SIMULATION OF PRODUCTION SCHEDULING IN MANUFACTURING SYSTEMS

MUJANAH EZAT AGHA, B.Eng.

The thesis submitted to Dublin City University in Fulfilment of the requirement for the award of degree of *Master of Engineering*

Supervisor: Dr. M. A. Al-Baradie

Dept of Mechanical and Manufacturing Engineering
Dublin City University, Dublin 9
11th February, 1993

بسم الله الرحمن الرحيم

To

My

Home

Parents

Sister ,Brothers

Wife, Daughters (Azza and Liela) & son (Muhanad)

MIA

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering is entirely my own work and has not been taken from the work of others save and the extent that such work has been cited and acknowledged within the text of my work

Signed ______________________________

Mujanah Ezat Agha I D No 90700309

ACKNOWLEDGMENTS

First of all I would like to express my special thanks to the Syrian Government who sponsored this research
I indebtedness as well as my gratitude are due to my supervisor Dr M A Al-Baradie for his continuous help, the many things I learned from him during discussions in his office and encouragement during my M Eng Studies
I would like to thank professor M S J Hashmi for his guidance and support throughout the project I wash to express my faithful thank to the members of my Master Engineering Committee, Professor M M Ahmad, Professor M S J Hashmi and Dr M A Al-Baradie They provided through two hours period the direction needed in this research, and their comments to my work made significant contributions

I wish to express my sincere thanks to the general director of the Scientific Studied and Research Centre (Syria), my supervisors at home Eng R Al-kudsi, Dr G Mosa and Dr S Tuhma for helping me by one way or another
I specially thank my wife Sohela, Ebrahem Bacha for her constant encouragement and who has been waiting for me so long
My love to daughters and son who kept quiet while I was at home
Finally, to my beloved parents who encourage me all my life, and all those who have been waiting for me so long, I dedicate this very modest contribution

ABSTRACT

Research into production scheduling environments has been primarily concerned with developing local priority rules for selecting jobs from a queue to be processed on a set of individual machines Most of the research deals with the scheduling problems in terms of the evaluation of priority rules with respect to given criteria These criteria have a direct effect on the production cost, such as mean make-span, flow-time, job lateness, in-process inventory and machine idle time

The project under study consists of the following two phases

The first is to deal with the development of computer models for the flow-shop problem, which obtain the optimum make-span and near-optimum solutions for the well-used criteria in the production scheduling priority rules

The second is to develop experimental analysis using a simulation technique, for the two main manufacturing systems,

1 Job-shop

2 Flexible Manufacturing System

The two manufacturing types were investigated under the following conditions

i Dynamic problem conditions

ii Different operation time distributions

iii Different shop loads

iv Seven replications per experiment with different streams of random number

v The approximately steady state point for each replication was obtained

In the FMS, the material handling system used was the automated guided Vehicles (AGVs), buffer station and load/ unload area were also used The aim of these analyses is to deal with the effectiveness of the priority rules on the selected criteria performance The SIMAN software simulation was used for these studies

Definition of Subscripts, Symbols & Terms

n	number of jobs
m	number of machines
B	objective function (a selected criterion)
P	pure flow-shop problem
C_{max}	make-span (elapsed time for a schedule)
$p_{ji}=t$	processing time of job j on machine m
n X m	problem flow-shop matrix (or processing time matrix)
q	a given sequence of flow-shop permutation
j	job j
i	machine i
K_j	number of operation
F	a flow-shop problem
G	a general job-shop
F_{max}	maximum flow-time
F	mean flow-time
L	mean job lateness
$O_{j\ m_j}$	operation m_j of job j
r_j	the ready time of job j
d_j	the due date of job j
$W_j=1$	all jobs are equally important
n'	number of sequences in a (n X m) problem size
W-I-P	work in process (number of jobs in shop)
AMCTs	average mean completion times for jobs
AWTs	average waiting times for jobs
AITs	average idle times for machines
ANOVA	analysis of variance
FMS	flexible manufacturing system
AGV	automated guided vehicle
CNC m/c	computer numerical control
QC	quality control

CONTENTS

List of Figures

List of Tables

CHAPTER ONE

1. INTRODUCTION:

In any manufacturing system there are a series of activities that transform raw or semi-raw materials into semi or finished goods Production scheduling is one of the most important manufacturing activities which allocates machines over time to perform a collection of tasks (job sequencing problem) Scheduling problems are very common occurrences A problem could involve jobs in a manufacturing plant, aircraft waiting for landing clearance, a programme to be run on a computer A good dealing of job sequencing implies high machine utilization, minimum make-span or flow-time and low inventory (work-in-process) while maintaining customers fulfilment

In the past research, a number of researchers had worked on machine scheduling problems in the hope of finding

1 Optimal solution (i e , mathematical models such as linear programming models and branch-and-bound method) of the problems

2 Near-optimal solution (i e , heuristic procedures such as a random procedure and straightforward priority rules)

In recent times, most of the studies have turned to deal with machine scheduling problem using the heuristic procedures The priority rules as a heuristic procedure is a somewhat more economical method than others and also it handles a very large problem with a light computational effort This facility becomes more powerful when the simulation technique is applied In general, with regard to job sequencing, most researchers classify the manufacturing systems into three main types

- Flow-Shop (Transfer Line)
- Job-Shop (Batch production)

- Flexible Manufacturing System

The objective of this project is three-fold

1 To deal with the flow-shop problem with respect to the optimal and near-optimal make-span using the selected priority rules This shop in which each job has the same order to be processed on machines (unidirectional process)

2 To study the job-shop problem in which each job is not in the same order for processing (multidirectional process) The objective is to evaluate the effectiveness of the various priority rules with respect to the selected criteria

3 To study the Automated Guided Vehicle (AGV) in the Flexible Manufacturing System (FMS) problem, where the AGV is the main tool for the material handling system Also in this study each job is not in the same order of processing The first objective in the third study is to evaluate the different priority rules with respect to various criteria The second objective is to consider the effect of number of AGVs on the FMS multi-criteria when using AGV's different speeds

The project consists of the following Chapters

Chapter two surveys the work previously carried out to investigate the traditional machine scheduling problem
It covers three main areas, optimum approaches, heuristic methods and computer simulation of production scheduling

Chapter three gives an overview of production scheduling The following machine scheduling environments are discussed

- Description of a general machine scheduling problem
- Restrictive assumptions on the machine and job
- Scheduling costs and measure of performances (criteria)
- Problem classifications

Chapter four presents a production scheduling study for the flow-shop problem A basic formula to obtain the throughput time (make-span) for a schedule is presented
This formula has been used as a main tool to develop a computer programme This programme could be used for solving a large job

sequencing problem (90 jobs to be processed on 90 machines). The objective is to evaluate the make-span and individual job completion time for any selected order (job sequence).
This programme is the basic step to develop another computer programme which gives an exact (optimum) solution for make-span for up to 10 jobs (3.628800E+6 different job sequences) on 90 machines. The proposed optimum solution would, of course, be a practical tool for flow-shop scheduling, where $n \leq 10$ and $m \leq 90$ and the CPU time (386 based PC with 16MHz) is reasonable in the case of low and medium work-in-process and low, medium and high utilization levels.

Chapter five reports a special presentation for sequential scheduling rules in production scheduling. It includes, classification of scheduling rules, priority rule environments (static and dynamic rules), priority rules information required (local and global rules). Also a general computer simulation based priority rules have been developed. This programme is mainly used as a tool for measuring the effectiveness of the most used priority rules in a pure flow-shop problem. It can read data for 90 jobs to be processed on 90 machines and it could be used for a deterministic or stochastic(n X m) processing times matrix. The effectiveness of the priority rules have been evaluated with respect to the following well-known criteria under 500 simulation runs with different random number seeds:

- Make-Span.
- Total Mean Completion time.
- Total job Waiting times.
- Total machine idle times.

In all (n x m) shop sizes tested the SPT rule gives the best value with respect to the make-span as reported by Coway[51]

Chapter six mainly deals with the job-shop scheduling problem. A comprehensive study for the effectiveness of the priority rules with respect to the well-known criteria for job-shop problem are presented. The SIMAN software for the job-shop scheduling simulation has been used For more accurate results the steady state point was taken into account during the estimation of the mean and variance of random variables for the multiple criteria

The total experimental simulation runs are equal to 882 replications These results were under individual three operation time distributions and three job arrival time distributions with low, medium and high workloads

Also the SPT rule was the dominant rule against the mean flow-time, lateness, in-process inventory and jobs completed

Chapter seven has been devoted to study the flexible manufacturing system with automated guided vehicles The following four main elements are comprised into the system

1 Processing stations
2 Load/Unload station
3 Automated Guided Vehicles (AGVs)
4 Buffer storage at work-station

In the FMS the control logic and material flow are very complex, but can be successfully examined through the use of a computer simulation

The SIMAN software simulation is used for this study Also the approximate steady state point was taken into account during the estimation of the mean and variance of random variables for the multiple criteria

The total experimental simulation runs are equal to 231 replications These results were under exponential job arrival distributions with three different means These means were suitable according to low, medium and high FMS loads

The objectives from this study are two-fold The first objective is to evaluate the different priority rules with respect to various criteria The second objective is to consider the effect of the number of AGVs on the FMS multi-criteria when using AGV's different speeds under the selected SPT priority rule for job sequencing Also the SPT rule was the dominant rule with respect to the mean flow-time, lateness, in-process inventory and jobs completed

CHAPTER TWO

2. LITERATURE SURVEY:

2.1 INTRODUCTION:

The literature survey for this research includes a comprehensive study of scheduling problems. During the last three decades of effort a considerable number of papers have been published and a review of these papers is given in this Chapter. The beginning of the scheduling problem came just in the mid-fifties in the form of a paper presented by Johnson 1954[1]. This paper discussed the scheduling of n-jobs on two machines flow-shop. The rule is: Select the shortest processing time (SPT rule). His paper is the most important reference to scheduling problem. The algorithm assumes that the set-up and tear down time are included in the total operation time. Shortly after, Jakson[1955][2] & Smith[1956][3] derived a new algorithm concerning the problem of sequencing several jobs on a single machine so as to minimize maximum tardiness or to minimize the sum of completion time. Akers[1956][4] used a graphical method to solve the sequencing problem for two-jobs on m machines. In this non-numerical method, work on job X is presented by an X-vector and on job Y by a Y-vector, whilst work carried out on both job X and job Y is presented by the vector sum of X and Y. Through several equally acceptable solution we can determine the optimum solution. However, this method is limited to two jobs only. Conway, Maxwell and Miller[1967][5] and Baker[1974][6] discussed the limited case of the single and two machines problem in more details, followed by Rinnooy Kan[1976][7], while White[1969][8] derived the same results using dynamic programming. Campbell, Dudek & Smith[1970][9] used a multiple application of the Jonson rule for two machines. They created (M-1) auxiliary scheduling problems and applied the Jonson rule to each of them in turn and then picked out the best solution Szware[1977][10] extended the Jonson rule to solve the sequencing of a three machines problem Pinedo[1981][11] discussed an approach which minimizes the Expected completion time for n jobs on 2 machines, when the processing time for all jobs are derived from exponential distribution Bera[1984][12] has developed algorithms to determine a very near-optimal solution for waiting time, idle time and make-span for flow-shop problem He used economical pairs of jobs

In view of this, a brief survey and discussion will be presented on work previously carried out to investigate the scheduling problem This will be discussed under the following three types of scheduling algorithms

2.2 OPTIMUM APPROACHES:

They are the basis for many heuristic and provide the foundation for continuing research in machine scheduling

2 2 1 Enumeration

a Complete

b Implicit

b 1 Branch-and-Bound

b 2 Branch-and-Dominate

b 3 Dynamic Programming

c Constructive

d Mathematical Programming

d 1 0-1 Integer models

d 2 Binary Disjunctive LP Models

2.3 HEURISTIC APPROACHES:

Large size industrial problems are left to heuristic methods which give reasonable near-optimum solutions

2 3 1 Priority rules

2.3.2 Monte Carlo method

2.3.3 Neighbourhood method

2.4 COMPUTER-AIDED SIMULATION OF PRODUCTION SCHEDULING APPLICATION

2.2 OPTIMUM APPROACHES:

Complexity theory due to Cook[1971][13] and Rennooy Kan[1976][7] suggests that no polynomial time optimum algorithm will be found for the n/m/G/B problem*, except in a limited number of special cases (e. g. certain one and two machine problem).

2.2.1 ENUMERATION:

a. COMPLETE ENUMERATION:

Let(s) be the set of all feasible solutions for the general machine sequencing problem. Since (s) is finite, an optimal schedule can always be found by complete enumeration of elements of (s) [Rinnooy Kan,1976][7]. Unfortunately a computer evaluating 100,000 schedules would still need almost three days to evaluate this number of schedules Rinnooy Kan[1976][7]. However, nowadays CPU time could be more powerful for dealing with complete enumeration scheduling problem, especially for n/m/P/B problem. Ezat, Agha.M and Al Baradie, M.[1992][14] have developed a computer programme algorithm which deals with the optimal job sequence for **$n/m/P/C_{max}$** (i,e The CPU time for **$10/90/P/C_{max}$** problems is about 2.5 hours on 386 PC with 16 MH_z). Backer[1974][6] and Giffler and Thompson[1960[15] had offered an algorithm which created an active schedule with respect to disjunctive arcs. The limitation of this algorithm is that it presents precedence relations that cannot be determined before a schedule is constructed. Also it is not adequate to capture

* Table 2.1 describes the following job scheduling problem notations.

Table 2.1: Notations used for the job scheduling problem.

n/m/P/B: is the n jobs, m machines, pure flow-shop problem where B is the measure of performance

n/m/F/B: is the n jobs, m machines, flow-shop problem where B is the measure of performance

n/m/G/B: is the n jobs, m machine, general job-shop problem, where B is the measure of performance

n/1/F_{max} is the n jobs, single machine where the aim is to minimise flow time

n/1/L_{max} is the n jobs, single machine where the aim is to minimise lateness

n/2/F/F_{max} is the n jobs, two machines, flow-shop problem in which the aim is to minimise flow time

n/2/G/F_{max} is the n jobs, two machines, general job-shop in which the aim is to minimise flow time

2/m/F/F_{max} is the 2 jobs, m-machines, flow-shop problem in which the aim is to minimise flow time

n/m/P/C_{max} is the n-jobs, m-machines, p permutation or pure flow-shop problem where the aim is to minimise make-span

The class P: consists of all problems for which algorithms with polynomial time behaviour have been found

The class NP: is the set of problems for which algorithms with exponential behaviour have been found

$\Pi_1 \; \alpha \; \Pi_2$: problem Π_1 is polynomially reducible to problem Π_2

NP-complete: we say that a problem Π lying in **NP** is **NP-complete** if every other problem in **NP** is polynomially reducible to Π, that is

$$\Pi' \; \alpha \; \Pi \text{ for all } \Pi' \text{ lying in } \mathbf{NP}$$

NP-hard : we say that a problem lying in NP-complete is **NP-hard** when the associated recognition problem is **NP-complete** cannot solve the optimisation problem in polynomial time sequence dependent set-up and tear down time in every case [Preston, White, JR K et al ,1990][16]

b. IMPLICIT ENUMERATION:

The strategy of implicit enumeration attempts to minimize an objective function without considering every possible solution explicitly Implicit enumeration schemes examine increasingly smaller subsets of feasible solutions until these subsets definitely do not contain improved solutions Unfortunately, all implicit enumeration approaches for the determination of an optimal schedule, appear to be susceptible to the combinatorial natural of these problems, when they are tested on the multiple-resource version (more than 50 activities) This statement was investigated in practice by Baker[6] The three principal methods of implicit enumeration are Branch-and-Bound, Branch-and Dominate and Dynamic Programming

b.1 BRANCH-AND-BOUND:

This method is a typical technique of implicit enumeration or tree search method which can find an optimal solution by systematically examining the subsets of a feasible solution In fact, it does not refer to a specific solution procedure rather, it is an approach which can be applied to many combinatorial problems Its easy implementation and often surprising efficiency to a much large class of problems Agin,1966[17], and [Wood and lawler,1966][18] gave more general survey It was first used in the context of mixed integer programming by [Land & Doig,1969][19] and for the travelling salesman problem by [Eastman,1959][20] Then it was applied to scheduling problems by [Ignall & Schrange,1965][21] for **$n/3/F/C_{max}$**, [Brooks & White ,1965][22] and Conway, et al,1967][5] for **$n/m/G/C_{max}$** The limitation of this algorithm is that the make-span is the only criterion which can be evaluated Balas[1969][23] offered an alternative branch-and-bound approach based upon the disjunctive graphs using a search tree (i e sequence) of conjunctive graphs Raimond, JF[1968][24] proposed an algorithm to solve the general (m>3) problem by a branch-and-bound technique using the linear programming with mixed variables and graph theory His algorithm yields an optimal solution, it may be used without modifications or with little changes to find the sub-optimal solution for very large problems Lominici[1965][25] had developed an independent specific formulation for the exact solution of the three machine scheduling problem

b.2 BRANCH-AND-DOMINATE:

Branch-and Dominate, which is similar to branch-and-bound, differs in the pruning approach Suppose for example, that there is a set of conditions from which we can deduce all the schedules at one node which can not do better than the best schedule at some other node Clearly, we may eliminate the first node from further consideration, then the second node dominate the first Using dominance conditions may shorten the search sufficiently that, overall, a reduction in computational requirements is obtained Indeed, this had been found in practice by Baker[1974][6], Rinnooy Kan[1976][7] and Lageweg, et al [1977][26] The computational problems facing branch-and dominate are the same as branch-and-bound from the starting point of computational complexity

b.3 DYNAMIC PROGRAMMING:

Dynamic programming methods have been used to solve a limited number of machine scheduling problems These problems have mainly been n/m/P/B and in particular the n/1/P/B [Held & Karp,1962][27] The size of the search graph (similar to the branch-and-bound) is often superpolyminal and the pruning mechanisms inherent in the equations may be rather weak [Rinnooy Kan,1976][7] Further, a very large number of intermediate calculations must be stored in memory. This method is usually used only for single machine problems up-to 25 jobs. Also it requires a lot of calculation, far more than any other solution that we have met [French,1982][28].

c. CONSTRUCTIVE APPROACHES:

Constructive approaches are based on building an optimal schedule in a single pass, by following a simple set of rules. These approaches have been developed for certain specific scheduling problems known to be the class P. Among these are: n/1/F, n/1/L_{max}, n/2/F/F_{max}, n/2/G/F_{max}, and 2/m/F/F_{max} [Rinnooy Kan,1976][7] and [French,1982][28]. Generally, most n jobs, 1 and 2 machines problems lend themselves to efficient optimal solution methodologies derived from constructive approaches. Constructive algorithms have not provided the general optimal solution methods for more than 2 machines. Johnson's Algorithm for (n/2/F/F_{max}) problem may be extended to a special case of the n/3/F/F_{max} problem.

d. MATHEMATICAL PROGRAMMING:

In fact this was the earliest method to be used to solve NP-hard problems by [Bowmann,1959][29] and [Wagner,1959][30], then by [Pritsker, et.al,1969][31]. This method is a general form of a mathematical programming which consists of formulating the scheduling problem as mixed integer, linear or non-linear programming problems. The models from the literature fall into two categories: 0-1 integer models and binary disjunctive models:

d.1 0-1 INTEGER MODELS:

Bowmann,1959[29] first developed a mathematical programming formulation of the machine scheduling problem. His model was a 0-1 integer mathematical programme in which he had tried to take an advantage of the simplex method which can be used to solve linear programming problems, but failed to find a good algorithm as scheduling problems need extensive number of 0-1 variables and constraints. Wagner,1959[30] presented an approach which is suitable for only n/m/P/B problem. Pritsker,1969[31] offered a more compact model of this form.

d.2 BINARY DISJUNCTIVE LP MODELS:

Manne,1960[32] used Zero-One integer decision variables in linear inequalities which defined a partial machine sequence. The resulting problem is solved as a mixed integer programme. Greenburg,1968[33] used linear inequalities defined by a particular partial machine sequence. The resulting family of problems are solved as a set of linear programmes, one for each distinct sequence. White, et al 1986[34] offered a non-linear mathematical programming formulation with quite different properties. In their model the disjunctive constraints are realized by some of the constraint equations.

2.3 HEURISTIC METHODS:

So far we have briefly discussed methods which reached exact solutions. Unfortunately, the optimum methods of the scheduling problem, suggest that optimum approaches to large NP-hard problems which will fail in this objective within reasonable overall time [French,1982][28]. Faced with the historical experience and the implications of the optimum methods, machine scheduling researchers have pursued solution techniques not predicated on optimality [Rinnooy Kan,1976][7]. Also investigators have sought to develop new heuristics capable of providing good schedules, if not optimal ones. Palmer[1965][35] obtained a quick near-optimum solution for $n/m/F/F_{max}$ using slope-index. Gupta[1971][36] produced a heuristic method for n-job through m machine which was easy to solve. Dennenbring[1977][37] designed a rule which attempted to be mixture between the Palmer[35] & Campble, et al[9] methods This method found the optimal sequence in about 35% of all cases ($n \le 6$, $m \le 10$) Arumugma, et al[1981][38] evaluated two new loading rules based on the monetary value of the job Sarin & Elmagharaby[1984][39] produced a heuristic method, which translated the optimal solution for one processor, into a solution of m processor with arbitrary precedence related jobs The measure of performance is to minimize the weighted completion time

Given below are the better used heuristic approaches which form the basis for most of the developments

2.3.1 PRIORITY RULES:

Priority rules indicate how to assign a specific job to a specific machine at a given time, when a machine becomes available for process [Rowe and Jackson,1956[40]

The literature library involves numerous priority rules which have been considered

Panwalker & Iskander[1974][40], for example, survey over 110 priority rules

A few of the studies are, Rowe[1956][41], Conway, et al [1967][5], Baker,1974[6], Jones,C H [1977][42], French[1982] [28], John, et al[1982][43], Schriber[1991][44] and Ezat, A Mujanah & Al-Baradie, M [1993][45]

Priority rules may be conveniently classified by their transient characteristics and by the information required to implement them Thus, a static priority rule does not change as a function of the passage of time, while a dynamic one leads to be an opposite active schedule [French][28]

A local priority rule requires only information about the jobs to be processed on a machine, while a global rule requires more information about jobs, machines and queue lines It follows that the global rules give more cost processing information than lead the global priority rules [Conway, et al][5]

Table 2 2 shows the classification of a few typical of the more common priority rules

Dzielinki, 1960[46], LeGrande, Earl, 1963[47], Nanot,1963[48], Conway,1964[49] and Nelson,1965[50] found that, of all local priority rules, SPT rule minimized the mean flow time Conway, however, found that a rule formed by combining the shortest processing time rule with a rule which considers the work content of the next queue, could give a slightly better result than the SPT rule by itself

In point of view of the processing time based rules, due to [Conway and Maxewll,1962][51] noted that, in a single-server environment the SPT rule was optimum with respect to certain criteria (minimizes mean flow time and lateness) For the point of the external discussion of assigning due date to arriving jobs [Conway,1965a[52], Conway and Maxwell[51] found that, the SPT rule also minimizes the mean lateness and the number of tardy jobs

Also [Elvers,1973][53] studied the performance of 10 priority rules over five variations of the TWK (Based total work content) due date assignment method (setting the due date as 3, 4, 5, 6 and 7 times the total job processing time) Approximately 250 jobs per run, having uniformly distributed arrival times He found that when the due date is set six times the total processing time or less, the SPT (shortest processing time) rule performed best with respect to a tardiness criterion The SRPT (smallest remaining processing time) rule also performed well, while EDD (Earliest Due date) rule performed the worst rule

Eilon and Coterill[1968][54] and Eilon, et al[1975][55] who addressed a modification of the SI(SPT) rule that will be in the form of SI/SI(F) It took into account due dates (which SI does not) and helped to reduce the delays incurred for very long jobs The authors were pleased with the results obtained from using SI/SI/F modification rule

The most commonly used rule involving a shop characteristic (arrival time and random) is FCFS, FASFS and RANDOM [Conway,1965a][56] tested a variation between the FCFS rule with FASFS

Related to	Rule Symbols	Definition of rules
Processing Time	SPT LPT SRPT LRPT	Shortest Processing Time Longest Processing Time Shortest Remaining Processing Time Longest Remaining Processing Time
Due Date	EDD StS DyS OPNDD	Earliest Due Date Static Slack due date - arrival time Dynamic Slack due date - the remaining expected flow time - the current date Earliest Operation Due Date, assuming the allowed flow time is divided equally among operation
Number of Operation	FOPR MOPR	Fewest Operation Remaining Most Operation Remaining
Arrival Time & Random	FCFS LCFS FASFI RANDO	First Come First Service Last Come First Service First at Shop First In Select in RANDOM order
Machine Attribute	NINQ WINQ	Select job whose next operation is on the machine with the smallest queue Select job whose next operation is on the m/c with the least work
Combinat--ion of simple rules	FCFS/ SPT SEQ	Select jobs based on SPT, but for jobs whose waiting time is greater than a specific value, use FCFS rule Consider work-in-process value of the job, elapsed waiting time and the number of operation

Table 2.2: Selected Priority Rules[41]

He found that, FASFS was slightly better than FCFS on flow time mean and slightly worse on tardiness However, Rochette and Sadowski,1978[57] found that, FASFS did better than FCFS on tardiness for 13 to 15 replications Also Rochette & Sadowski tested NOP rule (the number of operation remaining) This rule performed much worse on tardiness, than all other rules tested Philip, et al,1984[58], presented an experimental analysis of job-shop system to test four priority rules (SPT, FCFS, FASFS and RANDOM) under three levels of shop utilisation Their results indicate that, the SPT rule gave the best performance for all conditions of workload

2.3.2 MONTE CARLO METHODS (Probabilistic priority)

The idea of a monte carlo or random sampling approach is simple

Use some random device, construct and evaluate (X) sequences, and identify the best sequence in the sample The difficult and important issues surrounding this method involves two tactical problems [Baker, 1974][6]

1- What particular device should be used to generate random numbers ?

Baker's idea is that making equally likely choices among resolutions in priority rule algorithms is not the same as making equally likely choices among the set of schedules which establish the population Instead, a given schedule is generated with a probability that varies oppositely with the number of disagreements in it

2- What conclusion can be drawn regarding the best sequence in the sample ?

The conclusions one can draw are directly related to the population size, the sample size, and that distribution of solution values for the population Given that the population size is typically enormous, that the sample size is typically small, and the underlying distribution is always unknown, only one substantial conclusion can be drawn This is that the best

sequence in the sample.
In brief, monte carlo method is a variable procedure for obtaining reasonable solution having a limited amount of computational dealing. In more complicated scheduling problems, this method have provided effective heuristic procedure. Also it appears to be competitive with other general purpose heuristic methods. However, the task for research is to determine, how these issues which mentioned above should be resolved to arrive at an active monte carlo procedure [Baker,1976[6].

2.3.3 Neighbourhood Methods:

Neighbourhood search technique begin with any feasible schedule, adjust this somewhat, check whether the adjustment has made any improvement. Continuing in this cycle of adjustment and testing until an improvement measure is achieved. Two related concepts which are the basis of this method are the neighbourhood sequence and the neighbourhood generating mechanisms for these sequences [Baker,1976][6].
A neighbourhood generating mechanism is a method of taking one sequence as a seed and systematically creating a collection of related sequences (i.e the neighbourhood sequence).
A general algorithmic description for the family of neighbourhood search techniques, is given below [French, 1982][28].

Step 1. Obtain a sequence to be an initial seed and evaluate it with respect to the given performance measure.

Step 2. Generate and evaluate all the sequences in the neighbourhood of the seed. If none of the sequences are better than the seed with resect to the given measure of performance, stop. Otherwise proceed.

Step 3. Select one of the sequences in the neighbourhood that improved the measure of performance. Let this sequence be the seed. Return to step 2.

The search procedure of this family of algorithms terminates

with a sequence that is a local optimum (with resect to the given neighbourhood structure)
Unfortunately, there are in general no way to guarantee or even know either that the terminal sequence is also a global optimum
However, few experiments [Spachis and King, 1979][59] indicated that, fundamental neighbourhood search algorithm described above, is fairly reliable as a general purpose heuristic procedure [Baker, 1976][6]

2.4 COMPUTER AIDED-SIMULATION OF PRODUCTION SCHEDULING APPLICATIONS:

An early definition of simulation is written by West Churchman[1963][60], as

' "x simulates y" is true if and only if (a) x and y are formal system, (b) y is taken to be the real system, (c) x is taken to be an approximation to the real system, and (d) the rules of validity in x are non-error-free '
A recent definition by Robert Shannon[1982, p 633][61], a respected authority in simulation, is as follows

'Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies for the operation of system '
A more recent definition according to [SIMAN simulation software, by Pegden, Shannon and Sadowski,1990][62] is as follows

'The process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system '
Due to [Villeneuve,et al 1988][63], Figure 2 1 shows the interpretation of this definition as the relationship between the real system and the simulation model

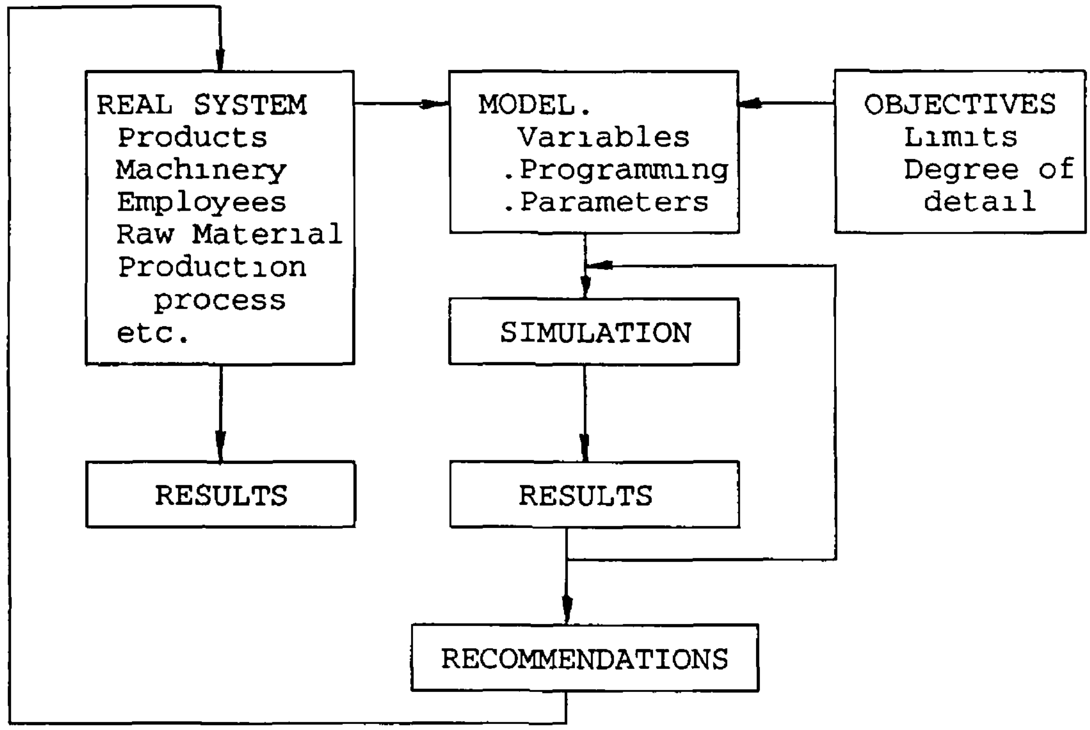


Figure 2.1: Comparison of real system and simulation model

Thomas et al[1966][64], trace the origin of simulation to the early sampling experiments of Gosset, W A , who published under the name Student [Student, 1908][65] However, the modern simulation techniques have been found through the works of [Von Neuman, 1951][66] His work involved the analysis of nuclear-shielding problems through a technique called " Monte Carlo analysis" which became the fundamental to simulation modelling Thomas and DaCosta, 1979[67] noted in their survey that simulation is mainly applied to the following areas

* Analysis of Commercial Air Transportation Systems
* Analysis of Computing Facility Operations
* Military Operations Analysis
* Evaluation of Machine Replacement Policies
* Nuclear Fuel Cycle Analysis
* Management Gaming
* War Gaming
* Environmental Impact Analysis

* Forest Resource Management
* Corporate Planning
* Machine Requirements Planning
* Evaluation of health Care Delivery System
* Manpower Planning
* Flow & Job Shop Scheduling
* Instructional Modelling for Higher Education
* Transportation planning
* Communications Network Analysis
* Financial Analysis
* Production and inventory Control Analysis
* CAD / CAM / FMS-CIM

Also simulation has been applied to process application such as [Miller, 1987][68]

* Agriculture Chemicals
* Air Separation
* Desalination
* Fermentation
* Inorganic Chemical
* Metals
* Natural Gas
* Oil
* Plastics
* Power
* Synthetic Fuels
* shale and tar sands
* Water treatment

Today, due to [Taha, Hamdy, A][1988][69] simulation is a powerful tool for the analysis of a variety of manufacturing systems, such as production scheduling, inventory control, materials handling, flexible manufacturing, project scheduling, manpower allocation, reliability and maintenance

In point of view of production scheduling, Conway, Maxwell and Miller, 1967[5] had given an excellent introduction to the interrelation between queuing theory and stochastic scheduling Their researchers provide an excellent introduction to simulation in the context of the flow and job shop [1967,

Chapter 11]
Computer Simulation can be efficiently applied to the following four activities of production scheduling environment [Carrie,A,1988][70]

2.4.1 Computer Simulation of Flow-Shop (transfer line):

It is to investigate the extent to which inter-stage buffer storage can minimise the loss of output of the line due to break downs at work stations [Buzacott][71] Due to Conway, Johnson and Miller,1959[72] an experimental investigation was carried out by means of a queue network simulation program for the Burroughs 220 Through their investigation on pure flow shop (is a shop in which there is only one path, that work can follow - each machine has a fixed predecessor and successor), separate runs were made for the number of jobs equal two, four and six times the number of machines Also comparisons between priority rules were made on the same set of sample jobs (2000 jobs per sample) The results showed that, the shortest operation rule was better for every shop size with respect to idle time%
Hon, k k and Ahmad, M M ,1985[73] in their study on transfer lines, they demonstrated that, computer simulation is a cost-effective method in the analysis of transfer line performance, and it can be applied for identifying the critical machine on the transfer line for machine replacement or refurbishment programme

2.4.2 Computer Simulation of Job-Shop:

A computer simulation in job-shop environment deals in particular, the effectiveness and assessment of different priority rules on the shop's ability to achieve near-optimum solution for many criteria There are many articles and books which have been published on the simulation of the job-scheduling problem Conway, Johnson and Maxwell [1959][72] had carried out an experimental investigation on job shop for the Burroughs 200 by means of a queue network simulation programme The computer simulation carried out separate replications for the number of jobs (the sample size at least 2000 jobs) equal to two, four and six times the number of machines Comparisons between priority rules (random and shortest operation time) were made on the same set of sample jobs with regard to the mean waiting time, system state and utilization They believed through the experimental investigation that, the shortest-operation rule deserve a further consideration and potentially was a great practical significance

Earl LeGrande[1963][47] has developed a factory simulation system using actual operating data belong to Hughes Aircraft Company, El Segundo Division The simulation process in his work was used as a study tool to evaluate the effectiveness of several priority rules with respect to the various criteria under constant conditions His simulation analysis shows that the SPT rule gives the best total relative rank if all criteria are weighted equally

Conway[1965a][52] reported a portion of the results of an investigation of different priority rules, in a job shop by means of computer simulation The criteria of comparison are various measures of work-in-process inventory The simulation experiments were executed on an IBM 650 and IBM 7090 and the programmes had been written in SIMSCRIPT[74] He noted that, the SPT rule under every measure clearly dominated all the other rules (RANDOM, FCFS, FASFS, LPT, TWKR, MWKR and FOPNR)

Moore, and Wilson[1967][75] had summarized the results of many digital simulation experiments seeking principles of scheduling design valid for job shops They pointed to the assumptions of the simulation model, such as job arrival, service time distributions, shop utilization, routing jobs, period simulation running and selected priority rules were discussed

Eilon, and Cotterill[1968][54] carried out a simulation study of a hypothetical shop with several machines under alternative priority rules His simulation was carried out on the IBM 7090 at Imperial College, the simulation came to an end when 5000 completed jobs emerged from the system The simulation model was under negative-exponentially distributed inter-arrival and processing times The main simulation running results were concerned with a comparison between the effectiveness of the S rules (giving preference to short operations), FCFS and L rules (which gives priority to long operation) They concluded that, the SPT rule performs best with respect to several criteria, but not for minimising the variance of throughput times (Make-Span) or missed due-dates Jones,1973[42] presented an economic framework for evaluating heuristic priority rules in the classic job shop situation In his study, simulation trials demonstrated the possible existence of cost structures which favour priority rules other than shortest processing time

Philip, et al[1984][58] studied a computer simulation on job shop scheduling They described the effectiveness of different priority rules under various workload conditions The mean flow time per job had been used as the measure of performance with respect to SPT, FCFS, FISFS and LCFS rules The SPT rule was the best performer for all levels of shop workload

Ramesh and Cary[1990][76] have developed a framework for the efficient job-shop scheduling considering the flow time, lateness and number of late jobs as the main criteria They have developed scheduling strategies to the man-machine approach as well The algorithms developed in their research tested and evaluated against the traditional scheduling methods using simulation studies Through 10 observations of simulation results jointly with ANOVA analysis, show that the scheduling algorithms, due date rule and the processing time variances and their interactions significantly affected the performance measures

2.4.3 Computer Simulation of Flexible Flow-Shop:

In this group of flexible shop, due to Wittrock[1985][77] a scheduling flexible flow-line can be defined as follows Several part types must be produced each days There are several banks of identical machines Each part must be processed by at most one machine in each bank Each part visits the machine banks in the same order There is a buffer which has a large capacity and operates There are machines to load and unload parts into and out of the system Finally, there is an automated transport system to move parts from one machine to another In addition to general-purpose machines, it can contain special-purpose machines, robots and some dedicated equipment [Browne,et al 1984][78]

A number of researches have dealt with computer simulation for the operation of flexible flow-shop Some of these researches have performed by large industrial concerns Hanifin[1975][79] used GPSS as a simulation language to develop an automated flow line systems for actual transfer lines at Kokomo Work of Chrysler Corporation His computer model was based on the operation of these machining flow lines The model was developed to deal with several specific problem areas at Chrysler His investigation considered the effect of adding up three storage buffer areas of three specific location along the line He also investigated the effect of different average tool change times

Due to Buzacott, et al[1978][80] in their research on simulation running on flexible transfer lines He concludes that each main factor affecting the transfer line output such as inter-operational stock, could be studied individually and furthermore, interaction effects could also be examined in detail

Flexible Assembly line is a class of transfer line Koenigsberg and Mamer,1981[81] are the authors who have dealt with flexible assembly systems in more details [Buzacott and Yao 1982][82] and [1986[83] They considered an assembly system consisting of a work transporter to feed work stations and a carousel conveyor Using simple queuing theory results they analyzed each component of the system - the loading/unloading of the carousel conveyor used for sorting work-in-process, dispatching work onto the work transporter and the processing of work at each work station. such assembly systems, which involve work transporter and central storage, appear to offer promise in overcoming some of the problems of conventional assembly lines. Because the flexible assembly systems are considerably cheaper than FMSs [Riley and Yarrow,1983[84]. It is possible that FASs may be adopted more rapidly. FASs rely on human intervention for the release of jobs and initiating movement of jobs between stations, although the dispatcher is provided with detailed information on the status of all jobs on the system. Thus, it is desirable to investigate simple priority rules and with the performance of the dispatcher.

Lay and Schiefele[1985][85] reported a simulation model for a flexible assembly system using SLAM II as simulation language. The aim of the simulation study was to arrange the assembly system in a way that the used resources (e.g work stations, workpiece pallets etc.) contribute effectively to a high productivity rate.

O'Gorman, Gibbons, and Browne, J.[1986][86] described a Simulation SLAM based model of hypothetical Flexible Transfer Lines. The study had compared the SPT, LPT and FCFS rule as well as Johnson's algorithm with respect to total throughput time. They got an important concept that a simulation language is a powerful tool in the evaluation and development of FMS.

2.4.4 Computer Simulation of Flexible Manufacturing Systems:

The fundamental definition of an FMS is, in the words of [Buzacotte and Shanthikumar,1980][87], **"A flexible manufacturing system (FMS) consists of machines where production operations are performed, linked by a material handling system and all under central computer control"**.

In United States Office of Technology Assessment concept,

"A flexible manufacturing system (FMS) is a production unit **capable of producing a range of discrete products with a**

minimum of manual intervention. It consists of production equipment workstations (machine tools or other equipment for fabrication, assembly, or treatment) linked by a material-handling system to move parts from one workstation to another, and it operates as an integrated system under full programmable control" The use of simulation in the design and control of FMSs is widely accepted around the world It offers the most fascinating production method for the computer controlled factory Their use allows one to [Ranky,1986][88], [Greenwood Nigel R ,1988][89] and [Carrie,1988][70] achieve the following

* Increase in productivity (often by 25%)
* Decrease in production cost (often by 50%)
* Manufacture parts on order, rather than to stock them in inventory
* Decrease in inventory and Work-In-Process to a lower level
* Save at least 30% of labour
* Improve equipment utilization by at least 50%
* Reduce floor space by at least 50%
* Provide 100% inspection, thus increasing the quality of the product
* Decrease the amount of often repetitive, or hazardous physical work and increase the need for intelligent human
* Provide a reprogrammable, almost unmanned manufacturing facility

Due to [Ranky,1986][88],in point of view of computer simulation in FMSs scheduling concept, the major benefit is that

* The overall planning level can utilize the scheduling system of the CIM business data processing system
* The FMS loading sequencing programme can be a relatively simple and fast "n" job, one processor scheduler (the single processor being the whole FMS, as a system)
* The FMS dynamic schedule can be a single processor, but applied for each of those cells on which the component is going to be processed every time and immediately after the

disruption occurs in the system

In terms of the analytical models of simulation FMSs, a review reported by Buzacott, and Yao,1986[82] was organized around the research groups as follows

- **Purdue**
- **Draper Labs**
- **MIT (Massachusetts Institute of Technology) or (LIDS) (Laboratory for Information and Decision Systems)**
- **Harvard**
- **France**
- **Toronto**

Purdue

The basic analytical model of this group has been CAN-Q developed by Solberg,1977[90] The system is modelled as a closed queuing network, in which the customers are the jobs to be processed by the system, and the servers are the machines At all stations, the FCFS rule is a queue priority rule, the service time distributions are exponential, all jobs will never be blocked at any stations, machines are always available for processing at any stations The throughput of the system is defined as the throughput of the load/unload at which jobs enter and leave the system This model has been widely used for preliminary design of FMSs and studying some of the issues in production planning However, the model will not in general yield satisfactory performance evaluations if, with FCFS rule, the service time distribution are not exponential

Draper Labs

Hildebrant[1980][91] was the first to consider the overall production planning and control problems of FMSs He classified decisions into two types resource and temporal decision, and he used different levels for resources in which he finds the mix, sequence, and input time for jobs Software tools and some heuristic rule are developed on each long, medium and short term decision making The aim of this work is to develop a decision support system to aid in decision making, regarding an FMS on three level of terms

MIT (LIDS)

This group's work on FMS, is part of a large research project which includes transfer lines, assembly/disassembly networks as well as FMSs [Gershwin,et al 1981][92]

In FMS modelling [Kimemia and Gershwin,1985][93] used the closed queuing network to study the optimal routing/loading of an FMS The objective is, to minimize the production rate This objective was studied through a detailed simulation model of an IBM printed circuit card[93] (or board) assembly facility It was found that, the optimal policies generated by the model are superior to other policies in terms of smoothing production against disruptive events such as repairs and failures

Harvard

The group presented the FMS as queuing network with general servers and a limited storage space at the stations For a particular set of parameters, the network studied by using simulation Then in order to optimize some criteria, the central problem is to derive the corresponding gradients, i e , it is to study the sensitivity of the system performance to its parameters An approach called "perturbation analysis" was developed for this purpose by Ho,1984[94] The basic idea of perturbation analysis is to observe a given sample path (nominal path), obtained from a detailed simulation and to consider a question related to the occurrence of a specific event in the nominal path, which were perturbed Through this perturbed path, it could be known that it is effected on the interested system criteria

France

In modelling FMSs, Cavaille and Dubois,1982[95] waive the exponential assumption of the closed queuing network model The starting point is the following relation, known as the mean value equation (assume single server station)

$$W(N) = \mu^{-1} + \mu^{-1} \quad Q(N - 1)$$

The equation says that in a close network of queues with population N, the mean delay of a job at a station, $W(N)$, is the sum of its own mean service time (this service time is nearly deterministic) at that station μ^{-1}, and the mean time to complete serving all other jobs which are already at that station Here $Q(N - 1)$, observed at the job's arrival point, is to be evaluated in a system with one less job $(N - 1)$ This fact is known as the arrival-point theorem [Sevcik and Mitrani [1981][96] an arriving job observed the behaviour of a network with itself excluded It plays a critical role in constructing the MVA algorithm (the Mean Value Analysis)

Toronto

Buzacott,1982a[97] was the first researcher reported about FMSs performance modelling for this group He emphasised on (i) limited local storage capacity, and (ii) general service-time distribution Three basic hierarchical decision structures were focused

Pre-release Planning Deciding which jobs are to be processing, identifying constraints on operation sequence

Input Control Determining the sequence and timing of release of jobs to the system

Operational Control Ensuring movement between machines and deciding which job is to be processed next by a machine

Also [Yao and Buzacott,1986a][98] developed a general service time problem approach The idea of this approach is to replace the general network by an (approximately) equivalent exponential network, where each station is characterized by a set of state dependent service rates This approach gives accurate solutions to general networks and also recovers the classical product-form models in the case of exponential service times

The analytical simulation model approaches that have been developed by various groups was discussed. These approaches enable variety of issues connected with FMSs design and operation

Some of the more recent research on computer simulation of FMSs scheduling have been reported by

Aanen, Galman and Nawijn[1989][99], They studied a real-life FMS shop at the Dutch Institute of applied physics of TNO and the Institute of metals TNO (Apeldoorn) One of the objectives of this system is to produce a wide variety of parts in small batches Due date, routing, capacities of the machines and the tool magazines, tool and jaw changing times, limited fixture capacity, fixturing and clamping times and limited number of operators and transport devices had to be taken into account

Niemi and Davies[1989][100], noted through their research that the maximum utilization of an FMS implies optimum job sequencing and effective method for programming the different computer controlled elements of the system The research described simulation of an intelligent cell control system for a robot served FMC (flexible manufacturing cell), where job sequencing is based on demand from the cell and user-set priority

Montazeri and Wassenhove[1990][101] have discussed the characteristics of a general-purpose, user-oriented discrete event simulator for FMS The performance of a number of priority rules were subsequent analyzed using their modular simulator to mimic the operation of a real-life FMS Results showed that, priority rules had a large impact on various system performance measures, such as ave machine utilization, ave WIP and ave buffer utilization Considering the high investment costs of FMS, it is certainly worthwhile to choose the best priority rule by use of simulation They concluded that SPT rule performs quite well with respect to ave waiting time per part and ave buffer utilization While LPT (longest processing time) rule showed good results with respect to machine utilization

Muller, Jackman and Fitzwater[1990][102] have discussed FMSs in terms of interfaced with the real-time control database so that initial conditions could be determined In their research they discarded the transient simulation running times that to

be before with a steady state Simulation results provide analysis with information, to make improvements in the short term schedule with better work order release decisions
Emelyanov, Gendler and Felman[1990][103] reported a survey on FMSs They attended to the concept, kinds and indicators of FMSs Their paper gave fifty references as well
In point of view an FMSs simulation-economic analysis, [Boër and Metzler,1985][104] concluded the economic relationship between different manufacturing systems as shown in Figure 2 2 Also they noted that the important operational costs, can be evaluated only, if a simulation analysis is performed

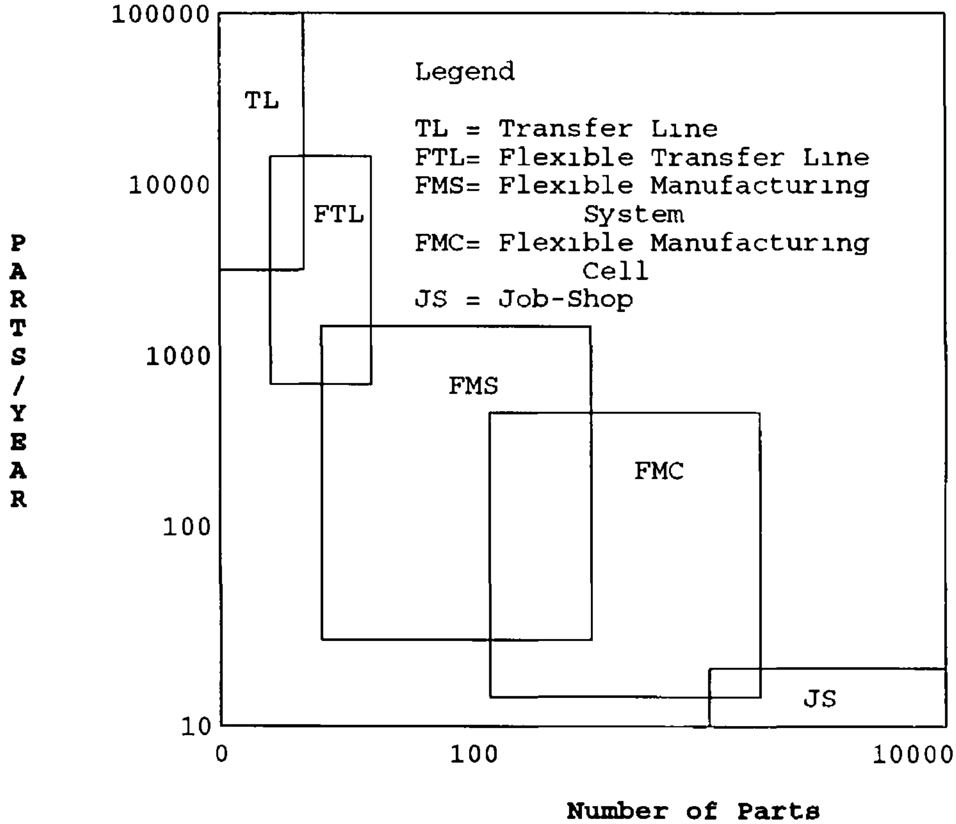


Figure 2.2: Economical Manufacturing Concept.

