# THE STUDY OF PASSING IN FLOW SHOP SCHEDULING PROBLEMS

Dissertation for the Degree of Ph. D.

MICHIGAN STATE UNIVERSITY

JAGDISH MANUBHAI MEHTA

1975

This is to certify that the

thesis entitled

THE STUDY OF PASSING IN FLOW SHOP
SCHEDULING PROBLEMS

presented by

JAGDISH MANUBHAI MEHTA

has been accepted towards fulfillment
of the requirements for

PH.D. degree in MANAGEMENT

Phillip L. Carter
Major professor

Date May 14, 1975

110

SEP 26 1974

# ABSTRACT

## THE STUDY OF PASSING IN FLOW SHOP SCHEDULING PROBLEMS

By

Jagdish Manubhai Mehta

In 1954, Johnson provided a simple algorithm to find an optimum solution minimizing the total make-span for a general N job, 2 machine and a special N job, 3 machine flow shop scheduling problem. Since then his 13-14 assumptions-restrictions and performance criterion for minimizing the total make-span have become popularized, being referred to as the "classical flow shop scheduling problem." Other researchers in the field have applied techniques such as combinatorial analysis, mathematical programming, heuristics, and Monte Carlo sampling to find optimum solutions to N job, M machine classical flow shop scheduling problems where M is equal to or greater than three, but with limited success.

Then in 1965, almost simultaneously, Ignall and Schrage in the United States, and Lomnicki in Great Britain discovered the Branch and Bound technique was useful for finding optimum solutions to large N job, M machine classical flow shop scheduling problems. Others have since tried to obtain optimum solutions to these problems, with one or more assumptions-restrictions removed, also with limited success.

In this dissertation, a study of the classical flow shop scheduling problem has been undertaken with the no passing restriction removed. The no passing restriction implies that once a job sequence has been selected for the first machine in the flow shop, the same job sequence should be used for all subsequent machines.

In the last 20 years, all but a few researchers have assumed in flow shop scheduling problems there is no need to change job sequences between machines. They probably thought any change in job sequence would only delay and in no way improve the value of the minimum total make-span. In preliminary analysis for this dissertation, a hundred 4 job, 4 machine flow shop scheduling problems were solved to see the effects of removing the no passing restriction from classical flow shop scheduling problems. Different job sequences on the first two machines than from the last two machines were permitted. The results were that in ten problems, passing was beneficial, while in the other ninety, it made no difference as far as minimizing the total make-span was concerned. The total make-span improvement in the ten problems where passing was permitted, was fairly small.

Conway, Maxwell, and Miller in Theory of Scheduling showed that changing the job sequence between the first and second machines, as well as between the last and second to last, does not improve the performance criterion of minimizing the total make-span. Thus, in a 3 machine flow shop, there is no need to change sequences between any two machines but in a 4 machine flow shop, changing the job sequence between the first and last two machines could prove useful. Taking this into account, a thousand problems each for 3 job, 4 machine; 4 job, 4 machine; and 5 job, 4 machine flow shop scheduling problems were constructed, using integer random numbers from Uniform probability distribution between 0 and 100 for processing times. The experiments on these 3,000 problems was divided into two parts: Phase One where the no passing restriction was maintained; and Phase Two where passing was permitted between the first two machines and the last two machines. Complete enumeration of all possible sequences in both phases was conducted for each problem, with any decrease in the total make-span in Phase Two compared to the minimum total make-span of Phase One noted. The number of optimum solutions of both phases and any correlation between them for each problem was noted. Previously, almost all researchers used processing times either chosen arbitrarily or generated from Uniform probability distribution. In research for this paper, Uniform probability distribution as well as Beta probability distribution were used to generate processing times for an additional 15,000 problems. Five sets of values for parameters of Beta probability distribution were chosen, and for each of them random numbers were generated for a thousand 3 job, 4 machine; 4 job, 4 machine; and 5 job, 4 machine flow shop scheduling problems. Complete enumeration studies of both phases for each of these problems were conducted.

It was determined that under the no passing restriction, the range between the maximum and minimum total make-spans is narrow. When the processing times are uniformly distributed, the range is only 24, 31, and 37 percent of the minimum total make-span for 3 job, 4 job, and 5 job, 4 machine flow shop scheduling problems and, as values of AB--parameters of Beta probability distribution--increase, values of the range decrease significantly. It was also found that 10.5, 13.0, and 18.5 percent of the 3 job, 4 job, and 5 job, 4 machine flow shop scheduling problems provide a lower value of the total make-span if passing is permitted, than that for the minimum total make-span under a no passing restriction when processing times are uniformly distributed. Therefore, as the number of jobs in the flow shop scheduling problem increase, the number of problems for which permitting passing lowered the minimum total make-span of Phase One in Phase Two also increases. This phenomenon was also observed in all five sets of problems whose processing times were Beta-distributed. The average reduction in the minimum total make-span where passing was permitted for 3 job, 4 job, and 5 job, 4 machine flow shop scheduling problems amounts to 0.415, 0.426, and 0.519 percent, respectively, when processing times are uniformly distributed. Although the average reduction in the minimum total make-span is small, it increases by 21.5 percent when the size of the flow shop scheduling problem increases by one job. Similar increases have been noted in the average reduction when processing times are Beta-distributed.

A most important finding in this dissertation is that as values of the parameters of Beta distribution for processing times increases, the number of problems for which permitting passing lowered the minimum total make-span of Phase One, and the average percentage of reduction in the minimum total make-span of Phase One, decrease significantly.

In addition, this dissertation provides three different heuristics or "search plans," each of which can provide solutions very close to the optimum solutions obtained with passing permitted, and with only a fraction of the effort. For example, for the one thousand 5 job, 4 machine flow shop scheduling problems, Plans One, Two, and Three provided 99.41, 80.80, and 80.13 percent of the possible reduction of the minimum total make-span with only 8.42, 4.24, and 0.36 percent of the computational efforts as compared to that required with complete enumeration.

# THE STUDY OF PASSING IN FLOW SHOP SCHEDULING PROBLEMS

By

Jagdish Manubhai Mehta

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Management

1975

# ACKNOWLEDGMENTS

Many people have assisted me in assembling this dissertation. I would like to acknowledge them here.

My special thanks goes to Professor Phillip Carter, who as dissertation committee chairman, encouraged and motivated me whenever I needed it. His insight, suggestions, and constructive criticism were a constant help to me. Professor Richard Gonzalez as a dissertation committee member and department chairman, was extremely helpful in solving all problems that came up during the process of preparing this paper. He was more than generous with his time, knowledge, and resources. And Professor Kamalesh Banerjee was always available whenever I needed help. His knowledge of the problem and his suggestions along the way were quite useful. These men all deserve a special note of thanks for the guidance they gave me. I am extremely grateful to them.

Many faculty and staff members of Bowling Green State University also assisted me and provided valuable resources. Professors Chan Hahn, David Hyslop, Michael Delaney, and Ms. Charlene McComas were especially helpful.

I am also indebted to Professors Leo Erickson and Richard Henshaw of Michigan State University and Professor Jan Kmenta of the University of Michigan for their help.

Finally, a special note of appreciation goes to my wife, Devyani, who was a constant source of help and encouragement to me along the way.

TABLE OF CONTENTS

| Chapter | | Page |
|---|---|---|

LIST OF TABLES

Table Page

Table Page

## LIST OF FIGURES

# CHAPTER I

## INTRODUCTION

The following statement, although written in 1967 for the book Theory of Scheduling, indicates to a large extent the present state of knowledge in the field of scheduling.

> Scheduling is a field in which there are some intriguing problems and some interesting answers. So far, however, the subject has not received the attention it deserves; work on it has been fragmented at best . . .[1]

In an industrial society such as ours, efficient and effective scheduling of operations of any kind is very important to utilize our resources of men and machines to the fullest extent. This fact, though very simple to understand and agree with, is generally not acted upon by industrialists responsible for the job of scheduling production operations. The organizational environment for them is such that neither an efficient schedule rewards nor an inefficient schedule punishes them. Professor W. F. Pounds

---

[1]Conway, Maxwell, and Miller, Preface of Theory of Scheduling (Reading, Mass.: Addison-Wesley, 1967).

studied the scheduling environment of our industries in great detail and reported:

> The job-shop scheduling problem is not recognized by most factory schedulers because for them, in most cases, no scheduling problem exists. That is, there is no scheduling problem for them because the organization which surrounds the schedulers reacts to protect them from strongly interdependent sequencing problems.[2]

On the other hand, some schedulers realize the scheduling problem, only to get frustrated by its extent and their inability to deal with it.

In both of these situations--the lack of knowledge of the problem and the lack of ability to handle it--people consciously or unconsciously schedule operations guided by their intuition. And scheduling on the basis of mere intuition, contrary to many people's beliefs, is not advisable because it does not produce good results in most cases. Although some people have a better scheduling intuition than others, most of us need some rules, guidelines and procedures to schedule operations effectively and efficiently.

Thus, the job for researchers of the scheduling area is two-fold. The first task is to make schedulers aware of the nature of the scheduling problem, its characteristics, and its effect on daily production efficiency. The second task is to help solve the problem of inefficient production

---

[2]W. F. Pounds, "The Scheduling Environment," Chapter 1 of Industrial Scheduling, edited by J. F. Muth and G. L. Thompson (Englewood Cliffs, New Jersey: Prentice-Hall, 1963).

scheduling by using guidelines, procedures, and rules so that the utilization of resources can be maximized or optimized. This does not seem particularly difficult except for the fact that researchers themselves do not have solutions to various scheduling problems. The situation is especially acute in the flow shop scheduling problem as few useful results have been obtained finding its solution. This lack of achievement is highlighted and analyzed in Chapter II where the present state of knowledge of the flow shop scheduling problem will be discussed.

## Brief Overview of Some Scheduling Concepts

To understand the area of scheduling, it is first necessary to become familiar with some key concepts.

SHOP, JOB, OPERATION and MACHINE are four interrelated basic terms that are used widely. The term "SHOP" refers to manufacturing facilities where various work functions are being completed. "JOB" refers to an item or a commodity that is being transformed in a shop from a stage which requires some work to be done to a stage where the required work is completed. The term "OPERATION" denotes a partial or full transformation process that is being applied to a job. Finally, the term "MACHINE" represents the transforming agent that performs an operation on a job in a shop. A shop can have one or more machines. The letter "M" is generally used to denote the number of machines in a shop.

If a shop has only one machine, the scheduling problem is sometimes known as "THE TRAVELING SALESMAN PROBLEM." A shop having two or more machines could be either a "FLOW SHOP" or a "JOB SHOP." The scheduling problem in these respective cases would be a "FLOW SHOP PROBLEM" and a "JOB SHOP PROBLEM." In a "FLOW SHOP" there is a natural ordering of M machines such that all the jobs that come to the shop to be processed go to machine 1 for their first operation, machine 2 for their second operation, . . . go to machine K for the Kth operation, where K is less than or equal to M. Thus a job in a "FLOW SHOP" with M machines has to go through exactly M number of operations, although a few of these operations may not be significant. In a "JOB SHOP" there is no natural ordering of machines and jobs, depending upon their technical nature, a job might go to any one machine for the first operation, another machine for the second operation, and so on. In the course of transformation, a job might not visit some machines and perhaps visit some machines more than once. Thus, in a "JOB SHOP" the number of operations a job goes through has no direct relationship with the number of machines in a shop.

A shop when operating usually processes many jobs simultaneously. The letter "N" generally denotes the number of jobs in a shop at a particular time. A shop processes jobs in a batch style or a continuous style depending upon how it is set up. When a shop is set up to process jobs in batch style, all of the N jobs enter the shop simultaneously

at a time $t_1$ and after processing, leave the shop simultaneously at a time $t_2$. Thus the shop has none of these N jobs just before the time $t_1$ or just after $t_2$, but exactly N jobs during the time period $t_1$ to $t_2$. This batch style of operation is known as "STATIC" operation since the number of jobs in a shop during a particular time period remains static.

Different from this batch operation is the case where the shop processes jobs in a continuous style. Jobs are allowed to enter individually at random times and to leave individually, when their own processing is completed. Thus, a shop has N number of jobs at any time where the value of N is fluctuating; going up, when a new job enters, and going down when a job which is processed leaves. Since the number of jobs in a shop which is operating on a continuous style fluctuate with time, it is known as a "DYNAMIC" operation.

In a shop each job undergoes many operations before it leaves the shop. Each of these operations, depending upon the job, the machine, and the amount of work involved in the operation, will require a certain amount of time. Transporting the job from one machine to another, cleaning and setting up the job and the machine will also require a certain amount of time. The total of the transportation, cleaning, setting up, and operation times is known as the "PROCESSING TIME" for the operation. The processing time generally is different for each operation and occasionally is dependent on the sequence of machines a job follows. In studying

scheduling problems, these processing times may be obtained from actual industrial situations or by sampling from theoretical probability distributions such as the Uniform, Normal, Lognormal, Exponential, Beta, and Gamma.

The usage of the two important terms "SCHEDULING" and "SEQUENCING" has created problems for many researchers. For many years some researchers used them arbitrarily as can be seen in the following excerpts:

> Sequencing Theory, when considered from an academic standpoint, is a combination of three specific areas within the field of operations research. The first is Sequencing, the second is Queueing or Dispatching, while the third is Scheduling.[3]

> It seems to be becoming fairly generally accepted that the central scheduling problem can be more usefully described as sequencing (or dispatching . . .).[4]

After some controversy, many people began to realize that no useful purpose is served by differentiating these two words. Furthermore, as expressed by Elmaghraby, Conway et al.:

> In many instances the word SEQUENCING is used synonymously with SCHEDULING. I would like to reserve the latter to the exact specification of the points in time at which certain events take place (similar to a train schedule). It is easy to see the reason for the frequent use by many authors of the two words interchangeably; a sequence also designates a schedule if the processing time of the first job on each facilitity is known and we are willing to

---

[3]A. H. Spinner, "Sequencing Theory--Development to Date," Naval Research Logistics Quarterly, Vol. 15, No. 2 (1968), 319-330.

[4]P. Mellor, "A Review of Job Shop Scheduling," Operation Research Quarterly, Vol. 17, No. 2 (1966), 161-171.

assume that each activity is started as early as possible. The two words will be used interchangeably in this paper, but only when these two conditions can be safely assumed.[5]

We considered drawing a fine distinction between Sequencing and Scheduling; to hold that the former is concerned only with the ordering of operations on a single machine, while the latter is a simultaneous and synchronized sequence on several machines. However, we found that no greater clarity resulted from such a distinction and the two terms are used essentially as synonyms in the following chapter.[6]

The American Heritage Dictionary defines scheduling as a production plan allocating work to be done and specifying deadlines, whereas sequencing is a following of one thing after another, succession.[7]

O'Brien, in his analysis of scheduling states that, "Scheduling involves the arrangement, coordination, and planning of the utilization of resources to achieve an objective."[8] O'Brien divides scheduling techniques into four classes: Time Scheduling, Resource Scheduling, Production Scheduling, and General Scheduling. Each of these four classes have anywhere from two to five separate techniques. He also gives three areas of scheduling applications, namely, Functional Scheduling, Topical Scheduling, and Production

---

[5] S. E. Elmaghraby, "The Machine Sequencing Problem--Review and Extension," Naval Research Logistics Quarterly, Vol. 15, No. 6 (1968), 205-232.

[6] Conway, Maxwell, and Miller, Theory of Scheduling op. cit.

[7] The American Heritage Dictionary of the English Language (Houghton-Mifflin Company, 1971).

[8] J. J. O'Brien, Scheduling Handbook (New York, New York: McGraw-Hill, 1969), p. 2.

applications. Each of these classes of applications have three to seven separate applications.

Thus it is clear that both terms, scheduling and sequencing, have far reaching meanings and a wide variety of applications.

Without putting forth any more arguments and illustrations, the two terms "SCHEDULING" and "SEQUENCING" will be used interchangeably in the remainder of this paper.

The ultimate aim of operating any shop is to maximize the profitability. Unfortunately, it is very difficult to determine precisely what variables will affect this profitability. Considering the need for customers and the availability of resources such as men, materials, and machines, some variables are chosen which conceivably can maximize the profitability of a shop. These variables are known as "THE MEASURES OF PERFORMANCE" or "THE PERFORMANCE CRITERIA." Then out of many possible alternatives of the scheduling problem, a solution is selected which optimizes the measure of performance. Numerous measures of performance are used in practice and research. Some of them include:

1. Finish the last job in the batch as soon as possible; that is, minimize the interval of time from start until finish of the batch processing. This measure is popularly known as the total make-span.
2. Finish each job as soon as possible, that is, minimize the sum of completion time of all

jobs, or minimize the mean of the completion time of N jobs.

3. Minimize the in-process inventory costs.
4. Minimize the costs that occur due to not meeting due dates exactly.
5. Minimize the distribution of lateness of jobs-- the length of time between the actual completion of a job and the desired completion.

It is inconceivable to assume that optimizing a single measure of performance will optimize the profitability of a shop. Attempts have been made to construct functions consisting of more than one measure of performance and then finding solutions to the scheduling problem to optimize value of the function. These attempts have been largely unsuccessful.[9]

## Statement of the Problem

The term "THEORY" implies a collection of systematically organized knowledge applicable in a relatively wide variety of circumstances; especially, a system of assumptions, accepted principles, and rules of procedure devised to analyze, predict, or otherwise explain the nature or behavior of a specified set of phenomena. Thus the "THEORY OF SCHEDULING" implies collection of knowledge as described above applicable to the scheduling problem.

---

[9] P. Meller, "A Review of Job Shop Scheduling," Operation Research Quarterly, Vol. 17, No. 2 (1966), 161-171.

This dissertation will be concerned with a small part of the scheduling theory. The problem and the reasons for selecting it are described below.

Flow Shop and Job Shop problems are closely related in the sense that the flow shop problem is considered a special case of the job shop problem. Many researchers have worked exclusively on the job shop problem, while others have worked on the job shop problem and applied their findings on the flow shop problem. Also, a few researchers have worked exclusively on the flow shop problem. Overall, the flow shop problem has received considerably less attention than the job shop problem. Surprisingly, the flow shop problem is not even mentioned, let alone discussed in the otherwise complete Scheduling Handbook written by James O'Brien in 1969. After my search for a suitable dissertation topic, I decided to work exclusively on the flow shop problem. It seemed clear to me then and more so now that there are many challenges and opportunities in the area of the flow shop scheduling problem.

After selecting the flow shop problem, I decided to work on its static operation rather than the dynamic operation. Traditionally, many of the researchers who have worked on the flow shop problem have chosen the static operation. The static operation is easier to work with, more basic and amenable to a mathematical formulation than a dynamic operation. Thus it is wiser to study the static operation first and then utilize the knowledge gained to understand the dynamic operation.

In 1954, S. M. Johnson provided a neat and simple algorithm to find an optimum sequence which minimizes the total make-span for a general N job, 2 machine and a special N job, 3 machine flow shop scheduling problem. He used fourteen assumptions and restrictions to define the flow shop scheduling problem and used the minimization of the total make-span as the performance criterion. This package of assumptions, restrictions, performance criterion and problem definition used by Johnson is used by many researchers; it will be referred to henceforth as the "classical flow shop scheduling problem."

For the research of this paper, I used all except one of Johnson's assumptions and restrictions. In this dissertation "Passing" is permitted in the solution of flow shop scheduling problems. Johnson assumed that a sequence of N jobs selected for the first machine of the flow shop should also be the sequence of the remaining M-1 machines. In other words, all of the M machines should have the same job sequence for processing N jobs. When a solution to a flow shop scheduling problem is such that it has different job sequences on different machines, then some jobs will bypass or "pass" over other jobs. Such a solution is not permitted under the no passing restriction used by Johnson, but is allowed when passing is permitted.

For whatever reason Johnson used the no passing restriction in his research on the classical flow shop scheduling problem, it was an intelligent decision. Later it was proved[10] that for the N job, 2 or 3 machine flow shop scheduling problem, the no passing restriction does not adversely affect the minimum value of the total make-span but effectively reduces the set of possible solutions, and in effect reduces efforts required for the search of an optimal solution. For example, under the no passing restriction for an N job, M machine flow shop problem there are only N! possible solutions, but if passing is permitted there are $(N!)^M$ possible solutions.

Since Johnson worked with only 2 and 3 machine flow shop scheduling problems, using the no passing restriction was a good choice for him. On the other hand, most of the researchers who have since worked on larger than 3 machine problems should not have used the no passing restriction. During preliminary research for this paper, it was discovered that for a 4 machine flow shop scheduling problem, removing the no passing restriction provides a lower value of the total make-span in some cases. On the basis of the preliminary work and the realization that very few researchers have attempted to remove the no passing restriction, I decided to study various effects of permitting passing

---

[10] Conway, Maxwell, and Miller, Theory of Scheduling, op. cit., p. 83.

between sequences in a flow shop scheduling problem for this dissertation.

The performance criterion I have chosen is minimization of the total make-span. In most of the research work on the static flow shop problem, two criteria are predominantly used, namely minimization of the total make-span and minimization of the mean completion time. Regarding this, Conway *et al*. states:

> Probably the most frequently cited paper in the field of scheduling is Johnson's solution to the two-machine flow shop problem. He gives an algorithm for sequencing n jobs, all simultaneously available, in a two-machine flow shop so as to minimize the maximum flow-time. This paper important, not only for its own content, but also for the influence it has had on subsequent work. In particular, it is likely that the general acceptance of minimizing the maximum flow-time as a criterion for the general job-shop problem can be attributed to Johnson's result.[11]

Following Johnson's lead and considering the prime advantage of easy comparability between solutions to the static flow shop problem, most of the researchers have used the total make-span criterion for their research.

The scheduling problem that emerges from the preceding discussion is defined as follows:

> Given N jobs to be processed on M machines in a flow shop and given processing times $t_{ij}$, for jobs i=1,2,...N, and for machines j=1,3,...M, the problem is to find the sequence of jobs on *each* of the M machines, so that the total make-span is minimized.

---

[11]Conway, Maxwell, and Miller, *Theory of Scheduling*, *op. cit.*, p. 83.

There are 14 assumptions and conditions that go with the problem definition:

1. All n jobs are simultaneously available at the beginning of planning period.
2. All jobs are of equal importance, i.e., no job has differential degrees of priority.
3. A single job cannot be processed simultaneously by more than one machine.
4. The processing time of each job is known and deterministic.
5. Set-up time and transportation time of a job is included in its processing time.
6. The processing time of a job is independent of the job sequence on any machine.
7. Every job requires processing on every machine and no job is processed more than once by any machine. If a certain job does not require processing on a particular machine, the corresponding operation will have a zero processing time.
8. There is no unexpected delay in processing. In other words, jobs are processed as soon as possible, subject to routing and sequencing requirements.
9. There is only one of each type of machine.

10. All M machines are available at the beginning of the planning period and are ready to take up any of the N jobs.
11. At most, only one job can be processed on a specific machine at any given time.
12. In-process inventory is allowable.
13. Each job follows the same machine sequence, i.e., each job goes to machine A first, then machine B, then machine C and so on.
14. No job splitting.

## Importance of the Problem

Looking at all those conditions and restrictions imposed on the problem listed in the problem definition section, one could assume that the way the problem is defined has no direct resemblance to any actual practice in the industry. Actually, the problem, even the way it is defined, has quite a bit of applicability in actual practice. This will be illustrated with some examples.

Example 1. A port with a nearby refinery has a docking facility for only one oil tanker. The tanker which brings crude oil to the port waits in the dock until the crude is refined and then carries the refined products back. The refinery tries to refine all the crude oil the tanker brings as soon as possible because keeping the tanker waiting for refined products and keeping the dock occupied with the tanker are both expensive. Moreover, other ships might be waiting to enter the dock.

The refinery has four chemical units which are similar to four machines in a flow shop. Crude has to be processed by all four units in a particular sequence as shown in Figure 1.1.

Usually a tanker delivers as few as three and as many as five grades of crude oil. Each of these grades of crude has to be processed separately and the processing times of each of the four chemical units differ from any other grade of crude. Each grade is tested in the laboratory first and its processing time on each of the chemical units is determined. When the testing of all the grades is completed, the processing time data are given to the scheduler. He then decides the sequence in which the grades of crude oil are to be processed in the refinery, to minimize the total make-span. The scheduler usually keeps the sequence of different grades of oil the same on all four chemical units, although if he wanted to, he could change it since the refinery has plenty of storage tank space for any in-process material.

Thus in this situation, the refinery is a flow shop with four machines. The number of jobs could be anywhere from three to five. The performance criterion is the minimization of the total make-span.

Example 2. This example is similar to the previous example. A company named Polychem makes polystyrene from styrene. Since the chemical styrene is in short supply, Polychem gets its limited varying daily quota at 8 a.m.

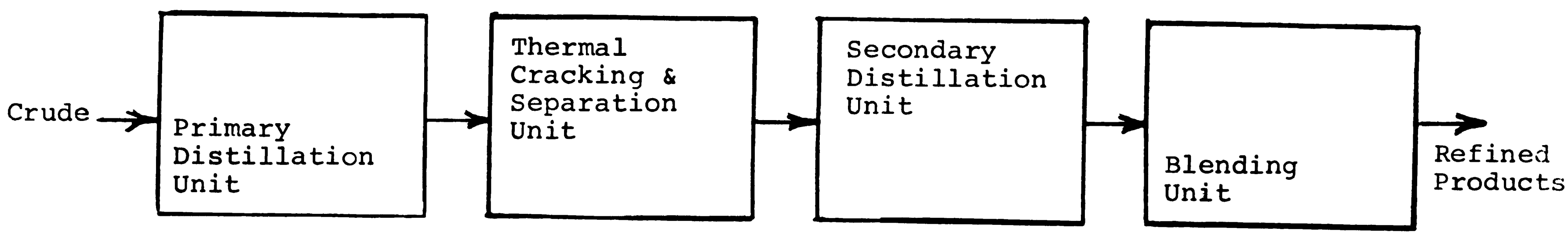


Figure 1.1 Refining of Crude Oil.

every day. The plastic polystyrene can be made in various grades of shape, color, strength and opacity. The conversation process from styrene to polystyrene needs processing on four chemical units as shown in Figure 1.2. As soon as the plant manager gets the information about the amount of styrene available for a particular day, he checks the pile of outstanding orders and decides which orders will be filled from that day's availability of styrene. He then calculates the processing time, temperature and pressure conditions for each of the four units for each of the orders. Since it is not advisable to store styrene overnight, it is all processed on the same day. The sooner it is done, the sooner all the employees can go home. Thus, in scheduling orders the performance criterion should be minimization of the total make-span.

Example 3. A dye company, El Chippo Dyes, has a steady demand for different color dyes. To minimize various production and storage costs the company manufactures only red, blue, yellow, and white dyes. Whenever an order for a particular dye arrives, the manager mixes these four basic colors in an appropriate amount and fills the order. As soon as one of the four basic dyes reaches a specified minimum level, production of all the basic dyes is triggered. The amount of production of the four dyes is decided on the basis of present inventory and future usage expectation. Each of the four basic dyes have to go through three vats in a particular sequence. The processing time of each

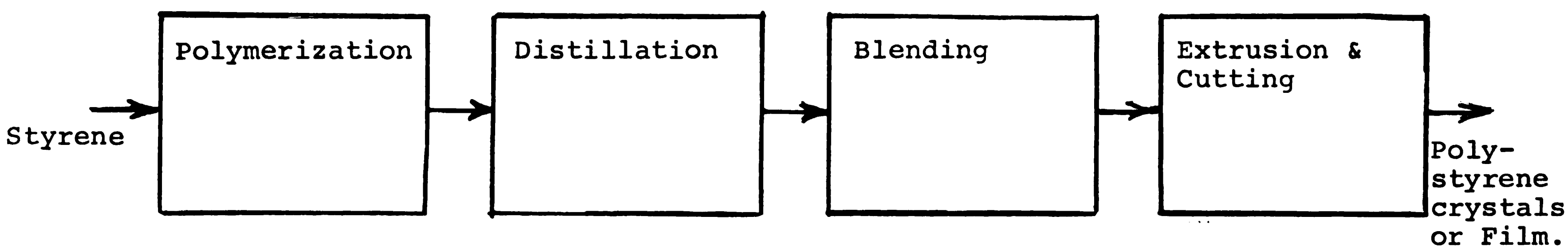


Figure 1.2 Polymerization of Styrene.

basic dye on each of the vats depends on the dye and the amount to be processed.

Since all four dyes are needed before an order is filled, the performance criterion is minimizing the total make-span.

All three examples given above show a static flow shop structure and present a definite need for rules, guidelines, and procedures to schedule jobs on machines to minimize the total make-span.

# CHAPTER II

## PAST RESEARCH ON THE STATIC FLOW SHOP SCHEDULING PROBLEM

There has been very little research work published in the area of the flow shop scheduling problem. It seems the widespread interest in developing the theory of the flow shop scheduling problem and its application in the industry has not materialized. Many experts of Management Science and Production-Operations Management consider the flow shop scheduling problem the stepchild of the job shop scheduling problem. They believe that once we have all the knowledge of the theory and application of the Job Shop Scheduling problem, the Flow Shop scheduling problem will be solved automatically. Perhaps this might happen, but I for one believe the Flow Shop scheduling problem is more amenable to analytical and optimizing techniques than the Job Shop scheduling problem. Pioneer research in the Flow Shop scheduling area and efforts done in the past to apply analytical and optimizing techniques to the Flow Shop are presented in this chapter.

Several mathematical techniques--combinatorial analysis, branch and bound technique, mathematical programming, heuristics, and Monte Carlo sampling--have been applied to solve the Static Flow shop scheduling problem.

## Combinatorial Analysis

The combinatorial optimization problem can be described in general terms as follows: Given a structure which can be arranged in a large but finite number of possible configurations--permulation or combination and some numerical data which allows a real valued cost or profit to be associated with each configuration, design the structure so as to optimize the cost or profit. In the static flow shop problem being discussed, the structure consists of N jobs, M machines (the configurations are partial or complete sequences of N jobs). The completion of processing time of these sequences on M machines are the costs that should be minimized. The solution procedure in the combinatorial analysis of the static flow shop problem involves changing one configuration or sequence of jobs to another by switching around jobs or choosing one job over others for a particular position in the configuration to reduce the total make-span, ultimately achieving the configuration that minimized the total make-span.

The work of Johnson (23), Dudek and Teuton (11), and Smith and Dudek (37) are prominent examples of combinatorial analysis applied to the classical flow shop problem.

Johnson's work (23), published in 1954, has proved to be a guidelight for the research done in the last twenty years. Although the procedure Johnson presented in that paper is applicable to finding solutions of only (i) the N job, two machine classical flow shop problem and (ii) the special case of N job, three machine classical flow shop problems, it is widely discussed in flow shop scheduling literature. Therefore, the procedure is described and explained in detail in the following pages.

S. M. Johnson

N Job, 2 Machine Problem

Johnson gave an algorithm based on the combinatorial analysis to find an optimum solution to any N job, 2 machine flow shop problem when the fourteen conditions listed on pages 14-15 of this dissertation are applicable and when the performance criterion is minimization of the total make-span. The algorithm given by him does not consider any "Passing" solution, i.e., any solution in which job sequences on different machines are different. Conway, Maxwell and Miller (8) proved that in a 2 or 3 machine flow shop problem with minimization of the total make-span as performance criterion, any "Passing" solution cannot be better than the best "Non-Passing" solution.

The necessary steps in this algorithm to achieve an optimum job sequence are listed and explained below.

1. Construct a table of the processing times $t_{ij}$, with columns for different jobs and rows for different machines or vice versa.

2. Construct two rectangular blocks, 1 and 2, in which you will put selected job numbers.

3. From the table of processing times, select the smallest processing time and note its job and machine number, e.g., $t_{ij}$, i and j. In case there is more than one, choose randomly.

4. If the machine number j is 1, put the job number i of that processing time in block 1, behind all the previously selected jobs of that block. If the machine number j is 2, put the job number 1 in block 2, ahead of all the previously selected jobs of that block.

5. Remove all processing times of the job i from the table of processing times.

6. If some processing times are left in the table, repeat steps 2-5.

7. If the processing time table has been exhausted, construct the optimum job sequence by putting the job sequence of block 1 ahead of the job sequence of block 2 and combine them.

Example: Processing time of a 6 job, 2 machine flow shop scheduling problem are given in Table 2.1.

In the processing time table, the smallest number is 1, corresponding to job 2, machine 1 and job 5, machine 2.

Table 2.1

Processing Time of 6 Job, 2 Machine Problem

| | Jobs | | | | | |
|---|---|---|---|---|---|---|
| Machines | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 7 | 1 | 2 | 10 | 9 | 7 |
| 2 | 3 | 9 | 15 | 6 | 1 | 4 |

Choosing randomly, job 2 goes to block 1 as the first number and job 5 goes to block 2 as the last number behind all future jobs.

| 2 Block 1 | 5 Block 2 |
|---|---|

Then the processing times of jobs 2 and 5 are removed from the table.

| | Jobs | | | |
|---|---|---|---|---|
| Machine | 1 | 3 | 4 | 6 |
| 1 | 7 | 2 | 10 | 7 |
| 2 | 3 | 15 | 6 | 4 |

Then the smallest number, 2, as processing time for job 3, machine 1 is selected and job number 3 is put in block 1 behind job number 2.

| 2 3 $\longrightarrow$ | $\longleftarrow$ 5 |
|---|---|
| Block 1 | Block 2 |

Repeating the steps of the algorithm, we will eventually have all six job numbers in two blocks as shown below:

| | |
|---|---|
| 2 3 | 4 6 1 5 |

The optimum sequence will then be "234615." The sequence 234615 is the best and optimum sequence out of the possible 6! = 720 sequences. The total make-span for it is 39 time units.

