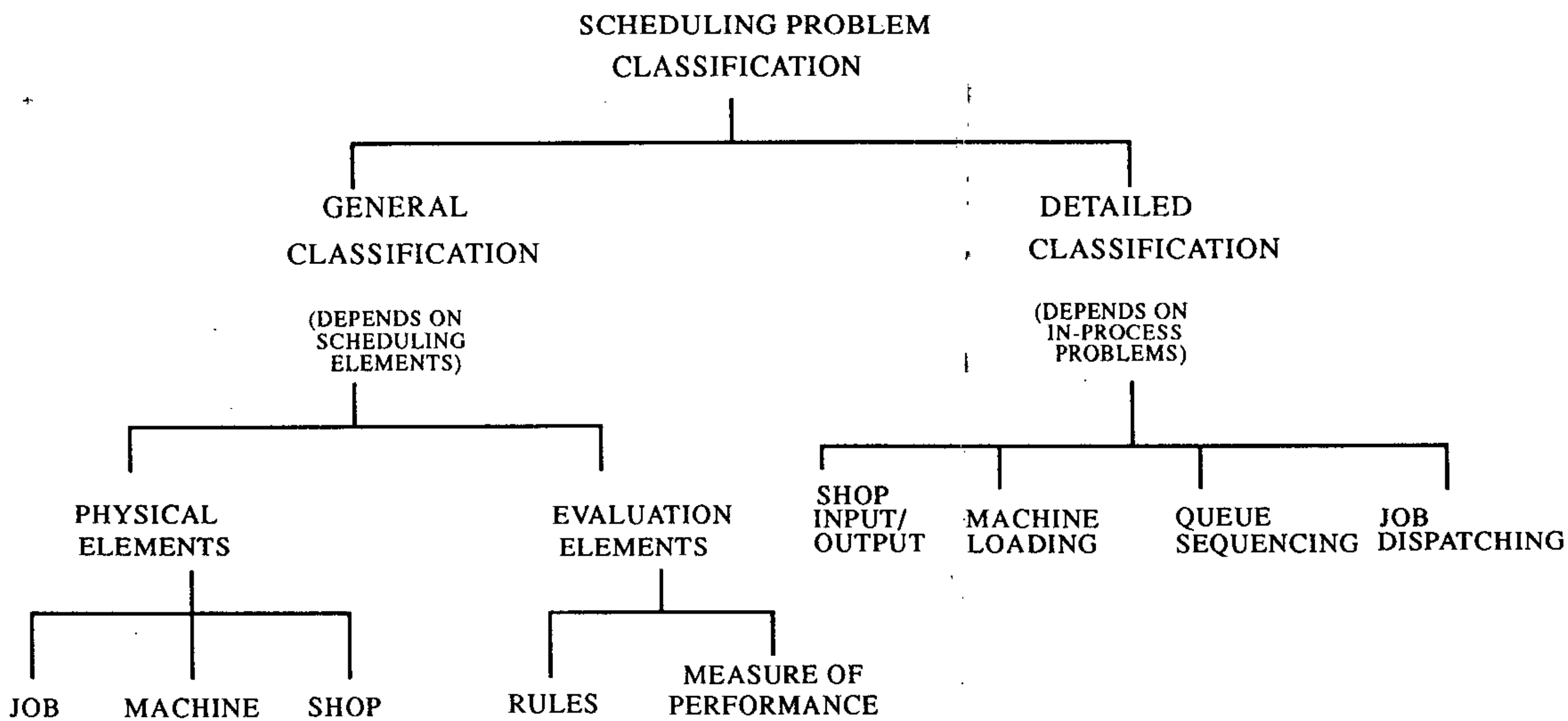


Fig 4 Scheduling problem classification

This specification may affect the problem in many instances. For example, long processing time may cause a bottleneck on a machine. It may also lead to the delay of some other orders, especially in a critical situation where many jobs have high priorities. As another example, due dates' tightness could lead to a similar problem. In addition to the role of a job's specification on the problem, the number of jobs in the system may also affect the situation.

The number of the received jobs by the system and the pattern of receiving are considered a part of the scheduling problem. However, many researchers have reported that the pattern in which jobs are received has no major effect on the relative evaluation of scheduling rules performance [46].

A.2 MACHINE:

This element is concerned with the number, type and flow structure of a machine. There are four basic structures which could be considered in this matter:

- Single machine,
- Parallel machines,
- Flow shop, and
- Job shop.

The single machine is the simplest form of a scheduling problem, because all jobs require one operation by one machine. The order of processing for different jobs is the same as the order in which they are completed (see Fig 5.a). The input sequencing order to a machine, in this case, will be equal to the output sequencing order.

The parallel machine scheduling problem is the same as for a single machine, i.e more than one machine can process the same job. Therefore, jobs are processed more frequently, and may not be completed in the same order of processing. The new problem here is to balance machine utilisation (see Fig 5.b). The sequencing order of input jobs may not be similar to the output one. It depends on the processing time of processed jobs.

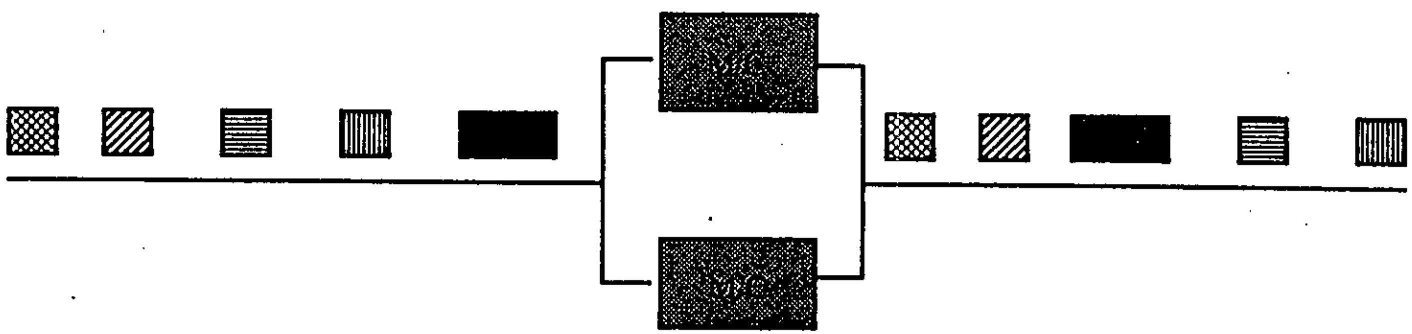
In flow shop, all jobs have an identical technological routine. Each job has to follow the same sequence on each machines. All jobs are finished in the same order in which they started on the first machine (see Fig 6). In this case the sequencing orders of input and output jobs are similar for all machines.

The job shop is the most difficult process to be scheduled [44], because there are no restrictions on jobs' routes; each job may have a unique route (see Fig 7). Input sequencing order of jobs usually is not similar to output one.

SHOP:

A shop is the area where the problem is taking place. There are two types of shops: the open and

(a) Single machine: same order of sequencing



(b) Parallel Machine: sequence depends on the processing time of each job

Fig 5 The Effect of machine on sequencing: (a) Single and (b) Parallel

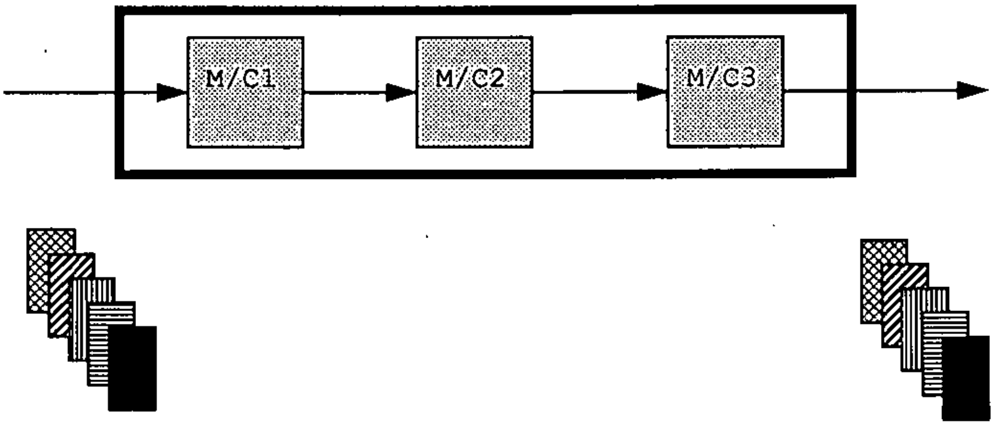


Fig 6 Flow shop: Jobs have the same sequencing order.

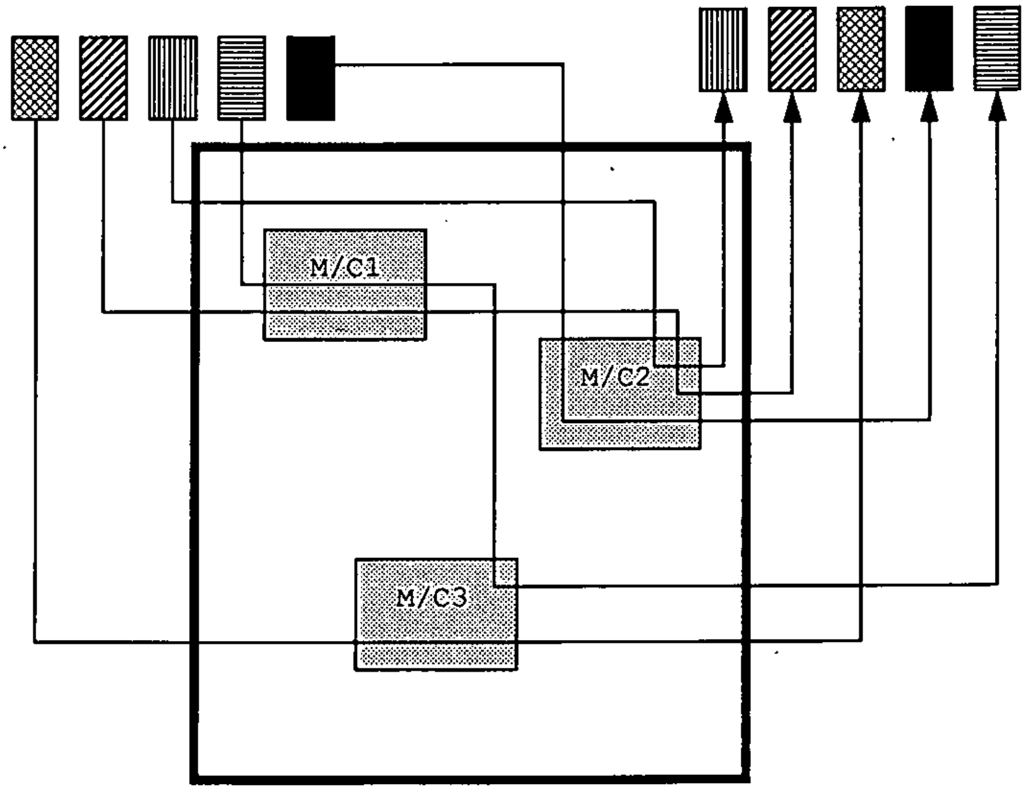


Fig 7 Job shop: Jobs have different sequencing order.

the closed shop. In an open shop, orders are received directly from the customer, and there is no stock to be used up. In a closed shop, all orders are serviced from inventory stock, which follows the replenishment process. In practice, a shop is not purely open or closed. It may be a combination of both.

In a shop, most of the more involved problems occur at shop input, machine loading, sequencing, or dispatching. These problems will be discussed below in section 3.2.4-B under detailed classification.

The shop element may be called the 'In-process' problem, because it is concerned with WIP. The main problem anticipated is high WIP, which may lead to congestion and long queues inside the shop. High WIP may also affect production cost. Furthermore, late jobs may also be expected - as a result of long waiting time.

(ii) EVALUATION ELEMENT:

As mentioned above in section 3.2.2-B and 3.2.3, this element depends on two major criteria: Scheduling rules, and measure of performance. The effectiveness of the schedule depends on the effectiveness of these two criteria. If the schedule has been built with a global rule and reasonable measure. A global rule should consider the whole situation in the shop and adjust the decision accordingly.

B- DETAILED CLASSIFICATION:

The aim of discussing classification in detail is to highlight the principal areas where a problem may occur. The detailed classification tends to explain the structure of problems which are concerned in this research. The main classes are:

1- Shop input/output,

2- Machine loading,

3- Queue Sequencing, and

4- Job Dispatching.

B.1 SHOP INPUT AND OUTPUT:

Input could be defined as the number of jobs received by a system. Input could be measured per unit of time. Jobs may rush directly into the shop causing In-process congestion. Jobs could be controlled in their rate of arrival at the shop [68-72]. Output could be defined as the rate at which jobs are completed and exit from system. This depends on the machines' capacity and WIP in the shop. Some output values could be used as a trigger to controlling jobs to be released into the shop [73]. Capacity of a machine is the maximum rate of output which can be achieved.

The relation between input, output, capacity and WIP is shown by the water system analogy in figure (Fig 8). A new order is received in tank A then

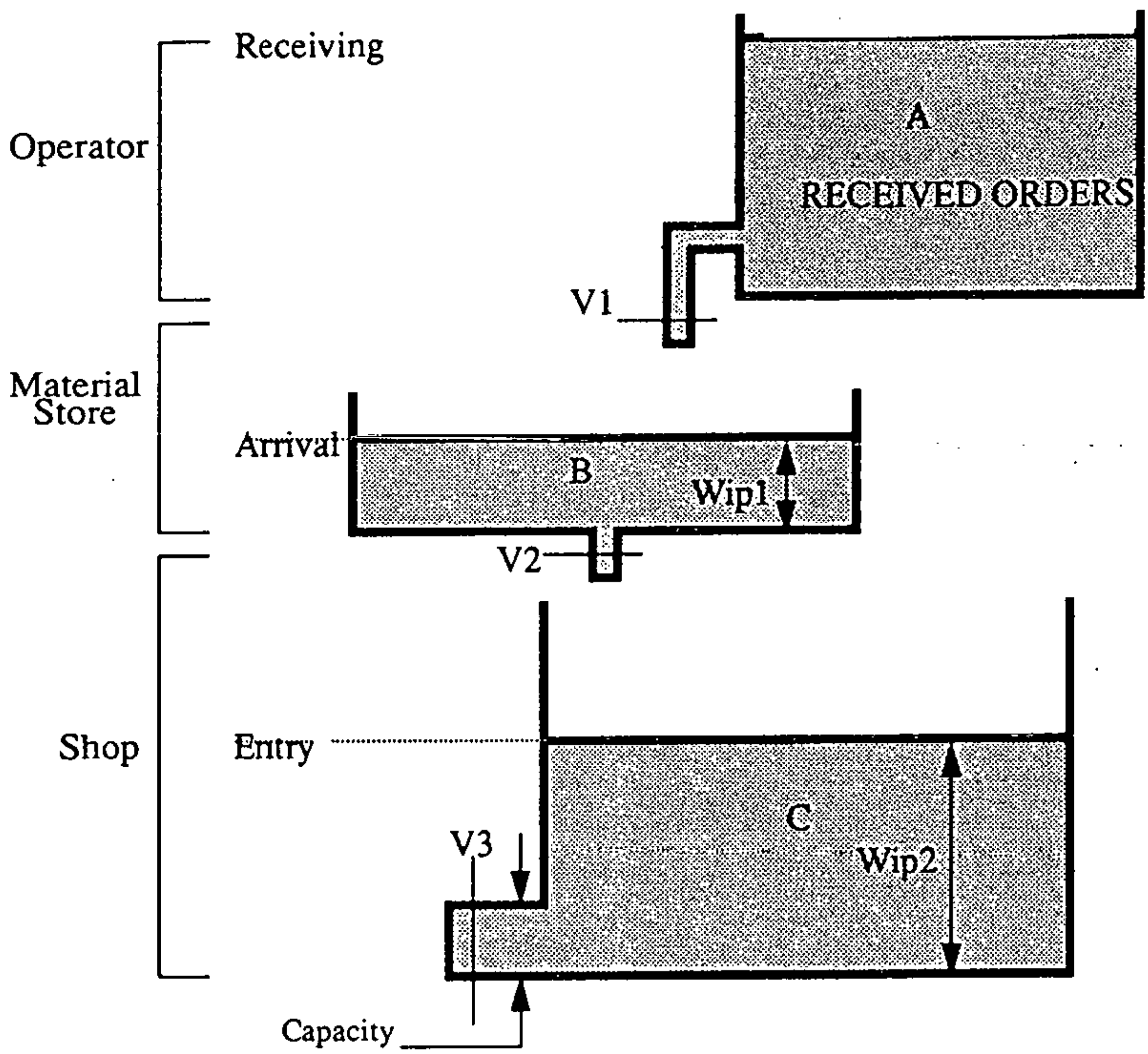


Fig 8 Water Representation of a Job Shop.

it flows into tank B through a valve (V1). Tank B represents the material store. Valve V1 control WIP tanks B and C. The valve also controls the arrival of orders. Valve V2 controls the entry of material to tank C. Tank C represents the shop. At V2, the flow represents the physical entry of orders into a shop. A discharge capacity is the maximum rate of flow at valve V3. To increase the output, it is obvious that the only way is to increase the diameter of pipe at valve V3, which represents the capacity. That could mean, increasing the number of machines in the shop (see Fig 8).

Low input to an output may cause low WIP. Idleness is then expected at some machines which means low utilisation. On the other hand, a high input may lead to high WIP. High WIP means that capital is tied up, with long queues, long waiting times, congestion, late or lost jobs, and low performance in the system. In general, the optimum situation is the steady state where input rate is approximately equal to output rate over the long term.

One method to increase output without changing the capacity is to expedite the critical jobs from WIP by the coloured tags, usually red for critical work. Coloured tags are only a short term solution because after a while the shop will turn out to be full of red or green tags and so forth [40,60].

Another means to increase the output rate without increasing capacity is by increasing the lead time. This method increases the volume of WIP in the shop. Increasing WIP inside the shop gives a chance to select a variety of different jobs. It also decreases machine idleness which leads to the maximum use of machines' capacity. Lead time is determined by subtracting the start processing time from the due date. Lead time may consist of a standard time plus a time which depends on the situation in the shop (number of In-Process jobs and on the lateness/earliness of finished jobs). This method can have much effect on the situation in the shop.

Expediting can be used to achieve a relatively good result if it is employed in a suitably organised manner. It should follow a technique to give high priority to certain jobs. If many jobs become red tagged (high priority), then it will disturb the schedule and it may expedite some jobs at the expense of others. Therefore, it is necessary to follow a scheduling rule. The scheduling rule should be concerned with decreasing the number of high priority jobs in the system.

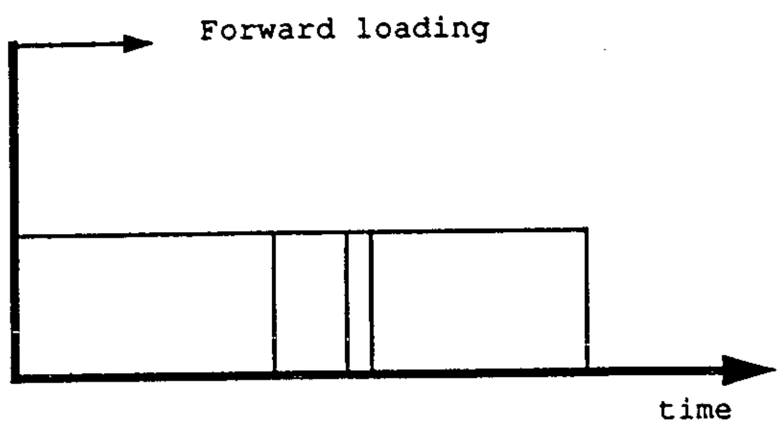
In conclusion, relating input to output rate over time helps in keeping WIP as minimum as possible. Expediting depends on the simplicity and efficiency of the scheduling rule applied. Therefore, a scheduling rule should follow up orders from the receiving point until completion. In this case expediting may have preference over increasing capacity in the system. Credit for this is due to the scheduling rule employed.

B.2 MACHINE LOADING:

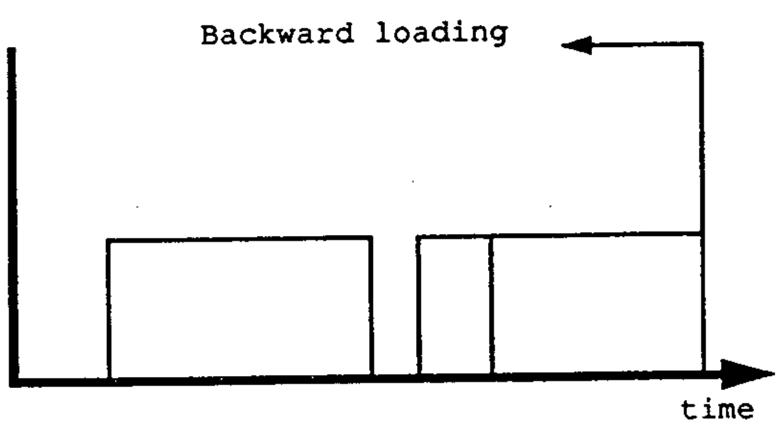
Loading is one of the first stages of scheduling procedure. When building a feasible schedule, the rate of loading of orders with reference to time on each machine is required. As soon as orders are received, they are inserted on the time scale of the machine concerned. In other words, loading may be considered as a reservation to find whether or not available capacity is enough to perform these orders. This method can be used to achieve a high equipment utilisation. There are two types of loading: forward and backward loading. They are as follow:

1- **Forward loading:** This type is concerned with the determination of the approximate completion date. It starts from present time, and loads jobs forward with reference to time according to each job's processing time (see Fig 9.a).

2- **Backward loading:** This calculates the required capacity for each machine over time. It starts from the due dates for each job and loads jobs backwards with reference to time according to each job's processing time (see Fig 9.b).



(a) Forward loading



(b) Backward loading

Fig 9 Loading: Forward and backward

B.3 QUEUE SEQUENCING:

Sequencing is concerned with developing an optimum sequence of jobs. This sequence is the order by which jobs should be processed on machines. One of the known sequencing methods is the Gantt chart method. It presents the schedule graphically. It is a horizontal bar chart showing the sequence of jobs for each machine against time.

The sequencing order is usually evaluated with respect to job and machine requirements. Makespan, minimum idleness, and delivery dates are some of fundamental measures of performance, which can be used to obtain the optimum sequencing order.

In the present study, sequencing and scheduling may use the same rules. In general, there is little difference between sequencing and scheduling. Sequencing is not concerned so much with timing, it is used to select the order by which jobs should be processed on machines. Scheduling is concerned mainly with the timing of machines and entire event within the shop. Measures of performance are concerned with the due dates and the receiving order of jobs. Inside the shop, there are further measures to be considered before choosing the next job to be loaded on a machine: machine utilisation and WIP in a shop. This will be discussed in more detail in chapter 5.

B.4 JOB DISPATCHING:

In practice, the situation in a shop often changes; machine breakdown may happen, materials are delayed, congestion occurs and late jobs result, new urgent jobs arrive, and so forth. Consequently scheduling is a difficult task. The monitoring and control of such a situation is required. The effect of these changes can be eliminated if a dynamic and flexible dispatching rule is applied.

B.5 SUB-SUMMARY:

A dispatching rule is of vital importance in solving many scheduling problems. A rule has to be understood by all levels in the system. Therefore, a simple, dynamic, and flexible dispatching rule is required to achieve a relatively optimised schedule, particularly in the long term. If a dispatching rule is simple and understandable, less effort will be required to follow up the schedule. Chapter 4 will discuss in more detail the classification of dispatching rules, and some examples will be given there. An effective scheduling procedure may combine and switch between the above procedures - controlling the input, machine loading, sequencing and dispatching - in order to follow up jobs from receiving until completion.

3.2.5 WIP AND LABOUR FORCE:

Congestion in a job shop generally means that a number of jobs have built up inside the shop forming queues. These in-process jobs are either waiting in queues or being processed on machines. The greater the number of in-process jobs the more waiting time is expected. Congestion may lead to priority congestion. Priority congestion means that many jobs are going to be late. Priority congestion in the shop is disruptive of the schedule because too many jobs have to be given top priority and the labour force may be disturbed by such a situation. Priority congestion could be difficult for the labour force to cope with. They should not be burdened with what should be managerial work, otherwise their capacity will not effectively used. That does not mean that workers should not know anything about decisions in the shop, but basically, it much better to let them know the basic decision principles so that they understand the scheduling procedures.

Secondly, the schedule which considers human nature is more likely to be performed effectively. Therefore, it is recommended that the rules used in building a schedule should bear in mind the basic ethical principles commonly employed in human decision making such as Fairness and the principle of Fair delivery.

Thirdly, ethics may generate some understanding and harmony in a shop. This harmony may higher the morale in the shop [19,60,80].

In the present research the rule developed herein - FDSSD - takes account of WIP minimisation which may affect the morale of the labour force. Minimisation of WIP is achieved by using a controlled arrival mechanism which controls the jobs flow into the shop. Loading to available capacity is the other current used method to decrease WIP in the shop. With regard to the morale of the labour force, the FDSSD rule relies on straightforward principles such as Fairness principle. As regards priority congestion where many jobs are late, the Fairness and Fair delivery principles should sort this problem out without unjust delays (see section 5.3.3).

3.2.6 TARDINESS PROBLEM:

Delivery date is the latest date that is acceptable for the completion of a job. In other words, it is the feasible due date. Delivery accuracy is one of the important aspect which customers are concerned with. Delivery performance is identified as the significant reason for attracting new customers - in addition to existing ones. Poor delivery may lead to a lack of confidence and subsequent customer loss.

The responsibility for poor delivery is put on every one in the system [40,74]. One of the most important factors is the labour who have the direct contact with material to translate a management decision into a physical reality. Therefore, the management decision should be reasonable, feasible, and morally acceptable [17,19,40,60]. Owing to the fact that scheduling rules effect the decision procedures significantly, an effort is required to inject ethics into the scheduling rules in a feasible and understandable way. The FDSSD rule promises to yield a fair and reasonable delivery performance.

3.3 THE PROBLEM AND PROPOSED SOLUTIONS:

In order to specify the problem, which is the concern of this section, it is necessary to bear in mind the general scheduling problem and the aims of the scheduling. As previously mentioned the general scheduling problem arises from a number of jobs needing to be performed on a number of machines in order to satisfy certain criteria. These criteria are chosen according to main aims of scheduling:

- Meeting delivery dates,
- Decreasing the WIP inventory, and
- Increasing machine utilisation and efficiency.

Next two sections will discuss the problems with which the current study is concerned. Also, some solutions are proposed.

3.3.1 THE PROBLEM:

Previous studies have drawn the attention to the need for improving Production Scheduling. They have been concerned with scheduling problem since the Sixties. Nowadays, many studies are still discussing similar problems if not the same ones. Most these studies carried out to find an optimum solutions [40,60].

One of the common problems is lateness among jobs. The other two common problems are the high WIP and the idleness at machines. These three factors are all quite contrary to the main objectives of most production systems (see Fig 10).

Late jobs mean that customers become dissatisfied, because their orders are not delivered within the agreed time. Although some customers may get their orders on-time, this does not a reasonable excuse to delay other orders especially if the later jobs came into the system first (see sections 3.2.3 and 3.2.6).

One could describe the main sources of the above mentioned problems, in five main points:

A. Unexpected events arriving, such as a particularly an urgent order.

B. Sometimes, there is no agreement between operations management and the marketing people who

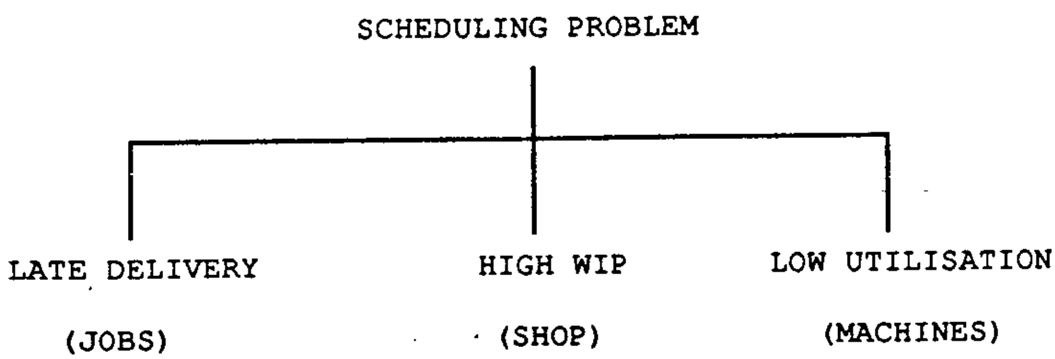


Fig. 10 Main Scheduling problem
in a job shop

are responsible for determination of delivery dates. This may be explained as they use a 'catch customer' policy instead of an 'attract customer' policy by improving the trust and confidence in the system [40].

C. The by-passing of customers at the cost of others who have been received first.

D. Jobs may be lost inside the shop because of high WIP inventory.

E. Some jobs may become late because of the decision rule followed. The decision rule may have one or more of the following characteristics:

E1. It is unrealistic or relatively difficult to be followed by the labour force[19].

E2. it is biased towards a single goal such as the shop, a machine requirement or towards some customers in particular.,

E3. It is static, and/or

E4. It may produce many similar high priorities at the same time for different jobs.

It is obvious that these three problems inter-related. One could say most of those five points are understood by the majority of companies. In practice, many companies have some of the above mentioned failings, i.e they are biased towards some customers or they may measure their customers according to how much expected profit or loss. Many firms evaluate the process purely in a blindly financial manner [15-17,67,75] (see Fig 11).

3.3.2 PROPOSED SOLUTIONS:

The main aim of this study is to find out an answer to scheduling problems using a scheduling dispatching rule as the key to the solution. Most current rules ignore either the customer's nature or the system's nature. The system's nature could mean the nature of equipment, staff and labour. For example, sometimes the top management make a decision which is difficult to be understood by those people who are responsible for executing that decision. This could lose harmony in the system which has an effect on the success of the schedule [19]. The customer's nature could mean his requirement. Some requirements of a customer are as follows:

- meeting delivery dates,
- treating all customers equally, and
- giving each customer a clear idea about how his order will be dealt with.

Sometimes, the applied rule is evaluated according to the amount of profit. It may ignore the value of long term customer relations, for example by not stressing the timely completion of important jobs. Many define important job as the job which gives the most profit. This is a misunderstanding of

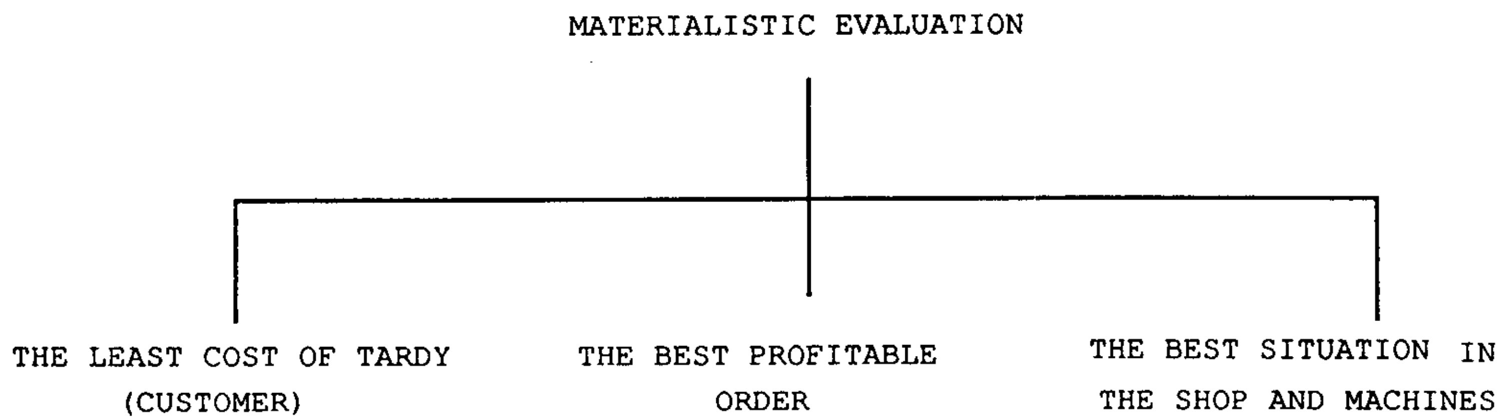


Fig.11 Materialistic view

the meaning of importance. The important job is the job which is going to be late. It has to be finished on the agreed date. If there is more than one job which is going to be late, then the important job is the one which first came to the system. The system should keep its promise, because what could someone expect from a system which does not?.

The proposed method should be relatively simple, and understandable from most levels in the system, especially from the labour force. The due date is a promise which should be kept. Therefore, marketing people and shop floor operations management should be in agreement. The current shop situation and the state of other jobs should be considered. A suitable arrival and entry mechanism may help in decreasing the WIP inventory and the lead times of jobs.

In the instance of a machine breakdown, the usual solution is to accelerate those jobs which are not going to that machine, until the machine is maintained. Jobs which would normally be going to it are either delayed or subcontracted till the machine is repaired. Decreasing the possibility of a machine breakdown can be achieved by paying more attention to preventative maintenance. This could be carried out in the idle time of a machine or by inserting an idle time for maintenance purpose.

In case all the solutions are not effective, subcontracting can be used when the shop became congested. This may also be used when an urgent order is received and there is no possibility to perform that order within the shop. Subcontracting may solve many scheduling problems. One of the most serious problems is machine breakdown. Although subcontracting may lead to extra costs, it would offer an acceptable and appreciated solution if it is used efficiently. Efficient subcontracting depends on:

- Selecting the critical order, and
- Finding the right subcontractor.

The most critical orders are either the most tardy ones or the one which has caused serious tardiness in the shop. In the latter case, an order usually has a long processing time. Some times the chosen order is the most urgent one.

Finding the right subcontractor needs considerable attention. They should be trusted to cooperate with the job shop in terms of delivery dates. A list of subcontractors should be ready. For making subcontracting more efficient, schedules of idleness for each subcontractor should be known, i.e updating the schedules is required.

3.4 SUMMARY:

In this chapter the scheduling problem is described as a number of jobs to be processed on a number of machines, following a specific order. The manner in which this is performed depends on a scheduling rule.

Scheduling problem elements are discussed. In general, these elements could be divided into physical elements (i.e job, machine and shop) and evaluation elements (i.e measure of performance and scheduling rules). Scheduling problem classification is made in accordance with these elements.

Another detailed classification is made. It classifies the problems into four parts: shop input/output, loading, sequencing, and dispatching. This classification presents In-Process scheduling procedures.

The problem concerned within this thesis is an In-Process problem which mainly concerns with the meeting of delivery dates, decreasing high WIP and improving machine utilisation. It is also concerned with balancing the conflict between customer and shop requirement based on Fairness principle.

CHAPTER 4

JOB SHOP SCHEDULING RULES AND INTRODUCTION TO THE PROPOSED RULE

CHAPTER 4

JOB SHOP SCHEDULING RULES AND INTRODUCTION TO THE PROPOSED RULE

4.1 INTRODUCTION:

As mentioned previously, the main goals of the job shop scheduling process is to ensure timely delivery, minimum WIP, and to achieve the most efficient use of the system possible. The impact of scheduling rules on schedule performance indicates that the rule employed can play a critical role in providing an effective solution [8,21,22]. An effective scheduling rule usually takes into account the primary objectives of a production system, in particular the improvement of delivery performance.

There are many different scheduling rules in existence which may provide different way of solution. Each solution depends, to some extent, on the problems complexity. Examples of these rules are called dispatching priority rules and heuristic rules

The literature uses the terms scheduling, dispatching and sequencing frequently. These terms may often have a similar meaning, especially when they are used separately. Scheduling is a broad term which is concerned with the timing of events in the shop, for example assigning operations to the right place at the right time. Dispatching is concerned with selecting the job which should to be processed first. Sequencing is concerned with arranging queues such that the job which has the highest priority is first in the queue.

Scheduling decisions, in a job shop production system, can be affected by the rules employed. A scheduling rule defines the manner in which a job is selected to be processed next on a machine. In most situations assignment strategy is based on setting priorities or setting procedures which select the job. Scheduling a number of jobs is a difficult task, because of the huge number of possible schedules that could be made. For example, in the case of n jobs queued at a machine, there are n! possible ways to sequence these jobs. The aim of obtaining a scheduling rule is to decrease this uncertainty. Heuristic rules have the same aim and are usually more effective than traditional scheduling techniques [2,37,53].

This study is concerned with developing a heuristic rule to schedule a number of jobs through a number of machines. This heuristic rule uses priority rules and state dependent procedures. In general, the rule strategy involves three main topics:

- The fairness principle,
- Delivery performance, and
- Current shop and machine state

In this thesis, the developed rule is called the Fair Delivery and Shop State Dependent heuristic rule, hereafter called FDSSD. This chapter discusses the environment of scheduling rules, scheduling operations, the classification of scheduling rules, and the framework of FDSSD rule.

4.2 ENVIRONMENT OF SCHEDULING PROCEDURES:

Scheduling procedures should be taken in accordance with scheduling problems because they both share the same physical elements; job, machine and shop, together with evaluation elements. Therefore, there are few repetitions. Section 3.2 in chapter 3 presents this in more detail.

4.2.1 ELEMENTS OF SCHEDULING PROCEDURES:

As mentioned previously, section 3.2.2 discussed the elements of the scheduling problem. These two elements are going to be discussed here from the point of view of their effect on scheduling rules and procedures.

A successful scheduling procedure would mainly depend on physical elements - job specifications and condition, machine state, shop situation - and evaluation elements. The greater number of these sub-elements considered, the greater chance there is of obtaining an optimum scheduling procedure.

Job specifications are taken into account by many of the traditional rules. For example, the SPT rule uses the processing times of jobs to indicate priority, and the EDD rule uses the delivery date. Many rules, such as the SPT rule, neglect the other specifications, such as due dates. Furthermore, the SPT and the EDD rules neglect some of the other physical elements found in scheduling problem such as receiving time and machine state.

This story is repeated again with rules which are concerned with machine utilisation, such as the WINQ rule. The WINQ rule is concerned with one of the main goals of production systems, but it again neglects the condition of available jobs. It also ignores the situation in the shop as a whole, WIP, queues, total waiting times, and so forth.

Again, many rules which are concerned with shop situation, ignore either job specifications, machine utilisation or both. The SPT rule could help in eliminating congestion in a shop due to high WIP. Although the SPT rule is an example of a rule that is concerned with decreasing the WIP inventory in a shop, it may worsen queuing problems. It prioritised jobs with short processing time to be finished quickly, leaving the shop with high processing time jobs waiting in queues. Utilisation of some machines in the shop may go down because of these lengthy jobs. The SPT rule in this case, again ignores machine condition.

Scheduling elements should be related to each other and a clear target should be defined. The elements to be evaluated are the measure of performance and the scheduling rules (see section 3.2.3). Determining the measure of performance is a vital problem, especially if it becomes biased towards one element at the expense of another one. Thus, in choosing a measure of performance, attention should be paid, firstly, to the deliver of jobs according to their due date, then to machines condition and the shop situation.

4.2.2 EVALUATION ELEMENTS:

The element evaluated is the reference by which the performance of the schedule can be measured. Therefore, it should be the main criteria in choosing a rule. The main basis of selection for scheduling rule should be according to the following measures:-

- To satisfy customer requirements by meeting the job's delivery dates.
- To meet machine requirements by minimising machine idleness.
- To meet shop requirements by minimising WIP and decreasing flow times.

Because of the conflict between these criteria, and because of the importance of each one, especially the first one, a rule should be developed which considers these all criteria together. It is obvious that the final procedures are not the optimum, from the point of view of each separate job, but they would be the optimum according to general situation.

Scheduling rules may have a direct influence on the situation in the shop. For example, congestion, WIP, or mean flow time could be minimised effectively by the SPT rule. This is because of the minimum flow time which can be achieved is obtained by applying the SPT rule. However the SPT rule may leave some jobs behind schedule, waiting in the shop for a long time, which could be avoided by applying the FRFS rule efficiently. Although a combination of the SPT and the FCFS rules has been developed in previous works [37], it does not consider neither jobs' due date nor machines' situation.

4.2.3 JOB SCHEDULING ON MACHINES:

A. Scheduling n jobs on one machine:

In this case, scheduling will differ from other cases because idleness is not expected. There will be many jobs waiting in the queue for processing. The measure of performance which commonly used in this instance is the flow time. Although the SPT rule is one of the best rules for achieving the minimum flow time in the shop, the SPT rule may result in high values of tardiness and earliness of some jobs. The SLACK rule can perform very well in decreasing tardiness and earliness values (see example 4.1)

EXAMPLE 4.1

Four rules are chosen to be tested here: First Received First Served (FRFS), Shortest Processing Time (SPT), Earliest Due Date (EDD), and the SLACK rule (see Table 4-1 and 4-2)(see Fig 12).

It is clear that the SPT rule has the minimum flow time and mean flow time when compared with the other rules. Because the SPT rule does not take account of due dates, lateness is relatively high. The EDD and the SLACK rules perform better than the SPT rule in achieving minimum mean tardiness and earliness, because both base their decision on due dates. This example demonstrates a static problem, but the need for an effective rule covering the case of dynamic and continues job arrival is required.

B. Scheduling n jobs on m machines:

The single machine scheduling problem has been discussed previously. It is the simplest form of the scheduling problem. Complexity is increased when more machines are involved. In general, a job shop scheduling problem centre around selecting the optimum scheduling procedure from large number of possible schedules which can be generated even for a small job shop. Because of the huge number of generated schedules (e.g if N is the number of jobs and m is the number of machines then the number of possible schedules equals $(N!)^m$), simulation is used to

JOB	Received order	Due Date	Processing Time	Slack Time	Flow rate				LATE (-)EARLY (+)			
					FRFS	SPT	EDD	SLACK	FRFS	SPT	EDD	SLACK
A	1	17	2	15	2	3	9	24	15	14	8	-7
B	2	20	15	5	17	40	25	22	3	-20	-5	-2
C	3	9	7	2	24	15	7	7	-15	-6	2	2
D	4	30	10	20	34	25	35	35	-4	-5	-5	-4
E	5	18	1	17	35	1	10	25	-17	17	8	-7
F	6	36	5	31	40	8	40	40	-4	32	-4	-4
Total Flow Rate					152	92	126	153				
Mean Flow Rate					25.3	15.3	21	25.3				
Number of Late Jobs					4	3	3	5				
Total tardiness									-40	-31	-14	-24
Total earliness									18	63	18	2
Conditional Mean Tardiness									-10	-10.3	-4.6	-4.8
Conditional Mean Earliness									9	12	6	2

Table 4-1 Example 4.1

Rule	Sequence of jobs
SLACK	C-B-A-E-D-F
FRFS	A-B-C-D-E-F
SPT	E-A-F-C-D-B
EDD	C-A-E-B-D-F

Table 4-2 Sequence of jobs -Example 4.1

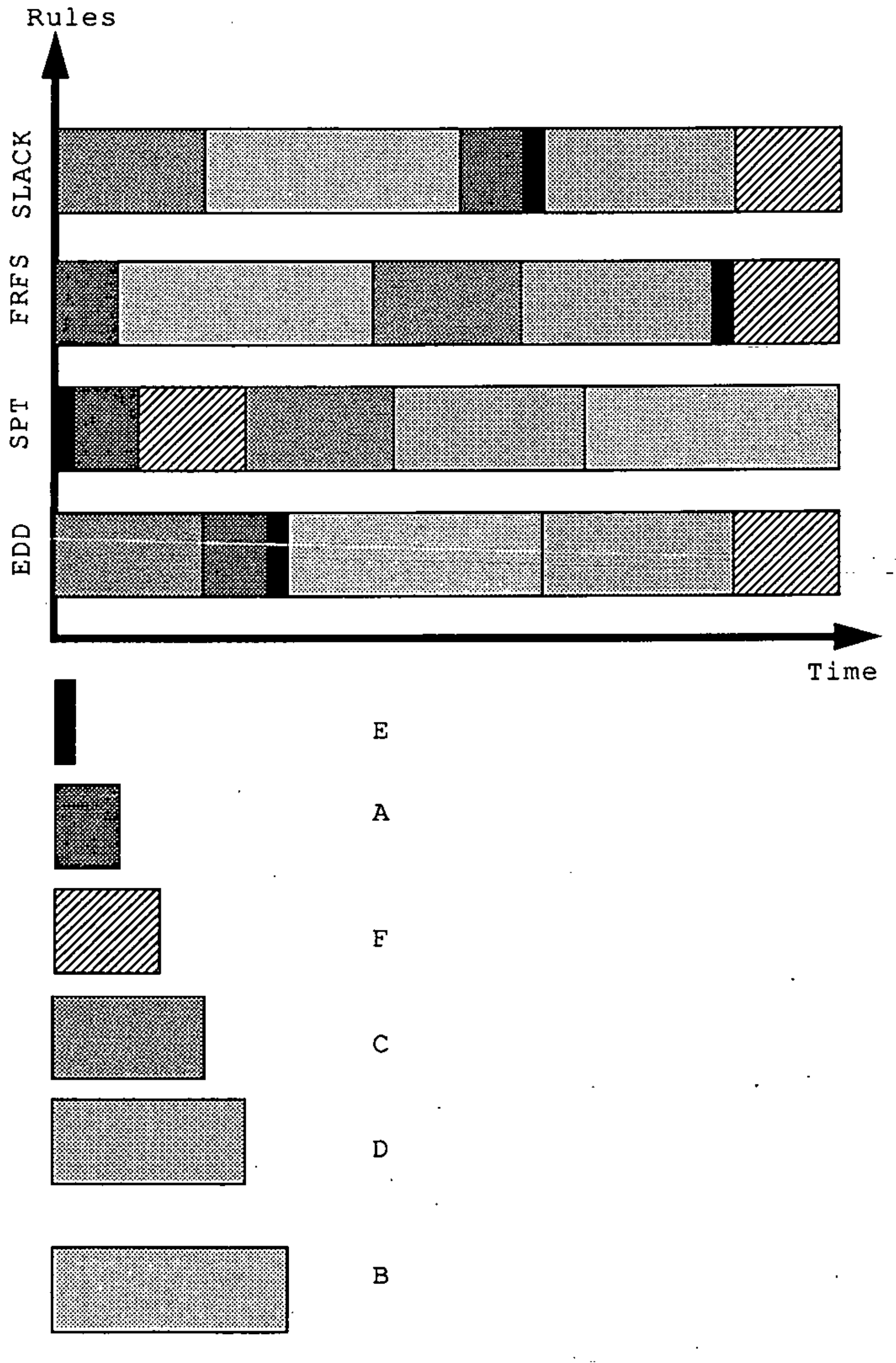


Fig 12 Example 4.1

study and investigate scheduling rules in different situations in order to find the optimal or near optimal scheduling procedures [1,54,55].