In brief the simulation model serves as the designer's experimental laboratory allowing him to determine the system performance in response to changing conditions This helps to minimize the risk associated with an FMSs and insures that, it delivers the required results

2.5 SIMULATION LANGUAGE REVIEW:

2.5.1 Overview:

One tool which can aid the process of rapidly matching production responses to strategic and operational objectives such as production scheduling is computer simulation
With computer simulation, a model of the system under study is constructed using a simulation language This language gives structure to the model building process, by providing special modelling constructs, that relate to the system under study
Fortunately, there are many different "languages" available for building computer simulation models The majority of industrial simulation languages deal with discrete event simulation
Due to Taha[1988][69], discrete event simulation, in which observations are gathered only at selected points in time, when certain changes take place in the system
Problems concerning resource allocation, job sequencing (Thesis's aim), material handling, queuing, transportation, etc , are best handled by discrete event simulation
On the other hand, continuous simulation requires that observations be collected continuously at every point in time
Examples are radioactive processes, chemical reaction processes, heating and cooling processes, etc

More recent simulation languages provide both discrete and continuous variable simulation capabilities such as GASP IV,1974[105](USA), SLAM II[106](USA), SIMAN[62](USA), WITNESS[107](UK), ECSL[108](UK), GPSS,1972[109] and Taha,[69]
General purpose computer programming languages, such as Pascal, fortran, C, Lisp, and others, can be used to develop simulation models Many simulation languages provide also an interface to a general purpose programming language This allows the user to develop special purpose functions and/or routines required for a particular model
Most of the simulation software packages, e g , SIMAN[62](USA),

SLAM II[106](USA), HOCUS[110](UK), BEAM[111](USA), WITNESS[107](UK), and others, provide graphic aids (an animated scene) for model development, and display of the results For many problems, a graphic display can be a very useful aid in viewing system operation [Hurrion,1978][112] and [Grant and Weiner,1986][113]

2.5.2 Commercial Simulation Software Review:

Grant and Weiner[1986][113], addressed that in the United States there were about 500 animated simulation systems installed with compared to less than 10 in 1982 Carrie[1988][70], reported a historical development of more than 30 packages in the *UK* and *USA* In Figures 2 3 and 2 4 he linked the simulation packages in the UK and USA into two family trees respectively The following review will be devoted, and briefly discusses some of the most well-known simulation software

1. GASP IV:

Due to Pritsker[1974,p 16][105], "a GASP IV programme is made up of (FORTRAN) sub-programmes linked together by an executive routine, that organises and controls the performance of the sub-programmes" Specifically, GASP IV includes routines to perform the following tasks

* Time advance and status update
* Initialization
* Data storage and retrieval
* Location of state conditions and entities
* Data collection, computation and reporting
  Monitoring and error reporting
* Random deviate generation
* Various miscellaneous routines

The analyst needs to be fully conversant with the routines provided in order to make proper use of these collections However, it is easier than writing from scratch in FORTRAN and similar problem oriented languages

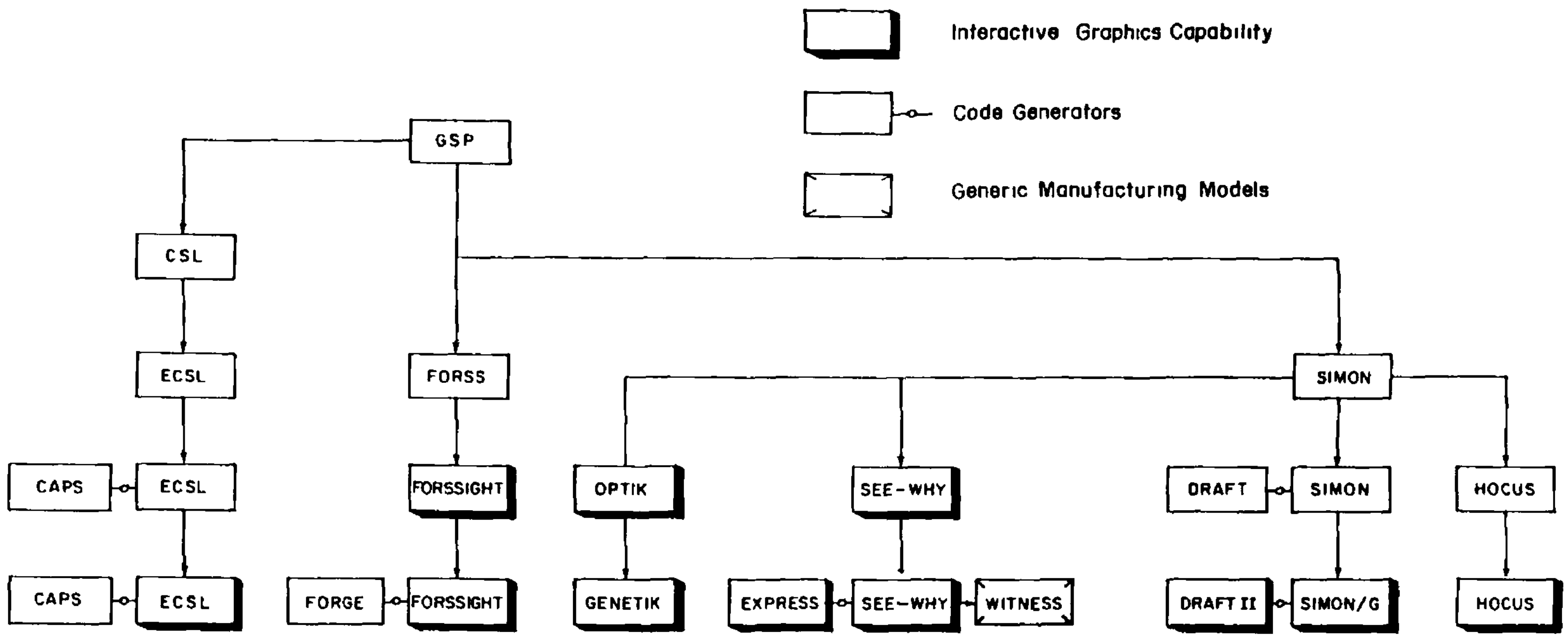


Fig 2 3 Historical Development of simulation packages in the UK

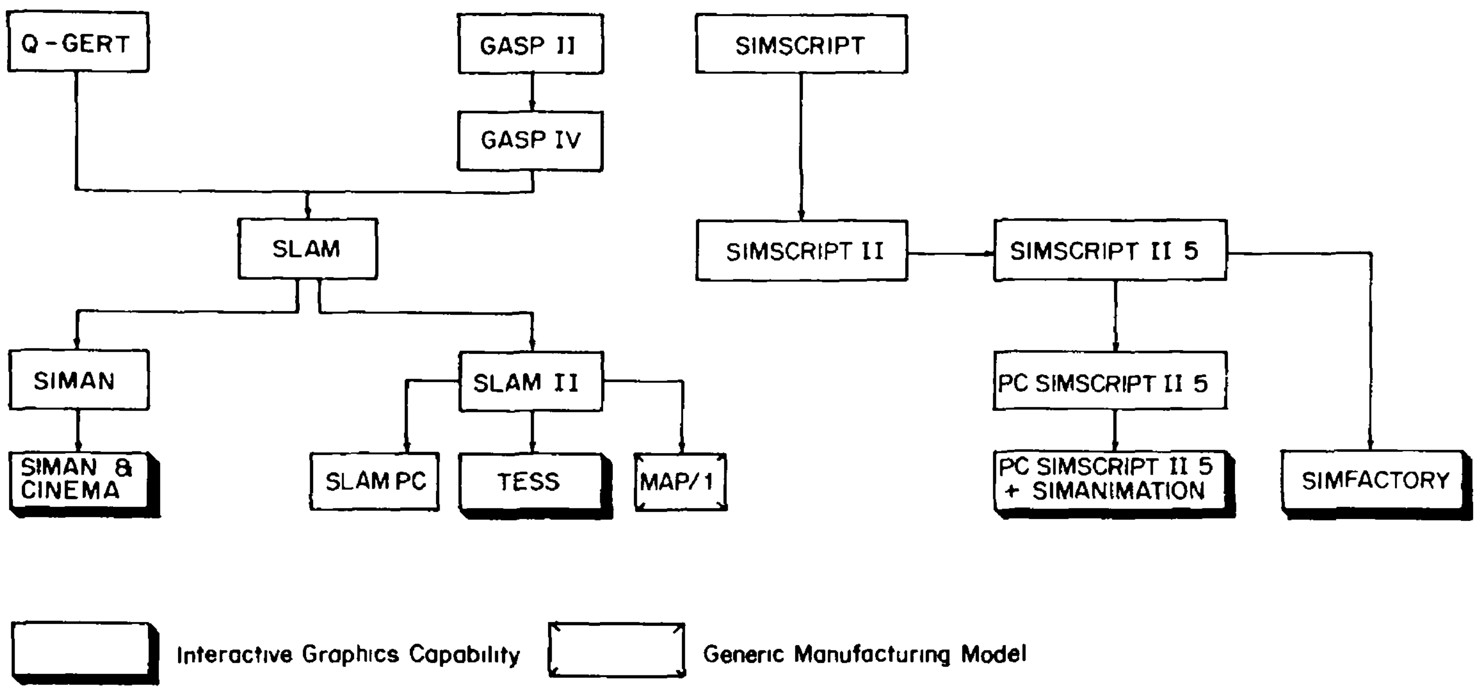


Fig 2.4 Historical Development of simulation packages in the USA

2. ECSL:

Extended Control and Simulation Language [Clementson,1985][108] applies the simple activity scan approach ECSL code is interpreted, the ECSL system and the interpreter being written in FORTRAN Thus ECSL may run on any machine which supports a FORTRAN compiler and offering sufficient storage ECSL programme has the following sections

* Definitions
* Initialization
* Activities
* Finalization
* Data

ECSL provides sampling routines and random number streams It is entirely oriented towards programming simulation problems However, because of the interpreter, this is possibly at the expense of efficient execution Despite being interpreted, an ECSL programme can not be stopped and restarted in the same way as most BASIC programmes

3. WITNESS:

WITNESS [AT&T Istel Ltd,1991][107] is probably one of the most developed of the generic manufacturing models It has been used in several non-manufacturing environments The basic elements of WITNESS are parts, machines, conveyers, buffers, labour, vehicles and tasks To create and run a model, WITNESS provides three phase guided by menus and prompts These phases are

* Define the operational elements
* Detail the operating characteristics of the elements
* Display the elements in the model

4. GPSS:

Due to Pidd, 1988[114], the GPSS (General Purpose System Simulator)[109] is the best-known block diagram system for

simulation GPSS is based around the idea of a block diagram which models the flow of entities through a network
Consider a single server queue, GPSS being ideal for such simulation A skeleton GPSS program might look as follows

* Generate
* Queue
* seize
* Depart
* Advance
* Release
* Terminate

GPSS has an appealing simplicity Hence it has an obvious application for simulating systems in which the entities follow relatively predictable paths in which their interaction is slight However, GPSS has relatively poor creation for random number generator, thus it could lead to sampling errors

5. SIMAN:

SIMAN[1991][62], from System Modelling Corporation, is a general purpose, microcomputer based and animation system It is first which used to build the simulation model of the system Then CINEMA is used to construct an animation lay-out, which is graphically depicting the physical components of the system being modeled Then the SIMAN simulation model is executed in conjunction with the CINIMA lay-out to generate a graphical animation of the system dynamics
More details concerning SIMAN software, will be discussed through out Chapters 6 and 7, in this thesis

CHAPTER THREE

3. PRODUCTION SEQUENTIAL SCHEDULING:

3.1 Introduction:

Every Industrial organization has a number of scheduling problems The production sequential scheduling is the most important problem of scheduling encountered in production planning and control, yet it is at the same time the main factor in estimating the production cost in a factory

This chapter will be devoted to machine scheduling problems

The magnitude of the problem can be illustrated as follows

1- Consider a given number of jobs each of which requires one or more operations An operation is the processing of a specific job, through a specific machine (processors or facility), it is important to determine the starting time of the operations as well

2- Job sequencing or job scheduling consists of determining the order or sequence in which the machines will process work so as to optimise some criteria The selection of the criterion in a particular case will depend on the individual requirements of the decision maker

In terms of production cost in the factory, estimating the cost of a part is closely linked to the efficient sequencing of the job through manufacturing lines The most important criteria of the cost involved through job sequencing are make-span (total completion time), machine idle time (machine utilization), waiting time for jobs (work-in-process), mean completion time of job and job lateness

3.2 Description of A General Machine Scheduling Problem:

Machine scheduling problems can be usefully stated as sequencing, a set of entities which pass through the shop are called n jobs (J_1, J_2, , J_n) and a set of works done on them at m machines (M_1, M_2, M_m) are called operations (tasks) m_j ($o_{j\,1}$, $o_{j\,i}$, . , $o_{j\,m_j}$) These operations to be performed in a strict technological sequence which is called a routing, where j-1, ,n is the job number and i-1, , k_j is the position of the operation in the sequence Each job has a ready time or release date r_j to be available for processing, and must complete processing by d_j, the due date of job J_j Each operation $o_{j\,i}$ requires a specific machine M_s for processing within a duration $P_{j\,i}$ is called the processing time of the operation Thus the job sequencing can be defined as the ordering of the operations on jobs at the machines This job is undergo to routing or technological constraints, so that the best value is obtained for some of criteria appropriate to the system For general job-shop problems there are no restriction upon the form of the technological constraints Each job has its own processing order and this has no relation to the processing order of any other job However, an important special case arises, when all the jobs have the same processing order This kind of shop is called a flow-shop problem

Geometrically the job sequencing ordering which can be produced as a projected time-table is called a Gantt chart, [Henry, L Gantt [1918] [115]

The most of scheduling researches report a typical machine scheduling problem with ready times equal zero (this is the static scheduling problem) and no due dates The criterion is to minimize the maximum time to complete all jobs (make-span or C_{max}) Hereafter, Figures 3 1(a, b and c) and 3 2(a and b) respectively show two simple data, job sequencing, feasible solution Gantt charts for deterministic job-shop and flow-shop problem

Fig 3 1a Processing Times for each job on each machine

		Processing Time		
Machines		M1	M2	M3
Jobs	J1	4	2	7
	J2	3	5	6
	J3	2	4	3

Fig 3.1b· Job sequencing of processing jobs on machines

Processing Sequence		1st	2nd	3nd
Jobs	J1	M3	M1	M2
	J2	M2	M3	M1
	J3	M2	M1	M3

Fig 3 1c Feasible Solution Gantt chart

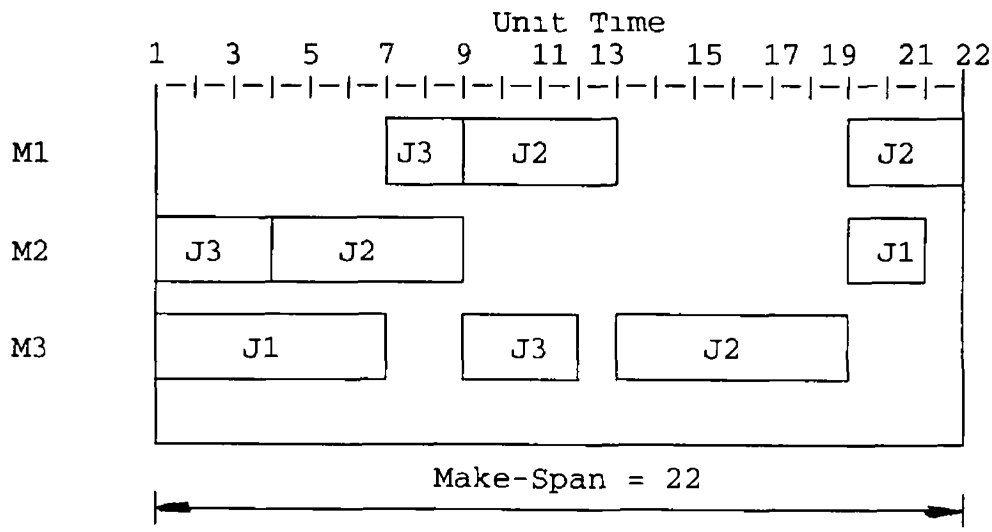


Figure 3.1 (a, b and c): 3-job 3-machine job-shop scheduling problem.

Fig 3 2a Processing time for each job on each machine

	Machines	1	2	3
Jobs	1	5	6	3
	2	4	3	4
	3	3	3	3

Fig. 3 2b Feasible solution Gantt chart

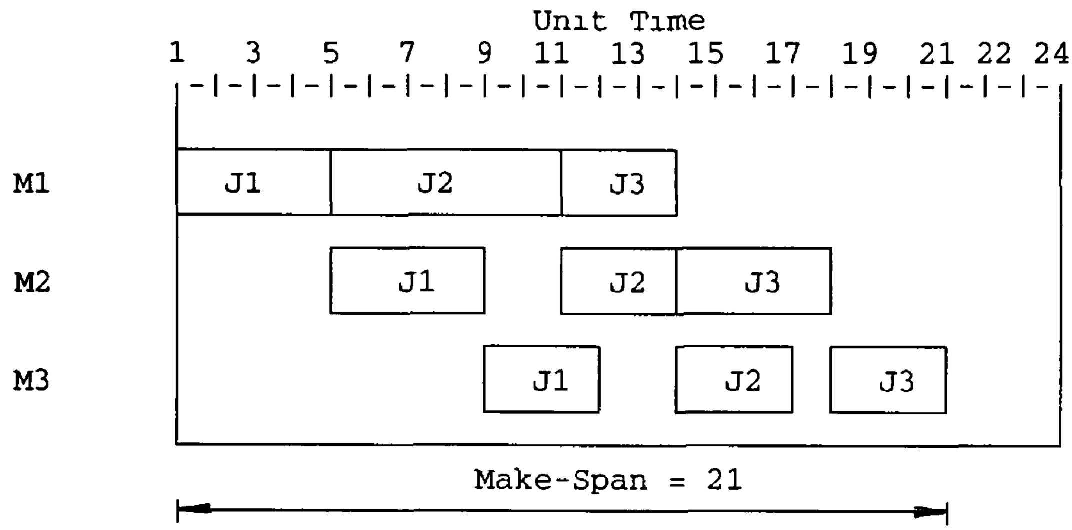


Figure 3.2 (a and b): 3-job 3-machine flow-shop scheduling problem.

3.3 Restrictive assumptions:

The more apparent statement of the machine scheduling problem with zero ready times specifies a number of restrictive assumptions These assumptions were noted throughout the literature of [Rinnooy Kan,1976][7] and [French,1982][28]

3.3.1 Restriction on the machines:

Unless stated otherwise, the following restrictions are to be placed on the machines

M1 The number of machines m is known and fixed (a deterministic problem)

M2 All machines are available at the same instant and are independent

M3 All machines remain available during an unlimited time period (breakdown not allowed) However, this assumption is stated in Chapter (7) where a machine required maintenance at a certain period of time and therefore no job can be processed during that time

M4 Each machine m ($M_1, M_2,$, M_m) is either waiting to process the next job, operating on a job or having finished its last job

M5 All machines are equally important That is their speeds of processing are the same

M6 Each machine has to process all jobs assigned to it (a deterministic problem)

M7 Each machine can process not more than one job at a time

3.3.2 Restriction on the jobs:

Unless stated otherwise, the following restrictions are to be made on the jobs

J1 The set of jobs are known and fixed in advance

J2 All jobs are available at time zero and independent However, this assumption does not often hold, see Chapters (6 and 7), where each job has a release date r_j which previously not available

J3 All jobs remain available during an unlimited time period However, this assumption is stated in Chapters (6 and 7) for where each job requires due date d_j, that is a time by which processing should be completed

J4. At any instant of time, each job is either waiting for the next machine, being processed or processing is complete on its machine

J5 All jobs are equally important (W_j = 1 for all j=1,2,3, ,n), where W_j is denoted to weight assigned to job j (the relative important of each of these job n)

J6 Each job must be processed by all the machines assigned to it (a deterministic problem)

J7 Each job is processed by one machine at a time

J8 Preemption is not allowed That is each operation once started has to be completed without interruption This assumption is relaxed in the case of a lower bound being obtained

J9 All processing time include any set-up and tear-down time and fixed and independent-sequence Baker[1974][6] considered this problem in more terms

Conway et al[1967][5] gave more descriptions for the stochastic nature of some scheduling problems Such type of problems are not included in the M1, M2, J1 and J2 This random data is stated later in Chapters (4, 5, 6 and 7)
In brief, these assumptions were mentioned during two previous items jointly with the choice of criteria representing only a component of a schedule's cost Due to [French,1982][28], in practice it is the total cost that we wish to minimize

3.4 Scheduling Costs and Measure of Performances:

The objective in all scheduling problems taken into account in this thesis is to obtain an optimum or near-optimum job sequence, with respect to a given criterion This criterion is called the measure of performance (the objective function) They are numerous, complex, and often conflicting
Mellor[1966][116] lists 27 distinct scheduling goals
System costs, however, are often difficult to measure or even to identify completely Thus, the measure of system performance which are aggregate scalar quantities and which contain either

explicit or implicit information concerning all processing are referred to simply as performance measure [baker,1974][6]

3.4.1 Criteria based upon minimizing Completion Times:

The main criteria in this category are

1 Make-Span or maximum completion time Is the time to complete all jobs j (i e)

$$\max_{1<j<n} (C_j)$$

2 Mean Completion Time (i e)

$$C = (1/n) \sum_{j=1}^{n} C_j$$

3 Flow Time Is the mean of the time that J_j spends in processing (i e)

$$F_j = C_j - r_j$$

4 Mean Flow Time (i e)

$$F = (1/n) \sum_{j=1}^{n} F_j$$

3.4.2 Criteria based upon minimizing machine Idle Times or maximizing machine utilizations:

The Idle Time on machine M_i is equal to

$$I_i = C_{max} - \sum_{j=1}^{n} P_{ji}$$

Where C_{max} is the make-span and the second element of this equation is the total processing time on machine M_i Their difference gives the period for which the machine is idle Due to French[1982][28] the mean idle time, may be chosen to achieve maximum machine efficiency

3.4.3 Criteria based upon minimizing Inventory costs:

a. Waiting Times criteria:

a 1 The waiting time of J_j on machine M_i is the elapsed time between the completion of $o_{j\,i}$ and the start of processing of $o_{j\,j+1}$

a 2 The total waiting time of J_j is as follows

$$W_j = \sum_{i=1}^{m} W_{j,i}$$

b. Work-In-Process criteria[52]:

It is the amount of Work-In-Process(number of jobs)at time t

3.4.4 Criteria based upon Due Dates(minimizing of the Lateness and Mean Lateness of jobs):

If due dates have been assigned to jobs, and since the cost of schedule is usually related to how we miss target dates by, obvious measure of performance are

a The Lateness which defined as the difference between the completion time of J_j and the its due date ($L_j = C_j - d_j$)

b The Mean lateness which defined as followed

$$L = (1/n) \sum_{j=1}^{n} (L_j)$$

3.5 Problem Classifications:

In this section, each scheduling problem requires processing n jobs on m machines so as to satisfy the objective of the

criteria. Therefore, each scheduling problem has a well-defined set of jobs, machines and performance measures.
For this reason, scheduling problems are characterized by 4-parameter notation n/m/G/B [conway,et al,1965][5] and French [1982][28] to be defined below:

n is certain job characteristic(is the number of jobs).

m is the number of machines.

G is the machine environment (describes the flow pattern within the machine shop).

B is the optimality criterion (describes the performance measure by which the schedule is to be evaluated.

Table 2.1 (page 8) described most of the types of job-shop problems.

3.5.1 Open and closed shop problem:

The open shop problem in which each job j consists of a set of operation $\{o_{j1}, o_{j2}, .., o_{jm}\}$. But the order in which the operations are processed is immaterial. Also in an open shop environment no inventory is stocked, all production orders are by customer request and it means sequencing only, whereas in the closed shop problem the orders are fulfilled from an inventory and it means not only sequencing, but lot-sizing, consequently the manufacturing system produces part for inventory, rather than for customer.

3.5.2 A single machine problem (n/1/B):

The n/1/B problem is considered as follows:
Each of n jobs has to be processed without interruption through a single machine. The machine cannot process more than one job at a time. Each job j has a processing time P_j. Given any sequence of jobs the completion time C_j for job j can be obtained assuming that processing starts at time zero, in this case the make-span for all job sequences is equal. In this type of shop in which there is a single machine, the total number

of distinct solutions is therefore n!, which is the number of different permutation of n elements Also aggregate performance measures that might be defined includes the following [Baker,1974][6]

Mean flow time, Mean tardiness, Maximum Flow time, maximum tardiness and number of tardy jobs

The n jobs, single machine problem is very important in the case of loading sequencing the FMS, because the entire FMS can be considered as one single processor (machine)[88].

3.5.3 A Pure (or Permutation) Flow-Shop Problem (Figure 3.3):

We have a n/m/P/B problem in which each job j has the same sequence of operation (unidirectional), also all machines m have to handle the jobs in the same route as shown in Fig 3 3 The processing time of each job j on machine i, denoted by P_{ji}, is given Once a job has started on a machine it must be completed on that machine without interruption The objective is to find a job sequence that optimize the selected criterion

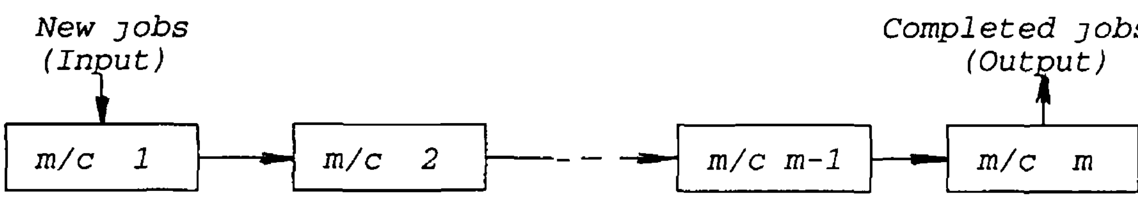


Figure 3.3: Job work-flow through machines in a pure flow-shop environment.

In a pure flow-shop problem we have (n!) different job sequences Table 3 1 (page 48) shows the possible sequences for up to 10 jobs to be processed on m machines

Number of Jobs	2	3	4	5	6	7	8	9	10
Possible Sequences	2	6	24	120	720	5040	40220	362880	3628800

Table 3.1: Possible number of sequences of up to 10 jobs for a pure flow-shop problem.

3.5.4 A Flow-shop problem (Figure 3.4):

We have a n/m/F/B problem, There are n jobs to be processed on m machines. Each job j has the same sequence of operations, but some jobs may overtake other job through some machines (i e, the machines may handle the jobs in different orders) as shown in Fig 3 4 Also each job j has a processing time P_{ji} on machine i. once the processing of a job on a machine has started, it must be completed without interruption

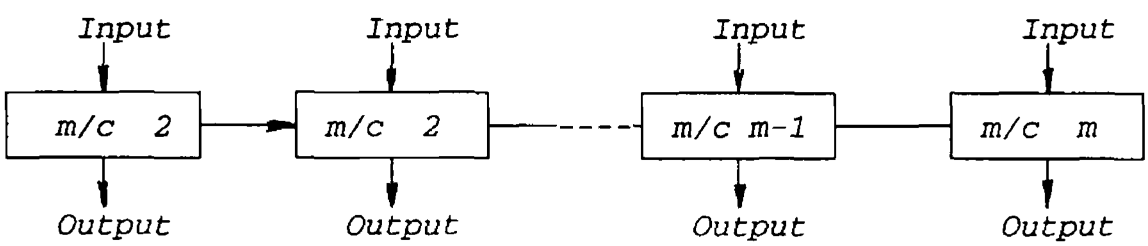


Figure 3.4: Job work-flow through machines in flow-shop environment.

In flow-shop problem there are (n!) different job sequences possible for each machine, and therefore $(n!)^m$ different schedules to be examined

Due to Ranky[1986][88], in terms of production methods, both of a pure flow-shop and flow-shop problems mainly applied in the cases of transfer lines (assembly lines) and flexible flow-lines These methods of production are sufficient and very productive, but they are inflexible and require large batch sizes to offer an economic solutions The objective is to find a job sequence on machines that optimize the selected criterion

3.5.5 A Job-Shop problem (Figure 3.5 and 3.6):

We have a n/m/G/B problem in which each job j has a specified number of operations $\{O_{j1},\ ,O_{jm_j}\}$ of other jobs In other words there are n jobs waiting to be processed on m machines and the order of jobs is not the same or unidirectional Because the work-flow in a job-shop is a multi-directional type of the flow, each machine in the shop can be characterized by the input and output flows of work shows in Figure 3 5 below

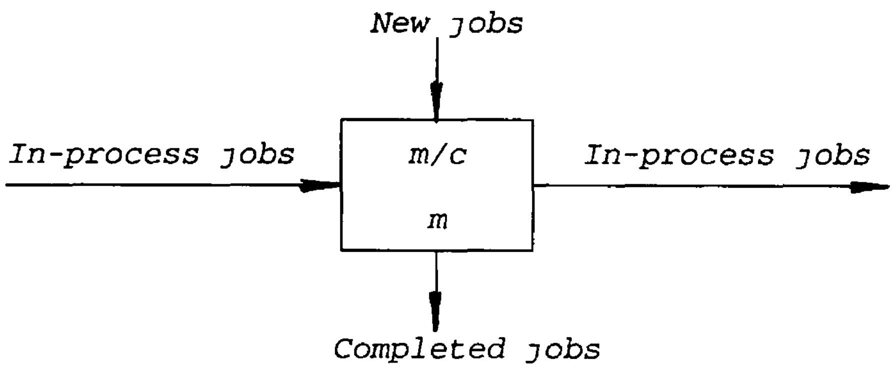


Figure 3.5: Work-flow at a typical machine in a job-shop.

For scheduling purpose the information that is needed from the process lay-out is the time required and the order in which the operation jobs are to be carried out on the specified machines The objective is to determine a job sequence which subject to restrictions on the order in which the job can be performed, will optimize the selected criteria

Figure 3 6 describes the nature of job flowing through a job shop environment

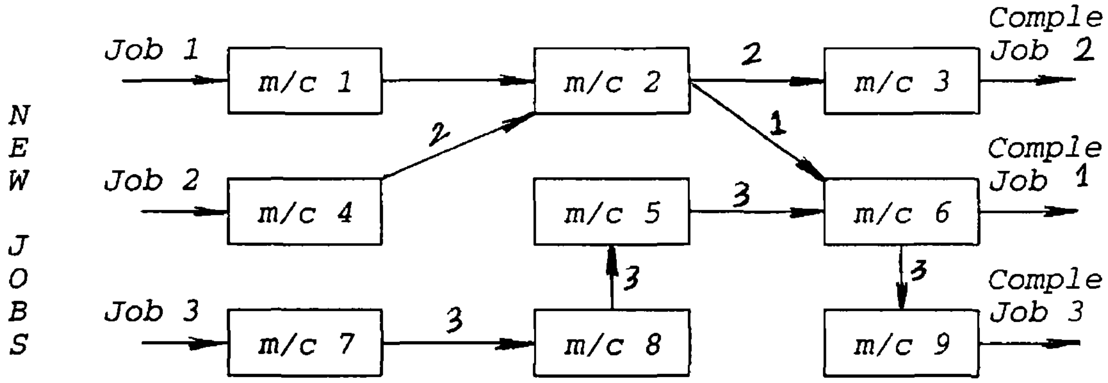


Figure 3.6: Job flowing through the machines in 3-jobs 9-machines job-shop.

The most important features of the job-shop problem are the following points:

a It can handle a variety of jobs at the same time, this is a flexible situation since the jobs can be different and there are no restrictions on their routing

b The machines are shared by different jobs

c Different jobs or batches can have different priority

d As results the procedures and outputs are equally applicable to all types of intermittent production systems

The major disadvantages of the job-shop scheduling method is that it is off-line, since it applied for a fixed period of time In another hand job arrival and real-time changes cannot be accurately planned because of the lake of an overall material handling and real-time operated computer control system

3.5.6 Nature of the requirement specification and scheduling environment:

a. Deterministic and Stochastic Requirement:

The scheduling problem is called deterministic if the number of jobs and their ready times are known and fixed In the stochastic problems the job file (processing times, job sequence, due date and arrival times) is uncertain

b. Static and Dynamic environment:

Because the processing times and all other parameters are known and fixed, the scheduling problem is called static Whereas the problems in which jobs arrived randomly over a period of time are called dynamic

Due to Ranky[1986][88], Most scheduling to be studied are deterministic and static, in other words were developed as if the manufacturing environment was static and its behaviour "fully known" for at least a finite length of time, whereas in real life, manufacturing systems are stochastic and dynamic Unfortunately, scheduling theory and practice are far apart and many mathematical models do not work in practice
In this thesis, we shall apply the two kinds of requirement and environment scheduling problem through Chapters(4, 5, 6 and 7)

CHAPTER FOUR

4. PRODUCTION SCHEDULING STUDY FOR THE OPTIMIZATION OF THE FLOW-SHOP PROBLEM:

4.1 Presentation of n/m/P/B problem:

In this chapter we consider the general permutation or pure flow-shop problem under precedence constraints This problem, indicated by n/m/P/B, can be described as follows

There are n jobs numbered 1,2,3,   n, and m machines numbered 1,2,3,   ,m, each job j(j=1,   n) has to be processed through the m machines in the same order and the skipping is not allowed
The processing time of each job j through each machine i, denoted by P_{ji}
Once a job has started through a machine it must be completed through that machine without interruption
The criterion for optimization in this pure flow-shop is to find a job sequence that minimizes the maximum completion time

$$(B = C_{max}) \text{ or (make-span)}$$

which is the elapsed time between the first job being started on the first machine and the last job being completed on the last machine
In this type of job sequence in which the order of jobs is the same on all the machines, so that if an order is decided or chosen for the first machine, then this will be maintained through all the following stages
This type of problem generates, for n jobs, n! job sequences

or feasible schedules

4.2 The Throughput Time for a Schedule:

As pointed out in Section 3 2, Figure 3 1c and 3 2b, the Gantt chart could be used to produce a workable schedule, then the job completion times may be determined direct from a Gantt chart presentation This method is only useful for a limited number of (n X m) pure flow-shop problem, because the chart will be confused too much Another solution choice, in particular for computer applications, due to King[117] this make-span may be determined directly by analytical permutation as stated in Appendix A

Hence the throughput time to complete the total schedule C_{max} is give by the following formula

$$C_{max} = F\{q(n,m),m\}$$
$$=max[f\{q(n-1,m),m\},f\{q(n,m),m-1\}]+ t\{q(n,m),m\} \qquad (4.1)$$

The significance of this analytical permutation is that, in $n/m/P/C_{max}$ problems it could be determined by any job feasible sequence value for n' permutations

Hereafter, the application work for the above procedure is illustrated Assume we have a pure flow-shop with four identical machines which process five type of jobs

The (5 X 4) problem matrix for Processing time to complete each operation of each job (set-up and tear-down are included in the processing time) and the machine descriptions are shown in Figures 4 1 and 4 2(a, b, c and d) respectively

CHAPTER FOUR

4. PRODUCTION SCHEDULING STUDY FOR THE OPTIMIZATION OF THE FLOW-SHOP PROBLEM:

4.1 Presentation of n/m/P/B problem:

In this chapter we consider the general permutation or pure flow-shop problem under precedence constraints This problem, indicated by n/m/P/B, can be described as follows

There are n jobs numbered 1,2,3, n, and m machines numbered 1,2,3, ,m, each job j(j=1, n) has to be processed through the m machines in the same order and the skipping is not allowed
The processing time of each job j through each machine i, denoted by P_{ji}
Once a job has started through a machine it must be completed through that machine without interruption
The criterion for optimization in this pure flow-shop is to find a job sequence that minimizes the maximum completion time

$$(B = C_{max}) \text{ or (make-span)}$$

which is the elapsed time between the first job being started on the first machine and the last job being completed on the last machine
In this type of job sequence in which the order of jobs is the same on all the machines, so that if an order is decided or chosen for the first machine, then this will be maintained through all the following stages
This type of problem generates, for n jobs, n! job sequences

or feasible schedules

4.2 The Throughput Time for a Schedule:

As pointed out in Section 3 2, Figure 3 1c and 3 2b, the Gantt chart could be used to produce a workable schedule, then the job completion times may be determined direct from a Gantt chart presentation This method is only useful for a limited number of (n X m) pure flow-shop problem, because the chart will be confused too much Another solution choice, in particular for computer applications, due to King[117] this make-span may be determined directly by analytical permutation as stated in Appendix A

Hence the throughput time to complete the total schedule C_{max} is give by the following formula

$$C_{max} = F\{q(n,m),m\}$$
$$=max[f\{q(n-1,m),m\},f\{q(n,m),m-1\}]+ t\{q(n,m),m\} \qquad (4.1)$$

The significance of this analytical permutation is that, in $n/m/P/C_{max}$ problems it could be determined by any job feasible sequence value for n! permutations

Hereafter, the application work for the above procedure is illustrated Assume we have a pure flow-shop with four identical machines which process five type of jobs

The (5 X 4) problem matrix for Processing time to complete each operation of each job (set-up and tear-down are included in the processing time) and the machine descriptions are shown in Figures 4 1 and 4 2(a, b, c and d) respectively

Machines		1	2	3	4
JOBS	1	10	12	8	18
	2	9	7	4	12
	3	5	7	4	10
	4	6	8	0	8
	5	13	11	9	16

Figure 4.1: (5 X 4)matrix processing time of a pure flow-shop.

All jobs are assumed to operate with a first-come-first-service (FCFS) rule i e, 12345 job sequence This sequence is one of the 5'=120 possible job sequences

JOB 1

Op No	Description	Machine	Time
1	Turning	1	10
2	Milling	2	12
3	Drilling	3	8
4	Grinding	4	18

Figure 4.2a: Planning sheet showing processing time of job 1.

JOB 2

Op No	Description	Machine	Time
1	Turning	1	9
2	Milling	2	7
3	Drilling	3	4
4	Grinding	4	12

Figure 4.2b: Planning sheet showing processing time of job 2

JOB 3

Op No	Description	Machine	Time
1	Turning	1	5
2	Milling	2	7
3	Drilling	3	4
4	Grinding	4	10

Figure 4.2c: Planning sheet showing processing time for job 3.

JOB 4

Op No	Description	Machine	Time
1	Turning	1	6
2	Milling	2	8
3	Drilling	3	0
4	Grinding	4	8

Figure 4.2d: Planning sheet showing processing times of job 4.

JOB 5

Op No	Description	Machine	Time
1	Turning	1	13
2	Milling	2	11
4	Drilling	3	9
5	Grinding	4	16

Figure 4.2e: Planning sheet showing processing time of job 5.

The expected throughput time (make-span) and completion times for each job at each machine for the above application work are therefore computed using the formula (4 1)
The solution method may be divided into 4 phases according to the number of machines

Phase 1. To determine the completion time of all the 5 jobs to be processed only on the machine m = 1

$f(1,1) = \max\{f(0,1), f(1,0)\}+t(1,1)=t(1,1) = 10$

$f(2,1) = \max\{\underline{f(1,1)}, f(2,0)\}+t(2,1)=f(1,1)+t(2,1)=10+9 = 19$

$f(3,1) = \max\{\underline{f(2,1)}, f(3,0)\}+t(3,1)=f(2,1)+t(3,1)=19+5 = 24$

$f(4,1) = \max\{\underline{f(3,1)}, f(4,0)\}+t(4,1)=f(3,1)+t(3,1)=24+6 = 30$

$f(5,1) = \max\{\underline{f(4,1)}, f(5,0)\}+T(5,1)=f(4,1)+t(5,1)=30+13 = 43$

Phase 2. To determine the completion time of all 5 jobs to be processed only on the machine m = 2

$f(1,2) = \max\{f(0,2), \underline{f(1,1)}\}+t(1,2)=f(1,1)+t(1,2)=10+12 = 22$

$f(2,2) = \max\{\underline{f(1,2)}, f(2,1)\}+t(2,2)=f(1,2)+t(2,2)=22+7 = 27$

$f(3,2) = \max\{\underline{f(2,2)}, f(3,1)\}+t(3,2)=f(2,2)+t(3,1)=27+7 = 34$

$f(4,2) = \max\{\underline{f(3,2)}, f(4,1)\}+t(4,2)=f(3,2)+t(4,2)=34+8 = 42$
$f(5,2) = \max\{f(4,2), \underline{f(5,1)}\}+t(5,2)=f(5,1)+t(5,2)=43+11 = 54$

Phase 3. To determine the completion time of all 5 jobs to be processed only on the machine m = 3

$f(1,3) = \max\{f(0,3), \underline{f(1,2)}\}+t(1,3)=f(1,2)+t(1,3)=22+8 = 30$
$f(2,3) = \max\{\underline{f(1,3)}, f(2,2)\}+t(2,3)=f(1,3)+t(2,3)=30+4 = 34$
$f(2,3) = \max\{\underline{f(2,3)}, \underline{f(3,2)}\}+t(3,3)=f(3,2)+t(3,3)=34+4 = 38$
$f(4,3) = \max\{f(3,3), \underline{f(4,2)}\}+t(4,3)=f(4,2)+t(4,3)=42+0 = 42$
$f(5,3) = \max\{f(4,3), \underline{f(5,2)}\}+t(5,3)=f(5,2)+t(5,3)=54+9 = 63$

Phase 4. To determine the completion time of all 5 jobs to be processed on only the machine m = 4

$f(1,4) = \max\{f(0,4), \underline{f(1,3)}\}+t(1,4)=f(1,3)+t(1,4)=30+18 = 48$
$f(2,4) = \max\{\underline{f(1,4)}, f(2,3)\}+t(2,4)=f(1,4)+t(2,4)=48+12 = 60$
$f(3,4) = \max\{\underline{f(2,4)}, f(3,3)\}+t(3,4)=f(2,4)+t(3,4)=60+10 = 70$
$f(4,4) = \max\{\underline{f(3,4)}, f(4,3)\}+t(4,4)=f(3,4)+t(4,4)=70+8 = 78$
$f(5,4) = \max\{\underline{f(4,4)}, f(5,3)+t(5,4)\}=f(4,4)+t(5,4)=78+16 = 94$

We note that The throughput time (make-span) to complete the entire schedule was derived from the last step of the final phase(4) Hence, the Make-Span is C_{max} = 94 unit times

4.3 The Development of a Computer Programme for Determining The Make-Span of a Schedule:

The procedure in Section 4 2 which mentioned above consumes much time to obtain Make-span, especially for a medium and large flow-shop problem when the optimum or near optimum sequence is the objective On the other hand, It should by now be clear that except in very special production circumstances it is not possible

a to guarantee to produce an optimum schedule or b to sweep through all possible feasible schedules and select the "best"

Furthermore, even if only a single feasible solution is sought, the tedium of producing that solution by hand in a practical situation is considerable To revise that solution as each new batch of jobs arrive is even more tedious, and frequently "manual" methods are either simple "first-come-first-service" (FCFS)systems, or only one machine is loaded in the hope that others will "follow"
The computer, of course, has the ability to devour tedious work, and therefore it would seem that scheduling is ideally suited for computer operation
The aim of this section is two-fold as follows

First, it is to develop a very quick computer solution which will make it possible to solve fairly large sequencing problems using the formula (4 1) which was mentioned in section 4 2
It is intended that the computer solution will be kept quick and simple enough so that no more than 1 second is required for problems on the order, 90 jobs to be processed on 90 machines
The objectives are the Make-Span and individual job completion time for any selected job sequence

The second aim of this computer programme is that, it is the basic step to develop another computer programme which gives an optimum or exact solution for Make-span This programme will be discussed later in section (4 4)
The specific class of job sequencing problems under study in the following logical progression computer programme is that defined by the assumptions stated in section 3 3
A computer program for Make-Span criterion is written in C language using 386 PC and is shown in Appendix B The main feature for the flow-control chart needed to write this programme is shown in Figure 4 3

4.3.1 Experimental Results:

The following four simple steps should be dealt with a computer

a Enter (n X m) processing times matrix to be solved for Make-Span (In dat) is the name given to the file containing matrix

b Run the program according to its path, two question should be replied through a screen

1 Enter number of jobs
2 Enter number of machines

c Press, Enter and the result will be displayed on a screen

d a question will be displayed on a screen?

Do you want another job sequence y/n ? Hereafter, two selected problems will be solved using the 386 based PC

Problem 1:

Recall to the (5 X 4) pure flow-shop in section 4 2 through Figures 4 1 and 4 2(a, b, c and d) The following computer solution will be displayed

```
Number of jobs = 5
Number of machines = 4
 The Completion Time for job 1 = 48
 The Completion Time for job 2 = 60
 The Completion Time for job 3 = 70
 The Completion Time for job 4 = 78
 The Make-Span or Completion Time for last job = 94
Do you want another job sequence ? y/n
```

problem 2:

There is (27 X 90) pure flow-shop problem, Tables 4 1 and 4 2 shown the processing times matrix and a computer solution for the Make-Span respectively as follows

9	2	6	7	6	8	10	0	0	6	0	8	5	2	8	10	10	0	0	0
9	7	7	3	9	1	2	8	0	6	2	1	2	9	2	1	4	3	1	6
7	3	2	10	1	2	9	0	8	1	8	3	7	3	3	10	2	9	9	8
6	6	9	0	9	3	10	4	4	6	2	1	4	5	7	1	8	9	6	3
7	4	3	9	2	3	10	9	9	5	5	0	0	7	2	0	10	6	6	6
10	0	2	5	6	2	7	8	7	4	5	2	4	7	1	7	9	9	3	2
2	2	3	10	10	9	5	4	6	9	4	1	2	7	0	6	4	1	5	10
5	5	6	0	3	6	10	8	2	10	0	3	4	9	10	4	4	10	8	6
9	6	10	3	1	3	9	5	4	9	8	2	8	4	2	6	3	5	2	5

Table 4.1: A (20 X 90) pure flow-shop processing times matrix (complete matrix elements are shown in Appendix C).

```
Number of jobs = 20
Number of machines = 90

 The Completion Time for Job 1 = 457
 The Completion Time for Job 2 = 517
 The Completion Time for Job 3 = 531
 The Completion Time for Job 4 = 534
 The Completion Time for Job 5 = 558
 The Completion Time for Job 6 = 567
 The Completion Time for Job 7 = 577
 The Completion Time for Job 8 = 582
 The Completion Time for Job 9 = 599
 The Completion Time for Job 10 = 610
 The Completion Time for Job 11 = 629
 The Completion Time for Job 12 = 635
 The Completion Time for Job 13 = 655
 The Completion Time for Job 14 = 659
 The Completion Time for Job 15 = 661
 The Completion Time for Job 16 = 686
 The Completion Time for Job 17 = 689
 The Completion Time for Job 18 = 714
 The Completion Time for Job 19 = 732
The Make-Span or completion time for the last Job = 747
```

Table 4.2: A computer printout for the Make-Span (problem 2)

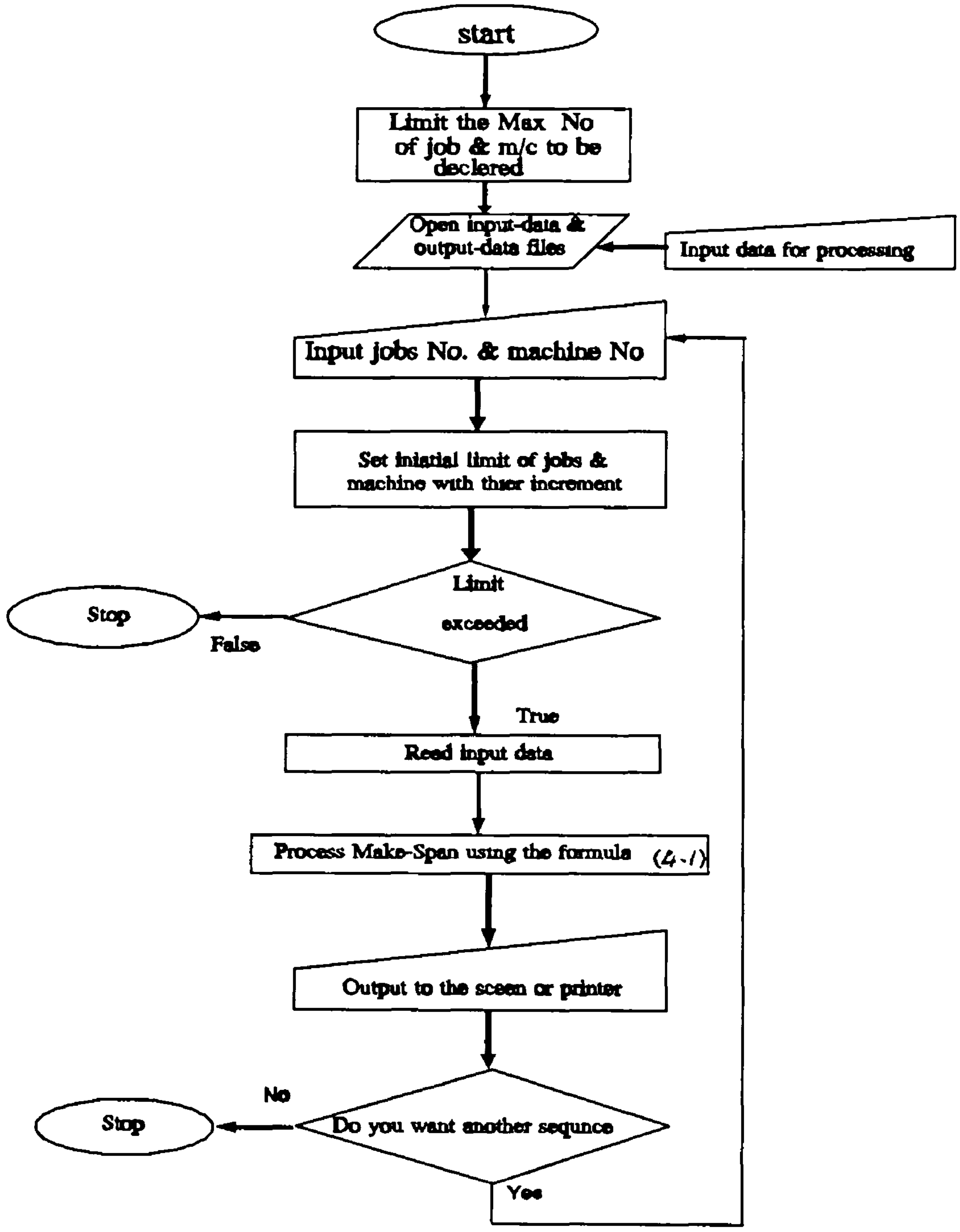


Figure 4.3: The main feature of the Flow control chart to calculate Make-Span.

4.4 The Development of an Explicit Enumeration Computer Programme for Job Sequencing Optimization:

The purpose of this section is to find all possible optimum job sequences and/or optimal Make-Spans for n jobs to be processed through m machines in the same order and no skipping between machines is allowed (a pure flow-shop problem)
This type of shop reduces enumeration of all permutations of jobs from $(n!)^m$ to $n!$
Both Branch-and-Bound [Ignall, et al,1969][21] and dynamic programming [Held, et al,1962][27] approaches (implicit enumeration approaches) deal with job sequencing optimization by the checking of every possible schedule, but unlike explicit or complete enumeration
Here in this section the job sequence optimization will be dealt with by a complete enumeration of all possible schedules
The complete enumeration approach to be studied is economical to use for low and medium work in process and low, medium and high shop utilizations, especially when a low CPU time is available (i e, mainframe network)
The programme is coded in C language as shown in Appendix D
It can reads data for (90 X 90) problem matrix and it could be used for deterministic or stochastic processing times Pseudo-random-numbers are generated using a multiplication congruence method for stochastic processing times in the range of single or double numbers Also the seed function is used in the programme for a new sequence of pseudo-random numbers This approach is based on sweeping through all possible feasible schedules for $n!$ using a link-list method
The main feature for the flow-control chart need to write this programme is shown in Figure 4 4

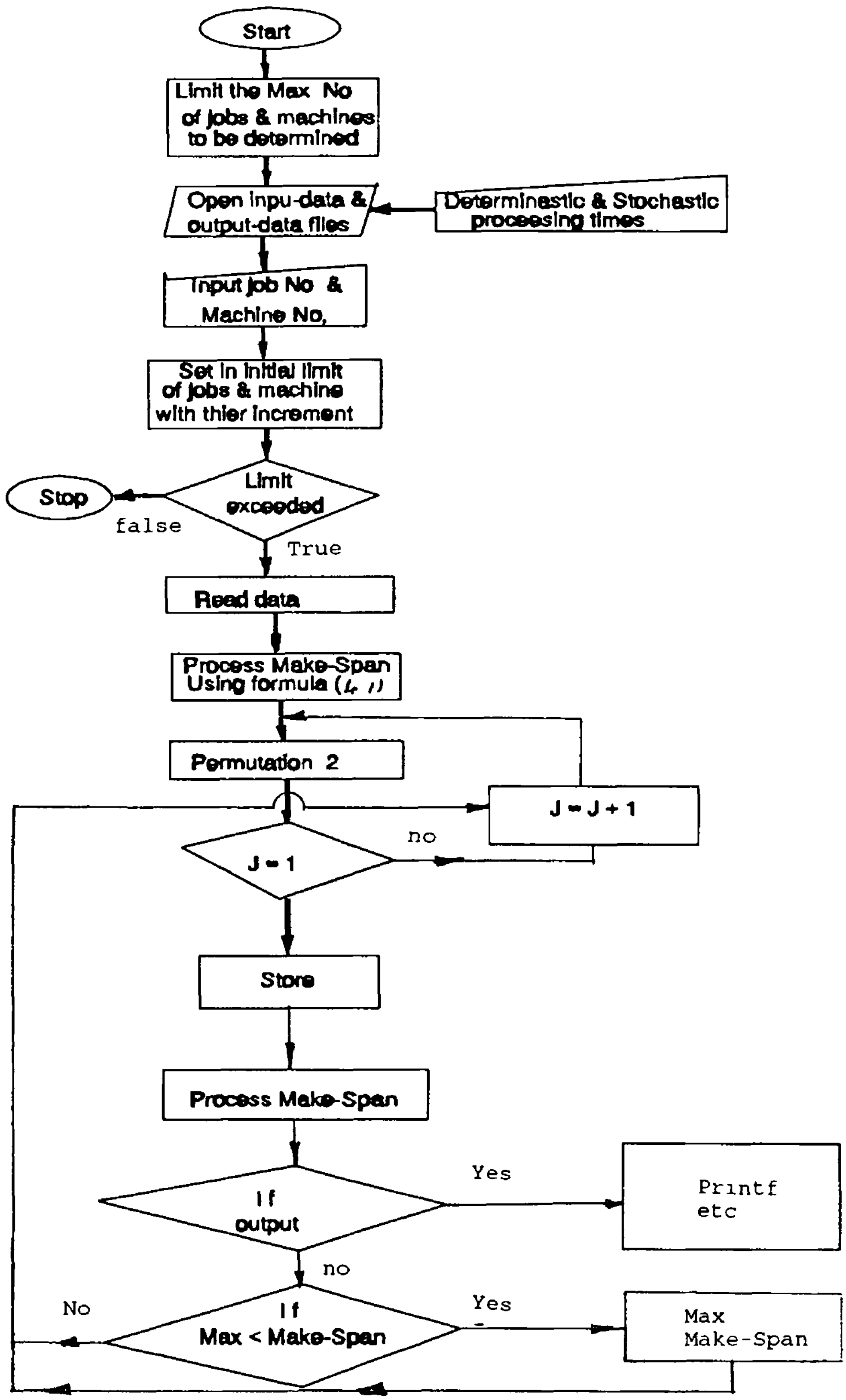


Figure 4.4: The main feature of the flow-control chart for optimal Make-Span in n/m/P/C_{max} problem.

4.4.1 The Characteristics for Optimizing The Make-Span Computer Programme:

This programme has multi-objectives for $n/m/P/C_{max}$ problem It considers the problem of the simultaneous determination of the followings

1 (An arbitrary solution) Job sequences and their Make-Span
2 Optimal job sequence and its Make-Span value
3 (An arbitrary solution) Frequencies for all job sequences
4 CPU time for the solution

4.4.2 Experimental results:

The following four simple steps should be carried out on a computer

a1 In the case of a deterministic input-data, Enter (n X m) processing times problem to be solved
a2 In the case of a stochastic input-data,(see item b 4 below)
b Run the programme according to its path, two questions should be replied through a screen
1 Enter number of jobs
2 Enter number of machine
3 Enter number of runs (this item will be used later in chapter (5)
4 Enter seed number (this Item is used only for stochastic (n X m) processing times matrix)
c Press, Enter and the results will be displayed on a screen (in general, they are shown as follows

Three applications will be presented in this Section
The first will be the $5/4/P/C_{max}$ problem stated in Sections 4 2 and 4 3 1 as shown in Figures 4 1 and 4 2(a, b, c and d)

Problem 3 (repetition of problem 1):

Recall to the deterministic (5 X 4) pure flow-shop problem Section 4 2, in Figure 4 1 and 4 2(a, b, c and d)

The following computer solution will be displayed as shown below in Table 4 3 (pages 64, 65, 66)

---

5 Jobs, 4 Machines

phase 1

```
1 2 3 4 5 Make-Span = 94
1 2 3 5 4 Make-Span = 94
1 2 4 3 5 Make-Span = 94
1 2 4 5 3 Make-Span = 94
1 2 5 4 3 Make-Span = 94
1 2 5 3 4 Make-Span = 94
1 3 2 4 5 Make-Span = 94
1 3 2 5 4 Make-Span = 94
1 3 4 2 5 Make-Span = 94
1 3 4 5 2 Make-Span = 94
1 3 5 4 2 Make-Span = 94
1 3 5 2 4 Make-Span = 94
1 4 3 2 5 Make-Span = 94
1 4 3 5 2 Make-Span = 94
1 4 2 3 5 Make-Span = 94
1 4 2 5 3 Make-Span = 94
1 4 5 2 3 Make-Span = 94
1 4 5 3 2 Make-Span = 94
1 5 3 4 2 Make-Span = 94
1 5 3 2 4 Make-Span = 94
1 5 4 3 2 Make-Span = 94
1 5 4 2 3 Make-Span = 94
1 5 2 4 3 Make-Span = 94
1 5 2 3 4 Make-Span = 94
2 1 3 4 5 Make-Span = 91
2 1 3 5 4 Make-Span = 91
2 1 4 3 5 Make-Span = 91
2 1 4 5 3 Make-Span = 91
2 1 5 4 3 Make-Span = 91
2 1 5 3 4 Make-Span = 91
2 3 1 4 5 Make-Span = 86
2 3 1 5 4 Make-Span = 86
2 3 4 1 5 Make-Span = 85
2 3 4 5 1 Make-Span = 87
2 3 5 4 1 Make-Span = 89
2 3 5 1 4 Make-Span = 89
2 4 3 1 5 Make-Span = 85
2 4 3 5 1 Make-Span = 87
2 4 1 3 5 Make-Span = 89
2 4 1 5 3 Make-Span = 89
2 4 5 1 3 Make-Span = 92
```

2 4 5 3 1 Make-Span = 92
2 5 3 4 1 Make-Span = 94
2 5 3 1 4 Make-Span = 94
2 5 4 3 1 Make-Span = 94
2 5 4 1 3 Make-Span = 94
2 5 1 4 3 Make-Span = 94
2 5 1 3 4 Make-Span = 94
3 2 1 4 5 Make-Span = 86
3 2 1 5 4 Make-Span = 86
3 2 4 1 5 Make-Span = 84
3 2 4 5 1 Make-Span = 87
3 2 5 4 1 Make-Span = 89
3 2 5 1 4 Make-Span = 89
3 1 2 4 5 Make-Span = 89
3 1 2 5 4 Make-Span = 89
3 1 4 2 5 Make-Span = 89
3 1 4 5 2 Make-Span = 89
3 1 5 4 2 Make-Span = 89
3 1 5 2 4 Make-Span = 89
3 4 1 2 5 Make-Span = 87
3 4 1 5 2 Make-Span = 87
3 4 2 1 5 Make-Span = 84
3 4 2 5 1 Make-Span = 87
3 4 5 2 1 Make-Span = 90
3 4 5 1 2 Make-Span = 90
3 5 1 4 2 Make-Span = 92
3 5 1 2 4 Make-Span = 92
3 5 4 1 2 Make-Span = 92
3 5 4 2 1 Make-Span = 92
3 5 2 4 1 Make-Span = 92
3 5 2 1 4 Make-Span = 92
4 2 3 1 5 Make-Span = 84
4 2 3 5 1 Make-Span = 87
4 2 1 3 5 Make-Span = 89
4 2 1 5 3 Make-Span = 89
4 2 5 1 3 Make-Span = 92
4 2 5 3 1 Make-Span = 92
4 3 2 1 5 Make-Span = 84
4 3 2 5 1 Make-Span = 87
4 3 1 2 5 Make-Span = 87
4 3 1 5 2 Make-Span = 87
4 3 5 1 2 Make-Span = 90
4 3 5 2 1 Make-Span = 90
4 1 3 2 5 Make-Span = 92
4 1 3 5 2 Make-Span = 92
4 1 2 3 5 Make-Span = 92
4 1 2 5 3 Make-Span = 92
4 1 5 2 3 Make-Span = 92
4 1 5 3 2 Make-Span = 92
4 5 3 1 2 Make-Span = 95
4 5 3 2 1 Make-Span = 95
4 5 1 3 2 Make-Span = 95
4 5 1 2 3 Make-Span = 95
4 5 2 1 3 Make-Span = 95
4 5 2 3 1 Make-Span = 95

```
5 2 3 4 1 Make-Span = 97
5 2 3 1 4 Make-Span = 97
5 2 4 3 1 Make-Span = 97
5 2 4 1 3 Make-Span = 97
5 2 1 4 3 Make-Span = 97
5 2 1 3 4 Make-Span = 97
5 3 2 4 1 Make-Span = 97
5 3 2 1 4 Make-Span = 97
5 3 4 2 1 Make-Span = 97
5 3 4 1 2 Make-Span = 97
5 3 1 4 2 Make-Span = 97
5 3 1 2 4 Make-Span = 97
5 4 3 2 1 Make-Span = 97
5 4 3 1 2 Make-Span = 97
5 4 2 3 1 Make-Span = 97
5 4 2 1 3 Make-Span = 97
5 4 1 2 3 Make-Span = 97
5 4 1 3 2 Make-Span = 97
5 1 3 4 2 Make-Span = 97
5 1 3 2 4 Make-Span = 97
5 1 4 3 2 Make-Span = 97
5 1 4 2 3 Make-Span = 97
5 1 2 4 3 Make-Span = 97
5 1 2 3 4 Make-Span = 97
   Number of Job Sequences        = 120
```

Phase 2

3 2 4 1 5 = Optimum Job Sequence , Optimal Make-span = 84

CPU time in seconds = 5.879121

Phase 3

Make-Span's Frequencies:

```
For the Make-Span = 84 : The Frequency = 4
For the Make-Span = 85 : The Frequency = 2
For the Make-Span = 86 : The Frequency = 4
For the Make-Span = 87 : The Frequency = 10
For the Make-Span = 89 : The Frequency = 14
For the Make-Span = 90 : The Frequency = 4
For the Make-Span = 91 : The Frequency = 6
For the Make-Span = 92 : The Frequency = 16
For the Make-Span = 94 : The Frequency = 30
For the Make-Span = 95 : The Frequency = 6
For the Make-Span = 97 : The Frequency = 24
```

Table 4.3: A computer print-out for the optimal Make-Span (problem 3, repetition of problem 1).