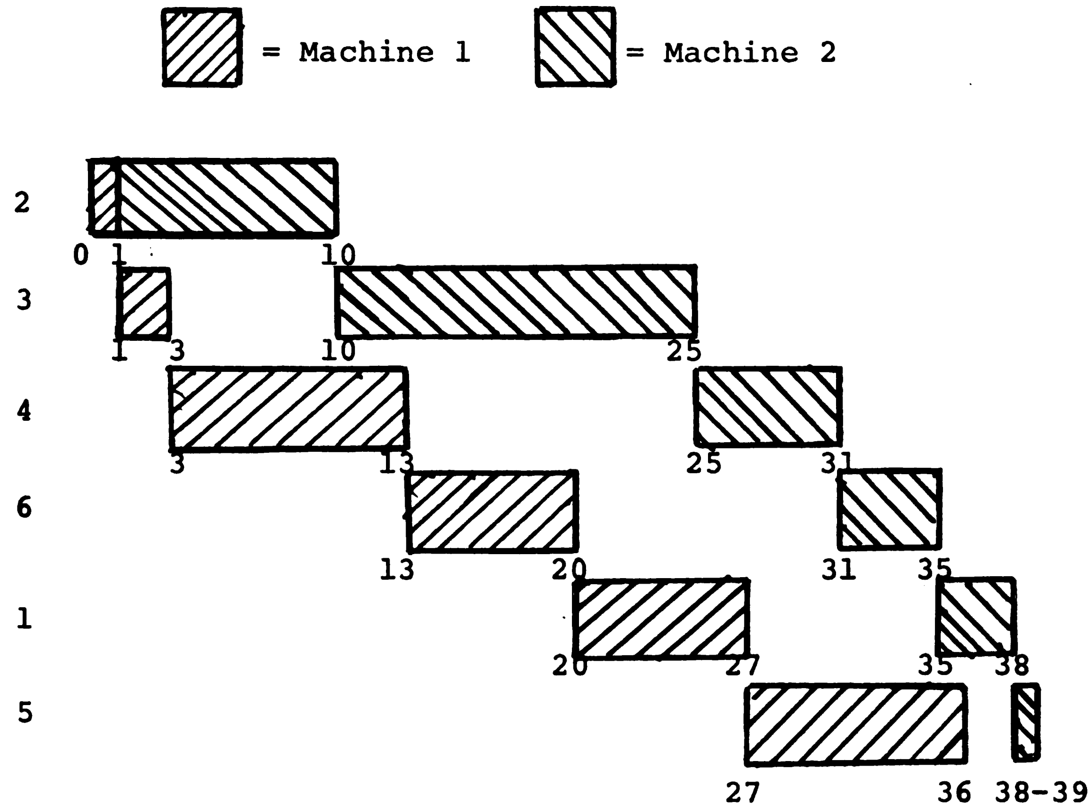


Figure 2.1. Gantt Chart for a 6 Job, 2 Machine Flow Shop Scheduling Problem.

Many researchers have attempted to extend the applicability of Johnson's N job, 2 machine flow shop algorithm to an N job, 3 machine or more than 3 machine flow shop problem. However, there has been only limited success. Johnson himself could not successfully extend the algorithm except for a special N job, 3 machine flow shop problem. The special case and the algorithm to obtain a solution for it are explained below.

N Job, 3 Machine Problem

The algorithm used for N job, 2 machine can be utilized for the N job, 3 machine problem only if all the previously-mentioned assumptions-conditions of N job, 2 machine are satisfied and the following conditions for processing times hold true.

$$\text{Maximum }[\text{Minimum }(t_{i1}),\ \text{Minimum }(t_{13})] \geq \text{Maximum }(t_{12})$$

In other words, the largest of all the processing times on machine 2 should be no bigger than the smallest of all the processing times on either machine 1 or machine 2 or both. If this condition holds for an N job, 3 machine problem, the algorithm that follows can be utilized to obtain an optimum solution to the problem.

1. The processing times of job 1 on machines 1 and 2 are added and the result is denoted as the processing time of job 1 on machine 1'. Similarly, the processing times of job 1 on machines 2 and 3 are added and the result is denoted

as the processing time of job 1 on machine 2'. The conversion process of processing times of machines 1, 2, and 3 to two machines 1' and 2' for job 1 is repeated for jobs 2, 3,. . . and N. As a result of the conversion, a table of processing times of N jobs, 2 machines is obtained.

$$t_{i1}' = t_{i1} + t_{i2} \; ; \; t_{i2}' = t_{i2} + t_{i3}$$

2. The algorithm presented earlier for N jobs, 2 machines is applied to the table of processing times obtained in Step 1, and the optimum sequence is computed.
3. The optimum sequence obtained in Step 2 is utilized for the original N jobs, 3 machines problem and the total make-span is computed. S. M. Johnson proved the above procedure provides a sequence with a minimum total make-span as compared to any other possible sequence for that particular flow shop problem.

   Example: An N job, 3 machine flow shop problem has processing times as shown in Table 2.2.

The condition is satisfied, so the problem is a special case to which the Johnson's 2 machine algorithm can be applied to compute the sequence which minimizes the total make-span.

First, by adding the processing times of machines 1 and 2, as well as 2 and 3, the processing times shown in Table 2.3 are obtained. Then the N job, 2 machine algorithm is applied to the above processing times table and the optimum sequence is determined.

Table 2.2

Processing Time of 6 Job, 3 Machine Problem

| | Jobs | | | | | | |
|---|---|---|---|---|---|---|---|
| Machines | 1 | 2 | 3 | 4 | 5 | 6 | |
| 1 | 6 | 7 | 14 | 5 | 10 | 7 | Min $(t_{i1})=5$ |
| 2 | 2 | 4 | 1 | 3 | 4 | 5 | Max $(t_{i2})=5$ |
| 3 | 3 | 10 | 8 | 7 | 9 | 2 | Min $(t_{i3})=2$ |

Maximum [Min $(t_{i1})$, Min $(t_{i3})$] $\geq$ Maximum $(t_{12})$

Maximum [ 5 , 2 ] $\geq$ 5

Table 2.3

Modified Processing Time of 6 Job, 3 Machine Problem

| | Jobs | | | | | |
|---|---|---|---|---|---|---|
| Machines | 1 | 2 | 3 | 4 | 5 | 6 |
| 1' | 8 | 11 | 15 | 8 | 14 | 12 |
| 2' | 5 | 10 | 9 | 10 | 13 | 7 |

$\boxed{4}$

Block 1

$\boxed{5 \quad 2 \quad 3 \quad 6 \quad 1}$

Block 2

The sequence "452361" for the original N job, 3 machine problem gives 54 time units as the total make-span. Out of the total 6! = 720 combination, this particular sequence "452361" gives the minimum total make-span.

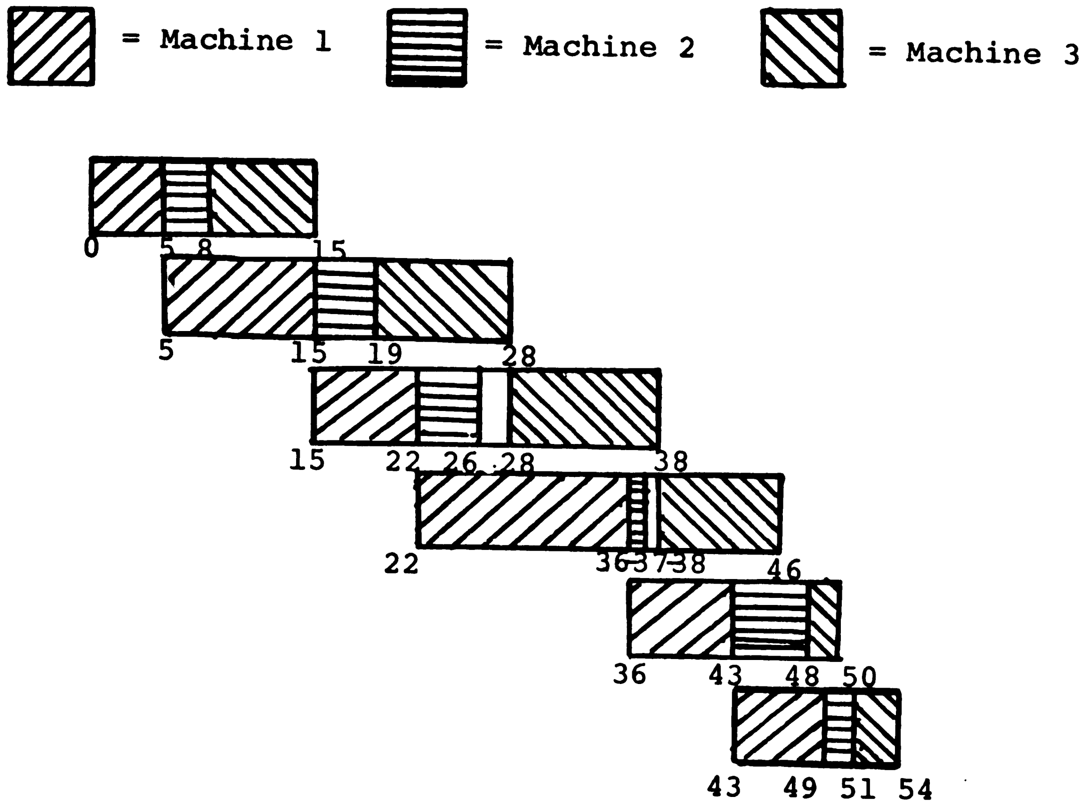


Figure 2.2 Gantt Chart for a 6 Job, 3 Machine Flow Shop Scheduling Problem.

Dudek and Teuton (11) worked on the N job, 3 machine classical flow shop problem, publishing their findings in 1964. In that paper they presented: 1) the theoretical development utilizing the combinatorial analysis to analyze the N job, M machine classical flow shop problem; 2) an algorithm to find a solution to the problem; and 3) an example and its solution. They claimed the algorithm will yield an optimum sequence to any N job, M machine classical flow shop problem. In response to the claim, William Karush (24), in a paper published in 1965, gave an example for which Dudek-Teuton's algorithm could not produce an optimal solution and showed that the claim of producing at least one optimum sequence to every N job, M machine classical flow shop problem by Dudek-Teuton's algorithm was not justified. Smith and Dudek (37), in a paper published in 1967, presented a modified Dudek-Teuton algorithm and gave theoretical proof that the modified algorithm would produce at least one optimum sequence to every N job, M machine classical flow shop problem. In the following pages Dudek-Teuton's algorithm, Karush's argument against the algorithm, and Smith-Dudek's modifications will be discussed.

## Dudek and Teuton

Dudek and Teuton's algorithm states the accumulated idle time on the last machine during the processing of N jobs is minimized, the total make-span will automatically be minimized. This is because:

Total make-span of N jobs on M machines = Total idle time on the Mth machine + $\sum_{i=1}^{N} t_{im}$

Where $t_{im}$ is the processing time of ith job on Mth (last) machine.

Therefore, Minimization (Total make-span)

= Minimization (Total idle time on the Mth machine)

$+ \sum_{i=1}^{N} t_{im}$ ;

the

obta

expl

sib

job

resp

sim

of

mul

the

for

sta

has

thi

fro

job

sys

defe

Since $\sum_{i=1}^{N} t_{im}$ is constant for a given problem.

Starting from the objective of the minimization of the total idle time on the Mth machine, Dudek and Teuton obtained an algorithm and a decision rule, which are explained below.

The set of N jobs is divided into two exclusive subsets, namely $\sigma$ and $\Pi$. The first subset $\sigma$ consists of J-1 jobs which are already scheduled for the first J-1 position respectively in the sequence of N positions. The second subset $\Pi$ consists of yet unscheduled jobs. At the beginning of the algorithm, $\sigma$ will be null and at the end $\Pi$ will be null. The algorithm in general consists of three steps: 1) choosing a job from the subset $\Pi$; 2) testing to see whether it meets the decision rule; and 3) scheduling it for the Jth position in the sequence.

Out of all the jobs in the subset $\Pi$, a job having the smallest sum of the processing times on the machine 1 through M-1 is chosen as the prospective candidate for the position J and is denoted by the sumbol 'A.' If more than one job has the smallest sum, the job having the maximum processing time on the Mth machine is chosen. Then the job A is removed from the subset $\Pi$, which is left with N-J jobs.

After deciding a job from the subset $\Pi$ as A, another job is chosen from $\Pi$ and denoted by the symbol 'B.' The symbol 'A' indicates the best prospective candidate or the defender for the position J in the sequence and the symbol

'B' indicates a challenger for the same position. One by one, each of the N-J jobs from $\Pi$, in any predetermined order takes its place as the challenger B against the defender A. To determine the superiority of A over B for the position J, two sequences S' and s" are constructed as S' = $\sigma$AB and S" = $\sigma$BA. The decision rule is applied to the two sequences S' and S" and to the processing times of the jobs in these sequences on the machines 1 through M. The decision rule consisting of M-1 inequality equations, M being the number of machines in the flow shop, is formulated by Dudek and Teuton from the initial objective of the minimization of the idle time on the last machine in the flow shop. These M-1 inequality equations would require too detailed a description to be explained here. For an accurate explanation, the reader should refer to the original article.

If S' gives the smaller idle time on the last machine compared to S" or, in other words, the decision rule indicates that S' is better than S", it means the job denoted by A is better suited than the job denoted by B for the position J in the sequence. Once this happens, a different job from $\Pi$ is denoted by B and the whole procedure of applying the decision rule on S' and S" is repeated. Now, if all of the jobs in $\Pi$ as B have "challenged" the job in A and have "lost," the job in A is assigned to the position J. The size of the subset and the value of J both are increased by 1, since one more job has been scheduled and a new job from $\Pi$ is chosen as the best prospective candidate for the for the next position in the sequence. Occasionally, a particular job in B makes the sequence S' inferior to S". In that case, the job in B replaces the job in A as the new prospective candidate and defender for the position J. The remaining jobs in Π will challenge the new defender one by one. When the new A weathers the challenge from the remaining jobs, it is assigned to the position J in the sequence. Sometimes for a particular job in A and a particular job in B, both S' and S" have equal results, indicating the job in A and the job in B are equally suitable for the position J. In this case, two σs are constructed, one σ having the job of defender and the other having a second job as defender. The remaining jobs in Π are that of challenger of the defender with the end result of occasionally more than two sequences with different jobs for the position J. If there is more than one sequence at the end of the algorithm, the choice of the optimal sequence is made on the basis of the minimum total make-span among those sequences.

Dudek and Teuton claimed that the algorithm produces at least one optimum sequence for the classical flow shop problem.

This paper describes an algorithm that will yield an optimum sequence for N jobs requiring processing through M machines when no passing is allowed.[1]

---

[1] R. A. Dudek, and O. F. Teuton, "Development of M-Stage Decision Rule for Scheduling N Jobs Through M Machines," Operations Research, Vol. 12, No. 3 (1964), p. 471.

William Karush gave a simple example of a 3 job, 3 machine classical flow shop.

| | Job | | |
|---|---|---|---|
| Machine | 1 | 2 | 3 |
| 1 | 3 | 22 | 20 |
| 2 | 22 | 20 | 14 |
| 3 | 2 | 20 | 18 |

When the Dudek-Teuton algorithm was applied to this problem, it gave an optimum sequence 1 2 3 with a total make-span of 83 time units. There are 3! = 6 possible sequences for the 3 job, 3 machine problem. Determining the total make-span for each of the six sequences indicates the minimum total make-span is 82 time units for the sequence 2 3 1. Thus the sequence produced by the Dudek-Teuton algorithm is not optimum.

The reason for this serious shortcoming of the Dudek-Teuton algorithm is that it compares the partial sequence $\sigma AB$ with $\sigma BA$ and concludes that if $\sigma AB$ is better than $\sigma BA$, the job A is more suitable in the position J irrespective of where the job B will be. Now, if $\sigma AB$ is better than $\sigma BA$, then $\sigma AB\ \Pi'\ \Pi''$ will be better than $\sigma BA\ \Pi'\ \Pi''$, but $\sigma A\ \Pi'\ B\ \Pi''$ may not be better than $\sigma B\ \Pi'\ A\ \Pi''$. The $\Pi'$ and $\Pi''$ are exclusive subsets of the subset $\Pi$. This means that although $\sigma AB$ is better than $\sigma BA$, A is not necessarily better suited for the position J as compared to B. If the Dudek-Teuton algorithm is applied to Karush's example, the partial sequence 1 2 will be better than 2 1 and the partial sequence 1 3 will be better than 3 1. Therefore, Dudek and Teuton would conclude that job 1 is better suited for position 1 as compared to jobs 2 and 3. The fallacy in this conclusion is that just because 1 2 is better than 2 1, it does not mean that 1 3 2 is better than 2 3 1. In fact, the total make-span of the sequence 1 3 2 is 85 whereas for the sequence 2 3 1, it is only 82.

There is another disadvantage in the Dudek-Teuton algorithm--the computational time requirement on the computer. Dudek and Teuton, in the same paper, reported that for a 3 machine, 3 job or 4 job and 5 machine, 3 job, 4 job, or 5 job problem, their algorithm would take approximately <u>three times as much time</u> as the complete enumeration would take on an IBM 1620 computer. Only for a 3 machine, 5 job, 6 job, 6 job or 7 job problem would their algorithm save time as compared to the complete enumeration. Thus, it seems clear that for a 3-5 machine problem it would be much better to attempt complete enumeration rather than the Dudek-Teuton algorithm.

## Smith and Dudek

Two years after Karush pointed out the fallacy in the Dudek-Teuton algorithm, Smith and Dudek in their paper (37) presented an improved algorithm correcting the fallacy. They redefined the two sequences S' and S" as $S' = \sigma AB\,\Pi'\,\Pi''$

and S" = $\sigma$B $\Pi$'a$\Pi$", where $\Pi$' and $\Pi$" represent any possible exclusive subsets of $\Pi$. The algorithm utilizing the decision rule determines whether S' is better than S" for each of the B's from the subset $\Pi$ and for each of the possible permutations of $\Pi$' and $\Pi$". If the decision rule indicates S' is indeed better than S", A is scheduled for the position J. Otherwise, for any one or more B's, S' is not better than S" and A and those B's are each scheduled for the position J, creating that many possible sequences. The decision rule, which involves M-1 inequality equations, is too technical to describe here.

Smith and Dudek in their algorithm demonstrated the computational time required to do 3 or 5 machine, and up to 8 job problems is almost half as much as would be required by the complete enumeration. Although this is quite an improvement over the Dudek-Teuton algorithm requirement of computational time, it is not as advanced as the computational time requirement of the Branch and Bound algorithm. J. N. D. Gupta in his paper (18), published in 1969, indicated the Branch and Bound algorithm given by McMohan and Burton (28), requires about 50 to 90 percent less time than the Smith-Dudek algorithm. Thus, it seems more than 2 machine classical flow shop problems, the computational time requirement of the algorithm of combinatorial analysis might be too great compared to other methods.

## Branch and Bound Method

As previously discussed, the combinatorial analysis gives an optimal solution for the N job, 3 or more machine classical flow shop problem, but the computational time requirement is very large. The Branch and Bound method can give optimal solutions to the classical flow shop problem with a reasonable computational time requirement.

As explained by Conway, Maxwell, and Miller (8, pp. 56) the Branch and Bound technique first used by Little, Murty, Sweeny, and Karel is an ingenious recursive computational procedure. The procedure involves the maintenance of a list of unsolved, closely-related problems and three processing routines to apply to the problems on the list. Initially, one places the original problem on the list. The original problem is then processed (and hence, removed from the list), by one of the routines, which will sometimes create other problems to be placed on the list. One simply continues applying these same routines to each problem until there are no more on the list, at which point the solution to the original problem has been revealed. The three routines are:

1. A solution routine, which directly solves a problem from the list if the problem is easy (small) enough.
2. An elimination routine, which discards a problem from the list if it can show that

the problem can make no contribution to the solution of the original problem.

3. A partitioning routine which replaces a problem too difficult to solve with several related subproblems.

For the classical flow shop problem in the Branch and Bound procedure, the set of all possible sequences is progressively divided into subsets in a systematic manner by "branching." After every division the newly-created subsets must be mutually exclusive and exhaustive. A partial sequence (one with less than N jobs sequences) is used to identify the subset of all possible sequences arising from the partial sequence. A lower bound on each subset is determined in such a way that the total make-span for any sequence in the subset is greater than or equal to this bound. After every branching, the subset with the lowest bound is chosen for further exploration. The procedure terminates when a complete sequence is obtained for which the total make-span is equal to or less than the lower bounds on all unexplored subsets. The complete sequence is then optimal.

At present, there are five published papers which show the utilization of the Branch and Bound technique to obtain optimal solutions to the classical flow shop problem. They are: Lomnicki (27); Ignall and Schrage (22); Brown and Lomnicki (6); McMahon and Burton (28); and Gupta (18). Only two of these five--that of Ignall and Schrage and one by McMahon and Burton--are useful enough to be discussed here in detail. The remaining three papers combine the information of the other two papers and thus will be mentioned only briefly.

Ignall and Schrage

A paper published in 1965 in Ignall and Schrage (22) was the first to clearly exemplify the applicability of the Branch and Bound technique for the classical flow shop problem. The objective function they used to compute the lower bound for each of the nodes in the tree is simple and easy to use. The objective function for the lower bound for a partial or complete sequence JS is as follows:

$$LB\ (JS) = \text{Max} \left[ \begin{array}{l} \text{Time } A(JS) + \sum\limits_{NJS} ai + \text{Min}_{NJS}\ (bi + Ci); \\ \text{Time } B(JS) + \sum\limits_{NJS} bi + \text{Min}_{NJS}\ (Ci); \\ \text{Time } C(JS) + \sum\limits_{NJS} Ci; \end{array} \right]$$

The time A (JS), B (JS), and C (JS) are the times at which machines A, B, and C, respectively, complete processing on the last job of the sequence JS. The term 'NJS' indicates the set of all jobs which are not on the sequence JS. The terms ai, bi, and ci represent the processing times of the ith job on machines A, B, and C respectively.

In comparison with the complete enumeration, the Branch and Bound technique is faster. The computational time and effort for the Branch and Bound technique is directly proportional to the number of nodes generated for a problem. For an N job, M machine classical flow problem, in the most difficult case, there will be 1 + N + N (N-1) + . . . + N! nodes. In the best possible case there will be 1 + 2 + . . . + N = 1/2 N (N + 1) nodes. For a 10 job problem solved by the Branch and Bound technique, at worst there will be 6,235,301 nodes and at best there will be 55 nodes. Using the objective function they devised, Ignall and Schrage worked 50 of the 10 job, 3 machine problems and found that on an average, each problem needs generation of only 212 nodes and 8.7 seconds on a CDC 1604 computer to arrive at an optimum solution. Since Ignall and Schrage did not make any comparisons of their method with other methods it is not possible to state definitely how efficient their method is. If a 10 job, 3 machine problem was done using the complete enumeration, there would have been a need for generation of 10! = 3,628,000 sequences. The generation of each sequence and computation of its make-span would require roughly the same amount of time as the generation of a node and computation of its lower bound. Therefore, the complete enumeration for one 10 job, 3 machine would require approximately $\frac{3628800}{212}$ * 8.7 seconds or 41.3 hours on CDC 1604. Although the complete enumeration would likely generate more than one optimum sequence, compared to only one optimum sequence by the Branch and Bound technique, the difference in computation times--41.3 hours compared to 8.7

seconds--is too great to consider any other factors. Actually, there seems to be no need for more than one optimum sequence. Thus the work of Ignall and Schrage appears quite impressive and useful for the classical flow shop problem.

Besides giving an objective function for the lower bound, explaining the Branch and Bound technique and the results for the previously published jobsets of Giglio and Wagner (15), Ignall and Schrage discussed the concept of dominated nodes in their work. They explained that if JS and IS are two different sequences on two different nodes, both containing the same set of jobs, then one of these two sequences may be dominated by the other sequence. For example, the sequence JS is such that the last job in it is completed on each of the machines A, B, . . . M at the same time or before the last of the jobs of IS on each of the machines A through M, thus the sequence JS is dominating the sequence IS. Any schedule which contains IS at the beginning cannot be hurt by replacing IS by JS. Thus even though the lower bound on the node with the IS as the partial sequence might be lower than the lower bound of the node with JS, it will not be necessary to branch from IS, and the node with IS can easily be discarded from the list of nodes that are under active consideration.

In the example that Ignall and Schrage used to illustrate the Branch and Bound technique, the lower bound of the partial sequence 1, as well as partial sequence 2, were both 55. [LB (1) = LB (2) = 55.] Therefore, it was necessary to branch from 1 as well as 2. The LB (12) was equal to 55 and LB (21) was 56. However, the sequence 12 was dominated by the sequence 21, so it was not necessary to branch from 12 so it was discarded even though LB (12) was smaller than LB (21).

Lomnicki, Brown and Gupta

The paper by Limnicki (27) was published in Great Britain at almost the same time as the paper by Ignall and Schrage in the United States. Both of these papers published in 1965 are based on the work of Little, Murty, Sweeny, and Karel,[2] On the traveling salesman problem, although Lomnicki also made use of Roy's Graph-theoretical interpretation of the job shop published in France in 1962.[3] Even though the objective function to find the lower bound used by Lomnicki is exactly the same as that of Ignall and Schrage, it is very difficult to see any similarity between them. The deceiving differences are caused by the use of unusual and strange notations by Lomnicki. In his paper Lomnicki briefly mentions a concept of a reversed order of

---

[2]Little, Murty, Sweeney, and Karel, "An Algorithm for the Traveling Salesman Problem," Operations Research, 11, 972-989, (1963).

[3]B. Roy, Cheminement et connexite dans les graphes - Applications aux problemes d'ordinnancement. Metra, serie speciale No. 1, Societe d'economie et de mathematiques appliquees, Paris. (1962).

machine

is reve

sequen

job se

choice

requir

solver

two or

result

compu

sugge

Burto

effec

1966

year.

Lonn

in s

Lonn

3 ma

Lonn

it

prob

is

cov

als

app

machines. If the order of the M machines A, B, C, . . . M is reversed to M, L, J . . . A for a problem, the optimum sequence obtained will be the exact reverse of the original job sequence. The advantage of this concept lies in the choice it offers the problem solver. One of these two ways requires less effort and computation time. If the problem solver can determine for a particular problem which of the two orders of machines--the original or the reverse--would result in the smaller number of nodes generated and less computation time, a lot of time can be saved. Lomnicki suggested the concept but did not use it. McMahon and Burton, whose work is discussed a little later, used it effectively.

The paper by Brown and Lomnicki (6) published in 1966 is an extension of Lomnicki's work of the previous year. Brown and Lomnicki gave a simple explanation of Lomnicki's previous work, presenting the objective function in slightly better form. The objective functions given by Lomnicki as well as Ignall and Schrage are particularly for 3 machine classical flow shop problems only. Brown and Lomnicki extended the objective function of Lomnicki so that it can be useful for more than 3 machine classical flow shop problems. The objective function given by Ignall and Schrage is in a fairly simple form so that anyone can extend it to cover more than a 3 machine problem. Brown and Lomnicki also gave some experimental results and analysis of the application of the Branch and Bound technique to some

problems and gave an algorithm in a form that can be used if the Branch and Bound technique is to be programmed into a computer.

J. N. D. Gupta published a paper (18) giving an algorithm to obtain an optimum to the classical flow shop scheduling problem. According to Gupta, the algorithm is based on the lexicographic search concept. The American Heritage Dictionary defines lexicography as the writing or compilation of a dictionary. What Gupta seems to be doing is "compiling a word from appropriate letters" or appropriately compiling a sequence from jobs. Actually, the algorithm Gupta presented in 1969 is no different from the earlier Branch and Bound technique. The objective function of Gupta can easily be put into the following form:

$$LB\ (JS) = \text{Time}\ M(JS) + \sum_{NJS} Mi$$

Once in this form, it is easy to see the function is the same as the last part of Ignall and Schrage's objective function with M instead of C. The M is the last machine in the flow shop. Other notations are the same as the one used for Ignall and Schrage, as well as Brown and Lomnicki, in the previous few pages.

Gupta compared the computation time his algorithm required on IBM 7040 for 3 to 7 jobs and 4, 6 and 8 machines with the computation times of Brown and Limnicki and Smith and Dudek. His algorithm requires about 10 to 20 percent less time than Smith and Dudek's algorithm. Thus, if nothing else, Gupta demonstrates Smith and Dudek's algorithm is inefficient compared to that of Brown and Lomnicki.

McMahon and Burton

In a paper published in 1967, McMahon and Burton presented some new concepts which are useful in applying the Branch and Bound technique to classical flow shop problems. The bounds that are obtained from the objective functions given by Ignall and Schrage, Lomnicki and Brown, and Lomnicki, are called "machine-based bounds" by McMahon and Burton. Against these "machine-based bounds," their own bounds from a different objective function are referred to as "job-based bounds." The objective function for job-based bounds is as follows:

$$LB(JS) = \underset{MAX}{} \begin{bmatrix} \text{Time } A(JS) + \text{Max}_{NJS} [a_i + b_i + c_i + {}_{NK}\sum \text{Min } (a_k, c_k)]; \\ \text{Time } B(JS) + \text{Max}_{NJS} [b_i + c_i + {}_{NK}\sum \text{Min } (b_k, c_k)]; \\ \text{Time } C(JS) + \sum_{NJS} c_i \ ; \end{bmatrix}$$

The LB(JS) is the lower bound for any node with a partial or complete sequence JS. Time A (JS), Time B (JS) and Time C (JS) are the completion times of the last job in the sequence, JS on the machines A, B, and C, respectively. The notations $a_i$, $b_i$ and $c_i$ are processing times of job i on the machines A, B and C, respectively. The NJS represents a job set containing jobs that are not yet assigned to the sequence JS. The term NK represents a set of jobs in the set NJS, minus the job i. The subscripts i and k represent one job from the jobset of NJS and NK, respectively. The $a_k$, $b_k$, and $c_k$ are processing times of the job K on machines A, B, and C, respectively.

The composite bound of McMahon and Burton is the bound equal to the greater of the machine-based bound and the job-based bound.

Composite bound = Max [Machine-based bound, Job-based bound]

It would have been nice if McMahon and Burton had experimented to determine the relative efficiency of machine-based bounds versus job-based bounds. It would have proved the superiority of one over the other. Instead, McMahon and Burton experimented with the classical flow shop problems of Branch and Bound technique using the machine-based bounds of Ignall and Schrage versus their own composite bounds. They concluded the composite bound is far better than machine-based bounds. They solved 50 problems each for the 4 to 10 jobs, 3 machine classical flow shop problems by the Branch and Bound technique, using both machine-based bounds and composite bounds. The mean number of nodes generated reduced from machine-based bounds to composite bounds by 22 percent, 28 percent, 39 percent, 40 percent, 48 percent, 40 percent, and 37 percent, respectively. The reduction in the mean computation times were 20 percent, 31 percent, 55 percent, 82 percent, 78 percent, 75 percent and 82 percent, respectively, for the same seven sets of 50 problems.

A significant reduction in the number of nodes generated and the computation time should convince anyone of the superiority of composite bounds for the Branch and Bound technique over machine-based bounds. On the other hand, it raises a question. If for each node generated the calculation of the lower bound by composite bounds requires almost double the amount of calculations required by machine-based bounds, (for composite bounds it is necessary to compute machine-based bounds as well as job-based bounds), how can the percentage reduction in the computation time from machine-based to composite bounds be greater than the percentage reduction in the number of nodes generated? For instance, for the 50 problems of the 10 job, 3 machine case, the average nodes generated decreased by 37 percent from machine-based to composite bounds. However, the computation time went down a whole 82 percent. Common sense suggests the percentage reduction in the total computation time should be less than the reduction in the total number of nodes generated, if each node generated required more computation in composite bounds than in machine-based bounds. In six out of seven sets of those 50 problems, the reduction in computation time is much greater than the reduction in the number of nodes generated. It seems McMahon and Burton should have tried to examine and explain this obvious discrepancy, but unfortunately they did not.

In the algorithm McMahon and Burton skillfully used the concept of reversed order. To determine which of the two orders, the original or the reversed should be used to solve the problem, they introduced a concept of the dominant machine. Of the two machines A and C, the one which has the greater total processing time is referred to as the dominant machine. From experiments of 50 problems for 4 to 10 jobs, they determined that the total number of nodes generated and the total computation time required is much less if the dominant machine is last compared to the number of nodes and computation times required if the dominant machine is first. Thus, between the two machines, A and C, if C is dominant, then the original order of the machines, i.e., ABC is used, but if the machine A is dominant, then the reversed order CBA should be used. In case the reversed order is used, the optimum sequence obtained by the Branch and Bound technique should be reversed to obtain the optimum sequence for the original problem.

## Mathematical Programming Approach

During the years 1958-1965 there was great interest among researchers in solving various scheduling problems using many mathematical programming techniques. Techniques like Linear Programming, Integer Programming, and Dynamic Programming were repeatedly used for the traveling salesman, job shop scheduling and flow shop scheduling problems. The work done to find a solution to the job shop was mainly carried out by Wagner in 1959 (42), Bowman in 1959 (5), Manne in 1960 (29, Chapter 12) and Giffler and Thompson in 1960 (14). Wagner in 1959 (42), Story and Wagner in 1961 (40), Giglio and Wagner in 1964 (15) specifically worked on solving flow shop scheduling problems using the integer linear programming technique.

In spite of the sophisticated techniques and inventiveness of the researchers, the mathematical programming approach did not prove a viable means of efficiently solving flow shop scheduling problems.

Wagner

Wagner in his paper (42), derived and presented an integer, linear programming model capable of representing the job shop scheduling problem. In the second section of his paper, he modified his model to specifically represent the flow shop scheduling problem. It is interesting to note that the model for the flow shop is capable of handling a problem even when the restriction of the same job sequence on different machines is removed. In other words, even when "no passing" or "no switching" restrictions are removed, the model for the flow shop can be used. In the third section, Wagner simplifies the model even further so that it can be used when there are only three machines in the flow shop and passing or switching is not necessary. Thus, Wagner has given three levels of the integer linear programming model, first for the job shop, then for the flow shop, and finally for the 3 machine classical flow shop problem. He does not attempt to present a solution to any problem in that paper, although some of his comments indicate a difficulty of solving a problem with his models.

"The model in its present form is computationally unwieldy except perhaps for situations with a very few machines and a limited number of items (jobs)." Thus, we have derived a fundamental system of the order of 4*N equations which probably can be solved for $N \leq 25$ on high-speed computing machinery; but it of course remains questionable whether or not such a computational proposal for finding an optimal solution is economically sound." (The above comment is in reference to the three machine classical flow shop problem.)