In the case of multiple machines in the shop, more problems may be raised, such as, WIP, In-Process queues or machines idleness. Therefore, the rule should be able to anticipate the future; 'where and when operations will take place'. In such a complex situation, a dynamic rule which considers due dates of jobs and shop requirements is required.

4.3 CLASSIFICATION OF SCHEDULING RULES:

In general, scheduling rules may be categorised as follows (see Fig 13):

1- Load information: Local or global.
2- Time effect: Static or dynamic.
3- Shop load situation.
4- Complexity.
5- Scheduling elements requirement and goals.

4.3.1 LOAD INFORMATION BASED FORM:

Scheduling rules in this category may be classified according to the quantity of information taken into account. The two types are:

- Local scheduling rules.
- Global scheduling rules.

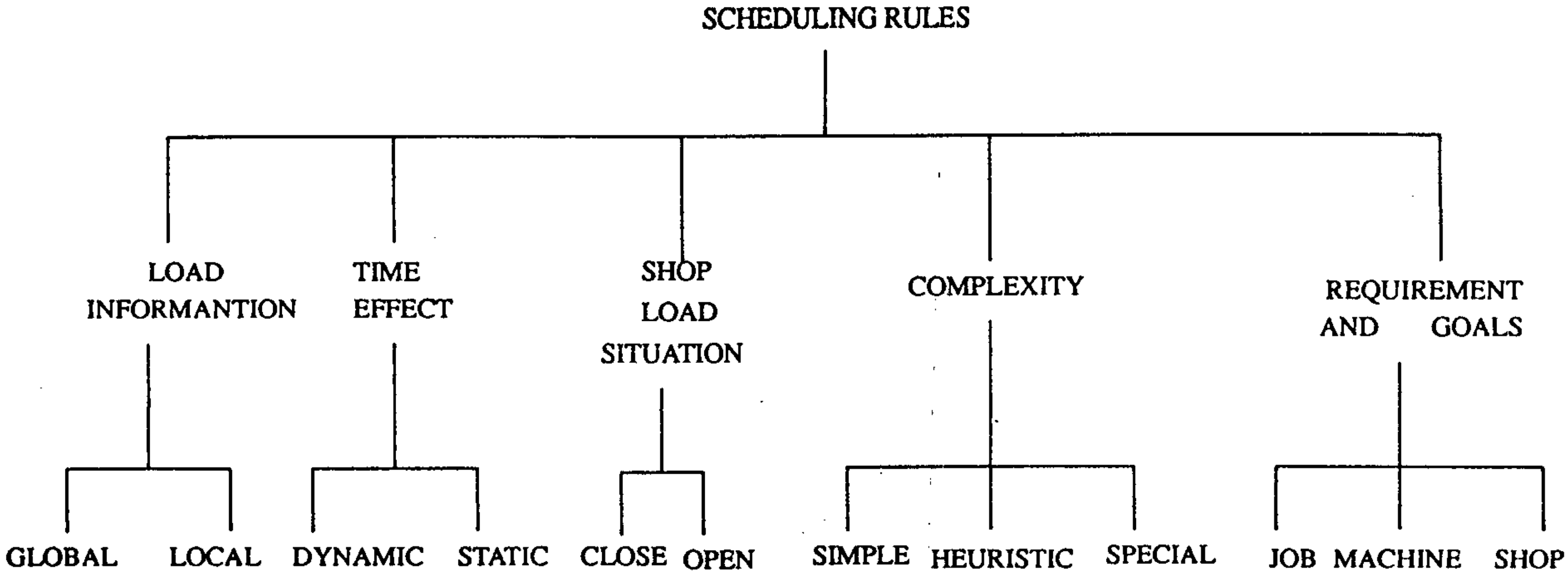


Fig 13 Scheduling Rules Classifications

Local rules use local load information which is concerned with the current machine and the current queue at that machine [32,52]. For example, the SPT rule is a local scheduling rule, because it is concerned with processing times of waiting jobs in a queue before the current machine.

Global rules, in addition to the local information, use load information on other machines and other queues in a shop [32,52]. WINQ rule is an example of a global scheduling rule which considers the information on the other queues.

4.3.2 TIME EFFECT BASED FORM:

There are two types of scheduling rules which are concerned with the role of information relating to the time. These two types are:

- Static scheduling rule.
- Dynamic scheduling rules.

Static scheduling rules are used to indicate the priorities for static situations. Since static priorities are not related to the passage of time, the priority of each job keeps the same value all the time. The priorities are usually based on job specifications such as due dates or processing times.

Dynamic scheduling rules perform in a more realistic way than static rules. Decisions can be changed with the passage of time. These scheduling rules usually depend on shop information in addition to job specifications. Sometimes priorities are a function of passing time, such as with the SLACK rule. With this rule, as more time passes, the lower value of remaining slack time is expected.

4.3.3 SHOP LOAD SITUATION BASED FORM:

From the last two forms, the job shop load situation could be derived. Scheduling rules could be classified according to their response to update information.The two classes are:

- Closed scheduling rules.
- Open scheduling rules.

Closed rules use fixed information and do not react to changes in the ongoing situation or to any new events in the shop.

On the other hand, open scheduling rules may react to new events in a shop and they may draw attention to anticipated events. A trigger mechanism can be used in this case. This type of rule illustrates the principle of using feedback information in making a decision. As well as this, many rules fall between these two classes.

4.3.4 COMPLEXITY BASED FORM:

Within this grouping, scheduling rules can be divided into three classes depending on the complexity procedure [2,23,32]. These classes are:

- Simple priority rules,
- Heuristics, and
- Other rules.

Simple priority rules usually use the information related to the job specification. A combination of simple priority rules can be expected in many instances. Simple priority rules and their combined rules may also be associated with a weighting factor allocated to each job.

Heuristic scheduling rules involve more complex considerations than simple priority rules; the anticipation of machine loading, for instance. Heuristic rules may involve human logic in a non-mathematical way [2,53]. The heuristic method is a rule of thumb. It is the reasonable method by which a solution to complicated problems can be obtained according to the previous experience of the management. However, although heuristic scheduling rules can help to cope with a problem, they can not guarantee optimal solutions.

The last category is the special purpose scheduling rules, which are designed for a specific shop.

4.3.5 REQUIREMENT AND GOAL BASED FORM:

As mentioned in section 4.2.1, elements of scheduling procedures - physical and evaluating elements - may be considered in conjunction with the main goals of a system. The main goals of a system are to meet delivery dates, to minimise WIP in shop, and to increase the utilisation of machines. This classification depends on the requirement of the physical elements of the scheduling. Therefore, scheduling rules could be divided into three classes:

A- Job requirement scheduling rules,
B- Machine requirement scheduling rules, and
C- Shop requirement scheduling rules.

A- Job requirement scheduling rules:

The class of job requirement scheduling rules is concerned with customer requirements. The EDD rule is an example concerned with the due dates of orders. Also, FRFS and FCFS are other examples of customer requirements rules where the criteria are receiving and arrival time, respectively. Both rules are based on the "Fairness" principle. This class of rules can be called "customer requirement scheduling rules", because these rules concentrate on customer satisfaction.

B- Machine requirement scheduling rules:

Machine requirement scheduling rules are concerned with keeping machines busy. In other words, these rules aim to decrease idle time at machines. For example, WINQ rule is concerned with queues at other machines.

C- Shop requirement scheduling rules:

Shop requirement scheduling rules also may include machine requirements. These rules are concerned mainly with the situation in the whole shop. Following up queues in the shop, machine utilisation and WIP are some of the main concerns of these types of rules. For example, the SPT rule is one of the rules which can be used to minimise WIP and decrease flow time in the shop.

4.4 LIMITATION OF EXISTING SCHEDULING RULES:

Traditional rules have had only limited success [37]. State dependent rules are introduced by Conway [38,52]. Many of these rules ignore the received order of jobs, jobs' due dates, available capacity or/and state of the shop. There are some rules which consider the received order such as the FCFS, FIFO and FRFS rules. They actually support the "Fairness" principle, but most of these rules ignore the delivery dates of jobs, which may lead to both early and late finished jobs at the same time. Also, these rules usually ignore the situation in the shop.

The SPT rule performs very well in keeping mean flow time as minimal as possible, but it leaves the jobs with a long processing time waiting behind in the queue. Because the SPT rule does not perform according to due dates, it has been combined with the EDD rule [36]. Even though such a combination is made, many jobs still become late while some others expedited. Rules based on due dates are performing relatively well [40], but again they do not consider the received order and they do not account for the shop situation.

Many traditional rules delay some jobs in order to prioritise others. The former job may have been received before the latter job, however the latter job finishes on time and the former job becomes late. Unfortunately, many firms may rely on these policies. The reason for employing by-passing policy is usually the importance of the latter job; the latter job is more expensive or more profit could be achieved. Suitable rules may have a sizable effect on that situation.

The principles of meeting delivery dates and maintaining good relationships, loyalty, trust and respect, along with concern for the customer, are supported by many of successful companies, because of the long term effect of such principles on the business itself. Many of the well-known Japanese firms are examples of this case [15,16].

The harmony in the system between all levels, labour force and the top managerial who are the decision makers, is required in order to achieve better understanding for the scheduling decision to be followed. Therefore, the more simple the scheduling rule, the more understanding within the system is expected. This is especially true if the rule is near to human principles [15,17,19,60].

One could conclude that many scheduling rules ignore the importance of meeting the delivery dates, either because of expediting most profitable jobs and delaying some others or because of expediting urgent jobs which are received late. Some other scheduling rules ignore the situation within the system; the machine state or shop situation. In this thesis, efforts are made to consider received order and delivery dates in making scheduling decisions by introducing the FDSSD rule; the proposed scheduling rule herein.

4.5 OPERATIONS SCHEDULING RULES:

Scheduling decision rules allocate available capacity to available jobs and activities through time. Scheduling is done on a time scale of few months, weeks, days, or hours. In other words, scheduling is done on a short-term basis.

Scheduling is concerned with the conflicting objectives of a production system: meeting delivery dates, increasing machine utilisation, and decreasing WIP. Increasing machine utilisation in the shop may require an increase in the amount of WIP. Increasing the amount of WIP may lead to congestion in the shop. Congestion usually leads to an increase in the lead times of jobs. This will cause a delay of some jobs, especially those with tight due dates. Therefore scheduling aims to find a balance between these conflicting objectives in a production system. In a job shop, WIP consists of a number of queues at machines. Therefore, shortening queues is one of the main targets to be achieved through scheduling.

In chapter 3 detailed scheduling problems are discussed. They are shop input/output, loading, sequencing, and dispatching. In this section, sequencing and dispatching rules are dealt with. Generally speaking, a dispatching rule is concerned with selecting a job from a number of jobs waiting to be processed. There are hundreds of rules for scheduling [23]. The next section will discuss some of these rules.

Those complex situations may be resolved through the use of an effective and dynamic scheduling rule. This rule may use loading in conjunction with dispatching and sequencing procedures. Loading will make reservations on the available capacity at the machines. Secondly, input/output procedures will balance jobs released to the shop, (referred to as 'arrival' henceforth) in order that the amount of WIP in the shop should remain within an acceptable limit. Finally, after sequencing arranges the queues, dispatching rules will select the right job at the right time, according to the previous reservation. In general, the above described procedures are the core of the proposed scheduling procedure used by the FDSSD rule. More detail is presented in chapter 5.

A number of rules which are generally known to researchers are listed below. More rules are listed in Ref. [8,23,45,46]:

- FCFS: (First Come First Served) This rule selects the job which arrived first on a queue or machine.

- FRFS: (First Received First Served) This rule is based on the received order of jobs. It selects the job which is received by the system first. In other words, it selects the job which has minimum receiving time first.

- EDD: (Earliest Due Date first) This rule selects the job with the most urgent due date to be processed first.

- OPNDD: (OPeratioNal Due Date) This rule selects the job from a queue according to current operation due date.

- SLACK: (minimum SLACK time, first) This rule selects the job which has minimum slack first.

  Slack = due date - processing time - lead time.

- SPT: (Shortest Processing Time, first) This rule selects the job with minimum processing time.

- WINQ: (Work in next queue) This rule selects the job whose next operation is at the machine with the minimum queue length. It called herein QINM.

- TSPT: (Truncated SPT) This rule performs similarly to SPT rule but it specifies a maximum time for a job to wait in a queue.

4.6 FRAMEWORK OF THE PROPOSED FDSSD RULE:

4.6.1 FAIR DELIVERY AND SHOP STATE DEPENDENT SCHEDULING RULE (FDSSD):

In general, the proposed heuristic rule (FDSSD) is a combination of different rules and strategies. These function in harmony to achieve the principle objective of meeting the delivery dates, i.e the job which has been received first in the system, has a higher priority with regard to the due date promised than jobs received afterwards.

The FDSSD rule is divided into four stages to make it more understandable and to form a cushion to absorb the changing situation in the shop (see Fig 14). These stages are:

- Receiving stage,
- Arrival stage,
- Entry stage, and
- In-Process stage.

Secondly, each stage has a certain way of dealing with jobs. For example, at the receiving stage, a reservation is placed to give each order the right time on a time scale of "when and where". At the arrival stage, jobs are selected according to their reserved time. If no job has arrived at the shop by that time, then one of the other jobs received later will be selected to go forward in the system in order to keep the machines busy.

Entry and In-Process stages have more or less similar strategies. Both are concerned with the next job to be entered to the shop or loaded onto the machine respectively. The main strategy is to arrange the queues according to the FRFS rule, then check in each queue to see whether if there is a job whose entry or machining time is due and it is not going to delay another job in the queue which has been received previously.

Finally, in all stages, the FDSSD's strategy is concerned with the state of the shop. Next chapter will discuss the FDSSD rule in more detail.

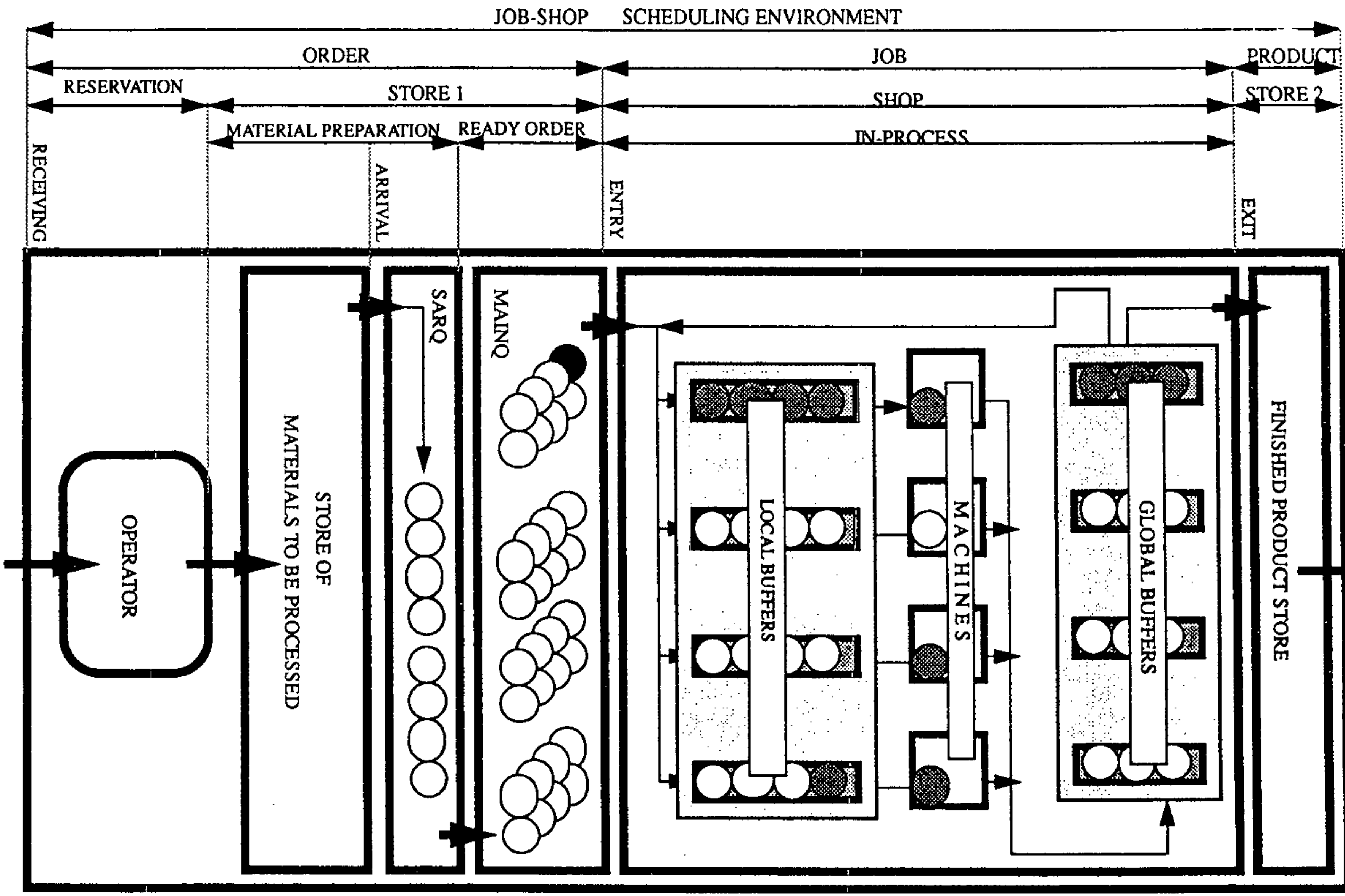


Fig. 14 Job Shop Scheduling System And Scheduling Stages

4.6.2 COMPUTER SIMULATION INVESTIGATION:

A program has been written in Fortran77 language on a Sun 3/50 work-station. The operating system is Unix. The program aims to investigate different rules, then to develop and investigate the proposed rule using identical data. The result has been studied and analysed. The model used is called JSSM. The model consists of many parts, (see Fig 15):

- Input data part: to read the data.
- Output result part: to analyse results.
- Rules selection part.
- Main block to control the model.

The main block is designed to communicate frequently with the rules' block. The program is designed to perform in two different ways. First at all, priority status is designated to each job, then the processes will take place in machines according to this status. (This is also be applied when traditional rules are used such as the EDD rule). Secondly, the priority could be set but may not be followed. This may happen as when heuristic rules are involved. Therefore, the performance will be according to the situation or according to the measure of performance. In chapter 6, the simulation model JSSM is presented in more detail.

This model is able to simulate a dynamic job shop problem using up to 49 machines. At the present

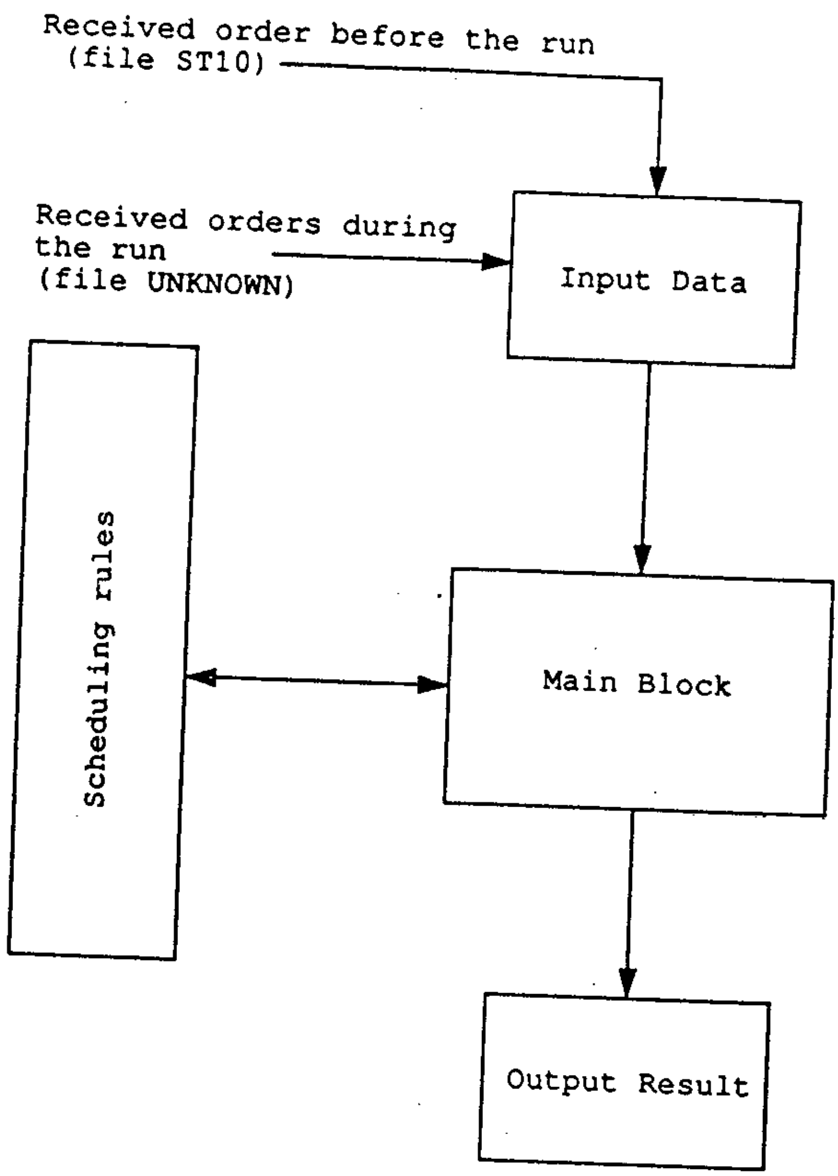


Fig 15 Input/Output and General structure of JSSM.

time the model is restricted to 9 machines. However, it could be changed in a few steps with a relatively simple procedure. The receiving rate can be controlled and changed according to the required data.

Inserting a rule is a matter of adding a subroutine to the model. The main connection between the subroutine and the model is the procedure of setting the priority index which the model will follow in making a decision. In case of some heuristic rules, such as the FDSSD rule, more steps are required. In-Process information is saved and can be recalled at any time. Changing any value in the model is relatively simple.

4.7 SUMMARY:

Scheduling decisions, in a job shop scheduling system can be simplified by means of scheduling rules. Scheduling rules may take into account following physical scheduling elements:

- Job specification,
- Machine state, and
- Shop situation.

Scheduling rules may be classified into five different forms according to:

- Load information: Local or global.
- Time effect: Static or dynamic.
- Shop load situation: Open and closed.

- Complexity: Simple, combination, heuristic and special purpose.
- Requirement and goals: Job, machine and shop requirement rules.

This study is concerned with developing a heuristic rule to schedule a number of jobs through a number of machines. The heuristic rule uses both, priority rules and state dependent rules in an ethically based manner in order to achieve better delivery performance. In other words, the FDSSD rule uses a combination of job requirement rules and shop requirement rules.

In this thesis, the effort is made to consider together the criteria of received order and delivery dates in a manner which tries to eliminate conflict between the two. The situation within the shop is also given due consideration.

CHAPTER 5

FAIR DELIVERY AND SHOP STATE DEPENDENT: THE PROPOSED SCHEDULING RULE

CHAPTER 5

FAIR DELIVERY AND SHOP STATE DEPENDENT: THE PROPOSED SCHEDULING RULE

5.1 INTRODUCTION:

In many job shop production systems, scheduling rules are concerned with the amount of profit that could be achieved. Due to monetary measure of performance (profit), many of the earlier received customers' orders may become late. Furthermore as mentioned in section 3.1, long waiting time and high WIP are common problems in many job shops [3,4,5,6]. Priority congestion (many jobs needing to go first), late jobs, machine idleness and the passing jobs are some typical examples of problems in a job shop production system.

The FDSSD rule is concerned with the general aims of a production system - meeting delivery dates, decreasing WIP and increasing machine utilisation. However, it considers the orders' receiving times as the basis of decision for late jobs (Fairness Principle), while delivery dates and the state of the shop are the basis in the normal situation. Some heuristic procedures are employed by the FDSSD rule (e.g. a release mechanism to control the inter-arrival of jobs into shop, loading, sequencing and dispatching procedures).

Drawing from the above, attention has been given to find a scheduling rule to be applied in the field of complex operations scheduling, especially in the West, where a lot of companies are looking for a suitable solution [15]. The rule is presented and developed in a manner which is practical and at the same time ethical. The computer simulation is used to investigate and compare several rules with the proposed rule.

The following sections will discuss in detail the developed rule and its structure. The operation in a hypothetical job shop using the rule is discussed.

5.2 FDSSD BACKGROUND AND JOB SHOP STRUCTURE:

5.2.1 GENERAL BACKGROUND AND OBJECTIVES:

Generally speaking, the rule highlights the importance of the time when a job is received into the system. Due dates and receiving times are used in the loading of machines. This procedure aims to improve delivery performance, decrease WIP and to increase machine utilisation. A balance between these aims is sought. The performance criteria contain some ethical principles. The Fairness principle (first come first served) is one of the basic principles used to avoid servicing one order at the expense of another. A list of ethical principles is provided in section 5.5.2.

Injecting some morality into the scheduling rules is a process that needsan extensive theoretical work. Many successful firms recognise the importance of moral consideration, especially the Japanese [15-20,75,76]. This work explores the ethical dimension in scheduling procedure which could be considered as the core of the system.

5.2.2 GENERAL STRUCTURE OF JOB SHOP:

5.2.2.1 DEFINITIONS:

There are many terms that need to be defined to facilitate and describe the job shop levels. Some of these are:

Order: This is the customer's request that a product should be manufactured. It is received by ordering department and forwarded into the preparation procedure.

Job: It is an order which has physically entered into the shop.

Product: This is the finished job.

Receiving: This term may be used to refer to the order being received. In general, it refers to the event when marketing peoples receive the confirmation from a customer that a certain product to be made.

Receiving order: This is the sequence in which orders have been received.

Reservation: As soon as an order is received, then the operator reserves the necessary time needed for this order, taking into account the Fair delivery principle.

Arrival: The event in which orders are given to the store to prepare the right materials on time for entry into the shop.

Entry: At this stage the orders' materials are entered physically to the shop.

Machining: This is the actual processing of an order on a machine or machinery.

Exit: The event where a job has no more process. It is defined also as the exit from the shop after completion.

Delivery date: The date when a job should be submitted. It is the final promise which has been given to the customer after reservation.

Due date: It is the delivery date that customer initially was promised. This date is the main target to be achieved, but if the shop is too busy, a new date may be set.

Local Buffer: It is located before each machine. The maximum capacity is 4 jobs.

Global Buffer: It is located inside the shop to store jobs between operations if the local buffer is full.

5.2.2.2 JOB SHOP STRUCTURE:

It is assumed that the job shop has the following structure as shown in Fig 14 and Fig 16:

(1) THE OPERATOR IN ORDERING DEPARTMENT:

They receive orders from customers and make subsequent reservations on a time scale to allow for their processing. After checking the new orders with the operation manager in the shop floor, they confirm delivery dates to customers. Their duty could be:

a- To check the validity of due dates. The estimation of feasible delivery dates requires detailed information on the current loading and other commitments in the shop.

b- To reach a compromise with the customer in case requested due dates are not possible. It is necessary to make the situation clear to the customer. No change in the reservation of previous orders should be made if permission has not been taken from customer. If the customer really needs his order at the time when the system is not up to it, then there are a couple of things that could be done. These courses of action that may offer a solution to the problem are:

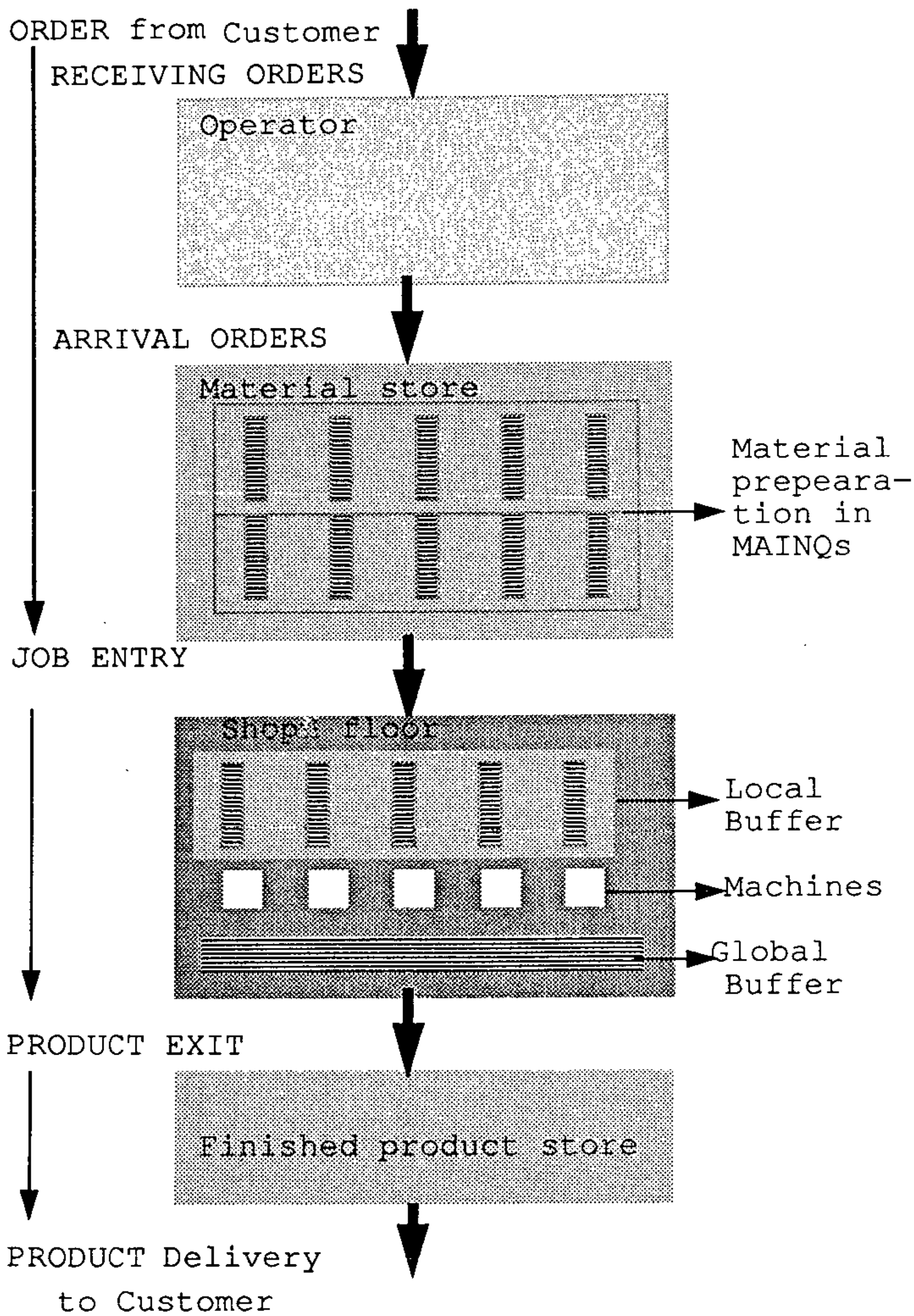


Fig 16 Job Shop Structure

b1- To check with an earlier customer and to delay his order so that the later customer may have his order suitably early.

b2- If this negotiation fails, then they may arrange for a subcontractor to do the work (see section 5.3).

d- To issues orders to the material store sufficiently early to allow time for material preparation.

e- To update the currently available information in the shop, e.g loading, and the new delivery dates.

(2) MATERIAL STORE:

This is the place where the materials for arrived orders should be prepared then either pushed into the shop at the right time (in normal situation) or pulled (in low WIP situation) by the operations manager into the shop floor.

(3) SHOP FLOOR:

After their entry into the shop floor the orders are henceforth called jobs. The shop floor consists of a number of machines, input buffers before each machine (local buffer) and the output buffer, which is called the In-Process machinery buffer (or global buffer). The capacity of the local buffer is four jobs for each machine. The global buffer represents the intermediate place between two operations. As soon as a job finishes on one machine, then it will normally go to next operation on the next machine. If there is not an empty space the local buffer then this job will be placed in global buffer. The volume of WIP in the shop can thus be controlled.

(4) PRODUCT STORE:

This is the place where finished jobs are stored as products to be delivered.

5.3 THE PROBLEM:

As mentioned in chapter 3, the problem in general is that a number of jobs are to be processed on a number of machines in such order that delivery dates are met, WIP is kept as minimum as possible, and machinery is efficiently utilised. Long waiting time in the shop and late jobs are well known problems. Late orders could be decreased by utilising efficient reservation and loading procedures. Although some improvement can be obtained by using the right loading procedure, usually much of the problem often still remains. This can be because of the arrival of an urgent customer or a particularly profitable job. This job may jump through the shop ahead of existing queues without any consideration to other jobs or even to the schedule that has already been made. Owing to the short-sighted considerations, i.e usually purely financial, many of early received orders are left behind schedule.

Because of the inherent conflict between the aims of the production system, some measure of performance should be considered. There are many measures that could be taken such as having minimum WIP inventory or aiming for minimum tardiness. There are many other principles to be considered that have been discussed in section 5.5.2.

The problem can be illustrated by two extreme cases. In the first example, the scheduler gives top priority to job requirement such as meeting some jobs' delivery dates thus leaving behind some other jobs that may have been received before those given priority. On the other extreme, the scheduler may emphasise a shop requirement such as minimising WIP, leaving some jobs waiting for long time because the scheduler is purely concerned with the cost of production or machine utilisation.

The aim of the proposed rule is to balance these conflicts between production system objectives in a more realistic manner. Delivery dates and the state of the shop are taken as the main basis of the rule when there are not any jobs behind schedule. If there are, then the received sequencing order may be used to determine priority until the delays have been cleared.

5.4 FRAMEWORK OF THE FDSSD RULE:

5.4.1 THE FDSSD RULE PROCEDURES:

The FDSSD rule depends basically on delivery dates requirement, receiving order and shop state. The manner of considering the criteria will be according to the state of each jobs. The more early jobs in the schedule the more consideration is given to the shop state (see section 5.4.4). Ethical consideration is given within the FDSSD rule, e.g. Fairness principle. Most of described heuristic rules in existing literature are based on dispatching, sequencing or loading procedures. The FDSSD rule employs these three procedures but in a manner that attempts to be fair to the customer. For more details on the heuristics within FDSSD rule, see section 5.4.4.

Using the Fairness principle in a critical or tie situation may offer a reasonable and acceptable decision. However, it is not sufficient for making optimum decisions. Delivery dates, also, on their own may not produce optimum decisions. Furthermore, adjusting the priorities according to the situation in the shop may not be enough to form the optimal schedule. Therefore, it is recommended that Fairness principle, delivery dates and shop state dependent criteria should be combined to form a more realistic scheduling procedures.

In brief, the FDSSD rule depends on the due date in making a time-based reservation for each job at each machine. In the case of due dates not being met, then new due dates are set which are then called in this case delivery dates. The reservation procedures for a machine depend on:

- Due date,
- When the machine is available.

If the machine has enough free time available to fit in the newly received job before its due date, then the due date equals the delivery date. Several heuristic techniques are employed in this instance (see section 5.4.3). Shop and machine state dependent procedures are considered inside the shop at the local buffers and at the material store. At the local buffers and the material store's queues, the jobs are sequenced according to the Fairness principle - the FRFS rule is used. Then the dispatching procedure selects the job which is going to be late in order that first job in the queue is not going to be late. If there is no job becoming late, then the dispatching procedure will consider the situation on other machine (e.g. queue length).

The ethical consideration given involves the Fairness principle during the reservation and at dispatching procedures. The due date is also given due consideration, i.e no promises should be made if the shop is not able to live up to its commitment.

Otherwise, compromise and negotiation with customers to fix a new delivery dates is required.

The simplicity of a scheduling rule and the ethical considerations which may motivate the production team (labour force) to keep up with the schedule if an urgent or a new unexpected event in the shop should take place [61,62]. Harmony among all people in the system (management and labour force) is required [19,62,80]. The FDSSD rule attempts to preserve that harmony by being close to commonly understood principles.

The customer may be considered as an indirect element within scheduling procedures because a job represents the customers order in the system. Therefore, the scheduling procedure should be performed in a manner that increases the confidence of the customer in the system; if the customer does not trust the system, he will probably not come back [40,74].

To conclude the above, the FDSSD rule combines the Fairness principle, delivery performance and state procedures in an ethical way to improve the morale of the job shop. The consideration shown to the customer and the simplicity of the FDSSD procedures may achieve a harmony among the people in the system. As the result, the customers confidence would be increased.

5.4.2 BASIC ETHICS PRINCIPLES:

There are some simple questions that could be asked by a customer that one may find it difficult to answer. For example, what answer should be given if the customer demands to be kept informed why his order is late. Should the manager tell him that the shop has congestion, high WIP or that waiting times are very high in the system? Not a suitable answer. Does the manager tell the customer frankly that there is another customer who came after him and because the other one is more important, he gets higher priority. Again, this is not to be expected. Some ethical consideration should be borne in mind in order to make the procedures more efficient.

There are a number of principles which may support the main goals of most production systems. Some of these principles are:

- **Fairness principle**: It uses the first come first serve basis.

- **Fair delivery principle**: It employs Fairness principle and due dates. The criterion is that no later received jobs are delivered at the expense of those received. Similarly, this principle could also be applied in dispatching or loading procedures.

- **Respect principle**: It means at all stages that dealing with an order means the dealing with the customer. Therefore, the receiving time of an order should be respected. Briefly, customer is respected by respecting his delivery date and by paying due attention when his order was received [20].

- **Promise principle**: The delivery date is a promise. No promise should be made if the shop can not live up to it [15,20].

- **State principle**: This determines that scheduling procedures according to the shop and machine requirement should be followed as long as this does not go against the customer's interests.

- **Harmony Principle**: This specify that the rule should be reasonably understandable to every one within the system in order to achieve a degree of harmony between the top management and the labour force [19,62].

- **Clarity principle**: Being clear with customers is important. The right of a customer to know about his order has to be given [15-17,20].

- **Low-for-long principle**: This states that having a number of long-term customers who give a relatively low profit on each order is preferable than to having a number of temporary one-off customers who just want a few highly profitable jobs done quickly [20,75].

- **Computer-Human principle**: This means feeding the computer programme with the human sense decision rules that would achieve the above Harmony principle. Using a computer may save time in one sense, but it may make the situation more complex in another sense if it is not programmed to employ the common-sense basis for human-style decision making [60].

5.4.3 THE FDSSD RULE STRUCTURE:

The FDSSD rule has three main strategies. Each strategy depends on the situation of the jobs and the shop. These strategies are:

1- To consider receiving times according to the FRFS rule.
2- To consider due dates based rules (e.g. the EDD rule).
3- To consider shop state according to WIP and machine utilisation.

As mentioned previously in section 4.6.1 and 5.4.1, the main structure of the FDSSD rule follows the structure of the job shop. Therefore, four stages are proposed to simplify scheduling procedures:

A- Receiving stage - could also be called the reservation stage.

B- Arrival stage.

C- Entry stage.

D- In-process stage - Shop-floor stage.

A- Receiving stage:

Reservation is one of the first stages where orders are scheduled across time according to due dates. Backward loading begins with the due date for each job, then proceeds by subtracting the processing time backward in time against each machine. If the available time is already scarce then a space is found in which to insert the job by means of backward checking. If no place becomes available this way then a forward loading procedure is used. Before backward and forward loading procedures take place, test is performed to try and create a sufficiently large gap close to the due date in which to fit that job. After the job has been settled on time scale then the new due date will be henceforth called the delivery date.

The customer should be told about any predicted delay and the new delivery date ought to be negotiated. If it is known that the job will definitely be late, then a subcontract could be arranged as a solution. It could be possibly just be left like that if the customer accepts the delay, especially if penalty costs were deducted in his benefit.

Arrival, entry and machining times could be set at this stage. They are calculated according to reservation procedures. By using due date and backward or forward loading, an order could be inserted into the idle time of a machine.

B- Arrival stage:

The second stage is the arrival stage where orders arrive at the material store. This stage could also be called the material preparation stage. Here materials are prepared to enter the shop at the right time.

This starts with a queue called the shop-arrival queue (SARQ). The job whose arrival time is due is placed in this queue. This queue is arranged according to the FRFS rule. There are two ways to control job arrival at the material store. The first method is to use queue length as a trigger, while the second way uses arrival time. The arrival time thus controls the arrival of orders at the material store.

The next job movement is to material store where queues are called main-queues (MAINQ). They are divided into a number of queues in accordance with the number of available machines. MAINQs again are arranged according to the FRFS rule. Selecting a job from that queue is a matter of determining which job is the most suitable for dispatch into the shop. More detail is shown in example 5.1. The following example shows the method of FDSSD rule in selecting the most suitable job to dispatch.

Example 5.1: (See table 5-1)

1- The sequence proposed according to the FRFS rule is A-B-C-D.

2- Processing job A first means it will finish 19 hrs. earlier than due date whilst job B and C are late. Note that job B or C may be processed before job A without any delay to job A.

3- Therefore, the sequence could be B-A-C-D, because job A would not be affected if job B comes before it.

4- Check job C with respect to job B, then with respect to job A. Job C is received later than B, and job C may delay job B if job A and C are processed before job B. Therefore, job B remains in its place while C will be checked again with job A.

5- Although job C is received later than job A, job C may be processed before job A without delaying it.

6- Sequence could be B-C-A-D.

7- Check job D with all jobs which precede it, as explained above.

JOB	RECEIVING ORDER	DUE DATE	PROCESSING TIMES	SLACK
A	1	29	12	17
B	2	16	13	3
C	3	9	3	6
D	4	22	6	16

Table 5-1 Example 5.1

8- Check job C with job B. Job C could precede job B without any problem.

9- Final sequence in this case could be as C-B-A-D.

C- Entry stage:

This stage is an intermediate stage between MAINQ and the shop. At this stage, the first step in the shop is to form main-in-process-queues (MINPQ). MINPQ and MAINQ follow more or less the same strategy, except that MAINQ is located out of the shop while MINPQ is located within it, i.e more consideration is given to the shop-floor in case of MINPQ. MINPQ has only four limited spaces in its local buffer at each machine while MAINQ is controlled by the scheduler. If MINPQ is full then jobs will be stored in the global buffer making another queue called hereafter Q23. The Q23 follows the FRFS rule in arranging jobs sequences.

D- In-process stage:

After the MINPQs are prepared then the same strategy which is applied to MAINQ, is followed in arranging the queues within the shop. In addition, selecting a job from the MINPQ is similar to the MAINQ, but with an extra factor- machine utilisation. Therefore, selecting a job from the MINPQ to be machined will be in accordance with the job requirement, the shop requirement, and the machine requirement. As mentioned in chapter 3, the job requirement means a consideration given between the received order and the due dates. The shop requirement is mainly concerned with WIP and queue length, while the machine requirement is concerned with utilisation and idleness (see example 5.1).

In example 5.1 difference between some of scheduling rules has been noticed (see table 5-1). The FDSSD rule is not the optimum rule in this case, but the FDSSD rule has an advantage over the other rules. It considers the early received job. As shown in table 5-2 that the EDD rule performs relatively well. One job is late 5 hours, while in the case of the FDSSD rule, the job is 12 hours late. The EDD rule caused job C to be processed before job A and B, and job A becomes late because of job C and other jobs such as D. In the case of the FDSSD rule, job A precedes job D, consequently job A would not be late. This applies also to job B and C. If job B precedes job C then C will become late (as with SLACK). If C precedes B it will be fine, i.e job B is not late. The main difference between the EDD and FDSSD rules is that the earlier the job is received the higher priority it will have in the queue in the case at it becoming late in the sequence.

In addition, the FDSSD rule could perform much better in a shop where every one in the system is aware of its use. This is because the FDSSD rule sup-

RULE	SEQUENCE	FINISHING TIMES	LATE(-)/EARLY(+)				LATE	N#	EARLY	N#
			A	B	C	D				
FDSSD	C-B-A-D	03 - 16 - 28 - 34	+1	0	0	-12	12	1	1	1
SPT	C-D-A-B	03 - 09 - 21 - 34	+8	-18	+6	+13	18	1	27	3
EDD	C-B-D-A	03 - 16 - 22 - 34	-5	0	+6	0	5	1	6	1
SLACK	B-C-D-A	13 - 16 - 22 - 34	-5	+3	--7	0	12	2	3	1
FRFS	A-B-C-D	12 - 25 - 28 - 34	+17	-9	-19	-12	40	3	17	1

Table 5-2 Example 5.1 output timing

ports principles (mentioned at 5.4.2) such that no right is given to any job to be finished first if it causes a job received earlier to become late.