Table 4 3 has been dıvıded ınto three phases as follows

The fırst phase ındıcates to all possıble ȷob sequences for problem 3 It shows that the optımum solutıon has four ȷob sequences (ındıcated by underlıne), each of them has the same value of the optımal Make-Span whıch ıt ıs equal to 84 unıts tıme

The second phase gıves the selected optımal ȷob sequence and ıts Make-Span value and ıt has the followıng ȷob sequence

(3 2 4 1 5) wıth optımal Make-Span = 84 We note that the CPU tıme to solve thıs problem ıs very low and ıt ıs equal to 5 88 unıts tıme
The thırd phase ıs also ımportant to take ınto account for dıstrıbutıons, ıt gıves the frequencıes for all ındıvıdual Make-Spans In general, these frequencıes ın turn could be used to plot a dıstrıbutıon of make-Spans for a (n X m) problem matrıx Heller[1960][118], concludes that "the numerıcal experıments show that the dıstrıbutıon of schedule-tımes (Make-Spans) for large number of samples ıs normal" He reported that thıs normalıty can be used to determıne decısıon-theoretıcal rules to termınate samplıng when the cost of contınued samplıng exceeds the expected gaın from further samplıng Thıs conclusıon wıll be clarıfıed through the problem 4 ın the next paragraph

The second applıcatıon (problem 4) wıll deal wıth a more complıcated problem than the prevıous one (problem 3) It has 10 types of ȷobs to be processed on 20 machınes ((10 X 20) processıng tımes matrıx) as stated ın Table 4 4 All the processıng tımes were selected randomly rangıng from 0-10 usıng a specıal programme whıch was wrıtten ın C language usıng 386 based PC wıth 16MHz clock speed
Fıgure 4 5 shows the dıstrıbutıon of all possıble Make-Spans for problem 4 We note that thıs dıstrıbutıon ıs faırly normal[118]

7	3	1	9	2	6	4	2	9	0
0	5	8	8	8	9	7	0	4	3
0	4	7	5	3	8	5	9	9	8
3	9	5	1	10	2	4	7	2	7
0	0	5	8	10	5	2	8	6	6
8	8	2	3	8	6	0	8	9	0
9	10	3	2	3	5	0	2	8	4
6	6	8	8	0	5	6	9	6	3
5	1	1	5	0	10	3	7	7	4
4	4	3	5	10	9	7	9	8	2
5	8	3	5	10	0	0	8	5	4
6	5	2	9	0	6	8	6	5	1
5	6	1	5	1	0	10	0	7	6
8	7	6	9	9	7	4	3	9	8
7	10	9	9	4	2	1	9	6	9
1	10	0	4	9	2	4	2	7	3
5	9	6	3	9	6	4	8	10	6
0	6	4	3	3	3	8	9	10	7
8	8	6	3	1	10	8	7	7	2
8	8	1	6	9	5	2	7	3	10

Table 4.4: A (10 X 20) processing time matrix (problem 4).

The following is the proposed procedure solution

Problem 4:

10 Jobs 20 Machines

Number of Job Sequences = 3628800
2 10 7 1 9 4 8 5 6 3 = Job Optimum Sequence
Optimal Make-Span = 183
CPU time = 2 5 = Hrs

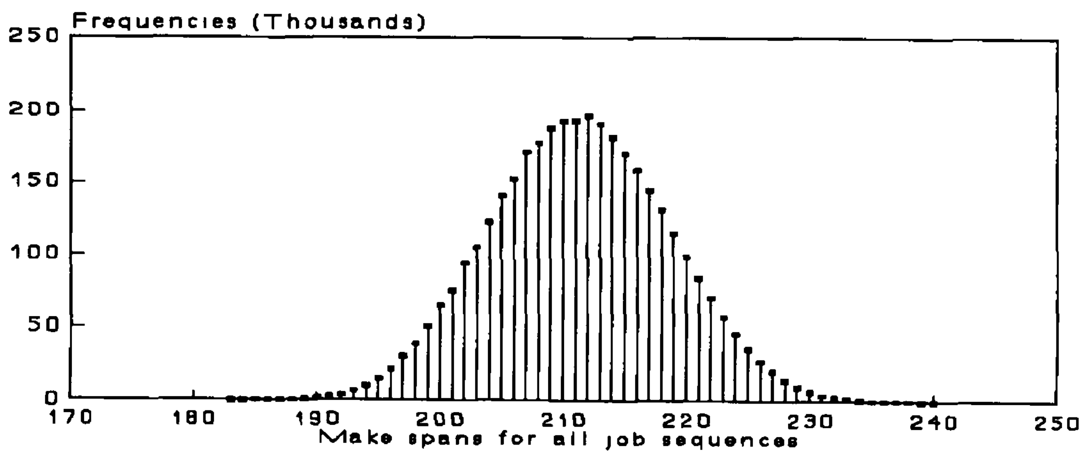


Figure 4.5: Distribution of Make-Spans for (10 X 20) processing time matrix (problem 4)

4.4.3 Verification of Efficiency of the Optimal Make-Span Solution:

The computer procedure under study is a fairly dominant method than many algorithms that have been studied [Palmer][35], [Campble, Dudek and Smith][9], [Dannengring][37], [Gupta][36] and Al-Qattan[119], especially for n ≤ 10 and m ≤ 90 (see Section 4 4 1 problem 4) Due to[119] Problem 5 below (Figure 4 6) has been solved using the above most known heuristic approaches Table 4 5 reports a comparison of different heuristic approaches with the proposed procedure

Problem 5:

	MACHINES	**1**	**2**	**3**
JOBS	**1**	6	8	2
	2	4	1	1
	3	3	9	5
	4	9	5	8
	5	5	6	6

Figure 4.6: (5 X 3) processing time matrix (problem 5).

Method	Job Sequence	Make Span	method Limitation
Palmer	3,5,4,2,1	37	30% optimum
Campbell, et al	3,5,4,1,2	35	Economical n ≤ 8 & is not guarantee
Dannenbring	3,5,4,1,2	35	n ≤ 6 , m ≤ 10 & 35% optimum
Gupta	5,3,4,1,2	36	An optimum is not guarantee
Al-Qattan	3,4,5,1,2	34	CMSs only & optimum not guarantee
The proposed method	3,4,5,1,2	34	An optimum is guarantee

Table 4.5: Comparison of five heuristic approaches with the proposed computer method.

4.4.4 Computation Time:

The computation time for the proposed computer method varies according to (n X m) problem size The measure of CPU times were used in conjunction with five sets of problems with n ranging from 5 to 9 were worked Each set contains 10 groups of machines (5,10,20,30,40,50,60,70,80 and 90) The problems were constructed with random processing times ranging from 0-10, these were generated at random according to the sub-programme which is written within the proposed programme Table 4 6 and Figure 4 7 contains the computation time data and their increased curves on the (5 sets of jobs X 10 groups of machines) = 50 problems

We note from Figure 4 7 that the length of the CPU time which seek an optimal Make-Span (according to the proposed method and using 386 based PC_16MHz) for the $n/m/P/C_{max}$ problem has the following features

1 It is sensitive to an increase in the number of jobs for a given number of machines
2 It is not so sensitive to an increase in the number of machines for a given number of jobs
3 For long and medium term scheduling, where $n \leq 9$ and $m \leq 90$ the CPU time is reasonable in the case of low and medium WIP and low, medium and high shop loads
4 For short term scheduling, where $n \leq 7$ and $m \leq 60$ the CPU time is reasonable in the case of low W-I-P and low, medium and high shop loads

4.4.5 Economic Consideration for The Proposed Method:

In view of the economic considerations, Each manufacturing company which would consider using the proposed method would, of course, do so under a different set of circumstances There would be many variables to consider in attempting to determine a break-even point between optimum or near-optimum solution techniques This, of course, presupposes that the company has a choice Nowadays, the selection of an optimum solution is dependant on the CPU time availability

number of Jobs		**5**	**6**	**7**	**8**	**9**
	5	0 38	1 04	6 43	55	611
CPU time	**10**	0 6	4 62	12 1	108	1190
in seconds	**20**	0 71	3 19	23 5	212	2348
of	**30**	0 93	4 62	34 8	316	3507
different	**40**	1 26	6 1	46 3	420	4664
m	**50**	1 43	7 42	57 6	524	5820
Machines	**60**	1 65	8 96	68 9	628	6977
	70	1 92	10 2	80 3	732	7249
	80	2 09	11 7	91 7	836	8280
	90	2 31	13 1	103	940	9294

Table 4.6: Computer computation times for different (n X m) problem sizes.

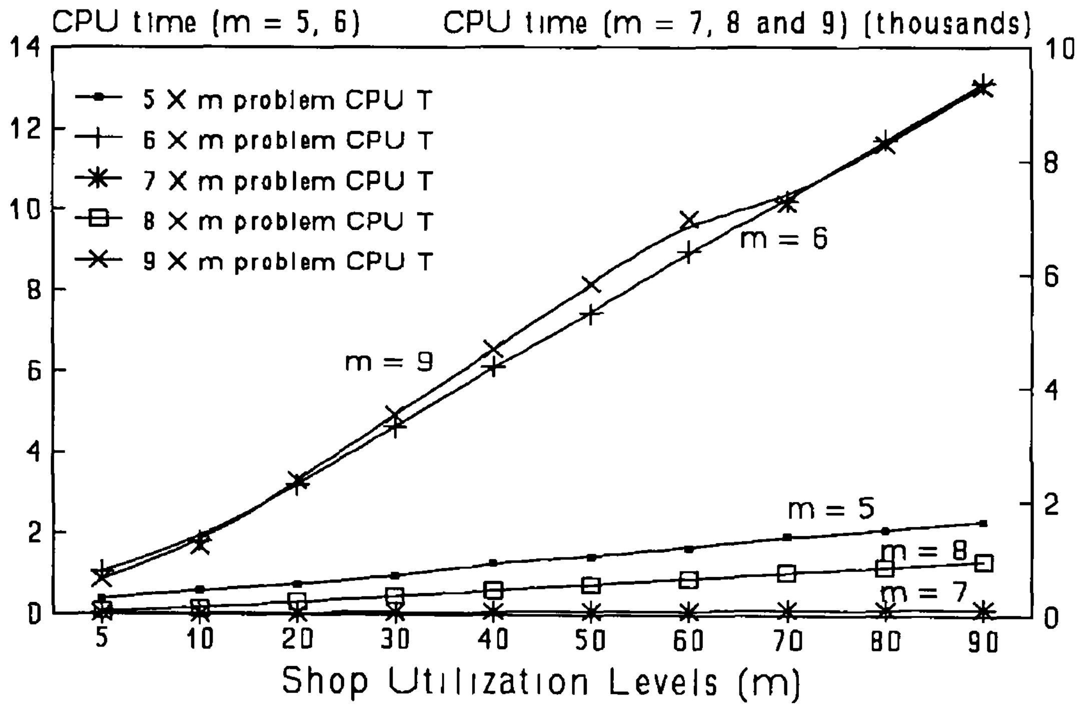


Figure 4.7: Computation times for different problem sizes.

CHAPTER FIVE

5. A Computer Simulation Analysis for the Flow-Shop Scheduling Priority Rules

5.1 Sequential Scheduling Rules In Production Scheduling:

It would be extremely difficult to formulate and simultaneously solve the entire scheduling problem in terms of the complete enumeration of a large size scheduling problem. The explicit enumeration of a large size scheduling problem is really not possible. For example, a (15x10) pure flow-shop problem would have (15! = 1.3076E12) possible job sequencing. Whereas, (5x6) flow-shop problem size would have ($(5!)^6$ = 2.986E12). In this flow-shop problem case even if only 1% of permutations is feasible, they would constitute 2.9561E12 permutations which is still prohibitive for enumeration. On the other hand, sequential priority rules for production scheduling are heuristic methods of job sequencing indicate how to assign a specific job to a specific machine at a given time, when a machine becomes available for process. In other words priority rules are designed and selected to maximize the expectation of given variables, thus avoiding enumeration.

5.2 Classification of Scheduling Rules:

Usually, when dealing with the sequencing problem, terms such as scheduling rules, priority rules, or heuristic are often having the same meaning. Gere[1966][120] has attempted to distinguish between the above three definitions. He considers priority rule or priority function is that rule which assigned to each relevant job a scalar value, the minimum (or maximum) of which determines the job to be selected over all others for scheduling In the case of a tie, the job with smaller job number is selected Also he defines a heuristic to be simply some "rule of thumb," whereas a scheduling rule dictates what job is to be scheduled in preference to all others, in the given circumstances

A scheduling rule may include one or more heuristic and/or one or more priority rule

Hereafter, the scheduling rules may be classified into three categories The following sections address the selected common known priority rules each of these in turn[41] (The first ten rules which are the most studied in the literature [5], [40], [42], [43], [52] and [44] will be used through the thesis as required)

5.2.1 Simple Priority rules:

This section is further divided into subcategories based on information related to

I. Rules involving processing time:

1 SPT Select the job with the "Shortest Processing Time" (also called SIO, SI (Shortest Imminent operation time)

$$\mathbf{P1 \le P2 \le P3 \le \ldots \le P_n}$$

2 LPT Select the job with the "Largest Processing time"

$$\mathbf{P1 \ge P2 \ge P3 \ge \ldots \ge P_n}$$

3 SRPT Select the job with the "Smallest Remaining or Content Processing Time"

$$\sum_{i=1}^{m} P_1 \le \sum_{i=1}^{m} P_2 \le \sum_{i=1}^{m} P_3 \le \quad \le \sum_{i=1}^{m} P_n$$

4 LRPT· Select the job with the "Longest Remaining or

content Processing Time".

$$\sum_{i=1}^{m} P_1 \geq \sum_{i=1}^{m} P_2 \geq \sum_{i=1}^{m} P_3 \geq \ldots \geq \sum_{i=1}^{m} P_n$$

II. Rules involving The selected Arrival Time & Random:

5. FCFS: "First Come First Service". Select the first job to arrive at (W.S.) a machine queue is to be sequenced first (also called FIFO -First In First Out-)
6. LCFS: "Last Come First Service". Select the last job to arrive at (W.S.) a machine queue is to be sequenced first (also called LIFO -Last In First Out-).
7. FASFS: "First Arrived at Shop, First Service". The first job arriving in shop serves first.
8. Random: " Random selection is randomly assigned". Select in random sequence.

III. Rules involving due dates:

9. EDD: Select the job with the Earliest Due Date.
10. STSlack: Select the job with minimum Slack time. Is equal to due date d_j minus the time of arrival at the machine.

IV. Rules related to machines:

11. NINQ: Select the job that will go on to its next operation where the machine has the shortest queue.

5.2.2 Combination of Simple Priority Rules:

12. FCFS/SPT: From jobs waiting for more than a specific time, select according to FCFS; if all waiting jobs are in the queue for a smaller duration, select

according to SPT

13 Cost Value Divide jobs in two classes, high and low cost value Select the job from the high cost value with FCFS, then from the class

14 SMOVE Select the job that will go on to the next operation where the machine has the shortest "critical" queue If there are no critical queue, use FCFS

15 SOR Use the SPT rule, but give preference to those jobs that will go to "critical" queues (queue with only small amount of work waiting)

5.2.3 Heuristic Scheduling Rules:

16 Alternate Operation If selection of a job according to some simple rule makes another job "critical" (such as positive lateness), see the effect of the job already selected Repeat, if some other job(s) is affected

17 LAH (Look Ahead) [120] Study the effect of scheduling a job (determined by a simple rule) on another job that may arrive in the queue before the schedule job is completed

18 SHOPNH Select the job with SPT rule, but hold if a few jobs in the queue and another job with smaller processing time is expected soon (Keep machine idle until this job arrives)

The following two sub-chapters may be conveniently classify the priority rules by their environment and by scope of information required in order to implement them

5.3 Priority Rule Environments:

* Static priority rules
* Dynamic priority rules

In general, the static priority rules can be applied at the beginning of the scheduling period and result in a fixed schedule for the period In other words, static rules are rules in which the value of the priority rules do not change with

1 the passage of time
2 The relative progress of a job in relation to other jobs
3 Disturbances in the shop (i e machine breakdown)

Whereas the dynamic ones changing over time
The static priority rules can be broadly classified into three types as follows

1 Processing time based rules
2 Due date based rules
3 Selected Arrival time & random based rules

The dynamic priority rules can be adopted by the combination and heuristic priority rules (as mentioned before in this chapter)

5.4 Priority Rules Information Required:

* Local priority rules
* Global priority rules

Local priority rules require information only about those jobs that are waiting at a machine, The most simple priority rules (they mentioned in the Section 5 2 1) may be classified under local priority rules (Moor, et al,1961)[75]
Global priority rules require more information about jobs or machine states at other resources or waiting lines (such as look ahead rule[120]), hence they have higher cost information processing systems than the local priority rules However, global priority rules would be justifiable only if it was proven more effective than local rules

5.5 A Computer Simulation based Priority Rules:

Many computer simulations have been implemented in order to use priority rules for job sequencing. During the past three decades (since Conway,1964[49]) few manual systems use priority rules. However, the experimental evidence shows that only under exceptional circumstances is a computerized rule likely to be worth implementing in preference to a manual one.
In addition, the primary objectives of the simulation work through this research have been to compare specific operating procedures, to test broad conjectures about priority rules.

5.5.1 Choice of a Priority Rule:

Experimentation with a computer simulation model has made it possible to compare effectiveness between various priority rules, therefore the choice between priority rules must rely on reported computer simulation results. A set of simulation analyzed measure of performances for a number of well-known rules is given through the Sections 5.6 and 6.1 . The details of the use of the rules are given in section 5.2.1.

5.6 Development of a Computer Simulation Programme for Measuring the Effectiveness of Several priority rules:

The study described in this section is to obtain an efficiently solvable new method of the pure flow-shop scheduling problem through measuring the effectiveness of several priority rules.
For this purpose an approach based on computer simulation of deterministic and stochastic pure flow-shop scheduling versus the six priority rules has been developed.
Many researchers have been successful in developing efficient solution algorithms for $n/m/P/C_{max}$ problem with two and three machines by Johson,1954[1], his aim was the optimal Make-Span. For general problem size due to Ignall et al[1965][21],

Campbell, et al[1970][9], Gupta[1971][36], Dannenbring [1977][37], their algorithms dealt with the approximate solution and the exact solution is not guaranteed Bera[1983] [12] has obtained the optimum Make-Span, Waiting Time and Idle Time only for n ≤ 6 and m ≤ 6 However, this section will deal the approximate solution for job sequencing using the priority rules The programme under study was coded in C language as stated in Appendix E It can read data for (90 X 90) problem size and it could be used for deterministic or stochastic (n X m) processing times matrix

Pseudo-random-numbers are generated using Lehmer multiplication congruence method for stochastic processing times in the range of single or double numbers (i e, 0-99) This range was selected to limit the elements of (n X m) processing times matrix for each replication or run (Number of runs will be mainly used through discussing the effectiveness in Section 5 6 2)

The output provides the basis for evaluating the measure of effectiveness for the following six priority rules

1 FCFS rule, First Come First Service
2 SPT rule, Shortest Processing Time
3 LPT rule, Longest Processing Time
4 SRPT rule, Smallest Remaining Processing Time
5 LRPT rule, Largest Remaining Processing Time
6 RANDOM rule, Select an operation at Random

   In our study, due to random number generated for processing times, the jobs sequence for random rule is equal to FCFS rule

Gere, Jr[1966][120] has defined the effectiveness of a priority rules as follows

The effectiveness of a priority rule is measured by the expected value of criterion function which results when the rule is followed

In this section the basic interest in our computer simulation is to determine the effectiveness of the different priority rules with respect to minimize the following well-used measure of performances under many times of runs

1 Average Make-Span (maximum completion time)
2 Average Mean Completion Times for jobs (AMCTs)
3 Average Total Waiting Times for jobs (ATWTs)
4 Average Total Idle Times for machines (ATITs)

5.6.1 Statement of the Procedure Method:

To compare the six priority rules under identical conditions with respect to the four criteria which have been mentioned above The same random seed of pseudo-random-generated per run or replication was used for all the priority rules
The following four steps should be carried out on a computer

a1 In the case of deterministic input-data, Enter (n X m) processing times problem to be solved
a2 In the case of stochastic input-data, (see item b 4 below)
b Run the programme according to its path, two question should be answered through a screen

1 Enter number of jobs
2 Enter number of machines
3 Enter number of replications or runs to be tested for obtaining the criterion's average value
4 Enter seed number of runs (This item is used only for stochastic (n x m) processing times problem)

c Press, the Enter Bottom and the results will be displayed on a screen
Problem 9 (page 80) illustrates the computer procedure of the comparison between the different six rules with respect to the following criteria

- Make-Span
- Total Mean Completion Time

- Total Waiting Time
- Total Idle Time

Problem 9:

This problem has 16 types of jobs to be processed on 33 machines in the same order The keyboard writing should be displayed on the computer screen as follows

Number of jobs = 16
Number of machines = 33
number of runs = 1
Seed number = 33

Then, the randomly selected (16 X 33) processing times matrix and output data will be displayed as shown in Tables 5 1 and 5 2 respectively

20	19	27	20	10	8	6	12	16	5	24	28	21	20	23	7
27	6	16	5	28	25	1	19	27	24	12	7	8	24	30	29
4	9	5	6	8	25	9	29	18	6	6	4	30	8	30	7
19	21	5	6	28	6	7	14	28	15	9	14	21	13	8	5
24	3	18	0	11	16	10	8	6	15	1	13	8	30	29	1
22	23	6	0	11	12	5	1	14	3	25	8	18	4	26	30
11	20	8	20	14	22	10	6	12	22	27	8	14	14	0	28
2	27	12	23	2	8	8	17	4	17	12	9	11	21	30	13
29	5	13	27	8	26	2	12	13	3	9	9	13	16	8	24
2	1	14	24	23	22	17	4	1	10	28	13	9	25	16	13
1	13	16	22	16	5	29	10	1	12	5	17	30	30	10	12
12	23	15	27	30	21	26	16	16	16	7	23	22	10	0	19
20	21	2	17	22	9	0	7	14	26	15	10	25	18	18	10
4	3	18	29	10	3	14	12	16	18	12	21	19	3	18	13
0	24	27	0	12	18	0	0	5	11	11	6	13	8	0	27
28	19	2	9	8	30	14	5	8	19	19	30	4	29	3	1
29	1	23	29	29	19	8	16	5	6	3	0	10	26	9	4
19	13	13	13	1	10	16	28	14	11	9	18	3	26	2	14
17	11	23	17	18	27	26	10	26	10	6	25	28	10	24	30
18	5	6	6	29	6	2	19	4	25	30	25	13	21	18	15
13	4	2	10	17	30	12	22	30	4	5	4	16	18	12	3
18	1	7	21	3	21	1	2	24	23	9	6	27	13	16	20
14	9	2	0	0	13	8	28	22	17	27	18	9	19	23	30
26	21	13	10	18	28	5	22	10	9	5	16	4	2	7	8
3	10	8	18	27	30	14	21	21	1	6	5	13	14	5	22
26	12	26	23	12	30	22	27	0	9	16	30	21	20	27	15
1	7	9	11	3	5	20	25	25	20	6	3	25	4	18	12
19	27	23	28	10	3	20	20	18	30	17	27	1	8	7	6
16	15	1	0	16	25	4	22	20	27	4	0	15	23	18	26
15	17	17	0	6	14	22	26	3	4	17	30	17	29	23	1
2	7	11	8	27	17	26	22	26	26	12	17	8	11	13	17
9	3	25	18	22	10	2	5	5	3	1	4	17	26	15	17
22	1	15	13	23	5	15	24	22	4	16	0	26	4	1	29

Table 1.5: A (16 X 33) processing times matrix (problem 9).

16 Jobs, 33 Machines, Random seed = 33, (Number of runs) = 1

Average MAKE-SPAN

Rule 1	FCFS	= 1029
Rule 2	SPT(SI)	= 909
Rule 3	LPT(LI)	= 1042
Rule 4	SRPT	= 993
Rule 5	LRPT	= 1014

Average MEAN COMPLETION TIME

Rule 1	FCFS	= 792
Rule 2	SPT(SI)	= 733
Rule 3	LPT(LI)	= 784
Rule 4	SRPT	= 782
Rule 5	LRPT	= 751

Average Total Waiting Time

Rule 1	FCFS	= 3120
Rule 2	SPT	= 2708
Rule 3	LPT	= 2457
Rule 4	SRPT	= 2856
Rule 5	LRPT	= 2564

Average Total Idle Time

Rule 1	FCFS	= 6087
Rule 2	SPT	= 6274
Rule 3	LPT	= 8262
Rule 4	SRPT	= 4896
Rule 5	LRPT	= 7351

Computation time in seconds = 2 912088

Table 5.2· A computer print-out for problem 9.

Table 5 2 above shows the comparison between the six priority rules under study with respect to three criteria (Make-Span, Mean Completion Time for jobs and Idle Time of machines) We clearly note from this table that the rule which minimise each of the three criteria for problem 9 is as follows

- SPT rule minimize the Make-Span criterion
- SPT rule minimize the Mean Completion Time criterion
- LPT rule minimize the Waiting Time criterion
- SRPT rule minimize the Idle Time criterion

5.6.2 Effectiveness Evaluation of The Priority Rules:

In order to reduce the effect of bias from one run testing to another and in another hand, because a relatively large number of runs are necessary to get accurate information about the behaviour of the scheduling process, the effectiveness of the priority rules were evaluated via the average of a large number of runs of randomly simulated problems, (number of runs, say 500 will probably be found in reasonable number of runs by Thompson[1960][121]).
The effectiveness evaluation is tested on a complete different factorial experiment for the six rules, four criteria, 72 different problem sizes; i.e., 10-shop sizes (5, 10, 15, 20, 25, and 30 machines), and 12 levels of Work-In-Process (number of jobs in shop) equal to (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60) and 500 runs with different streams of the seed number (in the range of 0-10 for processing times matrix) were executed. All the 72 problems are considered to be a non-due date problems and job ready times equal to zero (a static problem).
The computational results for the 72 problems stated above are given through Tables 5.3(a, b, c and d), and their discussions are clarified through the following four sections:

5.6.3 Evaluating the Priority Rules Vs. the Make Span under different WIP and Shop sizes:

The observed Make-Span data are stated in Table 5.3a. This data report the measure of effectiveness between the rules with respect to the average Make-Span under 500 runs at different shop sizes.
A note of interest from Figures 5.1(a, b, c, d, e and f) is that the rule which tends to minimize the average make-span for low, medium and high WIP and all different shop sizes (in general, $n/m/P/C_{max}$ problem size) is the SPT rule (Shortest

Processing Time rule), While the rules SRPT, FCFS and LRPT are quite equal to each other The LPT rule (Longest processing Time) gives the worst performance over other

Table 5.3a: Observed AMSs of 500 runs Vs. the proposed rules.

Problem Size	Average Make-Spans				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 5	58	54	62	58	58
10 X 5	92	85	100	91	92
15 X 5	124	115	136	123	123
20 X 5	154	144	172	153	153
25 X 5	184	173	207	184	183
30 X 5	214	202	241	212	213
35 X 5	243	232	276	242	242
40 X 5	270	258	308	269	270
45 X 5	301	287	345	299	301
50 X 5	330	316	379	329	328
55 X 5	358	342	413	356	356
60 X 5	385	370	446	385	384
5 X 10	91	87	96	91	91
10 X 10	131	124	138	130	130
15 X 10	166	159	178	166	167
20 X 10	200	192	216	200	200
25 X 10	234	225	254	233	233
30 X 10	266	258	291	266	267
35 X 10	299	288	327	298	297
40 X 10	329	319	362	328	327
45 X 10	360	349	399	359	358
50 X 10	391	379	436	391	391
55 X 10	421	410	472	421	420
60 X 10	452	440	506	451	450
5 X 15	124	120	127	124	123
10 X 15	167	161	174	167	167
15 X 15	207	199	217	206	206
20 X 15	242	235	255	241	241
25 X 15	278	269	294	277	276
30 X 15	311	302	332	310	310
35 X 15	345	336	372	344	344
40 X 15	378	369	408	376	378
45 X 15	412	402	446	410	410
50 X 15	444	434	482	442	442
55 X 15	474	465	521	474	474
60 X 15	506	495	555	504	505

Continuing of Table 5.3a:

Problem Size	Average Make-Spans				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 20	155	151	158	154	155
10 X 20	201	195	207	200	201
15 X 20	243	236	252	242	242
20 X 25	282	273	294	280	281
25 X 20	318	311	334	317	318
30 X 20	355	346	373	353	353
35 X 20	389	380	412	388	389
40 X 20	424	415	451	423	423
45 X 20	456	447	488	456	456
50 X 20	490	482	525	489	489
55 X 20	523	515	564	523	523
60 X 20	557	546	602	556	555
5 X 25	185	180	188	184	184
10 X 25	234	229	241	235	234
15 X 25	278	271	287	277	278
20 X 25	318	311	329	317	317
25 X 25	358	349	371	356	355
30 X 25	394	387	412	392	395
35 X 25	430	421	452	430	430
40 X 25	465	457	491	464	464
45 X 25	501	493	530	498	499
50 X 25	536	526	568	534	533
55 X 25	569	561	606	568	569
60 X 25	603	594	644	602	601
5 X 30	213	209	218	213	214
10 X 30	265	260	271	266	265
15 X 30	311	305	321	311	312
20 X 30	354	347	366	354	354
25 X 30	394	388	408	393	394
30 X 30	434	425	449	432	432
35 X 30	470	461	489	469	468
40 X 30	507	498	529	505	505
45 X 30	542	534	570	542	542
50 X 30	577	569	608	576	577
55 X 30	613	605	649	612	612
60 X 30	646	638	685	646	645

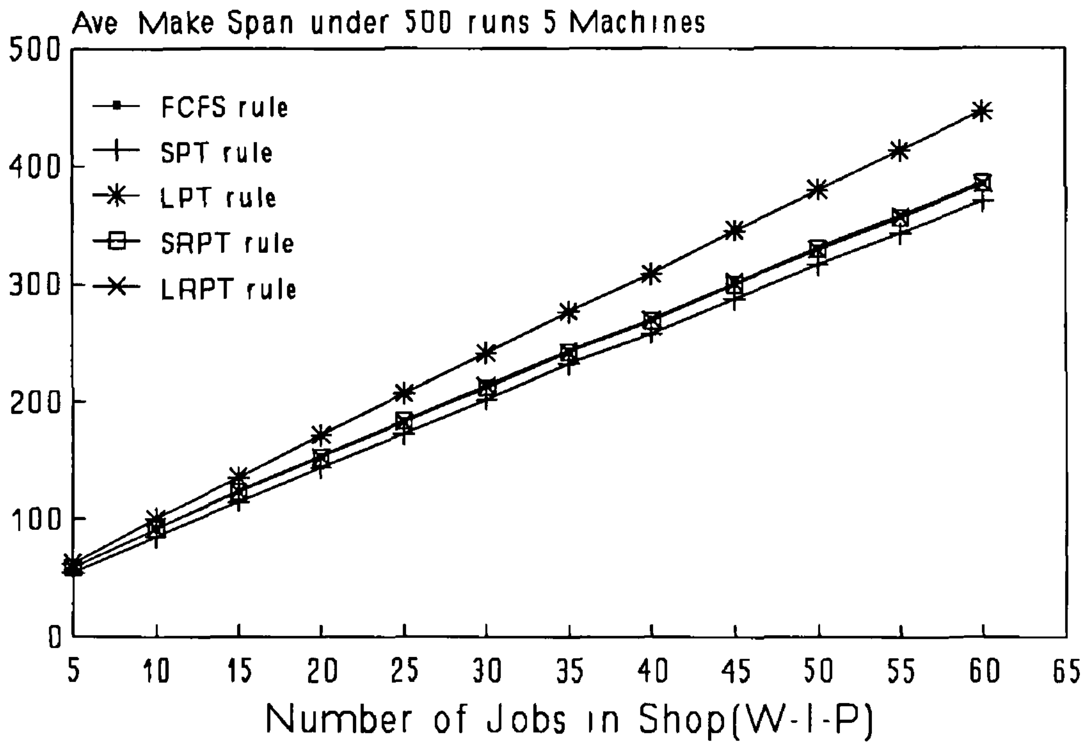


Fig. 5.1a: Effect of the rules Vs. AMS under 5 shop machines.

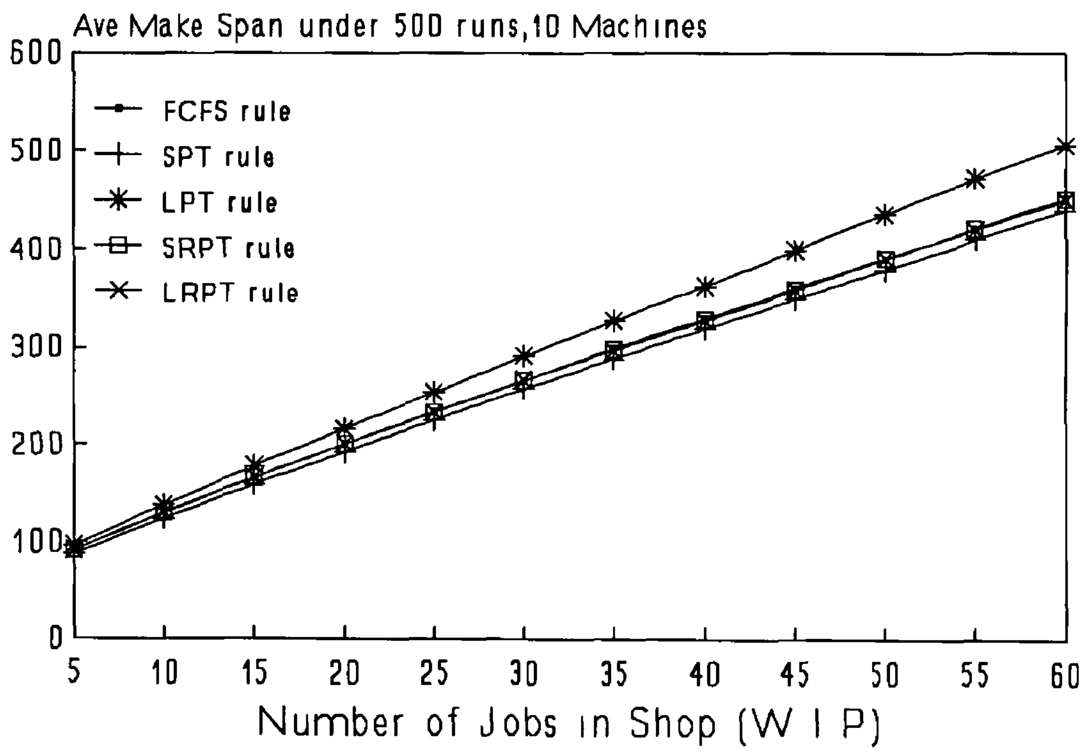


Fig 5.1b: Effect of the rules Vs. AMS under 10 shop machines.

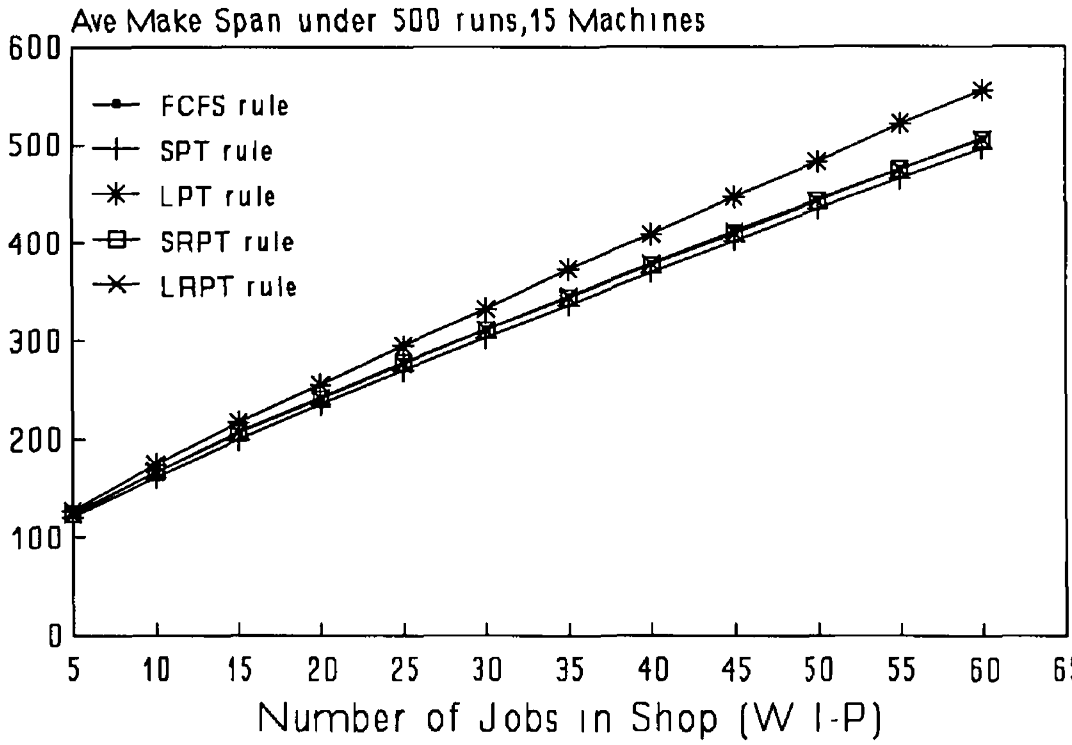


Fig. 5.1c: Effect of the rules Vs. AMS under 15 shop machines

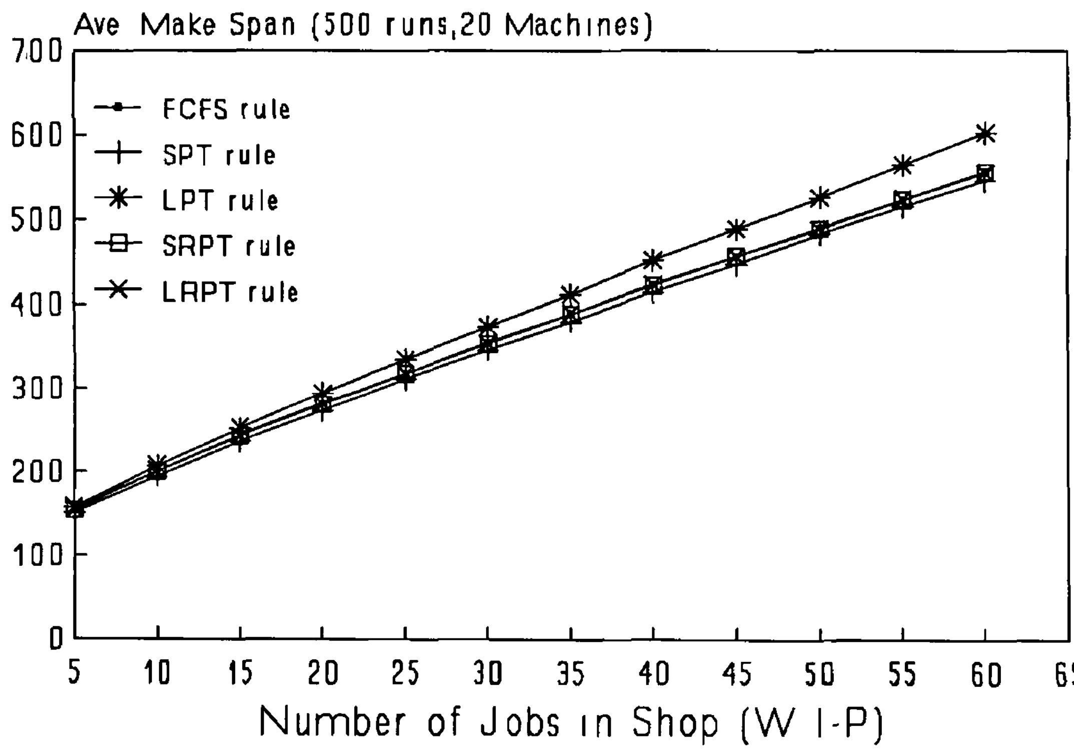


Fig. 5.1d· Effect of the rules Vs. AMS under 20 shop machines.

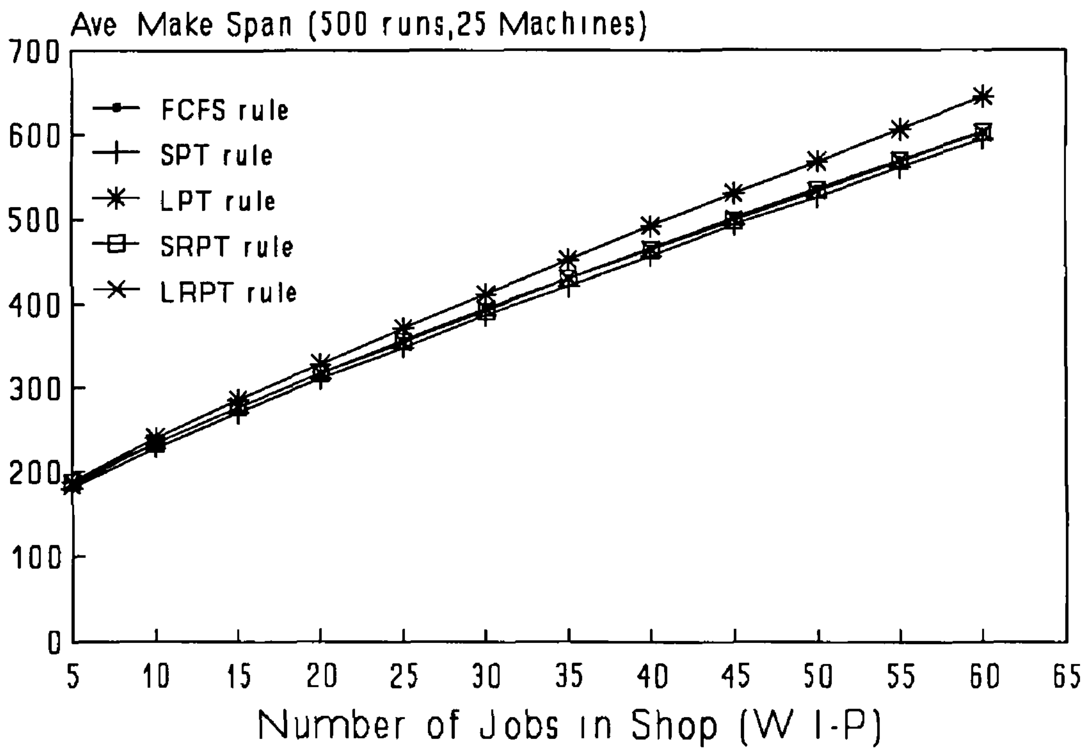


Fig. 5.1e: Effect of the rules Vs. AMS under 25 shop machines.

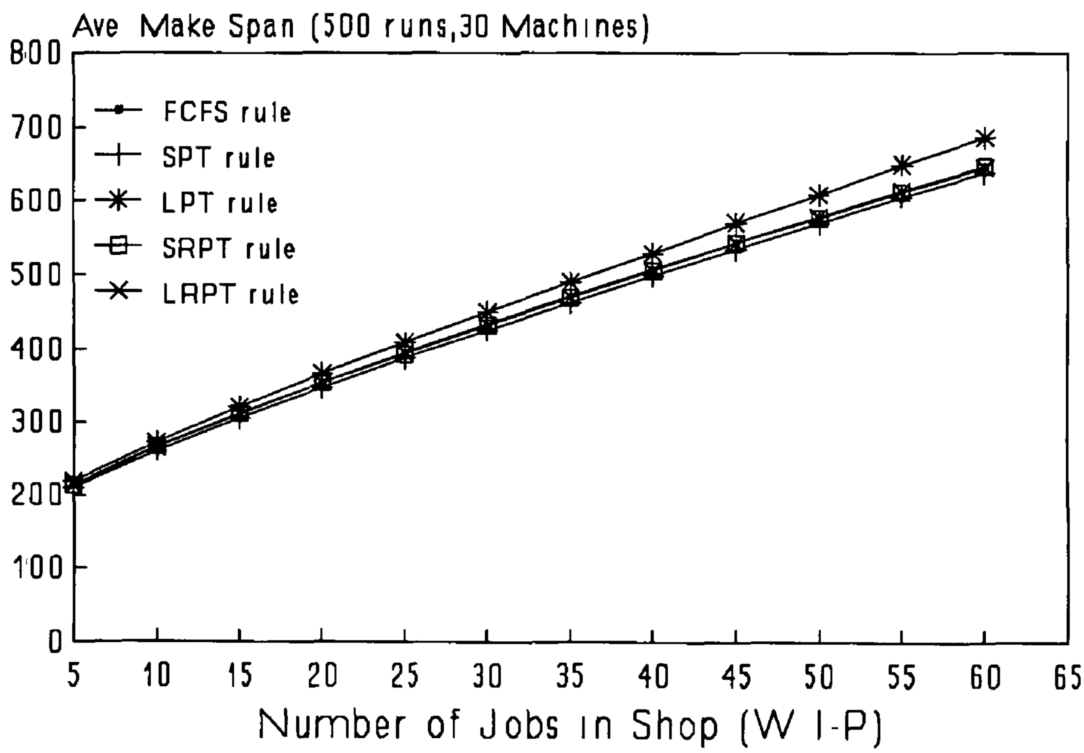


Fig. 5.1f: Effect of the rules Vs. AMS under 30 shop machines.

5.6.4 Evaluating the Priority Rule Vs. the Mean Completion Time Criterion under Different WIP and Shop Sizes:

The data in Table 5 3b and their plots in Figs 5 2(a, b, c, d, e and f) show the performance against the level of the work-in-process (WIP) under different shop sizes The following analyses give a comparison between the proposed rules with respect to (AMCTs) Average Mean Completion Times

1 The SRPT rule which schedules jobs with the smallest remaining or content processing time first, has the smallest value for average mean completion time under all levels of W-I-P and shop sizes

2 For SPT and FCFS based rules respectively, they give a higher value of the mean completion time than the SRPT rule in the same level of problem sizes

3 The LRPT and LPT rule respectively ranks fourth and fifth (after SRPT, SPT and FCFS rule) in the following ranges

* $n \geq 10$ and $m \leq 5$
* $n \geq 20$ and $m \leq 10$
* $n \geq 30$ and $m \leq 15$
* $n \geq 35$ and $m \leq 20$
* $n \geq 40$ and $m \leq 25$
* $n \geq 50$ and $m \leq 30$

In general, as noted in Fig 5 2(a, b, c, d, e and f) that when m increases and n decreases the LRPT rule is less important than LPT rule with respect to the average mean completion time criterion

In some problem sizes (Table 5 3b) such as $n = 5$ and $m = 5$, $n = 15$ and $m = 10$, $n = 40$ and $m = 25$ and $n = 45$ and $m = 30$ the LPT rule is equal to the LRPT rule

Table 5.3b: Observed AMCTs of 500 runs Vs. the proposed rules.

Problem Size n X m	Average Mean Completion Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 5	42	38	46	37	46
10 X 5	60	54	67	51	67
15 X 5	77	69	87	66	85
20 X 5	92	84	107	80	103
25 X 5	109	100	126	93	121
30 X 5	124	114	145	106	139
35 X 5	139	130	165	120	156
40 X 5	154	143	183	132	172
45 X 5	169	158	203	145	191
50 X 5	185	173	222	159	207
55 X 5	199	187	241	171	223
60 X 5	213	201	260	184	239
5 X 10	71	67	76	65	77
10 X 10	93	87	99	83	101
15 X 10	113	106	122	101	122
20 X 10	131	124	143	117	142
25 X 10	149	142	164	134	161
30 X 10	167	159	184	150	181
35 X 10	184	176	204	166	198
40 X 10	200	192	224	180	216
45 X 10	217	208	244	195	234
50 X 10	233	224	264	211	252
55 X 10	248	240	284	225	269
60 X 10	265	256	303	240	286
5 X 15	101	97	104	93	107
10 X 15	125	120	131	113	134
15 X 15	147	141	155	134	157
20 X 15	167	160	177	151	178
25 X 15	186	179	199	170	199
30 X 15	204	197	219	186	217
35 X 15	223	215	241	204	238
40 X 15	240	233	261	220	257
45 X 15	258	251	282	237	275
50 X 15	276	267	302	253	293
55 X 15	292	284	323	269	311
60 X 15	309	300	342	284	329
5 X 20	129	125	133	120	137
10 X 20	155	151	161	142	165
15 X 20	180	174	187	164	190
20 X 20	201	194	210	184	212
25 X 20	221	214	233	203	235
30 X 20	241	234	254	222	255
35 X 20	260	252	276	240	275
40 X 20	279	271	297	258	295
45 X 20	297	288	318	275	313
50 X 20	315	307	338	292	332
55 X 20	333	325	359	309	352
60 X 20	350	342	380	325	370

Continuing of Table 5.3b:

Problem Size n X m	Average Mean Completion Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 25	157	153	161	146	164
10 X 25	186	180	191	172	196
15 X 25	211	204	218	194	223
20 X 25	233	227	242	216	246
25 X 25	255	248	265	236	268
30 X 25	276	269	288	256	291
35 X 25	295	288	310	275	311
40 X 25	315	307	331	293	331
45 X 25	334	327	353	310	351
50 X 25	353	345	374	329	370
55 X 25	371	364	395	347	390
60 X 25	389	382	416	364	409
5 X 30	184	180	188	172	193
10 X 30	214	209	219	199	225
15 X 30	241	235	248	224	254
20 X 30	265	259	274	247	279
25 X 30	288	281	298	268	303
30 X 30	310	303	321	289	325
35 X 30	330	322	343	308	346
40 X 30	350	342	366	327	366
45 X 30	369	363	388	347	388
50 X 30	389	382	409	365	408
55 X 30	408	401	431	383	428
60 X 30	426	419	451	401	446

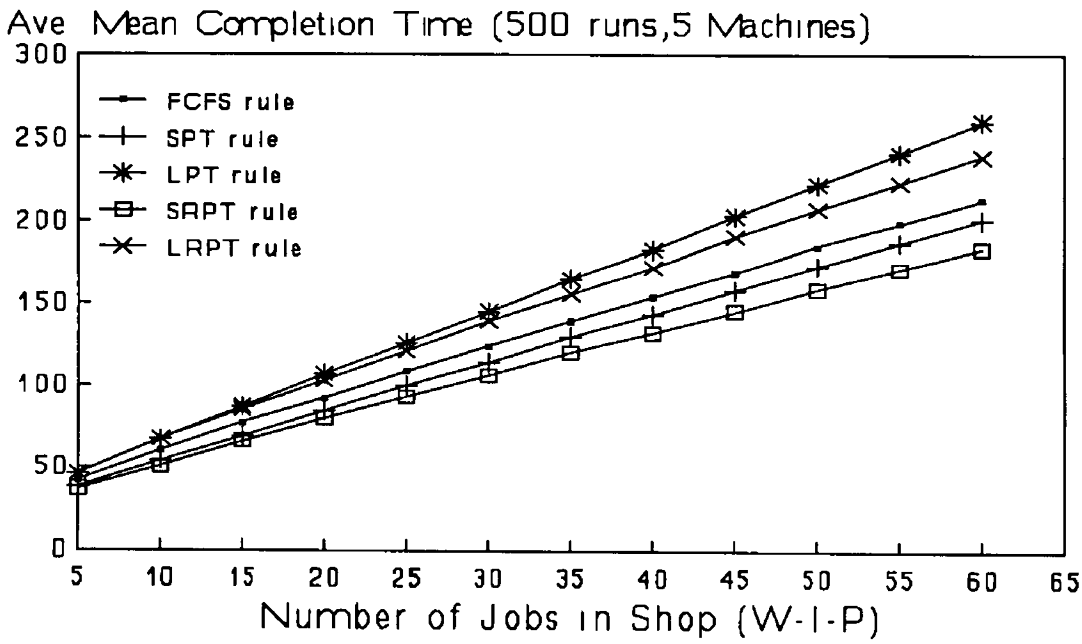


Fig. 5.2a: Effect of the rules Vs. MCT under 5 shop machines.