Story and Wagner

Story and Wagner (40) reported experiments on 4 to 9 jobs, 3 machine classical flow shop problems. First, they attempted to obtain optimum solutions to the 4 job, 3 machine classical flow shop problem by a few heuristic techniques like: a) Pairwise exchanges; b) end around cycling; and c) order reversal of jobs; starting from an arbitrary sequence. None of the three heuristics helped them to achieve the optimum solution. At this point, they set out to experiment with the N job, 3 machine classical flow shop given by Wagner (42). The integer programming algorithm they used was based on the approach suggested by R. Gomory,[4] and the integer programming package IP03 available for the IBM 7090 computer, by IBM Corporation. They arbitrarily constructed six problems each for 4 to 9 jobs, 3 machine classical flow shops. If they failed to obtain a solution to any problem in less than 1,000 iterations they terminated computations for that problem. For six 4 job, 3 machine problems, they obtained optimum solutions at an average of 119 iterations per problem. If they would have attempted to find optimum solutions to the 4 job, 3 machine problems they would have to do only twenty-four permutations per problem, instead of 119 iterations. Besides, the amount of computations in an iteration is much more than the amount of computation in a permutation. This fact in itself shows how inefficient integer programming can be in solving classical flow shop problems. Because of the thousand iterations limit on problems they could obtain optimum solutions to only 5, 4, 3, 2, and 1 problems out of six problems each for the 5 job, 6 job, 7 job, 8 job and 9 job, 3 machine classical flow shop cases. After this dismal performance story, Wagner also attempted to solve some 4 job, 3 machine problems by Gomory's mixed-integer programming method with very little success.[5]

---

[4] R. E. Gomory, "An All-Integer Programming Algorithm," Chapter 13, in the reference, 29.

[5] R. E. Gomory, "Mixed Integer Programming Algorithm," Unpublished Rand Memorandum, RM2597, 1960.

> Our conclusion is evident from the tests: We have not yet found an integer programming method that can be relied upon to solve most machine sequencing problems rapidly.

Giglio and Wagner

In their paper (15) of 1964, Giglio and Wagner tested integer linear programming, ordinary linear programming with answers rounded to integers, a heuristic algorithm and random sampling of six of the previously-used (40) 6 job, 3 machine classical flow shop problems. The heuristic algorithm and random sampling methods will be discussed later on in this chapter.

Giglio and Wagner applied Gomory's all-integer programming algorithm available as SHARE PKIP91 on IBM 7090 to the 3 machine flow shop model given by Wagner (42).[6] Giglio and Wagner in their paper (15, page 310) state that "previous experience with integer programming problems has demonstrated that the convergence properties of the algorithm are highly sensitive to the form in which the problem is stated initially. Consequently, we examined the effect on convergence of specifying six different input formats." The main differences among the six input formats were few extra constrains, or the presence of a lower bound on the objective function. They attempted to solve all six problems using each of the six different starting input formats

[6] R. E. Gomory, "An All-Integer Integer Programming Algorithm," Chapter 13 in the reference 29.

and restricting the maximum number of iterations to 10,000. They presented the actual number of iterations required for each of the 6 x 6 problems and Conway, Maxwell, and Miller (8, pp. 96) commented on the number of iterations by saying:

> This data does not appear very encouraging. In many cases the number of iterations is of the same order of magnitude as the number of possible permutations (720) and an iteration involves more computations than the generation of a permutation. However, they do confirm that the exact form in which the problem is stated has an important effect on the efficiency of the algorithm and further work is being directed at the development of more efficient constrains and bounds. At least, at the moment, the Branch and Bound technique appears to have a substantial advantage over integer programming as a practical computational procedure for the problem.

In the second part of their testing Giglio and Wagner generated a hundred 6 job, 3 machine problems. The processing times were randomly picked from the uniform distribution between 1 and 30, inclusively. For each of the 100 problems, they calculated the total make-span for each of the 6! = 720 permutations, computed the mean total make-span and noted the minimum make-span. Then they found the best sequences for each of those 100 problems by using the algorithm of linear programming on the model presented by Wagner (42), removing the restriction of the integer values only. The answers were rounded to the nearest integers to obtain the best sequences and the total make-span was computed for them. When Giglio and Wagner compared the total make-span obtained by linear programming with the

make-spans obtained by complete enumeration of each of the 100 problems, they found the linear programming make-spans were overall only a little better than the mean make-spans. If the minimum, mean, and maximum make-spans are given ranks of 1, 360 and 720, the best make-span obtained by linear programming would achieve an overall rank of 206. This result is far from satisfactory and Giglio and Wagner reported:

> In summary, we can conclude that the rounding process tends to produce solutions that are better than could be obtained from a single permutation drawn at random, but the method by no means produces optimal or nearly optimal solutions with a high degree of success.

## Heuristics

A heuristic is a rule of thumb which may have little, if any, theoretical foundation, but is found to generate very good, although not necessarily optimum, solutions. The advantages of a heuristic algorithm are that it would give a very good solution with relatively little computational time. In the classical flow shop scheduling area many researchers have given some of the best heuristics, Giglio and Wagner (15), Palmer (33), Campbell, Dudek and Smith (7) and Gupta (16) are some of them.

### Giglio and Wagner

In the previously mentioned paper (15) Giglio and Wagner provide a heuristic for N job, 3 machine classical flow shop problems. The algorithm is quite simple and gives reliable solutions. The steps of the algorithm are as follows:

For any N job, 3 machine problem the processing times of machine A and machine B for each of the N jobs are added and the sum is referred to as the processing time of machine A' for that particular job. Similarly, the processing times for machine B and machine C for each of the N jobs are added and referred to as the processing time of machine B' for that job. Finally, there will be two processing times for each of the N jobs. Johnson's algorithm for N job, 2 machine is applied over here and a sequence is obtained which should be a very close approximation to the optimum sequence for that problem.

Giglio and Wagner applied this heuristic on 20 problems whose processing times were randomly distributed from the uniform distribution over 1 to 30 both inclusive. In nine of the 20 problems, an optimum solution was produced and in eight of the remaining cases the solution produced was only one interchange away from the optimum. The average total make-span by heuristic was 1317 compared to the average minimum make-span of 1279. Thus, the heuristic seems to be quite useful.

Palmer

Palmer in 1965 presented a heuristic (33) for the N job, M machine classical flow shop problem. Considering the fact that this was a first attempt at devising a heuristic for more than a three machine flow shop, it is very efficient.

The algorithm is based on the principle of giving priority to jobs whose processing times increase with the increase in the machine number. The algorithm simply involves calculating a quantity called the slope index for each of the N jobs.

Slope index
for job i = Si =

$$-\frac{M-1}{2}t_{i1} - \frac{M-3}{2}t_{i2} - \frac{M-5}{2}t_{i3} \cdot \cdot \cdot + \frac{M-3}{2}t_{im-1} + \frac{M-1}{2}t_{im}$$

The jobs are then sequenced in the decreasing order of the slopes.

Palmer randomly generated 4 to 9 job, 3 machine problems and applied his heuristic on them. His solutions were very close to the optimum solutions. The sum of all six minimum make-spans was 418 units whereas the sum of the six make-spans by Palmer's heuristic was 437.

Giglio and Wagner (15) had used six problems for a linear programming algorithm. To check the comparative effectiveness of the heuristics of Giglio and Wagner with the heuristics of Palmer, I applied both heuristics to the six problems. The sum of the minimum make-spans for the optimum solutions was 397. Palmer's heuristic gave a make-span sum of 417 whereas Giglio and Wagner's heuristic resulted in 418. Thus it appears that for a very small

sample, both heuristics are almost equally efficient.
Palmer himself concluded his paper in this way:

> All general results must be approximations subject to the influence of single items or processes not being too great; they are upset if the schedule is dominated by one item or one process.

Campbell, Dudek and Smith

Campbell, Dudek and Smith in a paper (7) published in 1970 presented a reliable heuristic which has been fairly well received and is frequently reported by other researchers in the classical flow shop field. The heuristic involves converting an N job, 2 machine size which can be solved by using Johnson's N job, 2 machine algorithm.

The conversion of one problem of N job, M machine size to M-1 auxiliary problems of N job, 2 machine size is very simple. The original problem has processing times in N rows and M columns. Each of the auxiliary problems has two columns of processing times for the two machines and N rows for the N jobs. The first column of the processing times of the Kth auxiliary problem is constructed by adding row by row the processing times of the first K machines of the original problem. The second column of the same Kth auxiliary problem is constructed by adding row by row the processing times of the last K machines of the original problem. The value of K will progressively increase from 1 to M-1 for the M-1 auxiliary problems. For the first auxiliary problem, the first and second columns will be the same as the first and last column of the original problem.

For each of the M-1 auxiliary problems the optimum sequence and the minimum make-span is computed by using Johnson's N job, 2 machine algorithm. The lowest make-span from the M-1 make-spans and the corresponding N job sequence is noted. The sequence is chosen as the heuristic approximation of the optimum sequence and the make-span for the original problem using that sequence is computed.

These three men adopted a policy for error calculation on the basis of the percentage deviation of the make-span of the heuristic sequence from the optimum make-span.

$$\text{Error} = \frac{\text{Heuristic make span - Optimum make span}}{\text{Optimum make span}}$$

Then Campbell, Dudek and Smith generated integer random numbers from a uniform distribution between 01 to 99 for processing times of 340 problems ranging in size from 3 job, 3 machine to 8 job, 5 machines. For each of the 340 problems they computed the optimum solutions, solutions by their heuristic, and the solutions by Palmer's heuristics. They also calculated the percentage of errors and noted computation time required for both heuristics. The average amount of error percentages for their heuristics were about 2 percent compared to about 4.5 percent for Palmer's heuristics. The computation times for Campbell's heuristic were anywhere from two to three times the time required for

Palmer's heuristic. Thus it seems clear that Campbell, Dudek and Smith's algorithm provides closer approximations to the optimum solutions. The computation time comparison between their heuristic and Palmer's does not give a lot of insight except that Palmer's heuristic demands half or one-third as much time as Campbell's heuristic. A computation time comparison of Campbell, et al.'s heuristic, Palmer's heuristic, and any algorithm for optimum solutions, would have been beneficial.

Gupta

In 1971 Gupta published a paper (16) presenting a simple heuristic for the N job, M machine classical flow shop problem. This heuristic involves much less calculation than either Campbell, Dudek, and Smith's or Palmer's heuristics. In the heuristic, a function of (i) is calculated for each of the N jobs and the heuristic sequence is obtained by arranging jobs in the increasing order of the value of the function.

$$f\,(i) = \frac{A}{\underset{1 \le K \le M-1}{\mathrm{Min}} \left(t_{iK} + t_{iK+1}\right)}$$

where $A = +1$ if $t_{im} \le t_{i1}$, where $A = -1$ otherwise; Gupta used the same error function used by Campbell, Dudek, and Smith. He generated processing times from the uniform distribution between 0 and 999 from the 195 problems of 4 to 8 jobs, 4 to 8 machines classical flow shop problem.

For each of these 195 problems, Gupta computed the optimum solution, minimum make-span, the heuristic solution, and its make-span by his heuristic and by Palmer's heuristic. He also calculated error and computation time required by both his and Palmer's heuristics. The average error percentages of Gupta's heuristic ranged anywhere from 3 to 10.5 percent but Palmer's heuristic error percentage ranged from seven to twenty-two. The computation time required for Gupta's heuristic was on an average about 10 percent less than for Palmer's heuristic. Thus Gupta's heuristic seems to be better on both counts--closeness to optimum and computation time. However, an important question arises from the data presented by Gupta. Why did Campbell, Dudek and Smith find (7) on an average only 4.5 percent error with Palmer's heuristic, while Gupta found seven to 22 percent error with it when he used the same error function and similar size problems as Campbell and others?

## Monte Carlo Sampling

In many decision-making situations the Monte Carlo Sampling has proved to be a very useful and efficient technique to decide between many alternatives. Although it does not necessarily provide the best alternative, if used properly it can provide an excellent choice. On the other hand, if analytical methods are available to make a decision then the Monte Carlo technique might not measure up to the analytical methods in terms of optimality of the solution.

Heller in 1960 had published a paper showing some use of the Monte Carlo Sampling technique to find a good solution to the flow shop scheduling problem.

Heller

In 1960, Heller published a paper (19) showing the results of his experiments with a 100 job, 10 machine and a 20 job, 10 machine problem. From his results he made two conclusions. (1) The total make-spans are (approximately) normally distributed for a large number of jobs. This characteristic, together with the cost of sampling and the cost due to the amount of suboptimization, can be used to determine the size of the sample. And (2) that if we were looking for the minimum total make-span, we would sample (only) from those schedules which have the same ordering on all machines. These two conclusions, although erroneous, appeared to make sense to many of the other researchers at that time. Besides these erroneous conclusions there is another inaccuracy in the paper.

Heller generated a total of 1000 integer processing times distributed between 0 and 9 for the 100 job, 10 machine problem. He gave the impression that these 1000 numbers came from uniform distribution. Conway, Maxwell and Miller stated, "He generated a set of 100 jobs with integer processing times apparently uniformly distributed between zero and nine." (8, pp. 100.) Unfortunately, the distribution of where the numbers came from seems far from uniform and similar to a bell shape. This stems from the fact that there are only 31 zeros and 62 nines in the 1000 numbers, whereas there are 191 fives. (If the distribution were uniform, the expected number of zeros, nines, and fives should all be 100.)

Heller, for his experiment on the 100 job, 10 machine problem did 3000 trials. For each trial he randomly chose a permutation of 100 jobs, assumed this permutation as the job sequence on all of the 10 machines, and computed the total make-span. He drew a graph of make-spans versus the probability of obtaining them. The range of make-spans was 580 to 725 units. The shape of this graph was bell-like and thus he assumed that the make-spans are approximately normally distributed for large numbers of jobs. The conclusion of normality is not justified for two reasons, one of which is given by Conway, Maxwell, and Miller (8, pp. 100).

> If the distribution of schedule times (make spans) is approximately, and/or asymptotically, normal, the departures from normality will be most pronounced in the tails of the distribution, which is precisely the area of interest. No one is concerned with estimating the mean of the schedule-time distribution for a particular problem.

The second reason was that his sample size is only 3000, whereas the population size is 100! or approximately $1.26*10.^{157}$

For the second problem, the 20 job, 10 machine one, he used the processing times of the first 20 jobs from the 100 job problem. He examined and experimented with this problem in two different parts. In the first part he had the same job sequence on all of the 10 machines whereas in the second part he selected different job sequences on each of the 10 machines.

In the first part of the study for each trial he randomly generated a permutation of 20 jobs, used it as a job sequence for all the machines, computed the total make span and did 12,000 trials. In the second part, for each trial he randomly generated 10 different permutations of 20 jobs for each of the 10 machines and computed the total make-span. He repeated the procedure for 9037 trials. The make-spans versus the probability of obtaining it were drawn for both of the parts on the same paper using the same scale. Graphs for both parts were bell-shaped with the mean make-spans 169.93 and 727.82, respectively. The sample variances were 1.24 and 32.14, respectively, for the first and second part. Both of the graphs were far apart without a single point in common. From this observation Heller concluded that if the interest is in obtaining the minimum make-span, it is worthless to sample from the population of the second part.

The fallacy of the above conclusion is not difficult to understand. The population size of the first part (same job order or sequence on all machines) is 20! or approximately $2.43*10^{17}$ and the population size of the second part (different job order on each of the 10 machines) is $(20!)^{10}$ or approximately $7.24*10^{173}$. The first part population is also a part of the second part population. In fact, if a member from the second part is randomly selected there is a probability of $10^{-10}$ that it might be a member of the first one. With 9037 trials, there is a probability of .0000009037 that a solution could be the member that might be found in the first as well as the second parts. Obviously the probability is very small and Heller did not hit upon it. Another fact is that there are some members in the second part which are not members of the first part but still have lower make-spans than any member of the first part. With the total population of $7.24*10^{173}$ and sample size of only 9037 there is very little probability of finding any unusual member. (It is a fact that very few members of the second part might have a smaller make-span than the lowest make-span member of the first part.) In fact, Nugent working on the Monte Carlo Sampling for job shop scheduling examined the 20 job, 10 machine problem. Considering it as a special case of job shop scheduling, he applied his "probabilistic priority rules" and found a total make-span of 144 with slightly different job sequences on different machines. Nugent's work definitely proved that Heller's conclusion was erroneous.

To examine how a simple heuristic would perform, I applied Palmer's heuristic to Heller's 20 machine, 10 job problem. In half an hour's work, without the help of any electronic instrument, a (no passing) sequence was obtained which gave a total make-span of 156. Adding the processing times of the last five machines for each of the 20 jobs, then applying Johnson's algorithm of the N job, 2 machine case, a sequence was obtained. This second sequence, although quite different from the first, also resulted in a total make-span of 156. The second trial also took approximately thirty minutes. Heller had obtained make-spans between 150 and 200 for the problem doing 12,000 trials. Thus it is obvious that if some analytical or even heuristic techniques are available, the Monte Carlo Sampling technique would not be competitive.

## Non-Classical Flow Shop

There has been a fair amount of work done in the non-classical flow shop scheduling area (1, 2, 9, 19, 20, 21, 25, 26, 34). Out of these, only two, one by Heller (19, 20, 21), and the other by Krone (25, 26) would serve some useful purpose to launch a lengthy discussion here. Heller's classical and non-classical work is discussed in detail in the Monte Carlo Sampling section.

### Krone and Steiglitz

Krone and Steiglitz published a paper (26) in 1974 essentially based on the doctoral dissertation work of Krone (25) which he did in the department of electric engineering at Princeton University in 1970.

In the flow shop scheduling area, Krone has chosen a very difficult performance measure, the minimization of the mean completion time. The measure is seldom mentioned let alone used by many researchers in the flow shop scheduling area. The only other work done with the mean completion time on the N job, 2 machine flow shop is by Ignall and Schrage (22). There are two facts which show the amount of difficulty working with the minimization of the mean completion time. The first is that even for a 2 machine flow shop problem there is no simple algorithm to find an optimum solution if the performance measure is minimization of the mean completion time. Johnson's 2 machine procedure does not give an algorithm solution, nor according to Conway, Maxwell and Miller, does it produce good results. The second fact is that if the performance measure is minimization of the mean completion time, the 2 machine case is the only one where examining different sequences on different machines (passing) is not necessary. For the 3 machine case, passing between the first and second machines is not necessary, but between the second and third machines passing might produce better results than non passing. Thus for an N job 3 machine flow shop problem, the total number of permutations are $(N!)^2$ with minimization of the mean completion time whereas with the minimization of the total make-span only N! permutations need be examined. These two facts should convince anyone that the minimization of the mean completion time is a more difficult performance measure to work with than the minimization of the total make-span.

In his research, Krone finds the final solution for an N job, M machine flow shop problem with minimization of the mean completion time as the performance measure in two phases. In the first phase of his algorithm, he begins with a sequence of jobs which is the same for all M machines and is obtained by generating a pseudorandom starting solution. He then progressively improves the sequence by applying a local transformation or switching of jobs until the sequence can no longer be imporved. He calls this sequence the locally optimum sequence. He calls the local transformation a "perturbation" rule. The restriction of no passing is maintained throughout phase 1. (However, it is released in Phase 2.) "The program terminates when local optimality of the current trial solution is verified by searching its neighborhood exhaustively."

In the second phase he starts out with the locally optimum sequence he obtained in the first phase. The same "perturbation" rule is used to change the sequences of only some of the machines if the change improves the value of the mean completion time. In other words, he attempts to release the no passing restriction of phase one to further improve the value of the performance measure. When the "neighborhood is searched exhaustively" the algorithm in the second phase is completed.

For experimental work, Krone generated integer numbers uniformly distributed between 0 and 100, sufficient for fifty problems of 10 job, 5 machine size. For each of these fifty problems, he executed his two-phase algorithm twenty times. Since the final solution of his algorithm is dependent upon the initial solution, and since everytime he executed his algorithm he obtained a different initial solution for phase one, he obtained twenty intermediate solutions (solutions at the end of phase one), and twenty final solutions (solutions at the end of phase two). For each problem, he chose a best final solution and "normalized" the twenty intermediate and twenty final solutions assigning a value of 1.0 to the best final solution. By following this procedure he obtained 1000 intermediate and 1000 final solutions with at least fifty of the final solutions having a value of 1.0. He plotted a histogram for his 1000 intermediate and 1000 final solutions. The values of the intermediate solutions varied between 1.02 to 1.5 and the values of the final solutions varied between 1.0 and 1.05.

From his experience with these fifty problems and with a few other problems, he concluded that all the minimization of the mean flow time is accomplished in phase one, with phase two of the optimization process providing only slight improvement in general. Overall, he found an improvement of 0.66 percent in phase two.

From the wide range of intermediate solutions (1.02 to 1.50) Krone obtained a much narrowed (1.00 to 1.05) range of final solutions. Based on this fact and that both the intermediate and the final solution histograms start at almost the same values (1.02 versus 1.00) it seems fairly clear that a lot of "bad" intermediate solutions improve a great amount, but a lot of "good" intermediate solutions do not improve significantly during phase two. One could conclude then that if the task of finding locally optimum solutions in phase one is accomplished successfully, the task of improving them in phase two may not be necessary.

As mentioned earlier in the Branch and Bound section, Ignall and Schrage's algorithm to obtain optimum solutions for a 3 machine flow shop can easily be extended in cases of more than 3 machine flow shops (no passing) if the performance measure is minimization of the total makespan. This extension will be useful in this dissertation to obtain optimum solutions in the first phase. Ignall and Schrage also presented an algorithm to obtain an optimum solution for an N job, 2 machine flow shop when the performance measure is minimization of the mean completion time. Krone probably could have extended this algorithm (based on the Branch and Bound technique) to obtain solutions for more than a 2 machine flow shop (no passing) with the performance measure of the mean completion time.

There are some important similarities between Krone's dissertation work and this particular dissertation. Until all the conceptual work and experimental studies were done for this dissertation, Krone's work had not been examined. Both dissertations are motivated by a similar need, the critical evaluation of the effects of the removal of no passing restrictions from the N job, M machine flow shop. Both dissertations have an algorithm in two phases. The first phase solutions to flow shop under the no passing restriction are obtained and studied. In the second phase, the solutions of the first phase are improved upon by removing the no passing restrictions. At this point, the similarities end. As discussed earlier, Krone uses a more difficult performance measure than the one used for this dissertation. Another difference is in the quality of the solutions obtained in the first phase. Krone obtains locally optimum solutions whereas in this dissertation globally optimum solutions will be obtained in the first phase. In Krone's work, very little is achieved in the second phase. The efforts in the second phase are wasted on improving the "bad" solutions of the first phase. In this paper, although efforts in the second phase might seem greater than the achievements obtained, extra effort will not be useless. These efforts will bring definite results.

# CHAPTER III

## RESEARCH METHODOLOGY

This dissertation will attempt to examine the effects of removing the No Passing restriction from the classical flow shop problems. The implication of the No Passing restriction is that the same sequence of jobs should be used on different machines of the flow shop. In other words, once a job sequence is chosen for the first machine, the same job sequence is utilized for all subsequent machines and no job is allowed to "pass" over another job or jobs. In the last twenty years, all except for a few researchers have assumed that for the flow shop there is no need to change the job sequence from one machine to another. They must have assumed that any job "passing" could only delay and in no way improve the total make-span. During preliminary research with one hundred of 4 job, 4 machine problems for this dissertation, it became clear that in approximately 10-15 percent of the problems, the optimum solutions obtained under the No Passing restriction could be improved to a small extent by removing the restriction and permitting judicious differences between job sequences of the first two machines and the last two machines.

Conway, Maxwell and Miller provided proof that for a flow shop with the minimization of the total make-span as the performance measure, changing the job sequence between the first and second machines as well as changing the job sequence between the second to last and last machines is not necessary, and doing it will not improve the optimum solution (8, p. 81). This implies that for a two machine and a three machine flow shop, considering different job sequences on different machines is not necessary. For a four machine flow shop, "passing" between the first and second as well as the third and the fourth is not necessary. However, "passing" between the second and third machines can prove beneficial as far as the optimum values of the solution is concerned. With the no passing restriction on any size flow shop, N! permutations need be examined, but with Passing permitted $(N!)^{M-2}$ permutations $(M \geq 3)$ should be examined.

The number of permutations to be examined in a flow shop when passing is permitted increases very rapidly. For example, for a 4 machine flow shop with 4, 5 and 6 jobs, the total number of permutations are 576; 14,400; and 518,400. Thus, when the number of jobs in a 4 machine shop increases from 4 to 6, the total number of permutations increases from 576 to half a million. No wonder many researchers who are aware of the benefit of passing, hesitate to work with it.

The IBM 360 computer requires roughly about six to seven minutes to do a complete enumeration for a 6 job, 4 machine flow shop problem with passing permitted. Taking this tremendous computational burden into account, it was decided to work with 3 to 5 job, 4 machine flow shop problems for this dissertation. The various results obtained here will be applicable to larger flow shop problems once some trends in results are established, going from 3 job to 5 job problems.

The procedure used to obtain solutions for any problem in this dissertation is divided into two phases. In the first phase, optimum solutions are obtained under No Passing restrictions. In the second phase, first phase solutions are improved by permitting Passing between the second and third machines of the N job, 4 machine flow shop problem.

## Phase One

Two different algorithms are employed for finding optimum solutions in the first phase. One method employs the complete enumeration while the other uses a modified Branch and Bound procedure.

### Complete Enumeration

Although few other algorithms are available to obtain optimum solutions under the No Passing restriction, the complete enumeration algorithm is used for experiments reported in the fourth chapter of this dissertation. There are some advantages in using the complete enumeration for finding solutions to classical flow shop problems. For one, it not only gives the minimum total make-span for a problem, but if needed, provides a maximum total make-span and the complete frequency distribution of total make-spans for a problem. The range between the minimum and maximum total make-spans or frequency distribution of the total make-spans can be very useful in experimentation. Another advantage of complete enumeration is that if there is more than one sequence having the same minimum total make-span, complete enumeration can provide all of them. To obtain a satisfactory solution from a second phase algorithm, it is necessary that all optimum solutions of a problem in the first phase are known. The Branch and Bound technique normally provides only one optimum solution although there might be more than one present, and thus has to be modified to cause it to produce all of the optimum solutions a problem has. One disadvantage of the complete enumeration algorithm is that it requires more computation time as compared to most other algorithms. The computer problem used for the complete enumeration algorithm in this dissertation is explained below and listed in Appendix A.

The complete enumeration computer program consists of a main problem and three subroutines: SCHED, FCALCI, and FCALCZ. The input required for the main program are the number of jobs N, the number of machines M, and the total

N*M processing times for a problem. The main program first determines the value of N! and then for each of the natural positive integers between 1 and N! calls the subroutine SCHED to generate an appropriate sequence of N jobs. For example, for a 4 job, 4 machine flow shop, the following table shows the number between 1 and N! and their corresponding sequences. For each of the N! sequences that the

Table 3.1

Numbers in Sequences for the 4 Job, 4 Machine Flow Shop Scheduling Problem

| Number | Sequences |
|---|---|
| 1 | 1234 |
| 2 | 1243 |
| 3 | 1324 |
| 4 | 1342 |
| 5 | 1423 |
| 6 | 1432 |
| 7 | 2134 |
| " | " |
| " | " |
| " | " |
| 24 | 4321 |

main program has obtained from the subroutine SCHED, it calls the subroutines FCALC1 and FCALCZ to obtain the completion times of different operations of the N jobs on the 4 machines. The input for both the subroutines, FCALC1 and FCALCZ, are the N*4 processing times and the sequence generated by the subroutine SCHED. The subroutine FCALCZ also needs as an input the output of FCALC1. The subroutine FCALC1 computes the completion time of operations of N jobs on the first two machines and FCALCZ computes the completion times of N jobs on the last two machines. Although a single subroutine could easily do the job of the two subroutines, FCALC1 and FCALCZ, in the first phase, for the sake of computational efficiency, usage of the separate subroutines is necessary in the second phase where the job sequences are different on the first two machines from the last two machines. The total make-span for the particular sequence is equal to the completion time of the last (Nth) operation on the fourth machine from the subroutine FCALCZ.

Modified Branch and Bound Algorithm

For the first phase of experiments in Chapter V, the Branch and Bound algorithm used is essentially the same as the one used by Ignall and Schrage, with three important modifications.

The first modification is the extension of Ignall and Schrage's lower bound function of the 3 machine classical flow shop to the lower bound function for the 4 machine classical flow shop. Ignall and Schrage's function was present in the second chapter of this dissertation; the modified lower bound function is given below:

$$LB(JS) = \text{Max} \begin{bmatrix} \text{Time } A(JS) + \sum\limits_{NJS} a_i + \text{Min}\limits_{NJS} (b_i + c_i + d_i); \\ \\ \text{Time } B(JS) + \sum\limits_{NJS} b_i + \text{Min}\limits_{NJS} (C_i + d_i); \\ \\ \text{Time } C(JS) + \sum\limits_{NJS} c_i + \text{Min}\limits_{NJS} (d_i); \\ \\ \text{Time } D(JS) + \sum\limits_{NJS} d_i; \end{bmatrix}$$

Where JS is the set of jobs sequences corresponding to the node for which LB(JS) is the lower bound. Time A(JS), Time B(JS), Time C(Js) and Time D(JS) are the completion times of the last job in thessequence JS on machines A, B, C, and D, respectively. The terms $a_i$, $b_i$, $c_i$, and $d_i$ represent the processing times of job i on machines A. B, C, and D, respectively. The jobset NJS includes jobs which are not sequences in JS.

The second modification can save a great amount of computational time and effort when the Branch and Bound technique is used for large flow shop problems. The modification is based on the work done for the traveling salesman problem by Dannenbring (unpublished) and reported by Starr (39). It involves computing an upper bound to the problem before the tree branching and nodes creation are begun. If an upper bound to the total make-span is obtained, then any node which has a lower bound greater than the upper bound can easily be disposed of without branching from it. Various ways can be devised to obtain an upper bound. However, the upper bound should have a fairly low value so that it can serve some useful purpose without involving a lot of computational effort to find it. After some experimentation, a simple technique to compute an upper bound for this dissertation was decided. The technique is an extension of Giglio and Wagner's 3 machine heuristic technique (15) and involves part of Campbell, Dudek and Smith's heuristic algorithm (7). This technique involves adding the processing times of the first two machines and adding the processing times of the third and the fourth machines for each of the N jobs. Thus an N job, 4 machine problem is temporarily converted to an N job, 2 machine problem. Johnson's 2 machine algorithm can be applied to the problem and an "optimum" sequence obtained. The "optimum" sequence is used with the original N job, 4 machine problem and the total make-span computed. The sequence obtained by Johnson's algorithm will probably not be optimum for the N job, 4 machine problem, but some experience with the technique suggests that in most cases it will be quite close to the optimum. Thus, the total make-span obtained can serve quite well as the upper bound for the problem.

The third modification involves obtaining all the optimum sequences for a problem instead of just one from the Branch and Bound technique. While branching and computing lower bounds for nodes, if more than one node having the same lower bound was found, Ignall and Schrage chose one at random for further branching. At the termination, if any unbranched node or nodes having the same lower bound as the minimum total make-span were present, they were left as they were. Instead, in the Branch and Bound technique used in this dissertation, once an optimum sequence and its total make-span are obtained, all nodes having the same lower bound as the minimum total make-span are branched until all optimum sequences are found or until all the unbranched nodes have a higher lower-bound than the minimum total make-span.

The computer program for the modified Branch and Bound algorithm consists of a main program and five subroutines, namely, COMPUTE, SEND, BOUND, INTRO, and DESTRO. The Branch and Bound technique in general, and the modified Ignall and Schrage lower bound function, in particular, are programmed in such a way that they can be used for up to 20 job, 10 machine classical flow shop problems. The main program reads in the values of the number of jobs, number of machines, and the processing times. It outputs the minimum total make-span and all of the optimum sequences for a problem under the No Passing restriction. The subroutine COMPUTE computes an upper bound for a problem (using Johnson's algorithm). The subroutine SEND selects the most promising node from the list of available codes, branches out from it and creates nodes for the branches. The subroutine BOUND calculates the lower bound for all the newly created nodes. If in the future some other function for calculating lower bounds is to be used, it is necessary to change only the subroutine BOUND. The subroutine INTRO inspects all newly-computed lower bounds and compares them to the upper bound, and the corresponding node is introduced to the list of available nodes. The subroutine DESTRO searches for any node that is fully branched and has served its purpose, then eliminates it from the list of available nodes. Once a complete sequence emerges from the subroutine INTRO, the subroutine SEND reduces the value of the upper bound to the total make-span of the complete sequence. The complete listing of the computer program is given in Appendix B.

## Phase Two

### Standard of Measurement of Passing

After preliminary experiments, a need to define and measure the amount of passing between two sequences of a passing solution has emerged. The standard of measurement is named "SHIFT" and the amount of Passing is measured as 1 shift, 2 shift, et cetera. When a job which is in the Kth ($K<N$) position in the first sequence has completed its processing on the first set of machines and is ready to be processed on the next machine, L, ($L<M$), even though machine L is also ready and available for processing the job, both job and machine are made to wait. Then when the

K + lth job of the first sequence becomes available for processing on the Lth machine, it is permitted to be processed on it, resulting in 1 shift of Passing between the two job sequences. Here the measure "1 shift" is an indication of one job taken out its first sequence and made to wait. The measure "2 shifts" would indicate the presence of Passing such that two of the N jobs, when ready for processing on a machine, are made to wait until other jobs behind them in the first sequence finish processing on the machine. In a four machine flow shop, the jobs will wait for processing on the third machine. The important thing to understand is that it does not matter how much or how long a particular job is made to wait as long as it is pushed behind in the second sequence by at least one position.