Example 5.1 is a very simple case. If more machines are involved and more jobs queued, then the situation will be considerably difficult especially in the case of a dynamic job shop with a continuous arrival of jobs.

5.4.4 THE FDSSD RULE TECHNIQUES:

This section will explain the heuristic procedures employed within the FDSSD rule. As mentioned previously, job shop structure is divided into four main areas; operator, material store, shop-floor and product store. The structure of FDSSD rule follows the structure of the described job shop, i.e with receiving, arrival, entry and in-process strategy, in that order. For more detail see example 5.2.

EXAMPLE 5.2

The following example presents the main heuristic procedure at the reservation stage. Figure 17 represents a number of jobs that have already been received. Job E has just been received recently, so the following procedures are made:

The operational Due date of job i= DDi
The receiving time of job i= RTi
The start machining time of job i= STi
The operational Processing time of job i= PTi

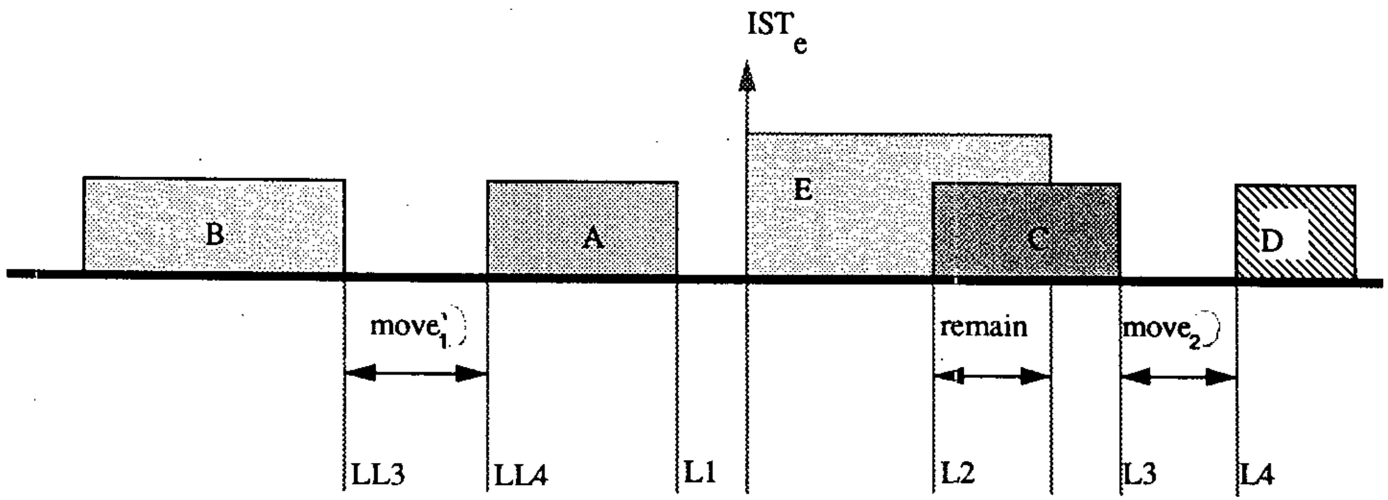


Fig 17 Example 5.2

1 - On the time scale of the machine check the time which equals STe (of job E).

2 - If the machine is busy at time STe then find the nearest preceding gap. If there is no preceding gap then go to step number 6.

3 - Determine the limits of the preceding gap. In this example the gap starts at L1 and finishes at L2.

4 - If PTe greater than the length of that gap (L2-L1), then calculate the difference between the processing time of job E and the gap length, then go to step 6.:

Diff= PTe - (L2-L1)

5 - Fit job E in such that the new due date (delivery date) is equal to L2.

6 - Determine if a gap exists after the DDe.

7 - If there is a gap then find its limits. This gap starts at L3 and finishes at L4.

8 - If Diff is less than L4-L3 then go to 11.

9 - Move block C forward and block A backward.

10- Fit job E into the extended gap.

11- If job E has not yet been inserted then put the job at the end of the scale (after D).

The FDSSD rule involves heuristic procedure in three principle different ways. First heuristic procedure is taking place at the reservation process. In general, the reservation processes applied within the FDSSD rule are Job Oriented Heuristic procedures. Job Oriented Heuristic procedures schedule one job at a time, i.e. all operations of a job are scheduled on all machines before considering the next job [37,53]. The rule herein, attempts to fit the operations of a job into a position as near as possible to its operational due dates. The procedures are designed to use both backward and forward loading procedures. If an operation does not fit before its operational due date then it could be inserted in the nearest preceding (i.e early) gap. If the gap is smaller than the operation processing time then the nearest late gaps. If there are several small gaps available around the required time it may be possible to re-schedule some of the other work in a manner which collects these gaps together into one useful period of time. This can only be done if other works are not adversely affected. The amount of delay allowed could be specified (see example 5.2).

The second heuristic procedure is used in order to form the SARQ queue - the main shop arrival queue. The FRFS rule is employed in ordering the queues. If a machine is idle and there is a late job then it could be forwarded in order to utilise the machine. Jobs thus inserted should not affect the schedule.

The third heuristic procedure takes place at MAINQs and MINPQs when selecting a job that is required to enter into the shop (in MAINQs case) or to be loaded onto machines (in MINPQs case). Dispatching heuristic procedures are followed to satisfy delivery date commitment. Alternate operation heuristic procedures are employed (see example 5.3). Firstly, the queues are scheduled according to the FRFS rule. Secondly, jobs that could be dispatched (green jobs) without creating a negative slack to the jobs that precede them are found. Again, from green jobs, selecting the proper job to be dispatched depends on the state of the shop - machine idleness and WIP.

Finally, a mechanism for releasing orders to the material store is employed in order to control the arrival of job to the material store. This consequently controls the WIP in the shop. Arrival, entry times and WIP are used as a trigger in this mechanism.

EXAMPLE 5.3

The slack of a job i = SLi
The processing time of a job i = PTi
The ordering sequencing number of job i = Mi
The total ordering number= M (see table 5-3)

Three jobs A, B, C are placed in a single queue. The following procedures are part of the FDSSD rule heuristic procedures in the third above mentioned category. The aim is to select a job and dispatch it.

JOBS THAT MAY BE DISPATCHED	VALUE OF 'M'
A	1
B	2
C	4
A & B	3
A & C	5
B & C	6
A, B & C	7

Table 5-3 Jobs that may be selected for dispatching (Example 5.3)

FDDSD solution procedures:

The different sequencing orders are A-B-C, A-C-B, B-A-C, B-C-A, C-A-B and C-B-A. The following procedures are going to:

A- Firstly, find the possible jobs to pass.

B- Secondly, compare between these possible jobs, then select the best job to be dispatched.

In the first step, the Fair delivery principle is used whilst in the second step, state dependent considerations are used.

The number of different sequences which are concerned herein is reduced in order to focus on the first job in the queue. The sequences are divided into three group:

- The first group is A-B-C and A-C-B.
- The second group is B-A-C and B-C-A.
- The third group is C-A-B and C-B-A.

In the first group, all cases are neglected since job A has already the highest priority to go first (according to FRFS rule). In the second group, job B is proposed to precede job A in both cases as then the problem could be reduced to a single procedure; comparison between job B and A. This is because in the second instance, job B has been scheduled to precede job A so in the first case it is also accepted in that position. In the third group, both cases should be considered because the first job in the queue (C) is at the tail of the queue when the FRFS rule is used. Therefore, the number of sequences is considerably reduced (see Fig 18).

There are two procedures in the FDSSD rule could be followed to Select a job to be dispatched (see Fig 18). These procedures are as follows:

A. Finding the jobs that could pass:

1- The FRFS rule is applied to put jobs in non-decreasing receiving time order. Put M= Ma (jobs A has Ma=1, B has Mb=2 and C has Mc=4) (see table 5-3). The considered sequences are: A-B-C and A-C-B.

2- If job A is late, then go to step 7.

3- If Slackof job A (SLa) is less than th processing time of job B (PTb), then go to 5.

4- If SLa >= PTb then M = M + Mb. The considered sequences are B-A-C and B-C-A.

5- If Job B is late, then go to 7.

6- If (SLa >= PTb + PTc and SLb > PT or SLb >= PTa + PTc and SLa > PTc), then M = M + Mc. The considered sequences then are C-B-A and C-A-B.

7- Go to the next step to compare and select the optimum job.

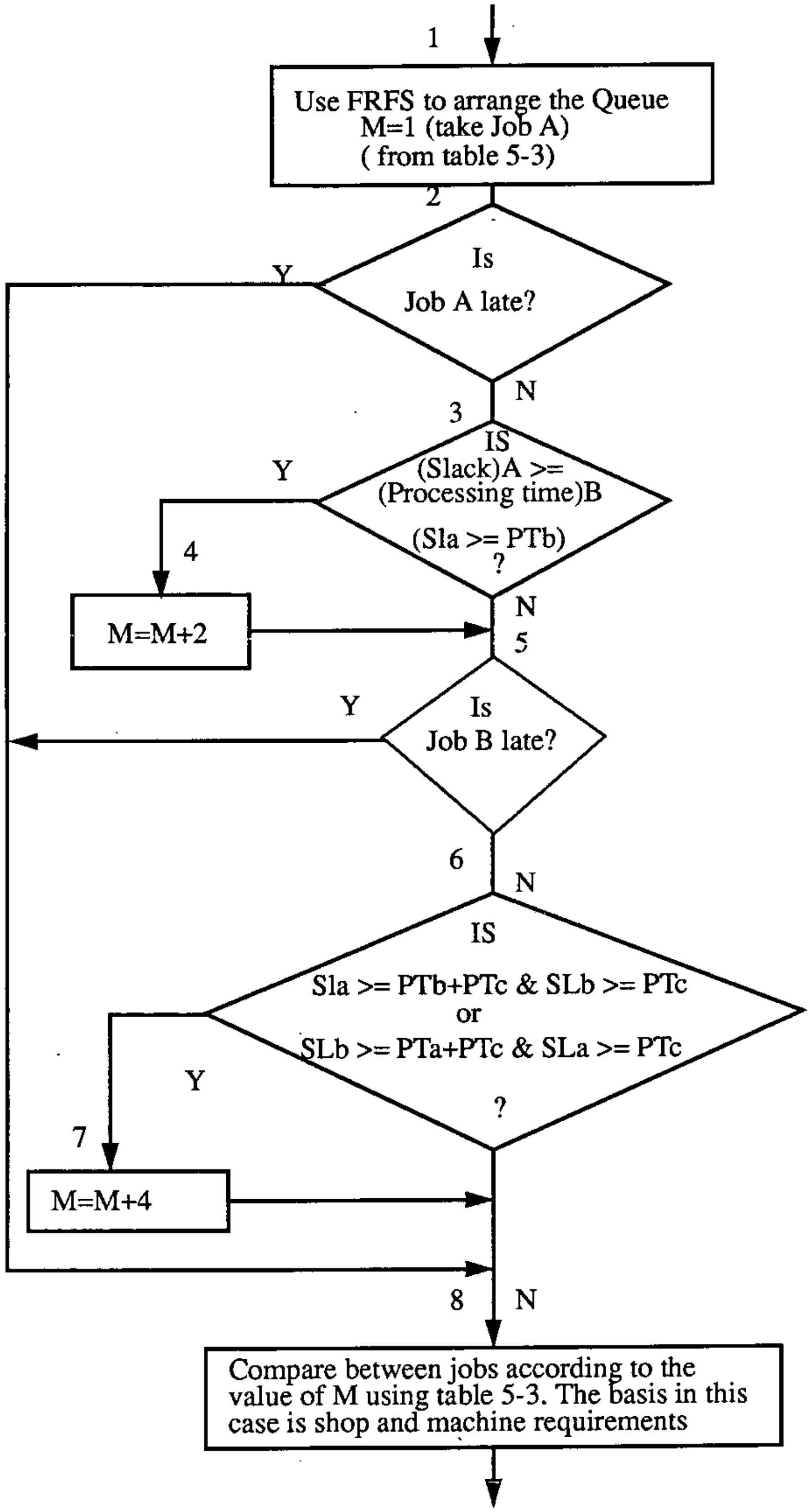


Fig 18 The procedures Of Fair Delivery principle

B. Comparing and selecting:

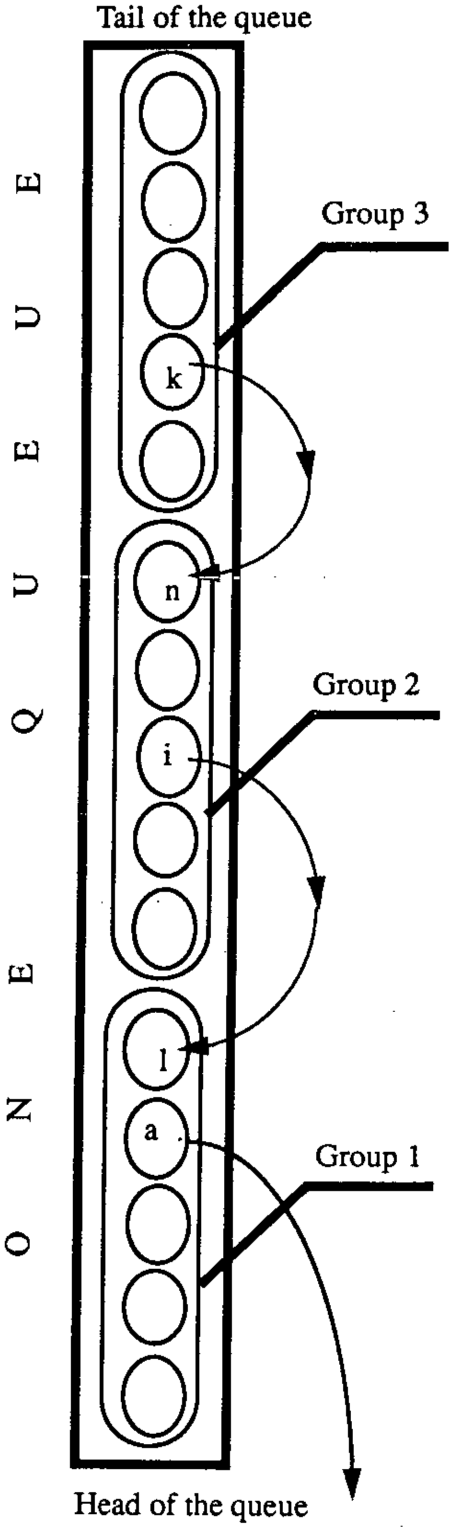
According to the value of M and the queues at other machines (if any), a comparison is made between jobs at the queue. Table 5-3 presents the corresponding value of M for each job in the queue.

Under JSSM the queue is divided into a number of groups each group consisted of 5 jobs and the previous procedures are applied to each group separately. The selected job from the last group in the queue is compared with the last job in the preceding group using the same basis (Fair delivery basis). Earlier jobs may be alternated with later jobs. Then the above procedures are repeated again in the preceding group till all groups have been completed (see Fig 19).

5.5 SUMMARY:

The FDSSD rule involves several heuristics such as job oriented heuristic and alternate operations. Decision making within the FDSSD rule employs the following strategies:

- Due date based rules are used at each machine,
- The FRFS rule, in case of jobs' due dates overlapping, is employed, and
- The shop's and machines' state, are considered in making in-process decisions.



(1) At each group apply FRFS rule to arrange the jobs in order the earlier received the nearer to the head of the queue.

(2) Use FDSSD rule to employ Fair delivery principle in order to select a proper job (e.g job k from last group.

(4) If group one is reached goto step 6

(3) Compare between job k and the last job in the preceding group (e.g job n). Using FDSSD rule, If job k is higher priority then k alternates with job n.

(5) Repeat the procedure from step 1 but on the preceding group with the provided selected job till group one is reached.

(6) Dispatch selected job from group one to machine.

Fig 19 Sequencing and dispatching within FDSSD

These strategies are applied differently in each of following stages in the shop: receiving, arrival, entry, and In-Process. First of all, at the receiving stage, a reservation is made to load each received job a on a machine's time scale. In this case, the first and second strategies are used. Secondly, in arrival stage, machine idleness is taken into account by using a release mechanism in conjunction with times that have been set at the reservation stage. The second strategy is used for SARQ sequencing. Thirdly, in the entry and in-process stages, several heuristics are involved; jobs are sequenced according to the FRFS rule, a dispatching procedures are made to distinguish the jobs that may by-pass, then a general assessment of the other machines' local buffers is followed by selecting the right job to be dispatched - entered into the shop or loaded at the machine.

The FDSSD rule is an attempt to translate and implement ethically based decisions into a practical form. Some of the ethics involved are given above principle are listed. At the same time, the rule's aims are similar to most other scheduling rules in a job shop environment; timely delivery, low WIP and high machinery utilisation.

CHAPTER 6

JOB SHOP SIMULATION MODEL

CHAPTER 6

JOB SHOP SIMULATION MOPDEL

6.1 INTRODUCTION:

A job shop simulation model (JSSM) is especially designed and developed, herein, for experiencing scheduling problems and investigating scheduling rules in a job shop production environment. A number of rules is included within the JSSM. Developing a new scheduling rule is possible, especially if the new rule involves a combination between available scheduling rules within the JSSM. The JSSM is a repository model because all related information in a simulation run is saved in a three dimensional array. The JSSM is able to construct a schedule of a number of jobs to flow through several machines.

In this research, the purpose of using computer simulation model is:

1- To develop a new scheduling rule such as the FDSSD rule.

2- To investigate, evaluate and compare between some scheduling rules and the developed the FDSSD rule.

3- To experience some scheduling problems such as queue building, congestion and tardiness problems.

4- To help scheduler in setting up feasible delivery dates.

The model is called Job shop Scheduling Simulation Model (JSSM). The program is written in Fortran77 language which is available at Sun workstations 3/50.

6.2 SIMULATION TECHNIQUE:

Simulation could be defined as a representation of an activity, by a simple form of another activity. Simulation, hereafter, may be used to mean computer simulation. In addition, simulation model could be used to gain some more experience or to understand a real system and its related problems. There are many types of simulation models; physical, analogue, schematic and symbolic model [66]. Because this thesis is concerned with computer simulation, it will use a symbolic simulation model.

Simulation is a technique by which many complex problems could be investigated efficiently and solved in more practical way than mathematical analysis techniques. Simulation technique is used, herein, to represent a job shop production system. It is a difficult task to provide schedules using an analytical formula or by solving a set of equations that may describe the operations scheme. Also, it is not feasible to carry out expensive trials on the real system itself; it would take too long and could upset normal production. To simulate a production system, two types of data could be employed. The first type of data can be based on real system data if there are sufficient available. The other type can be based on number generation from the property values which are involved in the process.

As mentioned above, the model is designed to investigate scheduling problems. A real system could be studied in a cheap way and without interruption. Computers can be used for this purpose. Computer simulation may represent a real system by constructing a program in which significant elements of the real system are included within the computer program, with scheduling rules and techniques that are going to be examined. In addition, a time advancing mechanism is required to obtain a dynamic behaviour situation. Briefly, simulation model is used as an evaluation tool to compare, measure some values and to present the production system in a relatively simple and cheap way. This model is used to determine when and how the decision is going to take place [24,32,66].

Simulation technique is one of well known methods in studying a job shop scheduling problem [32,33,54]. Analytical method is another technique which is used mostly in static job shops where data is fixed and required many assumptions to simplify the procedures. It uses mathematical parameters, functions and expressions to solve problems. Analytical method is relatively a complex procedure with respect to simulation method. The simulation method could be considered as the other main technique, especially, in studying and investigating the behaviour of a queuing system with many variables related to the scheduling problems [10]. Many researchers are concerned with simulation technique because of its ability to represent the dynamic behaviour of the job shop. Furthermore, it could be more practical so that various decisions can be examined under accepted real conditions without a great loss [1,34,38,48,58].

6.3 TYPE OF SIMULATION:

In the widest sense the term simulation refers to the use of the behaviour of the real life object or system. This could be small scale physical imitation of the real object. Also it could be a mathematical model where equations and logical rules represent the system under investigation.

Sometimes, discrete changes are involved. The developed model is concerned with the simulation of discrete systems. A sequence of data is processed according to the scheduling rules to study its behaviour.

Available data could be a real data when it is long enough to be processed. Also, it could be generated according to the available property of the real system. Processing times, due dates and route of each job are some examples of the data required.

6.4 BUILDING A SIMULATION MODEL:

Two items of building a simulation model can be distinguished and will be discussed at this stage. They are:

1- Model formulation: the rules which may describe the system are being studied.

2- Data generation: it is the set of numbers that is used by the model to represent a form of real data.

6.4.1 FORMULATION OF THE MODEL:

If the model is logically simple, then a set of rules to describe its behaviour can easily formulated. However, the system may be complex, such as the queuing problems which are common in production systems. These problems have a large proportion of simulation studies.

Generally speaking, production systems can be described as a number of machines through which many jobs flow. On the machines a number of activities is carried out, i.e processing or idling. A processing time is associated with each process. When a process is completed the job may flow to another machine if the later machine is appropriate. The job may flow out of the system, i.e exiting.

The system must be examined frequently to see how it is operating. Two methods to do this job are available. The first one is called slicing or constant time-step technique. It monitors the system at regular intervals and collects information from the shop. The second one is called discrete event or next event technique. It examines the system only when an event takes place in the system such as loading and unloading. In the second case, the simulated time in the model is advanced to the next earliest event.

6.4.2 GENERATION OF DATA:

The data required for a simulation study is usually a real data, i.e actual processing times and inter-arrival times. However, data in this form may be insufficient for a long simulation run and generally may not be as flexible as sampled data. A model could involve a representation of an actual data by generating it randomly according to the actual property which is common used in the real systems. In this study, the NAG library subroutines are employed to generate processing times, due dates and routing of each job. Also the inter-arrival times are generated in which loading rate could be determined.

6.5 VALIDATION OF THE MODEL:

Having built the simulation model, it must be thoroughly validated. The logic of the model and the data distribution must represent the real system.

In some systems, it may be practicable to compare simulation results with actual ones. In other words, simulating a small problem then comparing the results with hand simulated results may be possible. Validating a model could be made by printing the results regularly to investigate the logic within the model. In other words, validity could be insured by (in this work the fifth way is mainly employed):

1- Adequately defining the problem that the model is constructed to address.

2- Identifying the relevant model components.

3- Identifying all assumption employed in the model.

4- Observing the performance of the model under different conditions.

5- Comparing modelled result with real or with calculated output data.

6.6 THE JSSM MODEL DESCRIPTION:

This section will discuss the main goals of the developed model and how the model could represent a job shop system. The structure of the model is discussed. As mentioned previously, the problem is to investigate different scheduling rules in a job shop system. The JSSM could simulate up to 10 machines. Also, it is possible to simulate more number of machines if there is enough memory space to run on the computer. Fortran77 language is used to write the program. The program consists of three main parts; input, in-process and output block (see Fig 20).

The simulation model JSSM is designed to investigate and compare the effect of scheduling rules on the schedule performance in a job shop environment. The JSSM also could be employed to set a practical delivery dates that system may keep them up. There are many sub-goals that may be achieved in future such as using JSSM for learning purpose in building and understanding scheduling procedures. Also, the flexibility in changing any value in the model while it is running without interruption to the process, helps in understanding the effect of each value in the system on the scheduling procedures.

There are two types of timing procedures which are involved in simulation models. The first type is called slicing simulation. In this type, time is advanced to a new value (usually by one unit of time), then the system is scanned to find if there is any required action to be done. The second type is called discrete-event simulation. By using the second type, time is advanced according to the nearest next event. In the JSSM, both ways could be used separately or together.

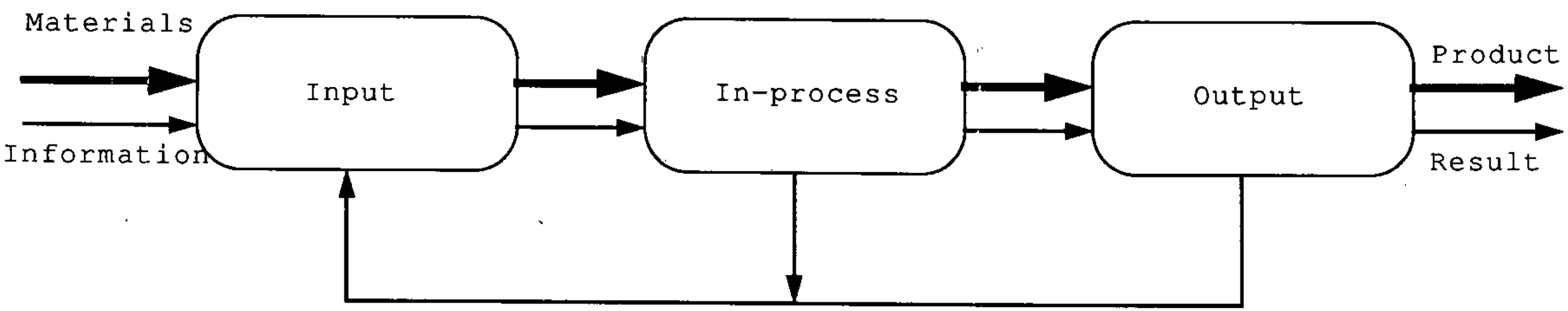


Fig.20 Flow of information and materials

Static and dynamic job shops could be simulated using the JSSM - static job shop means that no more accepted orders till the received jobs are completed while dynamic job shop accepts any new orders. Job receiving rate represents the number of received jobs within a unit of time. Setting receiving rate to zero, means the job shop is static. Job receiving rates may be called inter-arrival rate of jobs.

The JSSM is a repository model - all related information in a simulation run is stored in an array. The JSSM, however, could be made to use file storing. The array consists of three dimensions. Each dimension represents one physical element of a job shop; job, machine and shop. The information is stored according to these elements. Information could be called or changed at any time within the JSSM. For more detail see Fig.21. Data storing and structure will also be discussed below in section 6.7.4.

6.7 THE SIMULATION PROGRAM:

6.7.1 SIMULATION INPUT:

Input data could be either randomly generated or predetermined data. Predetermined data is relatively more practical than the other one, because under predetermined data a real one could be considered.

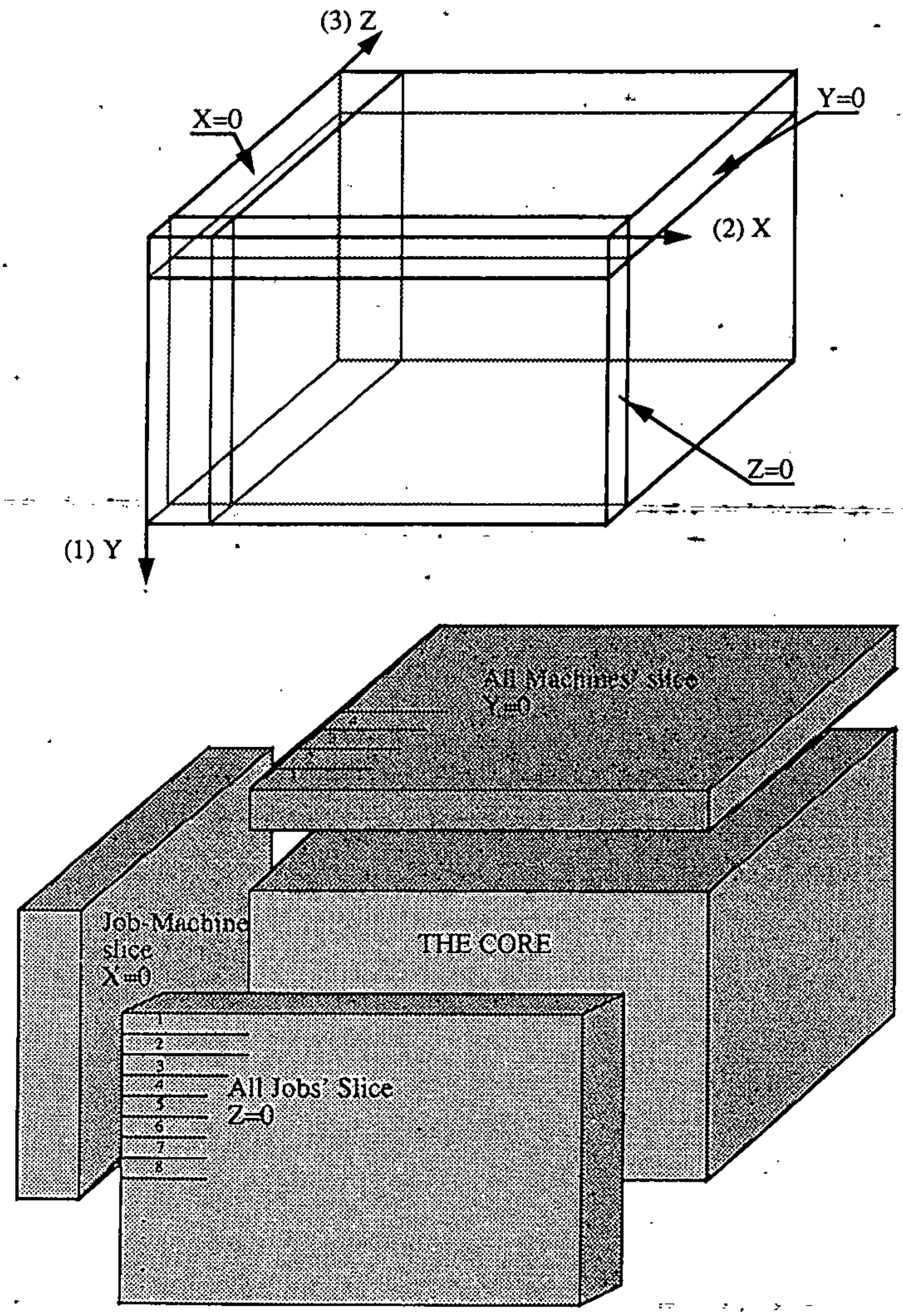


Fig 21· Array Structure: (A) Dimension and (B) Slices and Core

The input data of the model consisted of two types. The first one is the data that is received before a simulation run starts. This type of data is called, hereafter, initial data. It is the only used data, in static job shop simulation. The second type of data is called continuous data because it is received after the simulation run has commenced. Initial data is saved in file 'ST10'. Continuous received data is saved in files 'D.A' and 'D.N'. Initial data consists of the information of jobs and machines. Initial and continuous data are discussed in more detail in section 6.7.3-A.

In this thesis, generated data is selected to accomplish the scheduling task. In addition, the model could read real data if there is a sufficient available data. The NAG library subroutines are employed to generate processing times, due dates and routing of each job (routine). The G05ZYF subroutine and G05DYF are the subroutines that are used to generate routines and processing times respectively. (Appendix 2 includes the program that used to generate the input data).

6.7.2 THE PROGRAM STRUCTURE:

The program is written in Fortran77 on unix operating system at Sun 3/50 work-stations. The flowchart of main events is illustrated in Fig.22. Since the reservation is one of main events for loading machine time scale, a separate flowchart is

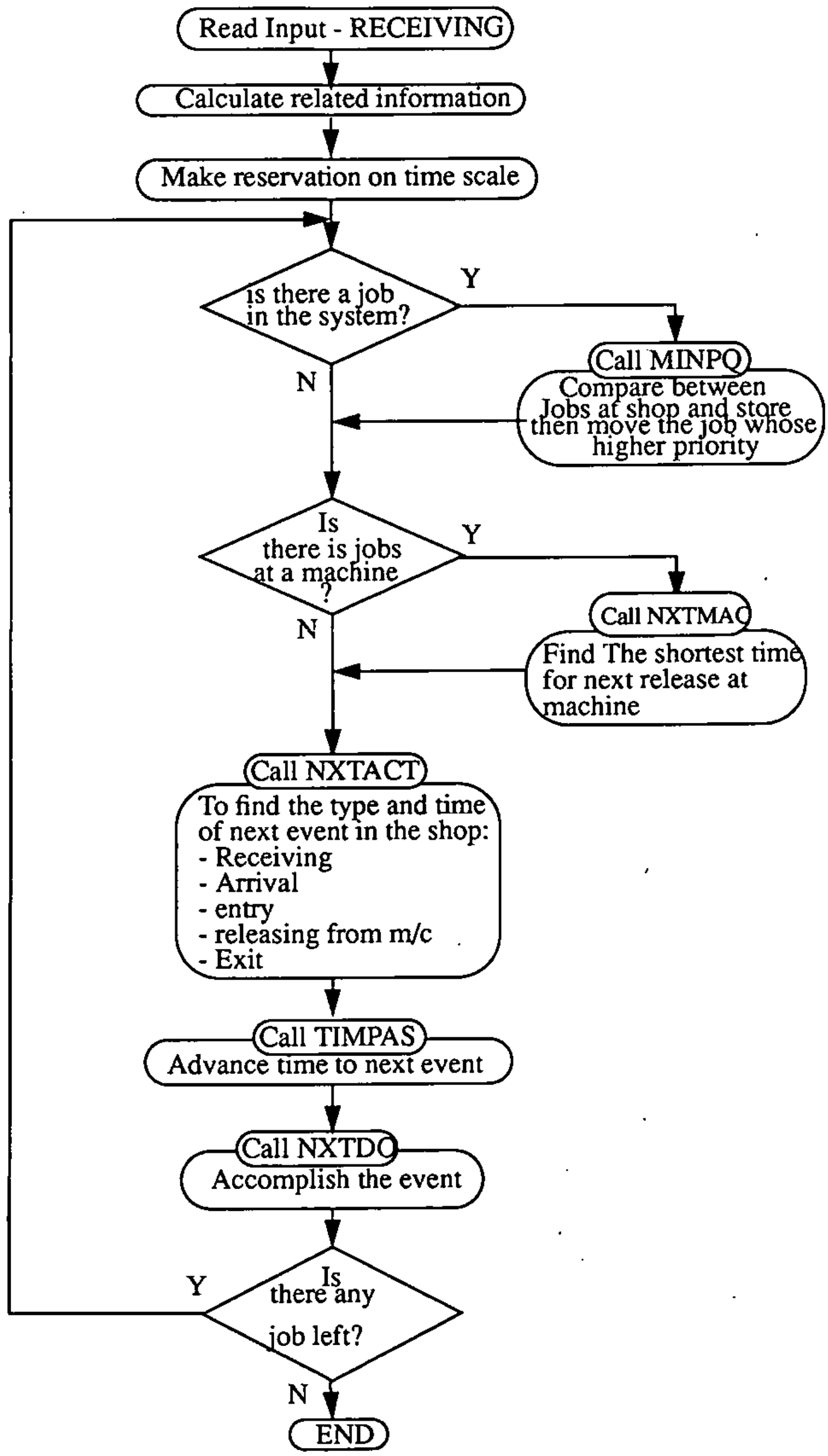


Fig 22 Model JSSM: Simulation procedures

shown in Fig 23. The subroutines of the program are presented in the following section 6.7.2-B.

A- THE PROGRAM:

The JSSM program, developed herein, consists of three major parts (see Fig 15). The first part is concerned with data collection (input). After the input is read, completing operational values are made. Reservation is one of the main processes which is carried out by INSERT subroutine (see Fig 23).

The second part represents the main block that investigates and performs the scheduling procedures. It is called In-process part. A subroutine which is called WIP, involves several subroutines in order to determine the next event; time advancing and event performing. Figure 22 illustrates the general followed computational procedures in the JSSM. The main considered events are as follows: orders receiving, orders arrival, jobs entry, machine loading and unloading.

The third part is concerned with output representation and reporting (see section 6.7.3-B).

There are general points could be listed below to conclude some of the main strategies in the program:

- Two types of input, could be used; initial and continuous.

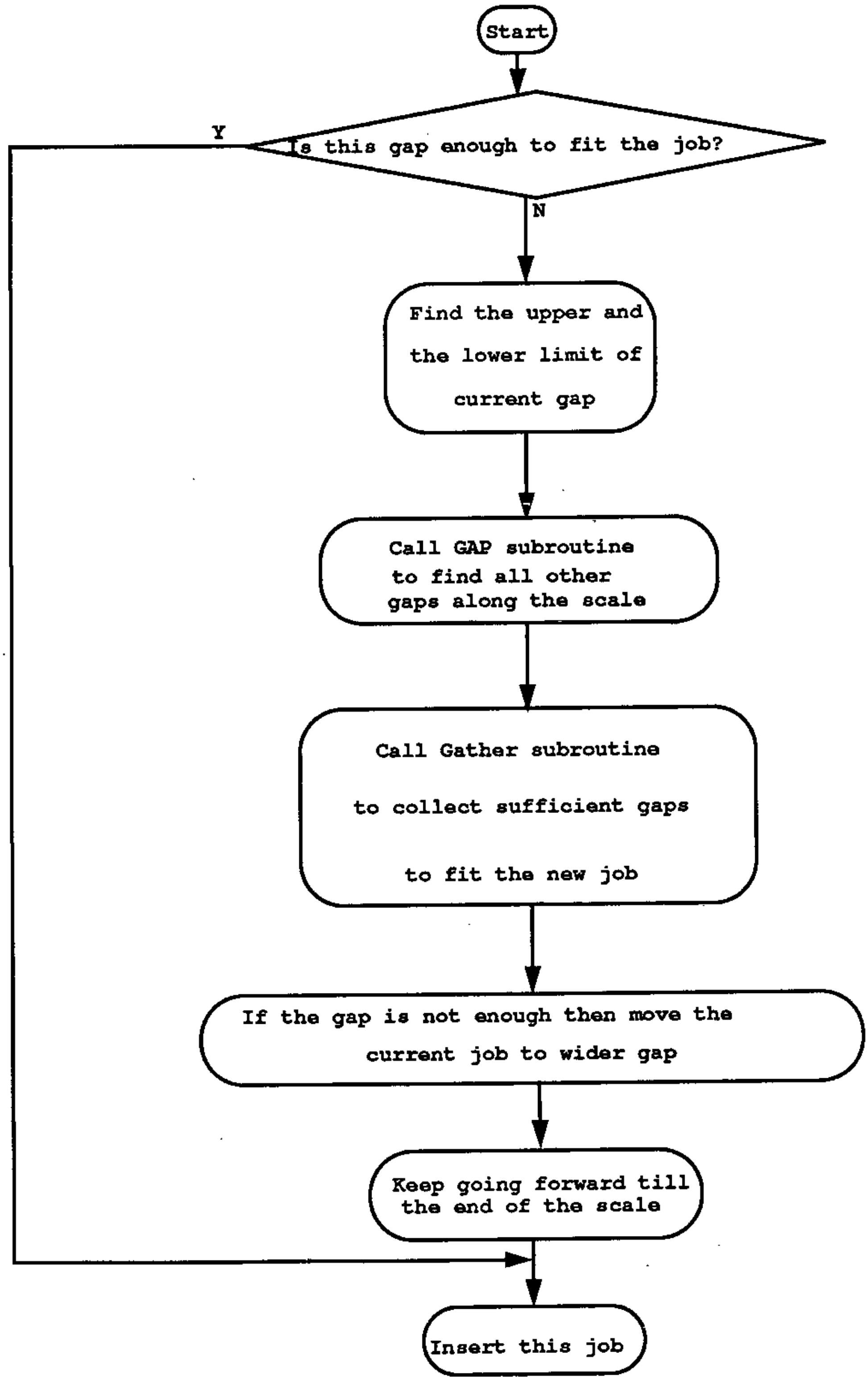


Fig 23 Subroutine INSERT structure

- All information could be adjusted and changed while the model is running.

- Static or dynamic process could be used. Also, alternating between them is possible.

- The capacity of parallel machines could be balanced.

- More events could be added such as breakdown. Rework process is also possible.

- Two types of priority are made; main and sub-priority (operational) index.

- Machine interference could be extended in future work since it has been involved within the program.

B- SUBROUTINES GUIDE LINES

The program consisted of eighty four subroutines. The following list is to specify the main subroutines' function and how it is related to each other. They are:

1 - Main program:

This is the main body of the program. It will call the main menu in MENU subroutine which will carry on scheduling activity. Main program will open few files to make it ready for data reading and result recording. For more detail see section 6,7,3, The MENU subroutine will be called. The menu as it appears in the model would be as follows:

1 - SET ALL TERMS & ELEMENTS. TIME=
2 - REPEAT LAST PROCESS
3 - NEW EVENTS & CORRECTIONS
4 - NEW KNOWN JOBS ARRIVED
5 - SEE AVAILABLE JOBS
6 - NEW SCHEDULE
7 - RETURN SCHEDULING TIME BACK
8 - SET THE APPEARANCE TIME OF THIS MENU
9 - CHANGE THE TECHNIQUE OF SCHEDULING.
10- SAVE & QUIT
11- QUIT
12- SHOW QUEUES
13- SHOW ANY VALUES OF THE ARRAY

2 - Related Subroutine to input data:

Subroutine SETALL sets the environment of the system: scheduling procedures and shop type. This subroutine is called from MENU subroutine. However, subroutine DEFAULT could set all related defaults without using SETALL subroutine. The following menu includes most facilities that arranged through this subroutine:

1-SET ALL TERMS
2-LINES IN BUFFER
3-SYSTEM TYPE
4-DATA ARRIVAL TYPE
5-EVENT TIMING
6-QUEUES
7-RULES AND MEASURES
8-DATE
9-TERMINAL

Subroutine REDSTR starts reading from a file. The default file is 'ST10' which is situated under the directory '.F.FILES'. It is called from MENU subroutine. This subroutine should be called once at the beginning of each simulation run. The model continues its reading of the input data from files 'A.D' and 'A.N' under the same directory mentioned above.

Subroutine REDCNT reads input data continuously from 'A.D' and 'A.N' files. It is called from subroutine NXTDO. When a number of jobs is read. The FILL subroutine will be called to fill all other related information in the storing array in the model. For example, it calculates how many machines that a job is required, expected finishing time and the remaining time.

Subroutine INSERT is called from FILL subroutine. The reservation part within the FDSSD rule is carried out by the INSERT subroutine. Subroutine INSERT1 is called from INSERT subroutine. The INSERT1 subroutine looks for a gap to fit a job in it. Then it inserts that job in, but If there is not a space at that place then the subroutine will calculate the limits of the nearest backward gap. If there is no gap available, then INSERT1 subroutine searches forward to a gap till the last reserved place. If the job is not fixed yet then the subroutine will put it at the tail of reserved scale. Subroutine GATHER1 is called from GATHER and INSERT1. The GATHER1 subroutine moves jobs forwards and backwards on time scale during reservation procedure in order to eliminate some small gaps. It adds them to form a useful gap. Subroutine GATHER is called from INSERT to remove first gap on time scale and add it to the next gap. Subroutine GAP is called from INSERT1 and GATHER to calculate:

- Where does a gap start?
- How long is this gap and the total length of gaps?

Subroutine FILLIN is called from INSERT1. It puts jobs and their related information in the related place on time scale.

Subroutine DATA displays the available data. It is called from MENU. Subroutine JOBINF displays the available jobs' information. It is called from DATA subroutine. The following menu presents the different provided information in order to monitor them:

1) ALL JOBS
2) A JOB
3) FINISHED JOBS
4) SUBCONTRACTED JOBS
5) CANCELLED JOBS
6) DELAYED JOBS
7) STOPPED JOBS
8) LATE JOBS

3 - Setting scheduling defaults:

Subroutine SETALL sets scheduling requirements such as a scheduling rule and a measure of performance. Subroutine CHNTKN is called from SETALL to set the required rule by using subroutine TEKNIK. The menu which will be obtained is as follows (see the following menu as it appears in the model):

```
1- WHICH RULE TO BE RUN?
======================
1 -FRFS.... 2 -EDD.... 3 -SPT.... 4 -FCFS....5 -MRPT
6 -S/ROP... 7 -SPT/EDD 8 -LRPT... 9 -QINM....10-SPTATM
11-COMPOSIT.12-SIMCU.. 13-SLACK.. 14-S/OP... 15-WSPT
16-WEDD.... 17- FDSSD. 18-COVERT. 19-N/A.... 20-LASQ

2- WHICH MEASURES OF PERFORMANCE TO BE CONSIDERED?
=================================================
1-FLOW RATE(SPT). 2-IGNORE TECHNS. 3-CONGESTION IN SHOP
4-MIN WIP........ 5-TARDINESS..... 6-IMPROVE USED RULE
7-CAPACITY........8-CUSTOMER...... 9-TIME SPENT IN SHOP
10-COSTS........ 11-LASQ......... 12-M/c INTERFERANCE
13-BOTTLENECK... 14-PARALEL M/CS. 15- MAINTENANCE
```

Subroutine CHOSNG is called to choose the following items:

1- Rule to be followed
2- Measure of performance.

Subroutine THEORY is called from FILL, NXTDO and TEKNIK. It includes all scheduling rules which are considered by the model. Some rules need a separate subroutine such as FCFS, QINM, SPTATM, SIMCU and COVERT. Some other rules use priority index.

There are two different type of priority: main and sub-priority. Main priorities are generated by PRITYM subroutine while sub-priorities are generated by PRITYS subroutine.

4 - In-process Subroutines:

The WIP subroutine is called from MENU subroutine to control the In-process procedures. Subroutines MINP, NXTMAC, NXTACT, TIMPAS and NXTDO are called. A brief description is provided for each mentioned subroutines:

A. Subroutine MINPQ

It is called from WIP and NXTDO. It compares between the jobs at Global Buffer and jobs at MAINQ queue. The criterion of comparison depends on the employed rule. The main operations in this subroutine are divided into three steps. Firstly, a comparison between the first job at the MAINQ and the first job at the Global buffer for each machine. This comparison based on priority index. In the case of the FDSSD rule, it based on entry time and receiving time. Then, selecting a job from MAINQ may take place if this job has received before the job at Global buffer and it is going to be late).