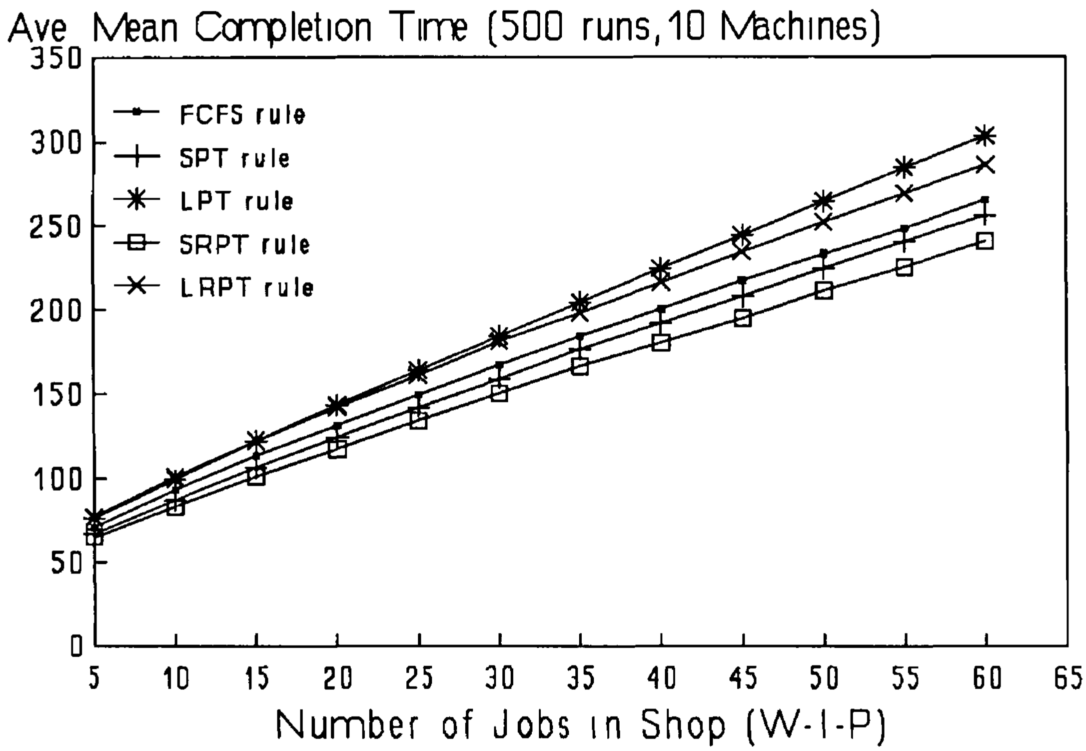


Fig. 5.2b: Effect of the rules Vs. MCT under 10 shop machines.

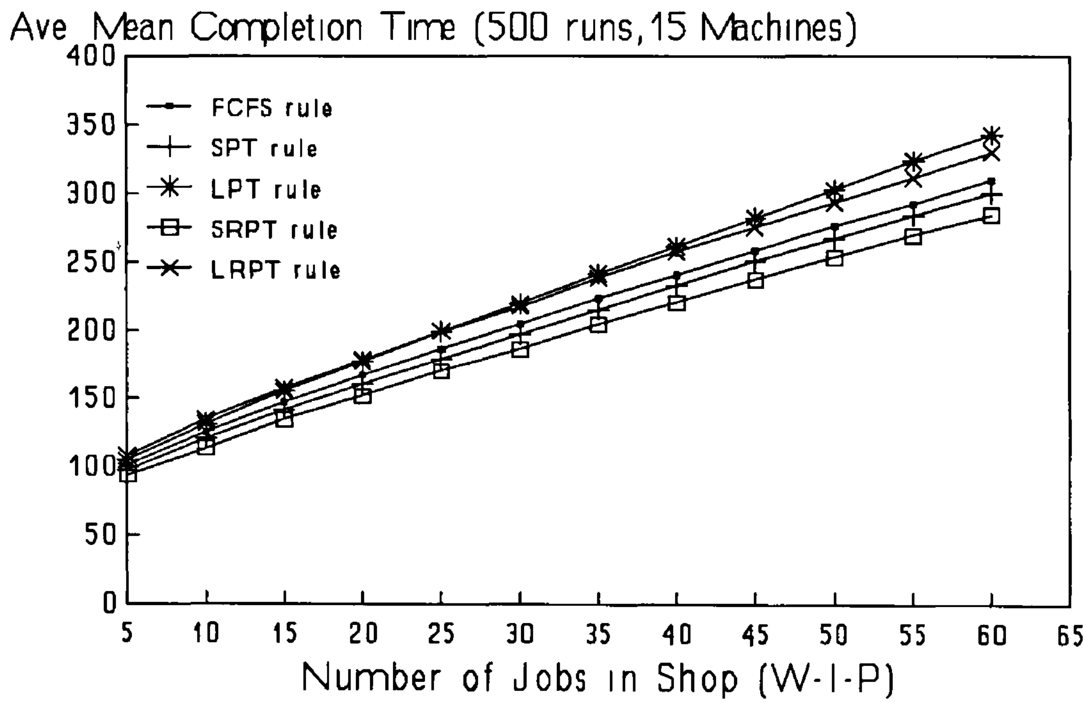


Fig. 5.2c: Effect of the rules Vs. MCT under 15 shop machines.

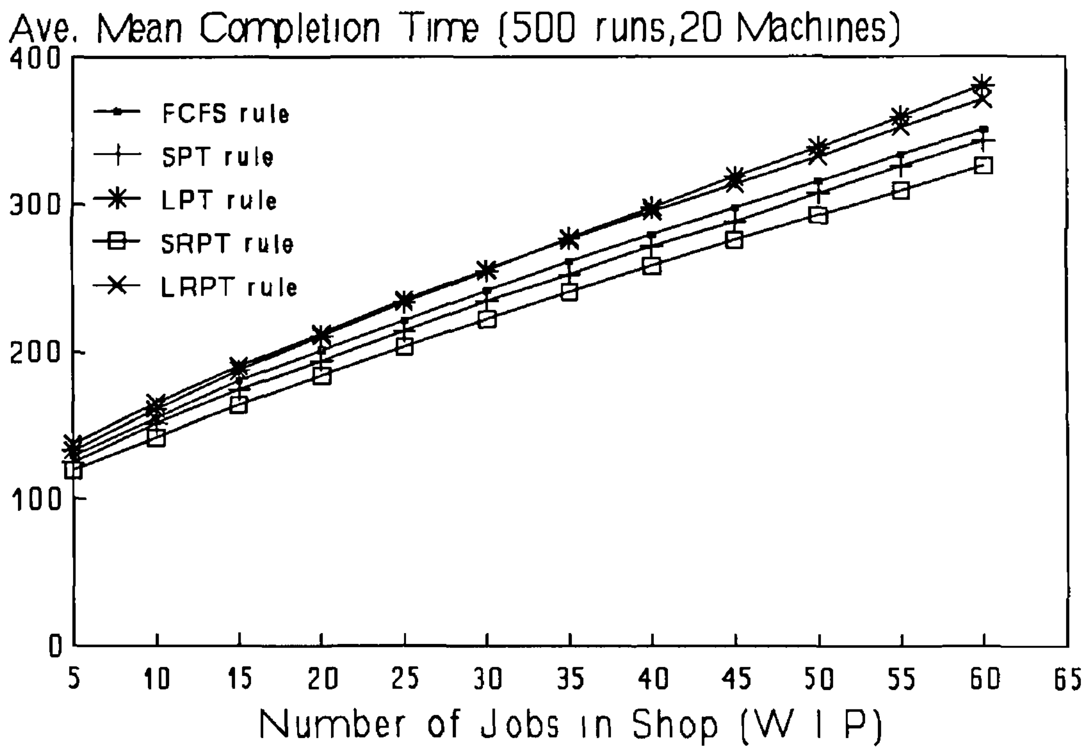


Fig. 5.2d: Effect of the rules Vs. MCT under 20 shop machines.

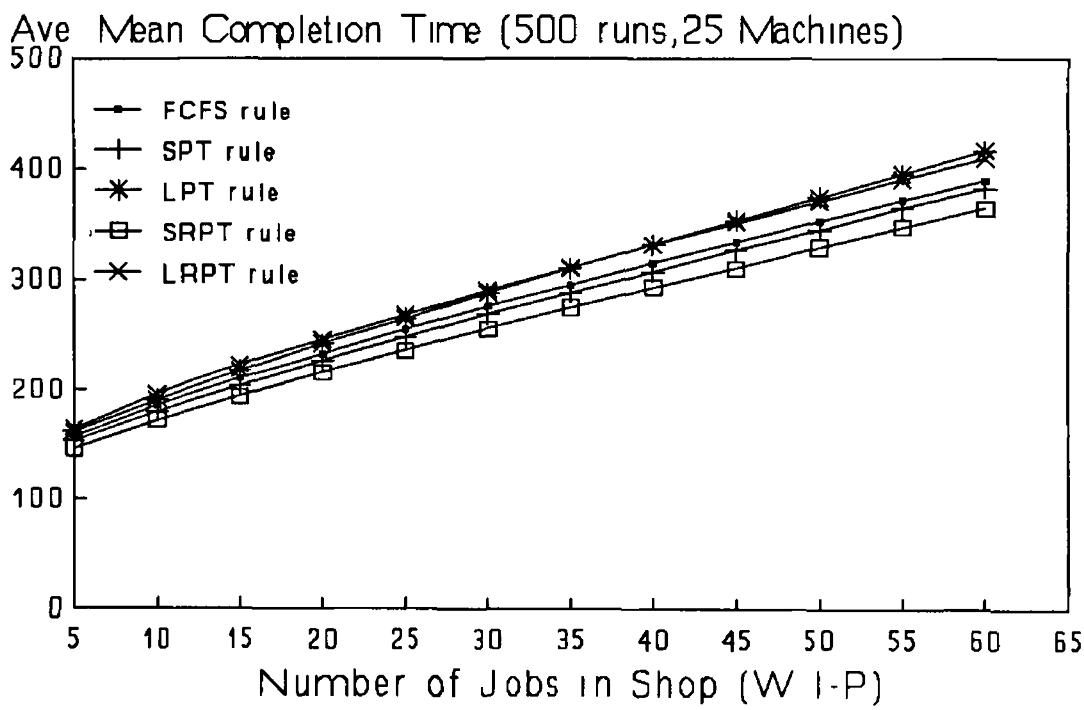


Fig. 5.2e: Effect of the rules Vs. MCT under 25 shop machines.

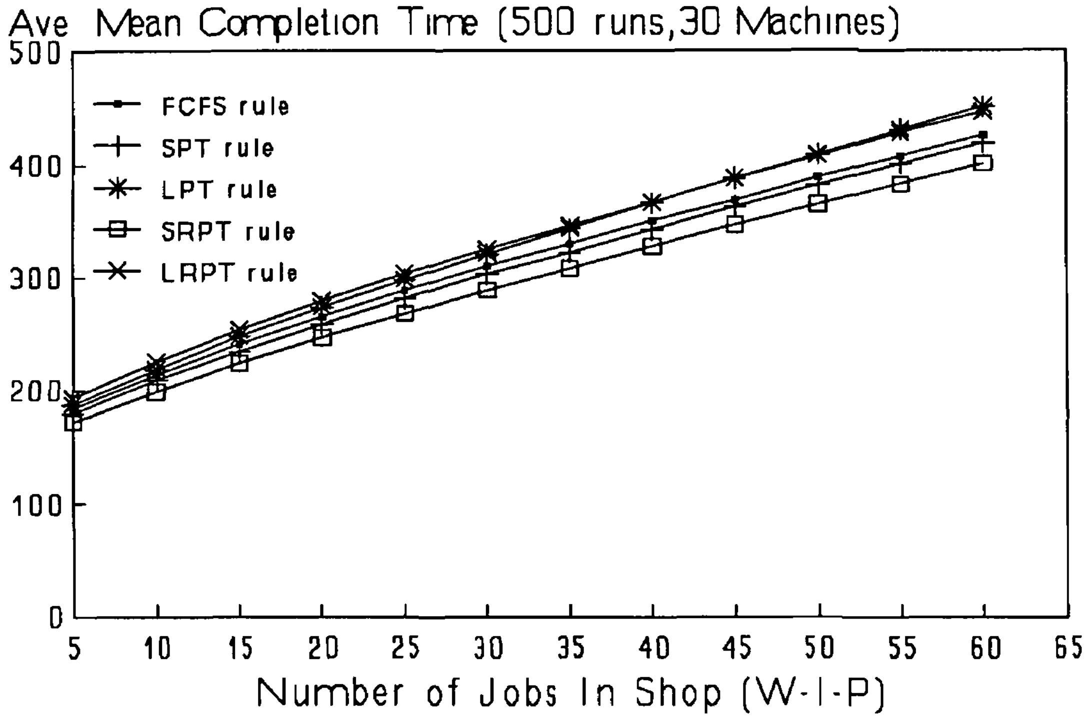


Fig. 5.2f: Effect of the rules Vs. MCT under 30 shop machines.

5.6.5 Evaluating the Priority Rules Vs. the Average Total Waiting Time Criterion under Different WIP and Shop Sizes:

The programme output data under 500 runs are scheduled and plotted in Table 5 3c and Figures 5 3(a, b, c, d, e and f) They show the results for the effect of the proposed rules with respect to the average waiting time at various problem sizes As mentioned before the programme running was carried out under the same randomly processing time matrixes for each run

The two lower curves illustrated in Fig 5 3a indicate that The SRPT and LPT are quit equal to each other when 5 shop machines is used (low shop load) at different WIP While in the range of $n \leq 25$ and $m \leq 30$ the SRPT rule is the dominate rule than other, in contract the LPT rule tends to be better than other in the range of $n \leq 30$ and $m \leq 30$ The SPT rule gives the poor performance than others when the problem size is greater than (5,10 x m) Finally the FCFS rule's curve shows a better performance than the LRPT rule curve with respect to the average waiting time for most problem sizes

Table 5.3c: Observed AWTs of 500 runes Vs. the proposed rules.

Problem Size n X m	Average Total Waiting Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 5	39	34	40	19	52
10 X 5	133	150	121	82	162
15 X 5	261	343	228	186	311
20 X 5	420	610	357	316	491
25 X 5	632	947	516	485	697
30 X 5	828	1335	687	651	921
35 X 5	1069	1803	873	852	1164
40 X 5	1317	2317	1077	1085	1418
45 X 5	1577	2870	1308	1304	1719
50 X 5	1901	3528	1548	1592	2011
55 X 5	2209	4264	1831	1846	2355
60 X 5	2492	5029	2114	2154	2679
5 X 10	61	59	65	34	84
10 X 10	225	249	205	148	276
15 X 10	425	522	372	306	507
20 X 10	705	898	585	532	808
25 X 10	1010	1375	821	796	1150
30 X 10	1374	1944	1105	1129	1537
35 X 10	1731	2540	1371	1418	1934
40 X 10	2120	3220	1677	1786	2347
45 X 10	2577	3972	2011	2157	2779
50 X 10	3047	4843	2361	2610	3312
55 X 10	3550	5819	2801	3119	3845
60 X 10	4097	6787	3184	3587	4368
5 X 15	83	79	82	46	111
10 X 15	286	315	271	192	351
15 X 15	571	662	500	412	671
20 X 15	912	1119	761	684	1040
25 X 15	1316	1678	1081	1044	1482
30 X 15	1751	2312	1416	1423	1941
35 X 15	2236	3059	1784	1841	2470
40 X 15	2732	3858	2148	2276	3023
45 X 15	3334	4800	2592	2848	3608
50 X 15	3976	5791	3058	3415	4304
55 X 15	4544	6841	3535	3956	4959
60 X 15	5272	7944	4034	4573	5640

Continuing of Table 5.3c:

Problem Size n X m	Average Total Waiting Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 20	97	96	99	55	132
10 X 20	344	377	317	231	422
15 X 20	672	777	592	483	797
20 X 20	1085	1294	924	832	1241
25 X 20	1552	1921	1291	1231	1754
30 X 20	2084	2656	1692	1685	2316
35 X 20	2674	3485	2135	2193	2959
40 X 20	3295	4393	2611	2773	3601
45 X 20	3914	5333	3079	3375	4255
50 X 20	4692	6522	3608	4007	5043
55 X 20	5385	7672	4166	4673	5854
60 X 20	6201	8920	4746	5393	6672
5 X 25	115	111	115	62	148
10 X 25	391	417	362	265	475
15 X 25	766	865	689	559	914
20 X 25	1221	1449	1061	953	1407
25 X 25	1794	2181	1510	1444	2016
30 X 25	2380	2962	1966	1914	2693
35 X 25	3034	3872	2462	2525	3370
40 X 25	3721	4847	2966	3133	4090
45 X 25	4508	6006	3578	3837	4935
50 X 25	5330	7141	4174	4569	5757
55 X 25	6140	8434	4806	5361	6671
60 X 25	7096	9846	5438	6181	7619
5 X 30	124	123	128	71	167
10 X 30	432	463	404	295	533
15 X 30	846	952	768	630	1012
20 X 30	1363	1593	1190	1059	1577
25 X 30	1967	2341	1670	1567	2233
30 X 30	2618	3212	2189	2131	2946
35 X 30	3379	4197	2766	2798	3754
40 X 30	4179	5278	3370	3497	4558
45 X 30	4977	6483	3990	4275	5486
50 X 30	5887	7718	4639	5070	6396
55 X 30	6760	9102	5333	5899	7380
60 X 30	7807	10562	6079	6851	8401

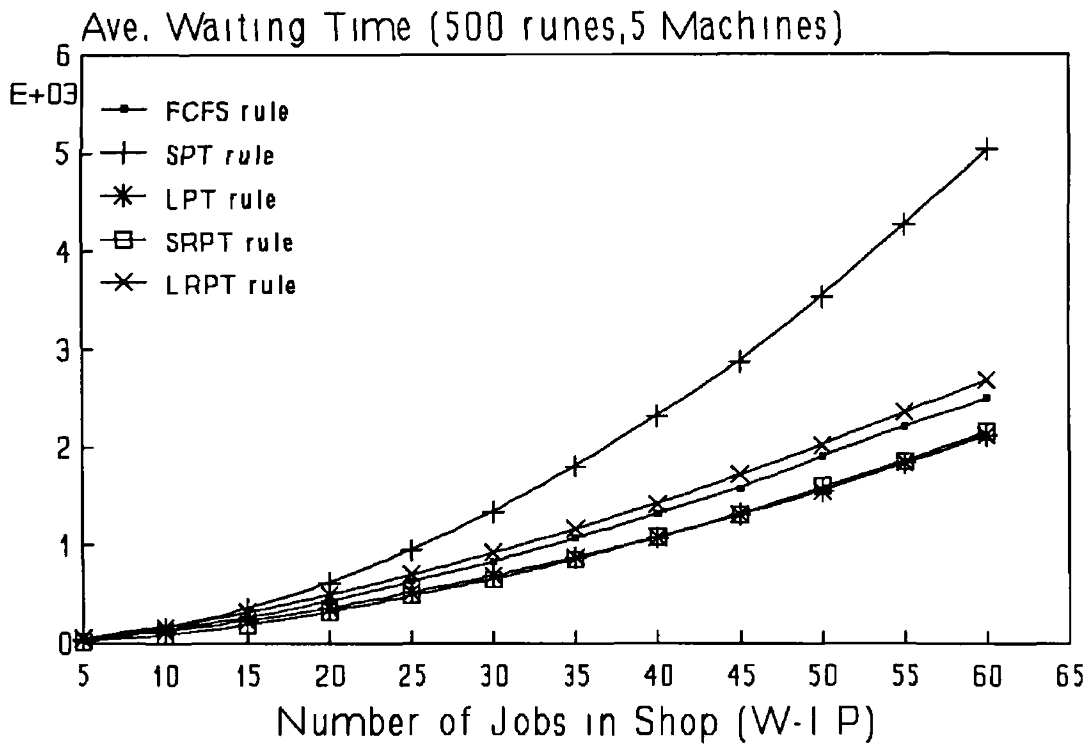


Fig. 5.3a: Effect of the rules Vs. AWT under 5 shop machines.

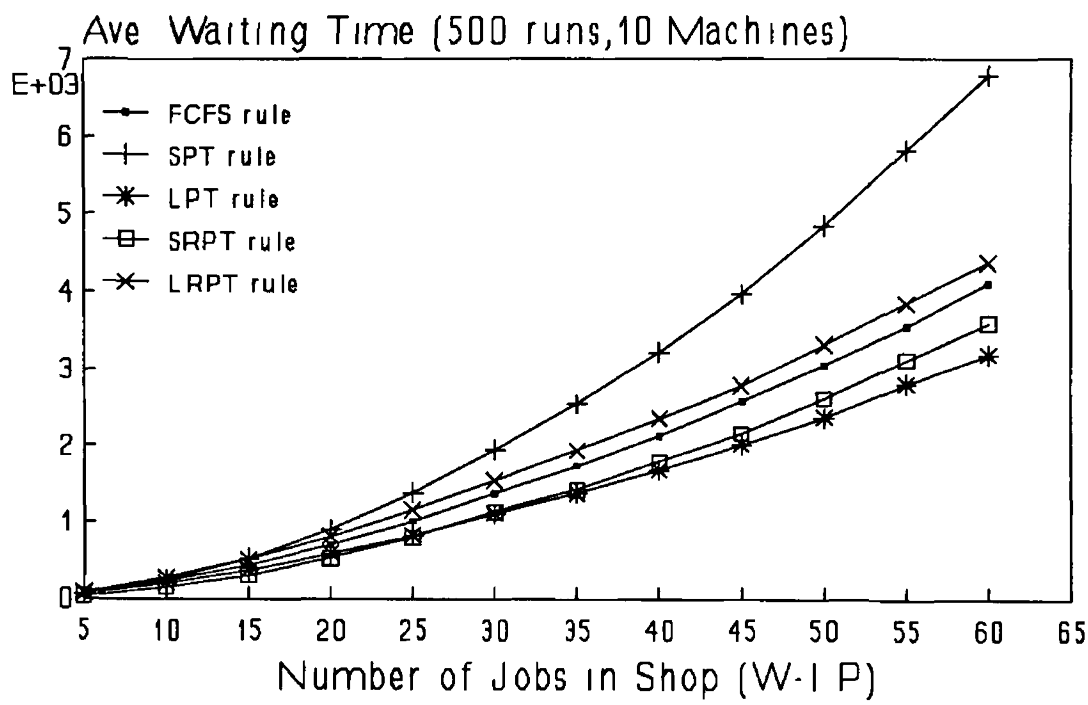


Fig. 5.3b: Effect of the rules Vs. AWT under 10 shop machines.

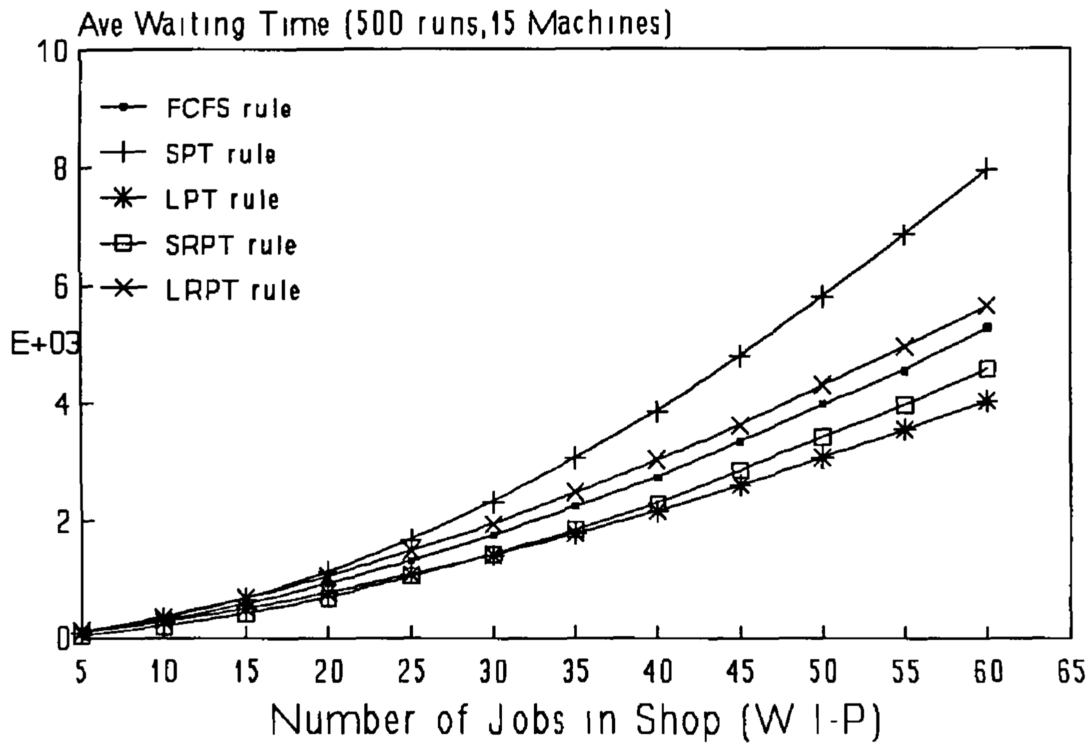


Fig. 5.3c: Effect of the rules Vs. AWT under 15 shop machines.

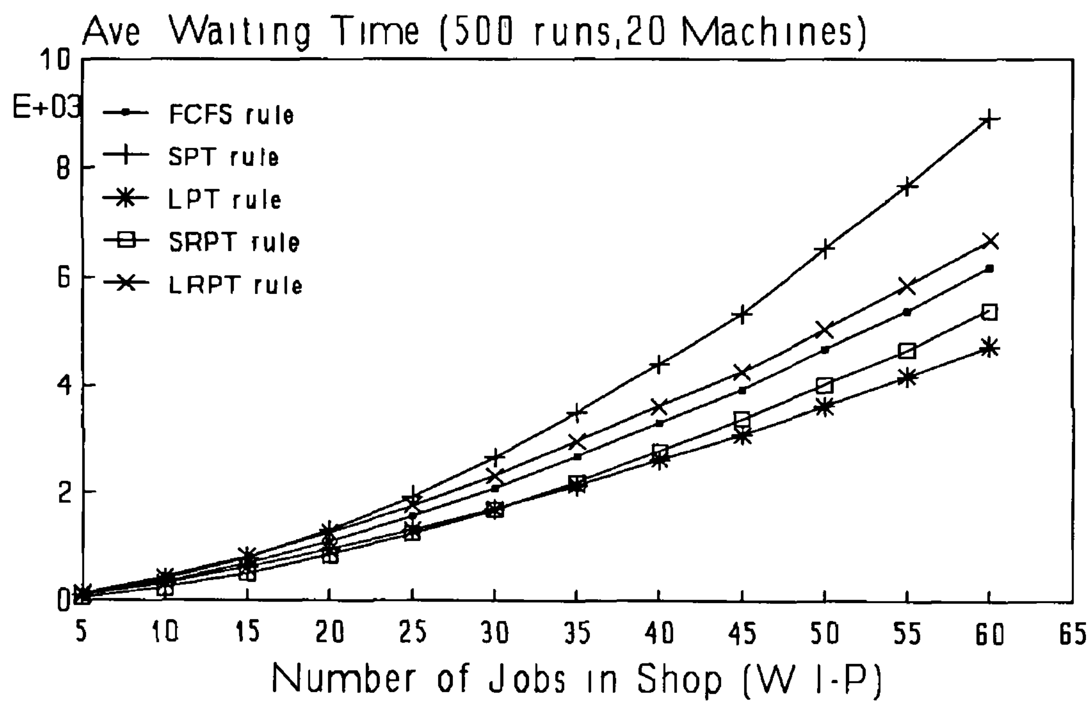


Fig. 5.3d: Effect of the rules Vs. AWT under 20 shop machines.

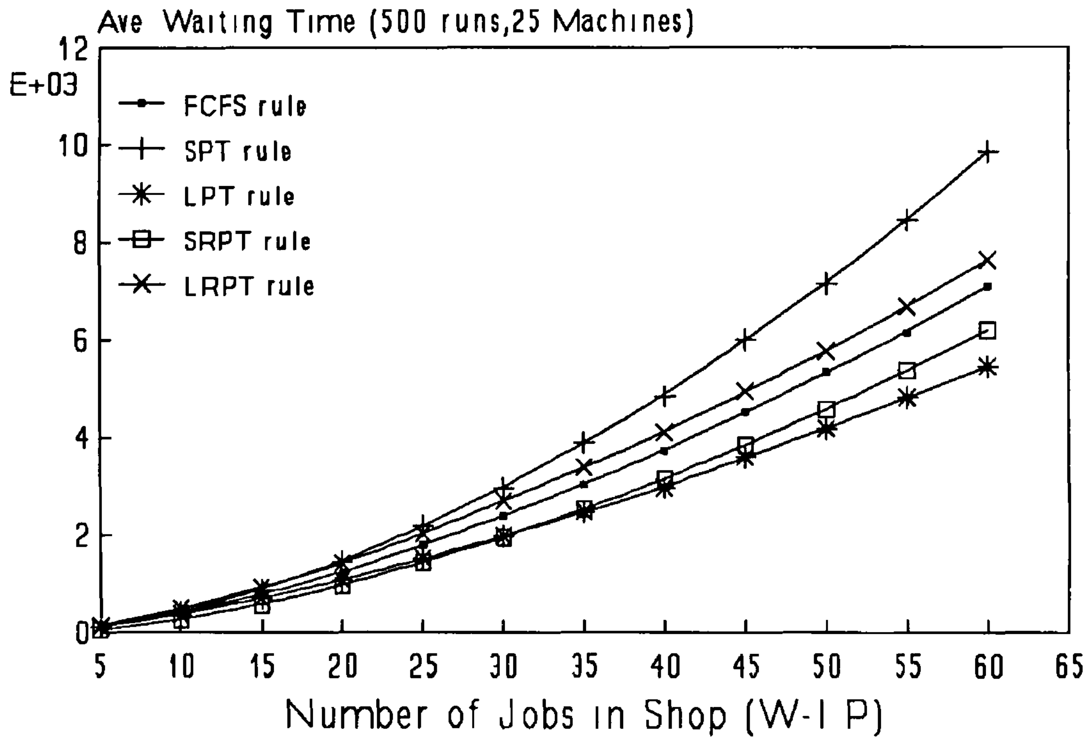


Fig. 5.3e: Effect of the rules Vs. AWT under 25 shop machines

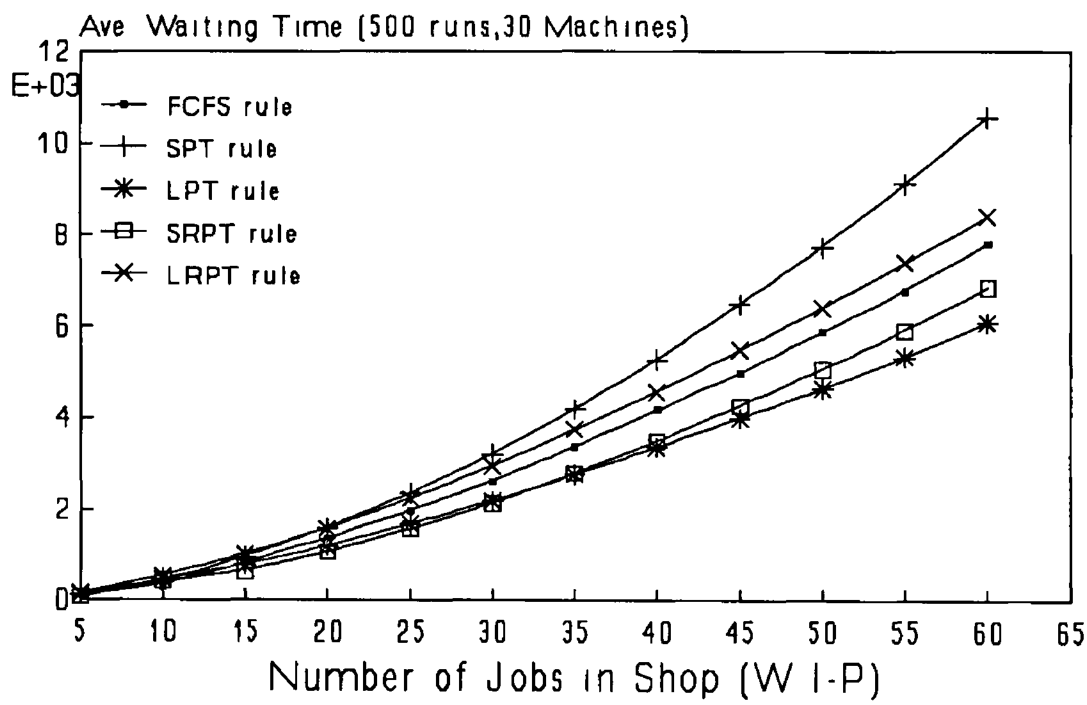


Fig. 5.3f: Effect of the rules Vs. AWT under 30 shop machines.

5.6.6 Evaluating the Priority Rules Vs. the Average Total Idle Time Criterion under Different WIP and Shop Sizes:

As discussed in the last three sections, the output programme data provide information (results) to evaluate the priority rules performance and to select a proper job sequence to be processed on available machines

The programme data on the Total Average Idle Time considered in this section are summarized in Table 5 3d

Figures 5 4(a, b, c, d, e and f) show the comparisons between the proposed rules as follows

Figures 5 4(a, b, c, d, e, and f) show that the FCFS rule performs the mid-quality performance between the other at all the proposed WIP and utilization levels

The LRPT rule performs better than any other rules for the ATIT criterion for the 5-machines shop, while the SPT rule improves its performance when the range $n \leq 40$ and $m \leq 5$ is used as illustrated in Figure 5 4a

In the range of $n \geq 10$ the LRPT rule is more dominate, compared to other at any utilization level as shown in Figures 5 4(b, c, d, e and f), while the SPT rule tends to be the second-quality performance in this range

Finally, the SRPT rule gives poor quality performance when the utilization level increases at a given WIP as cleared in Figures 5 4(a, b, c, d, e and f), while the LPT rule performs more effectiveness than the SRPT when the utilization level (shop load) decreases at a given WIP

Table 5.3d: Observed AITs of 500 runs Vs. the proposed rules.

Problem Size n X m	Average Total Idle Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 5	36	38	38	55	21
10 X 5	63	56	78	83	41
15 X 5	84	70	114	104	56
20 X 5	97	81	152	123	70
25 X 5	110	89	184	136	82
30 X 5	125	100	218	149	96
35 X 5	139	112	252	165	105
40 X 5	146	116	283	172	117
45 X 5	157	123	318	184	126
50 X 5	171	134	353	199	138
55 X 5	180	140	388	207	144
60 X 5	187	146	417	217	151
5 X 10	133	129	129	180	80
10 X 10	223	200	252	285	147
15 X 10	290	262	358	363	210
20 X 10	339	310	442	423	253
25 X 10	389	337	545	474	296
30 X 10	429	386	624	516	338
35 X 10	477	416	717	566	375
40 X 10	518	450	803	607	410
45 X 10	545	482	891	641	443
50 X 10	581	514	983	687	482
55 X 10	620	545	1061	713	508
60 X 10	652	573	1139	752	544
5 X 15	262	264	276	367	158
10 X 15	436	399	466	565	300
15 X 15	563	521	659	708	406
20 X 15	672	634	824	833	507
25 X 15	761	709	981	930	590
30 X 15	853	801	1138	1021	680
35 X 15	943	861	1300	1109	749
40 X 15	1014	917	1439	1183	813
45 X 15	1083	998	1585	1265	892
50 X 15	1152	1050	1722	1334	947
55 X 15	1222	1117	1875	1405	1018
60 X 15	1266	1146	1986	1450	1063
5 X 20	439	419	428	585	268
10 X 20	702	667	753	897	500
15 X 20	919	864	1043	1147	670
20 X 20	1089	1017	1291	1325	830
25 X 20	1237	1170	1501	1484	973
30 X 20	1368	1280	1727	1627	1087
35 X 20	1509	1404	1947	1774	1221
40 X 20	1620	1516	2150	1906	1329
45 X 20	1720	1610	2357	2013	1431
50 X 20	1826	1706	2562	2122	1517
55 X 20	1947	1814	2770	2239	1619
60 X 20	2051	1900	2996	2351	1712

Continuing of Table 5.3d:

Problem Size n X m	Average Total Idle Times				
	FCFS	SPT	LPT	SRPT	LRPT
5 X 25	612	608	631	847	409
10 X 25	1003	978	1091	1299	727
15 X 25	1323	1257	1461	1624	971
20 X 25	1557	1495	1793	1895	1184
25 X 25	1769	1677	2072	2099	1371
30 X 25	1956	1859	2377	2324	1569
35 X 25	2160	2031	2681	2548	1739
40 X 25	2302	2193	2984	2720	1892
45 X 25	2479	2354	3229	2864	2048
50 X 25	2639	2478	3497	3016	2181
55 X 25	2780	2626	3742	3183	2338
60 X 25	2898	2760	4026	3344	2444
5 X 30	838	816	834	1127	528
10 X 30	1382	1313	1432	1742	967
15 X 30	1750	1667	1936	2181	1311
20 X 30	2067	1972	2337	2537	1600
25 X 30	2368	2280	2748	2853	1876
30 X 30	2637	2501	3110	3118	2088
35 X 30	2861	2761	3493	3393	2322
40 X 30	3084	2964	3844	3596	2533
45 X 30	3306	3173	4207	3848	2742
50 X 30	3481	3346	4476	4037	2935
55 X 30	3730	3522	4869	4275	3124
60 X 30	3880	3685	5154	4455	3263

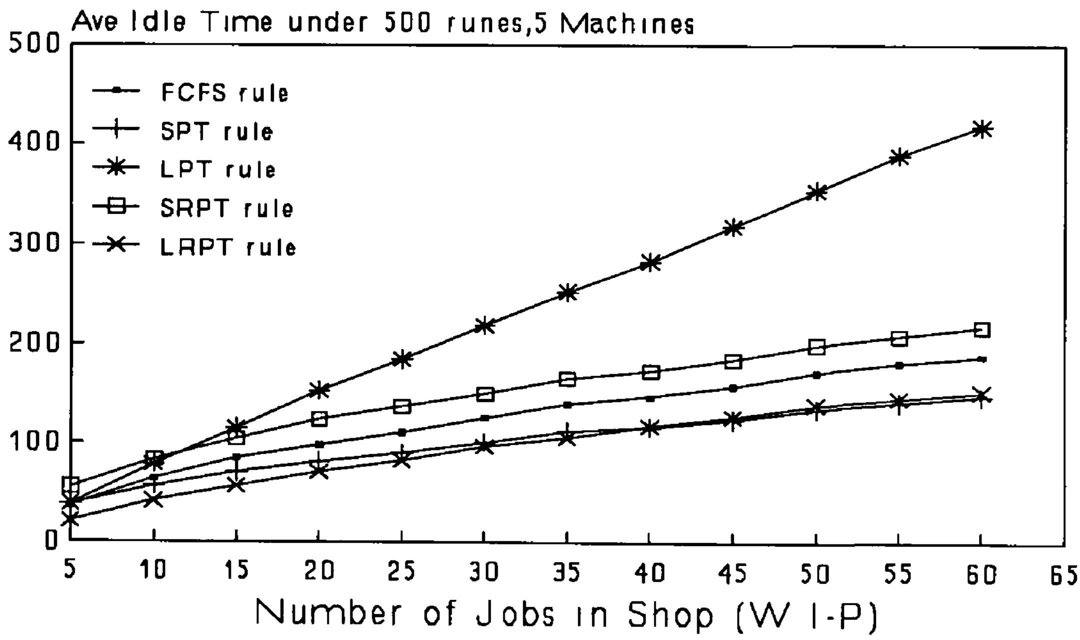


Fig. 5.4a: Effect of the rules Vs. AIT under 5 shop machines.

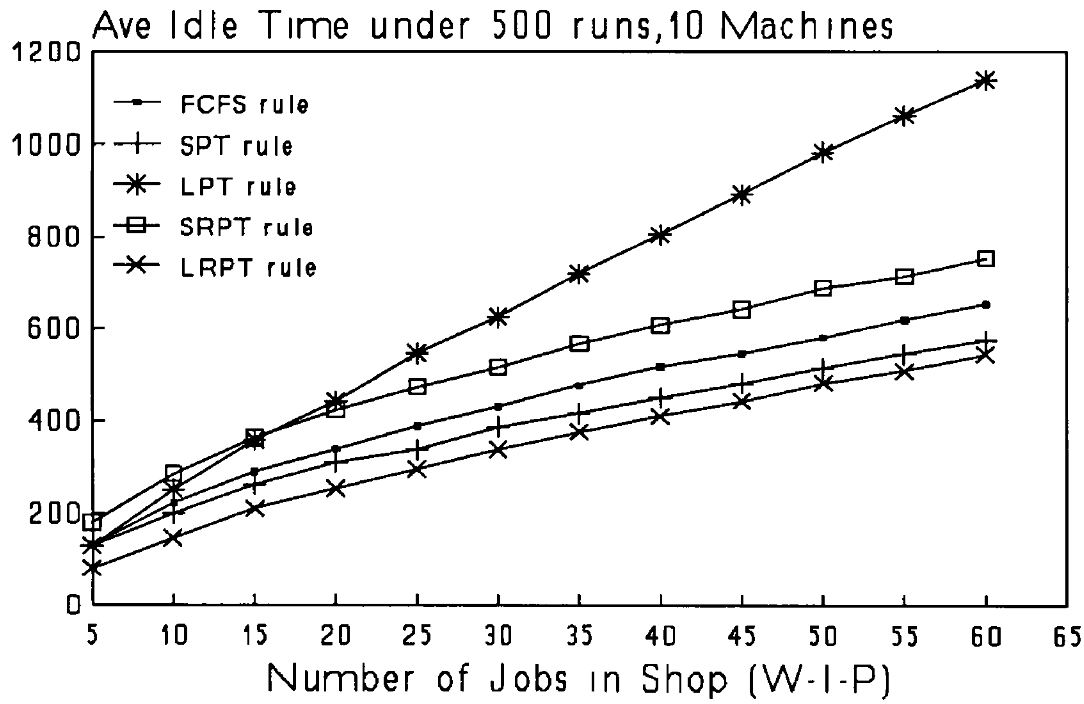


Fig. 5.4b: Effect of the rules Vs. AIT under 10 shop machines.

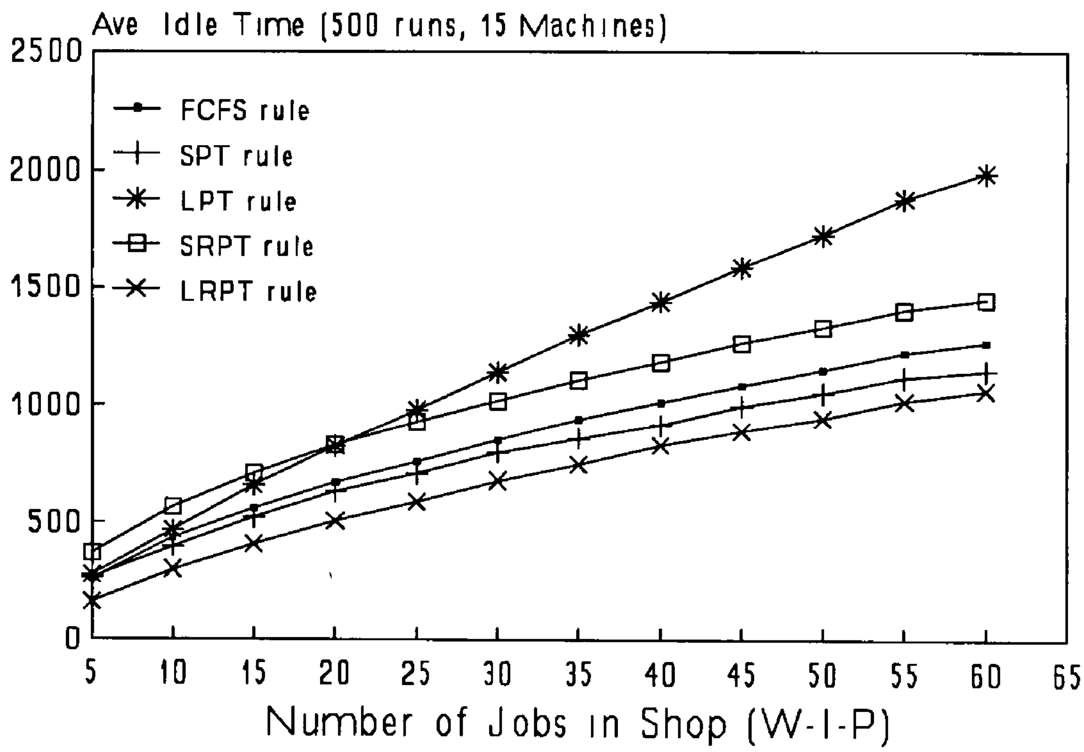


Fig. 5.4c: Effect of the rules Vs AIT under 15 shop machines.

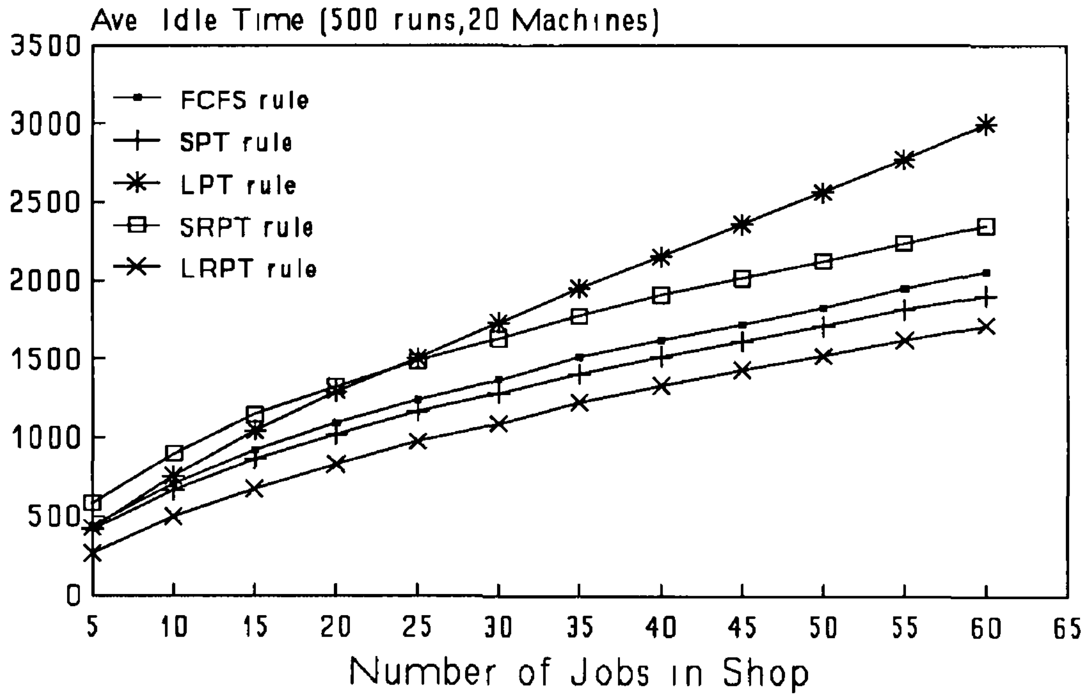


Fig. 5.4d: Effect of the rules Vs. AIT under 20 shop machines.

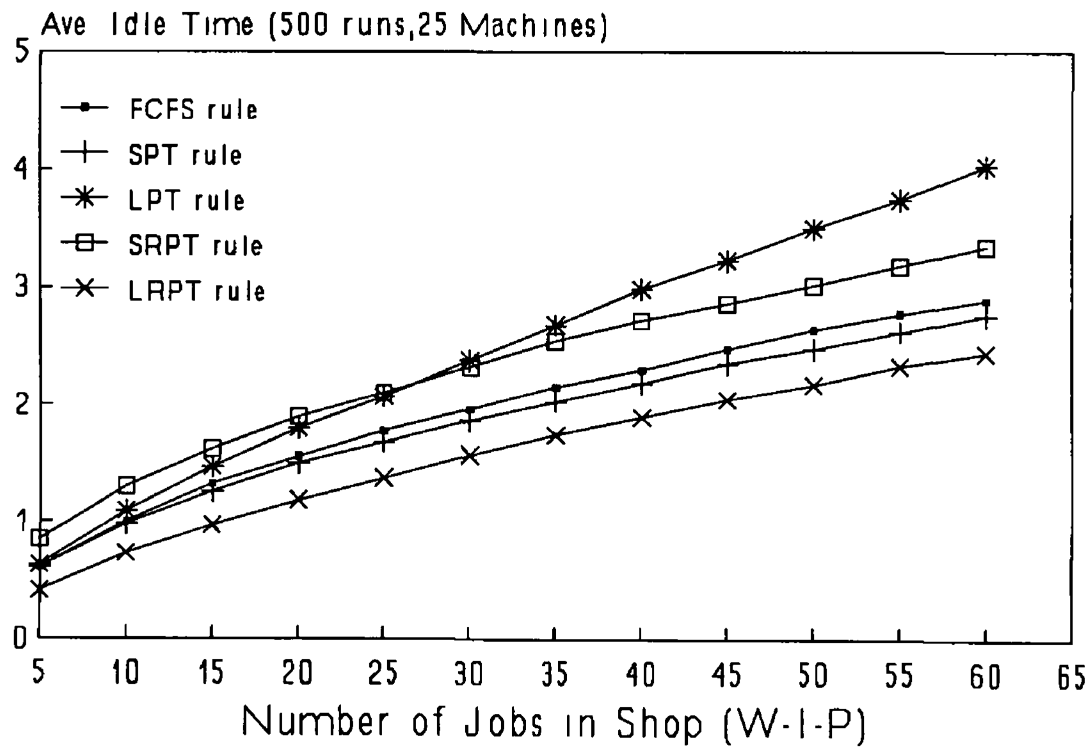


Fig. 5.4e: Effect of the rules Vs AIT under 25 shop machines.

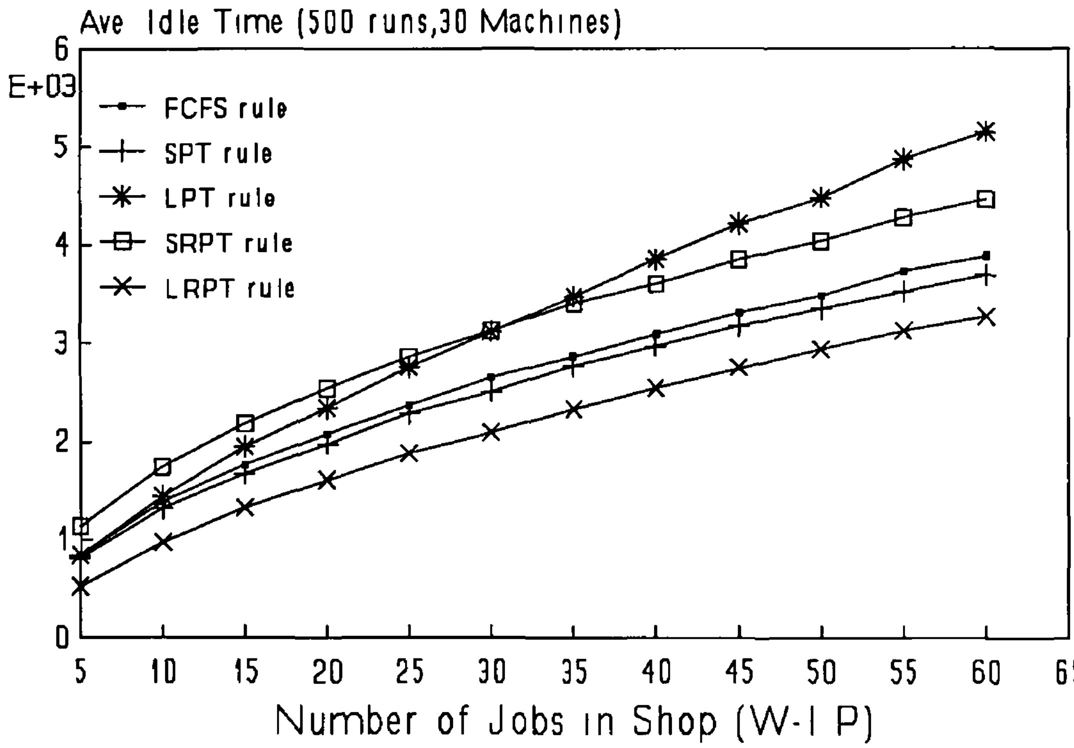


Fig. 5.4f: Effect of the rules Vs. AIT under 30 shop machines.

CHAPTER SIX

6. A Computer Simulation Analysis for Evaluating the Job-Shop Priority Rules:

In Chapters 4 and 5 the analytical studies have been limited to the flow-shop problem These studies in which the natural static problem, unidirectional process of jobs on machines and ready time and due date are equal to zero were concerned

In this Chapter the job-shop problem in which the dynamic version, flow job sequence is not unidirectional and ready times and due dates for jobs are not equal to zero will be considered in the analysis

In the dynamic problem, jobs arrive to the shop randomly over time, also scheduling is generally carried out by means of priority rules Using these rules the job sequence is selected for processing on a machine according to a specific routing of a good criteria

The effects of priority rules in dynamic job-shop environment are very difficult to analyze by the traditional analytical techniques The use of computer simulation has became a useful tool for this problem [5] and [6]

Experimental procedure with a computer simulation model has made it possible to compare alternative priority rules and generally develop a suitable environment for reducing the production cost

In terms of probability distribution selection, Table (6 1) gives the characteristics of different probability distribution that might influence a modeller to select a particular distribution to represent an activity (random variable)

Distribution	Parameters	Applications
Beta	(Alpha1,Alpha2)	* An absence data model * Activity durations in the PERT networks
Discrete	($CumP_1$,Val_1, $CumP_2$,Val_2,)	* Discrete assignments of the job type * An arriving entity for the batch size
Erlang	(ExpoMean,K)	* To complete the task time
Exponential	(Mean)	* To model random arrival * To breakdown processes * To lifetimes
Gamma	(Beta,Alpha)	* To complete some task * Machining, repair and breakdown times
Lognormal	(Mean,StdDev)	* Reliability models * Maintainability Engng
Normal	(Mean,StdDev)	* cycle times * The limit theorem applies
Poisson	(Mean)	* Time units(The number of arrival or departures)
Triangular	(Min,Mode,Max)	* For unknown distribution * The lack of reliable data
Uniform	(Min,Max)	* Is used when over a finite range are considered to be equally likely
Weibull	(Beta,Alpha)	* Reliability models such as the life time of a device * For non_negative task that are skewed to the life times

Table 6.1: Probability Distribution selection due to [62], [107] and Hines, W. W , and Montgomery, D. C.[1980][122].

Before dealing with the simulation of the job-shop under study the following section will discuss the conceptual performance plan of the job-shop problem

6.1 The Performance Plan for the Job-Shop problem:

The system being modeled is a machine job-shop which consists of six machine groups (Fig 6 1) Each group consists of a number of identical machines as shown In this Figure there are three specific workpieces (numbered 1, 2 and 3) which have to be processed on these machines The work flow pattern is not in the same order and skipping is allowed (multidirectional processes) as shown in Figures (6 2a, 6 2b and 6 2c) The ratio of the three workpieces distribution numbers were selected as follows 24% for workpieces 1, 44% for workpieces 2, and 32% for workpieces 3

Machine Group No	Machine Type	Number of Machines
1	Casting Units	14
2	Lathes Station	5
3	Planers Station	4
4	Drills Station	8
5	Shapers Station	16
6	Polishing Station	4

Fig. 6.1: Planning sheet for a number of identical machines to be used for process three workpieces in job-shop.