Some examples of 1 shift of Passing between two sequences are shown below:

| | Example 1 | Example 2 | Example 3 |
|---|---|---|---|
| First Sequence | 2 1 3 4 | 2 1 3 4 | 2 1 3 4 |
| Second Sequence | 2 3 1 4 | 2 3 4 1 | 2 1 4 3 |
| | 1 Shift | 1 Shift | 1 Shift |
| | Example 4 | Example 5 | |
| | 3 1 2 4 | 3 1 4 2 | |
| | 1 2 4 3 | 1 3 4 2 | |
| | 1 Shift | 1 Shift | |

Some examples of 2 shifts are given below. The two jobs that are made to wait, may or may not be the two consecutive jobs in the first sequence.

```
                    Example 1        Example 2
First Sequence      3 4 1 2          3 4 1 2
Second Sequence         1 3 4 2          1 2 3 4

                    Example 3        Example 4
                    3 4 1 2          3 4 1 2
                      4   2 1 3        4 3 2 1
                     2 Shift          2 Shift
```

In a Passing solution with two sequences of N jobs, there could be a maximum of N-1 Shift of Passing. Each of the following two examples show 3 Shift Passing between the two 4 job sequences.

```
                    Example 1        Example 2
First Sequence      1 2 3 4          1 2 3 4
Second Sequence           4 1 2 3          4 3 2 1
                     3 Shift          3 Shift
```

For a Passing solution with two N job sequences, given the first sequence, there are N! possible permutations for the second sequence. Out of these N! permutations, one would be exactly the same as the first sequence, resulting in a 0 Shift Passing between the two sequences. After a few calculations, it was determined that there are (N-1)! permutations or sequences out of the total N! permutations, each of which, if placed as the second sequence, would give 1 Shift Passing. There are (N-1)! more permutations, each when used as the second sequence, giving N-1 Shift Passing. Each of the remaining permutations N!-1-(N-1)!-(N-1)! will give more than one but less than N-1 Shift Passing if placed as the second sequence in a two sequence Passing solution. For example, for a given 4 job sequence "1 4 3 2," there are a total 4!=24 possible permutations or sequences which could be placed as the second sequence. One of them will give 0 Shift Passing. 3!=6 of them will give 1 Shift Passing, 3!=6 more sequences will give 3 Shift Passing, and 24-1-6-6- = 11 sequences will give a 2 Shift Passing if used as the second sequence with the first sequence "1 4 3 2."

## Preliminary Observations

As stated before, during experimental work with 4 job, 4 machine problems, approximately ten to fifteen percent of the problems provided lower total make-span values with passing allowed than with the No Passing restriction. As the optimum sequences obtained under No Passing and the better total make-span valued solutions under Passing were studied in detail for those ten to fifteen percent problems, the following phenomena were observed:

(i) For most problems there was more than one sequence providing the same minimum total make-span values under the No Passing restriction.

(ii) For the most problems there was more than one Passing solution having better values of the total make-span than the minimum total make-span obtained under the No Passing restriction. The amount of improvement the different Passing solutions showed over the No Passing solutions varied to some extent.

(iii) Each Passing solution of the 4 machine flow shop problem consists of two different job sequences, one being the sequence for the first two machines and the other being the sequence for the last two machines. In most of the Passing solutions that showed improvement over the minimum total make-span of the No Passing solutions, one of the two sequences matched exactly with one of the optimum No Passing sequences.

(iv) The differences between two sequences of a solution which improved the total make-span over the minimum total make-span under No Passing were minimal. Most of the 4 job, 4 machine Passing solutions that improved over the No Passing solutions had 1 Shift Passing only.

If the above preliminary observations are indeed typical of the underlying phenomena, they could prove to be more useful in obtaining lower total make-spans than the optimum total make-spans obtained under the No Passing restriction, without resorting to the complete enumeration of all the Passing permutations.

In the second phase of the experiment, complete enumeration of all the passing solutions with a passing amount equal to 1 through N-1 Shifts for each of the flow shop scheduling problems is undertaken. For an N job, 4 machine flow shop problem, there are (N!)*(N!-1) passing solutions. The total make-span for each of these solutions is obtained and compared with the minimum total make-span obtained in the first phase. Any passing solution that has a lower total make-span than the minimum total make-span of the first phase is saved. The amount of passing is measured for each passing solution saved and both sequences of each passing solution are compared with the optimum sequences obtained in the first phase of that problem. All similarities in the sequences are noted.

The computer program for the complete enumeration in the second phase is similar to the one used for the complete enumeration in the first phase. The main program calls the subroutine SCHED twice for each of the passing solutions to get two sequences--one for the first two machines and the other for the second two machines. The subroutine FCALC1 computes the completion times of various operations on the first two machines and the subroutine FCALCZ computes the completion times on the last two machines. The subroutines FCALC1 and FCALCZ both use different job sequences. Then the subroutine SHIFT computes the amount of shift between the two sequences of a passing solution and the main program notes all the similarities between the passing and non-passing optimum sequences. The computer program is listed in Appendix C.

## Generation of Processing Times

To determine the extent of reliability in the four observations mentioned, extensive experimentation with 3 job, 4 machine; 4 job, 4 machine; and 5 job, 4 machine flow shops was undertaken. The integer random numbers from the uniform probability distribution between and including 0 to 100 were generated in sufficient quantity to construct two sets of 500 problems each for the three different sizes of flow shops. Traditionally, researchers have chosen integer random numbers for the processing times of problems in their experiments from either uniform probability distribution or from some arbitrary, unknown distributions. For this dissertation, I have decided to work initially with integer random numbers from the uniform probability distribution. Using uniform probability distribution, it is easy to generate random numbers from it, and it is simple to compare these results with others; since almost all researchers in the flow shop area have used it. Once the results were obtained from experiments with problems whose processing times were generated from uniform probability distribution, I decided to work with processing times generated randomly from Beta probability distribution. Although Beta probability distribution is not popular with researchers in the scheduling area and the author has not encountered any research utilizing Beta distribution, there are two reasons for using it in this research. In the first place, Beta distribution is a close-range distribution. The normal range of random variates is from 0 to 1. Thus, unlike normal or exponential probability distributions, rejecting numbers beyond certain standard deviations from

the mean values is not necessary. Since the normal range of random variates is from 0 to 1, it is easy to get integer random numbers distributed in the Beta fashion between 0 and 100 or any other range by simply multiplying the variates appropriately and integerizing them. The second reason for choosing the Beta distribution pertains to its flexibility.

The Beta distribution curve changes its shape considerably with changes in the values of its two parameters, A and B. When the values of A and B are equal, Beta distribution curves are symmetrical around the mean value but when A and B have different values, the curves are skewed. The skewness increases with the increase in the difference between the values of A and B. When A and B are both smaller than 1, the curves are convex if observed from the bottom. When A and B are both equal to 1, the Beta distribution is exactly the same as the uniform probability distribution, thus the curve is simply a straight line. When both A and B are greater than 1, the Beta curves are concave from the bottom. As the values of A and B increase from 1, the curves look more and more like a bell. For these experiments, 28,800 random numbers distributed between and including 0 to 100 were generated for each of the six sets of values for A and B. Each of the sets had equal values of A and B, which are 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0. The frequency distributions of the first 20,000 numbers for each of the six sets are shown in Table 3.2

and in Figure 3.1 in the form of curves. The scientific subroutine package of IBM has a program called BDTR which computes the ordinate of the Beta density function for a given uniformly generated random number "X1." By using the rejection technique on the pair of random numbers "X1" and "X2" generated, the random variates of the Beta probability distribution are generated. The computer program is shown in Appendix D.

The processing times chosen from either the uniform or Beta probability distribution for this dissertation ranges from 0 through 100. Researchers have used many different ranges of processing times for their experiments in the flow shop scheduling area. Heller (19) and Palmer (33) worked with numbers ranging from 0 to 9, whereas McMahan and Burton (28) and Lomnicki (27) worked with numbers between 0 and 30. Giglio and Wagner (15) generated processing times ranging from 1 to 30. Campbell, Dudek and Smith (7) used numbers between 0 and 99. Gupta (16, 18) worked with numbers that ranged from 0 to 999 whereas Krone (25) used numbers from 0 to 1000. From studying published work in the flow shop scheduling area, it is clear that not only every researcher uses his own range of integer random numbers for processing times, but he does not present or explain his reasons for choosing the range he used. This makes it very difficult to choose an ideal range. From preliminary experiments an intuitive rule emerged. When integer random numbers are used for the processing times of

Table 3.2

Frequency Distribution of Random Numbers Generated Using Beta Probability Distributions

| | Parameters A and B | | | | | |
|---|---|---|---|---|---|---|
| Range | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 |
| 0-5 | 2,592 | 1,052 | 381 | 144 | 68 | 23 |
| 6-10 | 1,276 | 1,068 | 655 | 423 | 284 | 142 |
| 11-15 | 985 | 1,025 | 829 | 611 | 522 | 362 |
| 16-20 | 879 | 986 | 988 | 867 | 773 | 644 |
| 21-25 | 776 | 982 | 999 | 1,048 | 1,024 | 931 |
| 26-30 | 761 | 979 | 1,135 | 1,114 | 1,252 | 1,205 |
| 31-35 | 717 | 1,013 | 1,186 | 1,366 | 1,363 | 1,452 |
| 36-40 | 719 | 1,001 | 1,275 | 1,429 | 1,574 | 1,672 |
| 41-45 | 652 | 976 | 1,301 | 1,476 | 1,573 | 1,762 |
| 46-50 | 667 | 1,019 | 1,273 | 1,530 | 1,680 | 1,826 |
| 51-55 | 671 | 979 | 1,297 | 1,499 | 1,683 | 1,852 |
| 56-60 | 685 | 979 | 1,294 | 1,455 | 1,607 | 1,768 |
| 61-65 | 651 | 976 | 1,248 | 1,426 | 1,547 | 1,586 |
| 66-70 | 714 | 999 | 1,186 | 1,302 | 1,362 | 1,423 |
| 71-75 | 739 | 1,025 | 1,107 | 1,132 | 1,218 | 1,239 |
| 76-80 | 808 | 1,019 | 1,022 | 1,075 | 960 | 940 |
| 81-85 | 864 | 962 | 985 | 874 | 713 | 633 |
| 86-90 | 1,024 | 1,005 | 797 | 643 | 513 | 372 |
| 91-95 | 1,261 | 964 | 684 | 432 | 242 | 146 |
| 96-100 | 22,559 | 991 | 358 | 154 | 42 | 22 |
| | 20,000 | 20,000 | 20,000 | 20,000 | 20,000 | 20,000 |

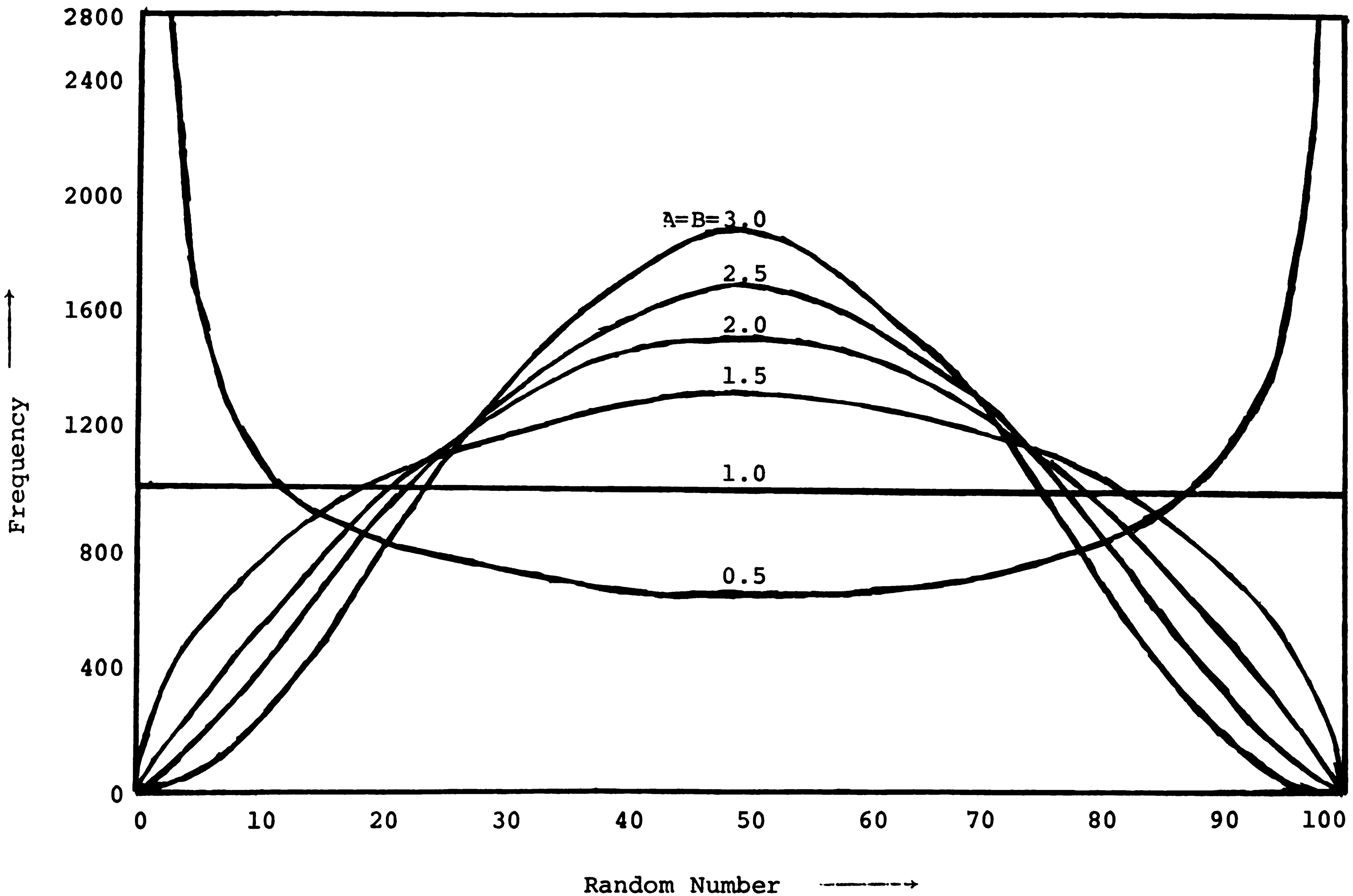


Figure 3.1 Frequency Distribution of Random Numbers Generated Using Beta Probability Distributions.

a flow shop scheduling problem, the value of the range of numbers should be fairly larger than the value obtained by multiplying the number of jobs times the number of machines in the flow shop. With a maximum size of 5 job, 4 machine flow shop problems in this experiment (N*M = 20), the range of 0 to 100 seems to be fairly large. The range could well have been 0 to 99, but there is one advantage to having the range up to 100. It gives the mean value of the distribution equal to 50 instead of the slightly odd amount of 49.5.

Many researchers selected ranges that were too narrow. For example, Heller (19) for his 100 job, 20 machine flow shop problem, used a range of 0 to 9. Giglio and Wagner (15) for their 6 to 9 job, 3 machine flow shop problems used a range of 1 to 30. These and a few other ranges are much too low to get satisfactory experimental results. When the range is too narrow compared to the value of N times M, the total number of optimum sequences under no passing becomes unduly large. Also, the difference between the minimum total make-span and the maximum total make-span becomes quite narrow. As a result, the discrimination between good sequences and bad sequences becomes very hazy. The decision to use a narrow range has a self-serving advantage when a heuristic technique is used in the experiments. When the range is narrow, a non-optimum sequence would seem to have much closer or even the same total make-span value as the optimum sequences would. This phenomenon would make the heuristic technique look much better than what it actually might be.

## Measurement Functions

The various experiments done on flow shop problems in this dissertation are for one of three different sizes of the flow shop and have processing times randomly generated either from the uniform probability distribution or from one of the five Beta probability distributions. Due to these differences in the various experiments, the numberical results and their frequency distributions could differ markedly. The minimum total make-span values, the total make-span when passing is permitted, and the extent of improvement in the total make-span by permitting passing, all could have different and varying frequency distributions, means and standard deviations from experiment to experiment. Unless some common basis of measurement or some way to place them on the same scale is utilized, the comparison between experiments and the common conclusions from all experiments would be very difficult.

Since this dissertation is mainly concerned with discovering and presenting improvement in the minimum total make-span of the classical flow shop by permitting passing between sequences, the different measurement functions suggested below are mainly centered around the measurement of improvement.

To put different measurement functions into formulas, the following notations are defined:

TMSP = Best total make-span when Passing is permitted.

MTMS = Minimum total make-span under the No Passing restrictions.

WTMS = Maximum total make-span under the No Passing restriction.

The measurement function for the extent of improvement could be one or more of the following four functions:

1. $$f1 = \frac{WTMS - TMSP}{WTMS - MTMS}$$

In this function the denominator is the difference between the best and the worst total make-spans under the no passing restriction. The denominator is the basis of comparison with the numerator, which is the difference between the total make-span when passing is permitted and the worst total make-span under the no passing restriction. When the permission of passing does not improve the total make-span value, the function has the value of 1 but as the improvement is greater, the function has increasingly larger values than 1.

2. $$f2 = \frac{MTMS - TMSP}{WTMS - MTMS}$$

This function is similar to the previous function except that the numerator has the difference of the total make-span under passing with the minimum total make-span under no passing. In the absence of any improvement, the function has a value of zero, but as the improvement increases, the value of the function goes up.

3. $f3 = \frac{TMSP}{MTMS}$

In this function, the ratio of the total make-span under passing is compared to the minimum total make-span. The ratio has a value of 1 if there is no improvement by permitting passing but as the improvement increases, the value of the ratio goes down.

4. $f4 = \frac{MTMS - TMSP}{MTMS}$

This function is similar to the third function, but instead of measuring the total make-span under passing, it measures the improvement of the total make-span under passing over the minimum total make-span under no passing. The value of the ratio is zero if there is no improvement but it increases directly with the increase in the improvement.

It is difficult to choose from these four functions. The first two require the value of the worst total make-span under the no passing restriction. If complete enumeration is done on the flow shop scheduling problem in phase 1 then the WTMS would be available, but the Branch and Bound algorithm does not provide WTMS. Thus, the first two functions have a slight disadvantage. After some trial work, it was decided to use the fourth function.

# CHAPTER IV

## RESULTS OF ANALYSES WITH UNIFORMITY DISTRIBUTED PROCESSING TIMES

All the results obtained from the experiments on the two sets of 500 problems each for the 3 job, 4 machine; 4 job, 4 machine; and 5 job, 4 machine flow shop scheduling problems are presented and analyzed in this chapter. The processing times of these 3000 problems are integer random numbers distributed from 0 through 100 from uniform probability distribution. The actual processing times for the problems used in this chapter and Chapter V are not presented in this dissertation. Anyone interested in repeating some or all of these experiments can easily obtain similar results by generating his own random numbers for the processing times as long as the range and the probability distributions are the same as the ones used here. The results for each of the three different sizes of flow shop scheduling problems are presented in the three corresponding sections of this chapter.

As discussed in the third chapter, the number of different possible solutions with passing between sequences of different machines is tremendously high. For a 3 job, 4 machine flow shop problem, there are $(3!)^2 = 36$ possible solutions when passing is permitted. Out of these 36 solutions, $3! = 6$ solutions have zero passing between sequences, so that there are $36 - 6 = 30$ solutions with some amount of passing between sequences. The number of solutions with greater than zero passing for a 4 job, 4 machine flow shop problem is $(4!)^2 - 4! = 552$ and for a 5 job, 4 machine problem it is $(5!)^2 - 5! = 14,280$. In Phase Two of these experiments, searching for improvements in the minimum total make-spans of Phase One, each of these 30, 552 or 14,280 different solutions are examined for each 3 job, 4 job or 5 job, 4 machine flow shop scheduling problem. In addition to this complete enumeration search, five different partial search plans are formulated for this dissertation. These five plans involve searching through a much smaller number of solutions than the complete enumeration search thus saving a considerable amount of computational time. Since these plans do less than a complete search, they do not produce all of the possible improvements. The results of search by complete enumeration and proposed plans are analyzed in this chapter. These five different plans of partial search are formulated from the four preliminary observations made during the experiments presented in the third chapter. The proposed plans are explained below.

Plan 1: This plan calls for search in Phase Two among solutions that have one shift of passing. Out of a total of 30, 552 and 14,280 possible solutions for the 3 job, 4 job and 5 job, 4 machine flow shop scheduling problems, respectively, only 18, 144 and 1200 solutions have exactly one shift of passing. Thus, in this plan the search for improvement of the minimum total make-span of phase one is restricted to just 60 percent, 26.17 percent and 8.4 percent of the possible solutions for those three sizes of flow shop scheduling problems. The plan is formulated on the basis of the preliminary observation that most of the optimum passing solutions found in Phase Two have one shift of passing. And it results in 40 percent, 73.83 percent and 91.6 percent savings in computational efforts for 3 job, 4 job and 5 job, 4 machine flow shop scheduling problems.

Plan 2: As was mentioned in the third chapter, during preliminary experiments, many times one of the two sequences of passing solutions that has a lower total make-span in Phase Two than the minimum total make-span of Phase One is exactly the same as one of the optimum sequences providing the minimum total make-span of Phase One. Based on this observation, a "less than complete" search plan for Phase Two computation is formulated. This plan involves constructing "R" number of possible passing solutions and examining each one to see whether or not it improves the minimum total make-span of Phase One. The plan is formulated in such a way as to make R as small as possible without losing a considerable amount of possible savings in Phase Two.

During the computation of results of Phase One, all of the optimum sequences providing a minimum total make-span are obtained. Let us say that "T" number of optimum sequences are obtained in Phase One. Each of the T sequences in turn is considered one of the two sequences of the possible passing solutions. Let us say that sequence "K" is one of the T optimum sequences of Phase One and is one of the two sequences of a possible passing solution "P." A set of all the possible sequences of N jobs minus the K sequence is constructed. This set will have N! - 1 sequences. Each of these N! - 1 sequences are then used one by one as the other sequence of the possible passing solution P. Thus, solution P is constructed from sequence K and a sequence from the set of (N!-1) sequences. Sequence K can go as the first or second sequence so there will be 2*(N!-1) possible passing solutions constructed using the Kth sequence. Thus, for T number of optimum sequences of Phase One there will be a total of T*2*(N!-1) possible passing solutions. Plan Two consists of examining these T*2*(N!-1) passing solutions and selecting the one with the lowest total make-span over the minimum total make-span of Phase One.

The amount of computation involved in this plan depends upon the number of sequences providing the minimum total make-span for a problem. For the thousand problems of 3 job, 4 machine flow shop scheduling problems whose results are reported in this chapter, the average number of optimum sequences in Phase One was 1.189 per problem. The frequency distribution from which the average number of optimum sequences per problem was obtained, will be presented later in this chapter. It implies that on an average 1.189*2*(3!-1) = 11.89 passing solutions should be examined according to Plan Two. The search through 11.89 possible passing solutions rather than 30 solutions means a savings of (30-11.89)/30 = 60.37 percent in computational effort over the complete enumeration search. On a similar basis, the computation effort that can be saved by following Plan Two is on an average 85.91 percent for a 4 job, 4 machine problem and 95.7 percent for a 5 job, 4 machine flow shop scheduling problem over the complete enumeration search.

Plan 3: This is simply a combination of Plans One and Two and is based on the two observations made during the preliminary experiments. The first observation is that most of the "useful" Phase Two passing solutions have one shift of passing. Secondly, most of the "useful" Phase Two passing solutions have one sequence the same as one of the Phase One optimum sequences. The possible passing solutions formulated and searched in Plan Three have one sequence the same as one of the optimum sequences obtained in Phase One <u>and also</u> have exactly one shift of passing between the two sequences.

The possible passing solutions are formulated by choosing one of the two sequences of the passing solution from the optimum sequences of Phase One and by choosing the other sequence from the set of sequences constructed by the permutations of N jobs <u>such that</u> the amount of passing between the two sequences is <u>exactly equal to one shift</u>. With any sequence of Phase One of the 3 job, 4 machine flow shop problem, there are only three sequences any of which if used together with the first one will have one shift of passing. For example, if the optimum sequence from Phase One is 231 and if it is put as the <u>first</u> of the two sequences in a passing solution, then there can be any one of the three sequences 213, 312, and 321 for the second sequence of the passing solution to have one shift of passing. If the sequence 231 is put as the <u>second</u> of the two sequences in a passing solution, there can be any one of the three sequences 123, 213, and 321 for the first place to have one shift of passing.

Possible Passing Solutions

| | 1 | 2 | 3 |
|---|---|---|---|
| First Sequence | 2 3 1 | 2 3 1 | 2 3 1 |
| Second Sequence | 2   1 3 |   3 1 2 |   3 2 1 |
| | **4** | **5** | **6** |
| First Sequence | 1 2 3 | 2 1 3 | 3 2 1 |
| Second Sequence |   2 3 1 | 2   3 1 |   2 3 1 |

Thus in this plan for the 3 job, 4 machine flow shop scheduling problem, for each optimum sequence of Phase One, there are 3 x 2 = 6 possible passing solutions to be searched in Phase Two. Since on an average there are 1.189 optimum solutions in Phase One, on an average for each 3 job, 4 machine problem Plan Three advocates searching through 6x1.189 = 7.134 possible passing solutions in Phase Two. This indicates Plan Three will save (30-7.134)/30 = 76.22 percent in computational effort over the complete enumeration search. And for a 4 job and 5 job, 4 machine flow shop scheduling problem, Plan Three saves 96.31 percent and 99.64 percent in computational effort over the complete enumeration search.

Plans 4 and 5: These plans are minor modifications of Plan Three. Since the total number of possible optimum sequences for Phase One is N!, there could be as many as 6, 24 or 120 optimum sequences for a 3 job, 4 job or 5 job, 4 machine flow shop scheduling problem at the end of Phase One. Although it is determined from the experiments on the 1,000 problems each for 3 job, 4 job and 5 job, 4 machine flow shop scheduling problems of which on an average there are only 1.189, 1.696 and 2.567 optimum sequences in Phase One respectively, an individual problem could have a very high number of optimum sequences. (Tables 4.3, 4.10, and 4.17 show the frequency distribution of the number of optimum sequences per problem for the 3 job, 4 job and 5 job, 4 machine flow shop scheduling problem, and also point out the value of the average number of optimum sequences per problem presented later in this chapter.) If a problem has high number of optimum sequences, Plan Three could ultimately advocate a search through a tremendous number of possible passing solutions. Plans Four and Five advocate

enforcing an artificial ceiling on the number of optimum sequences considered at the end of Phase One. Since the maximum number of optimum sequences of Phase One could be only six for the 3 job, 4 machine flow shop scheduling problem, a ceiling is not necessary and would be of little help. From Table 4.10, it can be seen that 85.4 percent of the total 1,000 problems have one or two optimum sequences, and 92.3 percent of the problems have one, two, or three optimum sequences in Phase One. Thus ceilings of three or two optimum sequences for 4 job, 4 machine flow shop scheduling problems would effectively cover most of the problems and restrict the amount of computational efforts, and remain fixed for the search of Plans Four and Five respectively. From Table 4.17, it can be seen that 90.1 percent of the total 1,000 problems have five or less optimum sequences and 93.9 percent have six or less optimum sequences in Phase One for 5 job, 4 machine flow shop scheduling problems. Thus ceilings of six or five optimum sequences for 5 job, 4 machine flow shop scheduling problems would effectively cover most of the problems and restrict the amount of computational efforts, therefore they were selected for search Plans Four and Five respectively.
Tables 4.10 and 4.17 demonstrate that Plan Four reduces the average number of optimum sequences per problem for Phase One from 1.696 and 2.567 to 1.544 and 2.24, respectively, for 4 job and 5 job, 4 machine problems. Plan Five further reduces the average number of optimum sequences in Phase One to

1.398 and 2.141 per problem for 4 job and 5 job, 4 machine flow shop scheduling problems.

## 3 Job, 4 Machine Flow Shop Scheduling Problems

The results obtained for each of the two replications of 500 problems for Phases One and Two are presented in Tables 4.1 and 4.2.

### Phase One

The minimum total make-spans and the maximum total make-spans are computed for each of the problems and presented in Tables 4.1 and 4.2 in the form of frequency distributions.

Each of the 500 problems for both replications has one or more optimal sequences in Phase One. Since Plans Two and Three in Phase Two for the 3 job, 4 machine flow shop scheduling problem attempts its search for the improvements of the total make-span on the basis of the optimal sequences of Phase One, it is useful to know the frequency distribution of problems according to their number of optimal sequences in Phase One. Table 4.3 provides such a frequency distribution.

Thus, there are a total of 1189 optimal sequences for the 1000 problems, or an average of 1.189 optimal sequences per problem in Phase One for the 3 job, 4 machine flow shop scheduling problem. A total of six different permutations or sequences can be formulated from a total of three different jobs. Since these 1189 optimal sequences of Phase

Table 4.1

Frequency Distribution of 3 Job, 4 Machine Problems According to the Minimum Total Make-spans Under No Passing

| The Range of Minimum Total Make-spans | Frequencies | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| 181-200 | 5 | 5 | 10 |
| 201-220 | 8 | 6 | 14 |
| 221-240 | 11 | 14 | 25 |
| 241-260 | 41 | 25 | 66 |
| 261-280 | 44 | 47 | 91 |
| 281-300 | 62 | 67 | 129 |
| 301-320 | 82 | 77 | 159 |
| 321-340 | 67 | 65 | 132 |
| 341-360 | 57 | 58 | 115 |
| 361-380 | 55 | 56 | 111 |
| 381-400 | 46 | 37 | 83 |
| 401-420 | 13 | 28 | 41 |
| 421-440 | 5 | 10 | 15 |
| 441-460 | 2 | 3 | 5 |
| 461-480 | 2 | 2 | 4 |
| Total | 500 | 500 | 1000 |
| Average total minimum make-span | 321.23 | 325.34 | 323.28 |
| Std. dev. of the total minimum make-span | 50.99 | 52.65 | 51.83 |

Table 4.2

Frequency Distribution of 3 Job, 4 Machine Problems According to the Maximum Total Make-Spans Under No Passing

| The Maximum Total Make-spans | Frequencies | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| Less than 260 | 2 | 1 | 3 |
| 261-280 | 8 | 8 | 16 |
| 281-300 | 15 | 11 | 26 |
| 301-320 | 18 | 27 | 45 |
| 321-340 | 29 | 32 | 61 |
| 341-360 | 38 | 42 | 80 |
| 361-380 | 59 | 51 | 110 |
| 381-400 | 63 | 57 | 120 |
| 401-420 | 87 | 67 | 154 |
| 421-440 | 67 | 62 | 129 |
| 441-460 | 53 | 58 | 111 |
| 461-480 | 27 | 40 | 67 |
| 481-500 | 19 | 27 | 46 |
| 501-520 | 11 | 12 | 23 |
| 521-540 | 3 | 4 | 7 |
| More than 540 | 1 | 1 | 2 |
| Total | 500 | 500 | 1000 |
| The average maximum total make-span | 400.72 | 403.21 | 401.96 |
| The std. dev. of maximum total make-span | 54.92 | 57.74 | 56.35 |

Table 4.3

Frequency Distribution of Problems According to the Number of Optimal Sequences Under No Passing

| | Replication 1 | | Replication 2 | | Total | |
|---|---|---|---|---|---|---|
| Number of Optimal Sequences O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. |
| 1 | 411 | 411 | 402 | 402 | 813 | 813 |
| 2 | 89 | 178 | 96 | 192 | 185 | 370 |
| 3 | 0 | 0 | 2 | 6 | 2 | 6 |
| Total | 500 | 589 | 500 | 600 | 1000 | 1189 |

One are for 1000 problems of 3 job, 4 machine flow shop type, each of them can be any one of the six possible permutations. The frequency distributions of these 1189 sequences according to the six different permutations are given in Table 4.4.

Table 4.4

Frequency Distribution of Optimal Sequences of Phase One According to the Different Possible Permutations of 3 Jobs

| Different Permutations or Sequences | Number of Sequences | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| 1 2 3 | 105 | 108 | 213 |
| 1 3 2 | 90 | 87 | 177 |
| 2 1 3 | 95 | 90 | 185 |
| 2 3 1 | 101 | 103 | 204 |
| 3 1 2 | 103 | 113 | 216 |
| 3 2 1 | 95 | 99 | 194 |
| Total | 589 | 600 | 1189 |

Jobs 1, 2, and 3 have the same value of average processing times and the same value of standard deviation of processing times on all four machines. So each of them--job 1, 2, or 3--are equally likely to be the first, second or third in any optimal sequence. Thus, in a long run, each of the six possible sequences should have an equivalent number of optimal sequences attributed to it. From the table it appears that all six sequences are almost equally popular. To check the above hypothesis of these six equally popular sequences, Goodness of Fit using the Chi-square tests was conducted on it. For the five degrees of freedom (Number of class - 1 = 6 - 1 = 5) $\chi_2$ .05 is equal to 11.070. The computed value of Chi-square from the frequency distributions of 1189 optimal sequences is 6.11. Thus, the null hypothesis of no difference between the popularity of the six possible sequences is accepted.

Phase Two

A complete enumeration of all the possible solutions with some passing between sequences for each of the 500 problems of both replications was performed. As a result, in 52 problems of the 500 problems in replication 1 and in 53 of the 500 problems in replication 2, the best Phase Two total make-spans were lower than their minimum total make-spans in Phase One. The actual amount of improvement was anywhere from 1 to 46 time units, averaging 11.44 and 13.98 time units, respectively, for 52 and 53 problems of replications 1 and 2. Each of the 52 + 53 = 105 problems had anywhere from one to seven solutions whose total make-spans in Phase Two were lower than the minimum make-span of Phase One. The average number of solutions per problem which had a lower total make-span than the minimum total make-span of Phase One was 1.37 and 1.42 for the 52 and 53 problems of replication 1 and 2, respectively. The explanation of the table and the tables that are presented in the following pages provide details of these problems in replications 1 and 2.

There are two tables for the two replications. The first column of these two tables lists the natural number of the 52 and 53 problems and the second column lists their original numbers from the 1 to 500 problems of their respective replications. The third column lists the minimum total make-spans obtained in Phase One and the fourth column lists the number of optimal sequences obtained in Phase One. The fifth column provides the best total make-span obtained in Phase Two and the sixth column lists the number of solutions which provided total make-spans in Phase Two which were lower than the minimum total make-span of Phase One. The seventh column lists the maximum amount of savings possible in the total make-span by permitting passing in the particular problem, being simply the difference between the third and the fifth columns. The eighth, ninth, and tenth columns provide the amount of savings in Phase Two that can be obtained by searching according to Plans One, Two and Three rather than using complete enumeration. These limited search plans were explained earlier in this chapter.