Secondly, according to the process of comparison, the subroutine either calls PRETOQ or MQTOQ subroutines in order to move the selected job from Global buffer or MAINQ, respectively, to Local buffer.

Thirdly, after the local buffers are arranged according to the applied rule (in the case of the FDSSD rule, the FRFS rule is used), machine loading procedure is carried out. LODMAC17 subroutine concerns with loading the selected job into machine. LODING subroutine would set all the related information.

Finally, concerning the FDSSD rule, the process to select a job from queue to be dispatched or loaded at machine, depends on the Fairness and the Fair delivery principles (see chapter 5 section 5.5 for more detail). Subroutines LODMAC17, LODMAC16 and LODMAC160 are used to select a job according to the FDSSD rule. The Fair delivery principle is presented in LODMAC0 subroutine which it could be called from LODMAC17 and LODMAC160. They are used in Local buffers and MAINQs respectively. The LODMAC0 subroutine uses two other subroutines, LODMAC01 and LODMAC02. They compare between some of in-process information such as queue length at different machines. The minimum queue length at other machines could be determined. It may help in making the decision within the FDSSD rule.

B. Subroutine NXTMAC

It specifies which machine has the smallest remaining processing time, i.e next machine to finish the current process is specified (releasing event).

C. Subroutine NXTACT

It finds the nearest event in the shop; receiving a new order, arrival to stores, entry to shop, release from a machine or/and repair a broken-down machine. It is possible to insert some other events. Then subroutine ACTION is called to compare between times of next events, then it decides which event will be next.

D. Subroutine TIMPAS

It is called from WIP. It advances time and the related times in the model. Advancing time is carried out by next-event technique.

E. Subroutine NXTDO

It calls the corresponding subroutine according to the specified next event in NXTACT subroutine. For example subroutine RELEAS is called from NXTDO to release job(s) from machine(s). Then subroutine RELEAS1 sets the related information. Another example, subroutine REPAIR is called from NXTDO. It considers the specified machine to be repaired according to the specified time. In arrival event, subroutine SARQ forms arrival queue (SARQ). It is called from NXTDO. Then subroutine NEWMNQ is called from NXTDO to add the new received jobs to MAINQ queues. Subroutine MAINQ takes jobs from SARQ queue and put them in separated queues according to first operation of a job.

5 - Changing available data:

Subroutine CHANGE is called from MENU. It may change related information of a machine or a job at any time.

To change some specification of a job, subroutine JBCHNG could be called from subroutine CHANGE. LOKJOB subroutine is called to find the corresponding job that has got the new change. CHANGE subroutine may change following job information:

1- CANCELLED job.
2- SUBCONTRACTED job.
3- DELAY FACTOR.
4- DUE DATE
5- WEIGHT FACTOR
6- COST PENALTY and
7- ROUTING & PROCESSING TIMES

In the case of a new change in a state of a machine, subroutine MACHNG could be called from subroutine CHANGE. LOKMAC subroutine is used to find the corresponding machine that has the new change. Following machine information could be changed while the model is running:

1- CANCELLED
2- BREAK-DOWN
3- DELAY FACTOR
4- MAINTENANCE
5- NEW MACHINE
6- WEIGHT FACTOR
7- MACHING COST
8- IDLE COST
9- STOPPED MACHINE
10-REPAIRED MACHINE

6 - Other subroutines:

Subroutine APPEAR hides the main menu for a specified time. It is called from MENU. Subroutine VALUE could be used to change and monitor any stored information value in the array. It is called from MENU. Subroutine QUEUE2 and subroutine QUEUE1 are called from subroutine QUEUE. Subroutine QUEUE presents queues: SARQ, MAINQ, MINPQ, machines and buffers. Subroutine MOVE moves jobs forward in queues. Subroutine ARANGE arranges queues according to priority index. Subroutine UPDOWN is called from NEWMNQ, THEORY and LODMAC16 to change the position of jobs in a queue oppositely.

Subroutine ZERO is called from TIMBK and NEWSHD when a new run is required. Subroutine ADD is used to add all values in one raw or one column in the array. It is called from FILL. Subroutine ERROR is used to produce error messages. Subroutine TIMBAK may be used to return scheduling back to a specified time and start again from that time. It is called from MENU.

7 - Output results and reporting:

Subroutine FINAL will write down the final report about the results. It is called from MENU and SAVE subroutines. Subroutine FINISH is called from RELEAS1. It records the full history of each job on each machine. (files name= RESULT1-3). Subroutine SEND is called from RELEAS1. It produces a full history of each job that exits from system.

Subroutine INNFFO is called from MINPQ. It records a full detail of each slice in the array used (INFO). In this subroutine a standard output file (fort,100) could be created. This file could be used to repeat a previous saved processes. This facility could be stopped, i.e info(0,23,0)=0. The full detail could be made short if info(0,23,0)=1 or long if info(0,23,0)=2. Subroutine SAVE is called from MENU. It saves all the latest situation in the shop in order to continue the process another time. Subroutine LAST could be used to repeat a last saved process that already saved previously. It is called from MENU. Subroutine ALL is called from INNFFO and FINAL to print all information which is stored in the array, into a file called RESULT7.

6.7.3 USED FILES: INPUT AND OUTPUT

A- INPUT FILES:

1. ST10:

It includes the initial data that the simulation model will start with. In static job shop simulation, this file is the only considered input (see table 6-1 (a) & (b)).

2. D.A:

It includes the orders that will be received after the simulation has commenced (see table 6-1 (a)). For more detail see section 6.7.1.

3. D.N:

It includes two main variables that determine the receiving rate. The first variable presents the number of jobs to be received, i.e number of jobs to be read from file 'D.A'. The second variable presents the period of time required till next receiving event take place.

Available machines = 3

New jobs to be received = 4

state	Job N#	Due Date	Weight	cost penalty	Job routine on machines	Operational processing times on each machine	Receive Time
0	1001	7	6	3	3	6	0
0	2003	125	1	3	2 3	5	0
0	3000	25	3	1	1 3	10	0
0	1004	39	7	3	1 2 3	10 10 3	0

(a) Job information

state	Machine code	Machine Number	available Process	Weight factor	Maintenance When	Maintenance Long	Capacity
0	1	0103*	1	8	12	12	
0	2	0200	2	6	100	10	
0	3	0300	1	2	35	23	
0	4	0400	5	4	70	8	

!! - first two characters indicates machine number
** - second two characters indicates number of next identical machine.

* 0103 means M/c1 and M/c3 are identical .

(b) Machine information

Table 6-1 Initial input ST10 file: a) Jobs and b) Machines

B- OUTPUT FILES:

The output of the model mainly consists of the following files which are formed by the model (see table 6-2 and 6-3):

1. File RESULT1:

This file mainly stores information of mean and total processing time, waiting times, passed time in the shop and how long that each job is going to be late or early.

2. File RESULT2

This file mainly stores information of times of each job at each queue in the shop.

3. File RESULT3

This file includes information of waiting and start machining times at each machine (see table 6-2).

4. File RESULT4:

In this file, the information consists of the number of received job, arrived job, WIP, exit jobs at each step of event in the shop. Also, the capacity and length of queues at each machine are included (see table 6-2).

FILE NAME	INCLUDE INFORMATION
==> PRG.EXP.FNL <==	Programmed due date. Expected due date which is delivery date. Actual delivery date
==> RESULT1 <==	Part number Due date Weight factor Cost penalty Number of operation that this job will have Mean procesing time tardy cost Total waiting time in-process waiting time Passed time in the shop (flow time)
==> RESULT2 <==	Part Number Receiving time Arrival time Entry time Start machining time Finished and exit time operationtional processing time
==> RESULT3 <==	Part Number Waiting time at each M/C Start machining time at each machine
==> RESULT4 <==	Machine to release next Time to the next release Total number of received jobs Total number of arrived jobs Number of WIP in the shop Number of job exit Current time Queue length at each machine Utilisation of each machine
==> SHD.FNL <==	JPart number Arrival ttime Entry time Start machining time at each machine Arrival ttime Entry time Start machining time at each machine

Table 6.2 Output files

OUT PUT RESULT6 File			
	M1	M2	M3
UTILISATION%	98	93	89
PASSED JOBS	106	116	92
IDLE TIMES	15	63	119
TOT.PROC. TIME	1118	1095	1054
TOT.IN-WAIT	2952	3467	3548
MACHINING TIME	87	939	883

THe Following results consider Due date:

M/CS STOP TOGETHER	= 233
THREE M/CS STOP TOGETHER	= 39
MEAN PROCESSING TIME	= 9
TOT IN-PROC WAITING TIME *	= 9967
TOT BF-ENTR WAITING TIME *	= 9385
TOT AFTR-PR WAITING TIME *	= 83833
MEAN FLOW TIME (ARRIVED JOBS)*	= 106
NUMBER OF JOBS	= 208
NUMBER OF MACHINES	= 3
COMPLETION TIME *	= 1002
MEAN COMPLETION TIME(EXIT JOBS)	= 5
# OF JOBS EXIT FROM SHOP	= 173
MEAN JOB RECEIVING(JOBS/HR)	= 12
N# OF TARDY JOBS *	= 98
N# OF EARLY JOBS *	= 72
N# OF ON TIME JOBS	= 3
MEAN TARDINESS *	= -45
MEAN EARLINESS *	= 28
CONDITIONAL MEAN TARDINESS *	= 48
CONDITIONAL MEAN EARLINESS *	= 65
% OF TARDY JOBS *	= 0
TOTAL TARDINESS *	= -7907
TOTAL EARLINESS *	= 4730
ROOT MEAN SQUARE OF COND. TARDINESS	= 114
ROOT MEAN SQUARE OF TARDINESS *	= 86
COST OF TARDY *	= 20961
COST OF EARLY *	= 108
MEASURES OF PERFORMANCE	= 8
SCHEDULING RULE ***	= 3
MAX WIP ALLOWED IN THE SHOP***	= 10
MAX WIP ALLOWED AT EACH MACHINE***	= 6
NUMBER OF EVENT IN SHOP	= 301
PRIORITY: MAIN(0) & SUB(1)	= 0

Table 6-3 File RESULT6: according to Due Date.

5. File RESULT6:

In this file, a brief description on the general items that could be used to evaluate each procedure (see table 6-3).

6. File RESULT7:

A long list of values in the used array is obtained. These values are recorded when the simulation is finished and completed.

7. File PRG.EXP.FNL

In this file the information consists of (see table 6-2):

a. Programmed due dates that are given in input file added to the received times,

b. Expected delivery date which are obtained by reservation using the FDSSD rule, and

c. Actual delivery dates which are simulated.

6.7.4 DATA STRUCTURE: STORING AND USAGE

The information is structured in a three dimensional array according to the physical elements of the scheduling problem. First dimension represents jobs' information. Second dimension represents machines' information. The third one presents shop's information. The array's dimensions are J,50 and M, where J is the number of jobs and M is the number of machines. J is along Y-axis and M along Z-axis. X-axis is limited to 50 spaces (see Fig 21).

A- FIRST DIMENSION: JOB INFORMATION

This slice of the array is a store of related job information. A slice could mean two dimensional sheet which represents only two dimensions (X-Y) at a zero value of the third dimension (Z=0). This slice is located in: Z=0, X= 0 - 50 and Y= 1 - J. jobs have been stored in a vertical two dimensional sheet with Z=0. All related main information of each job occupied the place along X-axis while the operational information is placed along Z-axis according to corresponding machine (see table 6-4). Some of the input information are:

- Job number,
- Job receiving order,
- Due date,
- Cost penalty,
- Machines route, and
- Operational processing times.

Another information is shown in Table 6-2. The priority index, receiving, waiting and remaining times are examples.

Job Status: 1=SARQ,2=ETRY, 5=sbcntr, 6=delayed, 7=cncl, 8=received	
Numbers as arrival order 11=finished	TIME 1
Par /Batch code number	2
Due date: when part is ready?	See que.(0.1 3
Weight factor: job importance. (0=normal .... 9=v.important)	Sched. No. 4
Job delay factor: part movement .(-1=No.delay 0=nrml 1=d 2=stop)	Dly Indicator 5
Cost penalty: $/day	6
Number of machines to be visited	N# of event 7
Where is the part now? (Current machine or next m/c)	Capacity % 8
Number of finished machines: How many?	N# of fini.proces 9
Main priority: the smaller in value the higher in priority.	Tot.proc.time(now) 10
Passed time in shop by the job: (see total at 0.15.0)	Tot.Idle time 11
Processing time of job at ALL machines : tot. process. time .	Tot Proc.Time(all) 12
Waiting time of each job at ALL machines = IN-PROCESS	Tot.I-P Waiting T. 13
Remaining machining time to be finished.	Tot.Rem.Proc. T. 14
Remaining time to be on due date.	Pas.T.All.J.in shop 15
Arrival time to SARQ queue: to shop	2 -stoppage - m/cs 16
Entry time from MAINQ to MINPQ in shop	3 -Stoppage - m/cs 17
Machining start time at first machine	4 - Stoppage - m/cs 18
Finishing time at last machine	Tot.brk-dwn mc T. 19
Waiting time BEFORE entry to the shop.	Tot.Bf Waiting T. 20
Waiting time AFTER finishing and Exit.	Tot.Af Waiting T. 21
Total Waiting time since arrival= 13 +20 +21	Tot.All Waiting T. 22
Job quantity	repeat=1 no=0 23
Number of processes finished so far +1	24
Number of processes job will make.	25
Mean processing time=12 / 25	mean proc.in shop 26
Total cost of delay Or earliness	tot.dly cost 27
Early cost penalty	load in shop 28
Allowance time that job can remain in shop	29
Receiving time of orders to model.	30
Receiving time of order to operator.	31
Machine indicator	32
	33
System Use	34
System Use	35
	36
	37
	38
	39
	40
	41
	42
	43
	44
	45
	46
	47
Z=0 Y-X	X 48
	49

Table 6.4 Front Slice: Job information

B- SECOND DIMENSION: MACHINE INFORMATION

This slice has the location: Y=0, X=1 - 50, and Z=1 - M, where J is the number of jobs and M is the number of machines. In other words, it represents the machine information at the horizontal two dimensional (X-Z) slice at Y=0. It stores the input, in-process, and output machines' information. Each machine has its value on Z-axis while the related operational information is located along X-axis (see Fig 24). A description of the information in this slice is shown in Table 6-5.

C- THIRD DIMENSION: SHOP INFORMATION

This slice has a general information about the shop. It is the two dimensional (Y-Z) vertical slice at X=0. This Slice has different type of information, (see Table 6-6). In general, this information is as follows:

- Timing of the events within the model: The available events are: receiving, arrival, entry, loading, breakdown, maintenance and release. It is also possible to add some more events to the model.

- Receiving control: It will be controlled by two things: random number and predetermined receiving rate to the system. Receiving rate could mean the number of jobs that system will receive within one unit of time. If receiving rate equals zero, then

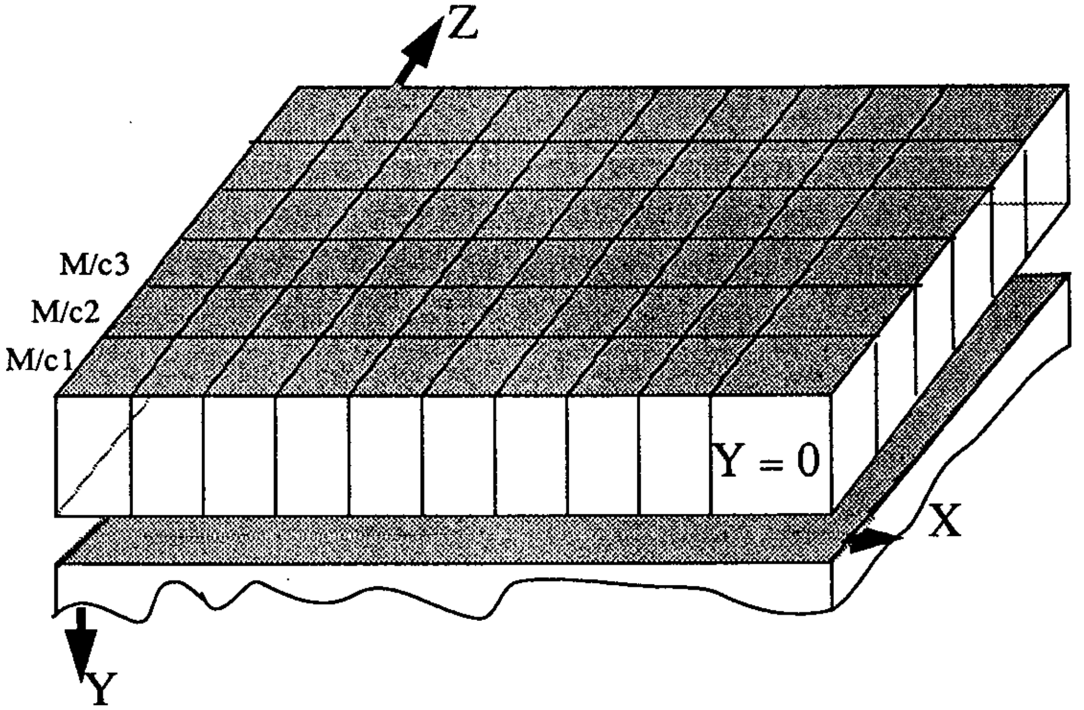


Fig.24 Machine Information Top Slice

0	Machine status: 0=norm ; 2=breakdown ; 5=maintenance ; 10=stopped	Z →
1	Machine system n#	
2	Machine code n#	
3	Type of process available at m/c	
4	Weight factor of m/c	
5	Machine delay factor -1=send any 0=norm 1=delay sending 2=stop sending to m/c	
6	maintenance ——When start.	
7	——How long	
8	Utilisation sofar	
9	N# OF JOBS PASSED THROUGH M/C SO FAR	
10	N# OF JOBS WILL PASS THROUGH M/C	
11	ACCUMULATED IDLE TIME OF M/C	
12	PROCESSING TIME OF ALL JOBS AT THIS M/C	
13	WAITING TIME OF ALL JOBS AT THIS M/C	
14	MACHINING TIME OF EACH M/C	
15	Q U E U E S	
16	Q U E U E S	
17	MACHINES /WAITING LINES SEE X=17	
18	Q U E U E S	
19	Breakdown total time	
20	Start breakdown time	
21		
22	Parallel m/cs	
23	Q U E U E 23	
24	How many visits	
25	machining cost	
26	idle cost	
27		
28		
29		
30		
31		
32		
33		
34	System Use	
35	System Use	
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48	▼X X=Z Y=0	
49		

Table 6-5 Top Slice: Machine information

Z		DUE Due date factor	SYS	SYS	Where and What 5. Arrival 6. Release 7. Load	At M/c	next M/c to release a job MAC	Max processing time	M/c has Max processiing time	Min. Processing time	M/c has Min. Processing time	N# M/c
Next arrival to MAINQ	Next arrival to SARQ	Next entry time	Next Repair time	More than one action 1.arr & Rel 2.Arr&load 3.Rel & load 4.All	Next ACTION time	Next load time	Next Release time	Next Receiving time	Receiving Rate How many jobs / unit of time	Next menu appearence time	RECEiveing Rate	main menu appearence every N# min
						SYS	Type of the receiving pattern of jobs				SYS	SYS
	SARQ allowed length	MAINQ allowed length	MAINQ (now)	# of jobs in the shop (now)		# of New Jobs to be received Next	# of old jobs when new jobs are received	# of subcontracted jobs	# of Arrived jobs	# of Exit jobs	# of Cancelled jobs	N# of jobs JOBS
		SYS	# of late jobs	# of on-time jobs	# of early jobs	SYS	total Earliness	Total Tardiness	Mean earliness	Mean tardiness	Conditional Mean Tardiness	Mean completion time
								SYS	SYS	SYS	Cost of Earliness	Cost of Tardiness
			Allowed WIP in the store	total processing time	Consider Mean or	Job's number to be serviced	Degree of customer importance	allowed time after due	Allowed time before due	Length time of a shift	SYS	SYS
Arrange MAINQ according To 1,36, ??.	When Model become Static	allowed WIP in the shop	MINPQ allowed length	RULE	Measure Of Performence	Priority: Main or operational	SYS	SYS	SYS	discrete or slice	Static or dynamic	System is 1= preemptive 0= non
						Operational information						Y

Table 6-6 Side slice: Y-Z at X=0

the system becomes static, i.e no more jobs are expected before all current jobs are completed.

- Scheduling rule selection: A number of scheduling rules is involved. A selection process is possible. Also, switching between rules is possible.

- Setting values and limits: There are many values may need to be set. Constants and starting condition of other variables.

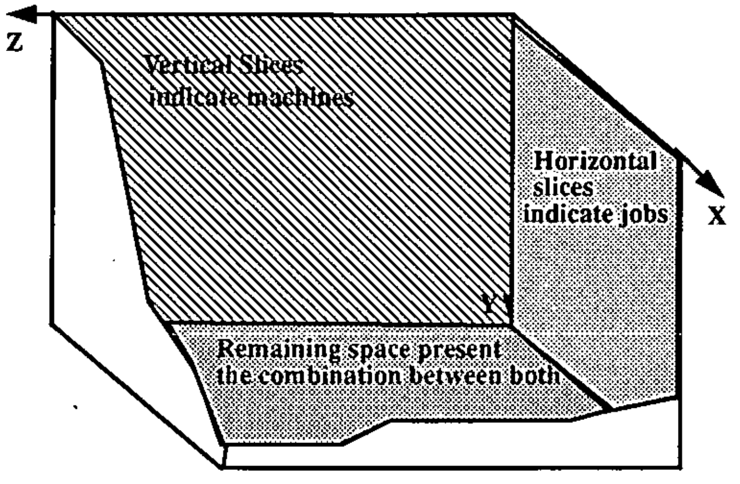
There is another slice may represent the shop queues, especially in-process queues. It is the vertical two dimensional (Z-Y) slice at X=17 (see Fig 25).

D- CORE OF THE ARRAY:

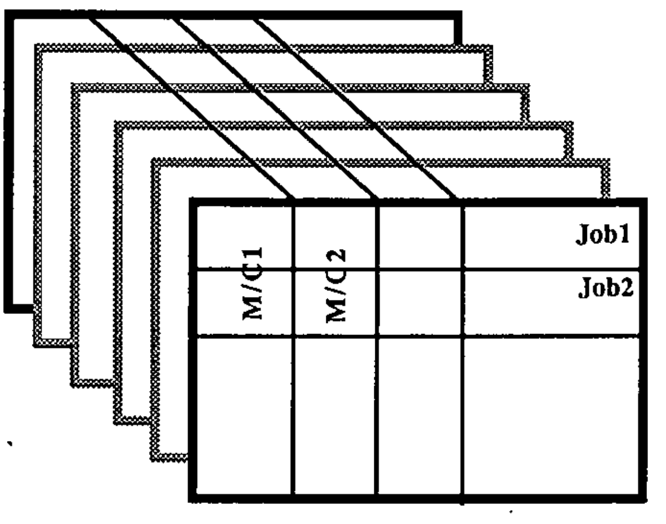
This is the part which is concerned mainly with in-process information and related operational conditions such as queues and timing. The core of the array starts from X=1 to 50, Y=1 to J, and Z=1 to M, where J equals the number of received jobs and M equals the number of machines. Each slice carries different information. The first ten vertical slices are a combination between jobs' and machines' information. Some of the stored information does not change during the simulation process and some other change due to time passing. For more detail (see Fig 26 and Table 6-7).

Machine 4	Machine 3	Machine 2	Machine 1	
				1
	Jobs in Queues			2
	Local Buffer			3
				4
	Expected finishing time			5
	Starting time on machines			6
	Queue length......Time			7
		......Number		8
Processing times of each job at machine				9
Remaining processing time on each machine				10
				11
	Start queuing time in			12
	local buffer			13
				14
	Time when machine become idle			15
	Queue length of global Buffers			16
Total processing time in the shop for each machine				17
				18
				19
				20

Fig.25 Machine and queues information in slice X=17.



(A)



(B)

Fig 26 The Core of the array

0	Z →
1	Routing : processes sequences
2	Subdue date 1
3	subdue date 2
4	In process weight factor for each process at each machine
5	In process delay factor
6	
7	
8	Where is the job now? 0=not started 1=In-queue 2=at m/c 3=finished from m/c
9	
10	Sub-priority 1
11	Start time at each machine
12	Processing time of each job at each machine
13	Waiting time of each job at each machine
14	Related Times ===V
15	Sarq, MachGoQ, MachInQ, MachOutQ, SextQ, EXT SarQall Go In Out(machine)
16	M A I N Q s
17	M I N P Qs
18	M A C H Qs
19	Start time at machine MACHQ
20	Finished time at machine MACHQ
21	M E X T Qs
22	Times
23	QUEUE 23 out put buffer of all machines to this m/c
24 now	Times
25 tot	Rout of each job (rout signed -1 means this process is finished)
26 mem	Corresponding Processing times
27	ENTRY time of each job to first machine
28	ROUTING 2
29	PROCESSING TIME 2
30	SUB_PRIORITY 2
31	
32	
33	
34	System Use
35	System Use
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	X-Z & Y= 1-JOBS
49	↓ X

Table 6-7 Core of the array: operational routing, times and queues

Finally, each value in this array could be called or changed at any time even while the simulation model is running. The array could involve more detailed information. In future the storing procedures could be changed slightly to avoid the huge size of memory. Therefore, a mechanism could be required to tidy up the finished jobs from the array to save them in a separate file till the whole process is completed.

6.8 SUMMARY:

Simulation technique is used to represent a job shop production system. The JSSM (Job shop Simulation Scheduling Model) is especially designed and developed to participate in investigating the performance of scheduling rules in a static and dynamic job shop. Five rules are tested. More rules are available within the model. The JSSM could be used to experience some scheduling problems such as queue building and tardiness problem. The logic within the model is discussed. Briefly, the problem is to represent a job shop production system which could be described as several machines through which a number of jobs may flow. A machine could be either processing or idle.

The input data is either a real or randomly generated one. In this study, data is generated by using the NAG library subroutines G05DYF and G05DZF. Jobs and inter-arrival rate are generated. Validation is made by printing out the results regularly then a direct comparison is made with expected ones.

Despite slicing simulation timing could be applied, a discrete-event simulation timing is used. It advances time according to the nearest next event whilst slicing simulation advances time according to a fixed interval of time (usually one unit of time).

The JSSM is built using Fortran77 language on Sun work-stations 3/50. The information is saved in a three dimensional array. Each slice of the array has a certain type of information. In general, the X-axis presents the information which is related directly to each job. Y-axis includes one job in each horizontal slice. Z-axis is concerned with machine's information.

The JSSM is used to investigate the effect of scheduling rules on the tardiness performance. The performance of scheduling rules, could be experienced. The EDD, FCFS, FRFS, SPT and FDSSD rules are examples of the scheduling rules which are built within the JSSM model.

In spite of the tardiness criterion is used as a measure of performance, long-term performance (customer satisfaction and system confidence are examples) could be beyond the JSSM limit. These long-term performance may not appear in the simulation run of the model.

CHAPTER 7

EXPERIMENTAL WORK

CHAPTER 7
EXPERIMENTAL WORK

7.1 EXPERIMENTAL DESIGN:

Several experiments are described. They aim to evaluate the performance of the proposed scheduling rules. The tardiness criterion is the measure of performance. The proposed scheduling rules are the EDD, FCFS, FRFS and SPT rules. These rules are compared with FDSSD rule.

The job shop consists of several machines (in this thesis they are four), and many jobs flow through these machines. These jobs are dispatched to the machines according to the rule selected. Later in this chapter, more explanation is provided. A simulation model of the shop (JSSM) was built and developed to simulate and test the proposed rules.

The main factors in the experiments are the shop receiving rate and the scheduling rules. Input data is generated at random using the NAG library subroutines. Six cases of receiving rate are employed with each rule. A number of replications is made to produce a set of results that could be statistically analysed.

7.1.1 JOB SHOP MODEL:

The simulation model represents a simple job shop system containing four non-identical machines (see Fig.14). The arrival order of jobs into the shop is random with inter-arrival times that are uniformly distributed. Orders are released into the shop at the time of receiving. Received jobs are assigned randomly from one to four operations. Each operation is randomly assigned a processing time from a distribution where mean value is eight time unit. The probabilities are equal for a job being routed to a particular machine, with a 50 percent chance of being visited. In other words, a total processing time of a job is randomly variable with mean of 16 unit of time [33,37]. Due dates are also generated at random from uniform distribution with mean of six times of total processing time of each job [33].

The JSSM is written in Fortran77, running on the University 'Sun' work-stations. The program contains subroutines for different tasks. The system is started with four jobs. Jobs are sent to each machine to keep machines busy until new random orders are received. The system is brought to steady state by monitoring the receiving rate and exit rate of the model during a "warm-up" period. The overall length of a simulation run is 5000 unit of time.

Several significant events are considered within the JSSM. These events are as follows:

1- Receiving: where new orders are accepted by the model.

2- Arrival: where orders are moved into shop.

3- Entry: when jobs are queued at machines.

4- Releasing: when jobs are released from machines.

5- Exit: when products are finished and they are going out of the shop.

Different seed numbers (for the random number generator) are used with the same treatment and level (rules and receiving rates respectively) to produce a range of results. The same seed numbers are used with the different rules, and then with the different receiving rates.

The receiving rate is the rate that jobs are received over time; 10%, 13.3%, 20%, 30%, 40% and 50%. These will result in varying machine utilisation and the load in the shop. The utilisation resulted from these different receiving rates are varied from 40% to 99.6%. Also, load is varied from under-loading (very short queues) to overloading (long queues). The number of replications is 20, 20, 29 and 41 replications for each rule at the following receiving rates; 50, 40, 30 and the others respectively. As mentioned previously, there are six receiving rates and five rules to be tested, i.e there are 30 cases. Therefore, 960 experiments are to be made.

7.1.2 MEASURE OF PERFORMANCE:

There are many measures of performance, which were discussed in chapter three (For more detail see also Table 3-1). The satisfaction or discontent of a customer activates the effort to consider the tardiness based criterion as a measure of performance in this study. Tardiness is the amount of time by which a job finishes after its due date. However, job tardiness results are emphasised because of their significance to production managers.

The simulation model is controlled for the following measures of performance: mean of total tardiness, total tardiness, earliness and percentage of tardy jobs and percentage of early jobs.

7.1.3 TESTED RULES:

The five selected rules for investigation are:

1- The Earliest Due Date first served rule (EDD)- this rule concerns directly with due dates.

2- The First Come First Served rule (FCFS) - for its simplicity and it is a form of fairness.

3- The First Received First Served (FRFS) - the most fair procedures among traditional rules.

4- The Shortest Processing Time first served (SPT) -for its superior performance in so many studies [37].

5- The Fair Delivery and Shop State Dependent (FDSSD)- to compare its performance with the other tested rules.

7.2 EXPERIMENTAL PROCEDURES:

The JSSM model is used to perform the experiment. The JSSM processes several different jobs, each one has a different order of operations (routine). Each job requires an operational time on each machine. All jobs have equal probabilities of being processed by any machine and the number of operations is a random variable between 1 and 4 inclusive. Four machines are described in the JSSM model, each machine can deal with one job at a time. The tardiness criterion is used for measuring the performance of each treatment.

As mentioned previously in section 7.1.3, that there are five rules to be examined under different receiving rates in the shop (see section 7.3.1). Thirty cases to be treated. A simulation run length is 5000 unit of time. The main number of experiments made are 960 experiments. Concerning light receiving rates; 10, 13.3 and 20, forty one replications are made. In receiving rate of 30, twenty nine replications are employed while in the remaining rates twenty replications are performed.

In general, the same set of input data is used with all five rules under the same level of receiving rate. In other words, under each level of receiving rate there are five treatments. A number of replications is made for each treatment. Each replication uses different set of input data. A different set of data means that a different seed number is used.

7.3 INPUT AND OUTPUT:

7.3.1 INPUT DATA:

There are two types of input data that the simulation model (JSSM) could use to perform the experiments: real and generated data. Since the real data is not available, a generated data is made. The NAG library subroutines are employed to generate at random the input data at the University 'Castle' mainframe (see section 6.7.3).

The generated data consists of two main parts. The first part represents jobs that they are going to be processed on machines. Each job requires three main variables to be generated, routing through machines, operational processing time on each machine, and the due date of each job. The G05DZF subroutine purpose is to produce for each job pseudo-random logical values (true and false) for each machine. True logical value means this job is going to visit that machine. All the values of routing are generate at random with all machines have equal chance to be visited. Also, each machine has equal chance of being busy or idle. Then, the G05DYF(m,n) subroutine is used to generate pseudo-random integer numbers. They are taken from uniform distribution over intervals (m, n) inclusive, where m is the minimum value in the interval and n is the maximum value. The mean operational processing time is eight unit of time.

The due date allowance varied uniformly from 2 to 10 times of the total processing time of a job. Therefore, each job has an average due date six times of the average of total processing time [33,37].

The second part of the generated data concerns shop load and inter-arrival time. It consists of two values, the period before the next receiving event will happen and the number of jobs to be received. The shop performance is largely affected by these two values [7]. These values determine how many jobs to be received (taken from the first generated part), and when the next receiving event is going to take place. As a result of those values, shop receiving rate is determined. Six shop receiving rates are proposed, 10, 13.3, 20, 30, 40 and 50 percent. For example, a receiving rate of 20 percent means the average of receiving is three jobs every 15 unit of time. These rates generated a general average utilisation equal to 41.8, 55.3, 81.6, 98.1, 99.5 and 99.6 percent.

7.3.2 OUTPUT RESULTS:

In previous section 7.3.1 the answer of the following question "what is the nature of the input data?" is reported. Section 7.2 discusses how the experiments have been performed. In this part the output results of these experiments are presented in Tables 7-1, 7-2 and 7-3. Three primary performance measures are considered: average job tardiness, total tardiness and the percentage of tardy jobs. The average time that a job spent in the system could be also measured.

All the output results of each treatment are used to calculate the average value of tardiness criteria. Calculation and analysis are carried out on the 'Castle' mainframe using the Minitab package.

RULES	Total tardiness of received jobs (Total tardiness of completed jobs)					
	Receiving rate% (average utilisation in the shop%)					
	10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
EDD	17.61 (17.61)	37.59 (37.59)	3892 (3839)	723323 (464023)	1826537 (985156)	1055625 (430989)
FCFS	63.37 (63.37)	298.5 (298.5)	10017 (9852)	682506 (225324)	1752325 (695912)	1035967 (337086)
FDSSD	11.66 (11.66)	48.02 (47.83)	5423 (5354)	673842 (478270)	1747093 (993603)	1038880 (459381)
FRFS	82.37 (82.37)	419.5 (419.5)	11304 (11181)	709898 (466415)	1814073 (995939)	1060603 (453427)
SPT	21.37 (21.37)	98.5 (97.8)	10026 (9472)	550702 (143807)	1464160 (406889)	885971 (213585)

* At 3000 unit of time.

Table 7-1 Total tardiness of jobs

Percentage of tardy jobs in the shop (%)						
RULES	Receiving rate% (average utilisation in the shop%)					
	10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
EDD	0.812	1.18	13.2	87.8	94.86	94.4
FCFS	1.517	3.77	24.73	86.2	93.99	93.9
FDSSD	0.572	1.02	12.71	87.4	93.71	93.3
FRFS	1.803	4.56	27.7	88.9	95.04	94.4
SPT	0.831	1.52	11.31	50.1	68.97	77.6

*** At 3000 unit of time.**

Table 7-2 Percentage of tardy jobs in The shop

Average time spent in the system						
RULES	Receiving rate% (average utilisation in the shop%)					
	10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
EDD	22.83	28.38	59.60	489	921.5	700.2
FCFS	23.23	29.42	64.66	277.34	661.9	548.9
FDSSD	22.9	28.43	59.06	490.2	901.2	722.4
FRFS	23.21	29.43	65.76	483.8	912.6	749.7
SPT	22.81	28.3	57.02	180.06	374.6	320.4

*** At 3000 unit of time.**

Table 7-3 Average time spent by a job in the system

7.4 ANALYSIS OF THE RESULTS:

Hypothesis tests were conducted for the differences among mean responses of various rules. A series of two sample comparison t-tests were conducted to classify and grade the selected scheduling rules using various performance measures. This analysis is carried out on the 'Castle' mainframe using the Minitab package. The desired total significant level of 95 percent was selected. Some results may be significant at the level of 99 percent. The F-test is conducted to find out if there is a significant effect of different rules on the tardiness criteria. Then the t-test is conducted to compare between the performance of each rule and the developed one (the FDSSD rule). These results are summarised in Tables 7-4, 7-5 and 7-6. These results are listed in the following tables in order of their performance.

Total tardiness:

Under this measure, two types of calculation can be obtained, total tardiness of received jobs and total tardiness of completed jobs. In general, the SPT rule was clearly dominant when the shop is heavily loaded, while the EDD rule performed well when the shop is moderately loaded. When the load in the shop is light, the FDSSD rule dominates all other tested rules with significant level of 95%. The FRFS rule (the most fair rule among the sched-

Receiving Rate% (Average Utilisation in the shop%)					
10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
FDSSD EDD SPT FCFS FRFS	EDD FDSSD SPT FCFS FRFS	EDD FDSSD FCFS SPT FRFS	SPT FDSSD FCFS FRFS EDD	SPT FDSSD FCFS FRFS EDD	SPT FCFS FDSSD EDD FRFS

* At 3000 unit of time.

Table 7-4 Performance of rules according to total tardiness of received jobs

Receiving Rate% (Average Utilisation in the shop%)					
10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
FDSSD EDD SPT FCFS FRFS	FDSSD EDD SPT FCFS FRFS	SPT FDSSD EDD FCFS FRFS	SPT FCFS FDSSD EDD FRFS	SPT FDSSD FCFS EDD FRFS	SPT FDSSD FCFS EDD FRFS

* At 3000 unit of time.

Table 7-5 Performance of rules according to the percentage of tardy jobs

Receiving Rate% (Average Utilisation in the shop%)					
10 (41.8)	13.3 (55.3)	20 (81.6)	30 (98.1)	40 (99.5)	50* (99.6)
SPT EDD FDSSD FRFS FCFS	SPT EDD FDSSD FCFS FRFS	SPT FDSSD EDD FCFS FRFS	SPT FCFS FRFS EDD FDSSD	SPT FCFS FDSSD FRFS EDD	SPT FCFS EDD FRFS FDSSD

* At 3000 unit of time.

Table 7-6 Performance of rules according to the average time in the system uling rules) exhibits a less performance than the FDSSD rule on all receiving rates, especially on low and moderate shop levels. Figure Fig.27-(a) represents the position of the FDSSD rule among the other rules according to the total tardiness of received jobs.

In general, it has been shown that the FDSSD rule performs significantly better than all others on the first receiving rate (low loading). On the next receiving rate the FDSSD rule performs also as well as the EDD rule. Also, Figure Fig.27-(b) exhibits almost similar conclusion according to the total tardiness of completed jobs.

Percentage of tardy jobs:

The FDSSD rule again clearly dominated all other rules under most of the receiving rates. Despite the fact that the SPT performs significantly better on heavy shop load, the FDSSD rule performs as well as other rules such as the EDD rule (see Fig.27-(c)).

7.5 SUMMARY:

Experiments are performed on the simulation model (JSSM) to investigate the performance of five selected scheduling rules under six receiving rates. The JSSM represents a job shop production system with four machines and several buffers to keep arrived jobs waiting in queues till a machine

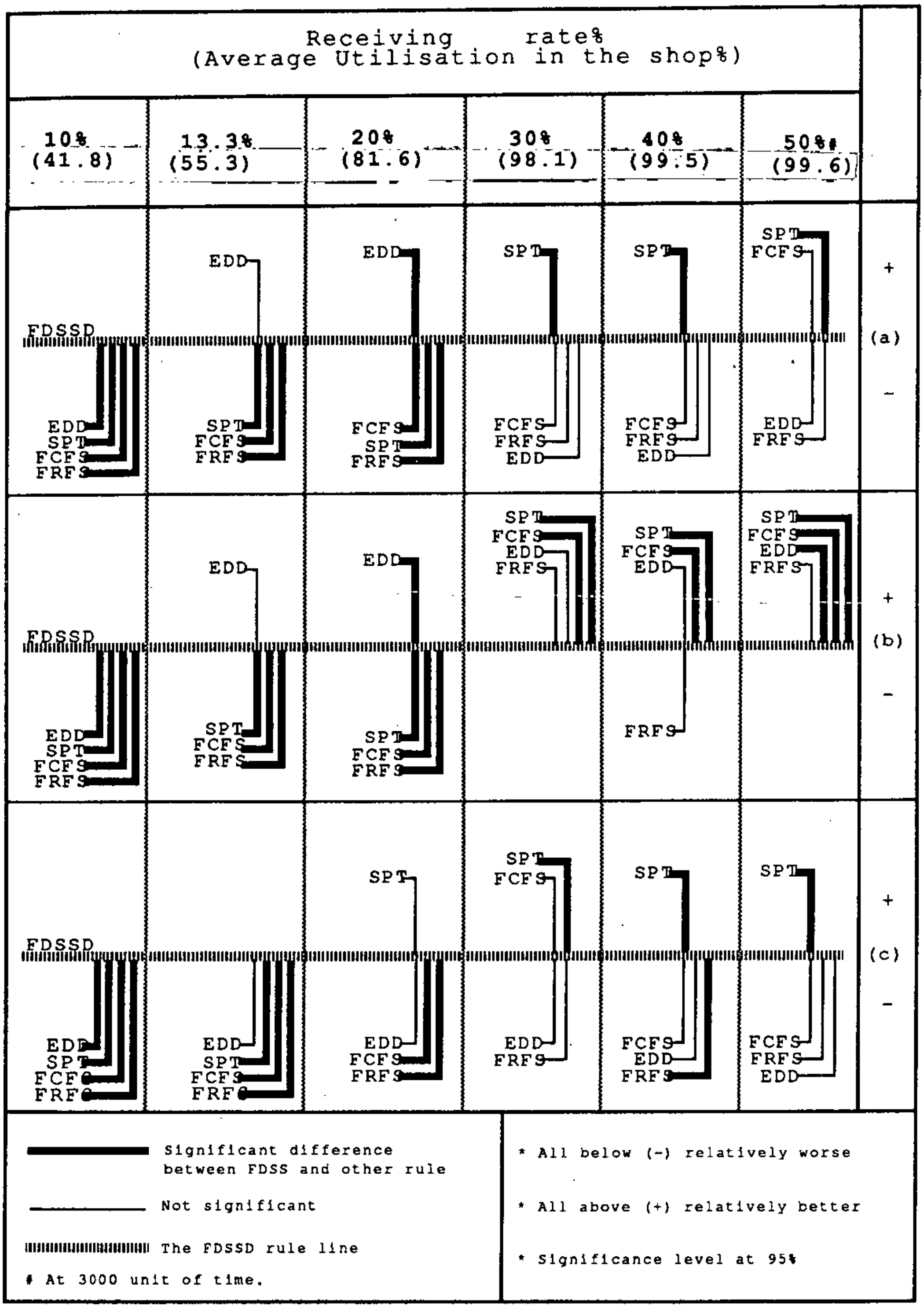


Fig.27 The FDSSD rule performance relative to other rules according to: (a) Total Tardiness of received jobs. (b) Total Tardiness of completed jobs. (c) Percentage of tardy jobs.

become idle. The five tested scheduling rules are the EDD, FCFS, FDSSD, FRFS and SPT rules. The six receiving rates are 10, 13.3, 20, 30, 40 and 50 percent.

Random input data is generated using NAG library subroutines. The output results are tested using the Minitab package to compare the performance of the FDSSD rule with other rules' performance. The total tardiness and percentage of tardy jobs are used as the measures of performance.

The results indicate that, generally, the performance of the FDSSD rule is significantly better than other tested rules on the lighter shop loads. Furthermore, the performance of the FDSSD rule is second best in the moderate shop loading (receiving rate at 20 percent). On heavy shop loads, the SPT rule performed significantly better than the FDSSD rule on total tardiness of received jobs. However, the FDSSD rule did perform as well as most other tested rules (see Fig.27 (a), (b) and (c)).

The FDSSD rule performed as well as most examined rules on all receiving rates. Furthermore, it performs significantly better than all others on light receiving rates.

CHAPTER 8

DISCUSSION - FURTHER EXTENSIONS AND CONCLUSION

CHAPTER 8

DISCUSSION - FURTHER EXTENSIONS AND CONCLUSION

8.1 DISCUSSION:

The scheduling problems addressed in this thesis have been repeatedly studied to find better procedures by developing or improving scheduling rules. These problems could be formulated as several jobs to be processed on some machines. Queues may be built up, thus increasing WIP. Machines may also become idle resulting in lower utilisation in the shop. The main objective of this thesis is to introduce and highlight the fairness consideration into scheduling rule. This was carried out by introducing and developing the FDSSD rule. The environment of the system is the job shop environment. The tardiness criterion has been selected because, the concern is with the satisfaction of customers who expect their orders to be delivered on time.

Five rules have been tested, the EDD, FCFS, FDSSD, FRFS and SPT rules. Then, the performance of the FDSSD rule -the developed one- has been compared with the other rules' performance. The

FDSSD rule takes into accounts the order in which jobs have been received with their delivery dates.

The JSSM has been constructed in this work to investigate and compare the performance of the rules mentioned above under different receiving rates. Input data is generated at random for use in the experiments.