Operation Sequence	Machine Type	Different Time Distributions		
		Exponential (Mean)	Uniform (Min,Max)	Normal (Mean,StdD)
1	(1) Casting	125	120,130	125,2 5
2	(3) Planing	35	20,50	35,7 5
3	(2) Turning	20	15,25	20,2 5
4	(6) Polish-ing	60	40,80	60,10
Exitsys				

Fig. 6.2a: Job sequencing sheet for workpiece 1, in job-shop under Uniform(300,550) distribution due date.

Operation Sequence	Machine Type	Different Time Distributions		
		Exponential (Mean)	Uniform (Min,Max)	Normal (Mean,StdD)
1	(5)Shaping	105	80,130	105,12 5
2	(4)Drilling	90	70,110	90,10
3	(2)Turning	65	55,75	65,5
Exitsys				

Fig. 6.2b: Job sequencing sheet for workpiece 2, in job-shop under uniform(300,550) distribution due date.

Operation Sequence	Machine Type	Different Time Distributions		
		Exponential (Mean)	Uniform (Min,Max)	Normal (Mean,StdD)
1	(1)Casting	235	170,300	235,32 5
2	(5)Shaping	250	100,400	250,75
3	(4)Drilling	50	20,80	50,15
4	(3)Planning	30	13,47	30,8 5
5	(6)Polish--ing	25	19,31	25,3
Exitsys				

Fig. 6.2c: Job sequencing sheet for workpiece 3, in job-shop under uniform(300,550) distribution due date.

In this machine job-shop, the problem will be discussed under the following factors which are considered important

1. Three levels of Arrival Time Distributions (the shop loads):

The experimental procedure under study will be dealt with at three levels of job arrival process or shop loads This arrival process makes the workpieces enter the system one at a time according to the following proposed probability distributions shown in Figure 6 3

		The shop load
The Proposed Three Different Arrival Processes	1- Exponential(9), (8 2) and (7) 2- Uniform (7,11), (6 5,10) and (4,10) 3- Normal (9,1), (8 25,.875)and (7,1 5)	Low load (%70) medi load(%77) High load (%85)
Where: (9), (8 2) and (7) are the mean for the expo patterns, (7,11), (6 5,10) and (4,10) are the (min.,max) for the uniform patterns and (9,1), (8.2,0.875) and (7,1 5) are the (mean,σ) for the normal patterns		

Fig. 6.3: The proposed job arrival distributions in job-shop.

2. The work flow pattern:

As mentioned before the machine job-shop is not a unidirectional flow and operation skipping between machines is allowed, e g , such a processing is possible

3 → 1 → 4 → 6 → 1 → 6 → 2

3. Processing Time Distributions:

The following three distributions have been selected as they are widely used in the job-shop simulation

1 Uniform, it has been used by Schriber[109]

2 Exponential, (commonly utilized in [52], [42] [43], [55] and [75]

3 Normal, it has been used by Jones[42]

4. Due Date Distribution for the expected lead time:

Jones, C [1973][42] has reported that the delivery commitments is the second important criteria for judging the efficiency of priority rules after shop utilisation Uniform Distribution with minimum (300 minutes) and maximum (550 minutes) (randomly selected for each job) is selected as being the most appropriate because, it offers the best balance for the lead times expected in practice

5. The Selected Job-Shop Priority Rules:

Due to Conway, R W[52], John, et al[43] and Moore, et al[75], the following seven commonly tested rules which have been selected for our investigation in this chapter are

1 Shortest Processing Time rule (SPT)- For its practical application and generally excellent performance in so many investigations Conway, et al[5]

2 Longest Processing Time rule (LPT)- to show that it maximizes whatever SPT rule minimize [5] In general, it has less practical applications than other rules

3 First Come First Service rule (FCFS)- for its simple application and the most democratic of all rules John, et al[42]

4 Last Come First Service rule (LCFS) Conway, et al [5] noted that this rule is used when the job arriving have been stacked in such a way (a high job-shop load) that the latest arrival is the most accessible and thus the one selected

5 First Arrived at Shop First Service (FASFS)

6 Earliest Due Date (EDD)(i e a lead time)

$$d_{j(1)} \le d_{j(2)} \le d_{j(3)} \le . \quad .....\le d_{j(n)}$$

7 Static Slack (StS)- It is equal to due date minus the time of arrival at the machine centre

6. Selecting the Measure of Performances:

Most sequencing systems can be put into several of numerous complex criteria according to whether their measure of performance is specified, Mellor[116] lists 27 distinct scheduling goals

Most researchers such as Conway[52], Earl Legrande[47] and Hollier[123] take into account that the basic problem of job shop operation is one of balancing the costs of carrying Work-In-Process, having idle machine or machine utilization level (or shop load) and meeting specified order due dates

To have a low degree of average idle machines, a shop process would need much waiting in machine queues so that machines were ever idle thus, if the orders (workpieces) have specified due dates the result will be higher in work-in-process costs and poor scheduling performance To have orders meet their due dates or lateness, the shop would need enough machines so that orders could be processed without delay This, in turn, would result in higher average idle machines

In this study of the job-shop scheduling, each order (job) must be planned and controlled according to the selected five criteria as follows

1. **Minimize Job Lateness:**
   It is the time between when a job is completed and when it was due to be completed

2. **Minimize Mean Flow Times:**
   It is the amount of time a job spends in the shop

3. **Minimize Machine Idle Times.**
   It is the fraction of time when the machines are non-productive

4. **Minimize Work-In-Process:**
   It is the unfinished workpieces during processing, it is usually due to workpieces waiting for available machines

5. **Maximize Completion Jobs or production rate:**
   It is the output produced in a given period of time Shop

capacity is often defined as the maximum production rate that can be obtained

7. Job-Shop Load Levels:

The shop will be simulated under three levels of loading

1 Low at approx 70%
2 Medium at approx 77%
3 High at approx 85%

Also in this study, there is a high degree of job-shop interaction due to the variety of orders, arrival time distributions, variable processing time distributions and due date distribution This interaction causes certain machines to become critical (idles), and waiting queues Orders which are held up at one point are affecting future machines through which they must pass Therefore, orders must be scheduled with an allowance for waiting and they must be dispatched in such a way that the schedules will tend to be met This affects the accuracy of the scheduling procedure (the priority rules selection) and hence the entire job-shop production cost

The following section (6 2) will discuss the concepts and methods for simulating the job-shop problem under study which mentioned above using the SIMAN language This software will be used for evaluating the performance of the priority rules with respect to the various criteria under different shop loads and processing time distributions

6.2 A Computer Simulation of Production Scheduling using the SIMAN Software:

The SIMAN modelling framework is divided into two frames They are the system model and the experiment model frame

The system model defines the physical elements of the system (machines, parts flow, worker, storage points, transporters, information, etc)

The experiment frame specifies the experimental conditions under which the model is to run, including elements such as initial conditions, resource availability, type of statics gathered, and the length of run The experiment frame also includes the analyst's specifications such as the schedules for resource (machines) availability, the routing (sequences), parts,etc

The two sections below will report the basic concepts which deal only with the model and experiment frame of computer simulation for the job-shop under study (C D Pegden, et al[1990][62] give more information about the simulation in manufacturing systems

6.3 The Model Frame of SIMAN's Simulation production Scheduling:

The objective is to use the simulation model to develop and evaluate the effectiveness of various priority rules with respect to different criteria under different shop loads and processing times distributions

Processes are modelled by using a block diagram

The block diagram is a linear, top-down flow graph depicting the process through which the entities (parts flow) in the system move

The block diagram may be constructed in either a graphical flow-chart form or in an equivalent statement (pseudo-code) form In our simulation study for the proposed job-shop described in section 6 1 the model programme will be created using the SIMAN statement form

The procedure in Appendix E will discuss step-by-step the proposed job-shop model

6.4 The Experiment Frame of SIMAN's Simulation production Scheduling:

As discussed earlier, a SIMAN simulation programme comprises both a model and experiment The Pseudo Codes statement that have developed through Section 6 2 for the proposed job-shop represents only the model frame (the first portion) of the simulation programme In this Section 6 2 the experiment frame has yet to be specified, it includes the length of the simulation run, the number of replications of the simulation, the characteristics of machines and queues, the measure of performances, the selected criteria output data and then plot files, etc The experiment programme codes are called elements and in the current problem they will be specified in statement form using text editor

In Appendix F, the step by step experiment programme will be developed for the proposed job-shop

6.5 The Selected Procedure for Estimating the Mean and Variance of random variables for the multiple Criteria:

Simulations are run in order to gain an understanding of the behaviour of the system under study The objective of the simulation analysis is to estimate the value(s) of one or more unknown parameters by applying appropriate statistical techniques to the data collected from the simulation

Pritsker, A Alan B[1986][106] proposed five procedures according to a considerable amount of research for estimating the mean and variance of random variables Each of these five approaches will be briefly represented as follows

1. Replication:

In the replication approach, several runs are executed, each with a different stream of random numbers After deleting the warm-up period (run-in period), each run is made to represent a single batch From run u, we obtain a value of X_u (a random variable) and the mean of the X_u and Variance[X_u] value over U runs are used as an estimate of the parameters of interest, those are respectively as follows

$$\overline{X_u} = \frac{\sum_{u=1}^{U} X_u}{U} \qquad (6\ 1)$$

$$Var[\overline{X_u}] = \frac{1}{U}\sigma \qquad (6\ 2)$$

The replication procedure has the desirable property that observations are truly independent Another advantage is that it can be used for both terminating and steady-state or transient warm-up period analysis However, this procedure has a disadvantage which is that a transient warm-up period must be deleted from each run

2. Sub-intervals:

In the sub-interval approach, only one simulation run is executed After deleting the initial transient warm-up period, the remainder of the run is divided into H equal batches, with each batch average representing a single observation Thus, if each batch has b samples of stochastic process $X_u(s)$, where s = 1, 2, 3, ,s then a batch sample mean, X_u, in computed from

$$X_u = \frac{\sum_{s=1}^{b} X_u(s)}{b} \qquad (6.3)$$

If the sub-interval are independent then Eq. 6.1 and 6.2 are used to estimate the mean X_u and variance Var[X_u] of random variable, respectively.

3. Regenerative Cycle:

This approach is similar to the sub-interval method in that it divides a simulation run into intervals which are referred to as cycles. A cycle starts when a specific state of the system is reached in which future behaviour is independent of the past behaviour. However, this method is actually designed to alleviate the problems associated with both the replication and sub-interval methods.

4. Parametric modelling:

This approach for employing parametric modelling involves the collection of sample values from a simulation and then fitting an equation(s) to the observed data values. This approach is similar to the one used when attempting to describe real world systems by fitting equations to data obtained from the system. This approach has a lack of knowledge of the reliability of the model.

5. Covariance and the use of Spectral Estimation:

This approach estimates the auto_variance from the sample output and use these in a spectral analysis. It has not produced reliable point estimates.

In our case of simulation for the proposed job-shop the replication method was selected as the statistical methodology used in the experimentation This selected replication procedure was chosen according to many reasons as follows

- It has more efficient independent observations than other methods
- Each replication has a different stream of random numbers and it represents as a single batch
- It is simple to determine the start of the steady-state range

Hereafter, section 6 5 1 below will discuss the effect of transient condition on the random variables and how could the steady-state be evaluated

6.5.1 Estimating the Mean for The Steady-State Job-Sop Simulations:

In a discrete and non-terminating simulations especially in stochastic job-shop simulations, they do achieve a steady-state Which pass through a transient phase
In modelling the proposed job-shop the system will begin the simulation in the empty and idle condition, the initial jobs will arrive at an un-congested system with idle machines Hence, the early arriving jobs will quickly move through the system and the performance measure will all be biased downward during the early part of simulation After the system has had time to "warm-up", queues will form, and the system will begin to exhibit its true long-term behaviour Observations collected after the Warm-up period will be representative of steady-state behaviour, whereas observations collected during the transient phase will make the proposed job-shop appear to function better than it really does
When trying to analyze steady-state performance, we must deal with the bias introduced by the starting conditions (This is not a problem in terminating systems-Seila[[1990][124]-, because we are specifically interested in evaluating the

transient response of these systems to their fixed starting conditions) However, Eilon & Hodgson[55] suggested that the transient period (run-in period, as they called it) have only a slight effect on priority rule comparison results These results were obtained through a simulation of job-shop model which was carried out at Imperial College, London

The question naturally arises as to how to select the warm-up period (or deletion amount) Due to Welch,[1983][125] and Law and Kelton,[1991][126], the simplest and most general technique for determining the warm-up period (or truncation point) is a replication procedure (see section 6 5 1) using a visual determination, i e , selecting the point from a mean random variable plot of the simulation response over time

In the plots with large fluctuations (or high frequency oscillations) in the response, this process can be improved using a moving average $\overline{X}_i(w)$ (where w is the window or moving average size and is a positive integer) to smooth the response

6.6 The Experimental analysis of the Simulation results under different Priority rules Vs. multiple criteria:

The following factorials summarise the conceptual plan of the job-shop scheduling to be simulated

1. **Three types of Workpiece Arrival Distributions** : Exponential, Uniform and Normal
2. **Jobs Work Flow** : Multiple processing
3. **Three types of Processing Time Distributions** : Exponential, Uniform and Normal
4. **Workpieces Due Date**: Uniform (300,550)
5. **Five Performance Measures** : Mean Flow-Time,

Mean Lateness,
Mean production Rate(Job Completed),
Mean Total Machine Idle Times and
Mean Total Work-In-Processes

6. How the Performance Measure Is Determined: The simulation duration will be achieved under 13120 Minutes (Approx 4 5 weeks, 8 hours per day) excluding the warm-up period (transient portion) The warm-up period will be individually determined for each replication using the replication visual determination procedure (see Chapter 6 5 1)

7. Seven Priority Rules: FCFS, LCSF, SPT, LPT, FASFS, EDD and Static-SLK

8. Three Shop Loads : Approx , Low %70, medium %77 and high %85

In order to compare the seven priority rules under as close to identical conditions as possible, each rule was presented individually with the same set of jobs to be processed The jobs arrive at the same time, in the same arrival patterns and the processing times from one run to another A "run" consisted of having the shop follow a particular rule throughout having a 13120 (Approx 4 5 week's operation) scheduling period
The only thing that changed from one run to another was the order in which the jobs were processed on the various machines i e , as determined by the priority rule employed
Finally, the simulation was executed on the job-shop under seven priority rules through the following two comparison phases

The first comparison phase:

7 priority rules were simulated under

3 job arrival distributions (Expo , Unif and Norm)

3 Shop loads (approx 70%, 77% and 85%)

1 Type of processing time distribution (Exponential)

7=R Replications with different stream random numbers, R ≥ 5 is the number of replications were found a reasonable number for testing by Law and Kelton[1991][126]

In total, the first comparison phase involved 441 experimental simulation runs

The second comparison phase:

7 priority rules were simulated under

1 Type of job arrival (Exponential)

3 Shop loads (approx 70%, %77% and 85%)

3 Type of processing time distributions (Exponential, Uniform and Normal)

7 Replications with different stream random number

In total, the second comparison phase involved 441 experimental simulation runs

The total experimental simulation runs for both comparison phase, are equal to 882, each run with data being taken on approximately 1300-1850 jobs

The following sections will discuss the experimental results for the first and second comparison phases respectively

6.6.1 The Selected Steady-State Points for the Two Comparison Phases.

The job-shop was analyzed only when the steady state conditions were reached and the transient period was eliminated from the analyses In eliminating the bias due to the initial condition, the simulation runs for the shop model was carried out for each case of the three arrival patterns (the first comparison phases) or the three operation time patterns (the second comparison phase) and seven priority rules through seven replications Hence, the simulation runs (3 loads x 7 rules x 7 replications) consisting of 147 runs for each comparison phase

A replication procedure for estimating the length of the transient portion for each case of the two comparison phases was used and it may be summarized within the following five items

1 The flow-time or its standard deviation has been taken as the variable to be estimated
2 The selected variable variation was plotted as a function of the simulation time for each case The moving average plot was used for the flow-time variable
3 Here, the estimate of the starting point of the approximate steady-state was defined by visual inspection (see the next paragraph)
4 The desired statistical observations were collected by rerunning the simulation model with the transient portion truncated
5 Some random number replication streams were rejected due to the long delay in achieving the steady-state point Hence, other suitable random number replications were selected

Tables 6 2 and 6 3 summarize the starting and ending points of the approximate steady-state for each case of the three arrival time distributions (Table 6 2) or the three operation time

distributions (Table 6 3) under seven priority rules were tested
The selected steady-state plots are shown in Figures (6 4-6 7)
These plots show the co-ordinates of some of the flow-time variable variations and their moving averages (or their standard deviation) as a function of simulation time

It was hoped that a single starting point of the steady-state could be used for all the seven priority rules for each case of the two comparison phases
This property is to aid the comparison of the results as shown in Figures (6 4-6 6)
Also we noted from Figures (6 4-6 7) the following features

1 Some of the proposed simulation cases have an individual steady-state starting point as shown in Figures (6 4 and 6 5)
2 The standard deviation and flow-time variation curves for the SPT rule are more horizontal and smooth than the other rules (especially the LPT rule) as shown in Figures (6 4, 6 6a and 6 6b) However, Eilon, et al's results[55] found that the bias was in favour of the SPT rule and against the LPT rule
3 The start of the steady-state points for the SPT rule are too close to the starting load points than other rules, especially in the low and medium shop-loads as shown in Figure 6 4

			The Type of Simulation Period		
The Comparison Phases	The Type of Load	The Pattern of ArrTime	Max Simulation Running Period	Warm_Up Period	Steady_state Period for Observation Data Collecting
The First Comparison Phase	%70 Load App	Expo Unif Norm	21120 45000 23120	8000 31880 10000	13120 13120 13120
	%77 Load App	Expo Unif Norm	21120 60000 19120	8000 46880 6000	13120 13120 13120
	%85 Load App	Expo Unif Norm	21120 29000 19120	8000 15880 6000	13120 13120 13120

Table 6.2: The simulation period types for different arrival time patterns (the first comparison phase).

			The Type of Simulation Period		
The Comparison Phases	The Type of Load	The Pattern of OpTimes	Max Simulation Running Period	Warm_Up Period	Steady_state Period for Observation Data Collecting
The Second Comparison Phase	%70 Load	Expo Unif Norm	21120 39620 37120	8000 26500 24000	13120 13120 13120
	%77 Load	Expo Unif Norm	21120 35620 55120	8000 22500 42000	13120 13120 13120
	%85 Load	Expo Unif Norm	21120 30620 28120	8000 17500 15000	13120 13120 13120

Table 6.3: The simulation period types for different operation time patterns (the second comparison phase).

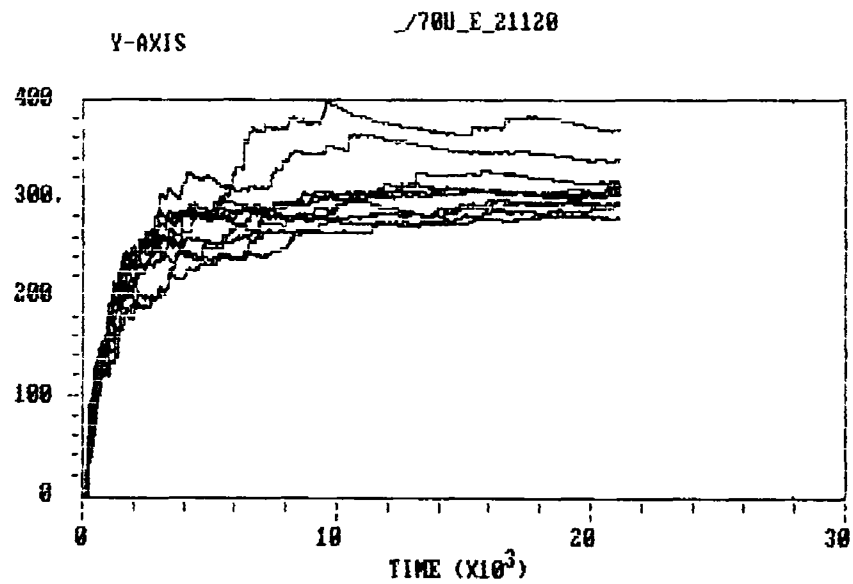


Fig. 6.4: The approx. steady-State points (31880) in the σ plots for the SPT rule flow-time variables under Unif. Arrival and Expo. Operation Time and 70% load.

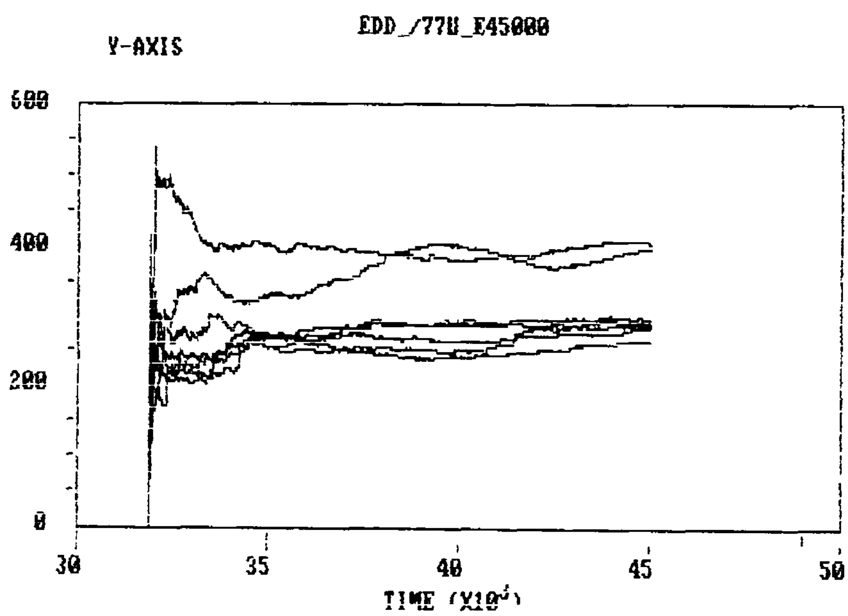


Fig. 6.5: The approx. steady-state points (46880) in the σ plots for the EDD rule flow-time variables under Unif. Arrival and Expo. Operation Time and 77% load.

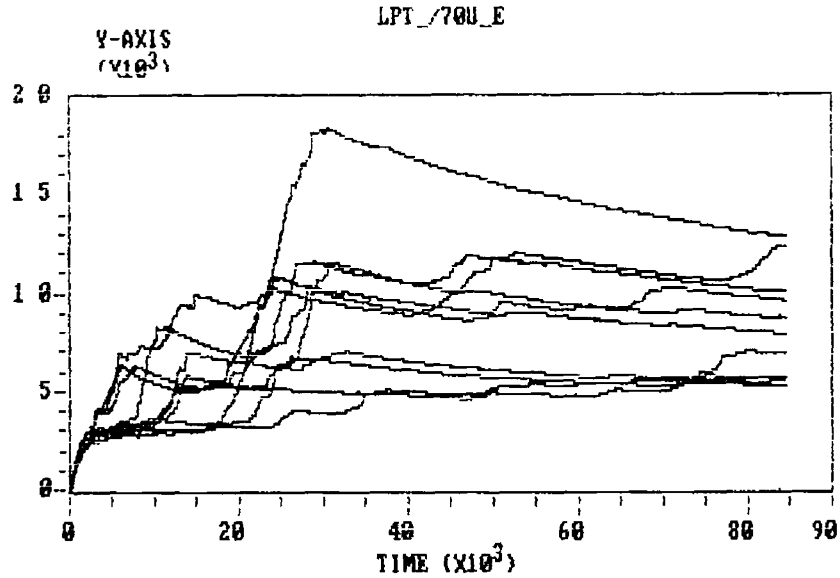


Fig. 6.6a: The approx. steady state points (31880) in the σ plots for the LPT rule flow-time variables under Unif. Arrival and Expo. Operation Time and 70% load.

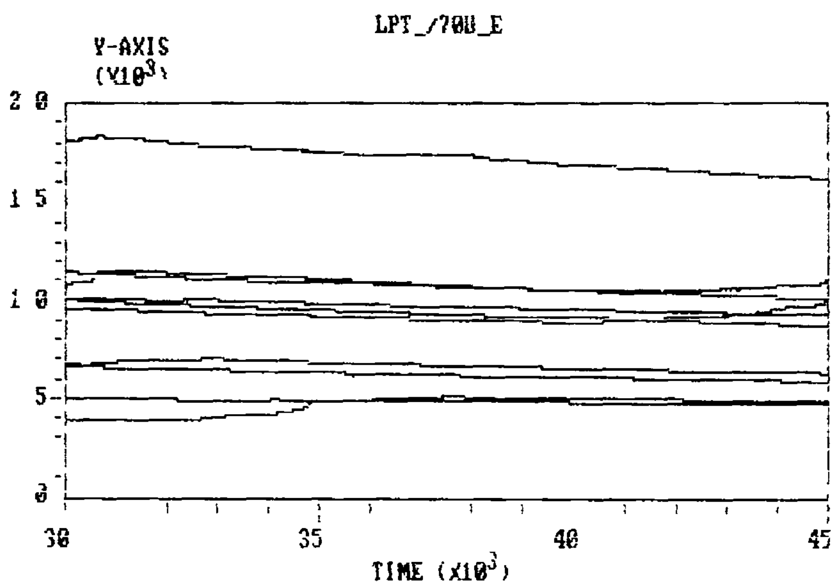


Fig 6.6b: The approx. steady state points (31880) in the σ plots for the LPT rule flow-time variables under Unif. Arrival and Expo. Operation Time and 70% Load.

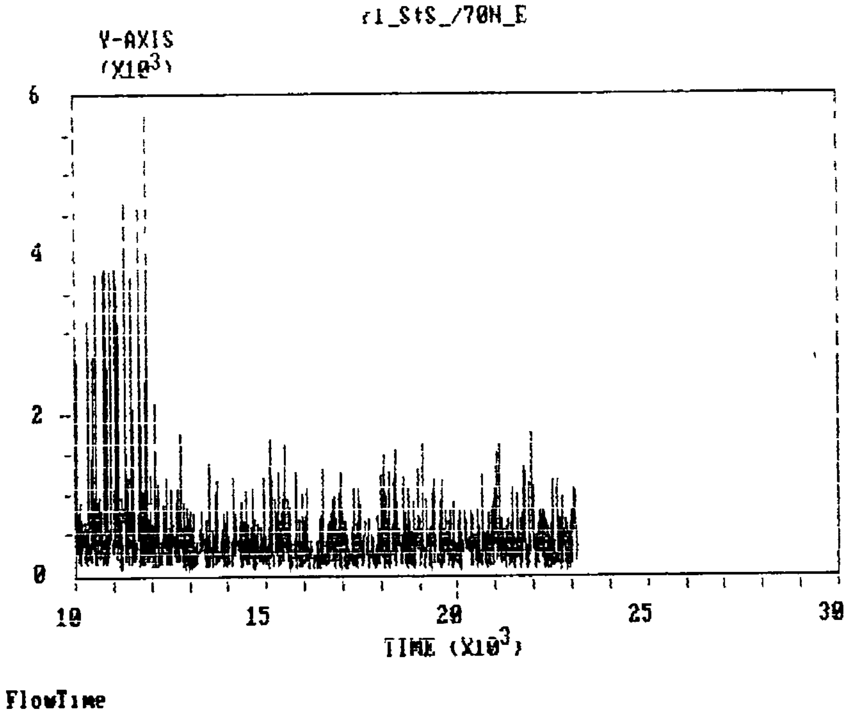


Fig. 6.7a: **The Static-SLK rule flow-time variables plot in the 1^{th} Rep.(run) under Norm. Arrival and Expo. Operation Time and 70% load.**

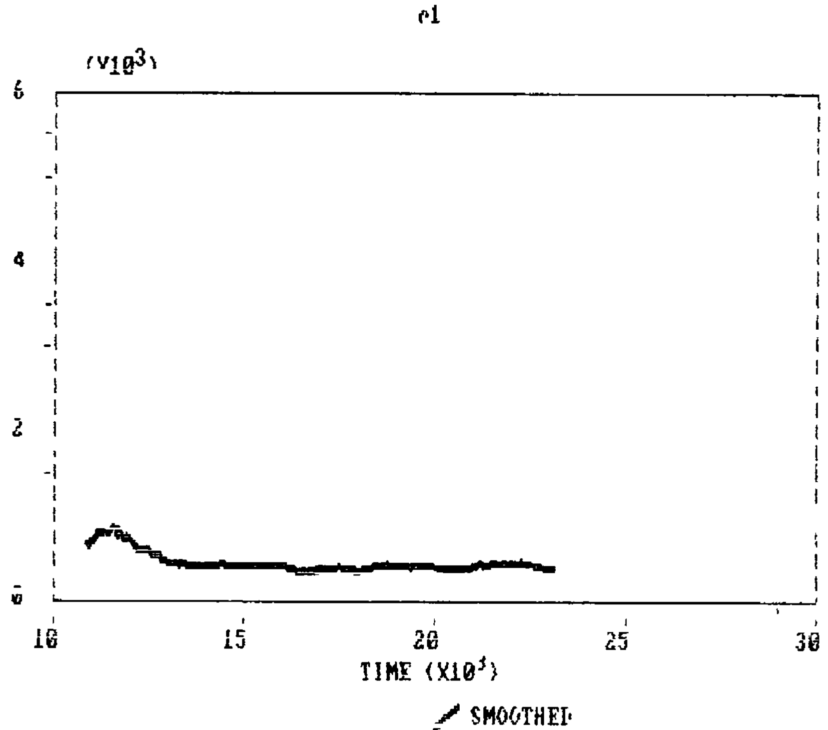


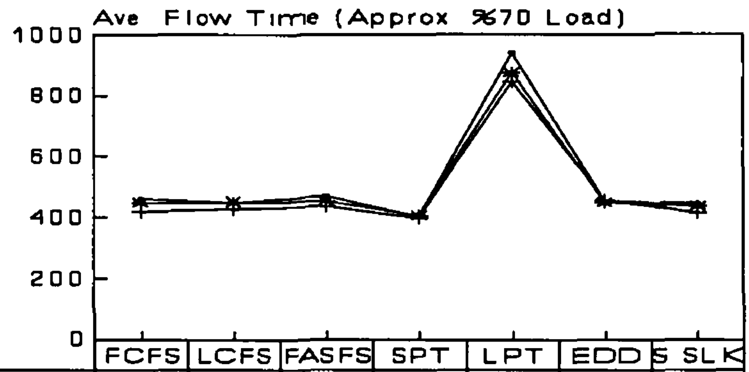
Fig. 6.7b: **The approx. steady-state point (10000) in the moving-average plot for the static-SLK flow-time variables plot in Fig. 6.7a.**

6.6.2 The Effect of the Various Priority Rules on the Proposed Criteria Under Three Types of Arrival Time Distributions and Shop Loads:

As mentioned in item 6 of Section 6.1 (page 110) that there are five selected criteria which will be considered for evaluating the job-shop priority rules.
Hereafter, the following items will discuss the analysis of the experimental simulation results for the chosen criteria each in turn of the first comparison phase:

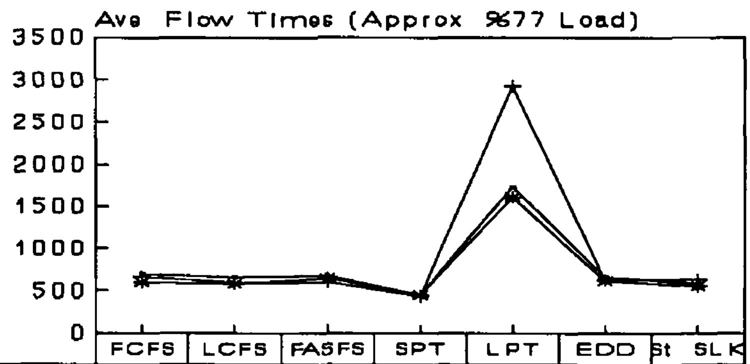
1. Mean Flow-Time: (Figs 6.8a, 6.8b and 6.8c)

The data and their plots in Figs. 6.8(a, b and c) show that the rule which minimize the mean flow-time is the SPT rule in the low, medium and high shop-loads and exponential, uniform and normal job arrival patterns. While the LPT rule has the highest values of the mean flow-time. This results is agreement with the earlier researchers such as Ram, and Schriber[1990][44].
On the other hand, EDD based rule gives the worst performance only when the high shop-load and uniform arrival pattern is applied as shown in Figure 6.8c. Elvers's studied [1973][53] gives a similar result for this rule with respect to similar job arrival pattern. The FCFS, LCFS, FASFS and St. SLK give different performances for each other especially in the low and medium shop-loads and all the proposed job arrival patterns. Also as shown in Figs. 6.8b and 6.8c for the medium and high shop-loads especially in the exponential and uniform arrival patterns that the FASFS tends to be slightly better than FCFS based rule. This result is in agreement to the result had been found by Conway[1965a][52] and Rochette et al[1976][57].
One important conclusion can be drawn from the above results which is that the effect of the different arrival patterns on the priority rules is not a significant variable for a given shop-load. This conclusion is in agreement to Elvers's studies[53].



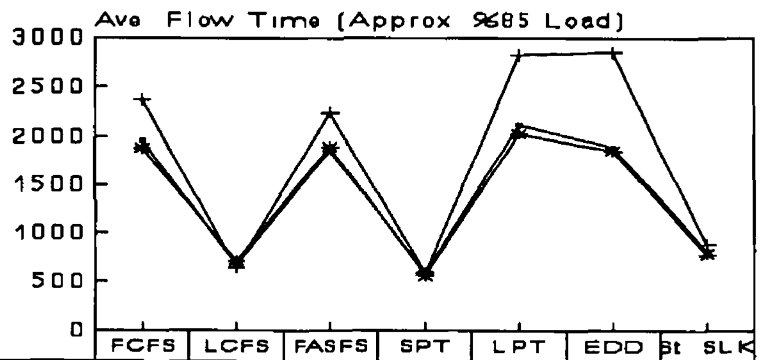
	FCFS	LCFS	FASFS	SPT	LPT	EDD	S SLK
Expo Arr Time Dis —▪—	458	448	469	402	937	450	444
Unif Arr Time Dis —+—	418	425	436	396	843	454	412
Norm Arr Time Dis —✳—	447 9	448 2	451	405	874	448	435

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis —▪—	691	664	675	446	1729	669	580
Unif Arr Time Dis —+—	665	608	599	432	2920	625	634
Norm Arr Time Dis —✳—	600	590	640	448	1610	622	557

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis —▪—	1965	675	1851	593	2123	1893	822
Unif Arr Time Dis —+—	2375	642	2243	600	2820	2846	882
Norm Arr time Dis —✳—	1875	713	1888	561	2031	1853	786

Priority Rules

Fig. 6.8 (a, b, and c). The mean flow-time Vs. the priority rules under different job arrival patterns and shop-sizes.

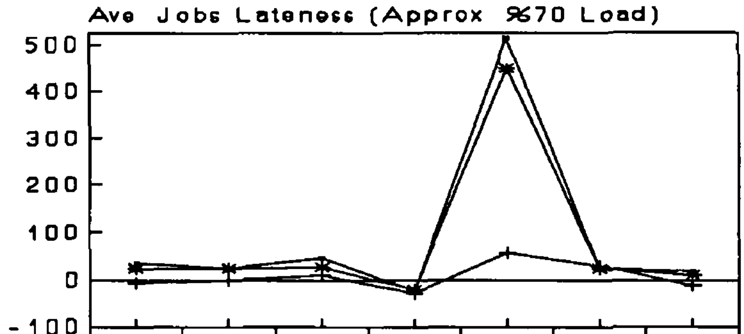
2. Mean Job Lateness: (Fig. 6.9a, 6.9b and 6.9c)

In this context a negative value for lateness means that the job finished early A job is tardy only if lateness is greater than zero
From comparing the performance characteristics of the priority rules, the data and their plots in Figures 6 9a, 6 9b and 6 9c show that

Under the SPT rule there are no lateness of any job The jobs begin to be tardy when the shop-load is increased However, under the medium and high shop-loads of the three arrival patterns and assigning the due date conditions the SPT rule gives better performance than the other with respect to the job lateness While the LPT rule tends to perform poorly at most of the job-shop conditions In the high shop-load and all the proposed arrival patterns the LCFS and St SLK rules give fairly good results respectively with respect to the job lateness While in the low and medium shop-loads and all arrival patterns the FCFS, LCFS, FASFS, EDD and Sts SLK perform quit equal to each other as shown in Figures 6 9b and 6 9c

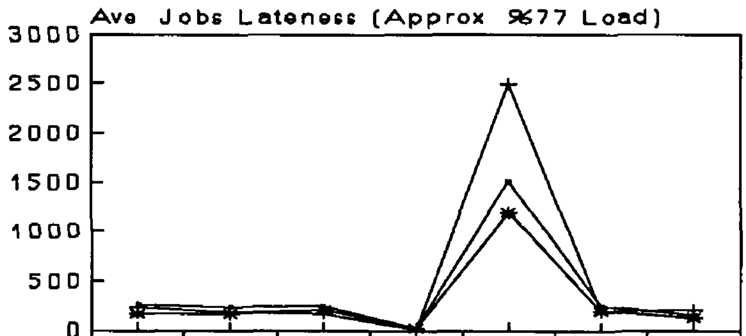
3. The Mean Idle Time: (Figs. 6.10a, 6.10b and 6.10c)

The results drawn in Figure 6 10a show that the LPT, SPT, FASFS and LCFS rules give the worst performance with respect to the machines idle time respectively This result is concluded just when the 70% load and exponential arrival pattern are applied While the LPT rule tend to be the best rule when the normal arrival pattern is applied The other rules perform equally well in all applied arrival patterns
In Figure 6 10b when the 77% load and all arrival patterns are applied the LPT rule gives the worst performance, while the other rules tend to be equal to each other
Also the LPT rule gives the worst performance when the 85% load and all arrival patterns are applied as shown in Figure 6 10c, While the SPT rule tends to be the best rule The other rules perform equally well



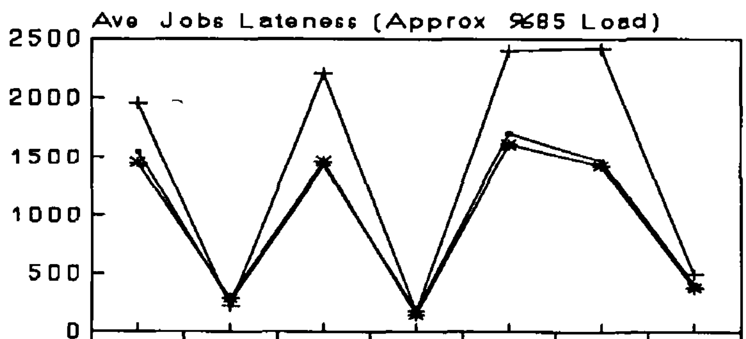
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	33 7	23	45 6	24	513	25 1	19 2
Unif Arr Time Dis	6 4	0 53	10 8	29	57	29 3	-12 9
Norm Arr Time Dis	23 8	23	25 5	20	448	22 8	9 3

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	265	238	250	20 9	1507	244	156
Unif Arr Time Dis	241	184	175	7 67	2494	200	208
Norm Arr Time Dis	175	165	215	22 8	1186	202	133

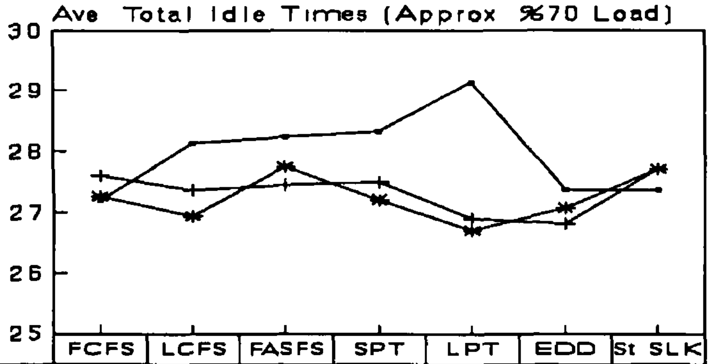
Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	1540	249	1426	176	1697	1468	410
Unif Arr Time Dis	1951	218	2211	176	2395	2421	490
Norm Arr Time Dis	1451	287	1462	138	1606	1428	374

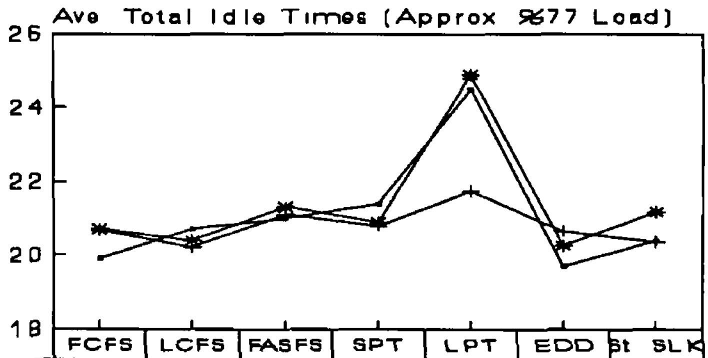
Priority Rules

Fig. 6.9(a, b and c): The job lateness Vs. the priority rules under different arrival patterns and shop loads.



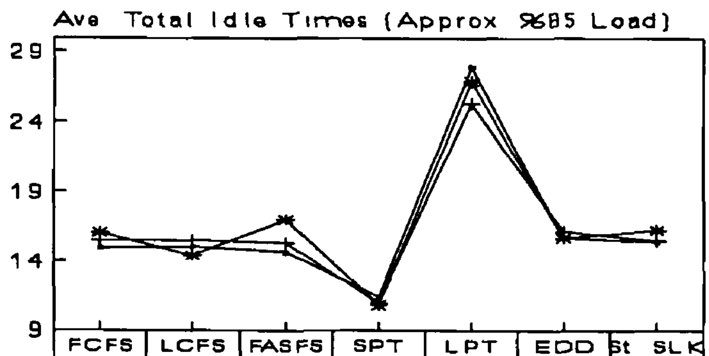
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	27 2	28 13	28 23	28 33	29 13	27 36	27 36
Unif Arr Time Dis	27 6	27 37	27 44	27 5	26 9	26 8	27 7
Norm Arr Time Dis	27 26	26 93	27 74	27 2	26 7	27 07	27 7

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	19 9	20 7	21	21 4	24 5	19 7	20 4
Unif Arr Time Dis	20 67	20 23	21 1	20 8	21 74	20 66	20 36
Norm Arr Time Dis	20 7	20 4	21 3	20 9	24 9	20 26	21 17

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	14 9	15	14 6	11 4	27 9	15 7	15 4
Unif Arr Time Dis	15 5	15 5	15 3	10 9	25 3	16 1	15 5
Norm Arr Time Dis	16	14 4	17	10 8	26 9	15 7	16 2

Priority Rules

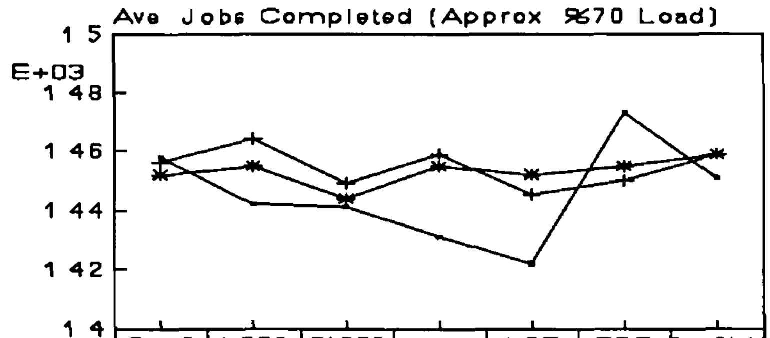
Fig. 6.10(a, b and c): The machines idle time Vs. the priority rules under three arrival patterns and shop loads.

4. The Mean Completed Jobs: (Figs. 6.11a, 6.11b and 6.11c)

In Figures 6 11c when the 85% shop-load and all arrival patterns are applied the SPT rule gives the best performance with respect to jobs completed While the LPT rule tends to be the worst rule
In Figure 6 11(a and b) the EDD rule gives good performance when the 70% and 77% shop-loads and Exponential pattern are applied Figure 6 11a show that in the case of the 70% shop-load and Normal arrival pattern the FSAFS rule tends to gives poor quality-performance than other While the LPT rule gives the worst performance in the other cases of the shop-loads and arrival patterns as shown in Figures 6 11a, 6 11b and 6 11c Also these Figures show that for all other rules (FCFS, LCFS, St SLK) perform equally well in all cases of the shop-loads and arrival patterns not mentioned before

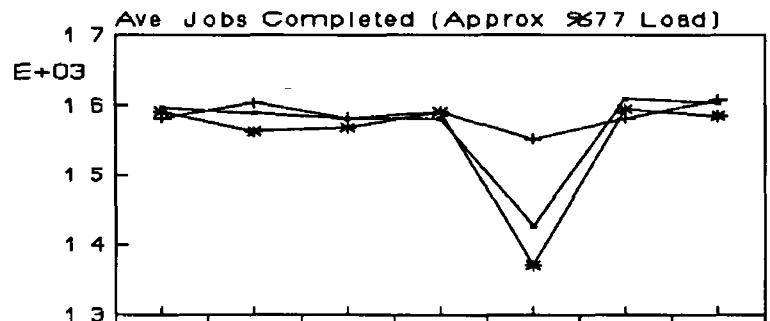
5. The Mean Total WIP (Figs. 6.12a, 6.12b and 6.12c)

In this context of the mean total WIP criterion under six machines groups Figures 6 12(a, b and c) show that the SPT rule is better for every condition tested
While the LPT rule gives the worst performance These results are in agreement with the results had been carried out by Conway[52]
The other rules (FCFS, LCFS, FASFS, EDD and St SLK) perform approximately equal values as shown in Figures 6 12a, 6 12b and 6 12c



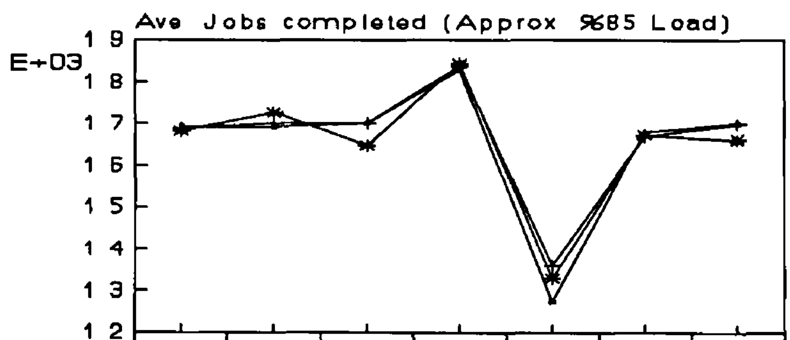
Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	1 458	1 442	1 441	1 431	1 422	1 473	1 451
Unif Arr Time Dis	1 456	1 464	1 449	1 459	1 445	1 45	1 459
Norm Arr Time Dis	1 452	1 455	1 444	1 455	1 452	1 455	1 459

Priority Rules



Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	1 596	1 588	1 581	1 582	1 427	1 609	1 604
Unif Arr Time Dis	1 582	1 603	1 582	1 591	1 552	1 582	1 608
Norm Arr Time Dis	1 59	1 562	1 568	1 59	1 37	1 595	1 585

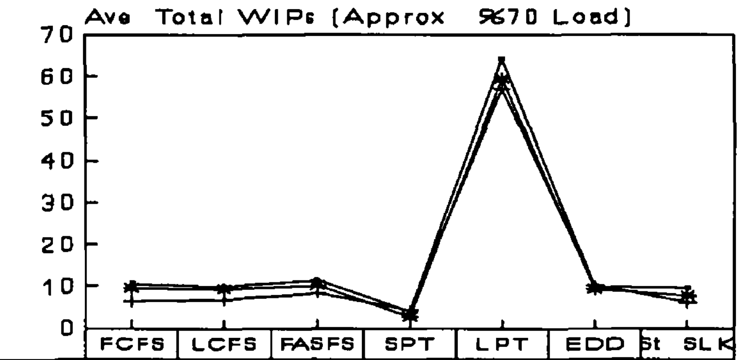
Priority Rules



Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis	1 693	1 691	1 701	1 829	1 274	1 68	1 7
Unif Arr Time Dis	1 688	1 702	1 701	1 837	1 359	1 67	1 699
Norm Arr Time Dis	1 681	1 725	1 647	1 843	1 33	1 674	1 662

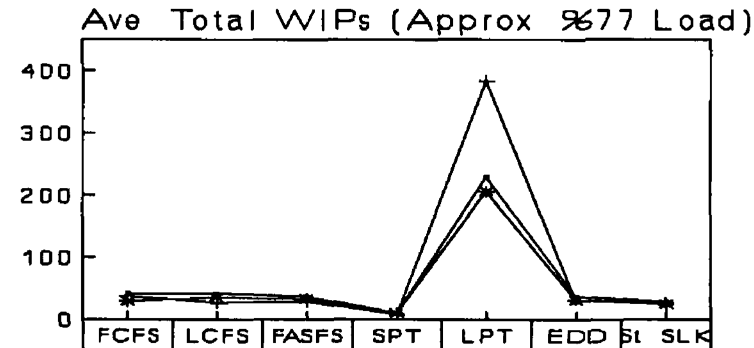
Priority Rules

Fig. 6.11(a, b and c): The mean jobs completed Vs. the priority rules under the three arrival patterns and shop loads.



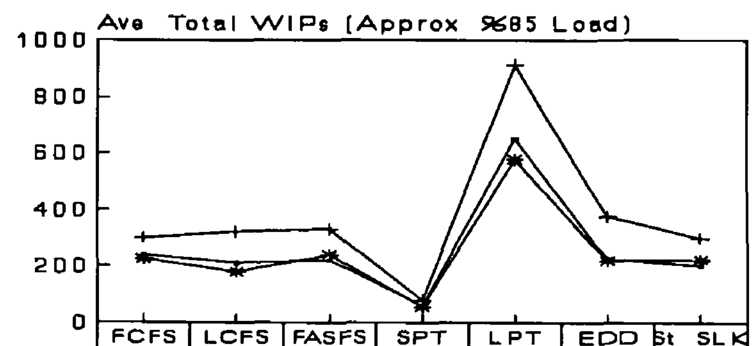
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis —•—	10 6	9 95	11 5	4	64	9 98	9 56
Unif Arr Time Dis —+—	6 48	6 76	8 4	4 14	57	10	6 23
Norm Arr Time Dis —✱—	9 5	9 27	10 1	2 6	59 3	9 29	7 8

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis —•—	42	41 3	38 4	10 2	229	37 2	29 2
Unif Arr Time Dis —+—	37 2	26 7	28 8	8 6	384	32 3	27 7
Norm Arr Time Dis —✱—	28 7	36	34 2	10 8	206	32	25 6

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Arr Time Dis —•—	240	209	220	58	650	224	201
Unif Arr Time Dis —+—	299	320	330	79	914	375	299
Norm Arr Time Dis —✱—	223	177	240	53	577	218	222

Priority Rules

Fig. 6.12(a, b and c): The mean total WIPs Vs. the priority rules under the three arrival patterns and shop loads.

6.6.3 The Effect of the Various Priority Rules on the Proposed Criteria Under Three Types of Operation Time Distributions and Shop Loads:

In the previous section 6 6 2 the evaluation of the six priority rules have been discussed under the three job arrival distributions (Expo Unif and Norm), shop-loads (Approx 70%, 77% and 85%) and a commonly known single pattern of the operation time distribution (exponential pattern) were applied In this section the job-shop simulation has been carried out under a commonly known single pattern of job arrival (exponential pattern), Three operation time distributions (Expo Unif and Norm) and also three shop-loads were applied

Hereafter, the following items will discuss the analysis of the experimental simulation results for the chosen criteria each in turn for the second phase

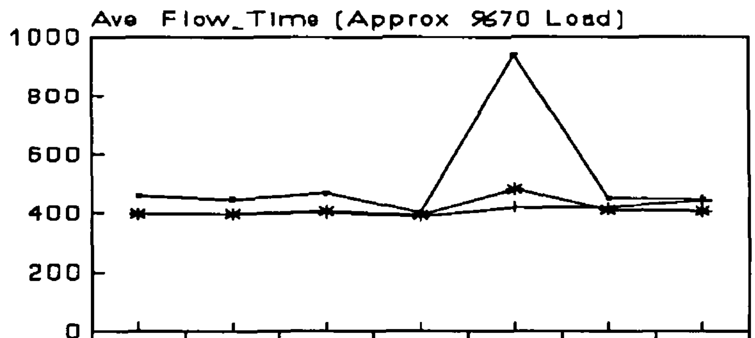
1. Mean Flow-Time (Figs 6 13a, 6.13b and 6 13c)

As shown in Figures 6 13a, 6 13b and (6 13c, only when the exponential operation time pattern is applied) the SPT rule is the best with respect to the mean flow-time This result has been found by the major comparative studies (LeGrande [1963] [47], Conway[1965a][52] and Eilon et al[1968][55])
While the LPT rule gives the worst performance, especially at the low and medium shop-loads (Figs 6 13a and 6 13b) The other rules (LCFS, St SLK, FCFS, EDD and FASFS) perform approximately equal quite well only in the low and medium shop-loads as shown in Figures 6 13a and 6 13b
In the uniform and normal operation time patterns and Approx 85% shop-load (Figure 6 13c) the SPT rule gives less performance with respect to the mean flow-time While the LCFS tends to be the first quality-performance (In general, the LCFS

rule is less important when used in job-shop environment As the author's Knowledge the LCFS rule has not yet been studied by the job-shop environment researchers)
The St SLK gives a good performance in this level of shop-load for all operation time patterns, while the FCFS and EDD rules gives less performance than the St SLK rule

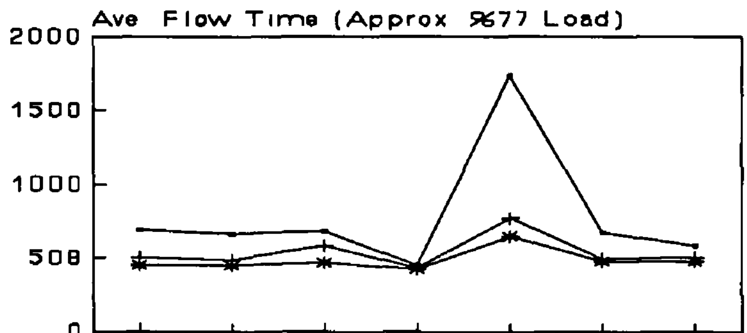
2. Mean Lateness: (Figs. 6 14a, 6.14b and 6.14c)

Since lateness is just the algebraic difference between the completion time and a given due date, due to conway[5] it can be expected that the mean of this criterion will be minimized by the rule in which the mean flow-time is minimized Hence, according to item one of this section and Figures 6 14a, 6 14b and (6 14c, only when the exponential arrival pattern is applied) the SPT rule is the best performance with respect to the mean lateness While the LPT rule gives the worst performance at the same shop conditions When the low and medium shop-loads and all operation time patterns are applied the other rules (FCFS, St SLK, LCFS, FASFS and EDD) give approximately quite equal well as shown in Figures 6 14a and 6 14b
In Figure 6 14c when the high load and uniform and normal operation time patterns are applied, the LCFS gives the best performance than other rules While the SPT rule tends to be the second quality-performance The St SLK ranks the third in the high load and all operation time patterns are applied, While the FCFS, FASFS and EDD respectively decrease in their performance when the high shop-load is applied



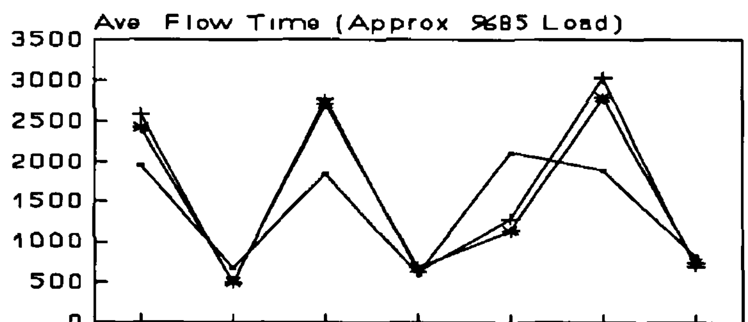
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	458	448	469	402	937	450	444
Unif Op Time Dist	396	397	401	390	421	417	441
Norm Op time Dist	398	397	404	393	482	412	405

Priority rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	691	664	675	446	1729	669	580
Unif Op Time Dist	499 1	485	581	435	764	495	498 5
Norm Op Time Dist	447	452	468	425	645	473	478

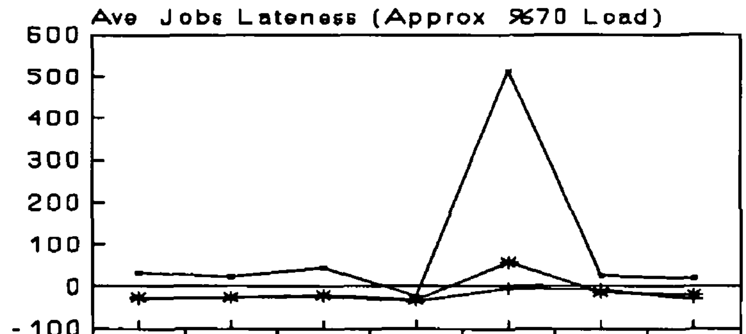
Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	1965	675	1851	593	2123	1893	822
unof Op Time Dist	2583	485	2771	617	1287	3034	693
Norm Op Time Dist	2426	496	2718	678	1132	2786	741

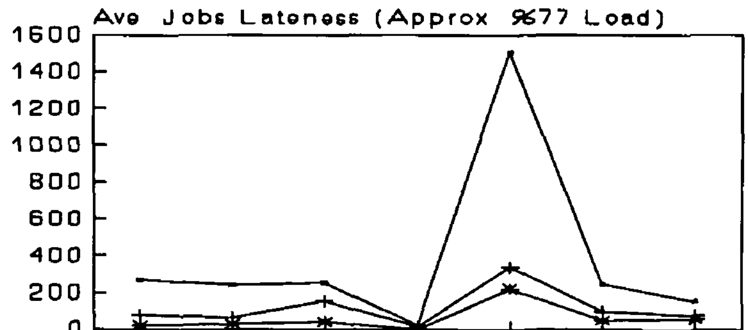
Priority Rules

Fig. 6.13(a, b and c): The mean flow-time Vs. the priority rules under the three operation time patterns and shop loads.