From Tables 4.5 and 4.6 it can be seen that in 105 problems out of a total 1000 problems (10.5 percent of the problems), the value of the best total make-span decreases if passing is permitted. The average minimum total make-span for the 105 problems at the end of Phase One computations

Table 4.5

3 Job, 4 Machine Flow Shop Scheduling Problems--Replication 1

| | No Passing | | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 1 | 17 | 308 | 1 | 292 | 1 | 16 | 16 | 16 | 16 |
| 2 | 48 | 323 | 1 | 305 | 2 | 18 | 18 | 18 | 18 |
| 3 | 61 | 211 | 1 | 206 | 1 | 5 | 5 | 5 | 5 |
| 4 | 93 | 381 | 1 | 354 | 1 | 27 | 27 | 27 | 27 |
| 5 | 101 | 260 | 2 | 246 | 1 | 14 | 14 | 14 | 14 |
| 6 | 112 | 306 | 2 | 305 | 1 | 1 | 1 | 1 | 1 |
| 7 | 121 | 417 | 2 | 410 | 1 | 7 | 7 | 7 | 7 |
| 8 | 132 | 322 | 1 | 313 | 1 | 9 | 9 | 9 | 9 |
| 9 | 143 | 292 | 1 | 283 | 1 | 9 | 9 | 9 | 9 |
| 10 | 150 | 316 | 2 | 314 | 1 | 2 | 2 | 2 | 2 |
| 11 | 167 | 285 | 2 | 271 | 1 | 14 | 14 | 14 | 14 |
| 12 | 179 | 386 | 2 | 373 | 1 | 13 | 13 | 13 | 13 |
| 13 | 183 | 323 | 2 | 302 | 1 | 21 | 21 | 21 | 21 |
| 14 | 200 | 324 | 1 | 318 | 2 | 6 | 6 | 6 | 6 |
| 15 | 225 | 370 | 2 | 366 | 1 | 4 | 4 | 4 | 4 |
| 16 | 228 | 374 | 1 | 357 | 1 | 17 | 17 | 17 | 17 |
| 17 | 234 | 269 | 2 | 261 | 1 | 8 | 8 | 8 | 8 |

Table 4.5 continued.

| | No Passing | | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 18 | 250 | 385 | 1 | 382 | 1 | 3 | 3 | 3 | 3 |
| 19 | 264 | 384 | 1 | 379 | 11 | 5 | 5 | 5 | 5 |
| 20 | 281 | 357 | 1 | 348 | 1 | 9 | 9 | 9 | 9 |
| 21 | 287 | 396 | 1 | 376 | 1 | 20 | 20 | 20 | 20 |
| 22 | 296 | 267 | 2 | 255 | 1 | 12 | 12 | 12 | 12 |
| 23 | 304 | 253 | 2 | 211 | 7 | 42 | 42 | 42 | 42 |
| 24 | 308 | 269 | 1 | 264 | 1 | 5 | 5 | 5 | 5 |
| 25 | 316 | 322 | 1 | 317 | 1 | 5 | 5 | 5 | 5 |
| 26 | 325 | 246 | 2 | 244 | 1 | 2 | 2 | 2 | 2 |
| 27 | 327 | 333 | 1 | 325 | 1 | 8 | 8 | 8 | 8 |
| 28 | 331 | 383 | 1 | 381 | 2 | 2 | 2 | 0 | 0 |
| 29 | 342 | 280 | 1 | 275 | 2 | 5 | 5 | 5 | 5 |
| 30 | 351 | 365 | 1 | 321 | 4 | 44 | 44 | 44 | 44 |
| 31 | 353 | 373 | 1 | 366 | 1 | 7 | 7 | 7 | 7 |
| 32 | 359 | 343 | 2 | 341 | 1 | 2 | 2 | 2 | 2 |
| 33 | 369 | 374 | 2 | 347 | 1 | 27 | 27 | 27 | 27 |
| 34 | 371 | 375 | 1 | 329 | 3 | 46 | 46 | 46 | 46 |
| 35 | 374 | 362 | 1 | 352 | 2 | 10 | 10 | 10 | 10 |
| 36 | 376 | 363 | 2 | 361 | 1 | 2 | 2 | 2 | 2 |

Table 4.5 continued.

| | | No Passing | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 37 | 384 | 360 | 2 | 341 | 1 | 19 | 19 | 19 | 19 |
| 38 | 386 | 372 | 1 | 362 | 1 | 10 | 10 | 10 | 10 |
| 39 | 389 | 346 | 2 | 334 | 1 | 12 | 12 | 12 | 12 |
| 40 | 396 | 311 | 1 | 307 | 1 | 4 | 4 | 4 | 4 |
| 41 | 402 | 285 | 2 | 273 | 1 | 12 | 12 | 12 | 12 |
| 42 | 414 | 328 | 1 | 324 | 1 | 4 | 4 | 4 | 4 |
| 43 | 429 | 399 | 2 | 391 | 1 | 8 | 8 | 8 | 8 |
| 44 | 440 | 396 | 2 | 388 | 3 | 8 | 8 | 8 | 8 |
| 45 | 441 | 334 | 1 | 317 | 1 | 17 | 17 | 17 | 17 |
| 46 | 444 | 264 | 2 | 261 | 1 | 3 | 3 | 3 | 3 |
| 47 | 445 | 384 | 1 | 369 | 1 | 15 | 15 | 15 | 15 |
| 48 | 457 | 344 | 2 | 336 | 1 | 8 | 8 | 8 | 8 |
| 49 | 471 | 333 | 2 | 324 | 1 | 9 | 9 | 9 | 9 |
| 50 | 477 | 361 | 1 | 354 | 1 | 7 | 7 | 7 | 7 |
| 51 | 486 | 465 | 2 | 458 | 2 | 7 | 7 | 7 | 7 |
| 52 | 500 | 375 | 1 | 370 | 1 | 5 | 5 | 5 | 5 |
| Total | | 17554 | 76 | 16959 | 71 | 595 | 595 | 593 | 593 |
| Average | | 337.58 | 1.46 | 326.13 | 1.37 | 11.44 | 11.44 | 11.40 | 11.40 |

Table 4.6

3 Job, 4 Machine Flow Shop Scheduling Problems--Replication 2.

| | No Passing | | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 1 | 8 | 338 | 2 | 332 | 1 | 6 | 6 | 6 | 6 |
| 2 | 14 | 393 | 1 | 391 | 1 | 2 | 2 | 2 | 2 |
| 3 | 22 | 306 | 1 | 302 | 2 | 4 | 4 | 4 | 4 |
| 4 | 24 | 337 | 1 | 329 | 1 | 8 | 8 | 8 | 8 |
| 5 | 28 | 240 | 1 | 223 | 2 | 17 | 17 | 17 | 17 |
| 6 | 35 | 403 | 2 | 390 | 2 | 13 | 13 | 13 | 13 |
| 7 | 38 | 401 | 1 | 377 | 2 | 24 | 24 | 0 | 0 |
| 8 | 44 | 431 | 1 | 386 | 3 | 35 | 35 | 35 | 35 |
| 9 | 49 | 330 | 2 | 324 | 1 | 6 | 6 | 6 | 6 |
| 10 | 57 | 289 | 1 | 278 | 1 | 11 | 11 | 11 | 11 |
| 11 | 76 | 404 | 1 | 383 | 1 | 21 | 21 | 21 | 21 |
| 12 | 83 | 420 | 1 | 418 | 1 | 2 | 2 | 2 | 2 |
| 13 | 85 | 406 | 1 | 403 | 1 | 3 | 3 | 0 | 0 |
| 14 | 105 | 305 | 1 | 301 | 1 | 4 | 4 | 4 | 4 |
| 15 | 120 | 410 | 1 | 403 | 1 | 7 | 7 | 7 | 7 |
| 16 | 128 | 325 | 1 | 293 | 2 | 32 | 32 | 32 | 32 |
| 17 | 131 | 270 | 1 | 262 | 1 | 8 | 8 | 8 | 8 |

Table 4.6 continued.

| | | No Passing | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 18 | 136 | 380 | 1 | 363 | 1 | 17 | 17 | 0 | 0 |
| 19 | 141 | 251 | 1 | 250 | 1 | 1 | 1 | 1 | 1 |
| 20 | 154 | 279 | 1 | 243 | 5 | 36 | 36 | 36 | 36 |
| 21 | 156 | 259 | 1 | 255 | 1 | 4 | 4 | 4 | 4 |
| 22 | 165 | 320 | 1 | 286 | 1 | 34 | 34 | 34 | 34 |
| 23 | 197 | 293 | 1 | 277 | 2 | 16 | 16 | 16 | 16 |
| 24 | 200 | 186 | 1 | 184 | 1 | 2 | 2 | 2 | 2 |
| 25 | 225 | 351 | 1 | 337 | 1 | 14 | 14 | 14 | 14 |
| 26 | 242 | 322 | 1 | 315 | 1 | 7 | 7 | 0 | 0 |
| 27 | 255 | 391 | 1 | 390 | 1 | 1 | 1 | 1 | 1 |
| 28 | 268 | 322 | 2 | 321 | 1 | 1 | 1 | 0 | 0 |
| 29 | 271 | 334 | 2 | 313 | 3 | 21 | 21 | 21 | 21 |
| 30 | 275 | 345 | 1 | 335 | 2 | 10 | 10 | 9 | 9 |
| 31 | 300 | 366 | 2 | 362 | 1 | 4 | 4 | 4 | 4 |
| 32 | 312 | 404 | 1 | 380 | 1 | 24 | 24 | 24 | 24 |
| 33 | 313 | 416 | 1 | 387 | 2 | 29 | 29 | 29 | 29 |
| 34 | 314 | 382 | 2 | 372 | 1 | 10 | 10 | 10 | 10 |
| 35 | 322 | 330 | 1 | 326 | 1 | 4 | 4 | 4 | 4 |
| 36 | 342 | 240 | 2 | 230 | 1 | 10 | 10 | 10 | 10 |
| 37 | 346 | 339 | 1 | 304 | 1 | 35 | 35 | 35 | 35 |

Table 4.6 continued.

| | | No Passing | | Passing | | | Saving by Plan | | |
|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 |
| 38 | 376 | 273 | 2 | 261 | 1 | 12 | 12 | 12 | 12 |
| 39 | 398 | 360 | 2 | 333 | 1 | 27 | 27 | 27 | 27 |
| 40 | 401 | 312 | 2 | 305 | 1 | 7 | 7 | 7 | 7 |
| 41 | 410 | 352 | 1 | 322 | 2 | 30 | 30 | 30 | 30 |
| 42 | 414 | 412 | 1 | 390 | 1 | 22 | 22 | 22 | 22 |
| 43 | 415 | 332 | 1 | 323 | 3 | 9 | 9 | 9 | 9 |
| 44 | 417 | 271 | 1 | 261 | 1 | 10 | 10 | 10 | 10 |
| 45 | 419 | 368 | 1 | 347 | 1 | 21 | 21 | 21 | 21 |
| 46 | 430 | 419 | 2 | 407 | 1 | 12 | 12 | 12 | 12 |
| 47 | 433 | 338 | 2 | 337 | 1 | 1 | 1 | 1 | 1 |
| 48 | 436 | 261 | 1 | 252 | 1 | 9 | 9 | 9 | 9 |
| 49 | 437 | 267 | 1 | 247 | 1 | 20 | 20 | 20 | 20 |
| 50 | 453 | 284 | 1 | 273 | 2 | 11 | 11 | 8 | 8 |
| 51 | 461 | 362 | 1 | 346 | 2 | 16 | 16 | 6 | 6 |
| 52 | 465 | 371 | 1 | 325 | 2 | 46 | 46 | 46 | 46 |
| 53 | 481 | 290 | 1 | 285 | 1 | 5 | 5 | 5 | 5 |
| Total | | 17780 | 66 | 17039 | 75 | 741 | 741 | 675 | 675 |
| Average | | 355.47 | 1.25 | 321.49 | 1.42 | 13.98 | 13.98 | 12.74 | 12.74 |

is 336.51 units and the average total make-span after complete enumeration in Phase Two is 323.79 units. This means an average of 12.72 units (336.51 - 323.79 = 12.72) improvement per problem for the 105 problems when passing is permitted between sequences. In the remaining 895 problems (1000 - 105 = 895) there was no improvement by permitting passing. The average minimum total make-span under the no passing condition for all 1000 problems, as presented in Phase One results, was 323.28 units. When passing is permitted the average total make-span for the 1000 problems was 321.94 units. Thus, by permitting passing, an average savings of 1.34 units in the total make-span (323.28 - 321.94 = 1.34) per problem for the 1000 problems is achieved. At first consideration, this amount of savings does not seem like a lot but if viewed in proper perspective, it can appear worthwhile. Under the no passing restriction, the average minimum total make-span is 323.28 units and the maximum total make-span is 401.96 units (as presented earlier in this chapter), or an average difference of only 78.68 units between the worst and best total make-span. When an average improvement of 1.34 units by permitting passing is compared with the average difference between the best and the worst total make-span of 78.68 units under the no passing restriction, the improvement seems worthwhile. According to the measurement function of improvement by passing discussed and selected in Chapter III, the value of improvement is as follows:

$$f4 = \frac{MTMS - TMSP}{MTMS} = \frac{323.28 - 321.94}{323.28} = \frac{1.34}{323.28}$$

$$= .00415 \text{ or } 0.415\%$$

Table 4.7 summarizes the amount of improvement or savings in total make-spans from Phase One to Phase Two by complete enumeration and by the three plans.

Table 4.7

Amount of Savings in the Total Make-span and the Amount of Search Work Involved with Passing

| | Complete Enumeration | Plans 1 | Plans 2 | Plans 3 |
|---|---|---|---|---|
| 1. Amount of savings in the total make-spans | | | | |
| Replication 1 | 595 | 595 | 593 | 593 |
| Replication 2 | 741 | 741 | 675 | 675 |
| Total | 1336 | 1336 | 1268 | 1268 |
| Percent | 100 | 100 | 94.9 | 94.9 |
| 2. Number of solutions searched per problem | | | | |
| Total | 30 | 18.00 | 11.89 | 7.14 |
| Percent | 100 | 60.00 | 39.63 | 23.78 |
| 3. Benefit-cost ratio in terms of total savings divided by number of solutions searched | 44.53 | 74.25 | 106.78 | 177.59 |

As can be seen from the tables, the three plans manage to capture most of the possible savings in total make-spans from Phase One to Two. The total amount of savings in Plan One is exactly the same as the savings obtained by complete enumeration for both replications. Although Plan One for 3 job, 4 machine flow shop scheduling problems involves only 60 percent as much work as complete enumeration, it captures 100 percent of the savings. Plan Two saves 99.65 percent of the maximum amount for replication 1 and 91.09 percent for replication 2. The total amount of work involved in Plan Two is just 39.63 percent but for both replications it saves 94.91 percent of the total make-span as compared to complete enumeration. Plan Three involves only 23.78 percent or one fourth as much work as complete enumeration but saves 94.91 percent of the maximum possible total make-span. Thus, it seems these three plans are quite successful in capturing almost all the savings in total make-spans from Phase One to Phase Two while avoiding a large volume of work searching for them in 3 job, 4 machine flow shop scheduling problems. The benefit-cost ratio also increases steadily, and for Plan Three is four times as large as that for the complete enumeration.

## 4 Job, 4 Machine Flow Shop Scheduling Problems

The results obtained for the two replications of 500 problems for Phases One and Two are presented in Tables 4.8 and 4.9.

### Phase One

First, the frequency distributions of a number of problems, according to minimum total make-spans and maximum total make-spans, are presented in Tables 4.8 and 4.9.

The frequency distribution of the number of problems according to the number of optimal sequences in Phase One is given in Table 4.10. The total number of optimal sequences a problem has is useful to know for Plans Two to Five in Phase Two.

There are a total of 1691 optimal sequences for the 1000 problems, or an average of 1.691 optimal sequences per problem. Plans Four and Five advocate a maximum of only three and two optimal sequences respectively for 4 job, 4 machine flow shop scheduling problems. From Table 4.10 it seems there are a total of 77 and 146 problems having more than three and two optimal sequences, respectively. Using only the first three and two optimal sequences reduces the average number of optimal sequences from 1.691 per problem to 1.544 and 1.398 per problem, respectively. The relative amount of savings in search work and its consequent loss in improving the amount of the total make-span

Table 4.8

Frequency Distribution of 4 Job, 4 Machine Problems
According to the Minimum Total Make-Spans
Under No Passing

| The Range of Minimum Total Make-Spans | Number of Problems | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| Less than 220 | 0 | 1 | 1 |
| 221-240 | 4 | 0 | 4 |
| 241-260 | 8 | 7 | 15 |
| 261-280 | 12 | 11 | 23 |
| 281-300 | 21 | 22 | 43 |
| 301-320 | 35 | 33 | 68 |
| 321-340 | 51 | 51 | 102 |
| 341-360 | 56 | 65 | 121 |
| 361-380 | 74 | 71 | 145 |
| 381-400 | 68 | 72 | 140 |
| 401-420 | 63 | 56 | 119 |
| 421-440 | 42 | 42 | 84 |
| 441-460 | 29 | 32 | 61 |
| 461-480 | 18 | 18 | 36 |
| 481-500 | 9 | 9 | 18 |
| 501-420 | 9 | 8 | 17 |
| 521-540 | 1 | 2 | 3 |
| Total | 500 | 500 | 1000 |
| Average minimum total make-span | 377.14 | 377.92 | 377.53 |
| Std. Dev. of the minimum total make-span | 56.49 | 56.12 | 56.30 |

Table 4.9

Frequency Distribution of 4 Job, 4 Machine Problems According to the Maximum Total Make-spans Under No Passing

| The Range of Maximum Total Make-spans | Number of Problems | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| 321-340 | 3 | 3 | 6 |
| 341-360 | 4 | 5 | 9 |
| 361-380 | 6 | 4 | 10 |
| 381-400 | 15 | 20 | 35 |
| 401-420 | 23 | 24 | 47 |
| 421-440 | 44 | 35 | 79 |
| 441-460 | 39 | 35 | 74 |
| 461-480 | 61 | 54 | 115 |
| 481-500 | 62 | 62 | 124 |
| 501-520 | 72 | 70 | 142 |
| 521-540 | 57 | 59 | 116 |
| 541-560 | 56 | 58 | 114 |
| 561-580 | 31 | 33 | 64 |
| 581-600 | 16 | 20 | 36 |
| Above 600 | 11 | 18 | 29 |
| Total | 500 | 500 | 1000 |
| Average minimum total make-span | 494.91 | 498.14 | 496.52 |
| Std. dev. of minimum total make-span | 57.21 | 59.22 | 58.22 |

Table 4.10

Frequency Distributions of Problems According to the Number of Optimal Sequences Under No Passing

| | Replication 1 | | Replication 2 | | Total | |
|---|---|---|---|---|---|---|
| Number Optimal Sequences O.S. | Number of Problems N.P. | Total Number of Sequences O.S.*N.P. | Number of Problems N.P. | Total Number of Sequences O.S.*N.P. | Number of Problems N.P. | Total Number of Sequences O.S.*N.P. |
| 1 | 297 | 297 | 305 | 305 | 602 | 602 |
| 2 | 123 | 246 | 129 | 258 | 252 | 504 |
| 3 | 41 | 123 | 23 | 84 | 69 | 207 |
| 4 | 20 | 80 | 15 | 60 | 35 | 140 |
| 5 | 7 | 35 | 8 | 40 | 15 | 75 |
| 6 | 11 | 66 | 15 | 90 | 26 | 156 |
| 7 | 1 | 7 | 0 | 0 | 1 | 7 |
| Total | 500 | 854 | 500 | 837 | 1000 | 1691 |

with other plans will be discussed later in this chapter after the results of the Phase Two experiments are presented.

Under the no passing restriction for 4 job, 4 machine flow shop scheduling problems there are a total of 4! = 24 possible sequences or permutations. Each of the 1691 optimal sequences obtained in Phase One could be any one of the 24 different possible sequences. The frequency distribution of the 1691 optimal sequences, according to the 24 possible sequences obtained, is presented in Table 4.11.

Since there are 1691 total optimal sequences distributed among the 24 possible sequences, each sequence should have an average of 70.46 optimal sequences. Goodness of Fit was tested on the frequency distribution using the Chi-square test. For the 23 degrees of freedom $X^2$ .05 is equal to 35.172. The computed value of the Chi-square from the frequency distribution of 1691 optimal sequences was only 24.75. The number 24.75 being much below 35.172, it is clear each of the 24 possible sequences are equally popular to be optimal.

## Phase Two

For 68 and 62 problems of the 500 problems each for replications 1 and 2, the best total make-spans in Phase Two were lower than the minimum total make-spans of Phase One. The actual amount of improvement obtained by complete enumeration for these 68 and 62 problems was anywhere from

Table 4.11

Frequency Distribution of Optimal Sequences of Phase One According to the Different Possible Permutations of 4 Jobs

| No. | Sequences or Permutations | Replication 1 | Replication 2 | Total |
|---|---|---|---|---|
| 1 | 1 2 3 4 | 31 | 30 | 61 |
| 2 | 1 2 4 3 | 32 | 26 | 58 |
| 3 | 1 3 2 4 | 33 | 40 | 73 |
| 4 | 1 3 4 2 | 35 | 43 | 78 |
| 5 | 1 4 2 3 | 32 | 30 | 62 |
| 6 | 1 4 3 2 | 35 | 31 | 66 |
| 7 | 2 1 3 4 | 46 | 31 | 77 |
| 8 | 2 1 4 3 | 37 | 31 | 68 |
| 9 | 2 3 1 4 | 42 | 32 | 74 |
| 10 | 2 3 4 1 | 36 | 32 | 68 |
| 11 | 2 4 1 3 | 29 | 38 | 57 |
| 12 | 2 4 3 1 | 31 | 34 | 65 |
| 13 | 3 1 2 4 | 36 | 23 | 59 |
| 14 | 3 1 4 2 | 28 | 38 | 66 |
| 15 | 3 2 1 4 | 33 | 28 | 61 |
| 16 | 3 2 4 1 | 32 | 45 | 77 |
| 17 | 3 4 1 2 | 37 | 44 | 81 |
| 18 | 3 4 2 1 | 40 | 32 | 72 |
| 19 | 4 1 2 3 | 33 | 34 | 67 |
| 20 | 4 1 3 2 | 35 | 41 | 76 |
| 21 | 4 2 1 3 | 37 | 35 | 72 |
| 22 | 4 2 3 1 | 39 | 41 | 80 |
| 23 | 4 3 1 2 | 42 | 47 | 89 |
| 24 | 4 3 2 1 | 43 | 41 | 84 |
| Total | | 854 | 837 | 1691 |

one to 48 units, averaging 12.07 and 12.69 units per problem, respectively, for replications 1 and 2. Each of these problems had anywhere from one to 24 passing solutions whose total make-span values in Phase Two were lower than the minumum total make-spans of Phase One. The average number of passing solutions with lower make-spans than the minimum total make-span of Phase One were 3.24 and 2.73 per problem, respectively. Tables 4.12 and 4.13 are similar to the tables for 3 job, 4 machine flow shop scheduling problems, with the addition of two more columns showing the amount of improvement in total make-spans by Plans Four and Five, respectively.

For 4 job, 4 machine flow shop problems there are 68 + 62 = 130 problems of the total 1000 problems in which permitting passing improves the total make-span. The average minimum total make-span at the end of Phase One for these 130 problems is 396.36 units and the average best total make-span when passing is permitted is 383.99 units. This is an average improvement of 12.37 units per problem for the 130 problems. In the remaining 870 problems, there was improvement in Phase Two computations. The average minimum total make-span for the 1000 problems in Phase One was 377.53 units and the average total make-span for the same 1000 problems in Phase Two was 375.92 units. Thus, by permitting passing, an average reduction of 1.61 units of the total make-span is achieved per problem.

Table 4.12

4 Job, 4 Machine Flow Shop Scheduling Problems--Replication 1

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 1 | 3 | 396 | 3 | 393 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 2 | 9 | 349 | 5 | 332 | 2 | 17 | 17 | 17 | 17 | 17 | 17 |
| 3 | 17 | 376 | 3 | 361 | 2 | 15 | 15 | 15 | 15 | 15 | 15 |
| 4 | 21 | 336 | 2 | 334 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 5 | 24 | 467 | 2 | 459 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 6 | 30 | 392 | 1 | 374 | 3 | 18 | 18 | 18 | 18 | 18 | 18 |
| 7 | 34 | 379 | 1 | 372 | 4 | 7 | 7 | 0 | 0 | 0 | 0 |
| 8 | 36 | 386 | 4 | 379 | 2 | 7 | 7 | 7 | 7 | 0 | 0 |
| 9 | 58 | 313 | 6 | 298 | 6 | 15 | 15 | 15 | 15 | 3 | 0 |
| 10 | 86 | 324 | 2 | 320 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| 11 | 90 | 455 | 2 | 431 | 1 | 24 | 24 | 24 | 24 | 24 | 24 |
| 12 | 103 | 401 | 1 | 391 | 2 | 10 | 10 | 10 | 10 | 10 | 10 |
| 13 | 112 | 407 | 2 | 399 | 2 | 8 | 8 | 8 | 8 | 8 | 8 |
| 14 | 115 | 433 | 1 | 423 | 2 | 10 | 10 | 10 | 10 | 10 | 10 |
| 15 | 116 | 434 | 2 | 428 | 1 | 6 | 6 | 0 | 0 | 0 | 0 |
| 16 | 118 | 350 | 2 | 329 | 3 | 21 | 21 | 21 | 21 | 21 | 21 |
| 17 | 122 | 330 | 2 | 314 | 2 | 16 | 16 | 16 | 16 | 16 | 16 |

Table 4.12 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 18 | 126 | 431 | 1 | 428 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 19 | 128 | 400 | 1 | 388 | 7 | 12 | 12 | 12 | 12 | 12 | 12 |
| 20 | 131 | 326 | 2 | 323 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 21 | 137 | 398 | 5 | 391 | 3 | 7 | 7 | 7 | 7 | 7 | 0 |
| 22 | 140 | 235 | 1 | 233 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 23 | 154 | 368 | 2 | 333 | 20 | 35 | 35 | 35 | 35 | 35 | 35 |
| 24 | 160 | 471 | 4 | 461 | 2 | 10 | 10 | 9 | 9 | 0 | 0 |
| 25 | 173 | 330 | 2 | 311 | 1 | 19 | 19 | 19 | 19 | 19 | 19 |
| 26 | 178 | 393 | 3 | 378 | 9 | 15 | 15 | 15 | 15 | 15 | 15 |
| 27 | 184 | 273 | 1 | 259 | 2 | 14 | 14 | 0 | 0 | 0 | 0 |
| 28 | 189 | 425 | 1 | 419 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 29 | 198 | 377 | 3 | 366 | 3 | 11 | 11 | 11 | 11 | 11 | 11 |
| 30 | 223 | 385 | 2 | 380 | 2 | 5 | 5 | 5 | 5 | 5 | 5 |
| 31 | 224 | 366 | 4 | 358 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 32 | 225 | 445 | 1 | 397 | 1 | 48 | 48 | 48 | 48 | 48 | 48 |
| 33 | 228 | 438 | 1 | 427 | 1 | 11 | 11 | 0 | 0 | 0 | 0 |
| 34 | 233 | 493 | 6 | 476 | 2 | 17 | 17 | 17 | 17 | 17 | 17 |
| 35 | 245 | 397 | 1 | 389 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 36 | 248 | 419 | 7 | 394 | 10 | 25 | 25 | 25 | 25 | 7 | 7 |

Table 4.12 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 37 | 252 | 333 | 1 | 332 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 38 | 261 | 365 | 3 | 363 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 39 | 266 | 317 | 1 | 286 | 17 | 31 | 31 | 31 | 31 | 31 | 31 |
| 40 | 268 | 309 | 1 | 293 | 1 | 16 | 16 | 16 | 16 | 16 | 16 |
| 41 | 280 | 397 | 2 | 391 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 42 | 298 | 392 | 2 | 383 | 2 | 9 | 9 | 9 | 9 | 9 | 9 |
| 43 | 299 | 419 | 1 | 395 | 3 | 24 | 24 | 24 | 24 | 24 | 24 |
| 44 | 301 | 374 | 2 | 360 | 1 | 14 | 14 | 14 | 14 | 14 | 14 |
| 45 | 307 | 539 | 2 | 529 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |
| 46 | 309 | 431 | 2 | 415 | 4 | 16 | 16 | 16 | 16 | 16 | 16 |
| 47 | 314 | 435 | 1 | 398 | 5 | 37 | 37 | 0 | 0 | 0 | 0 |
| 48 | 332 | 347 | 1 | 333 | 8 | 14 | 14 | 7 | 7 | 7 | 7 |
| 49 | 340 | 391 | 1 | 389 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 50 | 350 | 464 | 1 | 447 | 4 | 17 | 17 | 17 | 17 | 17 | 17 |
| 51 | 356 | 422 | 1 | 417 | 2 | 5 | 5 | 5 | 5 | 5 | 5 |
| 52 | 367 | 337 | 1 | 330 | 1 | 7 | 7 | 7 | 7 | 7 | 7 |
| 53 | 368 | 448 | 1 | 440 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 54 | 372 | 463 | 3 | 452 | 1 | 11 | 11 | 11 | 11 | 11 | 0 |
| 55 | 393 | 289 | 4 | 287 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |

Table 4.12 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 56 | 397 | 430 | 1 | 429 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 57 | 400 | 434 | 1 | 432 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 58 | 401 | 365 | 1 | 356 | 1 | 9 | 9 | 9 | 9 | 9 | 9 |
| 59 | 407 | 440 | 1 | 439 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 60 | 408 | 383 | 2 | 382 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 61 | 412 | 307 | 2 | 302 | 1 | 5 | 5 | 5 | 5 | 5 | 5 |
| 62 | 426 | 475 | 2 | 473 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 63 | 447 | 365 | 3 | 362 | 2 | 3 | 3 | 3 | 3 | 3 | 0 |
| 64 | 454 | 367 | 2 | 345 | 7 | 22 | 22 | 17 | 17 | 17 | 17 |
| 65 | 460 | 402 | 2 | 384 | 5 | 18 | 18 | 17 | 17 | 17 | 17 |
| 66 | 472 | 404 | 4 | 371 | 12 | 33 | 33 | 26 | 26 | 26 | 26 |
| 67 | 478 | 447 | 1 | 426 | 12 | 21 | 21 | 7 | 7 | 7 | 7 |
| 68 | 488 | 380 | 2 | 359 | 7 | 21 | 21 | 21 | 21 | 21 | 21 |
| Total | | 26,569 | 144 | 25,748 | 220 | 821 | 821 | 707 | 707 | 661 | 637 |
| Average | | 390.72 | 2.12 | 378.23 | 3.23 | 12.07 | 12.07 | 10.40 | 10.40 | 9.72 | 9.37 |

Table 4.13

4 Job, 4 Machine Flow Shop Scheduling Problems--Replication 2

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 1 | 29 | 459 | 4 | 439 | 3 | 30 | 20 | 20 | 20 | 4 | 4 |
| 2 | 41 | 379 | 1 | 365 | 1 | 14 | 14 | 0 | 0 | 0 | 0 |
| 3 | 52 | 372 | 1 | 359 | 2 | 13 | 13 | 13 | 13 | 13 | 13 |
| 4 | 57 | 442 | 1 | 429 | 3 | 13 | 13 | 13 | 13 | 13 | 13 |
| 5 | 66 | 407 | 1 | 388 | 1 | 19 | 19 | 19 | 19 | 19 | 19 |
| 6 | 79 | 446 | 2 | 425 | 1 | 21 | 21 | 21 | 21 | 21 | 21 |
| 7 | 84 | 308 | 2 | 303 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| 8 | 104 | 439 | 2 | 436 | 1 | 3 | 3 | 0 | 0 | 0 | 0 |
| 9 | 113 | 391 | 3 | 381 | 2 | 10 | 4 | 4 | 4 | 4 | 0 |
| 10 | 122 | 402 | 6 | 360 | 24 | 42 | 42 | 42 | 42 | 42 | 42 |
| 11 | 128 | 384 | 1 | 380 | 3 | 4 | 4 | 2 | 2 | 2 | 2 |
| 12 | 136 | 393 | 1 | 356 | 1 | 37 | 37 | 37 | 37 | 37 | 37 |
| 13 | 141 | 455 | 2 | 432 | 1 | 23 | 23 | 23 | 23 | 23 | 23 |
| 14 | 144 | 467 | 1 | 466 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 15 | 147 | 335 | 1 | 333 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 16 | 150 | 395 | 2 | 385 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |
| 17 | 154 | 449 | 1 | 428 | 1 | 21 | 21 | 21 | 21 | 21 | 21 |

Table 4.13 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 18 | 157 | 364 | 1 | 362 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 19 | 160 | 362 | 1 | 359 | 2 | 3 | 3 | 3 | 0 | 0 | 0 |
| 20 | 163 | 493 | 2 | 464 | 2 | 29 | 29 | 29 | 29 | 29 | 29 |
| 21 | 165 | 393 | 6 | 375 | 8 | 18 | 18 | 18 | 18 | 18 | 18 |
| 22 | 175 | 448 | 2 | 445 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 23 | 176 | 376 | 5 | 354 | 2 | 22 | 22 | 22 | 22 | 22 | 22 |
| 24 | 183 | 396 | 1 | 390 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 25 | 190 | 421 | 1 | 416 | 2 | 5 | 5 | 5 | 0 | 0 | 0 |
| 25 | 196 | 295 | 1 | 280 | 2 | 15 | 15 | 15 | 15 | 15 | 15 |
| 27 | 207 | 344 | 2 | 341 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
| 28 | 211 | 501 | 2 | 497 | 1 | 4 | 4 | 4 | 4 | 4 | 4 |
| 29 | 229 | 348 | 1 | 339 | 1 | 9 | 9 | 0 | 0 | 0 | 0 |
| 30 | 232 | 340 | 2 | 337 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 31 | 235 | 332 | 1 | 330 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 32 | 243 | 425 | 1 | 399 | 5 | 26 | 26 | 26 | 26 | 26 | 26 |
| 33 | 245 | 380 | 3 | 367 | 4 | 13 | 13 | 13 | 13 | 13 | 9 |
| 34 | 249 | 419 | 2 | 410 | 3 | 9 | 9 | 9 | 9 | 9 | 9 |
| 35 | 257 | 521 | 6 | 513 | 3 | 8 | 8 | 8 | 8 | 8 | 8 |
| 36 | 264 | 395 | 1 | 379 | 8 | 16 | 16 | 16 | 16 | 16 | 16 |

Table 4.13 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 37 | 271 | 495 | 2 | 485 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |
| 38 | 275 | 308 | 1 | 304 | 3 | 4 | 4 | 4 | 4 | 4 | 4 |
| 39 | 280 | 408 | 3 | 406 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 40 | 288 | 383 | 2 | 357 | 2 | 26 | 26 | 9 | 9 | 9 | 9 |
| 41 | 293 | 449 | 1 | 429 | 6 | 20 | 17 | 14 | 14 | 14 | 14 |
| 42 | 307 | 384 | 2 | 355 | 1 | 29 | 29 | 29 | 29 | 29 | 29 |
| 43 | 326 | 401 | 1 | 381 | 7 | 20 | 20 | 20 | 20 | 20 | 20 |
| 44 | 330 | 421 | 2 | 415 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 45 | 331 | 450 | 2 | 431 | 3 | 19 | 19 | 19 | 19 | 19 | 19 |
| 46 | 332 | 472 | 1 | 470 | 1 | 2 | 2 | 0 | 0 | 0 | 0 |
| 47 | 346 | 383 | 2 | 365 | 1 | 18 | 18 | 18 | 18 | 18 | 18 |
| 48 | 347 | 453 | 1 | 452 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 49 | 352 | 390 | 2 | 386 | 1 | 4 | 4 | 4 | 4 | 4 | 4 |
| 50 | 377 | 452 | 2 | 443 | 2 | 9 | 9 | 9 | 9 | 9 | 9 |
| 51 | 391 | 493 | 3 | 471 | 1 | 22 | 22 | 22 | 22 | 22 | 22 |
| 52 | 398 | 334 | 1 | 329 | 1 | 5 | 5 | 5 | 5 | 5 | 5 |
| 53 | 403 | 302 | 2 | 287 | 1 | 15 | 15 | 15 | 15 | 15 | 15 |
| 54 | 406 | 360 | 5 | 342 | 9 | 18 | 18 | 18 | 18 | 0 | 0 |
| 55 | 428 | 424 | 2 | 408 | 1 | 16 | 16 | 16 | 16 | 16 | 16 |

Table 4.13 continued.

| No. | Problem Number | No Passing: Minimum Total Make-span | No Passing: Number of Optimal Sequences | Passing: Lowest Total Make-span | Passing: Number of Sequences | Passing: Maximum Saving | Plans: 1 | Plans: 2 | Plans: 3 | Plans: 4 | Plans: 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 56 | 432 | 349 | 2 | 342 | 1 | 7 | 7 | 7 | 7 | 7 | 7 |
| 57 | 464 | 428 | 2 | 421 | 1 | 7 | 7 | 7 | 7 | 7 | 7 |
| 58 | 467 | 405 | 5 | 395 | 3 | 10 | 10 | 10 | 10 | 10 | 10 |
| 59 | 480 | 324 | 2 | 292 | 6 | 32 | 32 | 32 | 32 | 32 | 32 |
| 60 | 483 | 353 | 2 | 337 | 3 | 16 | 16 | 16 | 16 | 16 | 16 |
| 61 | 484 | 460 | 1 | 459 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62 | 499 | 401 | 2 | 387 | 6 | 14 | 14 | 14 | 14 | 14 | 14 |
| Total | | 29,958 | 126 | 24,171 | 169 | 787 | 778 | 728 | 720 | 686 | 668 |
| Average | | 402.55 | 2.03 | 389.85 | 2.73 | 12.69 | 12.55 | 11.74 | 11.61 | 11.07 | 10.77 |

The values of the measurement function for improvement by passing is as follows:

$$f4 = \frac{MTMS - TMSP}{MTMS} = \frac{377.53 - 375.92}{377.53}$$

$$= \frac{1.61}{377.53} = .00427 \text{ or } 0.427\%$$

Table 4.14 summarizes the amount of improvement or savings in total make-spans from Phase One to Phase Two by either complete enumeration search or by search of the five plans. It also summarizes the amount of search work invovled in the five plans, comparing it to the search work of complete enumeration, and provides benefit-cost ratios.