Although the FRFS rule is commonly used in practice [50] and it might be considered as the most fair scheduling rule, the performance of the FDSSD rule is significantly better at low receiving rates. On heavy and moderate receiving rates the FDSSD rule performs as well as the FRFS and most of other rules. Besides that, the FDSSD rule has another advantage that does not appear in the simulation run. This advantage concerns the achievement of customer satisfaction. This can be attributed to the way that the FDSSD rule attempts to employ some ethical principles such as fairness principle within delivery procedures.

8.2 FURTHER EXTENSIONS:

This work contributes to the scheduling area three main aspects. Firstly, a scheduling rule is developed. Secondly, a simulation model is established. Finally, the ethical view is introduced and highlighted for further studies to be considered in a more formal and academic way in scheduling procedures. Further work is discussed in more detail in the following sections.

8.2.1 SCHEDULING PROCEDURES:

Using ethics in scheduling area may generate some understanding and harmony in a shop. This may raise morale. Thus, it may result in a better performance in the shop, especially when it is congested. Furthermore, better results could be achieved when the scheduling rule, that has been used to build the schedule, incorporates the common-sense and some of the ethical principles such as the fairness principle.

It is recommended that some form of ethics should be injected into scheduling rules. This may deserve further consideration and it may have great practical implications. The SPT rule has very significant results on total tardiness and mean tardiness. It leaves some longer jobs very late in the shop. It also ignores due dates of jobs. Therefore, a new form of procedure could be reached to compromise between the SPT rule and fair delivery procedures in the light of the FDSSD rule.

8.2.2 THE SIMULATION MODEL (JSSM):

In this work, job shop scheduling problems were studied. Other types of production system could be studied using the JSSM, especially the flow shop. The assembly system could also be investigated. In the case of the assembly line, the main difference from other two systems, job shop and flow shop, is how to store the information. Tables 8-1, 8-2 and Fig.28 may represent a general outlook to the way that information is stored. Whenever there is an assembly process, then one of these jobs is chosen to be the main one. At the main job, the assembly operational processing time is inserted. The operational processing time of the other assembled job(s) is replaced by a negative value. This value indicates the number of the main job (where the assembly is going to take place).

As mentioned previously, JSSM can deal with other scheduling problems such as a machine maintenance or breakdown, parallel machines to be balanced and machines' interference to be decreased.

The JSSM model requires a visual improvement to be used in scheduling learning purpose. It could also be used to experience the influence of different rules on several measures of per-

Part (Job) Number	Operational processing times			
	M1	M2	M3	M4
1	20	0	5	10
2	5	0	15	- 1
3	6	2	- 1	0
4	0	- 3	0	0

Table 8-1 Example 8.1: Four jobs to form product A.

Part Number	Number of parts to be assembled (use the MAIN job)	Next part of the same product at System Code	Machine Routing				Operational Processing Time				Product
1	4	2	1	3	4		20	5	10		A
2		3	1	3	4		5	15	-1		
3		4	1	2	3		6	2	-1		
4			2				-3				
5	1										B
6	1										C
7	2	8	⋮				⋮				D
8											

Table 8-2 Input structure for assembly process

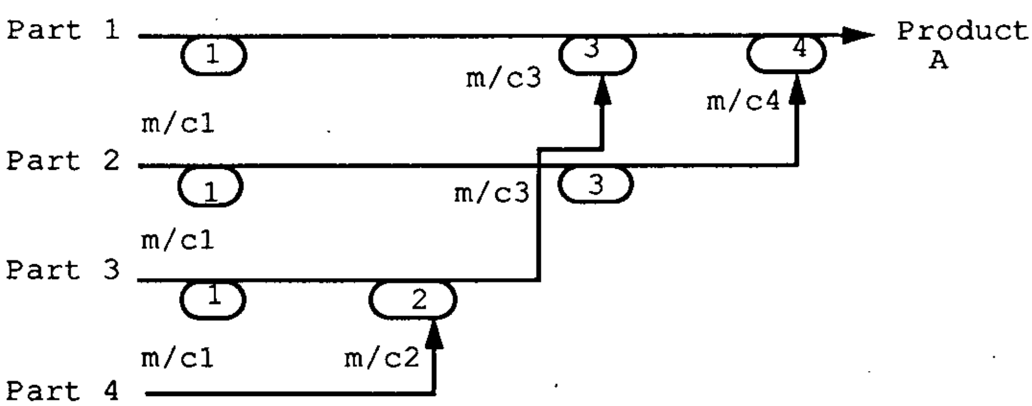


Fig.28 Example 8.1: Product A.

formance. Manual (external) switching between available scheduling rules is possible. It could be extended to benefit of each available rule by using a trigger value within the system. This value may be used to switch between rules internally.

More attention is required to study the subcontracting procedures to release the pressure from a congested system. An ethical base relation could be used to communicate with a group of firms to exchange their idle times in a cooperative way. An ethical base could mean being fair to customers, to other firms in the group for keeping the delivery dates.

This work could be extended to investigate non-delay schedules where machines could be kept idle (waiting for an expected coming job) while there are jobs waiting before that machine. Thus, an extra event should be considered within JSSM. This event related to machine loading.

Finally, there is very little work that deals with ethics in the scheduling area. Therefore, it may be worthy of more attention. It is recommended to widen the area of interaction among the other schedules in the system on a fairness basis. A better understanding and positive results may be expected.

8.3 CONCLUSION:

The scheduling problem as submitted for investigation was to schedule many jobs through several machines (four machines in this work). Each job has a different routine through the machines. The main aims are:

- To meet delivery dates,
- To minimise WIP in the shop, and
- To reduce machine idleness.

Owing to the above aims, investigations were formulated to determine that scheduling rule of tested ones is the most suitable to satisfy the requirements of both the customers and the shop. The tested rules are the EDD, FCFS, FDSSD, FRFS and SPT rules. Two main measures of performance are used the total tardiness and percentage of tardy jobs.

The main objectives of the study were achieved. The first objective is to develop a scheduling rule (FDSSD) that considers customer satisfaction besides its acceptable performance. The second one is to develop a simulation model (JSSM) to be employed in running some experiments to compare the performance of the tested rules. The third objective is to find out how the FDSSD rule performs relative to the other tested rules.

What is clear in this case, however, is that a reasonable compromise has been reached between improving the customer satisfaction and improving operational efficiency.

The FDSSD rule introduces the moral element into logical scheduling. Owing to this, the effect of complex scheduling procedures that could adversely affect customers can be diminished. The unnecessary unfairness may not be apparent to management until too much work has been done.

Most previous studies are concerned with obtaining better procedures to increase machine utilisation, to achieve timely delivery, to decrease WIP and/or to lower the production cost. However, many of these studies ignore the moral attitude towards the customer. The FDSSD rule gives the customer who came first a higher priority than those who come afterwards. The later customers may be served first if there is no danger of an earlier order becoming late.

A simulation model is developed to perform several experiments. In these experiments, five rules are tested under six receiving rates, 10, 13.3, 20, 30, 40 and 50 percent. The aim of these experiments is to compare the performance of the FDSSD rule with other rules.

The experimental observations show that scheduling rules that rely on simple common sense could achieve significant improvements [80]. It is believed that injecting the ethics into scheduling rules deserves further consideration and is of great practical significance.

The results demonstrate that the FDSSD rule performs as well as other tested rules, especially under low receiving rates where it dominated all other examined rules. Furthermore, the FDSSD rule compromises between the scheduling operation performance and customer satisfaction. Besides that, some of the FDSSD rule's advantages may appear over a long run. This can be attributed to the way that the FDSSD rule attempts to employ some ethics in its procedures such as fair delivery principle.

REFERENCES

REFERENCES:

[1]. Worrall, B., Bancroft, G. and M. Sivanesan, "Job shop scheduling rules incorporating customer importance and rush status." Proceeding of AIIE (SPRING Annual Conference), San Francisco, May (1979).

[2]. Gere, W. Jr., "Heuristics in job shop scheduling," Management Science, (1966) (13) (3) (167-190).

[3]. "Machine-tool technology," American Machinist, vol. 124, no. 10, pp. 105-128, 1980.

[4]. "Material flow in the factory", Industrial Computing, pp. 55-58, October, 1988.

[5]. Hon, K. B.,"Stabilised sequence planning system," in Proceedings of International Conference on Computer-Aided Production Engineering, editor. J. A. McGeough, p. 175-177, April, Edinburgh (1986).

[6]. Stommel, H., Betriebliche Terminplanning, De Gunter, (1976). (In German)

[7]. Dar-El, E. and Wysk, "Job shop scheduling - systematic approach," Journal of Manufacturing Systems, (1982) (v.1) (part.1) (77-88)

[8]. Neelamkavi, J., D. Rao and V. Thomson, "Use of dispatching Rules in Manufacturing Applications," 5th Canadian CAD/CAM and Robotics Conference, Toronto, Canada (June 1986).

[9]. Fox, M., R. Allen, S. Smith and A. Strohm, ISIS: A constraint-Directed Reasoning Approach to a job shop, Carnegie-Mellon University, CMU-RI-TR-83-8, 1983.

REFERENCES

[10]. Ham, I,K. Hitomi and T. Yoshida, Group of technology: Applications to Production Management, Kluwer-Nijhoff Publishing, USA (1985).

[11]. Conway R.W., "Priority Dispatching and Work-In-Process Inventory in a job shop," Journal of Industrial Engineering, (1965)(16)(March)(123-130)

[12]. Gilder, L.and P. Kleindorfer, "Coordinating Aggregate and detailed scheduling in the one machine job shop: computation and structure," Operation Research, (1975)(March)(23)(2)(312-324).

[13]. Abou-Zeid, M., El-Bouri, A. and U. Garg, "Computer solution to Single Machine Sequencing Problem," Industrial Engineering, (1985)(17)(8)(26-29).

[14]. Selen, W. and D. Hott, "A Mixed-Integer Goal Programming Formulation of the Standard Flow Shop Scheduling Problem," Journal of Operational Research Society, (1986)(37)(12)(1121-1128)

[15]. Trevor M. & I. Christie, Manufactures & Suppliers In Britain & Japan: competitiveness & the growth of small firms, Policy studies Institute. London (1988).

[16]. Morishima, M., Why has Japan 'succeeded'? Western technology and Japanese ethos, Cambridge University Press (1982).

[17]. Mente, B., Japanese Etiquette & Ethics In Business, 5th edition. Phoenix publishing associates ltd., Herts, UK (1987).

[18]. Hon, K. B.,"Sequence Planning System for Dynamic Manufacturing Environment," Proceedings of the third International Conference Manufacturing Engineering, p. 123-125, 4-6 August, Newcastle (1986).

REFERENCES

[19]. Cowhig, D. F., Explanation by culture or by economic rationality: The Japanese labour relations system, M.A Thesis Pennsylvania University, USA (1985).

[20]. SAC Company- 86 Via Campari, Sesto S. Giovanni, Milan, Italy (1984-1985).

[21]. Blackstone J., D. Phillips and R. Heisterberg, "A state of the art survey on Dispatching Rules For manufacturing Job shop operations," NSF/RANN, Report no. GEMS-4-77, Texas A&M University (1977).

[22]. Montazeri, M. and L. Van Wassenhove, "Analysis of scheduling Rules for FMS," International journal of production research, (1990)(V.28)(N.4)(785-802).

[23]. Panwalkar, S. and W. Iskander, "A Survey of Scheduling Rules," Operations Research, (1977)(25)(1)(January-February)(45-61).

[24]. Emery, J. C., "An Approach to Job Shop Scheduling Using a Larg-Scale Computer," Industrial Management Review, (1961)(no.3)(Fall)(78-96).

[25]. Muth, J. and G. Thompson (editors), Industrial Scheduling, Printice-Hall, Inc, Englewood Cliffs, New Jersey (1963).

[26]. Dudek, R. A. and O. Teuton, Jr., "Development of M-stage Decision Rule For Scheduling n Jobs through m Machines," Operations Research, (1964)(12)(3)(471-497).

[27]. Fox, M., Constraint-Directed Search: A case study of job shop scheduling, Carnegie-Mellon University, CMU-RI-TR-83-22, Dec. 1983.

REFERENCES

[28]. Stecke, K. and J. Solberg, "Loading and control policies for a flexible manufacturing system," International Journal of Production Research, (1981)(19)(5)(481-490).

[29]. Chang, Y. and R. Sullivan, "Schedule generation in a dynamic job shop," International Journal of Production Research, (1990)(28)(1)(65-74).

[30]. Charlton, J. and C. Death, "A method for general machine-scheduling problems," Operations Research, (1970)(18)(689-707).

[31]. Giffler, B. and G. Thompson, "Algorithms for solving production-scheduling problems," Operation Research, (1960)(8)(487-563).

[32]. Moore, J.and R. C. Wilson, "A Review of Simulation Research in Job Shop Scheduling," Production and Inventory Management, (1967)(vol.8)(January)(1-10).

[33]. Kiran, A. and M. Smith, "Simulation Studies in Job Shop Scheduling - A Survey," Computer and Industrial Engineering, (1984)(8)(2)(87-93).

[34]. Rowe, A., "Toward a Theory of Scheduling," Journal of Industrial Engineering, (1960)(XI)(2)(125-136).

[35]. Russell, R., E. Dar-El and B. Taylor, "A comparative analysis of the CoverT job sequencing rule using various shop performance measures," International Journal of Production Research, (1987)(25)(10(1523-1540).

[36]. Schultz, C., "An expediting heuristic for the shortest processing time dispatching rule," International Journal of Production Research, (1989)(27)(1)(31-41).

REFERENCES

[37]. Conway, R., W. Maxwell and L. Miller, Theory Of Scheduling, Addison-Wesley Publishing Company, Reading Massachusetts (1967).

[38]. Cantellow, D., R. Pitt, R. Saw and J. Hough, "Machine-Shop Problems: An Operational Research Approach," Operational Research Quarterly, (1973)(24)(4)(503-525).

[39]. Day, E. and M. Hottenstein, "Review of sequencing research," Naval Research logistics Quarterly, (1970)(17)(11-39).

[40]. Schroeder, R. G., Operations Management: Decision making in the operations function, 3rd Edition, McGRAW-Hill Book Company (1989).

[41]. Baker, K., "Sequencing rules and due date assignments in a job shop," Management Science, (1984)(30)(9)(1093-1104).

[42]. Jackson, J., "Some Problems in Queueing with dynamic priorities," Naval Research.logistics Quarterly, (1959)(7)(235-250).

[43]. Wacker, J. and D. Lucht, "Effective Dispatch Rules To Improve Performance in Job Shops," Production & inventory Management, (1985)(26)(77-87)

[44]. Graves, S., "A Review of production scheduling," Operations Research, (1981)(29)(4)(July-August)(646-574).

[45]. Montazeri, M. and Van Wassenhove, L., Scheduling rules for manufacturing systems, Working paper # 86-01, Afdeling Industrial Beleid, K. U. Leu. (1986).

[46]. Blackstone, J., D. Phillips and R. Heisterberg., State-Of-The-Art Survey on dispatching rules for manufacturing job shop operations, NSF-RA-770463, Texas A&M University Collage Station, Department of Industrial Engineering, (1977).

REFERENCES

[47]. Holloway, C. and R. Nelson, "Job shop scheduling with due date and variable processing time," Management Science, (1974)(20)(9)(1264-1275).

[48]. Buckley, J., A. Chan, U. Graefe, J. Neelamkavil, M. Serrer and V. Thomson, "A Production Planning and Scheduling system for manufacturing plants", 2nd International Conference - Computer Aided Production Engineering, Edinburgh, UK (April 1987).

[49]. Conway, R., An experimental investigation of priority assignment in a job shop, Rand Corporation Research Memorandum, RM-3789-PR (February 1964).

[50]. Eilon, S., I. Ghowdhury and S. Serghiou, "Experiments with the SI rule in job shop scheduling," Simulation, (1975)(21)(February)(45-48).

[51]. Stoeva, S., "A due date-based dispatching rule for flexible manufacturing systems," International Journal of Production Research, (1990)(28)(11)(1991-1999).

[52]. Vepsalainen, A. State Dependent Priority Rules For Scheduling, Carnegie-Mellon University, CMU-RI-TR-84-19 (1984).

[53]. Hastings, N and C. Yeh, "Job Oriented Production Scheduling", European Journal of Operational Research, (1990)(47)(35-48).

[54]. Hollier, R., "A Simulation Study of Sequencing in Batch Production," Journal of Industrial Engineering, (1968)(19)(4)(389-407).

[55]. Emery, J., "Job Shop Scheduling by Means of Simulation and an Optimum-Seeking Search", Proceeding of 3rd conference: Application of Simulation, ACM/AIIE/SHARE/SCI/TIMES, Los Angeles, (363-372)(Dec 1969)

REFERENCES

[56]. Jackson, J.,"Simulation research on job shop production," Naval Research and Logistic Quarterly, (1957)(4)(287-295).

[57]. Elion, S., "Research note: The job shop simulation program," International Journal of Production Research, (1973)(11)(3)(299-300).

[58]. Ashour, S. and S. Vaswani, "A GASP simulation of job shop scheduling," Simulation, (1972)(18)(1)(1-10).

[59]. "Eleven steps to simulating an automated system," Modern Material Handling (1984)(November)(5)(45-47).

[60]. Wight, O., Production and inventory management in the computer age, (1984).

[61]. "Management for Engineers: 7. The Japanese Example," International Journal of Operations and Management, (1990)(10)(6)(37-41).

[62]. Oliver, N., "Human Factors in the implementation of Just-In-Time Production," International Journal of Operations and Management, (1990)(10)(4)(32-39).

[63]. Browne, J. and B. Davis, "Scheduling in a batch or job shop production environment," Engineering Management international, (1982)(1)(173-184).

[64]. Kan, A. H. G. Rinnooy, Machine Scheduling Problems: Classification, complexity and computations, Martinus Nijhoff, Amsterdam, Holland (1976).

[65]. Baker, K., Introduction to sequencing and scheduling, John Wiely and sons (1974).

[66]. Chase, R. and Aquilano, N. Production and operations Management: a cycle approach, Home wood, Ill. Richard D. Irwin, (1985).

REFERENCES

[67]. Buffa, E., Modern Production/Operations Management, 7th edition, John Wiley and sons, (1983).

[68]. Bobrowski, PM and PS Park, "Work Release Strategies in a Dual Resource Constrained Job Shop," OMEGA International Journal of Management Science, (1989)(Vol.17)(No.2)(177-188).

[69]. Oliver Wight, "Input/output control a real handle on lead time," Production & inventory Management, (1970)(11)(3)(9-31).

[70]. Van Nunen, J. and M. Puterman, "Computing optimal control limits for G1/M/S Queuing systems with controlled arrivals," Management Science, (1983)(29)(6)(725-734).

[71]. Van Ommeren, J., "Input approximation for the batch arrival Mx/G/1 Queue," Operation Research, (1990)(38)(4)(678-685).

[72]. Ragatz, G., "An evaluation of order release mechanisms for job shops," Indiana University, PhD. (1985).

[73]. Baker, K., "The effect of input control on the performance of simple scheduling model," Journal of Operations Management, (1984)(4)(2)(99-112).

[74]. Nicholson, T., "Measuring customer Service and managing delivery," International handbook of production and operation management, (editor) Wild, R., Cassell Educational ltd, London (1989).

[75]. Caves, R. and M. Uekusa, Industrial organization in Japan, The Brooking Institution, DC., USA (1976).

[76]. Dunning, J., Japanese Participation in British Industry, Croom Helm ltd, Kent (1986).

REFERENCES

[77]. Edosomwan J. A "The Baldrige Award: Focus on Total Customer Satisfaction". Industrial Engineering (1991)(July)(24-26).

[78]. Betrand, J, "The use of work load information to control job lateness in controlled an uncontrolled release production system," Journal of Operations Management, (1983)(3)(2)(79-92).

[79]. Koulamas, C. and M. Smith, "Look-ahead scheduling for minimizing machine interference," International Journal of Production Research, (1988)(26)(9)(1523-1533).

[80]. Chandra, J. and Talavage, J. "Intelligent dispatching for flexible manufacturing.", International Journal of Production Research, (1991)(29)(11)(2259-2278).

Further references:

[81]. Baker, K. and G. Scudder, "Sequencing with earliness and tardiness penalties: A review," Operation Research, (1990)(38)(Jan-Feb)(22-36).

[82]. Conway, W. and W. Maxwell, "Network Dispatching by the shortest operation discipline," Operation Research, (1962)(10)(51-73).

[83]. Elvers, D., "The Sensitivity of the relative effectiveness of job shop dispatching rules with respect to various arrival distributions," AIIE Transactions, (1974)(6)(41-49).

[84]. Neville, A. and Kennedy, J., Basic statistical Methods for engineers and Scientists, International Textbook Co. (1964).

APPENDIX

1. THE PROGRAMME OF THE JOB SHOP SCHEDULING SIMULATION MODEL (JSSM)

```
-------------------------------------------
 SUBROUTINE UPDOWN(JJ)
C-------------------------------------------
CBROUT UPDOWN JOBS. (NEWMNQ, THEORY AND
LDMC16.)
 INCLUDE'.F.FILES/COM'
 DO 1 M=1,MACHS

IF(JJ.EQ.16.OR.JJ.EQ.23.OR.(M.EQ.1.AND.JJ.EQ.
15))THEN
 NUMBER=INFO(0,JJ,M)
 K2=NUMBER/2
 ENDIF
 IF(JJ.EQ.17)THEN
 NUMBER=4
 K2=2
 ENDIF
 DO 1 L=1,K2
 KK=INFO(L,JJ,M)
 INFO(L,JJ,M)=INFO(NUMBER-L+1,JJ,M)
 INFO(NUMBER-L+1,JJ,M)=KK
 1 CONTINUE
 RETURN
 END
C-------------------------------------------
SUBROUTINE JOBINF
C-------------------------------------------
C BROUT:FROM DATA.
 CHARACTER*20 STATUS(8)
 CHARACTER*50 WHERE(5)
 INCLUDE '.F.FILES/COM'
 INCLUDE'.F.FILES/FRM'
 1 WRITE(6,1350)
 1350 FORMAT('1)ALL JOBS 2)A JOB 3)FINISHED J
4)SUBCNTRCT J'
 @,/,'5)CANCELLED J 6)DELAYED J 7)STOPPED J
8)TARDY JOBS',/,
 @' TYPE 1-8 OR RTRN TO EXIT')
 READ(5,108)K1
 108 FORMAT(I4)
 IF(K1.EQ.0)GOTO111
 IF(K1.EQ.1)THEN
 31 WRITE(6,3006)
 3055
FORMAT('|',I1,'|',I3,'|',I4,'|',I5,'|',3(I1,'
|'),
 @3(I2,'|'),I3,'|',9(I4,'|'),I2,'|')
 DO 3 I3=1,JOBS
 WRITE(6,3055)(INFO(I3,IX,0),IX=0,19)
 3 CONTINUE
 ELSEIF(K1.EQ.2)THEN
 WRITE(6,*)'GIVE ME THE JOB NUMBER PLEASE.'
 READ(5,*)JN
 WRITE(6,3006)
 WRITE(6,3055)(INFO(JN,IX,0),IX=0,19)
 WRITE(6,*)'ROUTE','PROC.T','WAIT.T',
 @'ARRV.T','STRT.T','FINI.T'
 WHERE(1)='NOT START HERE.'
 WHERE(2)='IN QUEUE OF M/C'
 WHERE(3)='ON THE M/C'
 WHERE(4)='FINISHED FROM M/C'
 DO 12 I12=1,MACHS

WRITE(6,222)(INFO(JN,IX,I12),IX=25,26),INFO(J
N,13,I12),
@INFO(JN,21,I12),INFO(JN,16,I12),INFO(JN,19,I
12),
 @WHERE((1+INFO(JN,8,I12)))
 222 FORMAT(1X,I4,5(3X,I5),1X,A50)
 12 CONTINUE
 ELSEIF(K1.GE.3.OR.K1.LE.8)THEN
 STATUS(3)='FINISHED '
 STATUS(4)='SUBCONTRACTED'
 STATUS(5)='CANCELLED'
 STATUS(6)='DELAYED BY PROG'
 STATUS(7)='STOPPED BY PROG'
 STATUS(8)='LATE ABOUT'
 DO 7 I7=1,JOBS

IF(INFO(I7,0,0).EQ.K1.AND.K1.GE.3.AND.K1.LE.5
)THEN

WRITE(6,109)INFO(I7,2,0),STATUS(K1),INFO(I7,1
6,0)
 109 FORMAT('JOB NUMBER',I4,A20,I5)
 ENDIF
 IF(INFO(I7,5,0).EQ.1.AND.K1.EQ.6)THEN
 WRITE(6,109)INFO(I7,2,0),STATUS(K1)
 ENDIF
 IF(INFO(I7,5,0).EQ.2.AND.K1.EQ.7)THEN
 WRITE(6,109)INFO(I7,2,0),STATUS(K1)
 ENDIF

IF(INFO(I7,11,0).GT.INFO(I7,3,0).AND.K1.EQ.8)
THEN
 WRITE(6,109)INFO(I7,2,0),STATUS(K1)
 ENDIF
 7 CONTINUE
 ENDIF
 IF(K1.NE.0)GOTO1
 111 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE APPEAR(ITIME,ITTIME,NEXTM)
C-------------------------------------------
--------------------------
C BROUT: FROM MENU.
 WRITE(6,109)ITIME
 109 FORMAT('THE CURRENT INTERVAL TIME
IS',I5,'MIN Which IS THE '
 @'TIME CONTROL THE APPEARANCE OF MAIN MENU',/
,
 @' TYPE MMMMM MIN. <DEFAULT 0> RTRN TO EXIT')
 READ(5,1)ITIME
 1 FORMAT(I5)
 NEXTM=ITTIME+ITIME
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE KARKTR(CHARAC,NUMBER)
C-------------------------------------------
--------------------------
C BROUT:TO CHANGE CHARACTER TO NUMBER
C BROUT FROM (MENU.CHANGE.MACHNG & JBCHNG) AND
MENU
 CHARACTER*3 CHARAC
 OPEN(211,FILE='.F.FILES/CHARAC')
 WRITE(211,211)CHARAC
 REWIND(211)
 READ(211,212,ERR=111)NUMBER
 211 FORMAT(A3)
 212 FORMAT(I3)
 GOTO112
 111 WRITE(6,1)
 1 FORMAT('DONT USE CHARACTER. USE NUMBERS
ONLY')
 IGO=1
 112 CLOSE(211)
 RETURN
```

APPENDIX

```
 END
C--------------------------------------------
---------------------------
 SUBROUTINE MENU
C--------------------------------------------
---------------------------
C BROUT:THIS IS THE MAIN MENU OF THE PROGRAM.
C BROUT: FROM WIP-TWICE AND FROM MAIN PROGRAM
 CHARACTER*3 NEE
 INCLUDE'.F.FILES/COM'
 COMMON/NOT/MMM
 if(info(0,1,0).lt.info(0,46,0))goto113
 1 WRITE(6,99999)INFO(0,1,0)
99999 FORMAT(
 @'1 - SET ALL TERMS & ELEMENTS. TIME=',I6,/,
 @'2 - REPEAT LAST PROCESS........',/,
 @'3 - NEW EVENTS & CORRECTIONS...',/,
 @'4 - NEW JOBS ARE RECEIVED......',/,
 @'5 - SEE AVAILABLE JOBS.........',/,
 @'6 - NEW SCHEDULE ..............',/,
 @'7 - RETURN SCHEDULE TIME BACK..',/,
 @'8 - SET APPEARANCE TIME OF MENU',/,
 @'9 - CHANGE SCHEDULING TECHNIQUE',/,
 @'10- SAVE & QUIT................',/,
 @'11- QUIT.......................',/,
 @'12- SHOW QUEUES................',/,
 @'13- SHOW OR CHANGE ANY VALUES..',/,
 @' TYPE 1-13 RTRN TO EXIT')
CKKKKKKKKK Print*,'finish at????'
CKKKKKKKKK read*,info(0,46,0)
CKKKKKKKKKif(info(0,46,0).gt.0)info(2,0,3)=in
fo(0,46,0)
 if(info(0,46,0).GT.0)goto113

 READ(5,123)NEE
 123 FORMAT(A2)
 IF(NEE.EQ.'  ')GOTO111
 CALL KARKTR(NEE,NE)
 IF(NE.GT.14.OR.NE.LT.1)GOTO1
 111 continue
 goto 112
 113 if(Ne.eq.0)then
 NE=6
 else
 NE=12
 endif
 if(info(0,1,0).ge.8000)NE=11
 112 IF(NE.EQ.5)CALL DATA
 IF(NE.EQ.6.AND.MMM.EQ.0)CALL WIP
CCC
IF(NE.EQ.6.AND.MMM.NE.0)WRITE(6,*)'PROCESS IS
RUNNING'
 IF(NE.EQ.8)CALL
APPEAR(INFO(2,0,1),INFO(0,1,0),INFO(2,0,3))
 IF(NE.EQ.9)CALL TEKNIK
 IF(NE.EQ.10)STOP
 IF(NE.EQ.11)THEN
 LLL=111
 CALL FINAL(LLL)
 WRITE(6,19)
 19 FORMAT('THANK YOU ,GOOD BYE.')
 STOP
 ENDIF
 IF(NE.EQ.13)CALL VALUE
 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE VALUE
C--------------------------------------------
---------------------------
C BROUT:CHANGE ANY VALUE IN THE SYSTEM:FROM
MENU
 INCLUDE'.F.FILES/COM'
 III=IY
 11 jjj=0
 WRITE(6,1)
 1 FORMAT('- TYPE Y,X,Z VALUES IN THIS
ORDER.',/,
 @'- TO GET ONE SET IN ONE DIRECTION PUT IN ITS
PLACE 999.',/,
 @' EXAMPLE: THE COLUMN AT X=12 AND Z=0 THEN
TYPE => 999,12,0')
 IY=III
 READ(5,*,ERR=11)IY,IX,IZ
 IF(IY.EQ.999)THEN
 DO 2 I2=0,JOBS
 WRITE(6,3)I2,IX,IZ,INFO(I2,IX,IZ)
 3 FORMAT('(',I4,',',I4,',',I4,')=',I8)
 2 CONTINUE
 ENDIF
 IF(IX.EQ.999)THEN
 DO 4 I4=0,35
 WRITE(6,3)IY,I4,IZ,INFO(IY,I4,IZ)
 4 CONTINUE
 ENDIF
 IF(IZ.EQ.999)THEN
 DO 5 I5=0,MACHS*2
 WRITE(6,3)IY,IX,I5,INFO(IY,IX,I5)
 5 CONTINUE
 ENDIF

IF(IX.NE.999.AND.IZ.NE.999.AND.IY.NE.999)THEN
 WRITE(6,3)IY,IX,IZ,INFO(IY,IX,IZ)
 PRINT*,'-TYPE THE NEW VALUE'
 PRINT*,'-SAME VALUE REMAIN AS IT IS IF YOU
TYPE -999.'
 KKK=INFO(IY,IX,IZ)
 READ(5,123,ERR=11)JJJ
 123 FORMAT(I10)
 IF(JJJ.EQ.-999)INFO(IY,IX,IZ)=KKK
 IF(JJJ.GE.0)INFO(IY,IX,IZ)=JJJ
 ENDIF
 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE FINAL(LLL)
C--------------------------------------------
---------------------------
CBROUT TO WRITE DOWN THE FINAL RESULT (FROM
MENU&SAVE)
 CHARACTER*24 KLAM
 INCLUDE'.F.FILES/COM'
 72 FORMAT(A12,11('|',I5),'|')
CDIV
 INFO(8,0,14)=1000*INFO(0,15,0)/
(MAX(1,INFO(3,0,14)))
 info(8,0,15)=1000*info(0,15,0)/
(max(1,info(4,0,14)))
 DO 1 IX=8,14
 IF(IX.EQ.8)KLAM='UTILISATION%'
 IF(IX.EQ.9)KLAM='PASSED JOBS'
 IF(IX.EQ.10)KLAM='PROS Number'
 IF(IX.EQ.11)KLAM='IDLE TIMES'
 IF(IX.EQ.12)KLAM='TOT.PR TIME'
 IF(IX.EQ.13)KLAM='TOT.IN-WAIT'
 IF(IX.EQ.14)KLAM='MACHNG TIME'
 WRITE(721,72)KLAM,(INFO(0,IX,IZ),IZ=1,MACHS)
 1 CONTINUE
 KLAM='Mean Proc T'
 WRITE(721,72)KLAM,(info(0,36,iz),iz=1,machs)
 f26=info(0,26,0)
 f510=info(5,0,10)/1000.
 f59=info(5,0,9)/1000.
 f69=info(6,0,9)/1000.
 f610=info(6,0,10)/1000.
 f56=info(5,0,6)/1000.
 f55=info(5,0,5)/1000.

WRITE(721,390)info(8,0,10),F26,INFO(0,13,0),I
NFO(0,20,0),
```

APPENDIX

```
@INFO(0,21,0),INFO(0,1,0),INFO(7,0,1),INFO(7,
0,2),INFO(0,7,0)

@,F55,F56,INFO(5,0,11),INFO(6,0,11),F510,F610
,F59,F69,

@INFO(0,12,0),INFO(7,0,5),INFO(7,0,7),INFO(7,
0,6),INFO(7,0,3),
 @INFO(0,25,0)
390 FORMAT('RL=',I2,1x,'MPT=',F3.1,1x,'I
WT=',I8,1x,'A WT=',I8,1x,
 @'X WT=',I8,1x,'TIME=',I4,1x,'MX
WISH=',I4,1x,'MX Q MC=',I4,1x,
 @'EVENT=',I8,' RESV DD: RMS
TD=',F5.1,1x,'CRMS TD=',F4.1,1x
 @,'LT JF=',I4,1x,/,'ER JF=',I4,1x,'M T
F=',F4.1,1x,'M E F=',F4.1,
 @1x,'C M ','T F=',F4.1,1x,'C M E
F=',F4.1,1x,'T P T SHOP=',I8,1x,

@'WISH=',I5,1x,'SARQ(Q15)=',I3,1x,'MAINQs(Q16
)=',I3,1x,
 @'MAX WIST=',I5,1x,'T# Pros=',I8)
 f510=info(5,0,10)/1000.
 f59=info(5,0,9)/1000.
 f69=info(6,0,9)/1000.
 f610=info(6,0,10)/1000.
 f56=info(5,0,6)/1000.
394 FORMAT('STC=',I4,1x,'PASSED Mc
T=',I8,1x,'IDLE T=',I8,1x,
 @'REM P T=',I8,1x,'TOT PASSED T IN
SHOP=',I8,1x,'FIN PR=',I8,1x,
 @'REM PR=',I8,1x,'LEAD T(USED IN FDSSD:',1x,
 @'LT*
MC(37)=',I3,1x,'NTRY(36)=',I3,1x,'ARV(38)=',I
3)

WRITE(721,394)INFO(1,0,9),INFO(0,10,0),INFO(0
,11,0),INFO(0,14,0)
 @,INFO(0,15,0),INFO(0,24,0),INFO(0,25,0)-
INFO(0,24,0),
 @INFO(0,37,0),INFO(0,36,0),INFO(0,38,0)
CALL IF(LLL.EQ.111)THEN
CALL DO 3 K=1,JOBS
CALL
WRITE(752,7)K,INFO(K,16,0),INFO(K,17,0),(INFO
(K,11,N),N=1,MACHS)
CALL
WRITE(753,777)K,INFO(K,3,0),INFO(K,39,0),INFO
(K,19,0),INFO(K,30,0)
CALL @ ,INFO(K,3,0)-INFO(K,30,0)
CALL 7 FORMAT(I3,X,7(I4,X))
CALL 777 FORMAT(I3,X,5(I5,X))
CALL 3 CONTINUE
CALL LAST=1
CALL ENDIF
 if(info(0,15,5).le.0)goto111
 do 1245 Jo=1,info(0,15,5)
 if(info(jo,15,5).lt.0)goto1245
 if(info(info(jo,15,5),15,0).ne.0)
 @ info(jo,15,5)=
info(Jo,15,5)*info(info(jo,15,5),15,0)/
 @ max(1,abs(info(info(jo,15,5),15,0)))
 1245 continue
C RETURN if you like
write(200,9919)(info(jjj,15,5),jjj=1,120)
C RETURN if you like
write(200,9919)(info(jjj,15,5),jjj=121,240)
C RETURN if you like
write(200,9919)(info(jjj,15,5),jjj=241,360)
C RETURN if you like
write(200,9919)(info(jjj,15,5),jjj=361,info(0
,15,5))
9919 format(120(I4,x))
 111 RETURN
 END
CALLC-----------------------------------------
------------------------------
CALL SUBROUTINE FINISH(IFI)
CALLC-----------------------------------------
------------------------------
CALLCBROUT:RELEAS1. HISTORY JOB ON EACH M/
C(FILE NAME= RESULT1/3) MAX MACHS = 10
CALL INCLUDE'.F.FILES/COM'
CALL IWAIT=INFO(IFI,22,0)-INFO(IFI,21,0)
CALL
WRITE(718,1)(INFO(IFI,IX,0),IX=1,4),INFO(IFI,
7,0),(INFO(IFI,IX,0),
CALL
@IX=25,27),INFO(IFI,12,0),INFO(IFI,20,0),INFO
(IFI,13,0),
CALL @IWAIT,INFO(IFI,11,0),INFO(IFI,15,0)
CALL
WRITE(719,2)INFO(IFI,1,0),INFO(IFI,30,0),(INF
O(IFI,IX,0),IX=16,19)
CALL @,(INFO(IFI,12,J),J=1,MACHS)
CALL 1
FORMAT(I4,'|',I4,'|',I5,'|',I2,'|',I3,'|',I4,
'|',
CALL @I5,'|',I3,15('|',I5),'|')
CALL 2 FORMAT(I4,'|',5(I5,'|'),8(I4,'|'))
CALLC MACHS=8
CALL
WRITE(720,3)INFO(IFI,1,0),(INFO(IFI,13,IZ),IZ
=1,8),
CALL @(INFO(IFI,11,IZ),IZ=1,MACHS)
CALL 3 FORMAT(I4,'|',8(I4,'|'),8(I4,'|'))
CALL RETURN
CALL END
C-------------------------------------------
--------------------------
 SUBROUTINE QUEUE2(INO,KK)
C-------------------------------------------
--------------------------
C BROUT:FROM QUEUE1
 IF(INO.EQ.0)THEN
 KK=0
 ELSE
 KK=99999
 ENDIF
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE QUEUE1(II,K17,K38,L1,L2,L3,KF)
C-------------------------------------------
--------------------------
CBROUT: FROM QUEUE
 INCLUDE'.F.FILES/COM'
 CHARACTER*7 B(4)
 B(1)='SHP '
 B(2)='BUF '
 B(3)='MAC '
 B(4)='BUF '
 K10=INFO(II,K17,1)
 K20=INFO(II,K17,2)
 K30=INFO(II,K17,3)
 CALL QUEUE2(K10,KK)
 CALL QUEUE2(K20,K2)
 CALL QUEUE2(K30,K3)
 WRITE(6,200)B(KF),

@K10,MIN(KK,INFO(K10,10,0)),MIN(KK,INFO(K10,1
2,0)),

@MIN(KK,INFO(K10,K38,L1)),MIN(KK,INFO(K10,3,0
)),

@K20,MIN(K2,INFO(K20,10,0)),MIN(K2,INFO(K20,1
2,0)),
```

APPENDIX

```
@MIN(K2,INFO(K20,K38,L2)),MIN(K2,INFO(K20,3,0
)),

@K30,MIN(K3,INFO(K30,10,0)),MIN(K3,INFO(K30,1
2,0)),

@MIN(K3,INFO(K30,K38,L3)),MIN(K3,INFO(K30,3,0
))
 200 FORMAT(A4,3(5(I3,X),X))
 RETURN
 END
C---------------------------------------------
---------------------------
 SUBROUTINE QUEUE
C---------------------------------------------
---------------------------
CBROUT: TO SHOW QUEUES
 INCLUDE'.F.FILES/COM'
C THIS FOR 10 MACHINE MAX. & FOR STARTING
NUMBER OF JOBS IS 14
 DO 1010 MM=1,MACHS
 CALL MOVE(MM,16)
 1010 CONTINUE
C CONTINUE WRITE(6,11)INFO(0,1,0)
 11 FORMAT('TIME=',I5)
C CONTINUE
WRITE(6,1)(INFO(I1,15,1),I1=1,INFO(0,15,1))
C CONTINUE WRITE(6,72)INFO(5,0,14)
 DO 10 I2=1,100
 IF(
INFO(I2,16,1).EQ.0.AND.INFO(I2,16,2).EQ.0.

@AND.INFO(I2,16,3).EQ.0.AND.INFO(I2,16,4).EQ.
0.

@AND.INFO(I2,16,5).EQ.0.AND.INFO(I2,16,6).EQ.
0.

@AND.INFO(I2,16,7).EQ.0.AND.INFO(I2,16,8).EQ.
0.

@AND.INFO(I2,16,9).EQ.0.AND.INFO(I2,16,10).EQ
.0)GOTO12
 IF(MACHS.LE.3)CALL QUEUE1(I2,16,38,0,0,0,1)
C CONTINUE
IF(MACHS.GT.3)WRITE(6,2)(INFO(I2,16,I3),I3=1,
15)
 10 CONTINUE
 12 CONTINUE
C CONTINUEWRITE(6,72)INFO(7,0,5)
 DO 20 I4=4,1,-1
C CONTINUE
IF(MACHS.GT.3)WRITE(6,3)(INFO(I4,17,I3),I3=1,
15)
C THREE MACHINE ONLY
 IF(MACHS.LE.3)CALL QUEUE1(I4,17,12,1,2,3,2)
 20 CONTINUE
C CONTINUE
IF(MACHS.GT.3)WRITE(6,7)(INFO(0,8,IZ),IZ=1,MA
CHS)
C CONTINUE
IF(MACHS.LE.3)WRITE(6,700)(INFO(0,8,IZ),IZ=1,
3)
C CONTINUE
IF(MACHS.GT.3)WRITE(6,4)(INFO(0,17,I3),I3=1,1
3)
C THREE MACHINE ONLY
 IF(MACHS.LE.3)CALL QUEUE1(0,17,12,1,2,3,3)
C CONTINUE
IF(MACHS.GT.3)WRITE(6,7)(INFO(10,17,JH),JH=1,
13)
C CONTINUE
IF(MACHS.LE.3)WRITE(6,700)(INFO(10,17,JH),JH=
1,3)
 DO 30 I2=1,100
 IF(
INFO(I2,23,1).EQ.0.AND.INFO(I2,23,2).EQ.0.

@AND.INFO(I2,23,3).EQ.0.AND.INFO(I2,23,4).EQ.
0.

@AND.INFO(I2,23,5).EQ.0.AND.INFO(I2,23,6).EQ.
0.

@AND.INFO(I2,23,7).EQ.0.AND.INFO(I2,23,8).EQ.
0.

@AND.INFO(I2,23,9).EQ.0.AND.INFO(I2,23,10).EQ
.0)GOTO31
C CONTINUE
IF(MACHS.GT.3)WRITE(6,5)(INFO(I2,23,I3),I3=1,
15)
C THREE MACHINE ONLY
 IF(MACHS.LE.3)CALL QUEUE1(I2,23,12,1,2,3,4)
 30 CONTINUE
 31 CONTINUE
C CONTINUE WRITE(6,72)INFO(4,0,14)
C CONTINUE WRITE(6,6)(INFO(I7,15,5),I7=1,25)
C CONTINUE
WRITE(6,61)(INFO(I7,15,5),I7=26,44)
C CONTINUE
WRITE(6,61)(INFO(I7,15,5),I7=45,INFO(0,15,5))
 1 FORMAT('AR',40(I3,X))
 2 FORMAT('P-SHP',15(X,I3,X))
 3 FORMAT('I-BUF',15(X,I3,X))
 4 FORMAT('MACHN',15(X,I3,X))
 5 FORMAT('M-BUF',15(X,I3,X))
 6 FORMAT('EXT',26(I2,X))
 61 FORMAT(19(I3,X))
 72 FORMAT('--------------------------------',
 @ '---------------------------------N# OF
JOBS=',I3)
 7 FORMAT(
 @' ',I2,'== ',I2,'== ',I2,'== ',I2,'==
',I2,'== ',I2,'== '
 @,I2,'== ',I2,'== ',I2,'== ',I2,'== ',I2,'==
',I2,'== ',I2
 @,'== ')
 700 FORMAT(' ',I2,'=== ',I2,'=== ',
 @,I2,'=== ')
 III=INFO(2,0,3)
 100 CONTINUE
CC continue WRITE(6,40)
 40 FORMAT(
 @' TYPE 1 <MAIN MENU> RTRN TO EXIT')
 if(info(0,46,0).eq.0)then
 READ(5,99,ERR=100)INFO(2,0,3)
 IF(INFO(2,0,3).EQ.0)INFO(2,0,3)=III
 99 FORMAT(I4)
 else
ccccccccccccccc info(2,0,3)=info(1,0,9)*5
 info(2,0,3)=info(0,46,0)
 endif
 RETURN
 END
C---------------------------------------------
---------------------------
 SUBROUTINE TEKNIK
C---------------------------------------------
---------------------------
CBROUT TO SET THE TECHNIQUE REQUIRED (FROM
MENU AND FROM CHNTKN)
 CALL CHOSNG
 CALL THEORY
 RETURN
 END
C---------------------------------------------
---------------------------
 SUBROUTINE ADD(IY,IX,K1,M,ITOTAL,IPROC)
C---------------------------------------------
---------------------------
CBROUT FROM MENU.REDSTRT.FILL AND REDCNT.FILL
```