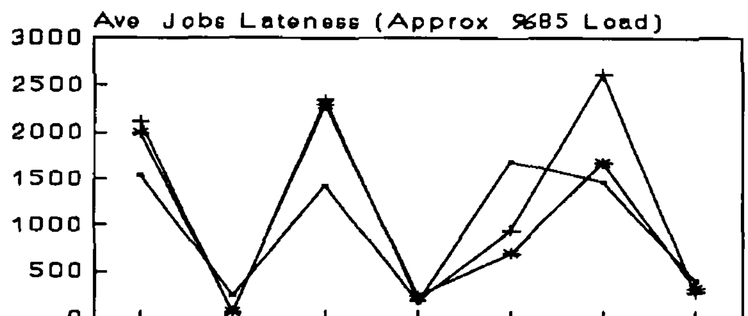
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	33 7	23	46	24	513	25 1	19 2
Unif Op Time Dist	28 6	27 2	23 6	34 2	4	7 3	28 3
Norm Op Time Dist	26	26 6	21	31 8	57	13 3	19 4

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	265	238	250	20 9	1507	244	156
Unif Op Time Dist	74 4	59 4	156	10 5	339	96 3	74 7
Norm Op Time Dist	21 6	27 3	43 3	0 3	221	48	53

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	1540	249	1426	176	1697	1468	410
Unif Op Time Dist	2121	60	2346	195	952	2610	282
Norm Op Time Dist	2001	70	2292	247	707	1679	328

Priority Rules

Fig. 6.14(a, b and c): The mean lateness Vs. the priority rules under the three operation time patterns and shop loads.

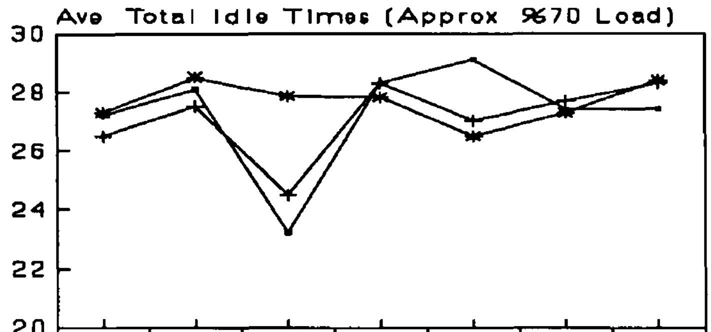
3. Mean Total Idle Times%: (Figs. 6 15a, 6.15b and 6.15c)

The data in Figures 6 15a and 6 15b when the low and medium shop-loads and three operation patterns are applied show the following results

1 The FASFS rule is the best rule when Exponential and uniform operation time patterns are applied, while the LPT rule tends to give the worst performance The other rules (SPT, FCFS, LCFS, St SLK and EDD give an approximately equal well performance in the same shop conditions

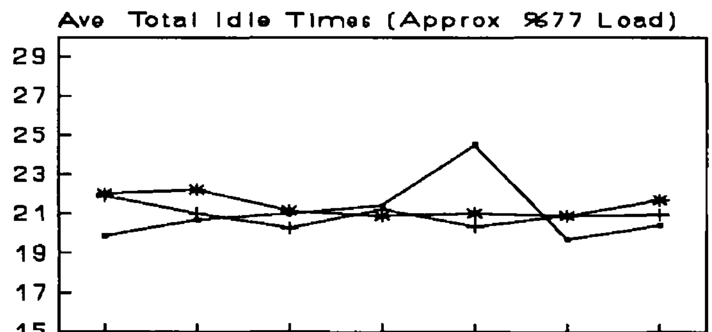
2 When the normal operation time pattern is applied, the LPT rule gives a good performance than the others While the St SLK rule gives less performance than the others The other rules give an approximately equal well performance with respect to the Mean total idle time

But Figure 6 15c shows that when the high shop-load (approx 85%) and the three operation time patterns are applied the SPT rule is the dominate rule over all other rules tested In contrast, the LPT rule gives the worst performance The other rules gives equal well to each other in the same shop conditions



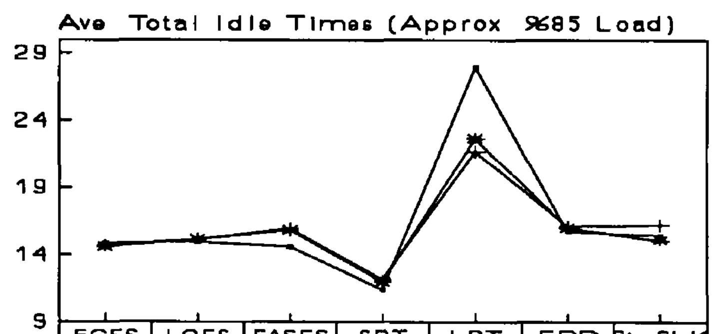
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	27 2	28 1	23 2	28 3	29 1	27 4	27 4
Unif Op Time Dist	26 5	27 5	24 5	28 3	27	27 7	28 3
Norm Op Time Dist	27 27	28 5	27 84	27 81	26 5	27 3	28 4

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	19 9	20 7	21	21 4	24 5	19 7	20 4
Unif Op Time Dist	21 9	21	20 26	21 2	20 37	20 9	20 93
Norm Op Time Dist	22	22 2	21 14	20 9	21	20 9	21 7

Priority Rules



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	14 9	15	14 6	11 4	28	15 7	15 4
Unif Op Time Dist	14 7	15 2	15 9	12 13	21 7	16 1	16 2
Norm Op Time Dist	14 6	15 2	15 8	12	22 7	16	15 1

Priority Rules

Fig. 6.15(a, b and c): The mean idle times Vs. the priority rules under the operation time patterns and shop loads.

4. The Mean Jobs Completed: (Figs. 6.16a, 6.16b and 6.16c)

In Figure 6.16a, when the shop-load is low (Apprix. 70%) and the three operation time patterns are applied, all rules tend to be approximately quite equal. But in figure 6.16b, when the shop-load is medium (Approx. 77%) and the exponential operation times are applied the LPT rule gives poor performance. While all the other rules give approximately good results with respect to mean jobs completed.

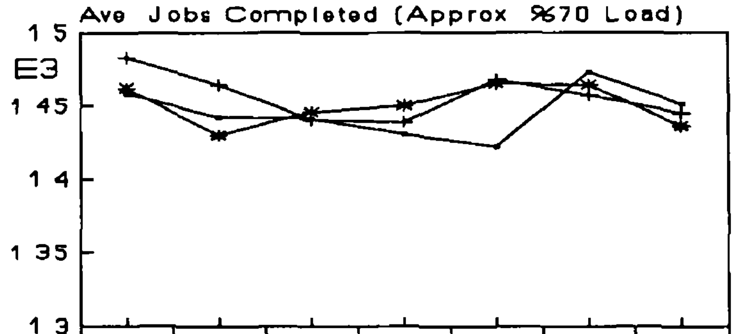
As shown in Figure 6.16c when the high shop-load (Approx. 85%) and the three operation patterns are applied, the SPT rule is the dominate rule over all the others. This result is according to most researcher's studies such as Jones[42]. While the same Figure 6.16c shows that the LPT rule gives the very worst performance especially when the exponential operation time pattern is applied. The other rules (FCFS, LCFS, FASFS, Static. SLK and EDD) respectively at the same conditions give fairly good results.

5. Mean Total WIP: (Figs. 6.17a, 6.17b and 6.17c)

Figures 6.17a, 6.17b and 6.17c show that the SPT rule tends to have its greatest advantage than the other rules with respect to the mean total WIPs. While the LPT gives the poor performance. This results tend to be more apparent when the high shop-load (Approx. 85%) and the three operation time patterns are applied. Most researchers such as Conway[52] and LeGrande[47] have reported similar results in their research.

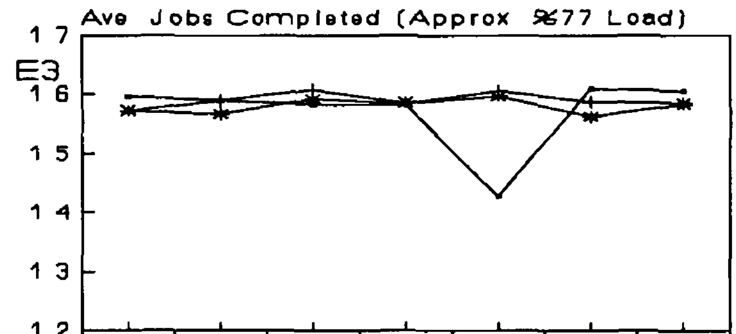
It is important to note that when the exponential operation pattern and high shop-load (approx. 85%) are applied all rules performance tend to be better than at the low or medium shop-loads with respect to mean WIP.

In Figures 6.17a, 6.17b and 6.17c the other rules (FCFS, LCFS, FAFSF, St. SLK and EDD) give approximately similar performances especially when all the operation time patterns and only medium and high shop-loads are applied.



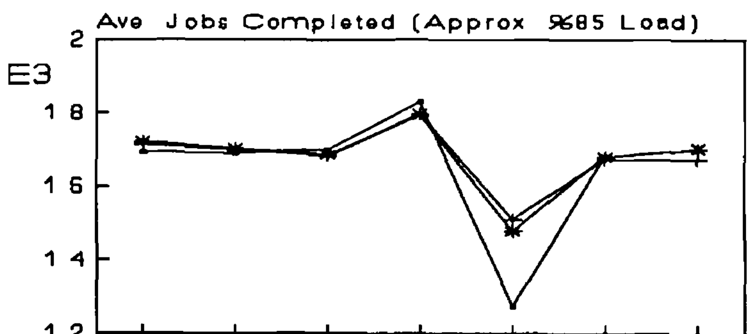
		FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	—▪—	1 458	1 442	1 441	1 431	1 422	1 473	1 451
Unif Op Time Dist	—+—	1 483	1 464	1 44	1 439	1 468	1 457	1 445
Norm Op Time Dist	—*—	1 462	1 43	1 446	1 45	1 465	1 464	1 436

Priority Rules



		FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	—▪—	1 596	1 588	1 581	1 582	1 427	1 609	1 604
Unif Op Time Dist	—+—	1 572	1 589	1 607	1 585	1 684	1 587	1 584
Norm Op Time Dist	—*—	1 572	1 567	1 591	1 585	1 596	1 562	1 583

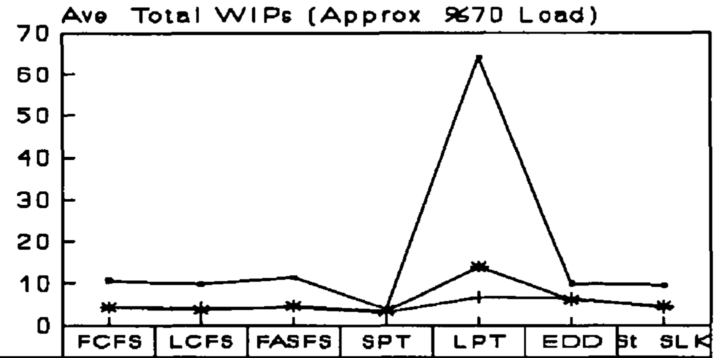
Priority Rules



		FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	—▪—	1 693	1 691	1 701	1 829	1 274	1 68	1 7
Unif Op Time Dist	—+—	1 715	1 703	1 682	1 798	1 511	1 672	1 672
Norm Op Time Dist	—*—	1 722	1 704	1 688	1 795	1 479	1 679	1 703

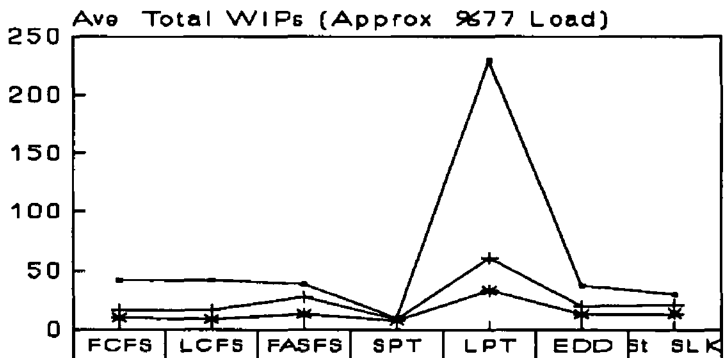
Priority Rules

Fig. 6.16(a, b and c): The mean jobs completed Vs. the priority rules under the three operation time patterns and shop loads.



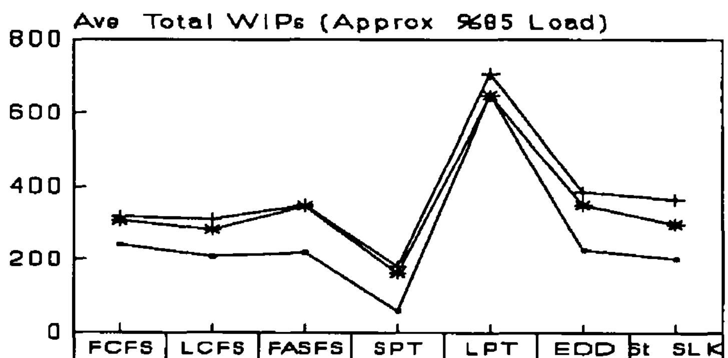
Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	10 6	9 95	11 5	4	64	9 98	9 6
Unif Op Time Dist	4 3	4 3	4 4	3	6 8	6 4	4 2
Norm Op Time Dist	4 3	3 7	4 7	3 5	14	6 1	4 6

Priority Rules



Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	41 9	41 3	38 4	10 2	229	37 2	29 2
Unif Op Time Dist	16 6	16 1	27	9 2	60 4	19 4	21 05
Norm Op Time Dist	10	8 8	13 2	7 7	32 5	12 6	13 6

Priority Rules



Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St SLK
Expo Op Time Dist	240	209	220	58	650	224	201
Unif Op Time Dist	319	310	351	179	707	385	364
Norm Op Time Dist	308	283	345	163	647	350	295

Priority Rules

Fig. 6.17(a, b and c): The mean total WIPs Vs. the priority rules under the three operation time patterns and shop-load.

6.7 The Experimental design of the Simulation Observations Under Study:

The results of the simulation experiment were then into an Analysis Of VAriance (ANOVA) technique This technique appears suitable for examining a number of factors significantly in its effect on the outcome of an experiment Various levels of factors can be tested in order to determine whether effects are consistent throughout the range of variation Since several factors are being varied simultaneously, information can be obtained concerning the various interactions as well as the main effects

Seven replications of the job-shop simulation experiments have been achieved by the different randomization procedure In view of the wide range of possible variability, replication provides a greater assurance that significant testes are not confounded with a large experimental error

In the present simulation the hypothesis to be tested are whether there is a significant difference among the proposed priority rules with respect to the proposed criteria under the following individual conditions studied

1 The three Exponential job arrival distribution workloads (approx 70%, 77% and 85% loads)
2 The three operation time patterns (Exponential, Uniform and Normal) and 85% workload of the exponential job arrival pattern is applied

Hereafter, The ANOVA computations using the IBM software for the above two conditions are presented in Tables 6 4 and 6 5 with respect to the five response variables (mean flow-time, lateness, total Idle times, jobs completed and total work-in processes This outcomes will discuss in turn as follows

1. The Two factorials ANOVA of the seven priority rules and three exponential job arrival patterns workloads:

Table 6 4 summarizes the ANOVA for the response variables (The seven criteria measured) Since the calculated for all "F_o" values (MS_{factor}/MS_{error}) (see the sixth column in table 6 4) of the response variables are greater than the critical $F_{05\ v1\ v2}$ (see the seventh column in Table 6 4 and Appendix F) values at the 0 05 confidence level for the following factors and interactions

- The proposed priority rules (7 levels)
- The exponential job arrival pattern workloads(3 levels)
- The interaction between the above two factors

These factors and its interactions can be said to have a significant effect on the output of the job-shop with respect to the response criteria, especially on the different arrival load factors

2. The two factorials ANOVA of the seven priority rules and the three operation time distributions:

Table 6 5 summarizes the ANOVA for the response variables (The seven criteria measured) Since the calculated for "F_o = 0 685" value (MS_{factor}/MS_{error}) of the Mean Total Idle Times criterion is less than the critical $F_{05\ v1\ v2}$ values = $F_{05\ 2\ 126}$ = 3 at the 0 05 confidence level (see the seventh column in Table 6 5 and Appendix F) for the three operation time factor These factor can be said to have not a significant effect on the output of the job-shop system with just respect to the WIP criterion

Also, since the calculated for "F_o = 1 35" value (MS_{factor}/MS_{error}) of the Mean Total WIPs criterion is less than the critical $F_{05\ v1\ v2}$ values = $F_{05\ 12\ 126}$ = 1 75 at the 0 05 confidence level for the interaction of the priority rules and the three operation time factors

These factors can be said to have not a significant effect on the output of the job-shop system with just respect to the WIP criterion

But, for the other "F_o" values (see the sixth column of table 6 5 and Appendix F) of the other factors, since they are greater than The critical $F_{05\ v1\ v2}$ values (see the seventh column of Table 17) at the selected 0 05 confidence level for the following factors and interactions

- The proposed priority rules (7 levels)
- The three operation time patterns under %85 workload (3 levels)
- The interaction between the above two factors

These factors and its interactions can be said to have a significant effect on the output of the job-shop with respect to the response criteria, especially on the different priority rules

Source of Variation	VAR	Degree of Freedom	Sum of Squares	Mean of Square	Fo Value	$F_{05\ 6\ 126}$ $F_{05\ 2\ 126}$ $F_{05\ 12\ 126}$ Values
The Priority Rules	FT	6	20398020	3399670	50 4	2 1
	IT	6	716 98	119 3	25 2	2 1
	La	6	19447790	3241298	43 2	2 1
	JD	6	831456	1064864	39 8	2 1
	WIP	6	1083083	180514	120	2 1
The Expo Arrival Patterns Loads	FL	2	20754500	10377250	154	3
	IT	2	3300 13	16501	34 9	3
	La	2	21956130	10978065	146	3
	JD	2	1064864	532432	152 8	3
	WIP	2	1602932	801466	531	3
Inter_ action	FL	12	11267230	938936	50 4	1 75
	IT	12	577 75	48 15	10 18	1 75
	La	12	10440410	870034 3	11 6	1 75
	JD	12	632128	52677 33	15 12	1 75
	WIP	12	5891083	49092 35	32 52	1 75
Error	FT	126	8491216	67390		
	IT	126	596 07	4 73		
	La	126	9451336	75010		
	JD	126	439104	3484 95		
	WIP	126	190210	1509 61		
Total	FT	146	60910960			
	IT	146	5190 92			
	La	146	61295670			
	JD	146	2967552			
	WIP	146	4365335			
Percent of Criteria Variabi_ lity	FT	%86				
	IT	%89				
	La	%85				
	JD	%85				
	WIP	%95				

Notations

FT is the Flow Time
IT is the Idle Time
La is the job Lateness
JD is the jobs completed
WIP is the Work_In_Process
VAR Is the response variables

Table 6.4: ANOVA for simulation data of the response variables under the two factorials of the priority rules and Job arrival pattern workloads.

Source of Variation	VAR	Degree of Freedom	Sum of Squares	Mean of Square	Fo Value	$F_{05\ 6\ 126}$ $F_{05\ 2\ 126}$ $F_{05\ 12\ 126}$ Values
The Priority Rules	FL	6	99003680	16500613	156 5	2 1
	IT	6	2065 66	342 8	130 8	2 1
	La	6	98549600	16424933	111 9	2 1
	JD	6	2009504	334917 3	112 2	2 1
	WIP	6	3340514	556752 3	108 3	2 1
The Operation Time Patterns	FL	2	1531328	765664	7 26	3
	IT	2	3 59	1 8	0 685	3
	La	2	1533712	766856	5 22	3
	JD	2	15744	7872	2 64	3
	WIP	2	402076	201038	39 2	3
Inter_action	FL	12	13039580	1086632	10 3	1 75
	IT	12	117 46	9 79	3 73	1 75
	La	12	13036500	1086375	7 4	1 75
	JD	12	169504	14125	4 73	1 75
	WIP	12	83048	6920 67	1 35	1 75
Error	FL	126	13289250	105470 2		
	IT	126	330 34	2 62		
	La	126	18498110	146810 4		
	JD	126	376192	2985,65		
	WIP	126	645982	5126 84		
Total	FL	146	126863800			
	IT	146	2508 34			
	La	146	13161790			
	JD	146	2570944			
	WIP	146	4471620			
Percent of Criteria Variabi_lity	FL	%90				
	IT	%87				
	La	%86				
	JD	%85				
	WIP	%86				

Table 6.5: ANOVA for simulation data of the response variables under the two factorials of the priority rules and operation time patterns.

CHAPTER SEVEN

7. A Scheduling of Automated Job-Shop or Flexible Manufacturing System:

7.1 Introduction:

This Chapter is mainly concerned with the application of the job sequencing of the priority rules to the AGV in flexible manufacturing system The well-known priority rules will be evaluated with respect to various criteria Also section (7 3 5) in this Chapter will discuss the effect of the number of AGVs on the FMS's Multi-criteria when using AGV's different speeds

In scheduling FMS, Due to a high variety of workpieces, automated transporter system and operating under computer control, the performance of scheduling priority rules depends very heavily on the criterion chosen as well as on the configuration of the production system in hand
In general, the scheduling FMS has greater flexibility to rush order and special customer requests especially when the random-orders are applied These properties are truly if the following main items are used in a FMS

1 Reprogrammable machines with automated tool or hand changing
2 Machine buffers
3 An automated material (workpieces) handling system
4 A sufficient number of tools to be a shift
5 The priority rules for each job are described in a part programme

Hereafter, the following section (7 2) will discuss the main types of facilities (activities) which may be included in the FMS

7.2 Elements of Flexible Manufacturing System (FMS):

There are four basic elements of an FMS

7.2.1 Processing Stations:

Most work-stations which they may be included in FMS are typically computer numerical control (CNC) machines (machining centres), inspection and QC stations, assembly operation areas sheet metal processing machines and forging stations

7.2.2 Load/Unload Stations:

The stations in which the workpieces have to be introduced or departs respectively are called load/unload stations The workpieces are fixed and placed on pallets by human operators or robots Usually load/unload stations are located at the same station

7.2.3 An Automated Material Handling System:

This system should perform the following four functions

1 Transports variety of workpieces and subassemblies between the processing stations or processing station buffers and load/unload stations

2 Allows the workpieces to move from any one processing station in the FMS to any another processing station

3 Achieves different job sequences on the various processing stations in the FMS, and to take substitutions when certain processing station are busy.

4 It is connected with the computer control network to

direct it to the different facilities.

Three common types of equipment that have been used to transfer workpieces between stations in an FMS include:

1. Conveyor systems (roller or cart-on-track conveyors).
2. Vehicles (Rail-cars or Automated guided vehicles), later on, the FMS simulation under study will deals with the AGVs as a tool for transporting.
3. Industrial Robots, later on the FMS simulation under study will deals with the stationary robots as a tool to handel the workpieces to and from the buffer machine stations.

7.2.4 Buffer Storage at Work-Stations:

In most FMSs, the buffer storage work-stations are used to serve the important function of providing load/unload buffers to the machines and are thus a form of in-process storage.
These buffers normally have two pallet-stand (machine shuttle) in front of each machine. One provides a queuing position for process waiting to go on to the machine, and the other a queuing position for transport waiting to be taken away from it. The main objective of machine shuttle is to maximize machine utilization.

7.3 A Simulation Study for Evaluating The FMS Priority Rules Vs. The Multiple Criteria:

The FMSs are quite expensive and efforts must be made to obtain low investment risk.
Simulation is found to be a very effective tool in design, implementation, operation and job sequencing of the dynamic interactions in the FMSs.
The following section will present the FMS case study. This study will deal with the evaluating of the various priority

rules with respect to different criteria under low, medium and high shop-loads (approx 70%, 77% and 85% respectively)

7.3.1 The System Elements Description:

The proposed case study analysis, the FMS which has the following elements

- One load/unload station with handling time according to the triangular (1,2,3) distribution
- Three different automatic work-stations (Fig 7 1) are used to process 3 workpieces (Fig 7 2a, b and c)
- One automatic co-ordinate measuring machine (Fig 7 1)
- Two AGV carrier pallets with 100 ft/minute, two circular loops with a common centre track, two spurs for entering and exiting workpieces and one AGV charge and staging area are provided The length of all the AGV's track (The spur length + processing track length + stage area loop + one common centre track) is equal to 810 ft as shown in Appendix J
- Four buffer work-stations (i e , each machine is served by a two-position pallet exchange mechanism to change the palletised components between the AGV and the work-station)
- Four stationary robots are located at each work-station for pick-up the work-pieces between the workstation's pallet mechanism and AGV

Machine No	Machine Type
1	CNC-Lathe m/c
2	Horizontal Machining
3	co_ordinate Measuring m/c
4	CNC_Milling m/c

Fig. 7.1: Planning sheet for a number automatic machines to be used for processing three work-pieces in FMS.

Operation Sequence	Machine Type	Exponential Operation Time Distribu
		Mean
1	(1) Turning	40
2	(4) Milling	30
3	(3) Measur	20

Fig. 7.2a: A Job sequencing sheet for the work-piece 1 in the FMS under Uniform (250,400) distribution due date.

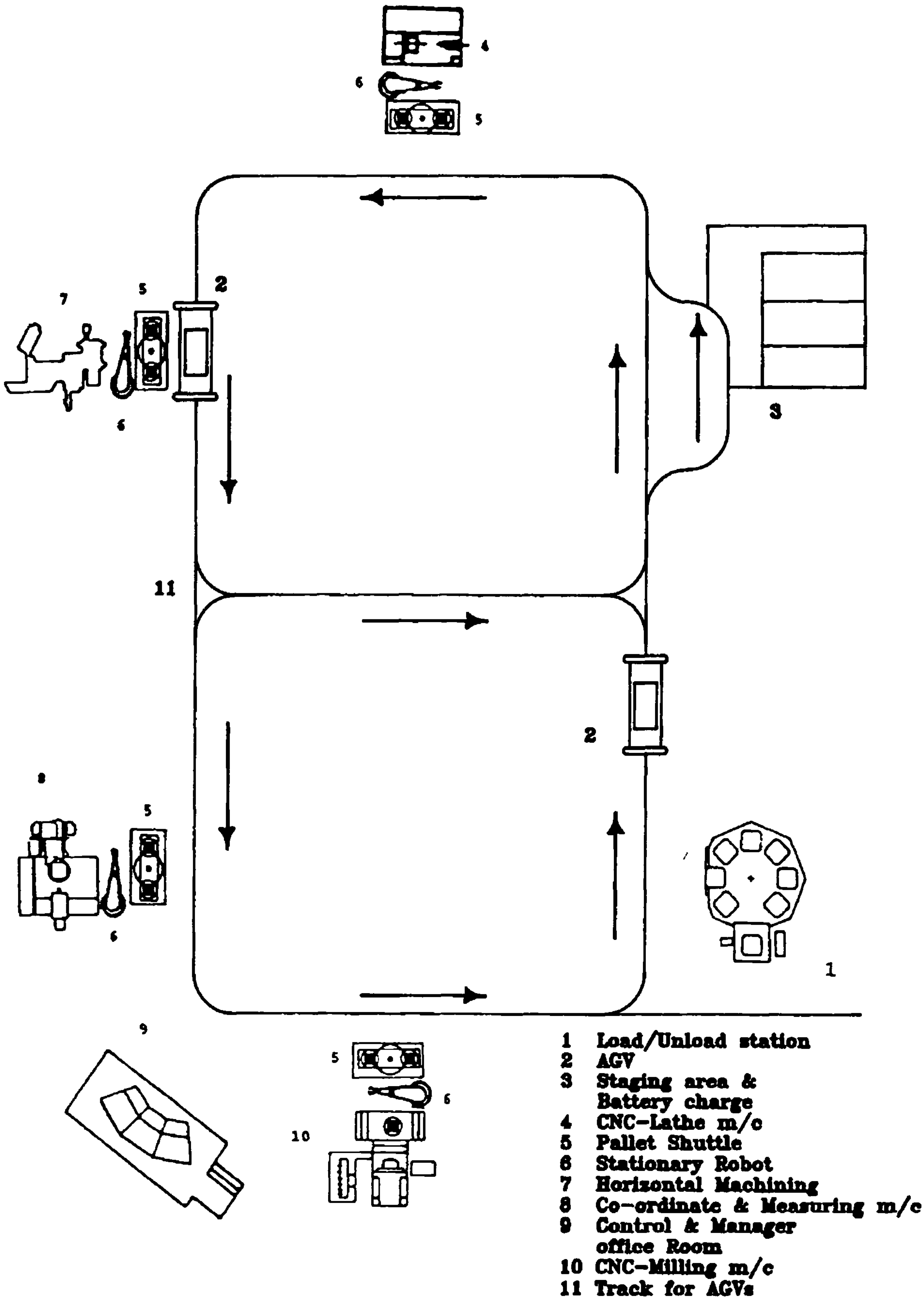
Operation Sequence	Machine Type	Exponential Operation Time Distribu
		Mean
1	(4)Milling	26
2	(2)Horizen	30
3	(1)Turning	25
4	(3)Measur	17

Fig. 7.2b: A job sequencing sheet for the work-piece 2 in the FMS under uniform (250,400) distribution due date.

Operation Sequence	Machine Type	Exponential Operation Time Distribu
		Mean
1	(2)Horizen	33
2	(4)Milling	25
3	(3)Measur	18

Fig. 7.2c: A job sequencing sheet for the work-piece 3 in the FMS under uniform (250,400) distribution due date.

Figure 7 3 below illustrates the FMS lay-out for the proposed system



7.3.2 The Performance Plane for the FMS Case Study:

In this FMS, the case study will be discussed under the following factors are considered important

1 The ratio of the three work-pieces distribution numbers were selected as follows (25% for work-piece 1, 40% for work-piece 2 and 35% for work-piece 3) (Fig 7 2a, 7 2b and 7 2c) These work-pieces enter the system according to the exponential job arrival distribution with three different means (32, 29 and 25), these means are suitable to the low, medium and high shop loads respectively as shown in Fig 7.4

		The FMS loads
The Proposed three different job arrival loads	1 Exponential (32) 2 Exponential (29) 3 Exponential (25)	Approx 70% (low load) Approx 77% (medium load) Approx 85% (high load)

Fig. 7.4: The proposed exponential job arrival distribution with three different FMS loads.

2 The work flow pattern will be multidirectional processes between the work-stations and skipping is allowed, e g , such a processing below is possible

3 → 5 → 1 → 4 → 2 → 1 → 5

3 Processing time distribution is exponential with different means as shown in Fig 7 2(a, b and c)

4 Due date distribution for the expected lead time of the

jobs is uniform with minimum 250 and max 400 minute

5 The seven well-known priority rules were selected as follows

1 First Come First Service rule (FCFS)
2 Last Come First Service rule (LCFS)
3 Shortest Processing Time rule (SPT)
4 Longest Processing Time rule (LPT)
5 First Arrived at Shop First Service rule (FASFS)
6 Earliest Due Date rule (EDD)
7 Static Slack rule (Static-SLK)

6 The seven important system performance criteria were selected for the priority rules evaluation are as follows

1 Minimize the mean flow time
2 Minimize the job lateness
3 Minimize total machine idle times
4 Minimize total work-in-process for machine queues
5 Minimize total waiting for pick up (AGV input queues + in-process queues)

6 Maximize the completion jobs or production rate
7 Minimize total AGV's idle times

7 The FMS will be simulated under three levels of loading such as follows

- The low levels at 70%
- The medium levels at 77%
- The high levels at 85%

8 22 production days for simulation replication with overall production time is equal to 10560 minutes (480 minutes per day) are applied

9 30 minuets break for each 8 hours.

10 The mean time between cleaning and re-lubrication for the

machines, is expressed as Lognormal distribution with mean 15 minutes and a standard deviation of 3 minuets [62]

As discussed in Chapter 6 that, the orders must be scheduled with allowance for in-process waiting, loading queue and the AGV's idle times as well Also they must be dispatched in such away that the schedules will be met This affects the accurate of the scheduling procedure (the priority rule selections) and hence the entire FMS production cost and reasonable AGV's idle times

For simulating the FMS case study which mentioned above, the SIMAN simulation language will be used for evaluating the following two comparison phases

The first comparison phase:

The effectiveness of the performance of the proposed priority rules with respect to the various criteria under the following conditions

- 7 priority rules were simulated under
- 1 job arrival distribution (exponential) (Fig 7 1(a, b and c),
- 3 FMS shop loads (approx 70%, 77% and 85%),
- 1 type of processing time distribution (exponential) (Fig 7 4),
- 2 AGV are used for transporting, 100 ft/minuets of the AGV speed,
- 7 replications with different stream of the random numbers and
- 147 simulation runs (7 different loads x 7 priority rules x 7 replications for each load) were executed for this comparison phase

The second comparison phase:

The effectiveness of the number of AGVs on the FMS's multi-criteria when using AGVs's different speeds under the following conditions

- The SPT rule will be the priority rule from which the job will be processed on the machines
- 1 type of job arrival distribution with mean 25 (approx 85% FMS load),
- 1 type of processing time distribution (exponential) as shown in Fig 7 2a, b and c),
- 1, 2, 3 and 4 the number of AGV will be separately used for simulating with
- 3 different AGV speed (60, 100, 140 and 180 ft/minute and
- 7 replications with different stream of random numbers
- 83 simulation runs (3 AGV x 4 different speeds x 7 replications for each case) were executed for this comparison phase

The Model and Experiment frames of the SIMAN's simulation production scheduling of the proposed FMS case study are stated in Appendix (I and J)
Also as discussed in sections (6 5) and (6 5 1) in the last Chapter that the selected procedure and steady-state point for estimating the mean and variance of random variables (observations) for the proposed criteria are the replication procedure and visual determination respectively

7.3.3 The Experimental Analysis for the FMS Simulation Results:

The proposed FMS's AGV also was analyzed when the steady state conditions were reached and the transient portion was eliminated from the simulation runs

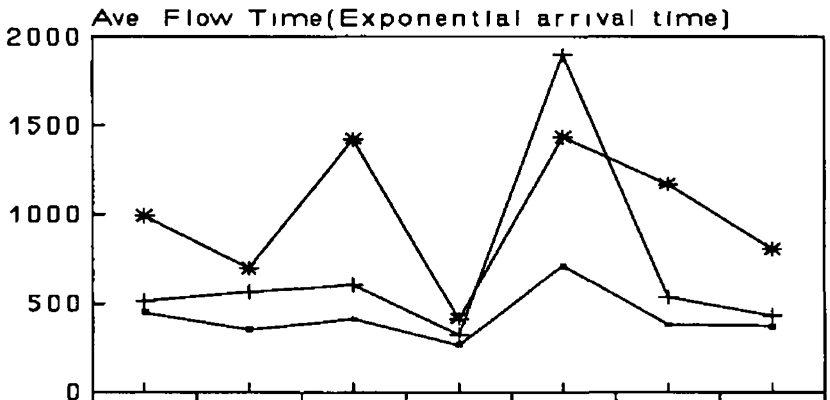
A replication procedure for estimating the length of the transient portion for each case of the two comparison phases was used as was reported in section 6 5 and 6 5 1
Also it was hoped that a single starting point of the steady state could be used for all the seven priority rules

The following two sections will discuss the output data of the experimental results for the two FMS comparison phases respectively

7.3.4 Effectiveness of the Various Priority rules on the proposed Criteria Under the Three Exponential Job Arrival Pattern Loads:

This discussion will deal with the FMS conditions in the first comparison phase (Section 7 3 2), where the exponential job arrival pattern is applied under three FMS loads and 2 AGVs with 100 ft/minute are used
Hereafter, the following items will discuss the analysis of the experimental simulation results for the chosen criteria which mentioned in Item 6 of Section 7 3 2

1. The Mean Flow-Time: (Fig. 7.5)



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load ——	449	354	410	265	709	378	370
Approx %77 load —+—	512	564	602	321	1895	535	431
Approx %85 load —*—	991	698	1420	411	1433	1170	808

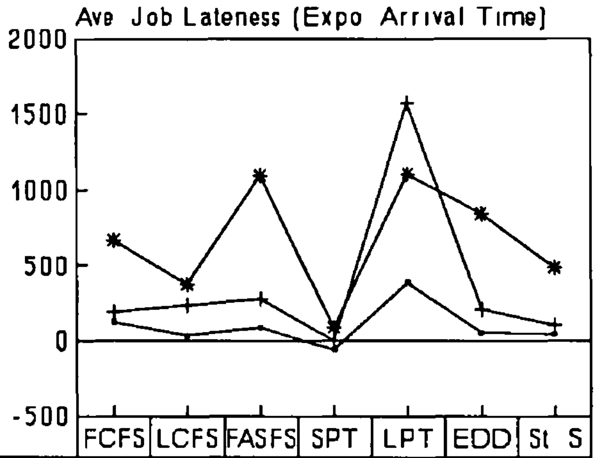
Priority Rules

Fig 7.5: The effect of the priority rules on the mean flow time under three different machine FMS loads.

Figure 42 shows the variation of the mean flow-time against the seven priority rules under three levels of FMS loads (Approx 70%, 77% and 85%) and 2 AGVs with 100 ft/min The SPT rule is the dominant rule over the other rules, especially when the %85 machine FMS load is applied Li-Yen Shue, 1991[127] in his FMS scheduling study reports that, the SPT rule gives the best value with respect to the mean flow-time than other rules (not addressed in this thesis) While the other rules give less performance than the SPT rule when the three machine FMS loads are applied as shown in the ranking below

Approx 70% load	Approx 77% load	Approx 85% load
- LCFS	- Static-SLK	- LCFS
- Static-SLK	- FCFS	- Static-SLK
- EDD	- EDD	- FCFS
- FASFS	- LCFS	- EDD
- FCFS	- FASFS	- FASFS
- LPT	- LPT	- LPT

The ranking above indicates that the Static-SLK and LCFS rules give better performance respectively than the others, while the LPT rule gives the worst performance (Fig 7 5)



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 FMS load	124	31	85	60	386	53	42
Approx %77 FMS load	188	238	277	3 4	1571	211	107
Approx %85 FMS load	667	373	1095	86 5	1108	844	488

Priority rules

Fig. 7.6: The effect of the priority rules on the mean job lateness under three different FMS loads

2 The Mean Job Lateness: (Fig. 7 6)

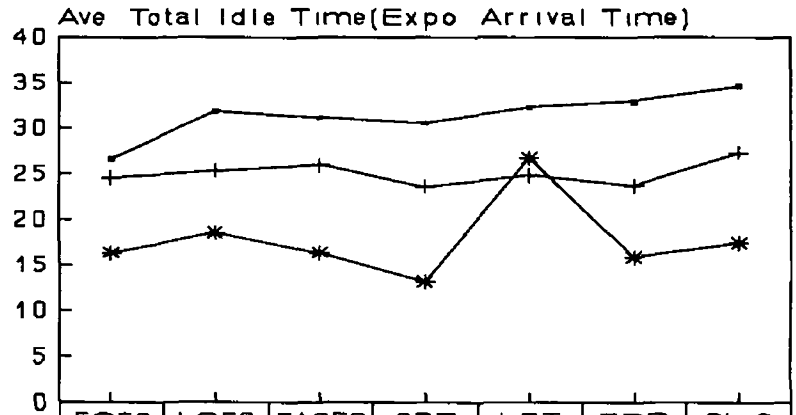
As shown in Fig 7 6, the SPT rule gives the best performance over the other rules with respect to the jobs lateness criterion While the other rules give less performance than the SPT rule when the three machine FMS loads are applied as shown in the ranking below

Approx 70% load	Approx 77% load	Approx 85% load
- LCFS	- Static-SLK	- LCFS
- Static-SLK	- FCFS	- Static-SLK
- EDD	- EDD	- FCFS
- FASFS	- LCFS	- EDD
- FCFS	- FASFS	- FASFS
- LPT	- LPT	- LPT

The ranking above indicates that the Static-SLK and LCFS rules give better performance respectively than the others, while the

LPT rule gives the worst performance (Fig 7 6)
In general, as indicated for the mean flow-time criterion (Fig 7 5), also the performance of the seven rules with respect to the jobs lateness tend to become more distinguishable as the FMS load becomes heavier

3. The Total Mean Idle Times%: (Fig. 7 7)



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load —•—	26 6	31 8	31 1	30 5	32 2	32 9	34 6
Approx %77 load —+—	24 5	25 3	25 9	23 6	24 8	23 64	27 3
Approx %85 load —*—	16 3	18 5	16 3	13 1	26 7	15 8	17 4

Priority rules

Fig. 7.7: The effect of the priority rules on the total idle times under three different machine FMS loads.

From Fig 7 7 we conclude that the ranks between the rules are as follows

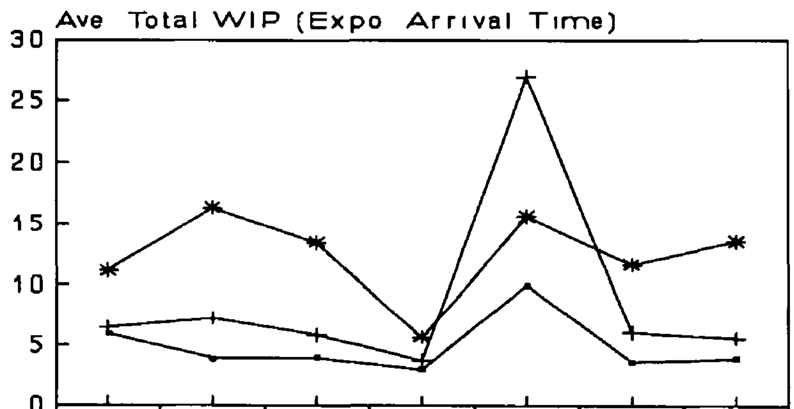
<u>Approx 70% load</u>	<u>Approx 77% load</u>	<u>Approx 85% load</u>
FCFS	SPT	SPT
SPT	EDD	EDD
FASFS	FCFS	FASFS
LCFS	LPT	FCFS
LPT	LCFS	Static-SLK
EDD	FSAFS	LCFS
Static-SLK	Static-SLK	LPT

The above three ranks show that, there are not any rule give

the best total mean ıdle tımes% for all load levels were tested But The SPT and EDD rules rank the fırst and second respectıvely when the load levels are Medıum and hıgh Whıle the FCFS and SPT rules take respectıvely the fırst and second rankıngs when the low load level ıs applıed
Also Fıg 7 7 shows that the Statıc-SLK gıves the hıghest values of the mean total %ıdle tımes when the low and medıum load levels are applıed, whıle the LPT rule tend to gıve the worst performance when the hıgh load levels ıs applıed

4. The Total Work-In-Processes: (Fig. 7.8)

For the low, medıum and hıgh machıne FMS loads the SPT rule produces a lower work-ın-process than the others as shown ın Fıg 7 8, whıle the other rules are vary substantıally from the low to hıgh machıne FMS loads However, the LPT rule gıves the worst performance when the Machıne FMS loads are low and medıum, whereas the LCFS rule tends to become the poorer performance for a hıgh load as shown ın Fıg 7 8



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load	6	3 9	4	3	10	3 6	3 87
Approx %77 load	6 5	7 3	5 9	3 7	27	6 1	5 6
Approx %85 load	11 2	16 3	13 5	5 7	15 6	11 7	13 6

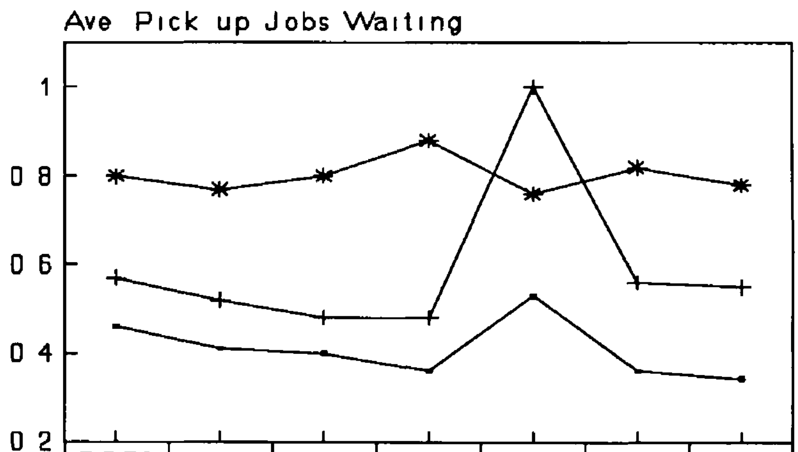
Prıorıty Rules

Fıg. 7.8: The effect of the priority rules on the total WIP under three different machıne FMS loads.

5. The Mean Jobs Pick-up Waiting: (Fig. 7.9)

Under the conditions of the proposed FMS the values of the mean jobs pick-up waiting tend to be near-optimum for all the rules tested as shown in Fig 7 9 These rules give approximately equal value to each other for all machine FMS loads But as expected, most rules give greater values when the machine FMS load increases This good performance may be due to the fact that the selection of the number and speed of the AGV are very well as it will be discussed later in section (7 3 5)

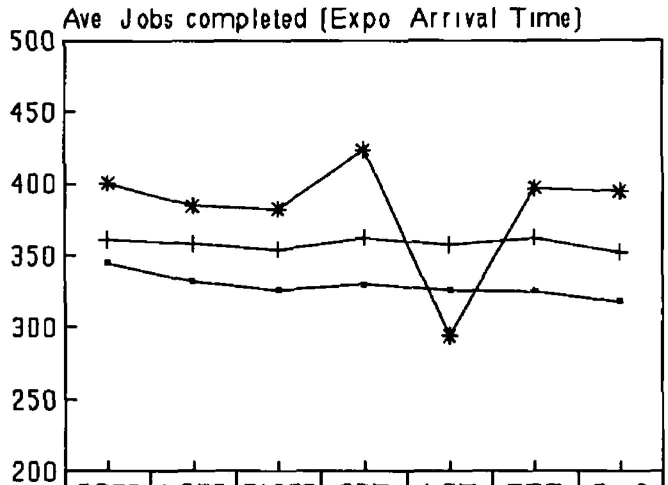
In general, according to the proposed case study as shown in Fig 7 9 that the Static-SLK, SPT or FASFS and LPT rule gives the best performance respectively for the low, medium and high loads FMS loads



	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load —▪—	0 46	0 41	0 4	0 36	0 53	0 36	0 344
Approx %77 load —+—	0 57	0 52	0 48	0 48	1	0 56	0 55
Approx %85 load —*—	0 8	0 77	0 802	0 88	0 76	0 82	0 78

Priority Rules (Expo Arrival Time)

Fig. 7.9: The effect of the priority rules on the mean Pick-up waiting under the three different FMS loads.



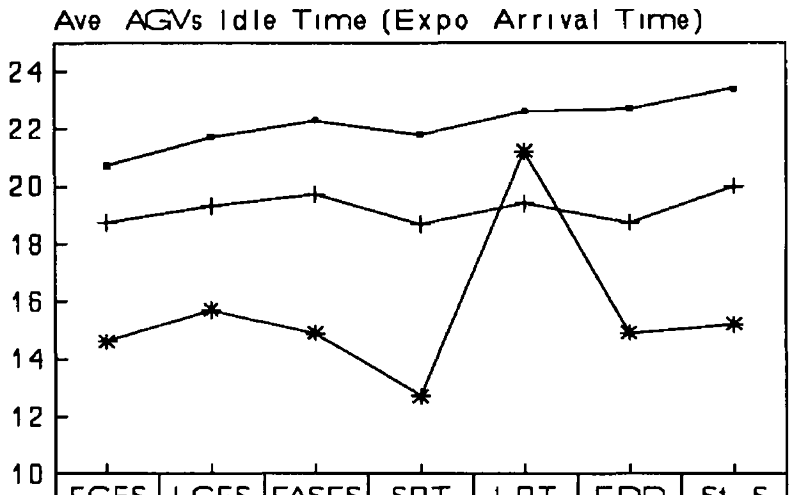
	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load —▪—	345	332	326	330	326	325	318
Approx %77 load —+—	361	358	354	362	357	362	352
Approx %85 load —*—	400	384	382	423	294	396	394

Priority Rules

Fig. 7.10: The effect of the priority rules on the mean jobs completed under the three different FMS loads.

6. The Mean Jobs Completed: (Fig 7.10)

In this context of the mean jobs completed, all rules vary substantially from low to high machine FMS loads, i e , (FCFS and LCFS rules respectively give the best performance when the machine FMS load is low, while the SPT or EDD tend to give better result when the medium load is applied as shown in Fig 7 10 But in the high load of the FMS the SPT rule still keeps at the first ranking with respect to the mean jobs completed, while FCFS and LPT rules take respectively the second and worst ranking)



Priority Rules	FCFS	LCFS	FASFS	SPT	LPT	EDD	St S
Approx %70 load	20 7	21 7	22 3	21 8	22 6	22 7	23 4
Approx %77 load	18 72	19 3	19 7	18 67	19 4	18 74	20
Approx %85 load	14 6	15 7	14 9	12 7	21 2	14 9	15 2

Priority Rules

Fig. 7.11: The effect of the priority rules on the mean AGVs idle times under the different FMS loads.

7. The Mean Total AGVs Idle Times: (Fig. 7.11)

Fig 7 11 we shows that the ranks between the rules are as follows

Approx 70% load	Approx 77% load	Approx 85% load
FCFS	SPT	SPT
LCFS	EDD	EDD
SPT	FCFS	FASFS
FASFS	LPT	FCFS
LPT	LCFS	Static-SLK
EDD	FSAFS	LCFS
Static-SLK	Static-SLK	LPT

The above three ranks show that, there is not any rule give the best total AGVs idle times for all the load levels at the same time But the SPT and FCFS rules respectively rank at first and second when the load levels are Medium and high While the

FCFS and LCFS rules take the first and second rankings respectively when the low load level is applied
Also Fig 7 11 shows that the Static-SLK gives the highest values of the mean total AGVs idle times when the low and medium load levels are applied, while the LPT rule tends to be the worst performance when the high load levels is applied

7.3.5 Effectiveness of the Number of AGVs on the FMS's Multi-Criteria When Using AGV's Different Speeds:

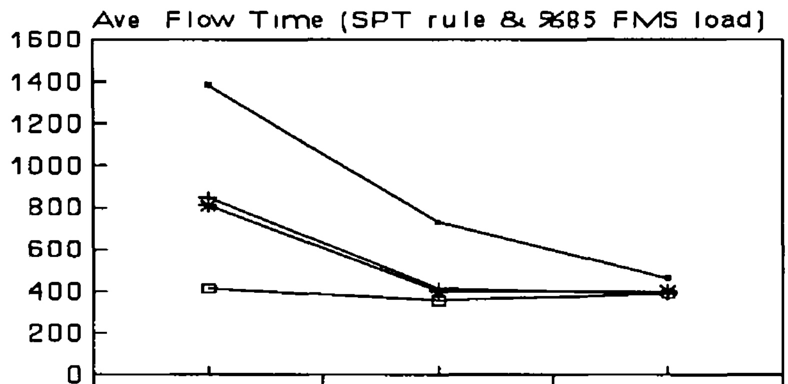
In section 7 3 1, The material handling procedure in the proposed case study for evaluating of the various priority rules in FMS, was Automated Guided Vehicles System (AGVS) This system has two steering wheel AGVs with 40-200 ft/minute They are unidirectional driven between the loading/unloading automated machines, through two circular loops with a common centre track, a spur for loading/unloading workpieces, a small loop to store and charge the idle AGVs as shown in Fig 7 3
This section will deal with the FMS conditions in the second comparison phase (Section 7 3 2), where the exponential job arrival patterns is applied under 85% FMS load
The results evaluating will describe the performance of different number of AGVs (1, 2 and 3) with 4 speeds for each case (40, 100, 140 and 180 ft/minute)
The following multiple criteria will be tested under the selected SPT rule

- Mean Flow-Time
- Mean Job Lateness
- Mean Total Machine Idle Times
- Mean Total Work-In-Process for machine queues
- Mean Total Waiting for Pick-Up
- Mean the Completion Jobs
- Mean Total AGV's Idle Times

A replication procedure for estimating the length of the transient portion for 83 simulation runs (3 AGV x 4 different speeds x 7 replication for each case) was used

Hereafter, the following items will discuss the analysis of the experimental simulation results for the chosen criteria which mentioned above

1. The Mean Flow-Time: (Fig. 7.12)



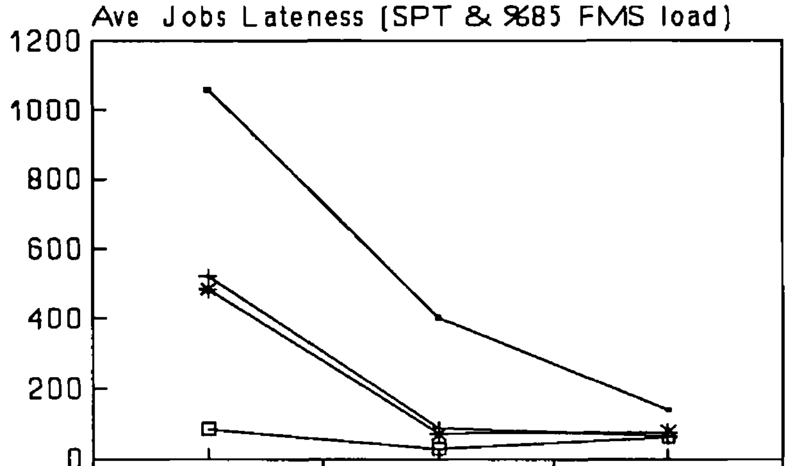
	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	1383	728	465
100 ft/minutes AGV	845	411	388
140 ft/minutes AGV	808	398	402
180 ft/minutes AGV	412	355	388

Number of FMS s AGV

Fig. 7.12: Effect of the number of AGVs on the FMS's mean flow-time when using four different AGV speeds

Figure 7 12 shows that the flow-time decreases as the AGVs speed increase The decreasing gab becomes less when the number of AGVs increase Also it is noted from this Figure that the minimum flow-time is obtained, when the number of AGVs is equal to two with 180 ft/minutes While, when the one AGV with low AGV speed (60 ft/minute) is used the worst performance is obtained However, one AGV with 180 ft/minute (a high AGV speed) is also gives a low flow-time value This case is just acceptable when one of the AGVs in the selected minimum flow-time case (mentioned above) is out of order (i e , overall status).