From Table 4.14 it appears the amount of work involved in Plans One, Two and Three, as compared to complete enumeration, decreases very rapidly, starting from 100 percent for complete enumeration to 3.69 percent for Plan Three. However, the amount of savings in total make-spans compared to the maximum possible savings by complete enumeration decreases quite slowly from 100 percent to 88.74 percent. Thus, the benefits involved in Plans One, Two and Three seem clear. Plan Three looks particularly attractive. Plans Four and Five do not seem particularly attractive compared to Plan Three. The reduction of search work in Plans Four and Five does not appear significant. And compared to the reduction in work, the loss of savings in total make-spans seems little higher. The benefit cot ratio, as

Table 4.14

Amount of Savings in Make-spans and Amount of Search Work Involved with Passing

| | Complete Enumeration | Plans 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1. Amount of savings in the total make-spans | | | | | | |
| Replication 1 | 821 | 821 | 707 | 707 | 661 | 637 |
| Replication 2 | 787 | 778 | 728 | 720 | 686 | 668 |
| Total | 1608 | 1599 | 1435 | 1427 | 1347 | 1305 |
| Percent | 100 | 99.44 | 89.24 | 88.74 | 83.77 | 81.16 |
| 2. Number of solutions searched per problem | | | | | | |
| Total | 552 | 144 | 77.79 | 20.29 | 18.53 | 16.78 |
| Percent | 100 | 26.09 | 14.09 | 3.69 | 3.36 | 3.04 |
| 3. Benefit-cost ratio in terms of total savings divided by number of solutions searched | 2.91 | 11.10 | 18.45 | 70.35 | 72.60 | 77.75 |

can be seen from the table, increases substantially from complete enumeration to Plan Three. For Plans Four and Five the increase is fairly slow. It was decided that for the experiments of Chapter V, involving 4 job, 4 machine flow shop scheduling problems, only Plans One, Two and Three, together with complete enumeration, will be utilized.

## 5 Job, 4 Machine Flow Shop Scheduling Problems

The results obtained for the two replications of 500 problems for Phases One and Two are presented below.

### Phase One

Frequency distributions of the number of problems according to the minimum total make-span and the maximum total make-span, are presented in Tables 4.15 and 4.16.

As previously discussed in the 3 job and 4 job flow shop sections, many flow shop scheduling problems have more than one optimal sequence. It is certain that the average number of optimum sequences per problem increases as the size of the flow shop increases. For the 3 job, 4 machine flow shop, there were 1.189 optimal sequences per problem, and for the 4 job, 4 machine flow shop, there were 1.691 optimal sequences per problem. For the 5 job, 4 machine flow shop, Table 4.17 provides the frequency distribution of the number of problems according to the number of optimal sequences. The average number of optimal sequences obtained

Table 4.15

Frequency Distribution of 5 Job, 4 Machine Problems According to Minimum Total Make Spans

| The Range of Minimum Total Make-spans | Frequencies | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| 261-280 | 3 | 1 | 4 |
| 281-300 | 1 | 6 | 7 |
| 301-320 | 6 | 8 | 14 |
| 321-340 | 15 | 3 | 18 |
| 341-360 | 32 | 30 | 62 |
| 361-380 | 40 | 45 | 85 |
| 381-400 | 60 | 59 | 119 |
| 401-420 | 76 | 77 | 153 |
| 421-440 | 63 | 69 | 132 |
| 441-460 | 73 | 64 | 137 |
| 461-480 | 48 | 45 | 93 |
| 481-500 | 41 | 38 | 79 |
| 501-520 | 19 | 29 | 48 |
| 521-540 | 12 | 16 | 28 |
| 541-560 | 10 | 7 | 17 |
| 561-580 | 0 | 3 | 3 |
| 581-600 | 1 | 0 | 1 |
| Total | 500 | 500 | 1000 |
| Average Total Minimum Make-span | 426.73 | 428.97 | 427.85 |
| Std. dev. of the Total Minimum Make-span | 54.33 | 54.84 | 54.58 |

Table 4.16

Frequency Distribution of 4 Job, 4 Machine Problems According to Maximum Total Make-spans

| The Range of Maximum Total Make-spans | Frequencies | | |
|---|---|---|---|
| | Replication 1 | Replication 2 | Total |
| 381-400 | 1 | 2 | 3 |
| 401-420 | 2 | 2 | 4 |
| 421-440 | 3 | 3 | 6 |
| 441-460 | 4 | 3 | 7 |
| 461-480 | 6 | 12 | 18 |
| 481-500 | 15 | 15 | 30 |
| 501-520 | 30 | 28 | 58 |
| 521-540 | 47 | 45 | 92 |
| 541-560 | 42 | 56 | 98 |
| 561-580 | 59 | 65 | 124 |
| 581-600 | 74 | 59 | 133 |
| 601-620 | 70 | 67 | 137 |
| 621-640 | 57 | 49 | 106 |
| 641-660 | 43 | 44 | 87 |
| 661-680 | 25 | 27 | 52 |
| 681-700 | 16 | 14 | 30 |
| 701-720 | 3 | 8 | 11 |
| 721-740 | 2 | 1 | 3 |
| 741-760 | 1 | 0 | 1 |
| Total | 500 | 500 | 1000 |
| Average Maximum Total Make-Span | 587.71 | 585.83 | 586.77 |
| Std. dev. of the Maximum Total Make-span | 57.13 | 59.10 | 58.20 |

Table 4.17

Frequency Distribution of the Number of Problems According to the Number of Optimal Sequences

| | Replication 1 | | Replication 2 | | Total | |
|---|---|---|---|---|---|---|
| Number Optimal Sequences O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. | Number of Problems N.P. | Total Number of Sequences N.P.*O.S. |
| 1 | 225 | 225 | 236 | 236 | 461 | 461 |
| 2 | 135 | 270 | 128 | 256 | 262 | 526 |
| 3 | 40 | 120 | 37 | 111 | 77 | 231 |
| 4 | 32 | 128 | 40 | 160 | 72 | 288 |
| 5 | 19 | 95 | 9 | 45 | 28 | 140 |
| 6 | 20 | 120 | 18 | 108 | 38 | 228 |
| 7 | 3 | 21 | 5 | 35 | 8 | 56 |
| 8 | 5 | 40 | 9 | 72 | 14 | 112 |
| 9 | 1 | 9 | 4 | 36 | 5 | 45 |
| 10 | 2 | 20 | 3 | 30 | 5 | 50 |
| 11 | 6 | 66 | 0 | 0 | 6 | 66 |
| 12 | 5 | 60 | 4 | 48 | 9 | 108 |
| 13-24 | 7 | 121 | 7 | 135 | 14 | 256 |
| | 500 | 1295 | 500 | 1272 | 1000 | 2567 |

are 2.567 per problem. This number is 51.8 percent higher than the 1.691 optimal sequences per problem for the 4 job, 4 machine flow shop and is 116 percent higher than the 1.189 optimal sequences per problem for the 3 job, 4 machine flow shop scheduling problem.

A total of 2567 optimal sequences for the 1000 problems, computes to an average of 2.567 optimal sequences per problem in Phase One for a 5 job, 4 machine flow shop scheduling problem. Each of these 2567 optimal sequences could be any one of the 5! = 120 different possible sequences. After studying the collected data, it seems clear that all 120 different possible sequences are equitably represented by the 2567 optimal sequences.

## Phase Two

For a total of 91 and 94 problems of the 500 problems each for replications 1 and 2, the best total make-spans in Phase Two were lower than the minimum total make-spans of Phase One. The actual amount of improvement acquired for these problems varied from 1 to 57 units, averaging 12.58 and 11.38 units per problem, respectively. Each of these problems had anywhere from 1 to 152 passing solutions whose best total make-span was lower than the minimum total make-span of Phase One, and it amounted to an average of 7.71 and 8.7 passing solutions per problem for replications 1 and 2. The two tables presented in the following pages correspond to replications 1 and 2 of the

Table 4.18

5 Job, 4 Machine Flow Shop Scheduling Problems--Replication 1

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 1 | 8 | 484 | 1 | 475 | 2 | 9 | 9 | 2 | 2 | 2 | 2 |
| 2 | 10 | 400 | 4 | 388 | 13 | 12 | 12 | 12 | 12 | 12 | 12 |
| 3 | 23 | 523 | 3 | 508 | 10 | 15 | 15 | 0 | 0 | 0 | 0 |
| 4 | 35 | 403 | 1 | 396 | 2 | 7 | 7 | 0 | 0 | 0 | 0 |
| 5 | 36 | 501 | 5 | 497 | 2 | 4 | 4 | 4 | 4 | 4 | 4 |
| 6 | 38 | 350 | 1 | 348 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | 49 | 379 | 2 | 376 | 5 | 3 | 3 | 3 | 3 | 3 | 3 |
| 8 | 50 | 432 | 2 | 405 | 15 | 27 | 27 | 6 | 6 | 6 | 6 |
| 9 | 51 | 486 | 1 | 475 | 3 | 11 | 11 | 11 | 11 | 11 | 11 |
| 10 | 56 | 537 | 6 | 480 | 21 | 57 | 57 | 57 | 57 | 57 | 57 |
| 11 | 68 | 433 | 4 | 429 | 1 | 4 | 4 | 4 | 4 | 4 | 4 |
| 12 | 72 | 428 | 2 | 403 | 21 | 25 | 25 | 0 | 0 | 0 | 0 |
| 13 | 86 | 428 | 1 | 404 | 2 | 24 | 24 | 17 | 17 | 17 | 17 |
| 14 | 97 | 386 | 1 | 379 | 13 | 7 | 7 | 7 | 7 | 7 | 7 |
| 15 | 101 | 503 | 2 | 494 | 10 | 9 | 9 | 9 | 9 | 9 | 9 |
| 16 | 103 | 473 | 8 | 472 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 17 | 128 | 421 | 1 | 400 | 6 | 21 | 21 | 12 | 12 | 12 | 12 |
| 18 | 133 | 451 | 6 | 435 | 2 | 16 | 16 | 16 | 16 | 16 | 16 |

Table 4.18 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 19 | 134 | 396 | 1 | 392 | 4 | 4 | 4 | 0 | 0 | 0 | 0 |
| 20 | 139 | 488 | 2 | 468 | 14 | 20 | 20 | 0 | 0 | 0 | 0 |
| 21 | 141 | 474 | 6 | 461 | 10 | 13 | 13 | 13 | 13 | 13 | 13 |
| 22 | 145 | 435 | 2 | 405 | 77 | 30 | 30 | 30 | 30 | 30 | 30 |
| 23 | 166 | 387 | 2 | 382 | 6 | 4 | 5 | 0 | 0 | 0 | 0 |
| 24 | 172 | 500 | 6 | 481 | 9 | 19 | 19 | 19 | 19 | 19 | 19 |
| 25 | 176 | 509 | 2 | 484 | 9 | 25 | 25 | 25 | 25 | 25 | 25 |
| 26 | 177 | 457 | 2 | 447 | 2 | 10 | 10 | 10 | 10 | 10 | 10 |
| 27 | 179 | 334 | 2 | 320 | 1 | 14 | 14 | 14 | 14 | 14 | 14 |
| 28 | 180 | 451 | 5 | 430 | 3 | 21 | 21 | 21 | 21 | 21 | 21 |
| 29 | 185 | 416 | 6 | 399 | 1 | 17 | 17 | 17 | 17 | 17 | 17 |
| 30 | 192 | 381 | 1 | 368 | 1 | 13 | 13 | 13 | 13 | 13 | 13 |
| 31 | 193 | 466 | 2 | 465 | 5 | 1 | 1 | 0 | 0 | 0 | 0 |
| 32 | 196 | 454 | 2 | 453 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 33 | 197 | 505 | 4 | 504 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 34 | 200 | 406 | 1 | 398 | 1 | 8 | 8 | 0 | 0 | 0 | 0 |
| 35 | 202 | 422 | 1 | 396 | 11 | 26 | 26 | 0 | 0 | 0 | 0 |
| 36 | 206 | 410 | 1 | 408 | 1 | 2 | 0 | 2 | 0 | 0 | 0 |
| 37 | 210 | 456 | 2 | 452 | 4 | 4 | 4 | 0 | 0 | 0 | 0 |

Table 4.18 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 38 | 216 | 452 | 4 | 431 | 27 | 21 | 21 | 18 | 18 | 18 | 18 |
| 39 | 227 | 437 | 1 | 429 | 1 | 8 | 8 | 8 | 8 | 8 | 8 |
| 40 | 228 | 522 | 12 | 508 | 5 | 14 | 14 | 14 | 14 | 10 | 0 |
| 41 | 230 | 396 | 1 | 380 | 3 | 16 | 16 | 16 | 16 | 16 | 16 |
| 42 | 232 | 490 | 6 | 477 | 4 | 13 | 13 | 13 | 13 | 13 | 13 |
| 43 | 235 | 494 | 5 | 464 | 10 | 30 | 30 | 30 | 30 | 30 | 30 |
| 44 | 238 | 477 | 7 | 440 | 62 | 37 | 37 | 37 | 37 | 37 | 30 |
| 45 | 239 | 357 | 20 | 352 | 4 | 4 | 5 | 1 | 1 | 0 | 0 |
| 46 | 243 | 366 | 6 | 363 | 26 | 3 | 3 | 3 | 3 | 3 | 3 |
| 47 | 254 | 473 | 1 | 468 | 2 | 5 | 5 | 5 | 5 | 5 | 5 |
| 48 | 258 | 405 | 6 | 401 | 2 | 4 | 1 | 4 | 1 | 1 | 1 |
| 49 | 259 | 423 | 1 | 417 | 2 | 6 | 6 | 6 | 6 | 6 | 6 |
| 50 | 274 | 419 | 2 | 400 | 27 | 19 | 19 | 4 | 4 | 4 | 4 |
| 51 | 275 | 450 | 1 | 431 | 19 | 19 | 19 | 19 | 19 | 19 | 19 |
| 52 | 279 | 315 | 2 | 309 | 3 | 6 | 6 | 6 | 6 | 6 | 6 |
| 53 | 281 | 402 | 1 | 401 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 54 | 287 | 532 | 2 | 510 | 16 | 22 | 22 | 22 | 22 | 22 | 22 |
| 55 | 289 | 383 | 4 | 362 | 2 | 21 | 21 | 21 | 21 | 21 | 21 |
| 56 | 290 | 498 | 5 | 477 | 1 | 21 | 21 | 0 | 0 | 0 | 0 |

Table 4.18 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 57 | 291 | 427 | 1 | 389 | 15 | 38 | 38 | 38 | 38 | 38 | 38 |
| 58 | 299 | 492 | 1 | 490 | 5 | 2 | 2 | 2 | 2 | 2 | 2 |
| 59 | 305 | 546 | 2 | 540 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 60 | 307 | 491 | 1 | 484 | 5 | 7 | 7 | 7 | 7 | 7 | 7 |
| 61 | 313 | 424 | 12 | 398 | 29 | 26 | 26 | 26 | 26 | 26 | 26 |
| 62 | 322 | 414 | 3 | 403 | 1 | 11 | 11 | 11 | 11 | 11 | 11 |
| 63 | 331 | 384 | 1 | 382 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| 64 | 334 | 399 | 2 | 394 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| 65 | 337 | 470 | 2 | 461 | 18 | 9 | 2 | 9 | 2 | 2 | 2 |
| 66 | 346 | 419 | 4 | 415 | 1 | 4 | 4 | 0 | 0 | 0 | 0 |
| 67 | 350 | 509 | 1 | 496 | 6 | 13 | 13 | 0 | 0 | 0 | 0 |
| 68 | 355 | 384 | 2 | 372 | 3 | 12 | 12 | 12 | 12 | 12 | 12 |
| 69 | 358 | 529 | 2 | 524 | 4 | 4 | 5 | 5 | 5 | 5 | 5 |
| 70 | 360 | 483 | 1 | 472 | 1 | 11 | 11 | 11 | 11 | 11 | 11 |
| 71 | 369 | 333 | 4 | 327 | 7 | 6 | 6 | 6 | 6 | 6 | 6 |
| 72 | 375 | 486 | 6 | 467 | 4 | 19 | 19 | 19 | 19 | 19 | 19 |
| 73 | 391 | 534 | 1 | 523 | 4 | 11 | 11 | 11 | 11 | 11 | 11 |
| 74 | 395 | 521 | 2 | 508 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| 75 | 402 | 405 | 5 | 398 | 1 | 7 | 7 | 7 | 7 | 7 | 7 |

Table 4.18 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 76 | 403 | 349 | 2 | 331 | 5 | 18 | 18 | 18 | 18 | 18 | 18 |
| 77 | 411 | 392 | 1 | 384 | 3 | 8 | 8 | 8 | 8 | 8 | 8 |
| 78 | 413 | 463 | 3 | 453 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |
| 79 | 429 | 459 | 1 | 455 | 1 | 4 | 4 | 0 | 0 | 0 | 0 |
| 80 | 430 | 508 | 1 | 502 | 1 | 6 | 6 | 6 | 6 | 6 | 6 |
| 81 | 437 | 433 | 1 | 419 | 1 | 14 | 14 | 0 | 0 | 0 | 0 |
| 82 | 443 | 426 | 1 | 424 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 83 | 445 | 342 | 6 | 332 | 4 | 10 | 10 | 10 | 10 | 10 | 10 |
| 84 | 448 | 334 | 3 | 332 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 85 | 456 | 477 | 4 | 460 | 7 | 17 | 17 | 17 | 17 | 17 | 17 |
| 86 | 479 | 446 | 2 | 393 | 12 | 53 | 53 | 53 | 53 | 53 | 53 |
| 87 | 481 | 459 | 4 | 453 | 5 | 6 | 6 | 6 | 6 | 6 | 6 |
| 88 | 485 | 470 | 1 | 451 | 6 | 19 | 19 | 0 | 0 | 0 | 0 |
| 89 | 494 | 412 | 9 | 408 | 13 | 4 | 4 | 4 | 4 | 4 | 4 |
| 90 | 499 | 450 | 1 | 449 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 91 | 500 | 435 | 2 | 430 | 4 | 5 | 5 | 5 | 5 | 5 | 5 |
| Total | | 40,161 | 283 | 39,016 | 702 | 1145 | 1133 | 888 | 876 | 871 | 854 |
| Average | | 441.33 | 3.11 | 428.75 | 7.72 | 12.58 | 12.45 | 9.76 | 9.63 | 9.57 | 9.38 |

Table 4.19

5 Job, 4 Machine Flow Shop Scheduling Problems--Replication 2

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 1 | 8 | 565 | 10 | 555 | 5 | 10 | 10 | 10 | 10 | 10 | 10 |
| 2 | 14 | 486 | 1 | 480 | 2 | 6 | 6 | 6 | 6 | 6 | 6 |
| 3 | 18 | 442 | 17 | 434 | 13 | 8 | 8 | 8 | 8 | 8 | 8 |
| 4 | 21 | 469 | 4 | 366 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 5 | 32 | 442 | 2 | 441 | 5 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 36 | 427 | 1 | 402 | 5 | 25 | 25 | 0 | 0 | 0 | 0 |
| 7 | 42 | 499 | 1 | 468 | 62 | 31 | 31 | 31 | 31 | 31 | 31 |
| 8 | 50 | 557 | 3 | 556 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 52 | 458 | 2 | 435 | 2 | 23 | 23 | 23 | 23 | 23 | 23 |
| 10 | 56 | 424 | 2 | 414 | 2 | 10 | 10 | 10 | 10 | 10 | 10 |
| 11 | 62 | 414 | 5 | 388 | 25 | 26 | 26 | 0 | 0 | 0 | 0 |
| 12 | 69 | 439 | 1 | 434 | 1 | 5 | 5 | 0 | 0 | 0 | 0 |
| 13 | 75 | 444 | 2 | 441 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 14 | 80 | 413 | 2 | 394 | 4 | 19 | 19 | 15 | 15 | 15 | 15 |
| 15 | 87 | 498 | 3 | 476 | 4 | 22 | 22 | 22 | 22 | 22 | 22 |
| 16 | 92 | 455 | 1 | 449 | 3 | 4 | 6 | 6 | 6 | 6 | 6 |
| 17 | 101 | 386 | 5 | 385 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 4.19 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 18 | 102 | 402 | 4 | 392 | 4 | 10 | 10 | 10 | 10 | 10 | 10 |
| 19 | 104 | 393 | 2 | 389 | 1 | 4 | 4 | 0 | 0 | 0 | 0 |
| 20 | 133 | 509 | 4 | 466 | 152 | 43 | 43 | 40 | 40 | 40 | 40 |
| 21 | 134 | 390 | 1 | 383 | 1 | 7 | 7 | 7 | 7 | 7 | 7 |
| 22 | 135 | 488 | 1 | 479 | 3 | 9 | 9 | 9 | 9 | 9 | 9 |
| 23 | 136 | 485 | 1 | 479 | 2 | 6 | 6 | 6 | 6 | 6 | 6 |
| 24 | 143 | 420 | 2 | 389 | 18 | 31 | 31 | 31 | 31 | 31 | 31 |
| 25 | 152 | 395 | 2 | 394 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 26 | 154 | 462 | 5 | 439 | 8 | 23 | 23 | 23 | 23 | 23 | 23 |
| 27 | 165 | 366 | 2 | 364 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 28 | 169 | 507 | 2 | 505 | 2 | 2 | 2 | 0 | 0 | 0 | 0 |
| 29 | 172 | 508 | 1 | 498 | 2 | 10 | 10 | 3 | 0 | 0 | 0 |
| 30 | 173 | 437 | 4 | 405 | 6 | 32 | 32 | 32 | 32 | 32 | 32 |
| 31 | 176 | 478 | 1 | 471 | 7 | 7 | 7 | 0 | 0 | 0 | 0 |
| 32 | 180 | 387 | 2 | 346 | 9 | 41 | 41 | 41 | 41 | 41 | 41 |
| 33 | 193 | 428 | 1 | 404 | 18 | 24 | 24 | 22 | 22 | 22 | 22 |
| 34 | 200 | 544 | 2 | 507 | 6 | 37 | 37 | 37 | 37 | 37 | 37 |
| 35 | 204 | 477 | 1 | 451 | 3 | 26 | 26 | 26 | 26 | 26 | 26 |
| 36 | 208 | 392 | 4 | 388 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 37 | 209 | 413 | 1 | 403 | 1 | 10 | 10 | 10 | 10 | 10 | 10 |

Table 4.19 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 38 | 221 | 464 | 2 | 460 | 1 | 4 | 4 | 0 | 0 | 0 | 0 |
| 39 | 224 | 406 | 1 | 392 | 2 | 14 | 14 | 14 | 14 | 14 | 14 |
| 40 | 232 | 486 | 1 | 482 | 1 | 4 | 4 | 4 | 4 | 4 | 4 |
| 41 | 239 | 401 | 2 | 394 | 5 | 7 | 7 | 7 | 7 | 7 | 7 |
| 42 | 240 | 429 | 4 | 414 | 24 | 15 | 15 | 0 | 0 | 0 | 0 |
| 43 | 246 | 410 | 2 | 407 | 12 | 3 | 3 | 3 | 3 | 3 | 3 |
| 44 | 248 | 405 | 1 | 403 | 7 | 2 | 2 | 0 | 0 | 0 | 0 |
| 45 | 250 | 506 | 1 | 498 | 2 | 8 | 8 | 8 | 8 | 8 | 8 |
| 46 | 256 | 486 | 24 | 476 | 45 | 10 | 10 | 10 | 10 | 4 | 4 |
| 47 | 258 | 467 | 7 | 466 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 48 | 263 | 346 | 1 | 340 | 2 | 6 | 6 | 6 | 6 | 6 | 6 |
| 49 | 272 | 356 | 1 | 340 | 3 | 16 | 16 | 16 | 16 | 16 | 16 |
| 50 | 274 | 517 | 4 | 505 | 6 | 12 | 12 | 12 | 12 | 12 | 12 |
| 51 | 276 | 436 | 3 | 414 | 4 | 22 | 22 | 22 | 22 | 22 | 22 |
| 52 | 278 | 491 | 2 | 472 | 2 | 19 | 19 | 19 | 19 | 19 | 19 |
| 53 | 286 | 406 | 1 | 404 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 54 | 287 | 436 | 18 | 419 | 47 | 17 | 17 | 17 | 17 | 17 | 17 |
| 55 | 291 | 408 | 1 | 397 | 2 | 11 | 11 | 11 | 11 | 11 | 11 |
| 56 | 303 | 368 | 1 | 348 | 13 | 20 | 20 | 11 | 11 | 11 | 11 |
| 57 | 306 | 424 | 1 | 381 | 16 | 43 | 43 | 43 | 43 | 43 | 43 |

Table 4.19 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 58 | 309 | 518 | 4 | 487 | 25 | 31 | 31 | 31 | 31 | 31 | 31 |
| 59 | 313 | 402 | 2 | 388 | 7 | 14 | 14 | 0 | 0 | 0 | 0 |
| 60 | 314 | 404 | 4 | 396 | 10 | 8 | 8 | 6 | 6 | 6 | 6 |
| 61 | 322 | 413 | 4 | 411 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 62 | 323 | 365 | 2 | 359 | 3 | 6 | 6 | 6 | 6 | 6 | 6 |
| 63 | 327 | 449 | 3 | 443 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 64 | 331 | 467 | 2 | 463 | 5 | 4 | 4 | 4 | 4 | 4 | 4 |
| 65 | 343 | 462 | 6 | 457 | 8 | 5 | 5 | 5 | 5 | 5 | 5 |
| 66 | 348 | 417 | 6 | 412 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 67 | 350 | 424 | 1 | 419 | 1 | 5 | 5 | 5 | 5 | 5 | 5 |
| 68 | 355 | 527 | 5 | 520 | 1 | 7 | 7 | 0 | 0 | 0 | 0 |
| 69 | 360 | 439 | 4 | 430 | 5 | 9 | 9 | 9 | 9 | 9 | 9 |
| 70 | 366 | 416 | 6 | 415 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 71 | 374 | 405 | 3 | 393 | 14 | 13 | 12 | 12 | 12 | 12 | 12 |
| 72 | 380 | 453 | 11 | 440 | 34 | 13 | 13 | 13 | 13 | 3 | 3 |
| 73 | 381 | 375 | 11 | 368 | 7 | 7 | 7 | 7 | 7 | 0 | 0 |
| 74 | 386 | 539 | 12 | 525 | 6 | 14 | 14 | 14 | 14 | 14 | 6 |
| 75 | 395 | 391 | 3 | 390 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 76 | 396 | 445 | 3 | 442 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 77 | 399 | 455 | 1 | 449 | 2 | 6 | 6 | 2 | 2 | 2 | 2 |

Table 4.19 continued.

| | No Passing | | | Passing | | | Plans | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Problem Number | Minimum Total Make-span | Number of Optimal Sequences | Lowest Total Make-span | Number of Sequences | Maximum Saving | 1 | 2 | 3 | 4 | 5 |
| 78 | 416 | 500 | 1 | 493 | 1 | 7 | 7 | 0 | 0 | 0 | 0 |
| 79 | 422 | 424 | 2 | 422 | 1 | 2 | 2 | 0 | 0 | 0 | 0 |
| 80 | 423 | 394 | 1 | 382 | 1 | 12 | 12 | 12 | 12 | 12 | 12 |
| 81 | 429 | 418 | 1 | 414 | 2 | 4 | 4 | 0 | 0 | 0 | 0 |
| 82 | 433 | 411 | 1 | 406 | 1 | 5 | 5 | 5 | 5 | 5 | 5 |
| 83 | 438 | 418 | 1 | 409 | 3 | 9 | 9 | 9 | 9 | 9 | 9 |
| 84 | 444 | 492 | 4 | 482 | 6 | 10 | 10 | 10 | 10 | 10 | 10 |
| 85 | 446 | 445 | 12 | 432 | 4 | 13 | 13 | 13 | 13 | 8 | 8 |
| 86 | 448 | 403 | 2 | 386 | 17 | 17 | 17 | 17 | 17 | 17 | 17 |
| 87 | 451 | 374 | 1 | 367 | 2 | 7 | 7 | 6 | 6 | 6 | 6 |
| 88 | 455 | 472 | 3 | 463 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 89 | 464 | 441 | 2 | 438 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |
| 90 | 468 | 363 | 1 | 362 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 91 | 477 | 394 | 1 | 393 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 92 | 485 | 505 | 3 | 482 | 15 | 23 | 23 | 14 | 14 | 14 | 14 |
| 93 | 498 | 456 | 1 | 432 | 10 | 24 | 24 | 24 | 24 | 24 | 24 |
| 94 | 500 | 455 | 1 | 452 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
| Total | | 41,378 | 313 | 40,304 | 818 | 1074 | 1073 | 905 | 902 | 874 | 866 |
| Average | | 440.19 | 3.33 | 428.77 | 8.7 | 11.43 | 11.42 | 9.63 | 9.60 | 9.30 | 9.21 |

5 job, 4 machine flow shop scheduling problem and are similar to earlier tables for 4 job, 4 machine flow shop scheduling problems.

From these tables it is easy to see the average savings in the total make-span by complete enumeration in Phase Two is 12.59 and 11.44 units per problem for the 91 and 94 problems of replications 1 and 2, respectively. Or for the total 91 + 94 = 185 problems the savings in total make-spans in Phase Two averages 11.99 units per problem. For the rest of 1000 - 185 = 815 problems there is no savings in total make-spans. Thus, for these 1000 problems, the average savings in total make-spans amounts to 11.99 * 185/1000 = 2.219 units per problem in Phase Two. The average minimum total make-span in Phase One was 427.85 units; thus the average minimum total make-span in Phase Two will be 427.85 - 2.219 = 425.63 units for the 1000 problems. The value of the measurement function of passing for the 5 job, 4 machine flow shop scheduling problem is:

$$f4 = \frac{\text{MTMS} - \text{TMSP}}{\text{MTMS}} = \frac{427.85 - 425.63}{427.85}$$

$$= \frac{2.219}{427.85} = .00519 \text{ or } 0.519\%$$

The actual values of the measurement function obtained for 3, 4, and 5 job, 4 machine flow shop scheduling problems, will be discussed later in this chapter.

As previously stated, the tables presented for the Phase Two experiments provide the amount of savings obtained in the total make-spans by complete enumeration and by the five search plans. Table 4.20 summarizes the relative savings in the total make-span and the amount of search work involved in the complete enumerations and five plans.

From Table 4.20, it seems the amount of search work involved in finding some improvement or savings in the total make-span reduces quickly, beginning at 100 percent for complete enumeration to 0.36 percent for Plan Three. But the amount of savings decreases very slowly starting from 100 percent for complete enumeration to 80.13 percent for Plan Three. Thus, the progressive increase in the usefulness of Plans One, Two and Three is established beyond doubt. Relatively speaking, Plan Four involves 87.3 percent as much work as Plan Three and provides 98.7 percent as much savings as that plan. Therefore, Plan Four seems advantageous in reducing the 12.7 percent of search work with a consequent 1.3 percent reduction in savings as compared to Plan Three. Plan Five when compared to Plan Four does not appear attractive. It amounts to 95.6 percent as much work as Plan Four, providing 98.57 percent as much savings as Plan Four. The reduction in the amount of work is 4.4 percent whereas the reduction in savings in the total make-span is 1.43 percent. Thus, although Plan Five seems adequate, it isn't as good as Plan Four. Only Plans One, Two, Three and Four for the 5 job, 4 machine flow shop scheduling problem will be used in Chapter V.