APPENDIX

```
 INCLUDE '.F.FILES/COM'
C ADD(STRT PNT Y.X.Z
,M(DIRECTION),TOTAL(SUM),NUMBER OF PROCES)
 GOTO(10,20,30),M
C Y-DIRECTION
 10 DO 1 I1=IY,JOBS
 IF(INFO(I1,0,0).EQ.0)THEN
 ITOTAL=ITOTAL+INFO(I1,IX,K1)
 IF(INFO(I1,IX,K1).GT.0)IPROC=IPROC+1
 ENDIF
 1 CONTINUE
 20 GOTO111
C Z-DIRECTION
C INFO(10,0,3)=MAX DIMENSION IN Z
 30 DO 3 I3=K1,MACHS
 ITOTAL=ITOTAL+INFO(IY,IX,I3)

IF(INFO(IY,IX,I3).GT.0.OR.(IX.EQ.25.AND.INFO(
IY,26,I3).GT.0))
 @IPROC=IPROC+1
 3 CONTINUE
 GOTO111
 111 CONTINUE
 RETURN
 END
C---------------------------------------------
------------------------
 SUBROUTINE CHOSNG
C---------------------------------------------
------------------------
C BROUT: TO CHOOSE(TEKNIK): 1-TECHNIQUE OR
THEORY OR 2-MEASURE OF PERFORMANCE
 INCLUDE'.F.FILES/COM'
 11 WRITE(6,4)
 4 FORMAT('THE DEFAULT MEASURE IS KEEPING WIP
AS MINIMUM &SPT THEORY'
 @,':-',/,'1 -CHANGE THEORY.',/,'2 -CHANGE
PERFORMANCE MEASURE.',/,
 @' TYPE 1-2 RTRN TO EXIT')
 READ(5,1,ERR=11)IT
 IF(IT.EQ.0)GOTO111
 IF(IT.EQ.2)THEN
 12 WRITE(6,2)
 IT2=INFO(8,0,9)
 READ(5,1,ERR=12)INFO(8,0,9)
 IF(INFO(8,0,9).EQ.0)INFO(8,0,9)=IT2
 IF(INFO(8,0,9).EQ.0)INFO(8,0,9)=8
 2 FORMAT('WHICH MEASURES OF PERFORMANCE',/
,'DO YOU WANT TO BE ',
 @'CONSIDERED?',/
,'=============================',/,' 1-FLOW
RAT',
 @'E(SPT) 2-IGNORE TECHNS 3-CONGESTION IN
SHOP',/,'.4-MIN WIP ',
 @' 5-TARDINESS 6-IMPROVE USED RULE ',/,' 7-
CAPACITY ',
 @' 8-CUSTOMER 9-TIME SPENT IN SHOP',/,'10-
COSTS ',
 @' 11-LASQ 12-INTERFERANCE',/,'13-BOTTLENECK
14-P',
 @'ARALEL M/CS 15-PREVENTIVE MAINTENANCE',/
,'16-CONTROLLED ARRI',
 @'VAL','17- FDSSD USE ORIGINAL DUE DATES',/,
 @' TYPE 1-20 <DEFAULT 4> RTRN TO EXIT')
 ENDIF
 IF(IT.EQ.2)IG=999
 IF(IT.NE.2)IG=0
 IF(IT.EQ.1)THEN
 13 WRITE(6,3)
 IT1=INFO(8,0,10)
 READ(5,1,ERR=13)INFO(8,0,10)
 IF(INFO(8,0,10).EQ.0)INFO(8,0,10)=IT1
 IF(INFO(8,0,10).EQ.12)THEN
 WRITE(6,* )'ENTER DEGREE OF IMPORTANCE
(9=V.IMP ==>> 0=NORMAL)'
 READ(5,1)IMPO
 IF(IMPO.EQ.0)IMPO=5
 INFO(8,0,11)=IMPO
 ENDIF
 1 FORMAT(I4)
 IF(INFO(8,0,10).EQ.0)INFO(8,0,10)=3
 3 FORMAT('TO SET PRIORITIES FOR EACH JOB :-
',/,'WHICH RULE DO YOU',
 @' WANT TO BE USED?',/
,'===================================',/,
 @'1-FRFS 2-EDD 3-SPT 4-FCFS 5-MRPT 6-S/ROP 7-
SPT/EDD 8-LRPT',/,
 @'9-QINM 10-SPTM 11-COMPST 12-SIMCU 13-SLCK
14-S/OP 15-WSPT',/,
 @'16-WEDD 17-FDSSD 18-COVERT 19- 20-LASQ',/,
 @' TYPE 1-20 <DEFAULT 3> RTRN TO EXIT)')
 ENDIF
 GOTO11
 111 RETURN
 END
C---------------------------------------------
-------------------------
 SUBROUTINE THEORY
C---------------------------------------------
-------------------------
C BROUT: FROM FILL-NXTDO-TEKNIK. TO SET DUE
TIME OF ENTRY TO SHOP
 INCLUDE '.F.FILES/COM'
 LP0=0
 LP1=1
 LP2=2
 LP4=4
 LP3=3
 LP9=9
 LP10=10
 LP14=14
 LP17=17
 LP12=12
 LP26=26
 LP29=29
 LP32=32
 LP31=31
 LP43=43
 LP35=35
 LP36=36
 LP38=38
 CALL PRITYM(LP3,LP10,LP0,LP0)
 DO 1921 I=1,JOBS
 IF(INFO(I,0,0).EQ.11)GOTO1921
 INFO(I,29,0)=INFO(I,3,0)-INFO(I,12,0)-
INFO(0,38,0)-
 @info(0,36,0)-info(0,37,0)
 1921 CONTINUE
 ITHEORY=INFO(8,0,10)

GOTO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,1
7,18,19,20)
 @,ITHEORY
C PRINT*,'FIRST ARRIVE IN SHOP FIRST SERVE'
 1 DO 204 I =1,JOBS
 IF(INFO(I,0,0).EQ.11)GOTO204
 INFO(I,10,0)=INFO(I,1,0)-INFO(4,0,14)
 204 CONTINUE
 GOTO111
C PRINT*,'EARLIEST DUE DATE FIRST'
 2 CALL PRITYM(LP3,LP10,LP0,LP0)
 CALL PRITYS(LP2,LP3,LP9)
 GOTO111
C WRITE(6,* )'SHORTEST PROCESSING TIME RULE'
 3 LP32=32
 IF(INFO(8,0,13).NE.1)CALL
PRITYM(LP12,LP10,LP0,LP0)
 IF(INFO(8,0,13).EQ.1)CALL
PRITYM(LP26,LP10,LP0,LP0)
 CALL PRITYS(LP12,LP29,LP32)
 GOTO111
C PRINT*,'FIRST COME FIRST SERVE RULE'
```

APPENDIX

```
 4 CALL FCFS
 GOTO112
C PRINT*,'MINIMUM REMAINING PROCESSING TIME'
 5 CALL PRITYM(LP14,LP10,LP0,LP0)
 GOTO111
C PRINT*,'SLACK PER REMAINING OPERATION'
 6 DO 200 I=1,JOBS
 FO14=INFO(I,14,0)
 FO15=INFO(I,15,0)
 FO7=INFO(I,7,0)
 FO9=INFO(I,9,0)
CDIV
 INFO(I,43,0)=((FO15-FO14)/MAX(1,(FO7-
FO9))+0.5)
 200 CONTINUE
 CALL PRITYM(LP43,LP10,LP0,LP0)
 GOTO111
C PRINT*,'SPT - EDD - FRFS'
 7 FO4=INFO(5,0,14)
 FO3=INFO(4,0,14)
CDIV
 P=100*((FO4-FO3)/
(1+FO3))*(MAX(1.,.5+(INFO(9,0,12)/10.)))
CDIV
 D=(200-P)*(MAX(1,INFO(9,0,12)/10.))
 DO 208 I=1,JOBS
 INFO(I,43,0)=MAX(0,(INFO(I,3,0)-
INFO(0,1,0)))-
 @ (P*INFO(I,12,0)+D*INFO(I,3,0) )
 208 CONTINUE
 CALL PRITYM(LP43,LP10,LP0,LP0)
 GOTO111
 8 PRINT*,'LONGEST REMAINING PROCESSING TIME'
 CALL PRITYM(LP14,LP10,LP0,LP1)
 GOTO111
C PRINT*,'QUEUE IN NEXT MACHINE RULE'
 9 continue
 GOTO112
C PRINT*,'SHORTEST PROCESSING TIME AT THIS
MACHINE'
 10 continue
 GOTO112
C PRINT*,'COMPOSIT RULE'
 1010
PRINT*,'IP,IDUE,IWEIGH,COST,IRMPRT,IRPROCN,IS
LACK,IWIP,FAFS'

READ(5,*)IP,IDUE,IWEIGHT,ICOST,IRMPROCT,IRMPR
OCN,ISLACK,IWIP,IFA
 11 IF(IP.EQ.0)GOTO1010
 ZZZZ=INFO(0,1,0)
 DO 999 I=1,JOBS
CDIV

INFO(I,31,0)=1000*IP*(10*(INFO(I,12,INFO(8,C,
8))/INFO(I,26,0)))+
CDIV
 @ IDUE*(INFO(I,3,0)/MAX(1.,ZZZZ))*10.+
 @ IWEIGHT*INFO(I,4,0)*10.+
 @
ICOST*INFO(I,6,0)*10.+IRMPROCT*INFO(I,14,0)+
 @ IRMPROCN*(INFO(I,9,0)-INFO(I,7,0))+
 @ ISLACK*MAX(1.,(INFO(I,3,0)-ZZZZ))+
CDIV
 @ IWIP*INFO(7,0,5)+10.*IFA/
MAX(1,INFO(I,1,0))
 999 CONTINUE
 CALL PRITYM(LP31,LP10,LP0,LP0)
 GOTO 111
C PRINT*,'SAVE IMPORTANT CUSTOMER RULE'
 12 WRITE(6,201)INFO(8,0,12)
 201 FORMAT('ENTER JOB'S NUMBER TO BE SAVED.
OR 0 TO ARRANGE ',/,
 @'QUEUE IN ORDER THE MORE IMPORTANT IS THE
FIRST.',/,' ',
 @' <DEFAULT',I3,' > RTRN TO EXIT')
```

```
 202 FORMAT(I4)
 READ(5,202)INFO(8,0,12)
 1200 IF(INFO(8,0,12).EQ.0)THEN
 CALL PRITYM(LP4,LP10,LP0,LP0)
 CALL ARANGE(15,-1,0)
 CALL ARANGE(16,MACHS,0)
 CALL UPDOWN(16)
 CALL ARANGE(17,MACHS,0)
 CALL ARANGE(23,MACHS,0)
 ENDIF
 GOTO112
C PRINT*,'SLACK RULE'
 13 DO 203 I=1,JOBS
 INFO(I,43,0)=INFO(I,15,0)-INFO(I,14,0)
 203 CONTINUE
 CALL PRITYM(LP43,LP10,LP0,LP0)
 GOTO 111
C PRINT*,'SLACK/ TOTAL N* OF OPERATION'
 14 DO 205 I=1,JOBS

CDIV
 INFO(I,43,0)=((INFO(I,15,0)-INFO(I,14,0))/
1.*MAX(1,INFO(I,7,0)))
 205 CONTINUE
 INFO(0,43,0)=0
 CALL PRITYM(LP43,LP10,LP0,LP0)
 INFO(0,43,0)=0
 GOTO111
C PRINT*,'WEIGTHED PROCESSING TIME'
 15 DO 206 I=1,JOBS

INFO(I,43,0)=(INFO(I,12,0)*MAX(INFO(I,4,0),1)
)
 206 CONTINUE
 CALL PRITYM(LP43,LP10,LP0,LP0)
 GOTO111
C PRINT*,'WEIGTHED EARLIEST DUE DATE'
 16 DO 207 I=1,JOBS
CDIV
 INFO(I,43,0)=(INFO(I,3,0)/
1.*MAX(INFO(I,4,0),1))
 207 CONTINUE
 CALL PRITYM(LP34,LP10,LP0,LP0)
 GOTO111
C HASD
 17 CALL PRITYM(LP38,LP10,LP0,LP0)
 CALL PRITYS(LP36,LP32,L3)
 GOTO111
 1118 WRITE(6,9191)
 9191 FORMAT('COVERT Q IN THE EQUATION 1-
0.01')
 READ(5,*)Q
 INFO(12,0,9)=Q*10
 18 IF(INFO(12,0,9).EQ.0)GOTO1118
 continue
 GOTO111
 19 CONTINUE
 CALL PRITYM(LP38,LP10,LP0,LP0)
 CALL PRITYS(LP36,LP32,LP3)
 GOTO111
C20 PRINT*,' LOOK AFTER SHOP QUEUES'
 115 WRITE(6,1130)
 1130 FORMAT('SET PRIORITY: 3-ACCORDING EDD',/
,'12- ACCORDING SPT')
 IP=INFO(10,0,10)
 IW=INFO(11,0,10)
 IC=INFO(12,0,10)
 READ(5,114)INFO(9,0,10)

WRITE(6,909)INFO(10,0,10),INFO(11,0,10),INFO(
12,0,9)
 909 FORMAT('<DEFAULT IP=',I2,' IW=',I2,'
IC=',I2,'> RTRN TO EXIT.'
 @ ,/,'VALUE OF IP')
 READ(5,*)INFO(10,0,10)
 PRINT*,'VALUE OF IW'
```

APPENDIX

```
 READ(5,*)INFO(11,0,10)
 PRINT*,'VALUE OF IC'
 READ(5,*)INFO(12,0,10)
 114 FORMAT(I2)
 IF(INFO(9,0,10).NE.3)INFO(9,0,10)=12
 20 IF(INFO(9,0,10).EQ.0)GOTO115
 IF(INFO(9,0,10).EQ.12)CALL
PRITYM(LP12,LP10,LP0,LP0)
 IF(INFO(9,0,10).EQ.3)CALL
PRITYM(LP3,LP10,LP0,LP0)
 DO 900 I1=1,JOBS
 IF(INFO(I1,0,0).EQ.11)GOTO900
 IF(INFO(I1,10,0).LE.0)GOTO209
CDIV
 PRI=(10.*JOBS/INFO(I1,10,0))*INFO(10,0,10)
CDIV
 209 WGT=(10.*INFO(I1,7,0)/
MACHS)*INFO(I1,4,0)*INFO(11,0,10)
 CST=INFO(I1,6,0)*INFO(12,0,10)*10.
C DELIVERY
 INFO(I1,35,0)=1000+INFO(I1,3,0)-
INFO(I1,14,0)-INFO(10,17,
 @ ABS(INFO(I1,25,1)))-INFO(0,38,0)-PRI-WGT-
CST
 INFO(I1,29,0)=INFO(I1,3,0)-INFO(I1,14,0)-
INFO(10,17,
 @ ABS(INFO(I1,25,1)))-INFO(0,38,0)
 900 CONTINUE
 CALL PRITYM(LP35,LP10,LP0,LP0)
 GOTO111
 111 CALL ARANGE(15,-1,INFO(8,0,8))
 CALL ARANGE(16,MACHS,INFO(8,0,8))
 CALL UPDOWN(16)
 CALL ARANGE(17,MACHS,INFO(8,0,8))
 CALL ARANGE(23,MACHS,INFO(8,0,8))
 IF(INFO(8,0,10).EQ.8)THEN
 CALL UPDOWN(15)
 CALL UPDOWN(16)
 CALL UPDOWN(17)
 CALL UPDOWN(23)
 ENDIF
 112 CONTINUE
 RETURN
 END
C-------------------------------------------
---------------------------*
 SUBROUTINE FCFS
C-------------------------------------------
---------------------------*
C BROUT: FROM THEORY, AND MOVE.
 INCLUDE'.F.FILES/COM'
 DO 1 II=1,MACHS
 DO 2 III=1,3
 DO 2 I=1,3

IF(INFO(I,17,II).GT.0.AND.INFO(I+1,17,II).GT.
0)THEN
 IKN=INFO(I,17,II)
 IKT=INFO(I+10,17,II)
 IF(IKT.GT.INFO(I+10+1,17,II))THEN
 INFO(I,17,II)=INFO(I+1,17,II)
 INFO(I+1,17,II)=IKN
 INFO(I+10,17,II)=INFO(I+1+10,17,II)
 INFO(I+1+10,17,II)=IKT
 ENDIF
 ENDIF
 2 CONTINUE
 DO 1 III=1,INFO(0,23,II)
 DO 1 I1=1,INFO(0,23,II)

IF(INFO(I1,23,II).GT.0.AND.INFO(I1+1,23,II).G
T.0)THEN
 IF(INFO(I1,24,II).GT.INFO(I1+1,24,II))THEN
 IKN=INFO(I1,23,II)
 IKT=INFO(I1,24,II)
 INFO(I1,23,II)=INFO(I1+1,23,II)
 INFO(I1+1,23,II)=IKN
 INFO(I1,24,II)=INFO(I1+1,24,II)
 INFO(I1+1,24,II)=IKT
 ENDIF
 ENDIF
 1 CONTINUE
 RETURN
 END
C-------------------------------------------
---------------------------*
 SUBROUTINE PRITYM(I1,I2,I,IUPDOW)
C-------------------------------------------
---------------------------*
C BROUT: (IN,OUT,SAMEZ) = 1(MIN. VALUE) IS
HIGHER PRIORITY THAN 10(E.G.)
C BROUT: IX1= X-VALUE OF COLUMN(Y=1-JOBS)THAT
WE NEED TO SET ITS PRIORITY
C BROUT: IX2= X-VALUE OF PLACE WHERE WE NEED
TO PUT THE RESULTED PRIORITY
C BROUT: IZ=Z-VALUE '' '' '' '' '' '''' ''
''TAKE&PUT'''' '' '' '' '' ''
C BROUT: JOBB= TOTAL N# OF PROCESSES
 INCLUDE '.F.FILES/COM'
 JOBB=INFO(0,34,0)
 IF(JOBB.EQ.0)JOBB=JOBS
 DO 1 M=1,JOBB

IF(INFO(M,I1,I).LE.0.OR.INFO(M,0,0).EQ.11)GOT
O3
 IE=-1
 IL=0
 IZERO=0
 DO 2 J=1,JOBB
 KK=0
 IF(INFO(J,I1,I).LE.0)THEN
 KK=1
 IZERO=IZERO+1
 ENDIF
 IF(KK.EQ.1)GOTO2
 IF(INFO(M,I1,I).EQ.INFO(J,I1,I))IE=IE+1
 IF(INFO(M,I1,I).LT.INFO(J,I1,I))IL=IL+1
 2 CONTINUE
 INFO(M,I2,I)=JOBB-IL-IE-IZERO-INFO(4,0,14)
 IF(IUPDOW.EQ.1)INFO(M,I2,I)=JOBB-IL+1-
INFO(4,0,14)
 3
IF(INFO(M,I1,I).EQ.0.OR.INFO(M,0,0).EQ.11)INF
O(M,I2,I)=-1
 1 CONTINUE
 RETURN
 END
C-------------------------------------------
---------------------------*
 SUBROUTINE PRITYS(L1,L2,L3)
C-------------------------------------------
---------------------------*
C BROUT: L1=X FIRST VISIT L2=SECOND : FROM
THEORY - FILL
 INCLUDE'.F.FILES/COM'
 DO 2 IZ=1,MACHS
 IYI=0
 DO 1 IY=1,JOBS
 IF(INFO(IY,L1,IZ).EQ.0)GOTO1
 IYI=IYI+1
 INFO(IYI,35,1)=INFO(IY,L1,IZ)
 IF(INFO(IY,L2,IZ).GT.0)THEN
 IYI=IYI+1
 INFO(IYI,35,1)=INFO(IY,L2,IZ)
 ENDIF
 1 CONTINUE
 INFO(0,34,0)=IYI
 LP35=35
 LP34=34
 LP1=1
 CALL PRITYM(LP35,LP34,LP1,LP0)
 IY2=0
```

```
 DO 4 IY1=1,JOBS
 IF(INFO(IY1,L1,IZ).EQ.0)GOTO4
 IY2=IY2+1
 INFO(IY1,10,IZ)=INFO(IY2,34,1)
 INFO(IY2,34,1)=0
 INFO(IY2,35,1)=0
 IF(INFO(IY1,L2,IZ).GT.0)THEN
 IY2=IY2+1
 INFO(IY1,30,IZ)=INFO(IY2,34,1)
 INFO(IY2,34,1)=0
 INFO(IY2,35,1)=0
 ENDIF
 4 CONTINUE
 INFO(0,34,0)=0
 2 CONTINUE
 RETURN
 END
C---------------------------------------------
--------------------------=
 SUBROUTINE MOVE(JATM,IQ)
C---------------------------------------------
--------------------------=
CBROUT: IT MOVES JOBS FORWARD. IN Q15, Q16, Q17
AND Q23
 INCLUDE'.F.FILES/COM'
 N15=0
 N17=0
 N23=0
 IF(IQ.EQ.17)THEN
 NUMBER=4
 N17=10
 ENDIF
 IF(IQ.EQ.23)THEN
 NUMBER=INFO(0,23,JATM)
 N23=1
 ENDIF
 IF(IQ.EQ.15)THEN
 NUMBER=info(0,15,1)
 N15=1
 JATM=1
 ENDIF
 DO 1 I1=1,NUMBER
 DO 1 I=1,NUMBER-1

IF(INFO(I,IQ,JATM).EQ.0.AND.INFO(I+1,IQ,JATM)
.GT.0)THEN
 INFO(I,IQ,JATM)=INFO(I+1,IQ,JATM)
 IF(N15.GT.0)GOTO3

INFO(I+N17,IQ+N23,JATM)=INFO(I+1+N17,IQ+N23,J
ATM)
 INFO(I+1+N17,IQ+N23,JATM)=0
 3 INFO(I+1,IQ,JATM)=0
 ENDIF
 1 CONTINUE
 RETURN
 END
C---------------------------------------------
--------------------------=
 SUBROUTINE ARANGE(IX,IZ2,IR1)
C---------------------------------------------
--------------------------=
CBROUT ARRANGE QUEUE X=IX Z=IZ2 ACCORDING
PRIORITY SHOWN IN PLACE Z=IPRIORITY
 INCLUDE '.F.FILES/COM'
 IR=IR1
 IZMAC=1
 IN=10
 if(info(8,0,10).eq.4)goto111
C IF IZ2 < 0 MEANS ARRANGE ONE M/C
 IF(IZ2.LT.0)IZMAC=ABS(IZ2)
 DO 1 IZ=IZMAC,ABS(IZ2)
 K17=INFO(0,IX,IZ)
 IF(IX.EQ.17)K17=4
C IR> 0 MEANS SUBPRIORITY
 IF(IR1.GT.0)IR=IZ
 IF(IX.EQ.15)IR=0

IF(INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6)THE
N
 IF(IX.EQ.15)IN=info(7,0,8)
 IF(IX.EQ.16)THEN
 IR=0
 IN=INFO(7,0,8)
 ENDIF
 IF(IX.EQ.17.OR.IX.EQ.23)THEN
 IR=0
 IN=INFO(7,0,9)
 ENDIF
 ENDIF
 DO 1 IY=1,K17
 DO 1 IY2=IY,K17

IF(INFO(INFO(IY,IX,IZ),IN,IR).GT.INFO(INFO(IY
2,IX,IZ),IN,IR))THEN
 KC=INFO(IY,IX,IZ)
 INFO(IY,IX,IZ)=INFO(IY2,IX,IZ)
 INFO(IY2,IX,IZ)=KC
 ENDIF
 1 CONTINUE
 IF(INFO(12,0,12).EQ.16)CALL QUEUE
111 RETURN
 END
C---------------------------------------------
--------------------------
 SUBROUTINE ERROR(LL)
C---------------------------------------------
--------------------------
CBROUT ERROR MESSAGES. COULD BE MADE ANY WHERE
IN THE PROGRAM.
 WRITE(6,10000)LL
10000 FORMAT('SOME THING WRONG IN ==>>',I3)
 RETURN
 END
C---------------------------------------------
---------------------------
 SUBROUTINE DATA
C---------------------------------------------
---------------------------
C BROUT:SHOW DATA : FROM MENU
 2 WRITE(6,3)
 3 FORMAT('1)JOBS.',/,'2)MACHINES.',/
,'3)SHOP.',/,
@' TYPE 1-3 RTRN TO EXIT')
 READ(5,1)K
 1 FORMAT(I4)
 IF(K.EQ.0)GOTO111
 IF(K.EQ.1)CALL JOBINF
 IF(K.NE.1)GOTO2
 111 CONTINUE
 RETURN
 END
C=====================================*DATE=
=====12-09-1991===========
 PROGRAM JOBSHOP
C---------------------------------------------
---------------------------
 INCLUDE'.F.FILES/COM'
 INCLUDE'.F.FILES/FRM'
CALL OPEN(718,FILE='.F.FILES/R3')
CALL OPEN(719,FILE='.F.FILES/R4')
CALL OPEN(720,FILE='.F.FILES/R5')
CALL OPEN(724,FILE='.F.FILES/R6')
CALL OPEN(753,FILE='.F.FILES/R7')
CALL OPEN(752,FILE='.F.FILES/R8')
CALL WRITE(752,750)
CALL WRITE(753,753)
CALL WRITE(718,718)
CALL WRITE(719,719)
CALL WRITE(720,720)
 OPEN(88,FILE='A.D',STATUS='OLD')
 OPEN(99,FILE='A.N',STATUS='OLD')
```

APPENDIX

```
 OPEN(721,FILE='.F.FILES/R1')
 open(200,file='.F.FILES/R2')
 write(200,700)
 WRITE(721,721)
 CALL DEFLT
 1 CALL MENU
 GOTO1
 END
C------------------------------------------------
-------------------------
 SUBROUTINE REDSTR(KLM)
C------------------------------------------------
-------------------------
 INCLUDE '.F.FILES/COM'
 INCLUDE'.F.FILES/FRM'
 OPEN(10,FILE='.F.FILES/ST10',STATUS='OLD')
 READ(10,305,ERR=151)MACHS
 READ(10,3051,ERR=151)JOBS
 INFO(3,0,14)=JOBS
 INFO(1,0,1)=MACHS
 DO 3 J=1,JOBS
 INFO(J,1,0)=J
 10
READ(10,306,END=14,ERR=151)INFO(J,0,0),(INFO(
J,N,0),N=2,4)

@,INFO(J,6,0),(INFO(J,25,K),K=1,8),(INFO(J,26
,M),M=1,8)
 3 CONTINUE
 14 READ(10,3054,END=151,ERR=151)
 DO 5 K=1,MACHS

READ(10,307,END=151,ERR=151)(INFO(0,L,K),L=0,
4),
 @(INFO(0,L1,K),L1=6,7)
 5 CONTINUE
 151 CLOSE(10)
 15 CONTINUE
 CALL FILL
 RETURN
 END
C------------------------------------------------
-----------------------
 SUBROUTINE REDCNT
C------------------------------------------------
-----------------------
C BROUT FROM NXTDO
 INCLUDE'.F.FILES/COM'
 JOBOLD=JOBS
 NEWJOB=INFO(7,0,12)
 IF(NEWJOB.EQ.0)NEWJOB=1
 JOBS=JOBS+NEWJOB
 INFO(3,0,14)=JOBS
 INFO(7,0,13)=JOBOLD
 INFO(7,0,12) =NEWJOB
 DO 2 I=1+JOBOLD,JOBS
 3
FORMAT(I2,I4,X,I5,X,I1,X,I2,X,8(I2,X),8(I3,X)
,I5)
 33
READ(88,3,ERR=15,END=119)(INFO(I,N,0),N=1,4),
INFO(I,6,0),(INFO

@(I,25,L),L=1,8),(INFO(I,26,M),M=1,8),info(i,
47,0)
 INFO(I,1,0)=I
CCC 6 FORMAT(I4,X,8(I2,X),8(I3,X))
CCC
WRITE(6,6)INFO(I,1,0),(INFO(I,25,M),M=1,8),(I
NFO(I,26,M),M=1,8)
 GOTO2
 119 WRITE(6,*)'CONTINUOS ARRIVAL FILE IS
EMPTY FILE IS REPEATED'
 REWIND(88)
 GOTO33
 2 CONTINUE
```

```
 GOTO111
 15 IF(I.EQ.(1+JOBOLD))GOTO 1110
 WRITE(6,109)
 109 FORMAT('ERROR! PUT DATA IN A PROPER WAY.')
 111 CALL FILL
CDIV
 info(2,0,14)=10000.*info(3,0,14)/
max(1,info(0,1,0))
 GOTO 1111
 1110 WRITE(6,*)'NO DATA'
 1111 continue
 RETURN
 END
C------------------------------------------------
---------------------------
 SUBROUTINE FILL
C------------------------------------------------
---------------------------
C BROUT:TO FILL UP INFORMATION. PR1,ROUT1
AT=>12,1--PR2,ROUT2=>29,28
C BROUT(MENU.REDSTR AND REDCNT)
 INCLUDE '.F.FILES/COM'
 JOBOLD=INFO(7,0,13)
 INFO(0,39,0)=JOBOLD
 DO 1 I1=1+JOBOLD,JOBS
 IF(INFO(I1,0,0).NE.0)GOTO1
C SETTING DELIVERY DATE
 INFO(I1,3,0)=INFO(I1,3,0)+INFO(0,1,0)
C TOT PROC TIME OF EACH JOB AT ALL MACHINES&
NUMBER OF PROC
 IPROCTIM=0
 IPROC=0
 NO=0
 CALL ADD(I1,26,1,3,IPROCTIM,IPROC)
C JOB HAS MAX N# OF PROCESSES
 IF(IPROC.GT.INFO(10,0,3))THEN
 INFO(10,0,3)=IPROC
 INFO(10,0,4)=I1
 ENDIF
C JOB HAS MAX PROCESSING TIME
 IF(IPROCTIM.GT.INFO(10,0,1))THEN
 INFO(10,0,1)=IPROCTIM
 INFO(10,0,2)=I1
 ENDIF
 INFO(I1,12,0)=IPROCTIM
 INFO(I1,25,0)=IPROC
 INFO(I1,7,0)=INFO(I1,25,0)
 I6=1
 6 IF(INFO(I1,25,I6).GT.0)THEN
C PUT PROCES TIMES&ROUTING AT PROPER PLACE. 1-
28 THEN31 IF GT 0
 IF(INFO(I1,1,(INFO(I1,25,I6))).GT.0)THEN
 IF(INFO(I1,28,(INFO(I1,25,I6))).GT.0)THEN
 WRITE(6,*)'MAXIMUM NUMBER OF VISITS ARE 2
ONLY'
 ELSE
 INFO(I1,28,INFO(I1,25,I6))=I6
 INFO(I1,29,INFO(I1,25,I6))=INFO(I1,26,I6)
C SUBDUE DATES
CDIV

INFO(I1,3,I6)=.5+INFO(I1,26,I6)*INFO(I1,3,0)/
INFO(I1,12,0)
C SECOND VISIT
 IF(INFO(0,24,I6).LT.(1))INFO(0,24,I6)=1
 INFO(I1,6,INFO(I1,25,I6))=1
 INFO(I1,25,I6)=-1*INFO(I1,25,I6)
 ENDIF
 ELSE
 INFO(I1,1,INFO(I1,25,I6))=I6
 INFO(I1,12,INFO(I1,25,I6))=INFO(I1,26,I6)
CDIV

INFO(I1,2,I6)=.5+INFO(I1,26,I6)*INFO(I1,3,0)/
INFO(I1,12,0)
C FIRST VISIT
```

APPENDIX

```
 INFO(I1,6,INFO(I1,25,I6))=0
C TO PREVENT CALCULAT IT AGAIN
 INFO(I1,25,I6)=-1*INFO(I1,25,I6)
 ENDIF
 ENDIF
 I6=I6+1
 IF(INFO(I1,25,I6).GT.0)GOTO6
 DO 66 I66=1,I6
C TO RETURN THE POSITIVE VALUE BACK.
 INFO(I1,25,I66)=INFO(I1,25,I66)*(-1)
 66 CONTINUE
C MEAN PROCESS TIME(TOT PROC.T./ N* OF PROCES)
CDIV
 INFO(I1,26,0)=(INFO(I1,12,0)/INFO(I1,25,0))
C REMAING TIME FOR PROCESSING
 INFO(I1,14,0)=INFO(I1,12,0)
 INFO(0,14,0)=INFO(0,14,0)+INFO(I1,14,0)
C REMAINING TIME TO DUE DATE
 INFO(I1,15,0)=INFO(I1,3,0)-INFO(0,1,0)
 1 CONTINUE
C INCREASE OLDJOB INFO(7,0,13)
 INFO(7,0,13)=INFO(7,0,13)+INFO(7,0,12)
C RETURN NEWJOB COUNTER TO 0.

IF(INFO(1,0,9).LT.INFO(0,1,0).OR.INFO(8,0,2).
EQ.0)GOTO9019
 READ(99,909)INFO(7,0,12),info(0,0,0)
 909 FORMAT(I5,x,I5)
C TOTAL PROCESSING TIME & TOT N# OF PROCES AT
EACH M/C.
 9019 DO 4 I4=1,MACHS
 ITOTP=INFO(0,12,I4)
 ITOTB=INFO(0,10,I4)
 GOTO(10,20),INFO(0,24,I4)+1
C SECOND
 20 CALL ADD(JOBOLD+1,29,I4,1,ITOTP,ITOTB)
C FIRST
 10 CALL ADD(JOBOLD+1,12,I4,1,ITOTP,ITOTB)
 INFO(0,12,I4)=ITOTP
 INFO(0,10,I4)=ITOTB
 TOTP=ITOTP
C MEAN PROCESSING TIME AT EACH MACHINE
CDIV
 INFO(0,36,I4)=.5+(TOTP/(MAX(1,ITOTB)))
C M/C HAS MIN MACHINING TIME(IDLEST)

IF(INFO(0,12,I4).LT.INFO(1,0,3).OR.INFO(1,0,3
).EQ.0)THEN
 INFO(1,0,2)=I4
 INFO(1,0,3)=INFO(0,12,I4)
 ENDIF
C M/C HAS MAX MACHINING TIME(BUSY)

IF(INFO(0,12,I4).GT.INFO(1,0,5).OR.INFO(1,0,5
).EQ.0)THEN
 INFO(1,0,4)=I4
 INFO(1,0,5)=INFO(0,12,I4)
 ENDIF
 4 CONTINUE
C TOTAL PROCESSING TIME IN THE SHOP ALL JOBS
ALL M/C
 ITOTAL=INFO(0,12,0)
 CALL ADD(JOBOLD+1,12,0,1,ITOTAL,IPROC)
 INFO(0,12,0)=ITOTAL
C TOTAL N# OF ALL PROCESSES
 IPR=INFO(0,25,0)
 CALL ADD(JOBOLD+1,25,0,1,IPR,NO)
 INFO(0,25,0)=IPR
C MEAN PROCESSING TIME IN SHOP
 PR=IPR
CDIV
 INFO(0,26,0)=(ITOTAL/MAX(1.,PR))
C ACCORDING PROCESSING TIME AT EACH M/C
 CALL PRITYS(12,29,32)
C DATA HAS BEEN READ=8
 DO 121 I=1,JOBS
 IS=INFO(I,0,0)
 IF(IS.EQ.0)INFO(I,0,0)=8
 121 CONTINUE
 CALL THEORY

IF(INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6.OR.
INFO(8,0,10).EQ.19)
 @ CALL INSERT
 DO 6062 JJ=1,JOBS
C EXPECTED DUE DATE

INFO(JJ,39,0)=max((INFO(JJ,36,abs(INFO(JJ,25,
INFO(JJ,25,0))))+
 @
INFO(JJ,12,abs(INFO(JJ,25,INFO(JJ,25,0))))),
 @
(info(jj,36,abs(info(jj,25,1)))+info(jj,12,0)
))
 6062 CONTINUE
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE TIMPAS(ISMAL)
C-------------------------------------------
--------------------------
CBROUT: (FROM WIP) IT ADVANCES TIME AND
ADVANCE RELATED TIMES
 INCLUDE '.F.FILES/COM'
 COMMON/P/PPASSA,PPASSE,PPASSM
 MINUS=INFO(2,0,8)
 INFO(2,0,8)=999
 IF(MINUS.GT.0)PPASSA=0
 IF(MINUS.GT.0)PPASSM=0
 IF(MINUS.GT.0)PPASSE=0
C INCREASE THE TIME UNTIL NEXT ACTION HAPPENS
 INFO(0,1,0)=INFO(0,1,0)+MINUS
 INFO(2,0,12)=INFO(2,0,12)-MINUS
 INFO(2,0,10)=INFO(2,0,10)-MINUS
 INFO(2,0,7)=INFO(2,0,7)-MINUS
 INFO(2,0,6)=INFO(2,0,6)-MINUS
 INFO(2,0,5)=INFO(2,0,5)-MINUS
 ZX=0
 DO 2 I2=1,JOBS
 IS=INFO(I2,0,0)

if(is.ne.11)info(i2,45,0)=info(i2,45,0)+minus
 if(is.eq.11)ZX=ZX+info(i2,45,0)
C REMAINING TIME OF THIS JOB UNTIL DUE
DATE(SLACK)

IF(INFO(I2,0,0).NE.11)INFO(I2,15,0)=INFO(I2,1
5,0)-MINUS
C PASSED TIME IN SHOP SINCE ENTRY TO MINPQ
 IF((IS.NE.8.OR.IS.NE.3.OR.IS.NE.4).AND.
 @ (IS.GT.0.AND.IS.LT.11))THEN
C PASSED TIME OF A JOB START AFTER ARRIVE TO
SHOP(SARQ)
 INFO(I2,11,0)=INFO(I2,11,0)+MINUS
 INFO(0,15,0)=INFO(0,15,0)+MINUS
 ENDIF
 2 CONTINUE
 info(7,0,11)=ZX
C avarage time spent since a job is received.
 info(4,0,9)=1000*ZX/max(1,info(4,0,14))
C UTILIZATION OF ALL MACHINES =MACHINES PROC
TIME/TOT TIME
CDIV
 INFO(0,8,0)=(10000*INFO(0,10,0)/
(MAX(1,MACHS)*INFO(0,1,0)))
 DO 11 I11=1,MACHS

IF(INFO(0,0,I11).EQ.0.AND.INFO(0,17,I11).GT.0
)
 @INFO(17,17,I11)=INFO(17,17,I11)-MINUS
```

APPENDIX

```
IF(INFO(0,0,I11).EQ.0.AND.INFO(10,17,I11).GT.
0)THEN
 INFO(10,17,I11)=INFO(10,17,I11)-MINUS
 INFO(7,17,I11)=INFO(7,17,I11)-MINUS
C REMAINING TIME OF ALL PROCESSING TIME OF THIS
JOB

INFO(INFO(0,17,I11),14,0)=INFO(INFO(0,17,I11)
,14,0)-MINUS
 ENDIF
C BREAKDOWN
1111
IF(INFO(0,0,I11).EQ.2)INFO(0,19,I11)=INFO(0,1
9,I11)+MINUS

IF(INFO(0,17,I11).GT.0.AND.INFO(0,0,I11).EQ.0
)THEN
C MACHINING TIME OF EACH MACHINE
 INFO(0,14,I11)=INFO(0,14,I11)+MINUS
C TOT REMAINING PROCESSING TIME IN SHOP
 INFO(0,14,0)=INFO(0,14,0)-MINUS
C TOTAL MACHINING TIME IN THE SHOP OF ALL
MACHINES
 INFO(0,10,0)=INFO(0,10,0)+MINUS
 ELSE
C TOTAL IDLENESS IN SHOP
 INFO(0,11,0)=INFO(0,11,0)+MINUS
 ENDIF
C UTILIZATTION OF EACH MACHINE
CDIV
 INFO(0,8,I11)=.5+(INFO(0,14,I11)*100./
MAX(1,INFO(0,1,0)))
C WAITING TIMES
C W.T. (BEFORE)
 DO 16 I16=1,INFO(0,16,I11)
 IF(INFO(I16,16,I11).GT.0)THEN
C MAINQ
C B.W.T.

INFO(INFO(I16,16,I11),20,0)=INFO(INFO(I16,16,
I11),20,0)+MINUS
C TOTAL A.W.T. OF A JOB

INFO(INFO(I16,16,I11),22,0)=INFO(INFO(I16,16,
I11),22,0)+MINUS
C TOTAL A.W.T. IN SHOP
 INFO(0,22,0)=INFO(0,22,C)+MINUS
C TOTAL B.W.T. OF ALL JOBS
 INFO(0,20,0)=INFO(0,20,0)+MINUS
 ENDIF
 16 CONTINUE
C F.W.T. (AFTER)
 DO 21 I21=1,INFO(0,21,I11)

IF(INFO(I21,21,I11).GT.0.AND.INFO(I21,15,0).G
T.0)THEN
C MEXITQ
C F.W.T.

INFO(INFO(I21,21,I11),21,0)=INFO(INFO(I21,21,
I11),21,0)+MINUS
C TOTAL A.W.T. OF A JOB

INFO(INFO(I21,21,I11),22,0)=INFO(INFO(I21,21,
I11),22,0)+MINUS
C TOTAL A.W.T OF ALL JOBS
 INFO(0,22,0)=INFO(0,22,0)+MINUS
C TOTAL A.F.W.T.
 INFO(0,21,0)=INFO(0,21,0)+MINUS
 ENDIF
 21 CONTINUE
C I.W.T. (IN-PROCESS 1)
 DO 17 I17=1,4
 IF(INFO(I17,17,I11).GT.0)THEN
C I.W.T

INFO(INFO(I17,17,I11),13,0)=INFO(INFO(I17,17,
I11),13,0)+MINUS
C TOT. A.W.T. OF A JOB

INFO(INFO(I17,17,I11),22,0)=INFO(INFO(I17,17,
I11),22,0)+MINUS
C TOT. A.W.T. OF ALL JOB
 INFO(0,22,0)=INFO(0,22,0)+MINUS
C TOT. I.W.T.
 INFO(0,13,0)=INFO(0,13,0)+MINUS
C TOT. W.T. OF EACH JOB AT EACH M/C

INFO(INFO(I17,17,I11),13,I11)=INFO(INFO(I17,1
7,I11),13,I11)+MINUS
C TOT W.T. OF ALL JOBS AT EACH M/C
 INFO(0,13,I11)=INFO(0,13,I11)+MINUS
 ENDIF
 17 CONTINUE
C W.T. (IN-PROCESS 2)
 DO 23 I23=1,INFO(0,23,I11)
 IF(INFO(I23,23,I11).GT.0)THEN
C MACHINES' BUFFER
C I2.W.T.

INFO(INFO(I23,23,I11),13,0)=INFO(INFO(I23,23,
I11),13,0)+MINUS
C TOT A.W.T. OF EACH JOB

INFO(INFO(I23,23,I11),22,0)=INFO(INFO(I23,23,
I11),22,0)+MINUS
C TOT A.W.T. OF ALL JOBS
 INFO(0,22,0)=INFO(0,22,0)+MINUS
C TOTAL I2.W.T.
 INFO(0,13,0)=INFO(0,13,0)+MINUS
C TOT W.T. OF EACH JOB AT EACH M/C

INFO(INFO(I23,23,I11),13,I11)=INFO(INFO(I23,2
3,I11),13,I11)+MINUS
C TOT W.T. OF ALL JOBS AT EACH M/C
 INFO(0,13,I11)=INFO(0,13,I11)+MINUS
 ENDIF
 23 CONTINUE

IF(INFO(0,20,I11).LE.INFO(0,1,0))INFO(0,0,I11
)=0
 11 CONTINUE
 111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE RELEAS
C--------------------------------------------
---------------------------
CBROUT NXTDO. RELEASE JOB(S) FROM M/C(S).
 INCLUDE '.F.FILES/COM'
 IDOUBL=0
 DO 12 I12=1,MACHS

IF(INFO(0,17,I12).GT.0.AND.INFO(10,17,I12).LE
.0)THEN
 IDOUBL=IDOUBL+1
 MAC=I12
 INFO(1,0,6)=MAC
 CALLRELEAS1
 INFO(2,0,6)=-1
 ENDIF
 IF(IG.EQ.999)GOTO12
 IF(INFO(8,17,I12).LT.3)THEN
 INFO(0,5,I12)=-1
 ELSE
 INFO(0,5,I12)=1
 ENDIF
 IF(INFO(0,23,I12).GE.1)INFO(0,5,I12)=2
 12 CONTINUE
 IF(IDOUBL.EQ.1)INFO(0,16,0)=INFO(0,16,0)+1
 IF(IDOUBL.EQ.2)INFO(0,17,0)=INFO(0,17,0)+1
```