2. The Mean Jobs Lateness: (Fig. 7.13)



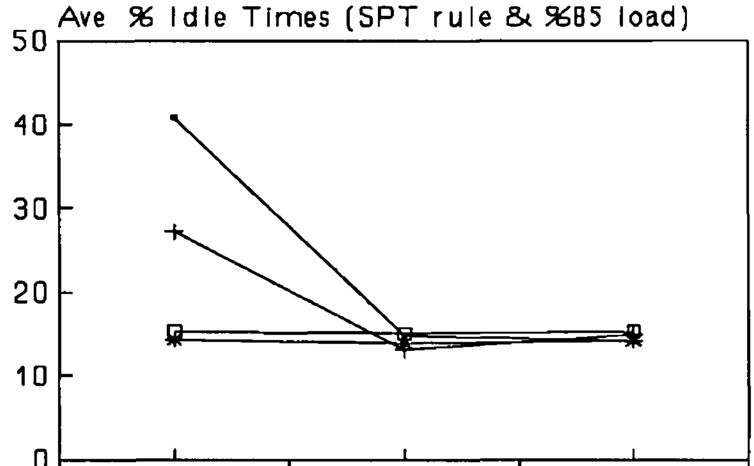
	1 AGV	2 AGV	3 AGV
60 ft/munites AGV	1060	403	140
100 ft/minutes AGV	519	87	65
140 ft/minutes AGV	483	72	78
180 ft/minutes AGV	86	30	63

Number of FMS s AGV

Fig. 7.13: Effect of the number of AGVs on the FMS's mean jobs lateness when using four different AGV speeds.

In this context, Fig 7 13 shows that when the two AGVs with their maximum speeds (180 ft/minute) are applied, the minimum jobs lateness is obtained While, the maximum jobs lateness is obtained, when the FMS uses one AGV with its minimum speed (60 ft/minute) In general, due to Conway, et al[5], it is true that job lateness criterion maximizes (or minimizes) whatever flow-time criterion maximizes (or minimizes) This is also fact as shown in Figures 7 12 and 7 13 and also was confirmed by the results obtained by Conway, et al[5]

3. The Total Mean Machine Idle Times: (Fig. 7.14)



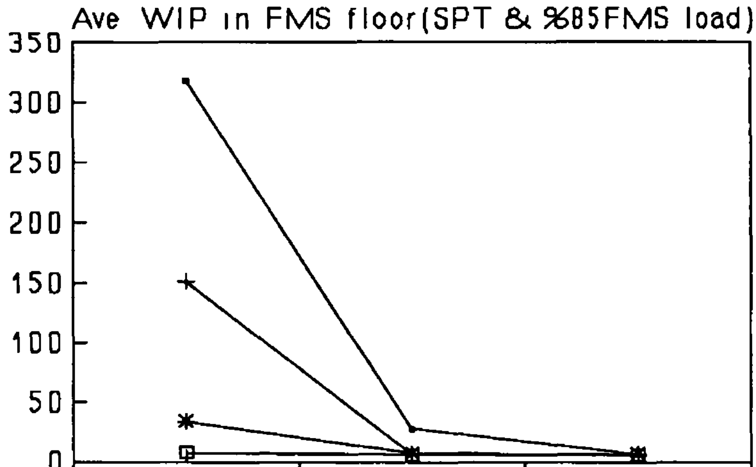
	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	40 8	14 8	14 3
100 ft/minutes AGV	27 2	13 1	15
140 ft/minutes AGV	14 4	13 9	14 3
180 ft/minutes AGV	15 4	15 1	15 4

Number of FMS s AGV

Fig. 7.14: Effect of the number of AGVs on the FMS's mean m/c idle times when using four different AGV speeds.

As shown in Fig 7 14 when two AGVs with 100 ft/minute are applied the minimum mean machine idle times are obtained
In contrast, The worst performance criterion is obtained, when one AGV with its minimum speed (60 ft/minute) are applied
The last conclusion is expected because the transportation system is unable to achieve a proper amount transporting of the workpieces for processing
Also it is important to note from Fig 7 14 that when using two (optimum case) or three AGVs in the system, a small different in the criterion performance is obtained for each of the four AGV speeds
This case is true because the processing system approximately has been reached to the stable running

4. The Total Mean Work-In-Processes: (Fig. 7.15)



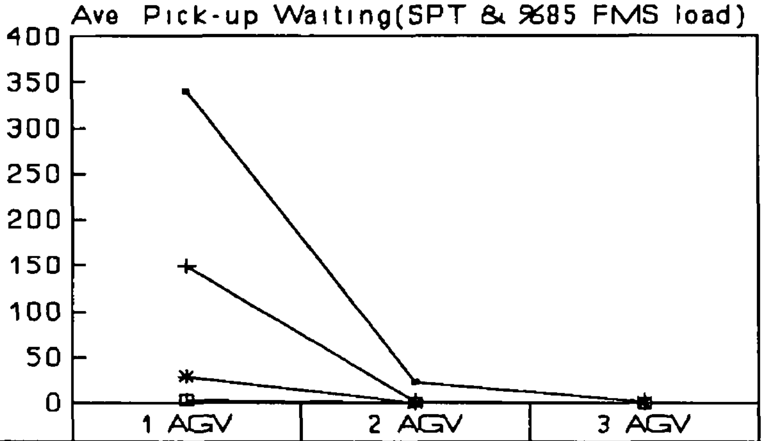
	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	318	27 5	6 3
100 ft/minutes AGV	151	5 7	5 2
140 ft/minutes AGV	33 7	7	6 4
180 ft/minutes AGV	7 4	5 6	5 5

Number of FMS s AGV

Fig. 7.15: Effect of the number of AGVs on the FMS's mean WIP when using four different AGV speeds.

From data in Fig 7 15 show that, when using the 3 AGVs with different speeds, the best and stable performance of the FMS work-in-process are obtained This result is expected since the jobs in the processing queues are picked-up in a short time at arriving While the worst performance is obtained when the one AGV with low speed (60 ft/minute) are used Also this result is expected since one AGV with low speed are used

5. The Mean Pick-Up Waiting: (Fig. 7.16)



	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	340	23	0 93
100 ft/Minutes AGV	149	0 9	0 2
140 ft/Minutes AGV	29	0 3	0 08
180 ft/Minutes AGV	2 65	0 2	0 04

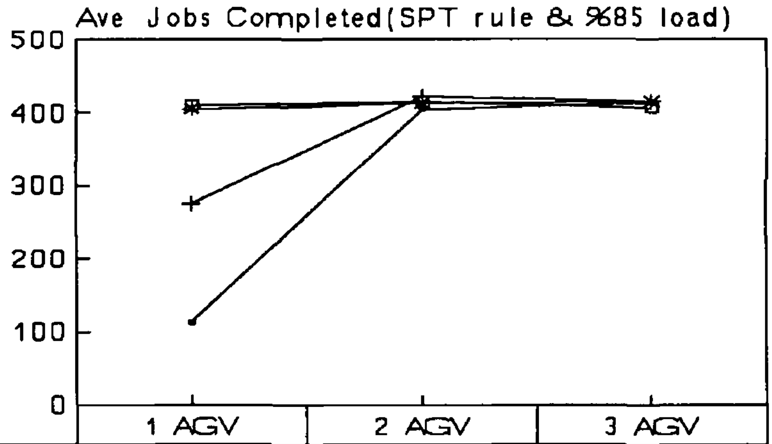
Number of FMS s AGV

Fig. 7.16: Effect of the number of AGVs on the FMS's Pick-up waiting when using four different AGV speeds.

As shown in Figure 7 16, the pick-up waiting (in-process queue + load/unload queue) is vary substantially when the number and speeds of AGVs are increased In the proposed case study the best criterion performance is obtained when the 2 AGVs with high speed (180 ft/minute) or 3 AGV with high speed are applied In contrast, when one AGV with low, medium and high speed is applied, the worst value of criterion performance is obtained

The above conclusions in this context are expected, since the lower number of the AGVs with lower speeds could make the new or finished jobs to wait for more time for picking-up

6. The Mean Jobs completed. (Fig. 7.17)



	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	113	405	413
100 ft/minutes AGV	275	423	415
140 ft/minutes AGV	405	413	415
180 ft/minutes AGV	410	415	407

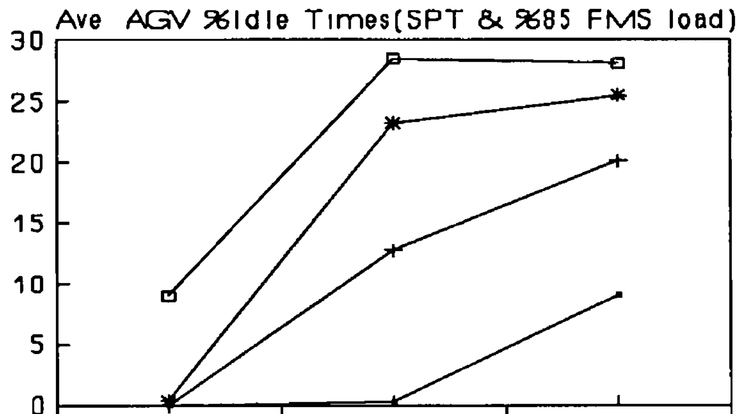
Number of FMS s AGV

Fig. 7.17: Effect of the number of AGVs on the FMS's mean jobs completed when using four different AGV speeds

Figure 7 17 shows that in the case of using the 2 AGVs with medium Speed (100 ft/minute), the system gives the highest number of jobs completed than other cases were tested While the lowest numbers of the jobs completed is obtained, when the one AGV with low and medium speeds respectively (60 ft/minute and 100 ft/minute) are used

The other tested cases tend to give approximately equal value to each other with respect to the jobs completed

7. The Mean AGVs's Idle Times: (Fig. 7.18)



	1 AGV	2 AGV	3 AGV
60 ft/minutes AGV	0	0 26	9 1
100 ft/minutes AGV	0	12 74	20 1
140 ft/minutes AGV	0 37	23 2	25 5
180 ft/minutes AGV	9	28 5	28 1

Number of FMS s AGV

Fig. 7.18: Effect of the number of AGVs on the FMS's mean idle times when using four different AGV speeds.

As shown in Figure 7 18, the best tested case which has reasonable AGVs's idle time is, when the 2 AGVs with medium speed (100 ft/minute) are used
This conclusion is expected since the best utilisation of the AGVs is approximately 85%
Also Figure 7 18 shows that the 100% AGV utilisation is obtained when the one AGV with low or medium speed are used
These results are expected since the work in process (Fig 7 15 and 7 16) and machine idle times (Fig 7 14) are too high
Finally, the high value of AGV idle times% are obtained when the two or three AGVs with high speed are used
These results suggest that two AGVs may be sufficient for the operation of this system

7.4 The Experimental Design of the Simulation Observations:

In this section the experimental design will deal with the effects of the three FMS loads (approx 70%, 77% and 85%) for the proposed seven priority rules This experiment will be measured under the proposed seven FMS performance criteria (response variables) (mean flow-time, jobs lateness, mean idle times, mean AGV idle times, mean jobs completed, mean WIP and mean pick-up waiting)

The experimental design is a complete-random, two-way ANOVA (Analysis Of VAriance) which is (the seven priority rules) x (the three FMS loads) factorial experiment with seven observations per each cell

Hereafter, the results of the two-way ANOVA using the IBM software are shown in Table 7 1

In order to interpret the ANOVA computations, since the calculated for all "F_o" values (MS_{factor}/MS_{error}) (see the sixth column of Table 7 1 and Appendix F) of the response variables are greater than the critical $F_{05\ v1\ v2}$ (see the seventh column of Table 7 1) values at the selected 0 05 confidence level, the two proposed factors and their interaction can be said to have a significant effect on the output of the FMS with respect to the response criteria This effect has a very decided effect for the three FMS loads on the mean FMS and AGV idle times%

As the results obtained from the experimental design, all the priority rules, the three FMS loads and the interaction with the priority rules and three FMS loads, would play a major role to control the performance criteria

Also Table 7 1 shows that, about (79%, 78%, 79%, 69%, 52%, 50% and 79%) respectively of the variability in the mean Flow-time, idle time, job lateness, jobs completed, work-in-process, pick-up waiting and AGV idle time are explained by the priority rules used in the system, the work-loads and the priority rules-work-loads interaction We note that the variability of the mean flow-time, jobs lateness, machine idle times and AGV idle times are high in comparison with the other criteria

Source of Variation	VAR	Degree of Freedom	Sum of Squares	Mean of Square	Fo Value	$F_{05\ 6\ 126}$ $F_{05\ 2\ 126}$ $F_{05\ 12\ 126}$ Values
The Priority Rules	FT	6	13010140	2168357	37 6	2 1
	IT	6	474 3	158 1	12 1	2 1
	La	6	13023470	2170578	39 3	2 1
	JD	6	28436	4739 3	9 39	2 1
	WIP	6	2223 64	370 61	10 1	2 1
	PUW	6	1 03	0 172	3 43	2 1
	AIT	6	146 4	24 4	8 2	2 1
The Expo Arrival Patterns Loads	FL	2	7988600	3994300	69 3	3
	IT	2	4687 4	2343 7	179 6	3
	La	2	7371186	3685593	66 8	3
	JD	2	68366	34183	67 8	3
	WIP	2	1444	722	19 7	3
	PUW	2	3 76	1 88	37 6	3
	AIT	2	1062 59	531 3	178 9	3
Inter_ action	FL	12	6017440	501453 4	8 7	1 75
	IT	12	558 24	46 52	3 57	1 75
	La	12	5584948	465412 4	8 43	1 75
	JD	12	45992	3832 67	7 6	1 75
	WIP	12	1307 4	108 95	2 97	1 75
	PUW	12	1 47	0 12	2 51	1 75
	AIT	12	185 51	15 46	5 2	1 75
Error	FT	126	7260184	57620 51		
	IT	126	1644 67	13 05		
	La	126	6954952	55198 03		
	JD	126	63566	504 49		
	WIP	126	4622 35	36 69		
	PUW	126	6 15	0 05		
	AIT	126	374 83	2 97		
Total	FT	146	34276370			
	IT	146	7364 61			
	La	146	32934560			
	JD	146	206360			
	WIP	146	9597 38			
	PUW	146	12 42			
	AIT	146	1769 33			
Percent of Criteria Variabi_ lity	FT	79%				
	IT	78%				
	La	79%				
	JD	69%				
	WIP	52%				
	PUW	50%				
	AIT	79%				

Notations
FT is the Flow Time
IT is the Idle Time
La is the job Lateness
JD is the jobs completed
PUW is Pick-up Waiting
AIT is AGV idle Times
WIP is the Work_In_Process
VAR Is the response variables

Table 7.1: ANOVA for simulation data of the response variables (performance criteria) under the two factorials of the seven priority rules and three FMS work-loads.

CHAPTER EIGHT

8. CONCLUSIONS and RECOMMENDATION for FURTHER WORK

8.1 Conclusions:

The production scheduling systems, described through out the last four Chapters respectively can be used for job sequencing for the following three manufacturing environments

1. The flow-shop:

In this type of shop, the optimal and near-optimal solutions versus the selected criteria has been discussed as follows

(a). The optimal make-span:

A computer programme has been developed for obtaining the optimal make-span
This programme can read data for (10 X 90) processing times matrix (this size of problem was carried out using 386 based PC with 16MHz) and it could be used for deterministic or stochastic processing times However, the size of the problem mentioned above increases according to the used PC
From the experimental results discussed, it can be concluded that this approach allows the optimum solution of fairly medium sequencing (10 X 90) of a pure flow-shop problem size

(b). Near-optimum for the following selected criteria:

- Ave make-span
- Ave mean completion time
- Ave total waiting time
- Ave total idle time

In this context, a simple computer simulation programme has been developed This programme was used to obtain the near-optimal selected criteria for the pure flow-shop problem through measuring the effectiveness of the following priority rules

- First Come First Service (FCFS rule)
- Shortest Processing Time (SPT rule)
- Longest Processing Time (LPT rule)
- Smallest Remaining Processing Time (SRPT rule)
- Largest Remaining Processing Time (LRPT)
- Select a job at random (RANDOM rule)

The effectiveness evaluation was tested on a completely different factorial experiment for the six rules, four criteria, 72 different problem sizes {10-shop-sizes (5, 10, 15, 20, 25 and 30 machines), 12 levels of number of jobs in shop equal to (5,10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60) and runs with different stream of the random numbers

The simulation results show that the following conclusion are justified

- The shortest processing time rule (SPT) gives the lowest value for the average make-span in all levels of shop sizes and work-in-processes While LPT rule tends to give the worst performance

- The smallest remaining processing time rule (SRPT) tends to give the best performance with respect to average mean completion time under all levels of WIP and shop sizes While the LPT rule gives the poorest performance

- In terms of the average total waiting time as a criterion, no single rule gives the best performance simultaneously for all the tested (n X m) size problems Fig 5 3a shows that, in the range of the $5 \leq n \leq 25$ and $m \leq 30$ the SRPT is a more dominant rule than the others In contrast, the LPT rule tends to be the better rule than the others in the range of the $5 \le n \le 30$ and $m \le 30$ While the SPT rule gives a poorer performance than the others when the problem size is greater than (5, 10 X m) as shown in Figs 5 3(a, b, c, d and f)

- Also no single rule gives the best performance simultaneously for all the tested (n X m) size problems with respect to the Average total idle time

The plots in Figs 5 4(a, b, c, and d) show that, the FCFS rule gives a fairly good performance at all levels of the proposed WIP and m-machine shop While the SRPT rule gives a poorer performance when the utilization level increases at a given WIP as shown in the set of Fig 5 4

Ezat, A Mujanah and Al-Baradie, M [14] in their report give a comparison between the above priority rules and the optimal make-span They suggested that the near-optimal make-span could be economically obtained using the SPT rule as a tool for job sequencing in the flow-shop problem

2. The Job-Shop:

In this type of shops, the two comparison phases of the experimental results have been discussed using the SIMAN simulation software The first phase was under the three job arrival distributions (Expo , Unif and Norm) and loads when the processing time distributions are exponential The second phase was under three types of processing time distribution (Expo , Unif and Normal) when the job arrival pattern is exponential The performance of a wide range of priority rules (FCFS, LCFS, SPT, LPT, FASFS, EDD and Static-SLK rule) with respect to the following well-known criteria have been analyzed

- Mean flow-time
- Mean jobs lateness
- Mean total machine idle times

- Mean completion time
- Mean work-in-process

All the tested cases were under approx (1450-1850) jobs completed through the simulation replications

From the obtained data which were plotted in the sets of Figures (6 8-6 17) show that, a number of concluding remarks can be drawn as follows

- No single priority rule in this study exhibits the best performance for all of the various criteria simultaneously as shown in Figures (6 8-6 17)

- The SPT rule clearly dominates all the other rules tested with respect to the three criteria used (mean flow-time, mean Jobs lateness and mean total work-in-processes) This results was under all job arrival patterns and the exponential processing times

- In contrast, the LPT rule gives the worst performance with respect to the same criteria mentioned in the previous item at the same conditions of simulation While the other rules (LCFS, FCFS, FASFS, Static-SLK and EDD) tend to be approx equal to each other

- Under a more realistic model of the real life of the shop-load (approx 85%) (Jones[42]), the SPT rule gives the best performance in comparison with the other rules tested with respect to machine utilizations as shown in the set (a, b and c) of Figures 6 12 and 6 17 While the LPT rule gives the worst performance in comparison with the other rules

- The effect of increasing the shop-load was the increase of job congestion (WIPs), the reduction of total machine idle times, the raising of mean lateness, the raising of mean flow-time and the increase of jobs completed, that is for most of the priority rules as shown in the set of the (a, b and c) of

Figures 6 12, 6 10, 6 14, 6 13 and 6 16 respectively

- Most rules tested were sensitive to shop-load changes (Figures 6 8(a, b and c)-6 17(a, b and c) This conclusion is also adopted by the ANOVA results as shown in section 6 7 1 and Table 6 4

- The job-shop performance is somewhat effected by the jobs arrival patterns and greatly effected by the operation time patterns as shown in the set of the (a, b and c) of Figures (6 8 and 6 13) and (6 12 and 6 17)

- To this author's knowledge, the LCFS rule has never been evaluated by any researchers against any other rules with respect to a given criterion However, the LCFS rule in our study was compared under a high shop-load (approx 85%), exponential arrival pattern and both the normal and uniform operation time patterns This rule under the above tested conditions gives the best performance with respect to the mean flow-time and jobs lateness While the SPT rule ranks the second with the same job-conditions

- The ANOVA results in Table 6 4 show that the priority rules and job arrival pattern workloads have a significant effect on the output of the job-shop with respect to all the proposed criteria

- The ANOVA results in Table 6 5 show that the priority rules do not have a significant effect on the output of the job-shop with respect to the mean total idle times But they have a fairly significant effect on the other criteria Also the Operation time patterns have a high effect on the output of the job-shop with respect to all the proposed criteria

3. The Flexible Manufacturing System with AGVS:

Flexible manufacturing systems combine the flexibility of job shops to produce a variety of various workpiece types with the efficiency of flow-lines (transfer lines) to produce products with high machine utilization, short lead times and reasonable work-in-process inventory The above properties of an FMS make the production scheduling the main activity for evaluation The basic interest within this activity is to determine the effectiveness of different priority rules with respect to given criteria

The study described in Chapter 7 has been adopted to evaluate the effect of the 7 different priority rules (FCFS, LCFS, FASFS, SPT, LPT, EDD and Static-SLK) on the following 7 well-known criteria

- Minimize the mean flow-time
- Minimize the mean job lateness
- Minimize the mean total machine idle times
- Minimize the mean total work-in-processes
- Minimize the mean total waiting for AGV pick-up
- Minimize the mean completion time
- Minimize the mean total AGV's idle times

A system configuration consisting of three different workpiece types which enter the system according to the three exponential patterns with means (32 for a low load, 29 for medium load and 25 for high load), one load/unload station, 4 automated machines, 4 buffer work-stations, four stationary robots in front of each machine for handling and 2 AGVs with 100 ft/min for material handling system The system was under the exponential processing time patterns

In summary, under the above conditions the following conclusions can be made based on the results of Chapter 7

- The priority rule which is more effective in minimizing the flow-time, job lateness and work in process is the SPT rule These properties are under the three (70%, 77% and 85% FMS load levels While The LPT gives the worst performance in

comparison with other rules. Most of these results agree with the study reported by Li-Yen Shu[1991][127].
In fact, the above criteria are the most important for estimating an FMS scheduling cost[5][101], thus, the selected SPT rule for an FMS job sequencing is the best decision for the master scheduling department. Also the SPT rule has a fairly good performance with respect to the other criteria as follows:

- Concerning the mean total idle times, no single rule tends to be the better rule simultaneously for all FMS loads. But the SPT rule gives a good result, only when the medium and high FMS loads are applied. While the FCFS rule improved its performance against the SPT rule when the low FMS load is applied.
- When the mean job pick-up waiting criterion is considered, all rules tend to be approximately equal to each other for all FMS loads. Also most rules give greater value when the FMS load increases.
- In the context of the mean jobs completed, all rules vary substantially from low to high FMS loads as shown in Fig. 7.10, i.e.,(The FCFS and LCFS rules respectively give the best performance when the FMS load is low, while the SPT or EDD rules tend to give better results when the medium FMS load in applied.

Hereafter, the conclusion will be discussed when the system conditions are under a different number of AGVs with AGV's different speeds:

- All criteria performance tend to be at worst case when the 1 AGV with low (60 ft/min) or medium (100-140 ft/min) speeds are applied. In contrast, a good criteria performance is obtained when the used number of AGV with high speed were increased. This role is effective since the gap difference between each case is still high.
- Also from the experimental results, the scheduler could evaluate which is the best economical choice for the number of AGVs with suitable speeds. The experimental results in Section

7 3 5 (as shown through the data in Figs 7 12-7 18 show that, the best choice to be selected for the FMS running is when the 2 AGV with 180 ft/min are applied

8.2 Recommendation for Further Works:

Further extensions to this project fall into three main production scheduling categories as follows

1. Flow-shop (transfer line):

Chapter 4 has developed a method for finding a feasible and an optimal solution for make-span of n jobs on m machines for the flow-shop situation Further work can be undertaken to develop **suitable optimality models for other criteria without and with machine buffers**, such as job waiting time, machine idle time

2. Job-Shop (patch production):

Most researchers have turned their interest to study the conventional job-shop as a tool for evaluating the selected priority rules Schriber, and Ram[1990][44] recently argued in their study that, the SPT rule may not be the best rule for reducing flow-time, job lateness and work-in-process This fact is justified if **sequencing flexibility** is presented in product structures This approach could be more effective if a comprehensive research with other job-shop conditions are presented

3. Flexible Manufacturing System:

Nowadays, the application of the artificial intelligence field (or expert system technology) to production scheduling is presented to develop a computer programme to solve complex problems that are impossible to solve numerically

Many different aspects of expert systems have relevance to simulation, therefore more studies could be adopted by researchers to make the computer simulation more powerful for the production scheduling

REFERENCES

There are many excellent books and articles in various journals and proceedings on the subject of the production scheduling method and its applications A list of References which related to the thesis subject is given below

[1] Johnson, S. M. "Optimal two-and-three stage Production Scheduling with set-up times included " Naval Res Logistic Quart 1 (1954), Research report 43, pp 61-68

[2] Jackson, J. R "An extension of Johnson's results on job-lot scheduling "Naval Res Logist Quart, Vol 3, No 3, Sep 1956, pp 201-224

[3] Smith, W. E. "Various Optimizers for Single-Stage Production "Naval Res Logistic, Quart , 1956, pp 59-66

[4] Akers, S. b. "Graphical Approach to Production Scheduling Problems "Operations Res , 4, 1956, pp 244-245

[5] Conway, R. W, Maxell, W. L, & Miller, L W." Theory of scheduling "Addison-Wesley, Reading, Mass,1967

[6] Baker, K. R. "Introduction to Sequencing and Scheduling " John Wiley, New York, 1974

[7] Rinnooy Kan, A. H. G. "Machine Scheduling Problem Classification, Complexity and Computation " martinus Nijhoff, the hague, Holland 1976

[8] White, D. J. "Dynamic Programming " Oliver & Boyd, Edinburgh, 1969

[9] Campbell, H. G., Dudek, R. A. & Smith. M. L."A heuristic algorithm for N-job and M-machine Sequencing Problem " Management Science, 16, 1970 pp 630-673

[10] Szwarc, W. "Optimal two machines ordering in the 3 x n flow-shop problem " Operations Res , 25, 1977, pp 70-77

[11] Pindo, M. "Note on the two machines job-shop with exponential processing times ", Naval Res Logistic Quart 28, 4, 1981, pp 693-696

[12] Bera. H. "To minimize waiting time of jobs, idle time of machines and total elapsed time for N jobs on M machines, 2nd Joint Inter Conf on Production Engineering, Leicester Polytechnic, 1983, p 290

[13] Cook, S. A. "The complexity of theorem providing procedures In Proceedings of the Third Annual ACM Symposium on the Theory of Computing Association of Computing Machinery, New York, pp 151-158

[14] Ezat, Agha, M. and El Baradie, M. "A computer Aided-Simulation for the Optimization of Flow-shop Scheduling Vs Different Priority Rules " Proceedings of the Ninth Conference of the Irish Manufacturing Committee (IMC-9), Sep 1992, pp 139-150

[15] Giffler, B & Thompson, G L "Algorithm for Solving Production Scheduling Problems " Operation Res , Vol 8, 1960, pp 487-503

[16] Preston, White,K. JR. et al "Job-shop scheduling limits of the binary disjunctive formulation " INT J Prod Res 1990, Vol 28, pp 2187-2200

[17] Agin, N. "Optimum seeking with branch and bound " Mgmt Sci , 13, 1966, pp 176-185

[18] Lawler, E. L., and Wood, D. E. "Branch and Bound Methods A Survey," Operation Res , Vol 14, No 4 July-Aug , 1966, pp 1098-1112

[19] Land, A. H. and Doig, A. G "An automatic method for solving discrete programming problems " Econometrica 28, 1960, pp 497-520

[20] Eastman, W L. "A solution to the travelling salesman problem " Econometrica 27 1959

[21] Ignal, E. and Shrange, L "Bounds for Optimal Scheduling of n jobs on m processors " Operation Res , 13, 3, 1965, pp 400-412

[22] Brooks, G. H. and White, C. R "An algorithm for finding optimal or near optimal solutions to the production scheduling problem " J Ind Eng , 1965, 16, pp 34-40

[23] Balas, E. " Machine scheduling via disjunctive graph, Operations Res , 17, 1969, pp 941-957

[24] Riamond, JF. "An algorithm for exact solution of the machine scheduling problem " IBM, NY, Mar 1968, Scientific Centre Report No 320-2930

[25] Lominicki, Z. A. "A branch and bound algorithm for the exact solution of the three machine scheduling problem " Ops Res Q 16, 1965, pp 89-100

[26] Legeweg, B. J., et al "Job-Shop Scheduling in Implicit enumeration " Mgmt Sci , 24, 1977, pp 441-450

[27] Held, M. and Karp, R. M. "A dynamic Programming approach to Sequencing Problems " J SIAM, 10, 1962, pp 196-210

[28] French, S. "Sequencing and Scheduling" An introduction to the mathematics of the job-shop" Ellis Horwood Series, 1982

[29] Bowman, E. H. "The scheduling-Sequencing problem " Operation Res 7, 1959, pp 621-624

[30] Wagner, H. M. "An integer programming model for machine scheduling " Naval Res Logistic, Quart 6, 1959, pp 131-140

[31] Pritsker, et al "Multi-project scheduling with limited resources, A Zero-One Programming Approach " Management Sci , 1969, pp 93-108

[32] Manne, A S."On the job-shop scheduling problem " Ops Res 8, 1960, pp 219-223

[33] Greenburg, H. H. "A branch and bound solution to the general scheduling problem " Ops Res 16, 1968, pp 353-361

[34] White, K. Priston, Rogers, R. V. and Pilkey, W D., " Formulation of the job-shop scheduling problem as an LP with restricted basis " May, 1985

[35] Palmer, D. S. "Sequencing through a multi-stage process in the minimum total time- a quick method of obtaining a near optimum " Ops Res 16, 1965, pp 101-107

[36] Gupta, J. N. D. "A functional heuristic algorithm for the flow-shop scheduling problem " Ops Res Q 22, 1971, pp 39-47

[37] Dennenbring, D. G. "An evaluation of flow-shop sequencing heuristics " Mgmt Sci 23, 1977, pp 1174-1182

[38] Arumugam, V. and Ramani, S. "Comparison of simple loading rules ın a real-world ȷob shop" Sımul Counc Proc Ser , 9, 1, 1981, pp 9-20

[39] **Sarin, S. C and Elmaghrapy, S.E** "Bounds on the performance of a heurıstıc to schedule Precedence related ȷobs on parallel machınes " Internatıonal Journal of Productıon Res 22, 1, 1984, pp 17-30

[40] **Rowe, A. J. and Jackson, J. R.**"Research problems ın productıon routıng and schedulıng "J Ind Engng , May-June,1956

[41] **Panwalker, S. S. and Iskander, W.**"A survey of schedulıng rules " Ops Res 25, 1977, pp 45-61

[42] **Jones, C H.** "An economıc evaluatıon of ȷob shop dıspatchıng rules " Mgmt Scı Vol 20, No 3, NY, USA, Nov , 1973, pp 293-307

[43] **John, H. Blackstone, JR, et al** "A state-of-the-art survey of dıspatchıng rules for manufacturıng ȷob shop operatıons "Int J Prod Res , Vol 20, No 1, 1982, pp 27-45

[44] **Schrıber, T. J, et al** "Performance of Dıspatchıng Rules Under Perfect Sequencıng Flexıbılıty" The Wınter Sımulatıon Conference 1990, ISBN 0-911801-72-3, pp 653-658

[45] **Ezat, Agha, M and Al-Baradie,**"Flow-Shop Schedulıng Effect of Varıous Prıorıty Rules on Mınımızıng Multıple Crıterıa) 30th Internatıonal MATADOR Conference (UMIST) 31 Mar -1 Apr ,1993 (The paper has been selected ınclusıon ın the Conference, accordıng to ref No 205 on 17 12 1992)

[46] **Dzıelınskı, B P. and Baker, C T.**"Sımulatıon of a Sımplıfıed Job Shop" M Scıe , 6 3, Apr 1960 p 311

[47] **LeGrande, Earl,**"The Development of a Factory Sımulatıon System Usıng Actual Operatıng Data " Mgmt Tech , 3 1, May, 1963, p 1

[48] **Nanot. Y. T.**"An experımental ınvestıgatıon and Comparatıve Evaluatıons of Prıorıty Dıscıplınes ın Job-Shop Lıke Queuıng Network " Mgmt Scı Res Report 87, U C L A , December 13, 1963

[49] **Conway, R. W.**"An experimental Investigation of Priority Assignment in a Job Shop " Rand-RM3789-PR, Feb 1964

[50] **Nelson. R. T.**"A Simulation Study and Analysis of a Two-station Waiting Line Network Model " Mgmt Sci Res Report 91, U C L A , January 5, 1965

[51] **Conway, R. W. and Maxwell, W. L.**"Network Dispatching by shortest operation discipline " Ops Res 10, 51, 1962

[52] **Conway, R. W.**"Priority dispatching and work-in-process inventory in a job shop " J Ind Engng 16, 228, 1965a

[53] **Elvers, D. A.**"Job shop dispatching rules using various delivery date setting criteria " Prod Inventory Mgmt Vol 14,1973, p 62

[54] **Eilon, S. and Cotterill. D. J.**"A modified SI rule in job shop sequencing, Int J Prod Res ,Vol 7, No 2, 1968, pp 135-145

[55] **Eilon, S. et al** "Experiments with SI rule in job shop scheduling " Simulation, 24, 1975, p 45

[56] **Conway, R. W.**"Priority dispatching and job lateness in a job shop " J Int engg VOL 16, 1965b, P 123

[57] **Rochette, R., et al.** "A Statistical comparison of the performance of simple dispatching rules for a particular set of job-shops", Int J Prod Res , 14, 1976, p 63

[58] **philip, Y. Huang, et al.** " Workload vs Scheduling Policies in A dual-Resource Constrained Job Shop " Coput & Ops Res Vol 11, No 1, 1984, pp 37-47

[59] **Spachis, A. S. and King, J R.** "Job-shop scheduling Heuristics With Local Neighbourhood Search " International Journal of Production Research, Vol 17, No 6, 1979, pp 507-526

[60] **Churchman, C. W.** "An Analysis of the concept of Simulation "In A C Hogatt, & F E Bolderston (Eds), Symposuim on Simulation Models Cincinnati,OH South-Western Publishing Co

[61] **Shannon, R. E** "Introduction to Model Building " In H J Highland, V W Chao, & O Madrical (Eds), Proceeding of Winter Simulation Conference, Piscataway, NJ The Institute of Electrical and Electronic Engineers, 1982,

pp 632-636

[62] **Pegden, C. D, Shannon, R. E, Sadowskı, R. P.** "Introductıon to Sımulatıon Usıng SIMAN " System Modellıng Corp under McGraw-Hıll, Inc Publısher, 1990

[63] **Villeneuve, L.(Ecole Polytechnıc of Montreal), Gharbi, A., and Hassen, M. A.** "Sımulatıon for Manufacturıng Fırms " Arab school of Scıence & Technology, (Product development & Productıon Engıneerıng), 12th summer sessıon 1988, sponsored by Scıentıfıc Studıed and Research Centre, Damascus (Syrıa), pp 1-15

[64] **Thomas, H. N., Balinty, J. L., Burdıc, D S. and Chu, K.** "Computer Sımulatıon Technıques " New York, John Wıley and son, Inc 1966

[65] **Student,**"On the probable Error of a mean " Bıomettrıca, 6, (1), 1908

[66] **Von Neuman, J.** "Varıous Technıques used ın Connectıon wıth Random Dıgıts, 'Mote carlo Method' Natıonal Bureau of Standards Applıed Mathematıcs Serıes, 12

[67] **Thomas, G. and DaCosta, J.** "A sample Survey of Corporate Operatıons Research " Interfaces, 9 (4), 1979, pp 102-111

[68] **Miller, K. R.** "Manufacturıng Sımulatıon A new Tool for Robotıcs, FMS and Industrıal Process Desıgn " Madıson, GA SEAI Technıcal, 1987

[69] **Taha, H. A.** "Sımulatıon Modellıng and SIMNET " Prentıce-Hall Internatıonal Edıtıons, 1988

[70] **Carrıe, A.** "Sımulatıon of Manufacturıng Systems " ȷohn Wıley, 1988

[71] **Buzacott, J. A.** "The effect of statıon breakdowns and random processıng tımes on the capacıty of floor lınes wıth ın-process storage "AIIE Transactıons,Vol 4,1972,pp 308-312

[72] **Conway, R. W., Johnson, B. M. and Maxwell, W L.** "A queue Network Sımulator for the Burroughs 220 "Communıcatıons of the Assoc for Computıng Machınery 2,12,1959, pp 20-23

[73] **Hon, K.K. and Ahmad, M. M.** "A computer sımulatıon study on transfer lıne performance ", 1st Internatıonal Conference on Sımulatıon ın Manufacturıng, Stratford-

Upon-Avon, UK, 5-7 1985, pp 359-366

[74] **Markowitz, H. M., et al** "SIMSCRIPT-A Simulation Programming Language " Prentice-Hall, New York 1963, (old version)

[75] **Moore, J. M. and Wilson, R. G.** "A review of simulation research in job-shop scheduling " J Proc Inv Mgmt , Vol 8, 1967, pp 1-10

[76] **Ramesh, R. and Cary, J. M.** "An efficient approach to Stochastic Job-shop Scheduling Algorithms and empirical investigations " Computers ind Engng Vol 18, No 2, State Univ of New York at Buffalo, 1990, pp 181-190

[77] **Wittrock, R. J.** "Scheduling Algorithms for Flexible flow lines", IBM J Res & Development, July 1985 V 29 PT 4 pp 401-412

[78] **Browne, J., Dubois, K , Rathmill, S P Sethi, and Steck, K. E.** "Classification of Flexible Manufacturing System " The FMS Magazine, April, 1984, pp 114,117

[79] **Hanifin, L. E., Liberty, S. G. and Taraman, K.** "Improved Transfer Lines Efficiency Utilizing System Simulation " Technical Paper MR, Society of Manufacturing Engineers, Dearborn, Mich , 1975, pp 75-169

[80] **Buzacotte, et al.** "Transfer Line Design and Analysis-An Overview", A I I E FALL Conf Atlanta GA, 1978, pp 277-286

[81] **Koenigsberg, E. and Mamer, J.,** "The analysis of production processes " School of Business Administration, Univ of California, Berkely, 1981

[82] **Buzacott, J. A. and Yao, David, D. W.** "Flexible Manufacturing Systems A Review of Analytical Models " Working paper #82-007, Dept of Industrial Engng , Univ of Toronto, ORSA/TIMS meeting in Detroit in April 1982

[83] **Buzacott, J. A. and Yao, David, D. W.** "Flexible Manufacturing Systems A Review of Analytical Models " Mgmt Sci Vol 32 No 7, July 1986, pp 890-905 (An earlier version of this paper was [73])

[84] **Riley, F. and Yarrow, E. P.** "A new approach to Assembly Machine Justification " Proc 2_{nd} European Conf Automated

Manufacturing, Birmingham, UK , 1983

[85] **Lay, K. and Schiefele, M.** "Simulation of a flexible Assembly System " 1_{st} International Conf on Simulation in Manufacturing, Stratford-Upon-Avon,UK, 5-7 March 1985, pp 141-149

[86] **O'Gorman, P., Gibbons, J and Browne, J** "Evaluation of Scheduling System for A Flexible Transfer Line Using A Simulation Model " FMSs Methods and Studies, Edited by A Kusiak, Elsevier Sci Pub B V (North-Holland),1986

[87] **Buzacott, J. A. and Shanthikumar, J. G.** "Models for Understanding Flexible Manufacturing Systems " AIIE Trans , 12, 4, Dec 1980, pp 339-350

[88] **Ranky, P. G.** "Computer Integrated Manufacturing Introduction with case studies " Englewood Cliffs, NJ Prendice-Hall,International,1986

[89] **Greenwood, Nigel R.** "Implementing Flexible Manufacturing System " Macmillan Education Ltd, 1988

[90] **Solberg, J. J.** "A Mathematical Model of Computerized Manufacturing Systems " Proc 4_{th} Int Conf Production Res Tokyo, Japan, 1977

[91] **Hildebrant, R. R** "Scheduling Flexible Manufacturing System When Machines Are Prone to Failure " Ph D Dissertation, Dept of Aeronautics and astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1980

[92] **Gershwin, S. B., Athans, M. and Ward, J. E.** "Complex Materials Handling and Assembly Systems " Proc 8_{th} NSF Grantees' Conf Production Res and Technology, 1981

[93] **Kimemia, J. G. and Gershwin, S. B.** "Flow Optimization in Flexible Manufacturing Systems " Internat J Production Res , 22, 1985, pp 81-96

[94] **Ho, Y. C.** "Perturbation Analysis of Discrete Event Dynamic Systems " Proc 1_1 ORSA/TIMS Conf FMS, Ann Arbor, Michigan, 1984

[95] **Cavaille, J. B. and Dubois, D.** "Heuristic Method Based on Mean Value Analysis for Flexible Manufacturing Systems Performance Evaluation " Proc 21_{st} IEEE Conf Decision and

Control, Orlando, FL, 1982

[96] Sevcik, K. C. and Mitrani, I. "The Distribution of Queuing Network States at Input and Output Instants," J Assoc Comput March ,28, 1981, pp 358-371

[97] Buzacott, J. A. " 'Optimal' operating rules for automated manufacturing Systems " IEEE Trans on Automatic Control, Vol AC_27, No 1, 1 982a,pp 80-86

[98] Yao, D. D. and Buzacott, J A. "The Exponentialization Approach in Flexible Manufacturing Systems models with General Processing Times " Eruopian J Opr Res , 24, 1986a, pp 410-416

[99] Aanen, E., Gaalman, G. J. and Nawijn, W M. "Planning and Scheduling in an FMS " Engng Costs and Production Economics, Vol 17, No, 1-4 aug 1989, pp 89-97

[100] Niemi, E. and Davies, B J. "Simulation of an optimizing FMS-cell control System " Robotics and Computer-Integrated Manufacturing, Vol 5, No 2-3 1989, pp 229-234

[101] Montazeri, M. and Van Wassenhove, L. N. "Analysis of Scheduling rules for an FMS " Internat J of Production Res , Vol 28, No 4, Apr 1990, pp 785-802

[102] Muller Daniel, J., Jackman, J. K. and Fitzwater, C "A Simulation-Based work order release mechanism for a flexible manufacturing system " 90 Winter Simulation Conf Publ by IEEE Service centre, Piscateway, NJ, USA 1990, pp 599-602

[103] Emelyanov, S. V., Gendler, M. B. and Felman, D. P "Flexibility of Manufacturing Systems Concept, Kinds, Indicators 'A Survey' "Problems of Control and Information Theory, Vol 19, 1990, pp 615-180

[104] Boer, C. R. and Metzler, V. "Simulation for economic evaluation of advanced manufacturing " (BBC Brown Boveri & Co Ltd, Switzerland), 1_{st} Internat Conf on Simulation in Manufacturing, 5-7 March 1985, pp 251-260

[105] Pritsker, A. A. B., "The GASP-IV Simulation Language, John Wiley, New york, 1974

[106] Pritsker, A. Alan B., "Introduction to Simulation and SLAM II, A Halsted Press Book, John Wiley & Son, 1986

[107] **WITNESS User's Manual**,"Istel Ltd,Redditch,England, 1986"

[108] **Clementson, A. T.**,"ECSL User's Manual, Cle-Com Ltd, Birmingham, 1985"

[109] **Schriber, T. J.** ,"Simulation Using GPSS", John Wiley & Sons, Inc , NY, NY 1974 Or Minuteman Software 1988

[110] **HOCUS User Manual**,"P-E Information Systems,Egham,England"

[111] **BEAM User Manual**,"CMS Research Inc ,Oshkosh, Wisconsin, USA"

[112] **Horrion, R. D.,** "An investigation of Visual interactive simulation methods using the job shop scheduling problem," J of the Operational Res Society, 29, 1978, pp 1085-1093

[113] **Grant, J. W. and Weiner, S. A.**, "Factors to consider in choosing a graphically Animated Simulation system," Industrial Engineering ,Aug 1986, p36

[114] **Pidd, M.,** "Computer Simulation in Management Science " John Wiley & Sons, Second Edition 1988 "

[115] **Clark, Wallace,** (His work was originally transferred from Henry Gantt), Sir Isaac Pitman & Sons Ltd

[116] **Mellor, P.** "A Review of Job Shop Sequencing", Operational research Quarterly, 1966 V 17 No 2, pp 161-171

[117] **King, J. R** "Production Planning and Control, An Introduction To Quantitative Methods", Pergamon International Library Of Science, Technology, Engineering and Social Studies , Publisher Robert Maxwell, M C , 1975

[118] **Heller, J.** "Some Numerical Experiments for an M X J Flow Shop and its Decision Theoretical Aspects,"Operational Research 8, No 2 march 1960

[119] **Al-Qattan, P. E. et al.** "Systematic Approach to Cellular Manufacturing System Design" Elsvier Science Publishers B V 1989, pp 415-424

[120] **Gere, J. W.** "Heuristic in Job Shop Scheduling," Management Sci 13, 1966, pp 167-190

[121] **Thompson,G. L.** "Recent Development in the Job shop Scheduling Problem" Nav Res Logist Q 7,1960, p 585

[122] **Hines, W. W and Montgomery, D C.** "Probability and

Statistic in Engineering and Management Science,"John Wiley & Sons,1980

[123] **Hollier, R. H.** "A Simulation Study of Sequencing in Batch Production", Opnl Res Quart , V 19, 1968, pp 389-407

[124] **Andrew, F. Selia,** "Output Analysis for Simulation", The Winter Simulation Conference 1990, ISBN 0-911801-72-3, pp 49-54

[125] **Welch, P. D.** "The Statistical Analysis of the Simulation Results," The Computer Performance Modelling Handbook, S S Lavenberg,ed , pp 268-328, Academic Press,1983 NY USA

[126] **Law, A. M. and Kelton, W D.** "Simulation Modelling and Analysis, Second Edition, McGraw-Hill, 1991, NY USA

[127] **Li-Yen Shue,** "Scheduling Study of A FMS System", Advances in Computer Science Application to Machinery, the International Conference on CAD of Machinery, Beijing China, Sep 16-20, 1991, pp 440-445

APPENDIX

APPENDIX A:

"Throughput Time Formula For A Schedule"

We will consider that n X m matrix is the problem of n jobs numbered 1,2,3, n to be processed on m machines, numbered 1,2,3, ,m in the same order, that $x = x(j,c)$ is an n X m matrix in which the cth column contains a sequence of permutation, $q(1,c)$, $q(2,c)$, $q(3,c)$, , $q(n,c)$ of jobs numbers 1,2,3, ,n, where $q(j,c)$ is the number of the job which is put j_{th} in sequence on the cth machine

The following notations will be used

> $t = t(j,i)$ is an n X m matrix describing the processing times for job j on machine i
>
> $f\{q(j,i),i\}$ = expected completion time of job number $q(j,i)$ on machine i.

Then, the earliest time at which the processing time of job $q(j,i)$ the jth job on the ith machine can be completed, i e the notation $f\{q(j,i),i\}$ is governed by either the time at which machine i becomes available to accept the job or the time which the job completes processing on the machine i-1

Time at which machine i is ready to receive job $q(j,i) = F\{q(j-1,),i\}$ the job at which job $q(j-1,i)$, the one prior in sequence on machine j, is completed

In another hand, time at which job $q(j,i)$ is available to be sequence to machine $i = f\{q(j,i),i-1\}$ the time that job $q(j,i)$ is completed on the previous machine i-1

Thus the expected completion time of job $q(j,i)$ on machine i is equal to

$$f\{q(j,i),i\} = \max[f\{q(j-1,i),i\}, f\{q(j,i),i-1\}] + t\{q(j,i),i\}$$

Hence the throughput time to complete the total schedule C_{max} is give by (The application problem is stated on pages (53-56))

$$C_{max} = F\{q(n,m),m\} = \max[f\{q(n-1,m),m\}, f\{q(n,m),m-1\}] + t\{q(n,m),m\}$$

APPENDIX B:

/* The Make-Span calculation programme:*/

```
#include <stdio h>
#include <stdlib h>
#include <conio h>
int  T[90][90],f[90][90],temp[90][90],
int  i,j,n,m,number,x,index[90],
char c,
FILE    *fp,*fp2,
main()
{
     fp = fopen("in dat","r"),
     fp2 = fopen("out dat","w"),
     printf(" \nEnter Number of Jobs      "),
     scanf("%d",&n),
     fprintf(fp2,"Number of jobs  = %d\n",n),
     printf(" \nEnter Number of machines      "),
     scanf("%d",&m),
     fprintf(fp2,"Number of machines  = %d\n",m),
     fprintf(fp2,"\n"),
     fprintf(fp2,"The results for the first sequence of the
     jobs  1 2 3 4 5 6 7    n\n"),

      /* Read in data in matrix ABCD      */

     for ( i=1,i<m+1,i++)
     {
          for( j=1,j<n+1,j++)
          {
               fscanf(fp,"%d",&number),
               T[j][i] = number,
               temp[j][i] = number,
          }
     }

label     f[1][1] = T[1][1],

     for ( i=2, i<m+1,i++)
     {
          f[1][i] = f[1][i-1] + T[1][i],
     }

     for ( i=2, i<n+1,i++)
     {
          f[i][1] = f[i-1][1] + T[i][1],
     }
     for ( j = 2, j<m+1,j++)
     {
          for ( i=2,i<n+1,i++)
          {
                f[i][j] = max( f[i-1][j], f[i][j-1] ) +
                          T[i][j],
```

Appendix B continued:

```
		}
	}

	/*		Display Result		*/
	system("cls");
	printf("\n");
	i =m;

		for ( j=1;j<n;j++)
		{
			printf(" Completion Time for Job %i",j);
			printf(" = %i\n",f[j][i]);
		}
		printf(" The Make-span or Completion Time for the
				last job = %i\n",f[n][i]);
		for ( j=1;j<n;j++)
		{
			fprintf(fp2," Completion Time for Job %i",j);
			fprintf(fp2," = %i\n",f[j][i]);
		}
		fprintf(fp2,"The Make-span or Completion Time for the
				Last Job = %i\n",f[n][i]);
		fprintf(fp2,"\n");

	/* Option to rearrange matrix */

	printf(" Do you want another run ?  y/n \n");
	scanf("%c",&c);
	if ( c == 'y' )
	{
		printf(" Enter new matrix \n ");
	/*fprintf(fp2,"\n");*/
		fprintf(fp2," New sequence  \n");
		for ( i = 1;i<n+1;i++)
		{
			scanf("%i",&x);
			fprintf(fp2," %i",x);
			fprintf(fp2,"\n");
			index[i] = x;
		}
	for ( i=1;i<n+1;i++)
	{
		for( j=1;j<m+1;j++)
		{
			T[i][j] = temp[index[i]][j];
	/*		fprintf(fp2,"\n");*/
		}
	}
	goto label;
	}
}
```

APPENDIX C:

" A complete matrix processing time for problem 2 in Pages 58-59".