Table 4.20

Amount of Savings in the Total Make-span and the Amount of Search Work Involved with Passing

| | Complete Enumeration | Plans | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1. Saving of Total Make-span in | | | | | | |
| Replication 1 | 1145 | 1133 | 888 | 876 | 871 | 854 |
| Replication 2 | 1074 | 1073 | 905 | 902 | 874 | 866 |
| Total | 2219 | 2206 | 1793 | 1778 | 1745 | 1720 |
| Percent | 100 | 99.41 | 80.80 | 80.13 | 78.64 | 77.51 |
| 2. Number of Solutions Searched Per Problem | 14280 | 1200 | 605 | 51.34 | 44.80 | 42.82 |
| Percent | 100 | 8.42 | 4.24 | 0.36 | 0.31 | 0.3 |
| 3. Benefit-cost ratio in terms of total savings divided by number of solutions searched | 0.16 | 1.84 | 2.96 | 34.60 | 38.95 | 40.15 |

From Table 4.21, few facts emerge clearly. The total number of problems whose best total make-span in Phase Two is lower than the minimum total make-span of Phase One increases steadily from 105 for the 3 job shop to 130 for the 4 job shop and to 185 for the 5 job, 4 machine flow shop. The average number of passing solutions with a lower total make-span than the corresponding minimum total make-span, increases from 1.39 passing solutions per problem for the 3 job shop to 2.99 passing solutions for the 4 job and to 8.21 passing solutions for the 5 job, 4 machine flow shop. The average savings in the total make-span per problem decreases slightly from 12.72 units for the 3 job to 12.37 units for the 4 job and to 11.99 units for the 5 job, 4 machine flow shop. This slight decrease in average savings in the total make-span together with a fairly large increase in the number of problems with lower total make-span in Phase Two as compared to Phase One makes the average savings in the 1000 problems increase from 1.34 units per problem for the 3 job to 1.61 units for the 4 job and to 2.22 units for the 5 job, 4 machine flow shop. The value of the measurement function increases with the increase in the size of the flow shop.

Table 4.21 also indicates the percent savings in total make-spans compared to complete enumeration in a particular plan decreases from 3 job to 5 job, 4 machine flow shops. For example, in Plan Three, the savings is 94.91 percent for the complete enumeration saving for the

Table 4.21

Results of Phases One and Two in Summary Form for 3 Job, 4 Job and 5 Job, 4 Machine Flow Shop Scheduling Problems

| | 3 Job | 4 Job | 5 Job |
|---|---|---|---|
| **Phase One** | | | |
| 1. Average minimum total make-span | 323.28 | 377.53 | 427.85 |
| 2. Average maximum total make-span | 401.96 | 496.52 | 586.77 |
| 3. Number of optimal sequences per problem | 1.189 | 1.691 | 2.567 |
| **Phase Two** | | | |
| 4. Number of problems out of 1000, whose total make-span improved by permitting passing between sequences | 105 | 130 | 185 |
| 5. Average number of passing solutions per problem whose total make-spans were lower than the minimum total make-span of Phase One | 1.39 | 2.99 | 8.21 |
| 6. Average savings in the total make-span per problem for number of problems of row 4 | 12.72 | 12.37 | 11.99 |
| 7. Average savings in the total make-span per problem for 1000 problems | 1.34 | 1.61 | 2.22 |
| 8. Average minimum total make-span in Phase Two for 1000 problems | 321.94 | 375.92 | 425.63 |
| 9. f4 in percent | 0.415 | 0.427 | 0.519 |
| **Percent Savings in Total Make-spans Compared to Complete Enumeration** | | | |
| 10. Complete enumeration | 100 | 100 | 100 |

Table 4.21 continued.

| | 3 Job | 4 Job | 5 Job |
|---|---|---|---|
| 11. Plan One | 100 | 99.44 | 99.41 |
| 12. Plan Two | 94.91 | 89.24 | 80.80 |
| 13. Plan Three | 94.91 | 88.74 | 80.13 |
| 14. Plan Four | --- | 83.77 | 78.64 |
| 15. Plan Five | --- | 81.16 | 77.51 |
| Number of Passing Solutions Searched in Phase Two | | | |
| 16. Complete Enumeration | 30 | 552 | 14280 |
| 17. Plan One | 18 | 144 | 1200 |
| 18. Plan Two | 11.89 | 77.79 | 605 |
| 19. Plan Three | 7.13 | 20.29 | 51.34 |
| 20. Plan Four | --- | 18.53 | 44.80 |
| 21. Plan Five | --- | 16.78 | 42.82 |
| Percent Amount of Search Work Compared to Complete Enumeration | | | |
| 22. Complete enumeration | 100 | 100 | 100 |
| 23. Plan One | 60 | 26.09 | 8.42 |
| 24. Plan Two | 39.63 | 14.09 | 4.24 |
| 25. Plan Three | 23.78 | 3.69 | 0.36 |
| 26. Plan Four | --- | 3.36 | 0.31 |
| 27. Plan Five | --- | 3.04 | 0.30 |
| Benefit Cost Ratio in Terms of Total Savings Divided by Number of Passing Solutions Searched in Phase Two | | | |
| 28. Complete enumeration | 44.53 | 2.91 | 0.16 |
| 29. Plan One | 74.25 | 11.10 | 1.84 |
| 30. Plan Two | 106.78 | 18.45 | 2.96 |
| 31. Plan Three | 177.59 | 70.35 | 34.60 |
| 32. Plan Four | --- | 72.60 | 38.95 |
| 33. Plan Five | --- | 77.75 | 40.15 |

3 job, but is only 88.74 percent in the 5 job, 4 machine flow shop. The reason for this is that the percent of search work done as compared to complete enumeration, decreases significantly from the 3 job to 5 job, 4 machine flow shop. For example, for Plan Three, 23.78 percent of the complete enumeration solutions are examined for the 3 job shop, but only 3.69 percent solutions are examined for the 4 job and just 0.36 percent passing solutions are examined in the 5 job, 4 machine flow shop.

Rows 28-33 in Table 4.21 provides values for the benefit-cost ratio. Since benefit and cost are in different measurement units, a value of the ratio by itself does not provide significant information, but when one value of the ratio is compared with another value, the comparison can provide useful information.

It is clear from the table that the values of the benefit-cost ratio for all three sizes of the flow shop, increases with the increase in plan numbers. These increases indicate Plan Three for the 3 job, 4 machine flow shop and Plan Five for 4, and 5 Job, 4 machine flow shops are most efficient as far as the benefit-cost ratio is concerned. For the complete enumeration and all five plans, the value of the benefit-cost ratio decreases with the increase in flow shop sizes. This signifies that for the same amount of savings in the total make-span, increasing amount of search work is necessary with any increase in the size of the flow shop.

Concluding from all presentations discussions and analysis of results in this chapter, removing the no passing restriction is quite useful. Although with passing permitted between sequences, the number of solutions to be examined increases tremendously, the benefit of a lowered total make-span could make the extra search work worthwhile. The results presented in this chapter establish the viability of using partial search plans rather than complete enumeration. These various plans, particularly Plan Three, permit more than 80 percent of the benefit of the removal of the no passing restriction to be achieved at a relatively small fraction of the search work of the complete enumeration.

# CHAPTER V

## RESULTS OF ANALYSES WITH BETA DISTRIBUTED PROCESSING TIMES

The results presented in Chapter IV establish the phenomena of the lower total make-span with the removal of the no passing restriction. All the results in Chapter IV were obtained from solving flow shop scheduling problems whose processing times were generated using uniform probability distributions between 0 and 100. In this chapter the results are obtained from solving flow shop scheduling problems whose processing times are generated by using Beta probability distributions. As discussed in Chapter III, Beta probability distribution is quite flexible with an entirely different shape of the density function curve for different values of its parameters A and B. For the experiments of Chapter V, different shapes of probability density functions are obtained by choosing five sets of A and B values equal to 0.5, 1.5, 2.0, 2.5 and 3.0 for each of the 3 job, 4 machines; 4 job, 4 machines and the 5 job, 4 machines flow shop sizes. Two replications of 500 problems for each of the five sets of A and B values are generated. In other words, for each of the three sizes of flow shops, enough random numbers are generated for a total of 5000 problems for the experiments. When the values of A and B are equal to 1.0, the Beta probability distribution is the same as the uniform probability distribution, which is already covered by the experiments whose results are reported in Chapter IV. For purposes of comparison and filling the gap between the values of A and B from 0.5 to 1.5, some of the results of Chapter IV will be used again in this chapter.

## Explanation of Terms

Following is a list of shortened terms which will be used in place of the longer names in this chapter:

A B refers to the Beta distribution parameters A and B. (Throughout the experiments, parameters A and B have equal values.)

Tamin is the mean minimum total make-span value of Phase One averaged from all the problems of replications 1 and 2.

Tamax is the mean maximum total make-span value of Phase One, averaged from all the problems of replications 1 and 2.

Trange is Tamax-Tamin = mean value of the difference between the maximum and minimum total make-span of Phase

One, averaged over the total problems of replications 1 and 2.

Nop is the total number of problems in both replications for which the lowest total make-span in Phase Two was found to be lower than the minimum total make-span of Phase One.

Tanos is the mean value of the number of optimal sequences per problem in Phase One averaged from all the problems of replications 1 and 2.

MTMS is the minimum total make-span under no passing.

TMSP is the lowest (best) total make-span under passing.

## 3 Job, 4 Machine Flow Shop Scheduling Problems

In the following pages, three tables of results obtained during experiments on 3 job, 4 machine flow shop scheduling problems are presented. Table 5.1 provides results of Phase One experiments. For each of the six sets of A and B values, it provides the results obtained in replications 1 and 2, the total of the two replications. Each replication has 500 problems; Table 5.1 also shows the value of the average minimum total make-span, the average maximum total make-span, the range between the average maximum and the minimum total make-span, and the average number of optimal sequences per problem, averaged over the

Table 5.1

Results of Phase One Experiments of 3 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Repli- cation | Average Minimum Total Make-span | Average Maximum Total Make-span | Range | Average Number of Optimal Sequences Per Problem |
|---|---|---|---|---|---|
| | 1 | 330.75 | 424.86 | 94.11 | 1.162 |
| 0.5 | 2 | 328.50 | 424.83 | 96.32 | 1.200 |
| | Total | 329.63 | 424.84 | 95.21 | 1.186 |
| | 1 | 321.23 | 400.72 | 79.49 | 1.178 |
| 1.0 | 2 | 325.34 | 403.21 | 77.87 | 1.200 |
| | Total | 323.28 | 401.96 | 78.68 | 1.189 |
| | 1 | 322.41 | 389.66 | 67.24 | 1.166 |
| 1.5 | 2 | 319.28 | 385.93 | 66.64 | 1.222 |
| | Total | 320.84 | 387.79 | 66.94 | 1.194 |
| | 1 | 315.64 | 377.14 | 61.50 | 1.182 |
| 2.0 | 2 | 319.66 | 380.60 | 60.95 | 1.208 |
| | Total | 317.65 | 378.87 | 61.22 | 1.195 |
| | 1 | 315.02 | 371.15 | 56.13 | 1.192 |
| 2.5 | 2 | 317.24 | 372.58 | 55.35 | 1.220 |
| | Total | 316.13 | 371.87 | 55.74 | 1.201 |
| | 1 | 315.04 | 365.72 | 50.68 | 1.220 |
| 3.0 | 2 | 318.37 | 368.14 | 49.78 | 1.226 |
| | Total | 316.70 | 366.93 | 50.23 | 1.223 |

500 problems. The values in the row designated "Total" are averaged from 1000 problems of replications 1 and 2.

As A and B (which will be called AB from now on) values increase from 0.5 to 3.0, the minimum total make-span values average from the total of replications 1 and 2 (which will be called Tamin from now on), decrease slowly from 329.63 to 316.70. The decrease in Tamin is small and not continuous because as AB increases from 2.5 to 3.0, there is a slight increase in its value. On the other hand, the value of the maximum total make-span averaged over the total of replications 1 and 2 (Tamax) decreases fairly steadily. As a consequence, there is a larger decrease in Tamax and a smaller decrease in Tamin. The range of the total of replications 1 and 2 between Tamix and Tamix (Trange) decreases quite substantially, from 95.21 to 50.53 as AB values increase from 0.5 to 3.0. On the basis of Trange equal to 100, for AB equal to 0.5, other Trange values for AB at 1.0, 1.5, 2.0, 2.5 and 3.0, are 82.6, 70.3, 64.2, 58.5 and 52.8, respectively. The value of Trange indicates amount of difference between the best and worst values of the total make-span. So as the value of Trange decreases with an increase in the value of AB, the discrimination between the best and worst total make-spans becomes less and less. The value of the average number of optimal sequences per problem for the total of replications 1 and 2 (Tanos) increases with the increase in AB's value. As AB increases from 0.5 to 3.0, Tanos increases from 1.186 to 1.223, a total increase of 3.5 percent.

Table 5.2 summarizes results of Phase Two experiments on 3 job, 4 machine flow shop scheduling problems. For each replication and for the total of both replications for each value of AB, the table provides the number of problems (Nop) for which the best total make-span in Phase Two was found to be lower than the minimum total make-span of Phase One. The table also shows the total savings in the total make-span for all Nop. The total savings for all Nop obtained and presented in Table 5.2 are by the complete enumeration process and by the partial search Plans One, Two and Three.

The most significant fact that emerges from Table 5.2 is that with the increase in AB values, the decrease in the Nop is quite substantial. As AB values increase from 0.5 to 3.0, the values of the Nop decreases from 141 to 23, amounting to 83.7 percent decrease. Another observation that can be made from the table is that the value of the total savings obtained by the complete enumeration search, decreases steadily with the increase in the value of AB. As AB values increase from 0.5 to 3.0, the total savings obtained by complete enumeration decreases from 2291 to 166 time units. On the basis of 100 units of total savings for AB equal to 0.5, the total savings value decreases to 7.25 time units for AB equal to 3.0.

The amount of savings by the three partial search Plans One, Two and Three for the different values of AB are almost as much as the savings by the complete enumeration search. The total savings by Plan One is almost 100 percent for all the values of AB, and the total savings by Plans Two and Three varies from 90.57 to 98.11 percent. For the six values of AB, overall savings by Plans One, Two and Three are 99.92, 95.19, and 95.19 percent of the savings by complete enumeration.

Table 5.2

Results of Phase Two Experiments of 3 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Number of Problems | Complete Enumeration | Total Savings by Plan 1 | Total Savings by Plan 2 | Total Savings by Plan 3 |
|---|---|---|---|---|---|---|
| 0.5 | 1 | 65 | 1188 | 1184 | 1128 | 1128 |
| | 2 | 76 | 1103 | 1103 | 1078 | 1078 |
| | Total | 141 | 2291 | 2287 | 2206 | 2206 |
| | Percentage | | 100.00 | 99.82 | 96.29 | 96.29 |
| 1.0 | 1 | 52 | 595 | 595 | 593 | 593 |
| | 2 | 53 | 741 | 741 | 675 | 675 |
| | Total | 105 | 1336 | 1336 | 1268 | 1268 |
| | Percentage | | 100.00 | 100.00 | 94.91 | 94.91 |
| 1.5 | 1 | 25 | 263 | 263 | 235 | 235 |
| | 2 | 24 | 194 | 194 | 187 | 187 |
| | Total | 49 | 457 | 457 | 422 | 422 |
| | Percentage | | 100.00 | 100.00 | 92.34 | 92.34 |
| 2.0 | 1 | 13 | 117 | 117 | 115 | 115 |
| | 2 | 18 | 147 | 147 | 144 | 144 |
| | Total | 31 | 264 | 264 | 259 | 259 |
| | Percentage | | 100.00 | 100.00 | 98.11 | 98.11 |
| 2.5 | 1 | 9 | 75 | 75 | 61 | 61 |
| | 2 | 20 | 190 | 190 | 179 | 179 |
| | Total | 29 | 265 | 265 | 240 | 240 |
| | Percentage | | 100.00 | 100.00 | 90.57 | 90.57 |
| 3.0 | 1 | 12 | 79 | 79 | 67 | 67 |
| | 2 | 11 | 87 | 87 | 87 | 87 |
| | Total | 23 | 166 | 166 | 154 | 154 |
| | Percentage | | 100.00 | 100.00 | 92.77 | 92.77 |

Table 5.3 presents the average minimum total make-span under no passing and passing. Using these two values, the values of the measurement function f4 is computed and shown in that table. As the value of AB goes up, the value of f4 decreases. The decrease in f4 values as AB increases from 0.5 to 3.0, amounts to approximately 92.5 percent.

## 4 Job, 4 Machine Flow Shop Scheduling Problems

In the following pages, three tables of results obtained during experiments on 4 job, 4 machine scheduling problems are presented. These tables are similar to the ones presented earlier for 3 job, 4 machine flow shop scheduling problems. In Table 5.4, the values of Tamin, Tamax, Trange, and Tanos are given for the six values of AB. The Tamin values decrease very slowly and unsteadily as the values of AB increase. The total amount of decrease in Tamin is only 3.5 percent. On the other hand, Tamax values decrease fairly steadily as AB values increase. As a result of large and steady decreases in Tamax, and small, unsteady decreases in Tamin, Trange decreases substantially from 142.95 to 75.63, as AB values increase from 0.5 to 3.0. On the basis of Trange equal to 100, for AB equal to 0.5, other Trange values for AB equal to 1.0, 1.5, 2.0, 2.5 and 3.0, are 83.4, 71.0, 64.0, 58.2 and 53.0, respectively. These numbers obtained for 4 job, 4 machine

Table 5.3

Value of the Measurement Function f4 for 3 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Average Minimum Total Make-span Under No Passing (MTMS) | Average Minimum Total Make-span Under Passing (TMSP) | (MTMS-TMSP) | f4 in Percent $\left(\frac{MTMS-TMSP}{MTMS}\right)$ |
|---|---|---|---|---|---|
| | 1 | 330.75 | 328.37 | 2.38 | 0.720 |
| 0.5 | 2 | 328.50 | 326.29 | 2.21 | 0.673 |
| | Total | 329.63 | 327.34 | 2.29 | 0.695 |
| | 1 | 321.23 | 320.04 | 1.19 | 0.370 |
| 1.0 | 2 | 325.34 | 323.86 | 1.48 | 0.455 |
| | Total | 323.28 | 321.94 | 1.34 | 0.415 |
| | 1 | 322.41 | 321.88 | 0.53 | 0.164 |
| 1.5 | 2 | 319.28 | 318.90 | 0.38 | 0.119 |
| | Total | 320.84 | 320.38 | 0.46 | 0.143 |
| | 1 | 315.64 | 315.41 | 0.23 | 0.073 |
| 2.0 | 2 | 319.66 | 319.37 | 0.29 | 0.091 |
| | Total | 317.65 | 317.39 | 0.26 | 0.082 |
| | 1 | 315.02 | 314.87 | 0.15 | 0.048 |
| 2.5 | 2 | 317.24 | 316.86 | 0.38 | 0.112 |
| | Total | 316.13 | 315.86 | 0.27 | 0.085 |
| | 1 | 315.04 | 314.88 | 0.16 | 0.051 |
| 3.0 | 2 | 318.37 | 318.20 | 0.17 | 0.053 |
| | Total | 316.70 | 316.53 | 0.17 | 0.052 |

Table 5.4

Results of Phase One Experiments of 4 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Average Minimum Total Make-span | Average Maximum Total Make-span | Range | Average Number of Optimal Sequences Per Problem |
|---|---|---|---|---|---|
| | 1 | 381.44 | 523.43 | 141.99 | 1.732 |
| 0.5 | 2 | 381.42 | 525.33 | 143.91 | 1.650 |
| | Total | 381.43 | 524.38 | 142.95 | 1.691 |
| | 1 | 377.14 | 494.91 | 117.77 | 1.708 |
| 1.0 | 2 | 377.92 | 498.14 | 120.22 | 1.674 |
| | Total | 377.53 | 496.52 | 118.99 | 1.691 |
| | 1 | 372.73 | 474.64 | 101.91 | 1.746 |
| 1.5 | 2 | 372.64 | 473.20 | 100.56 | 1.738 |
| | Total | 372.68 | 473.92 | 101.24 | 1.742 |
| | 1 | 368.52 | 460.31 | 91.80 | 1.664 |
| 2.0 | 2 | 370.73 | 461.69 | 90.96 | 1.654 |
| | Total | 369.62 | 461.00 | 91.38 | 1.659 |
| | 1 | 367.75 | 449.66 | 81.91 | 1.750 |
| 2.5 | 2 | 367.40 | 451.48 | 84.08 | 1.760 |
| | Total | 367.57 | 450.57 | 83.00 | 1.755 |
| | 1 | 368.63 | 443.69 | 75.06 | 1.880 |
| 3.0 | 2 | 367.09 | 443.30 | 76.21 | 1.714 |
| | Total | 367.86 | 443.49 | 75.63 | 1.797 |

problems are quite similar to the 3 job, 4 machine problems reported earlier. Thus, the discrimination between the best and the worst total make-spans decreases substantially as AB values increase. The values of Tanos shown in Table 5.4 for 4 job, 4 machine problems, do not follow any particular pattern, but instead fluctuate a great deal. The only observation that can be drawn from the Tanos values is that on an average, Tanos for 4 job, 4 machine problems are 44.0 percent higher than the Tanos for 3 job, 4 machine problems.

Table 5.5 provides values of Nop and the values of total savings for all Nops by complete enumeration and by search Plans One, Two and Three. As experienced previously, Nop values decrease substantially from 186 to just 28, as values of AB go from 0.5 to 3.0. On the basis of Nop equal to 100 for AB equal to 0.5, 15.1 Nop are obtained for AB equal to 3.0. The values of the total savings for all Nop by complete enumeration decrease even faster than the decrease in the Nop. The total savings values drop from 3192 time units to 252 time units, obtained by complete enumeration as AB increases from 0.5 to 3.0. The drop in the total savings value amounts to a 92.1 percent. The amount of savings by partial search plans are almost as much as the savings by complete enumeration. For Plan One, savings vary from 99.30 to 100 percent of the complete enumeration savings, whereas savings by Plans Two and Three vary anywhere from 85.25 to 92.06 percent of the complete enumeration savings. Overall savings by Plans One, Two and Three are 99.53, 89.40, and 89.07 percent of the savings by complete enumeration.

Table 5.6 shows the average minimum total make-span under no passing and passing, as well as using these two values to provide the values of the measurement function f4. With an increase in AB values, f4 values decrease

Table 5.5

Results of Phase Two Experiments of 4 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Number or Problems | Complete Enumeration | Total Savings by Plan | | |
|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 |
| 0.5 | 1 | 89 | 1527 | 1520 | 1334 | 1334 |
| | 2 | 97 | 1665 | 1665 | 1537 | 1528 |
| | Total | 186 | 3192 | 3175 | 2871 | 2862 |
| | Percentage | | 100.00 | 99.47 | 89.94 | 89.66 |
| 1.0 | 1 | 68 | 821 | 821 | 707 | 707 |
| | 2 | 62 | 787 | 778 | 728 | 720 |
| | Total | 130 | 1608 | 1599 | 1435 | 1427 |
| | Percentage | | 100.00 | 99.44 | 89.24 | 88.74 |
| 1.5 | 1 | 42 | 445 | 439 | 376 | 370 |
| | 2 | 41 | 409 | 409 | 362 | 362 |
| | Total | 83 | 854 | 848 | 738 | 732 |
| | Percentage | | 100.00 | 99.30 | 86.42 | 85.71 |
| 2.0 | 1 | 32 | 314 | 314 | 291 | 291 |
| | 2 | 40 | 354 | 354 | 322 | 322 |
| | Total | 72 | 668 | 668 | 613 | 613 |
| | Percentage | | 100.00 | 100.00 | 91.77 | 91.77 |
| 2.5 | 1 | 18 | 141 | 141 | 100 | 100 |
| | 2 | 12 | 137 | 137 | 137 | 137 |
| | Total | 30 | 278 | 278 | 237 | 237 |
| | Percentage | | 100.00 | 100.00 | 85.25 | 85.25 |
| 3.0 | 1 | 17 | 152 | 152 | 141 | 141 |
| | 2 | 11 | 100 | 100 | 91 | 91 |
| | Total | 28 | 252 | 252 | 232 | 232 |
| | Percentage | | 100.00 | 100.00 | 92.06 | 92.06 |

Table 5.6

Value of the Measurement Function f4 for 4 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Average Minimum Total Make-span Under No Passing (MTMS) | Average Minimum Total Make-span Under Passing (TMSP) | (MTMS-TMSP) | f4 in Percent $\left(\frac{MTMS-TMSP}{MTMS}\right)$ |
|---|---|---|---|---|---|
| | 1 | 381.44 | 378.38 | 3.06 | 0.802 |
| 0.5 | 2 | 381.42 | 378.09 | 3.33 | 0.873 |
| | Total | 381.43 | 378.24 | 3.19 | 0.836 |
| | 1 | 377.14 | 375.50 | 1.64 | 0.435 |
| 1.0 | 2 | 377.92 | 376.35 | 1.57 | 0.415 |
| | Total | 377.53 | 375.92 | 1.61 | 0.426 |
| | 1 | 372.73 | 371.84 | 0.89 | 0.239 |
| 1.5 | 2 | 372.64 | 371.82 | 0.82 | 0.220 |
| | Total | 372.68 | 371.83 | 0.85 | 0.228 |
| | 1 | 368.52 | 367.89 | 0.63 | 0.171 |
| 2.0 | 2 | 370.73 | 370.02 | 0.71 | 0.192 |
| | Total | 369.62 | 368.95 | 0.67 | 0.181 |
| | 1 | 367.75 | 367.47 | 0.28 | 0.076 |
| 2.5 | 2 | 367.40 | 367.13 | 0.27 | 0.073 |
| | Total | 367.57 | 367.29 | 0.28 | 0.075 |
| | 1 | 368.63 | 368.33 | 0.30 | 0.081 |
| 3.0 | 2 | 367.07 | 366.89 | 0.20 | 0.054 |
| | Total | 367.86 | 367.61 | 0.25 | 0.068 |

steadily. The amount of decrease in f4 values is 91.9 percent as AB values increase from 0.5 to 3.0.

## 5 Job, 4 Machine Flow Shop Scheduling Problems

In the following pages, three tables of results obtained during experiments on 5 job, 4 machine flow shop scheduling problems are presented. These tables are similar to the tables presented earlier for 3 and 4 job, 4 machine problems. Table 5.7 provides Tamin, Tamax, Trange and Tanos values. As AB values increase, the values of Tamin, Trange, and Tamax decrease. The decrease in Tamin is small amounting to only 2.88 percent, whereas the decrease in Tamax is a little larger, resulting in 16.45 percent as AB values increase from 0.5 to 3.0. Since the Trange values are the difference of the Tamax and Tamin values, and since the decrease in Tamax is greater than the Tamin, Trange values decrease steadily as AB values go from 0.5 to 3.0. The total decrease in Trange is from 191.80 to 102.07 units, amounting to a 46.75 percent decrease. It indicates the discrimination between the best and the worst total make-span for AB equal to 3.0 is half as much as the discrimination for AB equal to 0.5. As can be seen from the tables, Tanos does not follow any particular pattern of change. Overall, it seems that the Tanos values of 5 job, 4 machine problems are 65 percent greater than the Tanos values of the 4 job, 4 machine problems and are 137 percent

Table 5.7

Results of Phase One Experiments of 5 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Repli-cation | Average Minimum Total Make-span | Average Maximum Total Make-span | Range | Average Number of Optimal Sequences Per Problem |
|---|---|---|---|---|---|
| | 1 | 431.02 | 622.53 | 191.51 | 2.820 |
| 0.5 | 2 | 427.33 | 619.41 | 192.08 | 2.680 |
| | Total | 429.17 | 620.97 | 191.80 | 2.750 |
| | 1 | 426.73 | 587.71 | 160.98 | 2.590 |
| 1.0 | 2 | 428.97 | 585.83 | 156.86 | 2.544 |
| | Total | 427.85 | 586.77 | 158.92 | 2.567 |
| | 1 | 427.66 | 563.80 | 136.14 | 3.030 |
| 1.5 | 2 | 420.92 | 556.62 | 135.70 | 2.858 |
| | Total | 424.29 | 556.62 | 135.92 | 2.944 |
| | 1 | 422.22 | 544.68 | 122.46 | 2.880 |
| 2.0 | 2 | 423.28 | 544.51 | 121.23 | 3.278 |
| | Total | 422.75 | 544.59 | 121.84 | 3.079 |
| | 1 | 416.49 | 526.89 | 110.39 | 2.932 |
| 2.5 | 2 | 418.79 | 529.07 | 110.28 | 2.796 |
| | Total | 417.64 | 527.98 | 110.34 | 2.864 |
| | 1 | 416.78 | 520.45 | 103.67 | 2.786 |
| 3.0 | 2 | 416.84 | 517.31 | 100.47 | 2.882 |
| | Total | 416.81 | 518.88 | 102.07 | 2.834 |

greater than the Tanos values of the 3 job, 4 machine problems.

Table 5.8 provides the Nop and the total savings for all Nop by complete enumeration search and by the search of Plans One, Two, Three and Four. As experienced before, Nop values drop tremendously, from 237 to 45, amounting to a net 81 percent decrease as AB values increase from

Table 5.8

Results of Phase Two Experiments of 5 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Number of Problems | Complete Enumeration | Total Savings by Plan 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|
| 0.5 | 1 | 117 | 1688 | 1687 | 1323 | 1303 | 1275 |
| | 2 | 120 | 1960 | 1947 | 1426 | 1422 | 1386 |
| | Total | 237 | 3648 | 3634 | 2749 | 2725 | 2662 |
| | Percentage | | 100.00 | 99.60 | 75.40 | 74.75 | 73.00 |
| 1.0 | 1 | 91 | 1145 | 1133 | 888 | 876 | 871 |
| | 2 | 94 | 1074 | 1073 | 905 | 902 | 874 |
| | Total | 185 | 2219 | 2206 | 1793 | 1778 | 1745 |
| | Percentage | | 100.00 | 99.41 | 80.80 | 78.64 | 77.51 |
| 1.5 | 1 | 63 | 621 | 621 | 538 | 536 | 485 |
| | 2 | 63 | 619 | 588 | 543 | 512 | 498 |
| | Total | 126 | 1240 | 1209 | 1081 | 1048 | 983 |
| | Percentage | | 100.00 | 97.50 | 87.18 | 84.52 | 79.27 |
| 2.0 | 1 | 38 | 383 | 383 | 285 | 285 | 258 |
| | 2 | 41 | 362 | 361 | 296 | 296 | 292 |
| | Total | 79 | 745 | 744 | 581 | 581 | 550 |
| | Percentage | | 100.00 | 99.90 | 77.99 | 77.99 | 73.83 |
| 2.5 | 1 | 33 | 235 | 235 | 159 | 159 | 159 |
| | 2 | 31 | 200 | 174 | 143 | 130 | 130 |
| | Total | 64 | 435 | 409 | 302 | 289 | 289 |
| | Percentage | | 100.00 | 94.02 | 69.43 | 66.44 | 66.44 |
| 3.0 | 1 | 19 | 106 | 106 | 76 | 76 | 76 |
| | 2 | 26 | 158 | 158 | 142 | 142 | 142 |
| | Total | 45 | 264 | 264 | 218 | 218 | 218 |
| | Percentage | | 100.00 | 100.00 | 82.58 | 82.58 | 82.58 |

0.5 to 3.0. The amount of savings by the search plans are not as high as compared to the complete enumeration search as they were for the 3 and 4 job, 4 machine flow shop scheduling problems. Overall, the savings of Plans One, Two, Three and Four amounts to 99.0, 78.6, 77.5, and 75.3 percent of the savings by complete enumeration.

Table 5.9 provides the average minimum total make-span under both no passing and passing. Using these two values, the value of the measurement function f4 is computed and shown in the table. The value of the measurement function f4 drops steadily as the values of AB increase. The decrease in f4 values amounts to 92.7 percent as AB values go from 0.5 to 3.0.

## Summary

Table 5.10 shows a number of problems for which permitting passing between sequences improved the minimum total make-span of Phase One in Phase Two for the values of AB equal to 0.5 to 3.0 and for the flow shop sizes of 3 to 5 job, 4 machine. The number of problems given in each category are from a total of 1000 problems. It is obvious from the table that the number of problems increases with the increase in the number of jobs in the flow shop and also with the decrease in AB values. For example, in the case of the 3 job, 4 machine flow shop and for the AB value of 3.0, only 2.3 percent of the problems have a lower total make-span in Phase Two than Phase One whereas 23.7

Table 5.9

Value of Measurement Function f4 for 5 Job, 4 Machine Flow Shop Scheduling Problems

| A and B Values | Replication | Average Minimum Total Make-span Under No Passing (MTMS) | Average Minimum Total Make-span Under Passing (TMSP) | (MTMS-TMSP) | f4 in Percent $\left(\frac{MTMS-TMSP}{MTMS}\right)$ |
|---|---|---|---|---|---|
| | 1 | 431.02 | 427.64 | 3.38 | 0.784 |
| 0.5 | 2 | 427.33 | 423.41 | 3.92 | 0.917 |
| | Total | 429.17 | 425.53 | 3.65 | 0.850 |
| | 1 | 426.73 | 424.44 | 2.29 | 0.537 |
| 1.0 | 2 | 428.97 | 426.82 | 2.15 | 0.501 |
| | Total | 427.85 | 425.63 | 2.22 | 0.519 |
| | 1 | 427.66 | 426.42 | 1.24 | 0.290 |
| 1.5 | 2 | 420.92 | 419.68 | 1.24 | 0.295 |
| | Total | 424.29 | 423.05 | 1.24 | 0.292 |
| | 1 | 422.22 | 421.45 | 0.77 | 0.182 |
| 2.0 | 2 | 423.28 | 422.56 | 0.72 | 0.170 |
| | Total | 422.75 | 422.00 | 0.75 | 0.177 |
| | 1 | 416.49 | 416.02 | 0.47 | 0.113 |
| 2.5 | 2 | 418.79 | 418.39 | 0.40 | 0.096 |
| | Total | 417.64 | 417.21 | 0.43 | 0.103 |
| | 1 | 416.78 | 416.56 | 0.22 | 0.053 |
| 3.0 | 2 | 416.84 | 416.52 | 0.32 | 0.077 |
| | Total | 416.81 | 416.55 | 0.26 | 0.062 |

Table 5.10

Number of Problems Where Passing Improved the Total Make-span (Nop)

| A and B Values | Size of the Flow Shop: 3 Job, 4 Machine | 4 Job, 4 Machine | 5 Job 4 Machine |
|---|---|---|---|
| 0.5 | 141 | 186 | 237 |
| 1.0 | 105 | 130 | 185 |
| 1.5 | 49 | 83 | 126 |
| 2.0 | 31 | 72 | 79 |
| 2.5 | 29 | 30 | 64 |
| 3.0 | 23 | 28 | 45 |

percent of the problems have lower total make-spans in Phase Two compared to Phase One for the 5 job, 4 machine flow shop and AB value of 0.5.