APPENDIX

```
 IF(IDOUBL.EQ.3)INFO(0,18,0)=INFO(0,18,0)+1
 RETURN
 END
C-------------------------------------------
---------------------------
 SUBROUTINE RELEAS1
C-------------------------------------------
---------------------------
C BROUT: FROM RELEAS. IT TO RELEASE AND SET
RELATED INFO.
 INCLUDE '.F.FILES/COM'
C FINISH ROUTE & FINISH ONE PROCESS
 IF(INFO(0,17,MAC).EQ.0)GOTO111
 IN=INFO(0,17,MAC)

INFO(IN,25,INFO(IN,24,0))=INFO(IN,25,INFO(IN,
24,0))*(-1)
 INFO(IN,24,0)=INFO(IN,24,0)+1
 INFO(0,24,0)=INFO(0,24,0)+1
C TOT. N# OF BATCHES PASSED AT EACH MACHINE SO
FAR
 INFO(0,9,MAC)=INFO(0,9,MAC)+1
C TOTAL NUMBER OF FINISHED
PROCESSES=INFO(0,9,0)=INFO(0,24,0)
 INFO(0,9,0)=INFO(0,24,0)
C IDLENESS
 INFO(15,17,MAC)=INFO(0,1,0)
 DO 1 JATMA=1,MACHS
 IF(INFO(0,17,JATMA).EQ.0)THEN

INFO(0,11,JATMA)=INFO(0,11,JATMA)+INFO(0,1,0)
-INFO(15,17,JATMA)
 INFO(15,17,JATMA)=INFO(0,1,0)
 ENDIF
 1 CONTINUE
C MACHOUTQ COUNTER
 INFO(0,15,4)=INFO(0,15,4)+1
 INFO(0,15,10)=INFO(0,15,10)+1
C PUT JOB
 INFO(INFO(0,15,4),15,4)=IN
 INFO(INFO(0,15,10),15,10)=MAC
C TIME
 INFO(INFO(0,15,4),14,4)=INFO(0,1,0)
C FINISH TIME OF JOB
 INFO(INFO(0,18,MAC),20,MAC)=INFO(0,1,0)
C COUNTER 21 MEXTQ
 INFO(0,21,MAC)=INFO(0,21,MAC)+1
 INFO(INFO(0,21,MAC),21,MAC)=IN
 INFO(INFO(0,21,MAC),22,MAC)=INFO(0,1,0)

C NUMBER OF PROCESSES FINISHED
 INFO(IN,9,0)=INFO(IN,9,0)+1
C SITUATION OF THAT JOB ON MACHINE MAC IS
FINISHED=3
 INFO(IN,8,MAC)=3

C reservation changed
 IF(INFO(IN,36,MAC).GT.INFO(0,1,0)-
INFO(IN,12,MAC))THEN
CCCC print*,'in=',in,' mac',mac,'
proc.t',info(in,12,mac)
 DO 100 I=0,INFO(IN,12,MAC)-1
 INFO(0,45,MAC)=INFO(0,45,MAC)+1

INFO(INFO(0,45,MAC),45,MAC)=INFO(INFO(IN,36,M
AC)+I,40,MAC)
 INFO(INFO(0,45,MAC),44,MAC)=INFO(IN,36,MAC)
CCCCCCC print*,'in,36,mac',info(in,36,mac),'
I',I
CCCCCCC print*,'
INFO(IN,36,MAC)+I',INFO(IN,36,MAC)+I
CCCCCCC
print*,'inf(INF(IN,36,MC)+I,40,mc)',INFO(INFO
(IN,36,MAC)+I,40,mac)
 INFO(INFO(IN,36,MAC)+I,40,MAC)=0
100 CONTINUE
```

```
 ENDIF
C QUEUE LENGTH
 INFO(8,17,MAC)=INFO(8,17,MAC)-1
C IS THIS JOB GOING OUT SHOP (EXIT)
C ------------------------------------
 IF(INFO(IN,24,0).GE.INFO(IN,25,0)+1)THEN
C YES ==>JOB STATUS FINISH
C ------------------------
 INFO(IN,0,0)=11
C TOTAL N# OF EXIT JOBS& REMAINNING JOBS IN
SHOP
 INFO(4,0,14)=INFO(4,0,14)+1
 INFO(7,0,5)=INFO(7,0,5)-1
C CURRENT M/C IS 99999 MEANS EXIT
 INFO(IN,8,0)=99999
C SEXTQ
 INFO(0,15,5)=INFO(0,15,5)+1
 INFO(INFO(0,15,5),15,5)=IN
 INFO(INFO(0,15,5),14,5)=INFO(0,1,0)
C EXIT ORDER
 INFO(0,15,6)=INFO(0,15,6)+1
 INFO(IN,15,6)=INFO(0,15,6)
C FINISHED TIME
 INFO(IN,19,0)=INFO(0,1,0)
C EARLY of finished jobs
 IF(INFO(IN,15,0).GT.0)THEN
 INFO(4,0,13)=INFO(4,0,13)+1
 INFO(4,0,4)=INFO(4,0,4)+INFO(IN,15,0)
CDIV
 INFO(5,0,4)=1000.*info(4,0,4)/
MAX(1,INFO(4,0,14))
CDIV
 info(6,0,4)=1000.*info(4,0,4)/
max(1,info(4,0,13))
CDIV
 INFO(IN,27,0)=MAX(
(.5+info(in,28,0)*INFO(IN,15,0)/
 @ MAX(1,INFO(1,0,11)*INFO(1,0,12)
)),info(in,28,0) )
 INFO(1,0,13)=INFO(IN,27,0)+INFO(1,0,13)
 ENDIF

IF(INFO(IN,15,0).EQ.0)INFO(5,0,13)=INFO(5,0,1
3)+1
C LATE of finished jobs
 IF(INFO(IN,15,0).LT.0)THEN
 INFO(3,0,13)=INFO(3,0,13)+1
 INFO(4,0,1)=INFO(4,0,1)+ABS(INFO(IN,15,0))
C SQUARE TARDINESS

INFO(4,0,6)=INFO(4,0,6)+(INFO(IN,15,0)*INFO(I
N,15,0))
C RMS OF CONDITIONAL MEAN TARDINESS
CDIV
 info(4,0,7)=1000*SQRT(1.*INFO(4,0,6)/
MAX(1,1.*INFO(3,0,13)))
C RMS OF MEAN TARDINESS
CDIV
 info(4,0,8)=1000*SQRT(1.*INFO(4,0,6)/
MAX(1,1.*INFO(4,0,14)))
CDIV

INFO(IN,27,0)=MIN((.5+info(in,6,0)*INFO(IN,15
,0)/MAX(1.,
 @ INFO(1,0,11)*INFO(1,0,12) )),-
1*info(in,6,0))
 INFO(1,0,7)=ABS(INFO(IN,27,0))+INFO(1,0,7)
 ENDIF
CC ACCORDING TO THE RESERVATION DUE DATE.
 DDATE= INFO(IN,36,0)+INFO(IN,12,0)+
 @ INFO(0,37,0)+INFO(0,36,0)+INFO(0,38,0)
 EARLY=DDATE-INFO(0,1,0)
 IF(EARLY.LT.0)THEN
C LATE - TOT. TARDINESS
 INFO(5,0,12)=INFO(5,0,12)+abs(EARLY)
C N# OF TARDY
```

APPENDIX

```
 INFO(5,0,11)=INFO(5,0,11)+1
C MEAN TARDINESS
CDIV
 INFO(5,0,10)=1000*INFO(5,0,12)/
MAX(1,INFO(4,0,14))
C CONDITIONAL MEAN TARDINESS
CDIV
 INFO(5,0,9)=1000*INFO(5,0,12)/
MAX(1,INFO(5,0,11))
C TOT COST OF TARDINESS

INFO(5,0,8)=INFO(5,0,8)+(INFO(IN,6,0)*abs(EAR
LY))
C SQR OF TARDINESS
 INFO(5,0,7)=INFO(5,0,7)+EARLY*EARLY
C SQRT MEAN TARD
CDIV
 info(5,0,6)=1000*SQRT(1.*INFO(5,0,7)/
MAX(1,1.*INFO(4,0,14)))
C SQRT COND MEAN TARD
CDIV
 info(5,0,5)=1000*SQRT(1.*INFO(5,0,7)/
MAX(1,1.*INFO(5,0,11)))
C % OF TARDY JOBS
CDIV
 INFO(6,0,12)=1000*100.*INFO(5,0,11)/
MAX(1,INFO(4,0,14))
 ELSE
 IF(EARLY.GT.0)THEN
CC EARLY - TOT. EARLINESS
 INFO(6,0,7)=INFO(6,0,7)+EARLY
C N# OF EARLY
 INFO(6,0,11)=INFO(6,0,11)+1
C MEAN EARLINESS
CDIV
 INFO(6,0,10)=1000*INFO(6,0,7)/
MAX(1,INFO(4,0,14))
C CONDITIONAL MEAN EARLINESS
CDIV
 INFO(6,0,9)=1000*INFO(6,0,7)/
MAX(1,INFO(6,0,11))
C TOT COST OF EARLINESS

INFO(6,0,8)=INFO(6,0,8)+(INFO(IN,28,0)*EARLY)
 ENDIF
 ENDIF
C LOAD
 INFO(7,0,14)=10000.*info(4,0,14)/
(MAX(1,INFO(0,1,0)))

C NO (NO EXIT)
C -----------
 ELSE
C NEXT M/C TO MOVE

NEXT=INFO(IN,25,INFO(IN,24,0))
C PREPARED TO MACHINE (CURRENT MACHINE)
                IN1=IN
CCCCCCCCIF(INFO(8,0,9).EQ.14)CALLNXTMOV(NEXT,
IN1)
                INFO(IN,8,0)=NEXT
                IF(IN.EQ.0)CALLERROR(104)
                INFO(IN,8,NEXT)=1
C PUT IT IN M/CS' BUFFER
 INFO(0,23,NEXT)=INFO(0,23,NEXT)+1
 INFO(INFO(0,23,NEXT),23,NEXT)=INFO(0,17,MAC)
 INFO(INFO(0,23,NEXT),24,NEXT)=INFO(0,1,0)
C PROCESSING TIMES OF ALL JOBS IN QUEUE23.

INFO(16,17,NEXT)=INFO(16,17,NEXT)+INFO(INFO(0
,17,NEXT),26,INFO(
 @ INFO(0,17,MAC),24,0))
 ENDIF
C M/C SITUATION (IDLE)
 INFO(0,0,MAC)=0
C REMOVE JOB FROM M/C
 INFO(0,17,MAC)=0
 INFO(9,17,MAC)=0
 INFO(6,17,MAC)=0
 MAC=0
 INFO(1,0,6)=0
 111 RETURN
 END
C up to here include at the top

C-------------------------------------------
---------------------------
 SUBROUTINE INSERT
C-------------------------------------------
---------------------------
CBROUT: (FILL) DEAL WITH FDSSD TECHNIQUE
RESERVATION
 INCLUDE'.F.FILES/COM'
 JOBOLD=INFO(0,39,0)
 INFO(0,39,0)=0
C INSERT JOBS WHICH HAVE NO DOUBLE OVERLAP
 DO 1 J=1+jobold,JOBS
 IS=INFO(J,0,0)
 IF(IS.NE.8)GOTO1
C START ACCORDING TO ROUTINE
 MMM=0
 LM=0
 DO 200 MMM=1,INFO(J,25,0)
C THIS MACHINE
 M=INFO(J,25,MMM)
 call gap(m)
C LAST MACHINE
 IF(MMM.GT.1)LM=INFO(J,25,MMM-1)
C TO FIND POSSIBLE STARTING TIME
 IPASS=0
 201 CONTINUE
 IF(MMM.EQ.1)THEN
 IDUE=INFO(J,3,0)
 IST=IDUE-INFO(J,12,0)-INFO(0,37,0)
 IST1=MAX(IST,INFO(0,1,0)).
 ELSE
C MAKE SURE IT START AFTER FINISHING THE
PREVIOUS OPERATION
 IST1=INFO(J,36,LM)+INFO(J,12,LM)
 ENDIF
 if(info(0,1,0).lt.6000)goto202
ccccc include'.F.F'
 202 OK=0
C IF0 IF OK FILLIN
 NOT=1
 kmm=ist1
 IF(INFO(IST1,40,M).EQ.0)CALL
INSERT1(Kmm,M,J,OK,NOT,LM)
 IF(OK.EQ.1)GOTO205
C IF1 SEE EARLIER UP TO NOW IF MACHINE IS FIRST
OPERATION)
 IF(INFO(IST1,40,M).NE.0.AND.MMM.EQ.1)THEN
 DO 123 II=IST1,INFO(0,1,0),-1
 NOT=3
 IF(INFO(II,40,M).EQ.0)CALL
INSERT1(II,M,J,OK,NOT,LM)
 IF(OK.EQ.1)GOTO205
 123 CONTINUE
 ENDIF
C IF2 SEE NEAREST PLACE IF MACHINE IS NOT THE
FIRST OPERATION Forward
 IF(INFO(IST1,40,M).NE.0.AND.MMM.GT.1)THEN
 DO 10 II=IST1,INFO(0,42,M)+1
 NOT=2
 IF(INFO(II,40,M).EQ.0)CALL
INSERT1(II,M,J,OK,NOT,LM)
 IF(OK.EQ.1)GOTO205
 10 CONTINUE
 ENDIF
 IF(IPASS.EQ.1)GOTO203
 IPASS=1
C REMOVE FIRST GAP AND ADD IT TO THE SECOND
```

APPENDIX

```
ONE.
CC IF(OK.NE.1)CALL GATHER(M,IST1,OOK,LM,is)
CC IF(OOK.EQ.1234)GOTO202
 203 NOT=4
 IST2=IST1
C IF JOB NOT INSERTED YET THEN FITIT ANYWHERE
 222 IF(INFO(IST2,40,M).NE.0)GOTO2
 CALL INSERT1(IST2,M,J,OK,NOT,LM)
 IF(OK.EQ.1)GOTO205
 2 IST2=IST2+1
 GOTO222
205 CONTINUE
 200 CONTINUE
 if(info(0,1,0).lt.6000)goto1
ccccc include'.F.F'
 1 CONTINUE
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE INSERT1(IST1,M,J,OK,NOT,LM)
C-------------------------------------------
--------------------------
CBROUT: FROM INSERT
 INCLUDE'.F.FILES/COM'
C CHECK ALONG PROCESSING TIME=FORWORD=
 OK=0
 IOUT=0
 FIRST=0
 MANY=0
 again=0
 7 KKF=0
 KKB=0
 NOTIN=0
 FIT=0
 DO 3 KF=IST1,(IST1+INFO(J,12,M))
 IF(INFO(KF,40,M).NE.0)KKF=KF
 IF(KKF.GT.0)GOTO5
 3 CONTINUE
 5 IF(KKF.GT.0)THEN
C FORWORD IS BUSY-- TRY BACKWORD =
 DO 4 KB=KKF-1,(KKF-INFO(J,12,M)),-1
 IF(INFO(KB,40,M).NE.0)KKB=KB
 IF(KKB.GT.0)GOTO6
 4 CONTINUE
 6 ENDIF
C 1.1 INSERT JOB NO PROBLEM.

 IF(KKF.EQ.0)CALL FILLIN(M,IST1,J,OK)

 IF(OK.EQ.1)GOTO111
 GAPP=KKF-KKB
 TR=INFO(J,12,M)-(KKF-IST1)
 PERCNTG=(INFO(J,12,M)-GAPP)/GAPP
 TRopt=100*TR/info(j,12,m)
 TRmpt=100*TR/info(j,36,m)
C INSERT JOB EARLIER.

 IF(again.ge.3)goto4321
 1234 IF(KKF.GT.0.AND.KKB.EQ.0)then
 IF(M.EQ.INFO(J,25,1))THEN
 if(info(0,1,0).lt.6000)goto2
2 IST1=KKF-INFO(J,12,M)
 CALL FILLIN(M,IST1,J,OK)
 if(info(0,1,0).lt.6000)goto1
cc print*,'j',j,'at machine=',m, 'after
istl=kkf-info(j,12,m)'
cc
write(6,*)(info(jd,40,m),jd=info(0,1,0),istl+
info(0,1,0))
1 FIT=1
 else
 if(kkf-
info(j,12,m).le.info(j,36,lm)+info(j,12,lm))t
hen
 istl=kkf-info(j,12,m)
```

```
 fit=1
 endif
 endif
 ENDIF
 if(ok.eq.1)goto111
 IF(FIT.EQ.1)CALL FILLIN(M,IST1,J,OK)
 if(fit.eq.0)call reserv(j,m,kkf,again)
 IF(OK.EQ.1)GOTO111
 if(again.lt.3)goto7
4321 FIT=0
 ISHORT=0
 NOMORE=0
 NOTIN=0

 IF(KKF.GT.0.AND.KKB.GT.0) then

 if(NOT.eq.4)then
C MOVE JOB BAKWARD OR FORWARD THEN INSERT JOB.
 IF(INFO(0,43,M).GT.0)THEN
 L1=KKB
 L2=KKF
 L5=L1+INFO(J,12,M)
 DO 123 KI=1,INFO(0,43,M)
 IF(NOMORE.GT.0)GOTO124
 IF(INFO(KI,43,M).GT.KKB)THEN
 NOMORe=1
 L3=INFO(KI,43,M)
 L4=L3+INFO(L3,42,M)
 ENDIF
 IF(INFO(KI,43,M).LT.KKB)THEN
 LL3=INFO(KI,43,M)
 LL4=INFO(LL3,42,M)+LL3
 ISHORT=2
 ENDIF
123 CONTINUE
124 IOUT=0
 IF(NOMORE.EQ.0.AND.ISHORT.EQ.0.OR.
 @ (M.NE.INFO(J,25,1).AND.ishort.EQ.0)
)IOUT=333
 IF(IOUT.EQ.333)GOTO321
 MOVEB=0
 MOVEF=0
 ICALLB=0
 ICALLF=0
 IF(ISHORT.EQ.2.AND.M.EQ.INFO(J,25,1))THEN
C MOVE BACKWARD
 ICALLB=11
 MOVEB=MIN(LL4-LL3,L5-L2)
C RETURN BLOCK LL4,L1 BACKWARD ABOUT MOVEB AT
MACHINE M
 IF(MOVEB.GE.L5-L2)ICALLB=12
 ENDIF
 IF(ICALLB.EQ.12)GOTO125
 IF(NOMORE.EQ.1.AND.MOVEB.LT.(L5-L2))THEN
C MOVE FORWARD C VAR
 IF((L4-L3).GE.(L5-L2)-MOVEB.AND.
 @ (L5-L2)-MOVEB.LE.info(0,37,0))goto521
CALL MOVEF=L5-L2-MOVEB
CALL ICALLF=11
CALL ELSE
 MMB=KKF-KKB+MOVEB
 CALL GATHER1(L2,L3,MMB,1,M)
 FIRST=1+FIRST
CALL ENDIF
 521 ENDIF
 125 continue
 IF(ICALLB.EQ.11.or.icallb.eq.12)CALL
GATHER1(LL4,L1,MOVEB,1,M)
CCCCCCCCCC IF(ICALLF.EQ.11)CALL
GATHER1(L3,L2,MOVEF,-1,M)
 IF(ICALLF.EQ.11.OR.ICALLB.EQ.12)CALL
FILLIN(M,KKB-MOVEB,J,OK)
 IF(OK.NE.1)NOTIN=1
 endif
 ELSE
 IOUT=333
```

APPENDIX

```
 321 ENDIF

 endif
 MANY=1+MANY
 IF(MANY.GE.2)IOUT=333
 IF(IOUT.EQ.333)GOTO333
 IF(NOTIN.EQ.1.AND.FIRST.LT.2)GOTO7
 GOTO111
 333 L3=INFO(0,42,M)
 CALL FILLIN(M,L3,J,OK)
 111 RETURN
 END

 subroutine reserv(j,m,kkf,again)
 INCLUDE'.F.FILES/COM'
 end=0
 again=1+again
 goto111
 do 1 I=kkf,info(0,42,m)
 if(end.eq.999)goto1
 jj=info(i,40,m)
 if(jj.gt.0)then
 if(info(jj,25,1).eq.m)then
 do 2 kb=kkf-1,kkf-info(jj,12,m),-1
 if(info(kb,40,m).ne.0)kkb=kb
 if(kkb.gt.0)goto3
2 continue
3 if(kkb.eq.0)call mvjb(jj,kkf,i,m,end)
 if(kkb.gt.0)then
 mm=kkf-kkb
 if(info(0,43,m).gt.1)then
 nomor=0
 do 4 kk=1,info(0,43,m)
 if(nomor.eq.0)then
 if(info(jj,12,m).le.info(kk,42,m))then
 kkf=info(kk,43,m)+info(jj,12,m)
 call mvjb(jj,kkf,i,m,end)
 Nomor=1
 endif
 endif
4 continue
 endif
 endif
 endif
 endif
1 continue
111 return
 end

 subroutine mvjb(jj,kkf,i,m,end)
 INCLUDE'.F.FILES/COM'
 ist=kkf-info(jj,12,m)
 call remove(jj,i,m)
 call fillin(m,ist,jj,ok)
 end=999
 return
 end
 subroutine remove(jj,i,m)
 INCLUDE'.F.FILES/COM'
 do 1 k=i,i+info(jj,12,m)
 info(k,40,m)=0
1 continue
 return
 end

C--------------------------------------------
---------------------------
 SUBROUTINE GATHER1(IFROM,ITO,MOVE,ISTEP,M)
C--------------------------------------------
---------------------------
CBROUT: (GATHER&INSERT1)MOVE BLOCK
FORWARD(ISTEP=-1)/BACKWARD(1) START FROM
POINT IFROM TO POINT ITO -> EQUAL MOVE AT
MACHINE M
 INCLUDE'.F.FILES/COM'
cccccccc print*,' Gather1 at machine=',m,
info(0,1,0)
cccccccc
write(6,*)(info(jd,40,m),jd=info(0,1,0),ist1+
info(0,1,0))
 DO 2 N=IFROM,ITO,ISTEP
 IF(INFO(N,40,M).GT.0)THEN
 INFO(INFO(N,40,M),36,M)=
INFO(INFO(N,40,M),36,M)-ISTEP*MOVE
 INFO(INFO(N,40,M),36,0)=
INFO(INFO(N,40,M),36,0)-ISTEP*MOVE
 INFO(INFO(N,40,M),38,0)=
INFO(INFO(N,40,M),38,0)-ISTEP*MOVE
 ENDIF
 INFO(N-MOVE,40,M)=INFO(N,40,M)
 INFO(N,40,M)=0
 2 CONTINUE
cccccccc print*,'GATHER1 end at machine=',m,
info(0,1,0)
cccccccc
write(6,*)(info(jd,40,m),jd=info(0,1,0),ist1+
info(0,1,0))
 111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE GATHER(M,IST1,OOK,LM1,is1)
C--------------------------------------------
---------------------------
CBROUT: FROM INSERT-REMOVE FIRST GAP & ADD IT
TO NEXT ONE VAR0,26,0
 INCLUDE'.F.FILES/COM'
 CALL GAP(M)
 IF(INFO(0,43,M).EQ.0)GOTO111
C REMOVE FIRST GAP
 L1=INFO(1,43,M)
 L2=L1+INFO(L1,42,M)
 f26=info(0,26,0)
 IF(INFO(L1,42,M).LT.F26.and.
 @ L1.le.info(0,1,0)+f26 )THEN
 IF(INFO(0,43,M).GT.1)THEN
 L3=INFO(2,43,M)
 ELSE
 L3=INFO(0,42,M)
 ENDIF

MOVE=min(info(0,37,0)*INFO(0,26,0),info(11,42
,m))
 CALL GATHER1(L2,L3,MOVE,1,M)
 CALL GAP(M)
 OOK=1234
 ENDIF
 111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE FILLIN(M,IST,J,OK)
C--------------------------------------------
---------------------------
CBROUT: FROM INSERT1. IT PUTS A JOB IN A PLACE
AND RELATED INFO.
 INCLUDE'.F.FILES/COM'
C TIME SCALE
 IF(IST.GT.0)INFO(IST,40,M)=J
CCC print*,'job',J,'start at',ist,'on m/c',m
C STARTING MACHINNIG
 INFO(J,36,M)=MAX(0,IST)
C IF FIRST OPERATION
 IF(M.EQ.INFO(J,25,1)) THEN
C ENTRY TIME
 INFO(J,36,0)=MAX(0,INFO(J,36,M)-
 info(0,36,0))
C ARRIVAL TIME
 INFO(J,38,0)=MAX(0,INFO(J,36,0)-
 info(0,38,0))
 IF(INFO(J,3,0).LT.IST+INFO(J,12,0))THEN
 INFO(0,41,0)=INFO(0,41,0)+1
```

APPENDIX

```
 info(3,0,11)=info(0,41,0)
 INFO(INFO(0,41,0),41,0)=J
 INFO(J,0,0)=3
 ENDIF
 IF(IST.LT.(INFO(J,3,0)-INFO(J,12,0)))THEN
 INFO(0,42,0)=INFO(0,42,0)+1
 info(3,0,10)=info(0,42,0)
 INFO(INFO(0,42,0),42,0)=J
 INFO(J,0,0)=4
 ENDIF
 ENDIF
C TO SET ON TIME SCALE -1 -1 -1 -1
 DO 1 I=IST+1,IST+INFO(J,12,M)-1
CCCC goto12344
CCCC12345 print*,'FFFF','j',j,info(j,12,m)
CCCC include'.F.F'
CCCC
print*,'oldj',info(i,40,m),info(info(i,40,m),
12,0)
CCCC12344 if(info(i,40,m).ne.0)goto12345
 INFO(I,40,M)=-1*J
 1 CONTINUE
 MFUT=INFO(0,42,M)
 MPST=INFO(0,41,M)

IF(MFUT.LT.IST+INFO(J,12,M))INFO(0,42,M)=IST+
INFO(J,12,M)
 IF(MPST.GT.IST)INFO(0,41,M)=IST
 OK=1
 CALL GAP(M)
 RETURN
 END
C--OK---------------------------------------
---------------------------
 SUBROUTINE GAP(M)
C-------------------------------------------
---------------------------
CBROUT:FROM INSERT1 AND GATHER. TO DETERMINE:
WHERE GAPS STARTS=>~,43,M &GAP LONG=>~,42,M&
TOTAL GAPS LONG=>0,40,M
 INCLUDE'.F.FILES/COM'
 I=0
 ITOTGP=0
 ITOTGPS=0
 IC=0
 DO 1 I=INFO(0,1,0),INFO(0,42,M)
 IF(INFO(I,40,M).EQ.0)THEN
 IF(ITOTGP.EQ.0)THEN
 IC=IC+1
 INFO(0,43,M)=IC
C WHERE THE GAP IS STARTED
 INFO(IC,43,M)=I
 ENDIF
 ITOTGPS=ITOTGPS+1
 ITOTGP =ITOTGP +1
 ELSE
 IF(ITOTGP.GT.0)THEN
C HOW LONG IS THE GAP WHICH STARTS AT TIME
IC,43,M
 INFO(INFO(IC,43,M),42,M)=ITOTGP
 ITOTGP=0
 ENDIF
 ENDIF
 1 CONTINUE
 INFO(0,40,M)=ITOTGPS
 RETURN
 END
C-------------------------------------------
-------------------------
 SUBROUTINE DEFLT
C-------------------------------------------
-------------------------
 INCLUDE'.F.FILES/COM'
 open(102,file='DF',status='old')
102 format(I5)
 goto 122
110 print*,'err'
122 read(102,102,end=110) info(0,46,0)
 read(102,102,end=110) info(0,47,0)
 read(102,102,end=110) info(0,36,0)
 read(102,102,end=110) info(0,37,0)
 read(102,102,end=110) info(0,38,0)
 read(102,102,end=110) info(7,0,6)
 read(102,102,end=110) info(8,0,10)
 read(102,102,end=110) info(7,0,2)
 read(102,102,end=110) info(1,0,9)
 read(102,102,end=110) info(7,0,8)
 read(102,102,end=110) info(7,0,9)
 read(102,102,end=110) info(7,0,7)
 close (102)
C make it =0 if you like to see the menu
C return it if open deleted info(0,46,0)=9000
 INFO(1,0,1)=4
 MACHS=INFO(1,0,1)
C 0= SLack according(EXP.DD) J,39,0 1= Slack
according(Orig.DD) j,15,0
C return if no open 102  info(0,47,0)=4
 info(0,45,0)=0
C return if no open 102  INFO(0,36,0)=0
C return if no open 102  INFO(0,38,0)=0
C return if no open 102  INFO(0,37,0)=0
C ALLOWED DELAY OR EARLINES (SEE INSERT)$
 INFO(0,40,0)=1
C C1 MIN ARR(LEAD TIME )
 INFO(0,31,0)=0
C C3 MIN ENTRY(LEAD TIME )
 INFO(0,33,0)=0
C C2 MIN STRT M.
 INFO(0,32,0)=0
C IN INNFFO SUBROUTINE OUTPUT SLICE OR
DISCRETE
 INFO(0,23,0)=10
C RATE OF RECEIVING= ONE JOB EVERY 5 MIN (60/
12=5) OR 12 JOBS PER HOUR
 INFO(2,0,2)=12
 INFO(2,0,4)=.5+(60./INFO(2,0,2))
C DYNAMIC
 info(2,0,5)=0
 INFO(2,0,10)=-1
 INFO(2,0,6)=-1
 INFO(2,0,11)=-1
 INFO(2,0,12)=-1
 INFO(2,0,13)=-1
C ALLOWED MAINQ LENGTH
C return with no open 102
INFO(7,0,6)=60
C EARLY COST/UNIT OF TIME
 INFO(1,0,14)=1
C ONE DAY= MMM HR
 INFO(1,0,11)=1
C ALLOWED TIME BEFORE DUE
 INFO(1,0,12)=0
 INFO(8,0,1)=0
 INFO(8,0,2)=1
 INFO(8,0,4)=0
 INFO(8,0,5)=0
 INFO(8,0,6)=6
 INFO(8,0,7)=0
 INFO(8,0,8)=0
C TAKE CARE OF CUSTOMERS, BUT IF CONTROLLED
ARRIVAL THEN =16
 INFO(8,0,9)=8
C TECHNIQUE
C return with no open 102
INFO(8,0,10)=17
C LOCAL WIP (QUEUE LENGTH AT M/C)
C return with no open 102
INFO(7,0,2)=50
C return with no open 102
INFO(1,0,9)=2500
C ARRANGE MAINQS ACCORDING TO EITHER FRFS(1)
OR RESV (36)
```

APPENDIX

```
C return with no open 102
INFO(7,0,8)=36
C return with no open 102
INFO(7,0,9)=36
 INFO(9,0,12)=20
C SARQ NUMBER IN QUEUE15
C return with no open 102 INFO(7,0,7)=60
C TO SEE QUEUES FREQUENTLY AFTER ARRIVAL.
 INFO(12,0,12)=0
 INFO(7,0,3)=MAX(60,MACHS*INFO(7,0,6))
C GLOBAL WIP(IN THE SHOP)
 INFO(7,0,1)=MAX(200,MACHS*INFO(7,0,2))
 RETURN
 END
C-------------------------------------------
--------------------------=
 SUBROUTINE WIP
C-------------------------------------------
--------------------------=
 INCLUDE '.F.FILES/COM'
 COMMON/NOT/MMM
 MMM=1
 NO1=0
 NO2=0
 NO3=0
 NOS=0
 CALL REDSTR(KLM)
 2 CALL MINPQ
 IF(INFO(0,1,0).GE.INFO(2,0,3))THEN
 INFO(2,0,3)=INFO(2,0,3)+INFO(2,0,1)
 CALL MENU
 CALL QUEUE
 ENDIF
 CALL NXTMAC
C WHEN NEXT ACTION & WHERE NEXT ACTION WHEN &
WHERE
C-------------------------------------------
----------
 CALL NXTACT
 IF(INFO(12,0,12).EQ.28)CALL MENU
C SET PASSED TIMES
C---------------------
 IF(INFO(2,0,8).EQ.0)GOTO5
 CALL TIMPAS(ISMAL)
C CHECK TIME IF IT IS READY FOR NEXT ACTION
C-------------------------------------------
--
 5 CALL NXTDO(JUMP)
 IF(INFO(0,3,0).EQ.1)CALL QUEUE
CC rem
IF(INFO(0,1,0).GE.500.AND.INFO(0,1,0).LT.INFO
(1,0,9)
CC rem @ .AND.NO1.EQ.0)THEN
CC rem NO1=1
CC rem CALL FINAL(LLL)
CC rem ENDIF
CC rem
IF(INFO(0,1,0).GE.1000.AND.INFO(0,1,0).LT.INF
O(1,0,9)
CC rem @ .AND.NO2.EQ.0)THEN
CC rem NO2=1
CC rem CALLFINAL(LLL)
CC rem ENDIF
CC REM
IF(INFO(0,1,0).GE.4000.AND.INFO(0,1,0).LT.INF
O(1,0,9)
CC REM @ .AND.NO3.EQ.0)THEN
CC REM NO3=1
CC REM CALLFINAL(LLL)
CC REM ENDIF

IF(INFO(0,1,0).GE.INFO(1,0,9).AND.NOS.EQ.0)TH
EN
 NOS=1
 CALLFINAL(LLL)
 ENDIF
 GOTO2
 112 RETURN
 END
C-------------------------------------------
--------------------------=
 SUBROUTINE SARQ
C-------------------------------------------
--------------------------=
CBROUT: FORM SARQ (SHOP ARRIVAL QUEUE). (FROM
NXTDO)
 INCLUDE '.F.FILES/COM'
 COMMON/P/PPASSA,PPASSE,PPASSM
 NU=INFO(0,15,1)
 CALLMOVE(NU,15)
 INTO=0
 2 DO 1 J=1,JOBS
 IG1=0

IF(INFO(J,0,0).EQ.8.OR.INFO(J,0,0).EQ.3.OR.IN
FO(J,0,0).EQ.4)IG1=1
 IF(IG1.NE.1)GOTO1

IF(INFO(1,16,ABS(INFO(J,25,1))).EQ.0.AND.INFO
(0,15,1).EQ.0)GOTO124
 IF(INFO(0,15,1).GE.INFO(7,0,7).OR.
 @ INFO(7,0,4).GE.INFO(7,0,3))GOTO5
124
IF(INFO(1,16,ABS(INFO(J,25,1))).EQ.0.OR.INFO(
0,15,1).EQ.0)THEN
 IPASS=0

IF((INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6).A
ND.
 @
(info(j,38,0).le.info(10,17,info(j,25,1))+INF
O(0,1,0)))IPASS=17

IF((INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6).A
ND.INFO(J,0,0).
 @
EQ.3.AND.INFO(0,16,ABS(INFO(J,25,1))).LE.INFO
(7,0,6))IPASS=17

IF(INFO(8,0,10).NE.17.AND.INFO(J,29,0).LE.INF
O(0,1,0))IPASS=29

IF(INFO(8,0,10).EQ.17.AND.INFO(1,16,ABS(INFO(
J,25,1))).EQ.0)
 @ IPASS=17
 8010
IF(IG.EQ.999.OR.IPASS.EQ.17.OR.IPASS.EQ.29.OR
.INFO(0,15,1).EQ
 @ .0.
OR.INFO(1,16,ABS(INFO(J,25,1))).EQ.0)THEN
 INFO(J,14,1)=INFO(0,1,0)
 INFO(J,14,7)=INFO(0,1,0)
C ARRIVAL TIME TO SARQ
 INFO(J,16,0)=INFO(0,1,0)
 INFO(0,15,1)=INFO(0,15,1)+1
 INFO(0,15,7)=INFO(0,15,7)+1
C TEMP
 INFO(INFO(0,15,1),15,1)=INFO(J,1,0)
C ALL
 INFO(INFO(0,15,7),15,7)=INFO(J,1,0)
C TOTAL N# OF ARRIVAL
 INFO(5,0,14)=INFO(5,0,14)+1
C CHANGE STATE OF THE JOB
 INFO(J,0,0)=1
 INTO=1
 ENDIF
 ENDIF
 1 CONTINUE
 5 IF(INFO(12,0,12).EQ.15)CALLQUEUE
 PPASSA=PPASSA+1
 RETURN
 END
```

APPENDIX

```
C-------------------------------------------
--------------------------
 SUBROUTINE MAINQ(JJ)
C-------------------------------------------
--------------------------
CBROUT TAKE JOBS FROM SARQ&PUT IT IN SEPARATED
QUEUES ACCORDING FIRST MC(NEWMNQ)
 INCLUDE '.F.FILES/COM'
C NUMBER OF JOBS IN SARQ
 NSARQ=INFO(0,15,1)
 DO 1 JOB=1,NSARQ
C JOB NUMBER IN SARQ
 JSARQ=INFO(JOB,15,1)
C FIRSTPROCES ON MACHINE JATM.
 JATM=INFO(JSARQ,25,1)
C TO SET POINTER TO NEXT PROCESS
 IF(INFO(JSARQ,24,0).EQ.0)INFO(JSARQ,24,0)=1
C CURRENT MACHINE
 INFO(JSARQ,8,0)=JATM
 IF(JATM.LE.0)GOTO1
C COUNTER MAIN
 INFO(0,JJ,JATM)=INFO(0,JJ,JATM)+1
 JMAINQ=INFO(0,JJ,JATM)
C PUT IN MAINQ
 INFO(7,0,4)=INFO(7,0,4)+1
 INFO(JMAINQ,JJ,JATM)=INFO(JOB,15,1)
 INFO(JOB,15,1)=0
 INFO(0,15,1)=INFO(0,15,1)-1
 1 CONTINUE
 CALLMOVE(NSARQ,15)
111 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE NEWMNQ
C-------------------------------------------
--------------------------
CBROUT QUEUES ACCORDING FIRST MC. QUEUE UPSIDE
DOWN. NEW ARRIVAL (NXTDO)
 INCLUDE '.F.FILES/COM'
 COMMON /P/PPASSA,PPASSE,PPASSM
200 ITOTAL=0
 INFO(0,34,0)=0
 CALL MAINQ(34)
 IQUE16=0
 DO 1 JATM=1,MACHS
ccccccc print*,'A16',jatm,'-
',(info(k,16,jatm),k=1,info(0,16,jatm))
ccccccc print*,'A34',jatm,'-
',(info(k,34,jatm),k=1,info(0,34,jatm))
C PUT 16 IN TAIL OF 34
 CALL UPDOWN(16)
ccccccc print*,'K16',jatm,'-
',(info(k,16,jatm),k=1,info(0,16,jatm))
ccccccc print*,'K34',jatm,'-
',(info(k,34,jatm),k=1,info(0,34,jatm))

IF(INFO(0,16,JATM).GT.IQUE16)IQUE16=INFO(0,16
,JATM)
 DO 2 K=1,INFO(0,16,JATM)

INFO(INFO(0,34,JATM)+K,34,JATM)=INFO(K,16,JAT
M)
 2 CONTINUE

INFO(0,34,JATM)=INFO(0,34,JATM)+INFO(0,16,JAT
M)
ccccccc print*,'116',jatm,'-
',(info(k,16,jatm),k=1,info(0,16,jatm))
ccccccc print*,'134',jatm,'-
',(info(k,34,jatm),k=1,info(0,34,jatm))
C PUT 34 IN 16
 DO 3 K1=0,info(0,34,jatm)
 INFO(K1,16,JATM)=INFO(K1,34,JATM)
ccccccc
print*,info(k1,34,jatm),info(k1,16,jatm)
 INFO(K1,34,JATM)=0
 3 CONTINUE
ccccccc print*,'M16',jatm,'-
',(info(k,16,jatm),k=1,info(0,16,jatm))
ccccccc print*,'M34',jatm,'-
',(info(k,34,jatm),k=1,info(0,34,jatm))
 1 CONTINUE
 CALL UPDOWN(16)
 INFO(7,0,10)=IQUE16
 PPASSM=1
111 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE LDING(JATM)
C-------------------------------------------
--------------------------
CBROUT (LDM17) TO LOAD A JOB ON MACHINE.
 INCLUDE'.F.FILES/COM'
 IN=INFO(0,17,JATM)
 MI=INFO(0,1,0)
 INFO(0,5,JATM)=0
C TIME WHEN MACHINING START
 IF(INFO(IN,24,0).LT.2)INFO(IN,18,0)=MI
 INFO(IN,11,JATM)=MI
C START TIME OF PROCESS
 INFO(6,17,JATM)=MI
C IDLENESS
 INFO(0,11,JATM)=INFO(0,11,JATM)+(MI-
INFO(15,17,JATM))
C PUT PROCESSING TIME

INFO(9,17,JATM)=INFO(IN,26,ABS(INFO(IN,24,0))
)
C EXPECTED FINISHING TIME
 INFO(5,17,JATM)=MI+INFO(9,17,JATM)
C MACHINQ
 INFO(0,15,3)=INFO(0,15,3)+1
 INFO(0,15,9)=INFO(0,15,9)+1
 INFO(INFO(0,15,9),15,9)=JATM
 INFO(INFO(0,15,3),15,3)=IN
 INFO(INFO(0,15,3),14,3)=MI
C PLACE OF JOB
 IF(IN.EQ.0)CALLERROR(101)
 INFO(IN,8,JATM)=2
C REMAINING TIME
 INFO(10,17,JATM)=INFO(9,17,JATM)
C MACHQ
 INFO(0,18,JATM)=INFO(0,18,JATM)+1
 INFO(INFO(0,18,JATM),18,JATM)=IN
C START TIME
 INFO(INFO(0,18,JATM),19,JATM)=MI
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE MQTOQ(JATM)
C-------------------------------------------
--------------------------
CBROUT:TAKE JOB FROM MAINQ TO MINPQ AT MC
JATM(JOB WILL BE AT 4TH PLACE)(MINPQ)
 INCLUDE '.F.FILES/COM'
200 IF(INFO(0,16,JATM).EQ.0)CALLERROR(106)

IF(INFO(INFO(0,16,JATM),16,JATM).EQ.0)CALLERR
OR(107)
 IF(INFO(0,16,JATM).EQ.0)CALLERROR(106)

IF(INFO(INFO(0,16,JATM),16,JATM).EQ.0)CALLERR
OR(107)

INFO(4,17,JATM)=INFO(INFO(0,16,JATM),16,JATM)
 INFO(INFO(0,16,JATM),16,JATM)=0
C COUNTER AT 16
 INFO(0,16,JATM)=INFO(0,16,JATM)-1
C TIME OF ENTRY TO SHOP
```

```
 INFO(INFO(4,17,JATM),17,0)=INFO(0,1,0)
 INFO(7,0,5)=INFO(7,0,5)+1
 INFO(7,0,4)=INFO(7,0,4)-1
C THIS JOB IS ENTRED NOW TO THE SHOP.
 INFO(INFO(4,17,JATM),0,0)=2
C COUNTER AT 17
 INFO(8,17,JATM)=INFO(8,17,JATM)+1
C WHERE IS THE JOB X=8 (1 MEANS IN QUEUE)
 IF(INFO(4,17,JATM).EQ.0)CALLERROR(102)
 IF(INFO(12,0,12).EQ.40)
WRITE(6,2)INFO(4,17,JATM),JATM
 2 FORMAT('IF =0 THEN SOME THING WRONG
=',I3,X,'M/C',I2)
 INFO(INFO(4,17,JATM),8,JATM)=1
C QUEUE LENGTH (TIME) FOR THOSE BATCH CAME FROM
MAINQ ONLY

INFO(7,17,JATM)=INFO(7,17,JATM)+INFO(INFO(4,1
7,JATM),12,JATM)
C START QUEUEING TIME
 INFO(14,17,JATM)=INFO(0,1,0)
C MACHGOQ
 INFO(0,15,2)=INFO(0,15,2)+1
 INFO(0,15,8)=INFO(0,15,8)+1
 INFO(INFO(0,15,2),15,2)=INFO(4,17,JATM)
 INFO(INFO(0,15,8),15,8)=JATM
 INFO(INFO(0,15,2),14,2)=INFO(0,1,0)
C PROCESSING TIME OF NEXT MACHINES INCREASED
 DO 1 I=1,INFO(ABS(INFO(4,17,JATM)),25,0)

INFO(17,17,ABS(INFO(ABS(INFO(4,17,JATM)),25,I
)))=

@INFO(17,17,INFO(INFO(4,17,JATM),25,I))+INFO(
INFO(4,17,JATM),26,I)
1 CONTINUE
111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE PRETOQ(JATM)
C--------------------------------------------
---------------------------
CBROUT TAKE FROM ALL MACHINES' OUTPUT BUFFER
TO MINPQ. (MINPQ)
 INCLUDE '.F.FILES/COM'
 INFO(4,17,JATM)=INFO(1,23,JATM)

INFO(7,17,JATM)=INFO(7,17,JATM)+INFO(INFO(4,1
7,JATM),26,INFO(INFO
 @(4,17,JATM),24,0))
 INFO(8,17,JATM)=INFO(8,17,JATM)+1
 INFO(1,23,JATM)=0
C TIME
 INFO(14,17,JATM)=INFO(1,24,JATM)
 INFO(1,24,JATM)=0
 CALLMOVE(JATM,23)
 INFO(0,23,JATM)=INFO(0,23,JATM)-1
 INFO(16,17,JATM)=INFO(16,17,JATM)-
INFO(INFO(4,17,JATM),26,INFO(
 @INFO(4,17,JATM),24,0))
 RETURN
 END
C--OK----------------------------------------
---------------------------
 SUBROUTINE NXTMAC
C--------------------------------------------
---------------------------
 INCLUDE '.F.FILES/COM'
CBROUT:FIND MC WHICH HAS SMALLEST REM PROC
T.(10,17,MAC) MCS (0,17,IZ)(WIP)
 IF(MACHS.EQ.1)MAC=1
 IF(MACHS.EQ.1)GOTO111
 I8=MACHS
 MAC=MACHS+1
 5 MAC=MAC-1
 IF(MAC.EQ.0)MAC=999
 IF(MAC.EQ.999)GOTO112
 IF(INFO(0,17,MAC).EQ.0)GOTO5
 IF(INFO(10,17,MAC).LE.0)GOTO111
 1 IPASSED=0
 I8=I8-1
 IF(INFO(0,17,I8).GT.0)THEN
 IF(INFO(10,17,I8).LT.INFO(10,17,MAC))THEN
 MAC=I8
 IPASSED=1
 ENDIF
 ENDIF
 IF(I8.LE.1)GOTO111
 IF(IPASSED.EQ.1.OR.I8.GT.1)GOTO1
C NEXT MACHINE TO B RELEASED IS
MAC=INFO(1,0,6)
C-------------------------------------------
-------
 111 INFO(1,0,6)=MAC
C NEXT REALESING TIME AT MAC MACHINE IS
INFO(2,0,6)
C-------------------------------------------
----------
 INFO(2,0,6)=MAX(0,INFO(10,17,MAC))
 IF(INFO(0,17,MAC).EQ.0)INFO(2,0,6)=-1
112 RETURN
 END
C--OK---------------------------------------
-------------------------
 SUBROUTINE NXTACT
C-------------------------------------------
-------------------------
C BROUT: TO SET NEXT ACTION:RECEIVING A NEW
ORDER, ARRIVAL TO STORE, ENTRY TO SHOP,
RELEASE FROM MACHINE, REPAIR A MACHINE.
 INCLUDE'.F.FILES/COM'
 COMMON /P/PPASSA,PPASSE,PPASSM
 INFO(0,7,0)=INFO(0,7,0)+1
CXX RECEIVE
 IF(INFO(8,0,2).EQ.0)GOTO2

IF(INFO(1,0,9).GT.0.AND.INFO(1,0,9).LE.INFO(0
,1,0))GOTO2
 INFO(2,0,8)=INFO(2,0,5)
 INFO(1,0,8)=5
 INFO(2,0,9)=8
CXX SARQ ARRIVAL
2 IF(INFO(0,15,1).GE.INFO(7,0,7))GOTO2344
 IF(INFO(7,0,4).GE.INFO(7,0,3))GOTO2344
 IF(INFO(5,0,14).GE.JOBS)GOTO2344
 IF(PPASSA.GT.0)GOTO2344
 ITT=99999
 DO 1 I=1,JOBS

IF(INFO(I,0,0).EQ.8.OR.INFO(I,0,0).EQ.3.OR.IN
FO(I,0,0).EQ.4)THEN

IF((INFO(7,0,4).LT.INFO(7,0,3)).AND.INFO(0,15
,1).EQ.0)ITT=1
 IF(INFO(8,0,10).EQ.17)THEN
 KK=MAX(0,INFO(I,38,0))
 ELSE
 KK=MAX(0,INFO(I,29,0))
 ENDIF
 IF(KK.LT.ITT)ITT=MAX(0,KK)
 ENDIF
1 CONTINUE
 INFO(2,0,12)=MAX(1,ITT)
1000 CALLACTION(12,4)
C MAIN ARRIVAL (STORE)
2344 IF(PPASSM.GT.0)GOTO1235
 if(info(7,0,4).ge.info(7,0,3))goto1235
 KF=0
 IF(INFO(0,15,1).LT.1)GOTO1235
 DO 5 IM=1,INFO(0,15,1)
 IF(
```