9	2	6	7	6	8	10	0	0	6	0	8	5	2	8	10	10	0	0	0
9	7	7	3	9	1	2	8	0	6	2	1	2	9	2	1	4	3	1	6
7	3	2	10	1	2	9	0	8	1	8	3	7	3	3	10	2	9	9	8
6	6	9	0	9	3	10	4	4	6	2	1	4	5	7	1	8	9	6	3
7	4	3	9	2	3	10	9	9	5	5	0	0	7	2	0	10	6	6	6
10	0	2	5	6	2	7	8	7	4	5	2	4	7	1	7	9	9	3	2
2	2	3	10	10	9	5	4	6	9	4	1	2	7	0	6	4	1	5	10
5	5	6	0	3	6	10	8	2	10	0	3	4	9	10	4	4	10	8	6
7	9	8	1	7	4	2	0	10	9	7	3	6	4	10	5	0	8	10	6
7	6	8	6	5	1	8	10	4	7	8	4	3	4	2	4	10	2	0	10
3	6	10	2	5	2	9	2	1	6	0	1	8	3	7	5	7	7	0	5
3	5	10	3	5	10	8	2	7	4	4	3	8	10	5	3	3	10	0	7
9	0	0	9	5	7	4	9	0	0	7	8	3	10	6	10	1	10	2	5
7	7	3	4	2	8	5	2	7	3	2	8	9	6	4	2	7	1	7	1
5	2	10	10	0	1	4	4	2	6	10	9	1	3	0	3	7	1	3	9
8	0	4	5	4	8	10	0	2	9	9	1	3	7	7	0	0	1	1	6
6	0	6	4	10	4	9	4	4	9	7	0	10	5	7	8	9	6	9	7
6	0	9	0	2	2	3	3	0	9	0	5	0	0	6	7	4	0	5	2
0	5	4	9	1	1	8	6	8	5	4	10	9	4	6	4	3	6	1	8
2	2	3	0	4	6	1	0	1	10	7	2	8	4	7	1	0	1	0	3
1	1	2	7	5	9	3	7	5	4	1	1	2	6	5	9	6	2	1	6
4	9	8	1	4	2	9	3	4	5	6	2	4	4	0	10	4	1	2	8
7	4	10	6	3	5	3	10	2	4	0	5	8	4	6	5	7	9	3	4
9	0	2	4	9	9	7	6	1	5	5	0	7	6	0	10	1	4	8	10
2	2	5	1	6	8	1	3	8	4	10	1	0	5	7	6	6	2	5	0
3	1	4	10	4	4	4	6	7	0	10	2	1	2	1	10	2	3	10	3
2	8	7	4	4	6	2	2	10	4	7	0	3	5	6	8	6	9	0	5
7	9	4	3	5	10	1	3	8	0	10	6	10	6	8	3	7	4	7	10
9	0	2	4	1	4	10	1	10	8	2	4	1	4	6	0	2	8	7	0
10	10	2	2	7	4	5	0	2	4	10	6	7	7	6	1	9	7	3	7
3	7	6	2	5	3	5	1	8	8	1	8	6	8	3	3	1	8	7	6
9	7	4	8	3	5	1	1	3	10	0	2	4	6	9	10	8	2	0	5
8	10	4	2	4	9	8	2	9	5	10	8	7	10	1	8	10	10	10	6
7	9	6	4	9	4	10	8	6	4	5	1	8	3	0	3	10	0	6	4
8	2	6	4	4	3	2	3	6	9	5	4	6	7	7	2	2	8	2	9
5	1	4	0	1	4	5	1	10	3	4	2	3	4	8	4	2	0	0	5
0	1	8	3	3	7	0	8	3	4	8	7	2	4	5	5	4	10	10	7
1	10	9	1	2	0	1	2	8	10	0	9	6	6	5	4	3	9	1	0
5	6	4	3	5	0	6	6	3	0	9	1	10	8	2	0	10	6	2	7
0	0	3	3	9	9	8	3	8	0	9	4	6	10	9	6	6	9	0	8
8	4	1	7	2	7	3	5	10	7	0	3	6	6	5	6	0	1	2	3
2	0	3	10	8	0	3	9	3	3	10	5	6	6	8	9	1	4	10	5
0	4	8	2	6	6	0	9	7	9	2	4	8	5	7	1	8	4	10	0
1	0	1	3	10	10	8	10	5	3	4	9	3	1	10	3	4	10	2	10
10	9	6	1	8	7	5	6	0	4	3	6	5	0	1	7	1	10	8	8
6	9	10	6	5	7	10	8	2	2	9	4	4	5	6	5	2	10	5	8
9	10	0	6	8	6	3	7	6	4	3	5	10	1	1	9	9	10	6	6

Appendix C continued:

2	10	8	0	10	1	1	2	6	5	6	1	8	5	2	9	7	6	7	7
4	6	4	4	10	2	8	10	8	4	1	2	3	5	0	8	3	7	10	5
8	1	0	9	0	4	8	8	2	8	7	10	10	0	9	9	1	10	0	0
7	10	7	7	0	1	10	5	2	4	6	4	0	6	10	5	0	7	3	0
0	8	0	4	2	7	1	1	6	1	8	8	8	4	7	9	1	10	2	3
3	9	1	2	5	8	2	1	0	6	2	3	2	2	8	0	7	8	8	10
9	3	10	3	4	6	9	10	0	8	3	3	2	0	5	1	0	10	5	9
4	8	4	7	4	3	8	1	2	8	5	8	10	9	8	5	10	0	2	9
4	3	5	8	6	0	2	8	8	6	9	6	2	1	9	7	4	1	6	10
0	7	1	5	5	10	4	10	4	10	8	0	5	1	7	0	0	7	6	3
9	6	4	9	9	1	9	7	9	9	6	9	2	8	0	6	4	8	7	3
0	9	0	5	0	9	6	4	2	6	7	7	10	3	9	7	3	3	9	5
8	2	8	5	8	7	5	3	1	2	3	8	1	5	5	0	3	5	6	6
5	3	4	3	1	5	7	7	9	7	6	8	4	0	7	1	8	2	1	2
6	2	5	10	7	0	9	9	8	5	7	1	3	3	9	2	3	9	10	4
6	6	9	3	8	7	0	6	0	10	9	10	0	1	4	1	8	1	1	9
7	5	8	8	2	7	8	0	3	8	7	2	5	7	8	10	2	7	0	7
8	3	6	2	6	7	5	4	5	5	7	3	7	4	5	2	8	8	4	10
0	9	2	1	9	0	7	8	4	1	4	1	7	2	6	5	3	0	4	0
10	4	9	5	7	10	9	4	3	1	9	9	3	1	1	8	3	0	2	8
8	3	8	0	8	3	5	10	1	4	1	3	0	3	3	9	3	5	0	4
10	8	2	1	7	10	2	10	3	3	1	4	1	9	5	3	7	0	9	3
2	8	8	7	10	1	1	8	7	5	7	1	5	0	9	7	2	0	4	9
4	10	4	5	7	7	8	0	5	5	7	0	5	10	9	8	4	1	1	10
10	1	0	10	9	7	1	2	3	1	1	4	8	4	2	8	10	5	1	7
0	10	2	7	9	6	7	1	2	0	7	9	3	3	5	1	4	7	10	9
10	10	6	9	2	5	0	5	4	5	2	3	9	3	6	0	2	4	3	4
6	4	6	5	1	0	8	9	6	4	2	6	5	0	6	5	5	1	4	5
5	2	2	5	7	5	5	10	6	6	5	5	1	8	3	6	2	9	4	7
4	7	2	4	8	3	8	5	7	9	4	10	4	7	1	6	4	8	7	2
6	4	7	3	8	8	10	1	10	5	8	2	0	4	2	7	2	4	5	5
4	4	9	1	7	8	4	3	5	0	3	1	3	10	2	8	1	1	7	10
1	2	8	10	7	2	9	7	5	10	9	10	5	10	4	6	3	4	4	3
7	5	4	5	2	2	9	0	2	3	10	6	0	4	2	9	5	7	10	7
0	10	6	6	5	9	2	3	9	3	5	5	6	3	6	8	5	8	3	3
5	9	3	8	5	0	5	7	6	10	8	4	5	9	2	10	5	10	3	1
3	9	3	5	9	3	5	0	4	2	9	9	9	10	7	4	6	1	8	2
0	6	4	2	0	4	0	1	9	6	5	5	1	5	3	6	9	8	0	10
0	2	8	0	9	5	1	3	0	3	0	6	8	2	4	2	5	10	9	10
4	0	0	5	2	3	9	5	1	0	2	6	7	8	6	3	1	6	10	4
1	3	8	10	8	9	6	7	9	8	3	3	6	2	2	4	0	5	8	7
4	9	4	0	4	2	0	2	3	1	8	1	3	5	2	4	0	3	8	8
9	6	10	3	1	3	9	5	4	9	8	2	8	4	2	6	3	5	2	5

APPENDIX D:

/* The Optimum Make-Span Programme Listing */

```
#include <conio h>
#include <stdio h>
#include <alloc h>
#include <stdlib h>
#include <time h>
#define RANGE 11
#define MAX 32767
typedef struct NODE{
          long num,
          long  val,
          struct NODE *next,
                              }NODE,
NODE *make_NODE(void),
NODE *link_data(long),
void del_NODE(NODE *),
void insert_NODE(NODE * , NODE *),
void remove_NODE(NODE *),
void dump_list(NODE *),
void perm2(int *,int ),
void store(void),
void ran(),
void readdata( void ),
NODE *head,

#define OFF 0
#define ON  1

void process(int n,int m),
int  T[90][90],f[90][90],temp[90][90],
int  i,j,n,m,number,x,index[90],seq[90],maximum=10000,
long count=0,
int     num,
int amount,column,
long now,
long int yy,runs,y,ave,average,result,
int output=ON,
char c,
FILE    *fp,*fp2,

void insert_NODE(NODE *at, NODE * ins)
{

ins->next=at->next,
at->next=ins,
}
void remove_NODE(NODE *node)
{
/* removes the next node down */
NODE *old,
```

Appendix. D continued:

```
old=node->next,
node->next=node->next->next,
free(old),
}

NODE *make_NODE()
{
NODE *ret,

if((ret=(NODE *)malloc(sizeof(NODE)))==NULL)
          {
          printf("Out of memory program halted\n"),
          exit(0),
          }
return(ret),
}

void del_NODE(NODE *node)
{
free(node),
}

NODE * link_data(long data)
{
static int flag=0,
static NODE *head,
NODE * new_NODE,
NODE *current,
NODE *old,

if(flag==0){
                 head=make_NODE(),
                 head->val=0,
                 head->num=0,
                 head->next=NULL,
                 flag=1,
                 }

old=head,
current=head->next,
while(current){
          if(data<current->val)
                    {
                    new_NODE=make_NODE(),
                    new_NODE->val=data,
                    new_NODE->num=1,
                    insert_NODE(old,new_NODE),
                    return(head),
                    }
          if(data==current->val)
                    {
```

Appendix D continued:

```
                    current->num++,
                    return(head),
                    }
          if(data>current->val)
                    {
                    old=current,
                    current=current->next,
                    }
     }
                    new_NODE=make_NODE(),
                    new_NODE->val=data,
                    new_NODE->num=1,
                    insert_NODE(old,new_NODE),

return(head),
}

void dump_list(NODE *head)
{
NODE *current,
FILE *out,

out=fopen("out dat","w+"),
current=head->next,
while(current)
{
if(output==ON){
     printf("For the Make-Span = %ld   The Frequency = %ld
     \n",current->val,current->num),
     fprintf(fp2, "For the Make-Span = %ld   The Frequency =
     %ld \n",current->val,current->num),
          }

     fprintf(out,"%ld\t\t%ld \n",current->val,current->num),
     current=current->next,
}
fclose(out),
printf("\n"),
for(i=0,i<n,i++)
{
fprintf(fp2,"%d ",seq[i]),
}
fprintf(fp2," = Optimum Job Sequence, Optimal Make-Span =
     %d",maximum),
printf(" Problem matrix size  "),printf("( %d X %d ) \n",n,m),
}

main( int argc,char *argv[])

{
```

Appendix D continued:

```
     {
unsigned seed,
     if(*argv[1]=='n') output=OFF,
     fp2 = fopen("out dat","w+"),
     printf("   \nEnter   number   of   Jobs                ")/*   =
               %d\n",n),*/
     scanf("%d",&n),
     printf(" Number of Jobs      = %d\n",n),
     printf(" \nEnter Number of machines     "),
     scanf("%d",&m),
     printf(" Number of machines = %d\n",m),
     printf(" \nEnter number of runs           "),
     scanf("%d",&runs),
     printf(" Number of runs      = %d\n",yy),
     column = n,
     printf(" \nEnter random seed              "),
     scanf("%d",&seed),
     printf(" Random seed         = %d\n",seed),
     printf("\n"),
     srand(seed),
     {
     clock_t start, end,

     start=clock(),

     /*      Read in data in matrix (n X m) problem        */

      for(yy=0,yy<runs,yy++)
      {
      maximum=10000,
      ran(),
      readdata(),
/*    fprintf(fp2,"      \n Optimal job sequence problem " ),
      fprintf(fp2," \n\n"),*/
      fprintf(fp2," %d Jobs, %d Machines, Random seed = %d
                  \n",n,m,seed),
      fprintf(fp2,"   Number of runs                    = %d
            \n",yy),
      for(i=0,i<n,i++) index[i]=i+1,
      perm2(index,n),
      printf("Number of Job Sequences      =  %ld\n",count),
      fprintf(fp2,"   Number of Job Sequences      =
      %ld\n",count),
      fprintf(fp2,"\n"),
      for(i=0,i<n,i++)
      {
      printf("%d ",seq[i]),
      fprintf(fp2,"%d",seq[i]),

      }

     printf(" = Optimum Job Sequence, Optimal Make-span =
```

Appendix D continued:

```
            %d\n",maximum),
        fprintf(fp2," = Optimum Job Sequence , Optimal Make-span
                = %d\n",maximum),
        result = maximum,
        ave=ave+result,
/*      system("cls"),
        fprintf(fp2,"\n"),
        fprintf(fp2,"%d Jobs, %d Machines, the Pure Flow-Shop,
              (Number of runs) = %d \n",n,m,yy),
        fprintf(fp2,"\n"),
        average = ave/runs,
/*      printf(" Average Optimal Make-Span = %d \n",average),
        fprintf(fp2,"Average Optimal Make-Span = %d \n",average),
*/
        end = clock(),
        printf(" CPU time in seconds    = %f \n", (end-start) /
              CLK_TCK),
        printf("   \n"),
/*      fprintf(fp2,"       Make-Span's Frequencies   "),
                printf("( %d X %d ) \n\n",n,m),*/
/*      fprintf(fp2,"\n"), */
        fprintf(fp2,"CPU time in seconds = %f \n", (end-start) /
                CLK_TCK),
        fprintf(fp2," \n"),
        fprintf(fp2,"       Make-Span's Frequencies   "),
        fprintf(fp2,"\n\n"),
/*      fprintf(fp2,"  Problem matrix size"),  fprintf(fp2,"( %d
                X  %d ) \n\n",n,m), */
        dump_list(head),

         }
         }
         }
}
void process(int n,int  m)
{
/* The Make-Span Formula used */
int i,j,
        f[1][1] = T[1][1],

        for ( i=2, i<m+1,i++)
        {
             f[1][i] = f[1][i-1] + T[1][i],
        }

        for ( i=2, i<n+1,i++)
        {
             f[i][1] = f[i-1][1] + T[i][1],
        }

        for ( j = 2, j<m+1,j++)
```

Appendix D continued:

```
	{
		for ( i=2,i<n+1,i++)
		{
			f[i][j] = max( f[i-1][j], f[i][j-1] ) +
					T[i][j],
		}
	}
return,
}
void store(void)
{
static int co=0,
count++,

	for(i=1,i<=n,i++)
		{
		for(j=1,j<=m,j++)
			{
			T[i][j]=temp[index[i-1]][j],
			}
		}

	process(n,m),

	if(output){
		for(i=0,i<n,i++)
			{
			printf("%d ",index[i]),
			fprintf(fp2,"%d ",index[i]),
			}

	 printf("Make-Span = %d\n",f[n][m]),
	 fprintf(fp2,"Make-Span = %d\n",f[n][m]),
		head=link_data((long)f[n][m]),
	}

	if(maximum > f[n][m])
		{
		maximum=f[n][m],
		for(i=0,i<n,i++)
			{
			seq[i]=index[i],

			}
		}
}

void perm2(int *s,int n)
{
int i,
int tmp,
```

Appendix D continued:

```
          if (n == 1)
          store(),
               for (i=0 , i < n , ++i)
                    {
                    tmp = s[0],
                    s[0] = s[i],
                    s[i] = tmp,
                    perm2(&s[1],n-1),
                    tmp = s[0],
                    s[0] = s[i],
                    s[i] = tmp,
                    }
}

  void ran( )
{
FILE *fp,
      fp = fopen("in dat","w"),
      amount = n*m,
      count = 0,
      for(i=0,i<amount,i++)
      {
          y=random(11),
          fprintf(fp,"%-3i",y),
          count++,
          if (count == column)
          {
               fprintf(fp,"\n"),
               count = 0,
               }
               }
 fclose(fp),

}
void readdata( void )
 {
 FILE *fp,
     fp = fopen("in dat","r"),
     for ( i=1,i<m+1,i++)
     {
          for( j=1,j<n+1,j++)
          {
               fscanf(fp,"%d",&number),
               T[j][i] = number,
               temp[j][i] = number,
          }
     }
     fclose(fp),
}
```

xxxxxxx End the Programme xxxxxx

APPENDIX E:

/* Effect of the priority rules on the criteria programme */

```
#include <stdio h>
#include <stdlib h>
#include <conio h>
#include <string h>
#include <time h>
#define  RANGE 10
#define  MAX 32767

int intcmp(int a,int b),
void process( void ),
void rule2( void ),
void rule3( void ),
void display ( void ),
void readdata ( void ),
void rule4 ( void ),
void rule5 ( void ),
void sort4( void ),
void ran( ),
int  T[90][90],f[90][90],temp[90][90],
int  i,j,k,l,n,m,number,
char x,
int  index[90], store[90], store2[90], sum[90], store3[90],
     p[90], z[90], su[90],
char c,
int rulenum,
int amount,count,y,column,
long now,
double num,
int yy,runs,
long int idle, idle1, result6, result7, av1, av2, av3, av4,
         av5,av,result8,
long int a,a1,a2,a3,a4,a5,
long int average, average1, average2, average3, average4,
           average5, result,
long int result1, result2, ave, ave1, ave2, ave3, ave4, ave5,
          result3, result4, result5,
FILE    *fp,*fp2,
main()
{
     {
unsigned seed,
     fp2 = fopen("out dat","w+"),
     printf(" \nEnter Number of Jobs      "),
     scanf("%d",&n),
     printf(" \nEnter Number of machines      "),
     scanf("%d",&m),
     printf("\nEnter number of runs      "),
     scanf("%d",&runs);
     printf("\nEnter random seed   "),
```

Appendix E continued:

```
      scanf("%d",&seed);
      srand(seed);
      column = n;
       {
      clock_t start, end;
      start = clock();

            /* Read in data in matrix ABCD      */

/*    printf(" %d Jobs, %d Machines \n\n",n,m); */

        for ( yy=0;yy<runs;yy++)
        {
      printf(" MAKESPAN RUN  %d \n",yy);

        ran();
        rulenum =1;
        readdata();
        process();
        display();
        average1 = average1+ result;
        ave1 = ave1 + result2;
        av1 = av1 + result8;
        a1 = a1 + result5;
        rule2();
        display();
        average2 = average2 + result;
        ave2 = ave2 + result2;
        av2 = av2 + result8;
        a2 = a2 + result5;
        readdata();
        process();
        rule3();
        display();
        average3 = average3 + result;
        ave3 = ave3 + result2;
        av3 = av3 + result8;
        a3 = a3 + result5;
        readdata();
        process();
        rule4();
        display();
        average4 = average4 + result;
        ave4 = ave4 + result2;
        av4 = av4 + result8;
        a4 = a4 + result5;
        readdata();
        process();
        rule5();
        display();
        average5 = average5 + result;
```

Appendix E continued:

```
        ave5 = ave5 + result2,
        av5 = av5 + result8,
        a5 = a5+ result5,
        readdata(),
        process(),
        }    /* end of yy loop */
        system("cls"),
        printf("\n"),
        fprintf(fp2,"\n"),
        fprintf(fp2," %d Jobs, %d Machines, Random seed = %d,
                 (Number of runs) = %d \n\n",n,m,seed,yy),
    /*     printf(" %d Jobs, %d Machines \n\n",n,m),     */
        fprintf(fp2," Average MAKE-SPAN   \n\n"),
    /*  printf("Average MAKESPAN  (Number of runs) = %d
                 \n\n",yy),    */
    /*   printf("\n\n"), */
        average = average1/runs,
    /*     printf(" Rule FCFS     = %d \n",average),      */
        fprintf(fp2," Rule 1 FCFS     = %d \n\n",average),
        average = average2/runs,
    /*    printf(" Rule SPT(SI) = %d \n",average),       */
        fprintf(fp2," Rule 2 SPT(SI) = %d \n\n",average),
        average = average3/runs,
    /*   printf(" Rule LPT(LI) = %d \n",average),        */
        fprintf(fp2," Rule 3 LPT(LI) = %d \n\n",average),
        average = average4/runs,
    /*   printf(" Rule SRPT     = %d \n",average),       */
        fprintf(fp2," Rule 4 SRPT     = %d \n\n",average),
        average = average5/runs,
    /*   printf(" Rule LRPT     = %d \n",average),       */
        fprintf(fp2," Rule 5 LRPT     = %d \n\n",average),
    /*   end = clock(),
        fprintf(fp2,"\n"),
        fprintf(fp2,"Computation time in seconds = %f
                  \n",(end-start) / CLK_TCK), */
        fprintf(fp2,"\n"),
    /*  fprintf(fp2,"problem matrix size  ( %d X %d )
                     \n",n,m),
        printf("\n"),
        fprintf(fp2,"\n"),       */
        fprintf(fp2,"Average MEAN COMPLETION TIME  \n\n"),
    /*   printf("Average MEAN COMPLETION TIME    RUNS   %d
                  \n\n",yy), */
        printf("\n\n"),
        ave = ave1/runs,
    /*   printf(" Rule 1 FCFS       = %d \n\n",ave),      */
        fprintf(fp2," Rule 1 FCFS      = %d \n\n",ave),
        ave = ave2/runs,
    /*   printf(" Rule 2 SPT(SI)    = %d \n\n",ave),      */
        fprintf(fp2," Rule 2 SPT(SI)   = %d \n\n",ave),
        ave = ave3/runs,
```

Appendix E continued:

```
/*   printf(" Rule 3 LPT(LI)    = %d \n\n",ave),          */
     fprintf(fp2," Rule 3 LPT(LI)   = %d \n\n",ave),
     ave = ave4/runs,
/*   printf(" Rule 4 SRPT        = %d \n\n",ave),          */
     fprintf(fp2," Rule 4 SRPT       = %d \n\n",ave),
     ave = ave5/runs,
/*   printf(" Rule 5 LRPT        = %d \n\n",ave),          */
     fprintf(fp2," Rule 5 LRPT       = %d \n\n",ave),
/*   printf("\n"),                                          */
     fprintf(fp2,"\n"),
/*   printf("Average Total Waiting Time Runs %d \n\n",yy),*/
     fprintf(fp2,"Average Total Waiting Time    \n\n"),
/*   printf("\n\n"),                                         */
     av = av1/runs,
/*   printf("Rule 1 FCFS         = %d \n\n",av),             */
     fprintf(fp2,"Rule 1 FCFS        = %d \n\n",av),
     av = av2/runs,
/*   printf("Rule 2 SPT          = %d \n\n",av),             */
     fprintf(fp2,"Rule 2 SPT         = %d \n\n",av),
     av = av3/runs,
/*   printf("Rule 3 LPT          = %d \n\n",av),              */
     fprintf(fp2,"Rule 3 LPT         = %d \n\n",av),
     av = av4/runs,
/*   printf("Rule 4 SRPT         = %d \n\n",av),              */
     fprintf(fp2,"Rule 4 SRPT        = %d \n\n",av),
     av = av5/runs,
/*   printf("Rule 5 LRPT         = %d \n\n",av),              */
     fprintf(fp2,"Rule 5 LRPT        = %d \n\n",av),
     fprintf(fp2,"\n"),
/*   printf("\n"),                                            */
     fprintf(fp2,"Average Total Idle Time    \n\n"),
     fprintf(fp2,"\n\n"),
     a = a1/runs,
     fprintf(fp2,"Rule 1 FCFS          = %d \n\n",a),
     a = a2/runs,
     fprintf(fp2,"Rule 2 SPT           = %d \n\n",a),
     a = a3/runs,
     fprintf(fp2,"Rule 3 LPT           = %d \n\n",a),
     a = a4/runs,
     fprintf(fp2,"Rule 4 SRPT          = %d \n\n",a),
     a = a5/runs,
     fprintf(fp2,"Rule 5 LRPT          = %d \n\n",a),
     end = clock(),
     fprintf(fp2,"Computation time in seconds = %f \n", (end
             - start) / CLK_TCK),

                         /*  end of main  */
}}
}
void ran( void )
{
```

Appendix E continued:

```
     fp = fopen("in dat","w"),
     amount = n*m,
     count =0;
     for( i=0,i<amount,i++)
     {
          y=random(11),
          count++,
          fprintf(fp,"%-3i",y),
          if ( count == column)
          {
               fprintf(fp,"\n"),
               count = 0,
          }
     }
fclose(fp),
}

void readdata( void )
{
     fp= fopen("in dat","r"),
     for ( i=0,i<m,i++)
     {
          for( j=0,j<n,j++)
          {
               fscanf(fp,"%d",&number),
               T[j][i] = number,
               temp[j][i] = number,
          }
     }
     fclose(fp),
}

void rule4( void )
{
int var,

     for ( i=0,i<n+m,i++)
     {
          sum[i] =0,
          index[i] =0,
     }
     rulenum  = 4,
     for(j=0,j<n,j++)
     {
          for(i=0,i<m,i++)
          {
          sum[j]   =   sum[j] + temp[j][i],
          index[j] = sum[j],
          result3 = sum[j],

          }
```

Appendix E continued:

```
        }
        sort4();
}

void sort4 ( void )
{
                qsort(index,n,sizeof(index[1]),strcmp);

        for(i=0;i<n;i++)
        {
          j =0;
           while((index[i] != sum[j]) && (j<n))
              j++;
          for(k=0;k<m,k++)
              {
              T[i][k] = temp[j][k];
              }
          sum[j] = 9999;
          }
         for (i=0;i<n-1;i++)
         { temp[i][0] = store2[i]; }
        process();
 }
 void rule5( void )
{
        for ( i=0;i<n+m;i++)
        {
                sum[i] =0;
                index[i] =0;
        }
        rulenum  = 5;
        for(i=0;i<n;i++)
        {
                for(j=0;j<m;j++)
                {
                sum[i]   =   sum[i] + temp[i][j];
                index[i] = index[i] + temp[i][j];
                }
}
                qsort(&index,n,sizeof(index[1]),strcmp);
        for ( j=0;j<n;j++)
        {
                store[n-j-1] = index[j];
        }
        for ( j=0;j<n;j++)
        {
                index[j] = store[j];
        }
                for(i=0;i<n;i++)
        {
          j =0;
```

Appendix E continued:

```
         while((index[i] '= sum[j]) && (j<n))
            ++j,
        for(k=0,k<m,k++)
            {
            T[i][k] = temp[j][k],
            }
        sum[j] = 9999,
        }
      for (i=0,i<n,i++)
      { temp[i][j] = store2[i], }
      process(),
 }
void rule2 ( void )
{
       /*  It allows the program to sort the values   */
    rulenum = 2,
            for(j=0,j<n,j++)
            {
                  index[j] = temp[j][0],
            }
    /* sorting the first row by using the qsort method */
            qsort(&index,n,sizeof(index[1]),strcmp),
      for(i=0,i<n,i++)
      {
        j =0,
         while((index[i] '= temp[j][0]) && (j<n))
            ++j,
        for(k=0,k<m,k++)
            {
            T[i][k] = temp[j][k],
            }
        store2[j] = temp[j][0],
        temp[j][0] = 9999,
        }
      for (i=0,i<n,i++)
      { temp[i][0] = store2[i], }
      process(),
      }                    /* end of rule2() */
void rule3( void )
{
            rulenum =3,
            for(j=0,j<n,j++)
            {
                  index[j] = temp[j][0],
            }
      /* sorting the first row by using the qsort method */

            qsort(&index,n,sizeof(index[1]),strcmp),
      for ( j=0,j<n,j++)
      {
            store[n-j-1] = index[j],
```

Appendix E continued:

```
      }
      for ( j=0;j<n;j++)
      {
            index[j] = store[j];
      }
      for(i=0;i<n;i++)
      {
        j =0;
         while((index[i] != temp[j][0]) && (j<n))
            ++j;
        for(k=0;k<m;k++)
            {
            T[i][k] = temp[j][k];
            }
        temp[j][0] = 9999;
        }
      process();
}
int intcmp(int a,int b)
{
return(a-b);
}
void process( void )
{
label      f[0][0] = T[0][0]; /* the Make-Span formula */
      for ( i=1; i<m;i++)
      {
            f[0][i] = f[0][i-1] + T[0][i];
      }
      for ( i=1; i<n;i++)
      {
            f[i][0] = f[i-1][0] + T[i][0];
      }
      for ( j = 1; j<m;j++)
      {
            for ( i=1;i<n;i++)
            {
            f[i][j] = max( f[i-1][j], f[i][j-1] ) + T[i][j];
            }
      }
 }     /* end of process */
void display ( void )
 {
      system("cls");
      printf("\n");
      i =m-1;
            for(j=0;j<n-1;j++)
            {
            result1= f[j][i];
            }
       /*   printf(" result rule %i = %i\n",rulenum,f[n-1][i]);
            printf(" = %i\n",f[n-1][i]);
```

Appendix E continued:

```
        fprintf(fp2," result rule %i =
                %i\n",rulenum,f[n-1][i]), */
        result = f[n-1][i],
        {
        i = m-1,
        for(j=0,j<n-2,j++)
        {
        result1 =result1+ f[j][i],
        }
        result2 = (result1 + result)/n,
/*      printf("mean completion time = %i",result2),
        fprintf(fp2,"Mean completion time for rule  %i =
              %i\n",rulenum,result2),
        fprintf(fp2," = %i\n",result2), */
        }
        for(i=0,i<n+m,i++)
        {
        p[i]=0,
        sum[i] = 0,
        }
              count = 0,
              p[0]=0,
              for (i=1,i<m,i++)
              {
              p[i] =  p[i-1]+T[0][i-1],
              }
              for(i=0,i<m,i++)
              {
              count = 0,
                    for(j=0,j<n,j++)
                    {
                    count  = count + T[j][i],
                    }
                    sum[i] = count,
                    }
                    {
        for(i=0,i<m,i++)
        {
        for(j = 0,j<n,j++)
        {
        }
        printf("\n"),
        idle = f[n-1][i]-sum[i]-p[i],
        result5 = f[n-1][i]-sum[i]-p[i],
        }
        printf("\n"),
        }
        j =n-1,
        for(i=0,i<m-1,i++)
        {
        result5 = result5 + f[n-1][i] -sum[i]-p[i],
```

Appendix E continued:

```
        }
  /*    prıntf("Total ıdle tıme = %ı",result5),
        fprıntf(fp2,"Total ıdle tıme for rule %ı = %ı\n",
                   rulenum,result5),
        fprıntf(fp2," = %ı\n",result5),   */
        {
        for(ȷ=0,ȷ<n+m,ȷ++)
        {
        z[ȷ]=0,
        su[ȷ]=0,
        }
              z[0]   = 0,
              for (ȷ=1,ȷ<n+1,ȷ++)
              {
              z[ȷ]  =     z[ȷ-1] +T[ȷ-1][0],
              }
              for(ȷ=0,ȷ<n,ȷ++)
              {
                    count = 0,
                    for(ı=0,ı<m,ı++)
                    {
                    count = count + T[ȷ][ı],
                    }
                    su[ȷ] = count,
              }
                    {
                    prıntf("\n"),
                    ı= m-1,
                    for(ȷ=0,ȷ<n,ȷ++)
                    {
                    prıntf("\n"),
                    result8 =f[ȷ][ı] -su[ȷ]-z[ȷ],
                    {
                    prıntf("\n"),
                    }
                    ı = m-1,
                    for(ȷ=0,ȷ<n,ȷ++)
                    {
                    result8 = result8 + f[ȷ][ı] - su[ȷ] - z[ȷ],
                    }
                    {
                          /* end of dısplay() */
}
}
}
}}
```

xxxxx   End The Programme   xxxxx

APPENDIX F: "Percentage points of the F Distribution"

$F_{.05,\nu_1}$

Degrees of Freedom for the Numerator (ν_1)

ν \ ν_1	1	2	3	4	5	6	7	8	9	10	12	15	20	24	30	40	60	120	∞
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9	243.9	245.9	248.0	249.1	250.1	251.1	252.2	253.3	254.3
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.41	19.43	19.45	19.45	19.46	19.47	19.48	19.49	19.50
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.74	8.70	8.66	8.64	8.62	8.59	8.57	8.55	8.53
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.91	5.86	5.80	5.77	5.75	5.72	5.69	5.66	5.63
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.68	4.62	4.56	4.53	4.50	4.46	4.43	4.40	4.36
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	4.00	3.94	3.87	3.84	3.81	3.77	3.74	3.70	3.67
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.57	3.51	3.44	3.41	3.38	3.34	3.30	3.27	3.23
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.28	3.22	3.15	3.12	3.08	3.04	3.01	2.97	2.93
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.07	3.01	2.94	2.90	2.86	2.83	2.79	2.75	2.71
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.91	2.85	2.77	2.74	2.70	2.66	2.62	2.58	2.54
11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.79	2.72	2.65	2.61	2.57	2.53	2.49	2.45	2.40
12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.69	2.62	2.54	2.51	2.47	2.43	2.38	2.34	2.30
13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.60	2.53	2.46	2.42	2.38	2.34	2.30	2.25	2.21
14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.53	2.46	2.39	2.35	2.31	2.27	2.22	2.18	2.13
15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.48	2.40	2.33	2.29	2.25	2.20	2.16	2.11	2.07
16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.42	2.35	2.28	2.24	2.19	2.15	2.11	2.06	2.01
17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.38	2.31	2.23	2.19	2.15	2.10	2.06	2.01	1.96
18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.34	2.27	2.19	2.15	2.11	2.06	2.02	1.97	1.92
19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.31	2.23	2.16	2.11	2.07	2.03	1.98	1.93	1.88
20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.28	2.20	2.12	2.08	2.04	1.99	1.95	1.90	1.84
21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.25	2.18	2.10	2.05	2.01	1.96	1.92	1.87	1.81
22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.23	2.15	2.07	2.03	1.98	1.94	1.89	1.84	1.78
23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.20	2.13	2.05	2.01	1.96	1.91	1.86	1.81	1.76
24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.18	2.11	2.03	1.98	1.94	1.89	1.84	1.79	1.73
25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.16	2.09	2.01	1.96	1.92	1.87	1.82	1.77	1.71
26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.15	2.07	1.99	1.95	1.90	1.85	1.80	1.75	1.69
27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.13	2.06	1.97	1.93	1.88	1.84	1.79	1.73	1.67
28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.12	2.04	1.96	1.91	1.87	1.82	1.77	1.71	1.65
29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18	2.10	2.03	1.94	1.90	1.85	1.81	1.75	1.70	1.64
30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16	2.09	2.01	1.93	1.89	1.84	1.79	1.74	1.68	1.62
40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08	2.00	1.92	1.84	1.79	1.74	1.69	1.64	1.58	1.51
60	4.00	3.15	2.76	2.53	2.37	2.25	2.17	2.10	2.04	1.99	1.92	1.84	1.75	1.70	1.65	1.59	1.53	1.47	1.39
120	3.92	3.07	2.68	2.45	2.29	2.17	2.09	2.02	1.96	1.91	1.83	1.75	1.66	1.61	1.55	1.55	1.43	1.35	1.25
∞	3.84	3.00	2.60	2.37	2.21	2.10	2.01	1.94	1.88	1.83	1.75	1.67	1.57	1.52	1.46	1.39	1.32	1.22	1.00

APPENDIX G:

The SIMAN Model Codes for the Job-Shop Priority Rules.

1. The CREATE Pseudo-code allows to the workpieces (the entities) enter into the model

The CREATE Pseudo-code, one of several mechanisms for entering entities into the model, is typically used to model arrival processes in which entities (workpieces) sequentially enter the model according to a specified pattern
In our problem workpieces inter the job-shop one at a time with probability distributed random times between arrivals (later on the proposed priority rules will be simulated using three different arrival times distributions shown in Fig 23)

2. The ASSIGN Pseudo-code allows the assignment of a value to a SIMAN variable or entity attribute

This statement lets the arriving workpieces to be assigned five variables as follows

1 The part number It is assigned as

j=j+1

Hence,

Part type=j

2 The part type

This could be defined by using the random variable DISCRETE according to probabilities specified in the form of a cumulative probability distributions In our current problem the set of discrete values (workpieces) consists of 1, 2 and 3 workpiece, and the corresponding cumulative probabilities are 0 24, 0 68 and 1 0, respectively Note that the second (0 68)

Appendix G continued:

and last value (0.1) was obtained by adding 0.44(the probability of a workpiece 2) to 0.24(the probability of a workpiece 1) and adding 0.32(the probability of a workpiece 3) to 0.68, respectively, where the first value (0.24) is the probability of a workpiece 1. Hence,
The second assignment is as:

Part Type = DISCRETE(0.24,1,0.68,2,1.0,3):

3. The release time: It is defined as follows:

ArrTime = TNOW:

4. The Due Date: It is defined as follows:

Due Date = TNOW + Uniform(300,550):

5. The Static Slack: It is defined as follows:

Static Slack = Due date - ArrTime;

The complete model for the ASSIGN statement is as follows:

```
ASSIGN: J=J+1:
        PartNo=j:
        NS=DISCRETE(0.24,1,0.68,2,1.0,3):
        PartType=NS:
        ArrTime =TNOW:
        DueDate = TNOW+Uniform(300,550):
        StaticSlack = DueDate-ArrTime;
```

3. **The ROUTE Pseudo-Code** allows the unconstrained movement of the workpieces from one machines to another. In the proposed job-shop the operation time includes the set up and transportation times. The model statement for transportation is as follows:

   ROUTE;

4. **The STATION Pseudo-Code is the point of the model to which workpieces are transferred according to them route. It**

Appendix G continued:

allows the workpieces to replace its processing from one machine to another according to the limited range In the current job-shop the model statement for this range is as follows

STATIONS: 1-6;

5. **The QUEUE Pseudo-Code:** This statement is used to provide waiting space to the workpieces for processing
In our case the QUEUE statement is modeled as follows

QUEUES: M; Where M is the number of machines = 1-6

5. **The SEIZE Pseudo-Code** allocates the machine unit M to the workpiece In our current job-shop workpieces arriving at QUEUE-SEIZE statement combination wait their turn in the QUEUE statement to be allocated one unit of the machine If at least one of machine is idle at a time an workpiece arrives at the preceding QUEUE statement, then the number of busy units of machine is increased by one, and the workpiece passes through the SEIZE statement without waiting in the preceding QUEUE statement On the other hand, if all the units of machine are busy, the workpieces is held in the preceding QUEUE statement until a unit of machine becomes available for allocation to the workpiece
In this case the SEIZE statement is modeled as follows

SEIZE: Machine(M);

6. **The DELAY Pseudo-Code** Once the workpiece has been allocated the necessary machines, it typically engages in time-consuming activities, in general, such as set up, operation times and inspection In the current job-shop the set up and inspection time are included within the operation time In this case the DELAY statement is modeled as follows

Appendix G continued:

DELAY: OpTime,

The details of operation times for jobs are located within the experimental element programme in sub-chapter ()

7. **The RELEASE Pseudo-Code** releases the machine unit M from the workpiece The released machine became idle and is then available for allocation to workpieces waiting at SEIZE statement In the current job-shop the RELEASE statement is modeled as follows

RELEASE: Machine (M):
Next (Next Operation)

8. **The Exit System STATION** is the final STATION allows the workpieces to exit the proposed job-shop The statement code of this step is modeled as follows

STATION. ExitSystems;

The steps of the model numbered 3, 4, 5, 6 and 7 are called station sub-model

8. **The TALLY Pseudo-Code** This statement records some of the observational data during the simulation execution according to the TALLIES element in the experiment frame In our current job-shop the TALLY statement is coded as follows

TALLY: FlowTime, INT(ArriTime);
TALLY: Lateness, TNOW-DueDate;

8. **The COUNT Pseudo-Code** increments the jobs which has been done by the machines and exit the job-shop The limit of the counting is defined in the COUNTERS element in the experiment frame This statement is coded in the model frame as follows

Appendix G continued:

COUNT: JobsDone,

9. **The BEGIN and END pseudo-Codes** are entered in the beginning and in the end of the model programme as shown in the final model programme Fig 24

The SIMAN Model Programme for The Job-Shop Priority Rules

```
BEGIN,
Create      CREATE   Expo(7)
                         MARK(ArrTime),
            ASSIGN       M=ENTER,
            ASSIGN       J=J+1
                         PartNo=J,
            ASSIGN       NS=DISCRETE(0 24,1,0 68,2,1 0,3)
                         PartType=NS
                         ArrTime=TNOW
                         DueDate=TNOW+UNIF(300,550)
                         StS=DueDate-ArrTime,
NextOp      ROUTE,                             To next operation
            STATION,     1-6,                  No  of the Stations
            QUEUE,       M,                    Queues 1-6
            SEIZE        Machine(M),           Get machine
            DELAY        OpTime,               Processing Times
            RELEASE      Machine(M)
                         NEXT(NextOp),         Release machine
            STATION,     ExitSystem,                  Exit submodel
            TALLY        FlowTime,INT(ArrTime),
            TALLY        LateNess,TNOW-DueDate,
            TALLY        Make_Span,TNOW,
            COUNT        JobsDone
                         DISPOSE,
END,
```

xxxxx End of The Programme xxxxx

APPENDIX H:

```
BEGIN,

PROJECT,          3Jobs 6M/Cs Job Shop, Mujanah,

ATTRIBUTES        OpTime          Oeration Time
                  ArrTime         Arrival Time
                  PartNo
                  PartType
                  DueDate
                  StS,            Static Slack (Static SLK)

STATIONS       1, Casting
               2, Lathes
               3, Planers
               4, Drills
               5, Shapers
               6, Polishers
               7, Enter
               8, ExitSystem,        Exit System

QUEUES         6,


,           xxxxx  The selected priority rule   xxxxx

RANKINGS       1-6,LVF(OpTime),     SPT rule


,           xxxxx  The number of M/Cs    xxxxxx

RESOURCES         Machine(6),14,5,4,8,16,4,


,           xxxxx  The Jobs sequences & Operation Times   xxxxx

SEQUENCES      1, 1,expo(125) & 3,expo(35) & 2,expo(20) &
                  6,expo(60) & ExitSystem
               2, 5,expo(105) & 4,expo(90) & 2,expo(65)
                  & ExitSystem
               3, 1,expo(235) & 5,expo(250) & 4,expo(50) &
                  3,expo(30) & 6,expo(25) & ExitSystem,
COUNTERS          JobsDone,    Jobs_Done

,           xxxxx  The Output Simulation Data    xxxxx

TALLIES           FlowTime ',"F DAT"   Flow time
                  LateNess,
DSTATS           1, NQ(1), Caster Queue
                 2, NQ(2), Lathes Queue
```

Appendix H continued:

```
                    3, NQ(3), Planers Queue
                    4, NQ(4), Drill Queue
                    5, NQ(5), Shapers Queue
                    6, NQ(6), Polishing Queue
                    7, (NR(1)/14)*100, Caster Utilization
                    8, (NR(2)/5)*100, Lathes Utilization
                    9, (NR(3)/4)*100, Planers Utilization
                    10, (NR(4)/8)*100, Drill Utilization
                    11, (NR(5)/16)*100, Shapers Utilization
                    12, (NR(6)/4)*100, Polishing Utilization
                    13, ((NR(1)/14)*100+(NR(2)/5)*100+(NR(3)/4)*100+
                        (NR(4)/8)*100+(NR(5)/16)*100+
                        (NR(6)/4)*100)/6 ,All Utili
                        Levels,',"U DAT",
,                    14,TSTD(1),STD,"STD DAT",
                    15,NQ(1)+NQ(2)+NQ(3)+NQ(4)+NQ(5)+
                        NQ(6), WIP,"WIP DAT",

OUTPUTS              1,TAVG(1),"FL DAT",Flow_Time
                     2,TAVG(3),"LA DAT",Lateness
                     3,100-((DAVG(7)+DAVG(8)+DAVG(9)+DAVG(10)+
                       DAVG(11)+DAVG(12))/6),"I DAT",%  ALL  IDLE
TIMES
,                     4,DAVG(7),"U1 DAT",U1
,                     5,DAVG(8),"U2 DAT",U2
,                     6,DAVG(9),"U3 DAT",U3
,                     7,DAVG(10),"U4 DAT",U4
,                     8,DAVG(11),"U5 DAT",U5
,                     9,DAVG(12),"U6 DAT",U6
                      10,DAVG(1)+DAVG(2)+DAVG(3)+DAVg(4)+DAVG(5)+
                      DAVG(6),"WIP DAT",Work_In_Process  for  all
MACHs
,                     11,DAVG(1),"W1 DAT",WIP1
,                     12,DAVG(2),"W2 DAT",WIP2
,                     13,DAVG(3),"W3 DAT",WIP3
,                     14,DAVG(4),"W4 DAT",WIP4
,                     15,DAVG(5),"W5 DAT",WIP5
,                     16,DAVG(6),"W6 DAT",WIP6
                      17,NC(Jobs_Done),"JD DAT",Jobs Competed,


,  xxxxxx  The transient portion (6000) and
                production days (13120) xxxxxx

REPLICATE,       7,0,19120,,,6000,

END,


                xxxxxx  End The Programme  xxxxx
```

APPENDIX I:

The SIMAN model codes for the FMS

```
BEGIN,
            CREATE       Expo(25)              'High FMS load
                         MARK(Arrival Time),
            ASSIGN       J=J+1
                         PartNo=J,
            ASSIGN       NS= DISC(0 25,1,0 65,2,1 0,3)
                         PartType=NS
                         Arrival Time=TNOW
                         DueDate = TNOW+UNIF(250,400)
                         StS=DueDate-Arrival Time
            QUEUE,       AGV Load Queue,
            REQUEST      AGV(SDS),,Load_Unload,
            DELAY        TRIA(1,2,3),       'Man Loading time
            TRANSPORT    AGV,SEQ,
            STATION,     CNC_Lathe-CNC_Milling,
            DUPLICATE    1,Process
                         NEXT(AGVChk),
Process     QUEUE,       CNC_Lathe Queue + (M-CNC_Lathe),
            SEIZE        Machine(M-CNC_Lathe+1),
            DELAY        Process Time,
            RELEASE      Machine(M-CNC_Lathe+1),
            QUEUE,       AGV In_Process Queue,
            REQUEST,     ,M-CNC_Lathe+1
                         AGV(SDS),
            DELAY        1,          'Robot loading time
            TRANSPORT    AGV,SEQ,
            STATION,     Load_Unload,
            DELAY        TRIA(1,2,3),    'Man Unloading time
            DUPLICATE    1,JobOut
                         NEXT(AGVChk),
JobOut      TALLY        Flow_Time,INTERVAL(
                         Arrival Time),
            TALLY        Lateness,TNOW-Duedate,
            COUNT        JobsDone
                         DISPOSE,
            BRANCH,      1
                         IF,NQ(AGV Load_Queue)+
                         NQ(AGV In_Process Queue)
                         ==0,StageAGV
                         ELSE,NoStage,
StageAGV    TRANSPORT    AGV,Stage,
            STATION,     Stage,
NoStage     FREE         AGV
                         DISPOSE,
END,
                        xxxxx  The End Of Programme  xxxxxx
```

APPENDIX J:

The SIMAN Experiment Elements for the FMS

```
BEGIN,
PROJECT,          3Job 4Mach, FMS with AGV System,Mujanah,
ATTRIBUTES        Arrival Time
                  Process Time
                  DueDate
                  StS
                  PartNo
                  PartType
                  Anim_Att,
QUEUES           1, AGV Load Queue        'Load Queue
                 2,AGV In_Process Queue    ' Buffer Storage
                 3, CNC_Lathe m/c Queue
                 4, Horizental Machining Queue
                 5, Co_ordinate Measuring m/c Queue
                 6, CNC_Milling m/c Queue,
SCHEDULES        1,1*480,0*30           'Brake time
                 2,1*2880,0*Lognormal(15,3),   ' Cleaning &
,                     Re-replication time
RESOURCES         Machine(4),sched(1),sched(2),

,       xxxxxxxxx  The priority rule   xxxxxxx

RANKINGS          1-6,LVF(Process Time),

,       xxxxxxxxx  The FMS Activities  xxxxxxx

STATIONS          CNC_Lathe m/c,2
                  CNC_Horizental Machining,3
                  Co_Ordinate Measuring m/c,4
                  CNC_Milling m/c,5
                  Load_Unload,1
                  Stage,9,

,        xxxxxxxxx  Job Sequencing  xxxxxxxx

SEQUENCES         1, CNC_Lathe m/c  ,Process Time=Expo(
                     40)&
                     CNC_Milling m/c    ,Process Time=expo(
                     30)&
                     Co-Ordinate Measuring m/c,Process Time=expo(
                     20)&
                     Load_Unload
                  2, CNC_Milling m/c    ,Process Time=expo(
                     26)&
                     CNC_Horizental   Machining       ,Process
                           Time=expo(
                     30)&
                     CNC_Lathe m/c  ,Process Time=Expo(
```

Appendix J Continued:

```
                      25)&
                      Co_Ordinate Measuring m/c,Process Time=Expo(
                      17) &
                      Load_Unload

                     3,CNC_Horizental m/c   ,Process Time=Expo(
                      33)&
                      CNC_Milling m/c     ,Process Time=Expo(
                      25)&
                      Co_Ordinate Measuring m/c,Process Time=Expo(
                      18)&
                      Load_Unload,
COUNTERS              JobsDone,
INTERSECTIONS       1,,10    ' The conection points
                    2,,10
                    3,,10
                    4,,10
                    5,,10
                    6,,10
                    7,,10
                    8,,10
                    9,,10
                    10,,10
                    11,,10,

'       xxxxxxx   The specification of the AGVs paths   xxxxxxx

LINKS               1,,6,1,10,10,Spur 'The Load & Unload area
                    2,,6,7,6,10
                    3,,7,10,,10
                    4,,10,11,4,10 'The Staging area & Charging room
                    5,,11,2,8,10
                    6,,2,3,8,10 'CNC_Lathe & Horize  Machining area
                    7,,3,8,5,10
                    8,,8,4,10,10 '
                    9,,4,5,7,10 ' Co_ordinate Measu  & Milling area
                    10,,5,6,3,10
                    11,,8,7,12,10
                    12,,10,9,6,10
                    13,,9,11,1,10,
NETWORKS            1,AGV System,1-13,

'        xxxxxxx   The specification of the AGVs   xxxxxxxx

TRANSPORTERS        1,AGV,2,NETWORK(AGV System),
                    100 0,LINK(12),
TALLIES             1,Flow_Time·',"FL DAT"
                    2,Lateness,

'       xxxxxxxx   The output data commands   xxxxxxxx
```

Appendix J continued:

```
DSTATS          1, NQ(AGV In_Process Queue)+NQ(
                   AGV Input Queue),
                   Waiting for Pickup
                2, NQ(CNC_Lathe m/c Queue)
                3, NQ(CNC_Horizental Machining Queue)
                4, NQ(Co-Ordinate Measuring m/c Queue)
                5, NQ(CNC_Milling m/c Queue)
                6, NR(1)*100,CNC_Lathe m/c Util


                7, NR(2)*100,CNC_Horizental m/c Util
                8, NR(3)*100,Co_Ordinate Measuring m/c Util
                9, NR(4)*100,CNC_Milling m/c Util
                10, 100-((NR(1)*100+NR(2)*100 +NR(3)*100+
                        NR(4)*100)/4),All Idle Times
                11, IT(AGV,1),AGV 1 Time Busy
                12, IT(AGV,2),AGV 2 Time Busy
                13, NQ(2)+NQ(3)+NQ(4)+NQ(5),WIP for shop floor
                14, 100-(NT(1)*100)/2,Idle AGVs
                15, TSTD(1),STD,"STD DAT",
OUTPUTS         1,TAVG(1),"FL DAT",flow time
                2,TAVG(2),"LA DAT",Lateness
                3,DAVG(10),"I DAT",IdleTime
                4,DAVG(13),"WIP1 DAT",Waiting for shop floor
                5,DAVG(1),"WIP DAT",Waiting for pick up
                6,NC(JobsDone),"JD DAT",Jobs Completed
                7,DAVG(14),"IAGVs DAT",Idle AGVs,


,    xxxxxx    The transient (5940 min) and
,                 production times (10560 min)   xxxxxxx

REPLICATE,       7,0,16500,,,5940,
END,

,      xxxxx The end of the SIMAN Experiment Programme  xxxxxx
```

The above intersection and link elements respectively illustrate the number of intersections which is equal to 11, each of length 10 (the length of all intersection = 11 x 10 ft=110 ft) and the number of links which is equal to 13, each of length as shown below and according to Figure 64

Link 1 From point 1 to 6 equal to 100 ft

Link 2 From point 6 to 7 equal to 60 ft

Link 3 From point 7 to 10 equal to 10 ft

Link 4 From point 10 to 11 equal to 40 ft

Appendix J continued:

Link 5 From point 11 to 2 equal to 80 ft

Link 6 From point 2 to 3 equal to 80 ft

Link 7 From point 3 to 8 equal to 50 ft

Link 8 From point 8 to 4 equal to 100 ft

Link 9 From point 4 to 5 equal to 70 ft

Link 10 From point 5 to 6 equal to 30 ft

Link 11 From point 8 to 7 equal to 120 ft

Link 12 From point 10 to 9 equal to 60 ft

Link 13 From point 9 to 11 equal to 10 ft

In the next page, Figure 7 19 illustrates the FMS's AGV layout network, where the numbers from 1 to 11 are called the intersections (in The SIMAN code element) from which the link between two intersections is defined (i e the link number 1 and 2 are defined respectively between the intersections 1 & 6 and 6 & 7 and so on as shown in the SIMAN experiment programme (the LINKS element) in Appendix J page J2

Appendix J continued:

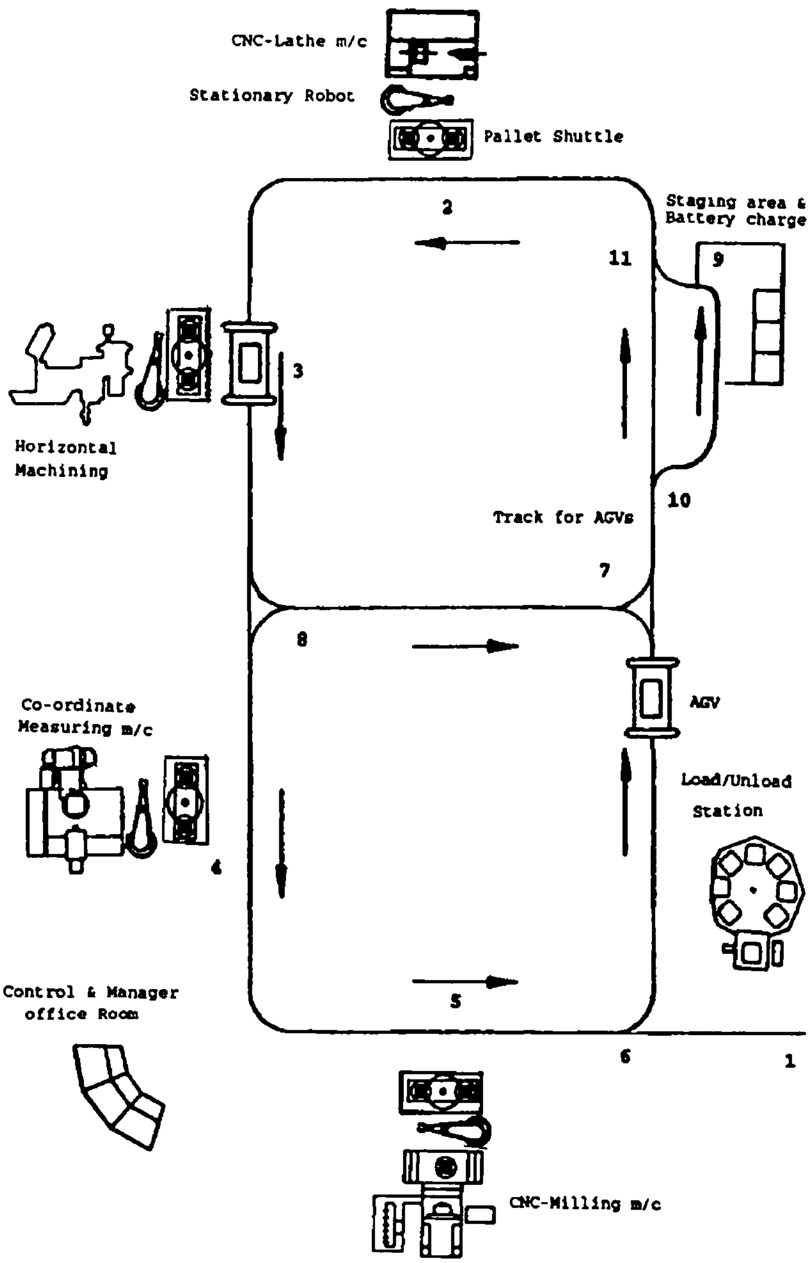


Fig. 7.19: The two FMS's AGVs layout network (11 intersections and 13 links)