APPENDIX

```
INFO(1,16,ABS(INFO(IM,25,1))).EQ.C.AND.INFO(0
,16,
 @
ABS(INFO(INFO(IM,15,1),25,1))).LT.INFO(7,0,6)
)INFO(2,0,13)=0
 5 CONTINUE
 IF(KF.EQ.1)CALLACTION(13,2)
CXX RELEASE
1235 CALLACTION(6,16)
 RETURN
 END
C---OK----------------------------------------
---------------------
 SUBROUTINE ACTION(IWHERE,IWHAT)
C---------------------------------------------
---------------------
CBROUT (NXTACT)SET VALUES OF 2,0,9(IT COULD BE
MORE THAN ONE EVENT AT ONCE)
CBROUT 2,0,8(TIME NEED TO HAPEN) &
1,0,8(WHICH5,6,11,10,12)
 INCLUDE'.F.FILES/COM'
 IF(INFO(2,0,IWHERE).GE.0 ) THEN

IF(INFO(2,0,IWHERE).EQ.INFO(2,0,8))INFO(2,0,9
)=INFO(2,0,9)+IWHAT
 IF(INFO(2,0,IWHERE).LT.INFO(2,0,8))THEN
 INFO(1,0,8)=IWHERE
 INFO(2,0,8)=INFO(2,0,IWHERE)
 INFO(2,0,9)=IWHAT
 ENDIF
 ENDIF
 RETURN
 END
C--OK-----------------------------------------
---------------------
 SUBROUTINE NXTDO(JUMP)
C---------------------------------------------
---------------------
CBROUT (WIP)CALLCORROSPONDING
SUBROUTINE:REPAIR,RELEASE,ENTRY,ARRIVAL(S &
M)
C AND RECEIVING. ALSO TO DETERMINE NEXT JOBS
TO ARRIVE ( HOW MANY)
 INCLUDE'.F.FILES/COM'
C RELEASE
 IF(INFO(2,0,9).GE.16)THEN
 CALLRELEAS
 CALLMINPQ
 INFO(2,0,9)=INFO(2,0,9)-16
 ENDIF
C RECEIVE
 KAKA=0
 IF(INFO(8,0,2).EQ.0.OR.

@(INFO(1,0,9).GT.0.AND.INFO(1,0,9).LE.INFO(0,
1,0)))GOTO11
 IF(INFO(2,0,9).GE.8.OR.JOBS.EQ.0)THEN
 CALLREDCNT
 CALLSARQ
 CALLNEWMNQ
C NOTE NEXT TIME TO RECEIVE A NUMBER OF JOBS
 INFO(2,0,5)=info(0,0,0)
909 FORMAT(I5)
 KAKA=1
 INFO(2,0,9)=INFO(2,0,9)-8
 ENDIF
C SARQ ARRIVAL AND PREPARE TO ENTER SHOP
 11 KOKO=0
 IF(INFO(2,0,9).GE.4)THEN
 CALLSARQ
 CALLNEWMNQ
 KOKO=1
 INFO(2,0,12)=-1
 INFO(2,0,9)=INFO(2,0,9)-4
 ENDIF
C MAIN ARRIVAL
```

```
 KIKO=0
 IF(INFO(2,0,9).GE.2)THEN
 IF(KOKO.EQ.0)CALLSARQ
 CALLNEWMNQ
 INFO(2,0,13)=-1
 INFO(2,0,9)=INFO(2,0,9)-2
 KIKO=1
 ENDIF
 info(2,0,14)=10000.*info(3,0,14)/
max(1,info(0,1,0))
 INFO(7,0,14)=10000.*info(4,0,14)/
(MAX(1,INFO(0,1,0)))
 CALL STDV
C SHOW RELEASE MESSAGE
 f32=info(3,0,2)/1000.
 f312=info(3,0,12)/1000.
 f43=info(4,0,3)/1000.
 f42=info(4,0,2)/1000.
 f52=info(5,0,2)/1000.
 f714=info(7,0,14)/100.
 f62=info(6,0,2)/1000.
 f214=info(2,0,14)/100.
 f53=info(5,0,3)/1000.
 f63=info(6,0,3)/1000.
Call
WRITE(724,109)JOBS,INFO(4,0,14),INFO(0,1,0),I
NFO(3,0,1),f32,f312,
Call
@info(4,0,1),f42,f43,f214,f714,f52,f62,f53,f6
3
CCCCCCCCCCC
WRITE(6,109)info(0,1,0),JOBS,INFO(4,0,14),INF
O(3,0,1),f32,
CCCCCCCCCCC @info(4,0,1),f42,f214,f714
109 FORMAT(
 @'T',I4,' R',I4,' X',I4,' A',I7,' M',F5.1,'
F',I6,' M',
 @F5.1,' R.rt',F4.1,' X.rt',F4.1,' AL',F4.1,'
FLF',F4.1,
 @' Rcv',F4.1,' Cmp',F4.1)

if(Ktime.eq.10.and.info(0,1,0).lt.1+lastT)the
n
 stop
 else
 ktime=ktime+1
 LastT=info(0,1,0)
 endif
 RETURN
 END
C-------ALL
 SUBROUTINE
LDM01(M,JQ0,LK0,NXM0,LQ0,IQU0,IP0,I0,ID,IDU,I
DU0,RSV0)
C---------------------------------------------
---------------------------
CBROUTSET
SLACK(LK0),NXM0,NXTQUE(IQU0),PROCT(IP0)&FIND
MIN DDATE IDU,AT JOB(ID)
 INCLUDE'.F.FILES/COM'
 IF(JQ0.GT.0)THEN
C DELIVERY
 IDU0=INFO(JQ0,3,0)
CCCCC print*,'JQ0',jq0,'
(-250)',info(jq0,25,0),
CCCCC @info(jq0,25,info(jq0,25,0))

INFO(JQ0,39,0)=max((INFO(JQ0,36,abs(INFO(JQ0,
25,INFO(JQ0,25,0))))
 @
+INFO(JQ0,12,abs(INFO(JQ0,25,INFO(JQ0,25,0)))
)),
 @
(info(JQ0,36,abs(info(JQ0,25,1)))+info(JQ0,12
,0)))
 RSV0=INFO(JQ0,36, INFO(JQ0,25,
```

APPENDIX

```
INFO(JQ0,25,0) ) ) +
 @ INFO(JQ0,12, INFO(JQ0,25, INFO(JQ0,25,0) )
)
 if(info(0,45,0).eq.0)goto2
 1 LK0=INFO(JQ0,15,0)-INFO(JQ0,14,0)
 goto3
 2 LK0=(INFO(JQ0,39,0)-info(0,1,0))-
INFO(JQ0,14,0)
3 IF(IDU.GT.IDU0.OR.IDU.EQ.0)THEN
 ID=I0
 IDU=IDU0
 ENDIF
 NXM0=INFO(JQ0,25,(INFO(JQ0,24,0)+1))
 IQU0=-1
 IF(NXM0.GT.0)IQU0=INFO(8,17,NXM0)
 IF(NXM0.GT.0)LQ0=INFO(7,17,NXM0)
 IP0=INFO(JQ0,12,M)
 ENDIF
 111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINELDM02
 @
(I,O0,IO,IP,IP0,IPN,IQ,IQU,ILQ,LQ,LQ0,IQU0,NX
T0,IQL,IQUL)
C--------------------------------------------
---------------------------
C TO FIND MIN PROC.T JOB, MIN QUEUE LENGTH NEXT
MC,
 IF(IP.GT.IP0.OR.IP.EQ.-1)IP=IP0
 IF(IP.GT.IP0.OR.IPN.EQ.-1)IPN=I
 IF(NXT0.EQ.0)THEN
 O0=I
 IO=I
 ENDIF
C SEE QUEUES OF NEXT MACHINE THAT JOBS IN
CURRENT QUEUE ARE GOING TO.
 IF(NXT0.GT.0)THEN
C SHORTEST NUMBER OF JOBS
 IF(IQUL.LT.IQU0.OR.IQUL.EQ.99999)IQUL=IQU0
 IF(IQUL.LT.IQU0.OR.IQL.EQ.-1)IQL=I
C SHORTEST IN TIME
 IF(LQ.LT.LQ0.OR.LQ.EQ.99999)LQ=LQ0
 IF(LQ.LT.LQ0.OR.ILQ.EQ.-1)ILQ=I
C LONGEST NUMBER OF JOBS
 IF(IQU.GT.IQU0.OR.IQU.EQ.-1)IQU=IQU0
 IF(IQU.GT.IQU0.OR.IQ.EQ.-1)IQ=I
 ENDIF
 111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE LDM17(JATM)
C--------------------------------------------
---------------------------
C FROM MINPQ
 INCLUDE'.F.FILES/COM'
200 M=JATM

IF(INFO(0,17,M).EQ.0.AND.INFO(8,17,M).EQ.1)GO
TO1

IF(INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6)THE
N
 JQ5=0

if(info(8,0,10).eq.17.or.info(8,0,9).eq.6)CAL
LLDM0(INFO(1,17,M)
 @
,INFO(2,17,M),INFO(3,17,M),INFO(4,17,M),JQ5,K
GOTO,MOVE,M,17)
 IF(KGOTO.EQ.1)GOTO1
 IF(KGOTO.EQ.5)GOTO5
 IF(MOVE.EQ.0)GOTO1
 LLL=INFO(1,17,M)
 INFO(1,17,M)=INFO(MOVE,17,M)
 INFO(MOVE,17,M)=LLL
 ENDIF
1 IF(INFO(1,17,JATM).EQ.0)CALLERROR(103)
 INFO(0,17,JATM)=INFO(1,17,JATM)
 INFO(1,17,JATM)=0
 INFO(11,17,JATM)=0
 CALLLDING(JATM)
 GOTO111
5 WRITE(6,*)'SOMETHING WRONG: NUMBER OF JOBS
IN QUEUE IS 0'
111 RETURN
 END
C--------------------------------------------
---------------------------
 SUBROUTINE
LDM0(JQ1,JQ2,JQ3,JQ4,JQ5,KGOTO,MOVE,M,IQU17)
C--------------------------------------------
---------------------------
C FROM LDM160 AND LDM17
 INCLUDE '.F.FILES/COM'
 IGOTO=0
 LK2=0
 LK3=0
 LK4=0
 LK5=0
 MOVE=0
 if(info(0,45,0).eq.0)goto2
 LK1=INFO(JQ1,15,0)-INFO(JQ1,14,0)
 goto3
 2 LK1=(INFO(JQ1,39,0)-info(0,1,0))-
INFO(JQ1,14,0)
C DELIVERY
 3 IDUE1=INFO(JQ1,3,0)
 IDUE=IDUE1
 ID=1

IF(JQ1.GT.0)NXTMC1=INFO(JQ1,25,INFO(JQ1,24,0)
+1)
 IQUE1=-1
 LQ1=99999
 IF(NXTMAC1.GT.0)IQUE1=INFO(8,17,NXTMC1)
 IF(NXTMAC1.GT.0)LQ1=INFO(7,17,NXTMC1)
 IP1=INFO(JQ1,12,M)
 IF(IQU17.EQ.16)GOTO16
17 IGOTO=INFO(8,17,M)
 IF(INFO(0,17,M).EQ.0)IGOTO=IGOTO+1
 GOTO 18
16 IGOTO=6
 IF(JQ5.EQ.0)IGOTO=5
 IF(JQ4.EQ.0)IGOTO=4
 IF(JQ3.EQ.0)IGOTO=3
 IF(JQ2.EQ.0)IGOTO=2
 IF(JQ1.EQ.0)IGOTO=1
18 GOTO(5,1,20,30,40,50),IGOTO
50
CALLLDM01(M,JQ5,LK5,NXTMC5,LQ5,IQUE5,IP5,5,ID
,IDUE,IDUE5,RESV5)
40
CALLLDM01(M,JQ4,LK4,NXTMC4,LQ4,IQUE4,IP4,4,ID
,IDUE,IDUE4,RESV4)
30
CALLLDM01(M,JQ3,LK3,NXTMC3,LQ3,IQUE3,IP3,3,ID
,IDUE,IDUE3,RESV3)
20
CALLLDM01(M,JQ2,LK2,NXTMC2,LQ2,IQUE2,IP2,2,ID
,IDUE,IDUE2,RESV2)
 IOUT=0
 IT=0
 IQ=-1
 IP=-1
 LQ=99999
 ILQ=-1
 IPN=-1
 IQUE=-1
 IQUL=99999
```

APPENDIX

```
 MOVE1=0
 MOVE2=0
 MOVE3=0
 MOVE4=0
 MOVE5=0
 MOVE23=0
 MOVE45=0
 IF( (JQ2.GT.0.AND.LK1.LT.IP2).AND.
 @
(JQ3.EQ.0.OR.(JQ3.GT.0.AND.LK1.LT.IP3)).AND.
 @
(JQ4.EQ.0.OR.(JQ4.GT.0.AND.LK1.LT.IP4)).AND.
 @ (JQ5.EQ.0.OR.(JQ5.GT.0.AND.LK1.LT.IP5)) )
MOVE1=1
 IF(JQ2.GT.0.AND.LK1.GE.IP2) MOVE2=2
 IF(JQ3.GT.0.AND.(
 @((LK1.GE.IP3).AND.(LK2.GE.IP1+IP3)).OR.
 @ ((LK1.GE.IP3+IP2).AND.(LK2.GE.IP3)) ) )
MOVE3=3
C4-1-2-3-5 or 4-1-3-2-5 or 4-2-1-3-5 4-3-1-2-
5 or 4-2-3-1-5 or 4-3-2-1-5
 IF(JQ4.GT.0.AND.(
 @(
LK1.GE.IP4.AND.((LK2.GE.IP4+IP1.AND.LK3.GE.IP
4+IP1+IP2).OR.
 @ (LK3.GE.IP4+IP1.AND.LK2.GE.IP4+IP1+IP3))
).OR.
 @
(LK1.GE.IP4+IP2.AND.LK3.GE.IP4+IP2+IP1.AND.LK
2.GE.IP4).OR.
 @
(LK2.GE.IP4+IP3+IP1.AND.LK1.GE.IP4+IP3.AND.LK
3.GE.IP4).OR.
 @( LK1.GE.IP4+IP2+IP3.AND.
((LK2.GE.IP4.AND.LK3.GE.IP2+IP4).OR.
 @ (LK2.GE.IP4+IP3.AND.LK3.GE.IP4)) ) ) )
MOVE4=4
 MM1=0
C5-3-4-2-1/5-4-3-2-1/5-2-4-3-1/5-4-2-3-1/5-3-
2-4-1/5-2-3-4-1/5-3-4-1-2/5-4-3-1-2
C5-1-4-3-2/5-4-1-3-2/5-3-1-4-2/5-1-3-4-2/5-1-
4-2-3/5-4-1-2-3/5-2-4-1-3/5-4-1-1-3
C5-1-2-4-3/5-2-1-4-3/5-3-1-2-4/5-1-3-2-4/5-2-
1-3-4/5-1-2-3-4/5-3-2-1-4/5-2-3-1-4
 IF(JQ5.GT.0.AND.(
(LK1.GE.IP2+IP3+IP4+IP5.AND.

@(LK2.GE.IP3+IP4+IP5.AND.((LK3.GE.IP4+IP5.AND
.LK4.GE.IP5).OR.
 @ (LK4.GE.IP3+IP5.AND.LK3.GE.IP5)) ).OR.

@(LK3.GE.IP2+IP4+IP5.AND.((LK2.GE.IP4+IP5.AND
.LK4.GE.IP5).OR.
 @ (LK4.GE.IP2+IP5.AND.LK2.GE.IP5)) ).OR.

@(LK4.GE.IP2+IP3+IP5.AND.((LK2.GE.IP3+IP5.AND
.LK3.GE.IP5).OR.
 @ (LK3.GE.IP2+IP5.AND.LK2.GE.IP5)) ) ).OR.
 @(LK2.GE.IP1+IP3+IP4+IP5.AND.

@(LK1.GE.IP3+IP4+IP5.AND.((LK3.GE.IP4+IP5.AND
.LK4.GE.IP5).OR.
 @ (LK4.GE.IP3+IP5.AND.LK3.GE.IP5)) ).OR.

@(LK3.GE.IP1+IP4+IP5.AND.((LK1.GE.IP4+IP5.AND
.LK4.GE.IP5).OR.
 @ (LK4.GE.IP1+IP5.AND.LK1.GE.IP5)) ).OR.

@(LK4.GE.IP1+IP3+IP5.AND.((LK1.GE.IP3+IP5.AND
.LK3.GE.IP5).OR.
 @ (LK3.GE.IP1+IP5.AND.LK1.GE.IP5)) ) ) )
)MM1=1
 IF(JQ5.GT.0.AND.(
(LK3.GE.IP2+IP1+IP4+IP5.AND.
 @(LK2.GE.IP1+IP4+IP5.AND.
((LK1.GE.IP4+IP5.AND.LK4.GE.IP5).OR.
 @ (LK4.GE.IP1+IP5.AND.LK1.GE.IP5)) ).OR.
 @(LK1.GE.IP2+IP4+IP5.AND.
((LK2.GE.IP4+IP5.AND.LK4.GE.IP5).OR.
 @ (LK4.GE.IP2+IP5.AND.LK2.GE.IP5)) ).OR.
 @(LK4.GE.IP2+IP1+IP5.AND.
((LK2.GE.IP1+IP5.AND.LK1.GE.IP5).OR.
 @ (LK1.GE.IP2+IP5.AND.LK2.GE.IP5)) ) ).OR.
 @(LK4.GE.IP2+IP3+IP1+IP5.AND.
 @(LK2.GE.IP3+IP1+IP5.AND.
((LK3.GE.IP1+IP5.AND.LK1.GE.IP5).OR.
 @ (LK1.GE.IP3+IP5.AND.LK3.GE.IP5)) ).OR.
 @(LK3.GE.IP2+IP1+IP5.AND.
((LK2.GE.IP1+IP5.AND.LK1.GE.IP5).OR.
 @ (LK1.GE.IP2+IP5.AND.LK2.GE.IP5)) ).OR.
 @(LK1.GE.IP2+IP3+IP5.AND.
((LK2.GE.IP3+IP5.AND.LK3.GE.IP5).OR.
 @ (LK3.GE.IP2+IP5.AND.LK2.GE.IP5)) ) ).OR.
MM1.EQ.1) )MOVE5=5

IF(MOVE1.EQ.0.AND.MOVE2.EQ.0.AND.MOVE3.EQ.0.A
ND.
 @ MOVE4.EQ.0.AND.MOVE5.EQ.0)MOVE1=1
 IF(MOVE1.EQ.1)GOTO1
 IF(IGOTO.LE.2)GOTO1
 CALLLDM02

@(1,OUT1,IOUT,IP,IP1,IPN,IQ,IQUE,ILQ,LQ,LQ1,I
QUE1,NXTMC1,IQL,IQUL)
 IF(IGOTO.GE.3.AND.MOVE2.EQ.2)CALLLDM02

@(2,OUT2,IOUT,IP,IP2,IPN,IQ,IQUE,ILQ,LQ,LQ2,I
QUE2,NXTMC2,IQL,IQUL)
 IF(IGOTO.GE.4.AND.MOVE3.EQ.3)CALLLDM02

@(3,OUT3,IOUT,IP,IP3,IPN,IQ,IQUE,ILQ,LQ,LQ3,I
QUE3,NXTMC3,IQL,IQUL)
 IF(IGOTO.GE.5.AND.MOVE4.EQ.4)CALLLDM02

@(4,OUT4,IOUT,IP,IP4,IPN,IQ,IQUE,ILQ,LQ,LQ4,I
QUE4,NXTMC4,IQL,IQUL)
 IF(IGOTO.GE.6.AND.MOVE5.EQ.5)CALLLDM02

@(5,OUT5,IOUT,IP,IP5,IPN,IQ,IQUE,ILQ,LQ,LQ5,I
QUE5,NXTMC5,IQL,IQUL)
 MOVE=0
C TWO IN QUEUE
 IF(IQUE.LE.3.AND.IQUE.GE.0)MOVEK=IQ
C IF THERE is A JOB TO BE OUT
 IF(IOUT.GT.0) IT=IOUT
C AND NEXT MACHINE HAS MORE THAN 3 JOBS
 IF(IQUE.GT.3.AND.IT.GT.0)MOVEK=IT
C IF THERE IS NOT OUT JOBS THEN SEE THE
SHORTEST JOB
 IF(IQUE.GT.3.AND.IT.EQ.0)MOVEK=IPN
 IF(MOVE2.GT.0.AND.MOVE3.GT.0)THEN
ccc IF(LK2.LE.LK3)THEN
ccc MOVE23=2
ccc LK23=LK2
ccc ELSE
ccc MOVE23=3
ccc LK23=LK3
ccc ENDIF
 IF(LQ3-INFO(JQ3,12,M).LT.LQ2-INFO(JQ2,12,M)
)THEN
 MOVE23=3
 LQ23=LQ3
 LK23=LK3
 JQ23=JQ3
 ELSE
 MOVE23=2
 LQ23=LQ2
 LK23=LK2
 JQ23=JQ2
 ENDIF
 ELSE
 IF(MOVE2.GT.0)THEN
```

APPENDIX

```
 MOVE23=2
 LQ23=LQ2
 LK23=LK2
 JQ23=JQ2
 ELSEIF(MOVE3.GT.0)THEN
 MOVE23=3
 LQ23=LQ3
 LK23=LK3
 JQ23=JQ3
 ENDIF
 ENDIF
 IF(MOVE4.GT.0.AND.MOVE5.GT.0)THEN
 IF(LK4.LE.LK5)THEN
 MOVE45=4
 LK45=LK4
 ELSE
 MOVE45=5
 LK45=LK5
 ENDIF
 IF(LQ5-INFO(JQ5,12,M).LT.LQ4-
INFO(JQ4,12,M))THEN
 MOVE45=5
 LQ45=LQ5
 LK45=LK5
 JQ45=JQ5
 ELSE
 MOVE45=4
 LQ45=LQ4
 LK45=LK4
 JQ45=JQ4
 ENDIF
 ELSE
 IF(MOVE4.GT.0)THEN
 MOVE45=4
 LQ45=LQ4
 LK45=LK4
 JQ45=JQ4
 ELSEIF(MOVE5.GT.0)THEN
 MOVE45=5
 LQ45=LQ5
 LK45=LK5
 JQ45=JQ5
 ENDIF
 ENDIF
 IF(MOVE23.GT.0.AND.MOVE45.GT.0)THEN
 IF(LQ23-INFO(JQ23,12,M).LT.LQ45-
INFO(JQ45,12,M))THEN
 MOVE=move23
 ELSE
 MOVE=MOVE45
 ENDIF
 ELSE
 if(move23.gt.0)move=move23
 if(move45.gt.0)move=move45
 ENDIF
 GOTO111
1 MOVE=1
 KGOTO=1
 GOTO111
5 KGOTO=5
111 RETURN
 END
C--------------------------------------------
--------------------------
 SUBROUTINE LDM16(M)
C--------------------------------------------
--------------------------
 INCLUDE'.F.FILES/COM'
CBROUT IS CALLD FROM MINPQ. IT IS USED IN DSD.
 L=0
 LL=0
200 CALLUPDOWN(16)
1 IF(INFO(0,16,M).GE.2)CALLLDM160(M,0)
2 IF(INFO(0,16,M).GT.6)CALLLDM160(M,5)
300 IF(INFO(0,16,M).GT.11)CALLLDM160(M,10)
500 IF(INFO(0,16,M).GT.16)CALLLDM160(M,15)
600 IF(INFO(0,16,M).GT.21)CALLLDM160(M,20)
700 IF(INFO(0,16,M).GT.26)CALLLDM160(M,25)
800 IF(INFO(0,16,M).GT.31)CALLLDM160(M,30)
900 IF(INFO(0,16,M).GT.36)CALLLDM160(M,35)
510 IF(INFO(0,16,M).GT.41)CALLLDM160(M,40)
520 IF(INFO(0,16,M).GT.46)CALLLDM160(M,45)
530 IF(INFO(0,16,M).GT.51)CALLLDM160(M,50)
540 IF(INFO(0,16,M).GT.56)CALLLDM160(M,55)
550 IF(INFO(0,16,M).GT.61)CALLLDM160(M,60)

 DO 3 I=1,INFO(0,16,M)

IF(I.LE.5.AND.INFO(I,16,M).GT.0)L=L+INFO(INFO
(I,16,M),12,0)

IF(I.LE.10.AND.INFO(I,16,M).GT.0)L2=L2+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.15.AND.INFO(I,16,M).GT.0)L3=L3+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.20.AND.INFO(I,16,M).GT.0)L4=L4+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.25.AND.INFO(I,16,M).GT.0)L5=L5+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.30.AND.INFO(I,16,M).GT.0)L6=L6+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.35.AND.INFO(I,16,M).GT.0)L7=L7+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.40.AND.INFO(I,16,M).GT.0)L8=L8+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.45.AND.INFO(I,16,M).GT.0)L9=L9+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.50.AND.INFO(I,16,M).GT.0)La=La+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.55.AND.INFO(I,16,M).GT.0)Lb=Lb+INFO(I
NFO(I,16,M),12,0)

IF(I.LE.60.AND.INFO(I,16,M).GT.0)Lc=Lc+INFO(I
NFO(I,16,M),12,0)
 3 CONTINUE
 IF(INFO(0,16,M).GT.60)THEN

IF(INFO(INFO(60,16,M),36,0).GE.Lc+INFO(INFO(6
1,16,M),12,M))THEN
 K3=INFO(60,16,M)
 INFO(60,16,M)=INFO(61,16,M)
 INFO(61,16,M)=K3
 CALLLDM160(M,55)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.55)THEN

IF(INFO(INFO(55,16,M),36,0).GE.Lb+INFO(INFO(5
6,16,M),12,M))THEN
 K3=INFO(55,16,M)
 INFO(55,16,M)=INFO(56,16,M)
 INFO(56,16,M)=K3
 CALLLDM160(M,50)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.50)THEN

IF(INFO(INFO(50,16,M),36,0).GE.La+INFO(INFO(5
1,16,M),12,M))THEN
 K3=INFO(50,16,M)
 INFO(50,16,M)=INFO(51,16,M)
 INFO(51,16,M)=K3
 CALLLDM160(M,45)
```

APPENDIX

```
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.45)THEN

IF(INFO(INFO(45,16,M),36,0).GE.L9+INFO(INFO(4
6,16,M),12,M))THEN
 K3=INFO(45,16,M)
 INFO(45,16,M)=INFO(46,16,M)
 INFO(46,16,M)=K3
 CALLLDM160(M,40)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.40)THEN

IF(INFO(INFO(40,16,M),36,0).GE.L8+INFO(INFO(4
1,16,M),12,M))THEN
 K3=INFO(40,16,M)
 INFO(40,16,M)=INFO(41,16,M)
 INFO(41,16,M)=K3
 CALLLDM160(M,35)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.35)THEN

IF(INFO(INFO(35,16,M),36,0).GE.L7+INFO(INFO(3
6,16,M),12,M))THEN
 K3=INFO(35,16,M)
 INFO(35,16,M)=INFO(36,16,M)
 INFO(36,16,M)=K3
 CALLLDM160(M,30)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.30)THEN

IF(INFO(INFO(30,16,M),36,0).GE.L6+INFO(INFO(3
1,16,M),12,M))THEN
 K3=INFO(30,16,M)
 INFO(30,16,M)=INFO(31,16,M)
 INFO(31,16,M)=K3
 CALLLDM160(M,25)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.25)THEN

IF(INFO(INFO(25,16,M),36,0).GE.L5+INFO(INFO(2
6,16,M),12,M))THEN
 K3=INFO(25,16,M)
 INFO(25,16,M)=INFO(26,16,M)
 INFO(26,16,M)=K3
 CALLLDM160(M,20)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.20)THEN

IF(INFO(INFO(20,16,M),36,0).GE.L4+INFO(INFO(2
1,16,M),12,M))THEN
 K3=INFO(20,16,M)
 INFO(20,16,M)=INFO(21,16,M)
 INFO(21,16,M)=K3
 CALLLDM160(M,15)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.15)THEN

IF(INFO(INFO(15,16,M),36,0).GE.L3+INFO(INFO(1
6,16,M),12,M))THEN
 K3=INFO(15,16,M)
 INFO(15,16,M)=INFO(16,16,M)
 INFO(16,16,M)=K3
 CALLLDM160(M,10)
 ENDIF
 ENDIF
 IF(INFO(0,16,M).GT.10)THEN

IF(INFO(INFO(10,16,M),36,0).GE.L2+INFO(INFO(1
1,16,M),12,M))THEN
 KL=INFO(10,16,M)
 INFO(10,16,M)=INFO(11,16,M)
 INFO(11,16,M)=KL
 CALLLDM160(M,5)
 ENDIF
 ENDIF
400 IF(INFO(0,16,M).GT.5)THEN

IF(INFO(INFO(5,16,M),36,0).GE.L+INFO(INFO(6,1
6,M),12,M))THEN
 KLL=INFO(5,16,M)
 INFO(5,16,M)=INFO(6,16,M)
 INFO(6,16,M)=KLL
 CALLLDM160(M,0)
 ENDIF
 ENDIF
222 CALLUPDOWN(16)
111 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE LDM160(M,K)
C-------------------------------------------
--------------------------
CBROUT FROM LDM16IS USED IN DSD.
 INCLUDE'.F.FILES/COM'
 MOVE=0
 KGOTO=0
 J1=INFO(K+1,16,M)
 J2=INFO(K+2,16,M)
 J3=INFO(K+3,16,M)
 J4=INFO(K+4,16,M)
 J5=INFO(K+5,16,M)
 IF(INFO(12,0,12).EQ.16)CALLQUEUE
 CALLLDM0(J1,J2,J3,J4,J5,KGOTO,MOVE,M,16)
 MOVE=MOVE+K
 IF(KGOTO.EQ.1.OR.KGOTO.EQ.5)GOTO1
 LLL=INFO(K+1,16,M)
 INFO(K+1,16,M)=INFO(MOVE,16,M)
 INFO(MOVE,16,M)=LLL
1 CONTINUE
 RETURN
 END
C-------------------------------------------
--------------------------
 SUBROUTINE MINPQ
C-------------------------------------------
--------------------------
C BROUT: COMPARE Q23 AND Q16 ANCHOOSE ONE TO
FORWARD TO MINPQ (WIP & NXTDO)
 INCLUDE '.F.FILES/COM'
 COMMON/P/PPASSA,PPASSE,PPASSM
 NOMORE=0
200 JATM=0
 IQUE16=0
 CALLUPDOWN(16)
 DO 1 JATM=MACHS,1,-1
 IF(INFO(8,0,8).EQ.1)MSUB=JATM

IF(INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6)CAL
LLDM16(JATM)
51 N16=INFO(0,16,JATM)
 J16=INFO(N16,16,JATM)
 J23=INFO(1,23,JATM)
 CALLMOVE(JATM,17)
C LOCAL BUFFUR

IF(INFO(4,17,JATM).GT.0.OR.(J23.EQ.0.AND.J16.
EQ.0))GOTO6
 IGO=0
C SEE PRIORITY
15
IF(INFO(8,0,10).EQ.17.OR.INFO(8,0,9).EQ.6)GOT
O17
3 IF(J23.GT.0.AND.J16.GT.0.AND.
 @
```

APPENDIX

```
(INFO(J23,10,MSUB).LE.INFO(J16,10,MSUB)))GOTO
4
33 IF(J23.GT.0.AND.J16.GT.0.AND.
 @
(INFO(J23,10,MSUB).GT.INFO(J16,10,MSUB)))GOTO
5
17 IF(J23.EQ.0)GOTO50
 IF(J16.EQ.0)GOTO4

IF(INFO(J23,36,JATM).GT.INFO(J16,36,JATM).AND
.
 @ INFO(J23,36,JATM).GT.INFO(0,1,0) )goto5

if(info(j16,36,jatm).lt.info(0,1,0).and.J23.G
T.J16)GOTO5
 GOTO4
50 IF(J16.EQ.0)GOTO6
 GOTO5
C LOCAL BUFFUR
4
IF(INFO(4,17,JATM).EQ.0.AND.J23.GT.0)CALLPRET
OQ(JATM)
 GOTO6
5 IF(INFO(0,0,JATM).NE.0)GOTO1
C LOCAL BUFFUR

if(info(8,0,10).eq.17.or.info(8,0,9).eq.6)the
n

if((info(J16,36,0).le.info(0,1,0)+info(10,17,
jatm).and.
 @ (INFO(4,17,JATM).EQ.0.AND.J16.GT.0)).or.
 @ info(info(0,47,0),17,jatm).eq.0)CALL
MQTOQ(JATM)
 Else
 IF(INFO(4,17,JATM).EQ.0.AND.J16.GT.0)CALL
MQTOQ(JATM)
 endif
6 IF(INFO(8,0,10).NE.12)CALL
ARANGE(17,MACHS,INFO(8,0,8),10)
 CALL MOVE(JATM,17)

IF(INFO(0,17,JATM).EQ.0.AND.INFO(1,17,JATM).G
T.0)CALLLDM17(JATM)
 CALL MOVE(JATM,17)
 NOMORE=NOMORE+1
C LOCAL BUFFUR

IF(INFO(4,17,JATM).EQ.0.AND.NOMORE.LE.4)GOTO5
1

IF(INFO(0,16,JATM).GT.IQUE16)IQUE16=INFO(0,16
,JATM)
1 CONTINUE
 CALL UPDOWN(16)
 PPASSE=PPASSE+1
 INFO(7,0,10)=IQUE16
 info(2,0,11)=-1
111 RETURN
 END
C----------------------
 SUBROUTINE STDV
C----------------------
 INCLUDE'.F.FILES/COM'
C SD=SQRT(TOT((X-XM)*(X-XM))/N)
 ATOT=0
 XM2TOT=0
 AC2TOT=0
 AM2TOT=0
 fc2tot=0
 fm2tot=0
 NN=0
 LA=0
                  nnnn=0
 kkkk=0
 AAC=0
 AAM=0
 ffm=0
 ffc=0
 XXM=0
                  DO 1 I=1,JOBS
 IF(INFO(I,15,0).LT.0)THEN
 A=ABS(INFO(I,15,0))
 LA=LA+1
 ATOT=ATOT+A
 ENDIF
1                 CONTINUE
 info(6,0,14)=LA
 info(3,0,1)=ATOT
 AM=ATOT/max(1,info(3,0,14))
CDIV
                  XM=ATOT/max(1,info(4,0,14))
 INFO(4,0,5)=1000*XM
CDIV
 fc=1.*info(4,0,1)/max(1,info(3,0,13))
CDIV
 fm=1.*info(4,0,1)/max(1,info(4,0,14))
CDIV
 f32=ATOT/max(1,info(3,0,14))
 info(3,0,2)=1000*f32
CDIV
 AC=ATOT/max(1,info(6,0,14))
 INFO(6,0,5)=1000*AC
 DO 2 II=1,JOBS
 if(info(ii,15,0).gt.0)then
                  KKKK=KKKK+info(ii,15,0)
 NNNN=NNNN+1
                  endif

if(info(ii,15,0).lt.0.and.info(ii,0,0).eq.11)
then
 f=abs(info(ii,15,0))
 ffc=F-fc
 ffm=f-fm
 ffm2=ffm*ffm
 ffc2=ffc*ffc
 fm2tot=fm2tot+ffm2
 fc2tot=fc2tot+ffc2
 endif
 IF(INFO(II,15,0).LT.0)THEN
 A=ABS(INFO(II,15,0))
 AAC=A-AC
 AAM=A-AM
 XXM=A-XM
 AAC2=AAC*AAC
 XXM2=XXM*XXM
 AAM2=AAM*AAM
 AC2TOT=AC2TOT+AAC2
 XM2TOT=XM2TOT+XXM2
 AM2TOT=AM2TOT+AAM2
                  ENDIF
2                 CONTINUE
 info(3,0,4)=kkkk
 info(6,0,13)=nnnn
 f35=1.*info(3,0,4)/max(1,info(6,0,13))
 info(3,0,5)=1000*f35
 Vcf=fc2tot/max(1,info(3,0,13))
                  VM=XM2TOT/
max(1,info(4,0,14))
 Vfm=FM2TOT/max(1,info(4,0,14))
 VCa=AC2TOT/max(1,info(6,0,14))
 Vam=AM2TOT/max(1,info(3,0,14))
 SDM=SQRT(VM)
                  SDca=SQRT(VCa)
 SDfm=SQRT(Vfm)
 SDcf=SQRT(Vcf)
 SDam=SQRT(Vam)
 INFO(3,0,3)=1000*SDM
 INFO(5,0,3)=1000*SDca
 info(4,0,3)=1000*SDfm
 info(6,0,3)=1000*SDcf
 info(3,0,12)=1000*SDam
```

APPENDIX

```
 f42=1.*INFO(4,0,1)/MAX(1,INFO(4,0,14))
 INFO(4,0,2)=f42*1000
 F52=1.*info(3,0,1)/(MAX(1,INFO(6,0,14)))
 info(5,0,2)=f52*1000
 f36=1.*info(3,0,1)/max(1,info(3,0,13))
 info(3,0,6)=1000*f36
 f62=1.*info(4,0,1)/max(1,info(3,0,13))
 info(6,0,2)=1000*f62
 PFlate=1.*info(3,0,13)*100/
max(1,info(4,0,14))
 PAlate=1.*info(6,0,14)*100/
max(1,info(3,0,14))
 info(4,0,11)=PAlate*1000
 info(4,0,10)=PFlate*1000
 RMS=info(4,0,8)/1000.
 RMSC=info(4,0,7)/1000.
 f214=info(2,0,14)/100.
 f714=info(7,0,14)/100.
 f54=info(5,0,4)/1000.
 f64=info(6,0,4)/1000.
 f612=info(6,0,12)/1000.
 f49=info(4,0,9)/1000.
 f80=info(0,8,0)/100.

 if(info(0,1,0).gt.info(1,0,9)+100)stop

if((info(0,1,0).gt.2494.and.info(0,1,0).lt.25
06).OR.
 @
(info(0,1,0).gt.994.and.info(0,1,0).lt.1006).
OR.
 @ (info(0,1,0).gt.info(1,0,9)-
7.and.info(0,1,0).lt.
 @ info(1,0,9)+7))then
 write(200,701)info(8,0,10),

@info(0,1,0),info(3,0,1),info(4,0,1),info(3,0
,14),

@info(6,0,14),info(4,0,14),info(3,0,13),PFlat
e,PAlate,f32,f52,

@f42,f62,f49,f214,f714,info(3,0,4),info(0,14,
0),info(4,0,4),f80
ccc
f612,RMS,RMSC,info(3,0,4),info(4,0,4),f54,f64
,f35

700 Format('RL',1x,
 @ ' Time',1x,' Trds.R',1x,' Trds.X',1x,'
Rcv','#','L.R',1x,'eXt'
 @,'#','L.X',1x,' LxX%',1x,' LrR%',1x,' T/
R',1x,' T/LR',1x,' F/X'
 @ ,1x,' F/XL',1x,'Avrg',1x,' R.rt','~','
X.rt',1x,'T.Erly',x,
 @ 'RM.wrk',x,'X.erly','cp%')
CCC ' DD%',1x,' RMS',1x,'
CRMS*Erl.Rc',1x,'Erl.Xt',1x,' M.EX',1x,'
CM.E',1x,' M.ER')

701 Format(I2,' ',I5,1x,I7,1x,I7,' ',I4,'
',I4,' ',I4,' ',I4,1x,2(
 @ F5.2,' '),4(F5.1,1x),F5.1,' ',2(F5.2,'
'),I6,x,I6,x,I6,x,F5.2)
CCC ,F4.1,'%
',F5.1,1x,F5.1,'*',I6,1x,I6,1x,3(F5.2,1x))
 endif

 RETURN
 END
```

APPENDIX

2. THE PROGRAMME OF DATA GENERATOR

```
 character*5 ix1,ix2,ix3,ix4
 LOGICAL X1,x2,x3,x4
 LOGICAL G05DZF
 EXTERNAL G05DZF
 EXTERNAL G05CBF
 INTEGER G05DYF
 EXTERNAL G05DYF
open(999,file='sys1',status='scratch')
open(888,file='sys2',status='scratch')
 open(99990,file='SEED',status='old')
123 WRITE (6,*) 'TYPE SEED number:'
 Read(99990,321)kk
321 format(I4)
 NOUT1=KK+1000
 Nout2=KK+2000
 CALL G05CBF(KK)
 DO 20 I = 1,2000
* Processing times
 mk1 = G05DYF(1,15)
 mk2 = G05DYF(1,15)
 mk3 = G05DYF(1,15)
 mk4 = G05DYF(1,15)
* Routine 0-1
 X1 = G05DZF(0.5D0)
 X2 = G05DZF(0.5D0)
 X3 = G05DZF(0.5D0)
 X4 = G05DZF(0.5D0)
* Number of jobs to be received next
 Nm = G05DYF(1,5)
* After how many minutes to receive the Nm jobs
 Nwhen=G05DYF(1,29)
 write(Nout2,901)Nm,Nwhen
901format(i5,x,i5)
 Nout=1
 rewind(999)
 write(999,90000)x1,x2,x3,x4
 rewind(999)
read(999,9999)ix1,ix2,ix3,ix4
 iK1=0
 ik2=0
ik3=0
ik4=0
 if(iX1.eq.' T')ik1=1
 if(iX2.eq.' T')ik2=2
 if(iX3.eq.' T')ik3=3
 if(iX4.eq.' T')ik4=4
 j=5
if(ik1.eq.0.and.ik2.eq.0.and.ik3.eq.0.and.ik4
.eq.0)j=G05DYF(1,4)
 goto(1,2,3,4,5),j
1 ik1=1
 t=t+1
 print*,t
 goto5
2 ik2=2
 t=t+1
 print*,t
 goto5
3 ik2=3
 t=t+1
 print*,t
 goto5
4 ik2=4
 t=t+1
 print*,t

5 ip1=ik1*mk1
 ip2=ik2*mk2/2
ip3=ik3*mk3/3
 ip4=ik4*mk4/4

* IW wight- ICST cost- ITOT total processing
time- IDD Due Date
 iw=g05dyf(1,9)
 icst=g05dyf(1,9)
 itot=ip1+ip2+ip3+ip4
idd=itot+itot*g05dyf(1,9)
kk2=0
kk3=0
kk4=0
 kp2=0
 kp3=0
kp4=0
 rewind(888)
 if(ik1.gt.0)write(888,*)ik1,ip1
 if(ik2.gt.0)write(888,*)ik2,ip2
 if(ik3.gt.0)write(888,*)ik3,ip3
if(ik4.gt.0)write(888,*)ik4,ip4
 rewind(888)
read(888,*)kk1,kp1
read(888,*,end=124)kk2,kp2
```

APPENDIX

```
read(888,*,end=124)kk3,kp3
read(888,*,end=124)kk4,kp4
124WRITE (NOUT1,99999)i+1004,idd,iw,icst,
 @ kk1,kk2,kk3,kk4,kp1,kp2,kp3,kp4
 20 continue
 goto123
9999 format(4A2)
99999 FORMAT (x,x,I4,x,I5,x,i1,x,-
i2,1X,4(i2,x),4(' .'),4(i3,x))
90000 FORMAT (4L2)
 END
```

Fig 24

181

170

fig 26

table 6-7

table 6-6

table 6-2

table 6-3

Table 6-5