Figure 5.1 pictorially shows the effect of the increases in AB values and the effect of increases in the size of the flow shop on the number of problems in which the lowest total make-span of Phase Two was lower than the minimum total make-span of Phase One. The increase in the number of problems with the increase in the size of the flow shop for AB values equal to 0.5, 1.0, and 1.5 is phenomenal.

Table 5.11 presents values of the measurement function f4 for 3, 4 and 5 job, 4 machine flow shop scheduling problems and for the AB values of 0.5 to 3.0. The values

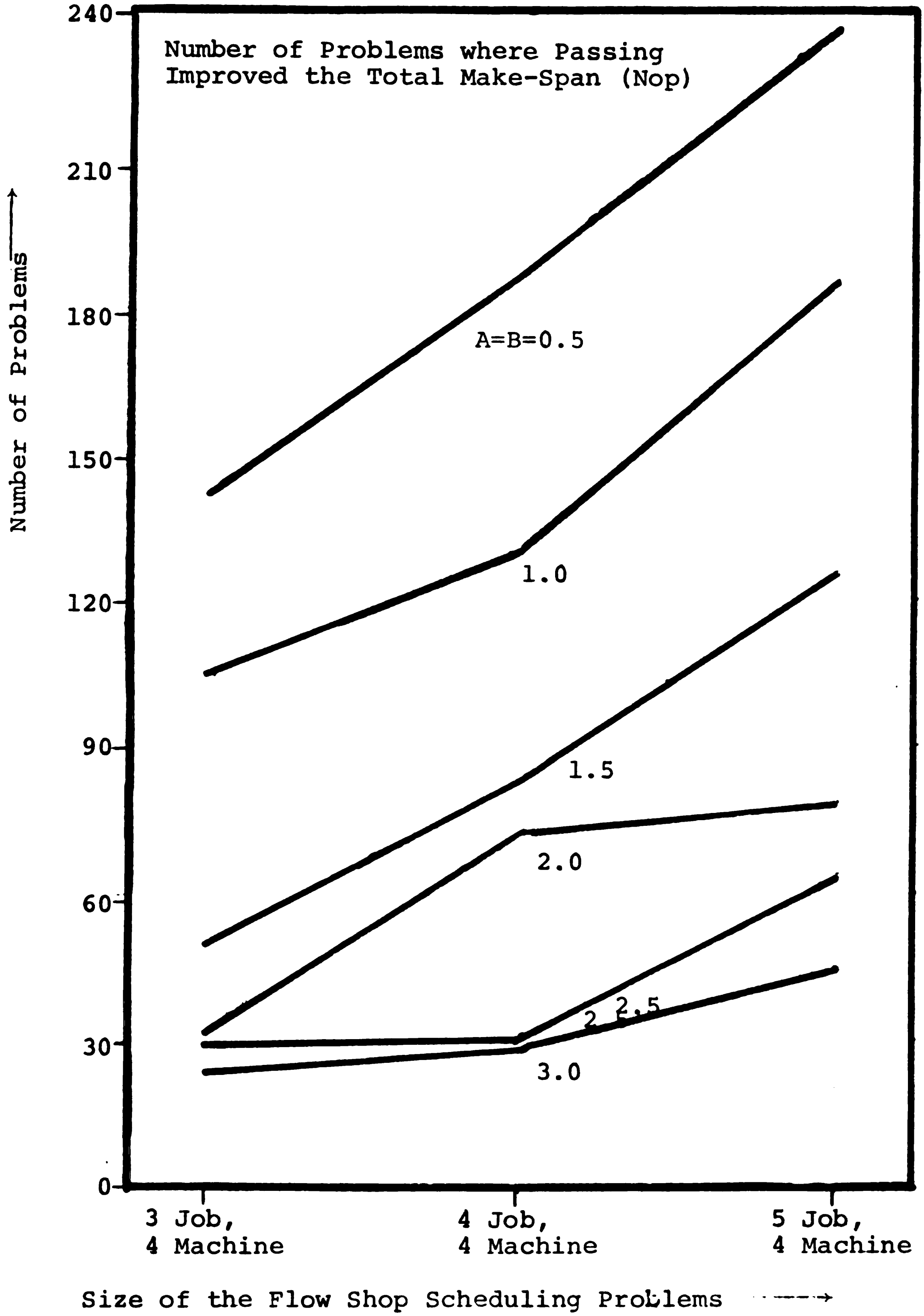


Figure 5.1 Number of Problems where Passing Improved the Total Make-Span (Nop).

Table 5.11

Value of the Measurement Function f4 in Percent

| | Size of the Flow Shop | | |
|---|---|---|---|
| A and B Values | 3 Job, 4 Machine | 4 Job, 4 Machine | 5 Job, 4 Machine |
| 0.5 | 0.695 | 0.836 | 0.850 |
| 1.0 | 0.415 | 0.426 | 0.519 |
| 1.5 | 0.143 | 0.228 | 0.292 |
| 2.0 | 0.082 | 0.181 | 0.171 |
| 2.5 | 0.085 | 0.075 | 0.103 |
| 3.0 | 0.052 | 0.068 | 0.062 |

of f4 as given in the table varies from 0.052 to 0.850 percent. The f4 values for each of the three flow shop sizes, except in one instance, decrease with the increase in AB values. Thus it can be safely assumed that the values of f4 will decrease with the increase in AB values for a given size of flow shop scheduling problems.

In a similar way, for a given AB value, as the size of the flow shop scheduling problem increases, the value of the measurement function f4 increases, except in a few instances. Figure 5.2 shows pictorially the behavior of the measurement function with the increase in the size of the flow shop and with the increase in AB values. From Table 5.11 and from Figure 5.2, it can be said that in general the value of f4 increases with both the increase in

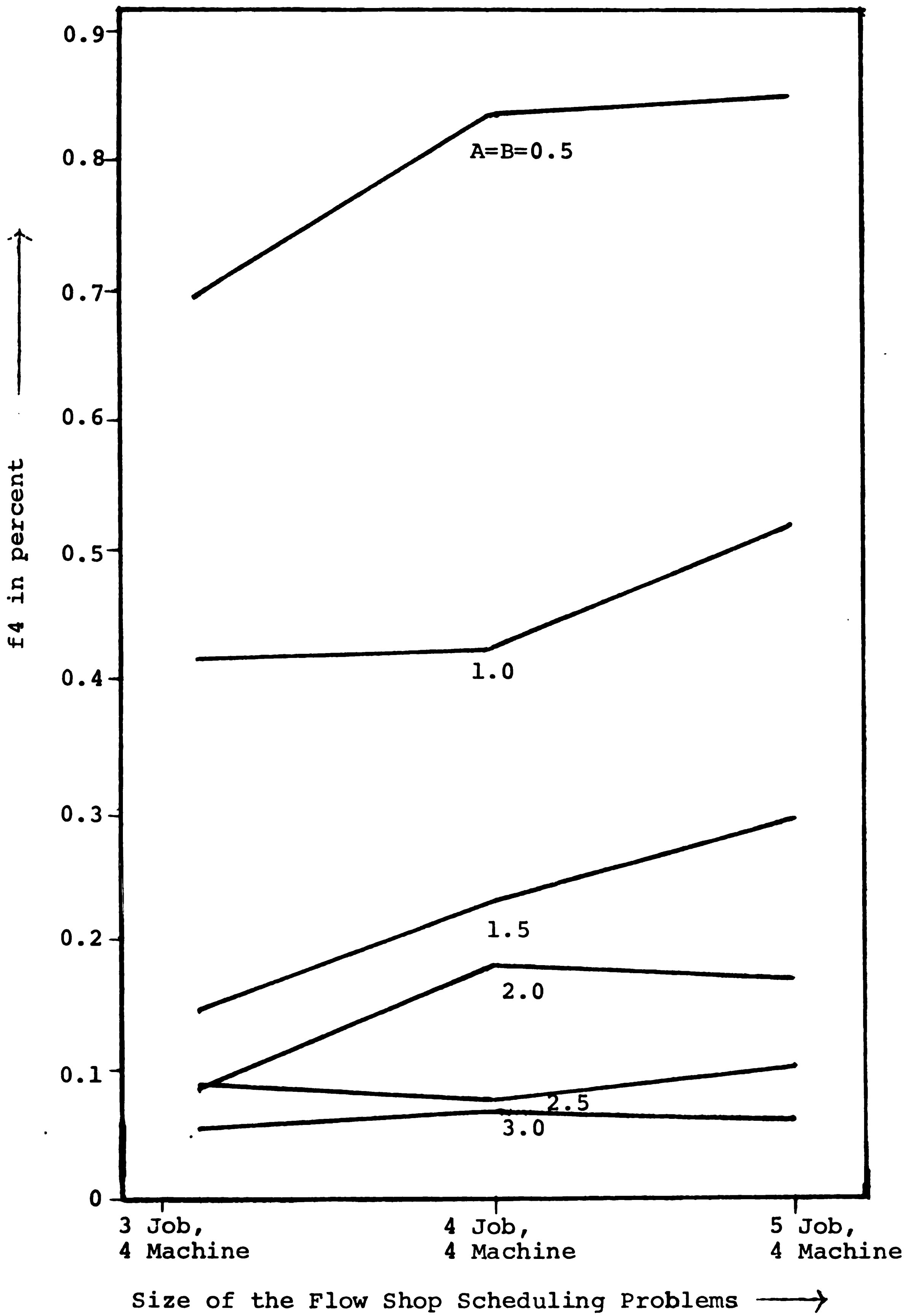


Figure 5.2 Value of the Measurement Function f4.

the number of jobs in the flow shop and with the decrease in the values of AB.

# CHAPTER VI

# SUMMARY OF FINDINGS AND RECOMMENDATIONS FOR FURTHER RESEARCH

## Findings and Conclusions

The main purpose of this dissertation was to study and analyze the effects of permitting passing between sequences of solutions to flow shop scheduling problems. The various results presented in Chapters IV and V proved beyond a doubt that, in a large number of problems, permitting passing improves the minimum total make-span obtained with no passing. Thus, if global optimization of a flow shop scheduling problem is desired, it is necessary that complete enumeration of all possible solutions with passing between sequences be considered. The complete enumeration of all no passing and passing solutions for even a small size flow shop scheduling problem involves a great deal of effort. So if global optimization in a particular situation is not necessary, then some heuristic, if available, can provide fairly good solutions to flow shop scheduling problems with much less computational efforts.

Until now not even a single heuristic was available to reduce the minimum total make-span beyond the no passing stage for flow shop scheduling problems where passing was permitted.

This dissertation provides three heuristics or plans, each of which can provide solutions which are very close to global optimum solutions with only a small fraction of effort required as compared to that required by complete enumeration. Plans Three, Two, and One gradually require more and more computational efforts to obtain progressively closer approximations to the global optimum solutions to flow shop scheduling problems. For example, for the one thousand of 5 job, 4 machine flow shop scheduling problems, Plans Three, Two and One provided 80.13, 80.80, and 99.41 percent of the possible reduction of the total make-span between the local optimum of no passing and global optimum of passing with 0.36, 4.24, and 8.42 percent computational efforts compared to 100 percent reduction and 100 percent computational efforts by complete enumeration. These three plans can be used only if all or most of the optimal solutions under the no passing restriction are available.

To obtain all optimal solutions under no passing, two approaches have been suggested in this dissertation. One is the complete enumeration of all the no passing solutions; the other is by a modified version of Ignall and Schrage's branch and bound technique. A user who is looking for a heuristic will find three choices in this dissertation, and he can make his choice on the basis of the availability of computational efforts and his needs. Although these three plans are specifically tailored for the smaller flow shop scheduling problems with exactly four machines, they are simple enough so that any user can modify them to make them applicable to larger sizes of flow shop scheduling problems.

The results in Chapter IV illustrate that if the processing times are uniformly generated, then 10.5, 13.0, and 18.5 percent of the 3 job, 4 machine; 4 job, 4 machine; and 5 job, 4 machine flow shop scheduling problems have a lower value of total make-span with passing compared to the minimum total make-spans obtained with no passing. It is easy to see that as the number of jobs in the flow shop scheduling problem increase, the number of problems for which permitting passing lowered the total make-span, also increased. Results presented in Chapter V also support this contention. In fact, in observing the results of Chapter V, it seems that the number of problems for which passing was permitted lowered the minimum total make-span increase on an average of 49.1 percent as the size of the flow shop scheduling problem increases by one job. Such a large increase may or may not continue as the number of jobs in the flow shop increases beyond five. But if it does, it is conceivable that for a relatively large size problem such as a 10 job, 4 machine problem, in each instance passing can provide a lower total make-span compared to the minimum total make-span with no passing.

It was discovered through experimentation that the amount of reduction in the total make-span by permitting passing was relatively small. For example, the expected reduction in the minimum total make-span of no passing, by permitting passing for 3 job, 4 machine, 4 job, 4 machine, and 5 job, 4 machine flow shop scheduling problems, was only 0.415, 0.426, and 0.519 percent, respectively, when the processing times were generated from uniform distribution. These values are comparable to the expected reduction in total make-span obtained by Krone (25), which was 0.66 percent by using the heuristic technique on the 8 job, 5 machine flow shop scheduling problems with the minimization of mean completion time as the performance measure. Although the expected reduction in the total make-span by permitting passing is small, it increases by 21.5 percent with every increase of one job in the flow shop.

In the last twenty years of work, almost all researchers have used processing times either chosen arbitrarily or generated from the uniform probability distribution for their research on flow shop scheduling problems. In this research, besides a uniform probability distribution, beta probability distributions were also used for generating processing times. The beta distribution has two parameters, A and B. For this dissertation, five sets of A and B values were chosen and corresponding probability distribution functions were used to generate five sets of processing times. For each set of processing times, passing and no passing studies were made and reported in Chapter V. These results clearly indicated that as the values of A and B increased, the number of problems for which permitting passing reduced the minimum total make-span of no passing, decreased significantly for each of the three different sizes of flow shops. The expected reduction in the minimum total make-span of no passing by permitting passing decreased even faster as the A and B values increased. For example, for 5 job, 4 machine scheduling problems, the expected value of the reduction in the minimum total make-span of no passing was 0.850 percent when A and B were equal to 0.5, but was only 0.062 percent when A and B were each 3.0.

During this investigation it was found that under no passing the range between the maximum and minimum total make-spans was quite narrow. When the processing times were generated from the uniform probability distribution, on an average the range was only 24, 31, and 37 percent of the minimum total make-span for 3 job, 4 job, and 5 job, 4 machine flow shop scheduling problems. Also, as values of A and B increased in the beta probability distribution, values of the range decreased significantly. For example, when A and B were equal to 3.0, the ranges were only 16, 20, and 24 percent of the minimum total make-span for 3 job, 4 job, and 5 job, 4 machine flow shop scheduling problems.

Many researchers have presented heuristics for flow shop scheduling problems under no passing. Few of them have led us to believe that their heuristics are totally reliable, because they provide solutions whose total make-spans are "only" 5 to 15 percent away from the minimum total make-span. In the absence of any knowledge about the range between the maximum total make-span (worst solution) and the minimum total make-span (best solution), their claim might seem justifiable, but if the value of the range is known and if it is narrow compared to the minimum total make-span, then the claim of "good heuristics" can be discounted.

## Suggestions for Further Research

The most important and useful research in the area of flow shop scheduling at this time would be the development of the branch and bound technique to obtain global optimum solutions under passing. Since the branch and bound technique for no passing is available, the task of modifying it for passing seems possible. After spending a great deal of time and effort working with passing in flow shop scheduling problems, the author of this dissertation was convinced that the task of modifying the branch and bound techniques is feasible but will require a lot of effort and analysis by other researchers.

Other research that could be quite useful is an extension of the studies and analysis done herein to flow shop scheduling problems with more than four machines. With N job, 4 machine flow shop scheduling problems and minimization of the total make-span as the performance criterion, passing need be considered only between job sequences of the second and third machines. But with M machines in the flow shop where M is more than four, M-3 instances of passing should be considered to obtain a global optimum solution. Various plans (heuristics) presented here could be modified through further experimentation. The biggest disadvantage of the proposed extension is that for any job larger than a 5 job, 4 machine flow shop scheduling problem, tremendously large computational efforts would be required. When a complete enumeration of six thousand 5 job, 4 machine flow shop scheduling problems with passing permitted was done for this dissertation, more than 72,000 seconds of computational time were needed on an IBM 360/75 computer. By the same token, one problem of 5 job, 5 machine flow shop size with passing permitted would require 1,440 seconds of computer time to do a complete enumeration. Unless a much larger computer with plenty of computation time is available, a study involving complete enumeration to obtain globally optimum solutions is not feasible.

Examination of the applicability of the various plans presented in this dissertation for N job, 4 machine flow shop scheduling problems where N > 5 could also be an interesting idea for further research. These plans were quite successful for small flow shop scheduling problems. Whether they are equally useful for larger ($N > 5$) flow shop scheduling problems or whether some modifications are required could be explored in future research.

Throughout the study and analysis of permitting passing in flow shop scheduling problems, it was assumed that all the optimum solutions a problem has under no passing can be easily obtained; and in this study, were obtained. It would be interesting to develop heuristics and analyze their performance when only one or a few optimum solutions per problem were obtained. Techniques required for obtaining only one optimum solution under no passing per problem will demand less computational effort than techniques providing all the optimum solutions a problem has under no passing. It would be beneficial to determine whether the initial savings in computational effort by obtaining only one optimum solution under no passing per problem is worthwhile.

The list of possible topics for further research is almost endless. Besides the above mentioned topics, there are many unanswered questions in the flow shop scheduling field under no passing which could be researched. It is the hope of the author that this dissertation will generate more interest in the flow shop scheduling area and that more people will take it upon themselves to find answers to the remaining questions in this field.

# BIBLIOGRAPHY

## BIBLIOGRAPHY


1. Akers, S. B. and Friedman, J. "A Non-Numerical Approach to Production Scheduling Problem." Operations Research, Vol. 3, No. 4 (1955), pp. 429-442.

2. Akers, S. B. "A Graphical Approach to Production Scheduling Problems." Operations Research, Vol. 4, No. 2 (1956), pp. 244-245.

3. Bakshi, M. S. and Arora, S. R. "The Sequencing Problem." Management Science, Vol. 16, No. 4 (1969), pp. B247-B263.

4. Bellmore, M. and Nemhauser, G. L. "The Traveling Salesman Problem: A Survey." Operations Research, Vol. 16, No. 3 (1968), pp. 538-558.

5. Bowman, Edward H. "The Schedule-Sequencing Problem." Operations Research, Vol. 7, No. 5 (1959), pp. 621-624.

6. Brown, A. P. G. and Lomnicki, Z. A. "Some Applications of the 'Branch and Bound' Algorithm to Machine Scheduling Problems." Operational Research Quarterly, Vol. 17, No. 2 (1966), pp. 173-186.

7. Campbell, H. G., Dudek, R. A. and Smith, M. L. "A Heuristic Algorithm for the N Job, M Machine Sequencing Problem." Management Science, Vol. 16, No. 10 (1970), pp. B630-B637.

8. Conway, R. W., Maxwell, W. L., and Miller, L. W. Theory of Scheduling. Reading, Mass.: Addison-Wesley, 1967.

9. Corwin, B. D. Some Flow Shop Scheduling Problems Involving Sequence Dependent Setup Times. Ph.D. Thesis, Case Western Reserve University, 1969.

10. Day, J. E. and Hottenstein, M. P. "Review of Sequencing Research." Nav. Res. Log. Quarterly, Vol. 17, No. 1 (1970), pp. 11-39.

11. Dudek, R. A. and Teuton, O. F. "Development of M-Stage Decision Rule for Scheduling N Jobs Through M Machines." Operations Research, Vol. 12, No. 3 (1964), pp. 471-497.

12. Elmaghraby, S. E. "The Machine Sequencing Problem--Review and Extensions." Naval Research Logistics Quarterly, Vol. 15, No. 2 (1968), pp. 205-232.

13. Fabrycky, W. J., Ghare, P. M., and Torgersen, P. E. Industrial Operations Research. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1972.

14. Giffler, B. and Thompson, G. L. "Algorithms for Solving Production Scheduling Problems." Operations Research, Vol. 8, No. 4 (1960), pp. 487-503.

15. Giglio, R. J. and Wagner, H. M. "Approximate Solutions to the Three-Machine Scheduling Problem." Operations Research, Vol. 12, No. 2 (1964), pp. 305-324.

16. Gupta, J. N. D. "A Functional Heuristic Algorithm for the Flowshop Scheduling Problem." Operational Research Quarterly, Vol. 22, No. 1 (1971), pp. 39-47.

17. Gupta, J. N. D. "M-Stage Scheduling Problem--a Critical Appraisal." International Journal of Production Research, Vol. 9, No. 2 (1971), pp. 267-282.

18. Gupta, J. N. D. "A General Algorithm for the NXM Flowshop Scheduling Problem." International Journal of Production Research, Vol. 7, No. 3 (1969), pp. 241-247.

19. Heller, J. "Some Numerical Experiments for an MXJ Flow Shop and Its Decision--Theoretical Aspects." Operations Research, Vol. 8, No. 2 (1960), pp. 178-184.

20. Heller, J. "Combinatorial Properties of Machine Shop Scheduling." NYO-2879, AEC Research and Development Report, 1959.

21. Heller, J. "Combinatorial, Probabilistic, and Statistical Aspects of an MXJ Scheduling Problem." NYO-2540, AEC Research and Development Report, 1959.

22. Ignall, E. and Schrage, L. "Application of the Branch-and Bound Technique to Some Flow Shop Scheduling Problems." Operations Research, Vol. 13, No. 3 (1965), pp. 400-412.

23. Johnson, S. M. "Optimal Two and Three Stage Production Schedules with Set-up Times Included." Nav. Res. Log. Quart., Vol. 1 (1954), pp. 61-68.

24. Karush, W. "A Counterexample to a Proposed Algorithm for Optimal Sequencing of Jobs." Operations Research, Vol. 13, No. 3 (1965), pp. 323-325.

25. Krone, M. J. Heuristic Programming Applied to Scheduling Problems. Ph.D. Thesis, Princeton University, 1970.

26. Krone, M. J. and Steiglitz, K. "Heuristic-Programming Solution of a Flowshop-Scheduling Problem." Operations Research, Vol. 22, No. 3 (1974), pp. 629-638.

27. Lomnicki, Z. A. "A 'Branch-and-Bound' Algorithm for the Exact Solution of the Three-Machine Scheduling Problem." Operational Research Quarterly, Vol. 16, No. 1 (1965), pp. 89-100.

28. McMohan, G. B. and Burton, P. G. "Flow Shop Scheduling with the Branch-and-Bound Method." Operations Research, Vol. 15, No. 3 (1967), pp. 473-481.

29. Muth, J. F. and Thompson, G. L. Industrial Scheduling. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.

30. Nugent, C. E. On Sampling Approaches to the Solution of the N-by-M Static Sequencing Problem. Ph.D. Thesis, Cornell University, 1964.

31. O'Brien, J. J. Scheduling Handbook. New York, New York: McGraw-Hill Book Company, 1969.

32. Page, E. S. "An Approach to Scheduling Jobs on Machines." Journal of Royal Statistical Society, Series B, Vol. 23 (1961), pp. 484-492.

33. Palmer, D. S. "Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time--A Quick Method of Obtaining a Near Optimum." Operational Research Quarterly, Vol. 23, No. 3 (1972), pp. 323-331.

34. Reddi, S. S. and Ramamoorthy, C. V. "On the Flow-Shop Sequencing Problem with No Wait in Process." Operational Research Quarterly, Vol. 23, No. 3 (1972), pp. 323-331.

35. Sisson, R. L. "Sequencing in Job Shops--A Review." Operations Research, Vol. 7 (1959), pp. 10-29.

36. Sisson, R. L. "Sequencing Theory." Chapter 7 in Progress in Operations Research. Edited by R. L. Ackoff. New York: Wiley, 1961.

37. Smith, R. D. and Dudek, R. A. "A General Algorithm for the Solution of the N Job, M Machine Sequencing Problem of the Flow Shop." Operations Research, Vol. 15, No. 1 (1967), pp. 71-82.

38. Spinner, A. H. "Sequencing Theory--Development to Date." Nav. Res. Log. Quart., Vol. 15, No. 2 (1968), pp. 319-330.

39. Starr, M. K. Systems Management of Operations. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971.

40. Story, A. E., and Wagner, H. M. "Computational Experience with Integer Programming for Job-Shop Scheduling." Chapter 14 in Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, N.J.: Prentice-Hall, 1963.

41. Szware, W. "Solution of the Akers-Friedman Scheduling Problem." Operations Research, Vol. 8, No. 6 (1960), pp. 782-788.

42. Wagner, H. M. "An Integer Programming Model for Machine Scheduling." Nav. Res. Log. Quart., Vol. 6, No. 2 (1959), pp. 131-140.

43. Wismer, D. A. "Solution of the Flowshop-Scheduling Problem with No Intermediate Queues." Operations Research, Vol. 20, No. 3 (1972), pp. 689-697.

# APPENDICES

APPENDIX A

```
      DIMENSION T(5,4),F(5,4),A(5),B(5),JY(5),IFACT(5),
     1IQ(150),IM(150),JTOT(120),KTOT(120)
      DIMENSION ISMLUI(150),NIU(150),ISMU(150),IFLAGU(150),
     1 AVICT(10),ISMI1(500),ILRG1(500),TOT1(500),IA(80)
      INTEGER A,B,T,F,TS,PLAN,TIME
      DATA IFACT/1,2,6,24,120/
      DATA LM1,LM2/001,500/
      DATA JY/6,2,1,1,1/
      NNN=4
      JRN=0
      ITOT2=0
      IF1=IFACT(NNN)
      DO 1 M=1,IF1
      JTOT(M)=0
    1 KTOT(M)=0
      DO 40 MN=LM1,LM2
      READ(5,101) ((T(I,J),J=1,4),I=1,NNN)
      ISML1(MN)=100000
      ILRG1(MN)=0
      TOT1(MN)=0
      DO 19 M=1,IF1
      CALL SCHIL(M,A,JY,NNN)
      CALL FCALC1(A,T,F,NNN)
      CALL FCALC7(A,T,F,NNN)
      TS=F(A(NNN),4)
      TOT1(MN)=TOT1(MN)+TS
      IF(TS.LE.ILRG1(MN)) GO TO 16
      ILRG1(MN)=TS
   16 IF(TS-ISML1(MN)) 17,18,19
   17 ISML1(MN)=TS
      N=0
   18 N=N+1
      IQ(N)=M
   19 CONTINUE
      TOT1(MN)=TOT1(MN)/IF1
      ITOT2=ITOT2+N
      WRITE(6,204) MN,N,ISML1(MN),TOT1(MN),ILRG1(MN)
     1,(IQ(N1),N1=1,N)
      KTOT(N)=KTOT(N)+1
      DO 20 N1=1,N
      JTOT(IQ(N1))=JTOT(IQ(N1))+1
   20 CONTINUE
   40 CONTINUE
      WRITE(6,201) ITOT2
      WRITE(6,205) (JTOT(I),I=1,IF1)
```

```
      WRITE(6,205) (KTOT(L),L=1,IF1)
101   FORMAT(4I4,2X,4I4,2X,4I4,2X,4I4,2X)
201   FORMAT(1H1,30X,I5,9X,20I4)
202   FORMAT(10I5)
203   FORMAT(1H0,I4,2I8,I7,1X,3I8,3I6)
204   FORMAT(1X,3I5,5X,F5.1,11I5)
205   FORMAT(1H0,5(24I5,//))
206   FORMAT(1H1,//)
207   FORMAT(1H0, 8X,23H--------PASSING--------, 1X
     1,24H-------NO PASSING-------)
208   FORMAT(1H0,8X,23HPROBLEM MINIMUM NUMBER ,1X
     1,24H TOTAL    NUMBER MAXIMUM ,18H--SAVING BY---     )
209   FORMAT(17X,15H TOTAL   OF OPT., 1X,11X,2HOF,16X,4HPLAN)
210   FORMAT(32H   NUMBER   NUMBER MAKESPAN SEQ.   ,1X
     1,24HMAKESPAN  SEQ.    SAVING ,14H 1      2      3)
211   FORMAT(//,6HOTOTAL,7X,I8,I7,1X,3I8,3I6)
212   FORMAT(8H0AVERAGE, 5X, F8.2,F8.2,1X,3F8.2,3F6.2)
213   FORMAT(9H0STD.DEV.,4X,2F8.2)
214   FORMAT(1H1)
215   FORMAT(1H0,I6,'-',I3,I10)
      STOP
      END
```

```
      SUBROUTINE SCHED(M,JA,JY,NNN)
      DIMENSION JA(5),JY(5),K(5),IT(5)
      DO 1 I=1,NNN
    1 IT(I)=I
      IY=M
      DO 2 I=1,NNN
      K(I)=((IY-1)/JY(I))+1
      K1=1
      DO 3 J=1,NNN
      IF(IT(J).EQ.0) GO TO 3
      IF(K(I).EQ.K1)GO TO 4
      K1=K1+1
    3 CONTINUE
    4 JA(I)=IT(J)
      IT(J)=0
      IY=IY-(K(I)-1)*JY(I)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FCALC1(A,T,F,NNN)
      DIMENSION A(5),T(5,4),F(5,4)
      INTEGER A,T,F
      F(A(1),1)=T(A(1),1)
      DO 2 I=2,NNN
    2 F(A(I),1)=F(A(I-1),1)+T(A(I),1)
      F(A(1),2)=F(A(1),1)+T(A(1),2)
      DO 3 I=2,NNN
      IF(F(A(I-1),2)-F(A(I),1))5,5,4
    4 F(A(I),2)=F(A(I-1),2)+T(A(I),2)
      GO TO 3
    5 F(A(I),2)=F(A(I),1)+T(A(I),2)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FCALC2(B,T,F,NNN)
      DIMENSION B(5),T(5,4),F(5,4)
      INTEGER B,T,F
      F(B(1),3)=F(B(1),2)+T(B(1),3)
      F(B(1),4)=F(B(1),3)+T(B(1),4)
      DO 7 J=3,4
      DO 8 I=2,NNN
      IF(F(B(I-1),J)-F(B(I),(J-1)))9,9,10
    9 F(B(I),J)=F(B(I),(J-1))+T(B(I),J)
      GO TO 8
   10 F(B(I),J)=F(B(I-1),J)+T(B(I),J)
    8 CONTINUE
    7 CONTINUE
      RETURN
      END
```

# APPENDIX B

```
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IES(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITEMP,MARK
      INTEGER PROCTI,TIME
      JOB=20
      MACHIN=10
      JONSON=156
      JAG=0
      K=0
      READ(5,200)((PROCTI(I,J),J=1,MACHIN),I=1,JOB)
      ISTOP=0
      JOBSCH(1)=0
      LIST=1
      LOWBON(1)=1
      DO 1 I=1,JOB
      MATRIX(1,I)=0
    1 CONTINUE
    2 CALL SEND
      IF(NAG.EQ.0)GO TO 3
      CALL DESTRO
      NAG=0
      JAG=0
    3 IF(JAG.LT.10)GO TO 4
      CALL DESTRO
      JAG=0
    4 JAG=JAG+1
      IF(ISTOP-1)2,6,5
    5 WRITE(6,206)
      CALL DESTRO
      CALL CHOOSE
      GO TO 10
    6 WRITE(6,201)JONSON
      DO 7 I=1,LIST
      WRITE(6,202)(MATRIX(I,J),J=1,JOB)
    7 CONTINUE
  200 FORMAT(10I2)
  201 FORMAT(10I5)
  202 FORMAT(20I3)
  203 FORMAT(5X,3I5,20I3)
  206 FORMAT(95H THERE WERE MORE THAN 500 ITEMS IN THE LIST.
     1 THE PROGRAM STOPPED AND PRINTED CARDS.)
   10 STOP
      END
```

```
      SUBROUTINE SEND
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IPS(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITLIM,MARK
      INTEGER PROCTI,TIME
      NAG=0
      ITEMP=100000
      DO 1 LM=1,LIST
      IF(JOBSCH(LM).EQ.JOB)GO TO 1
      IF(ITEMP.LT.LOWBON(LM))GO TO 1
      ITEMP=LOWBON(LM)
      LM1=LM
    1 CONTINUE
      IF(ITEMP.NE.100000)GO TO 2
      ISTOP=1
      RETURN
    2 NJOB=JOB-JOBSCH(LM1)
      MJOB= JOBSCH(LM1)
      DO 4 I=1,MACHIN
      IF(MJOB.NE.0)GO TO 3
      TIME(I)=0
      GO TO 4
    3 TIME(I)=PROCTI(MATRIX(LM1,1),I)
    4 CONTINUE
      IF (MJOB.LE.1)GO TO 8
      DO 7 I=2,MJOB
      TIME(1)=TIME(  1)+PROCTI(MATRIX(LM1,I),1)
      DO 6 J=2,MACHIN
      IF(TIME(J).GT.TIME(J-1))GO TO 5
      TIME(J)=TIME(J-1)+PROCTI(MATRIX(LM1,I),J)
      GO TO 6
    5 TIME(J)=TIME(J)    +PROCTI(MATRIX(LM1,I),J)
    6 CONTINUE
    7 CONTINUE
      IF(NJOB.GT.1)GO TO 8
      NAG=1
    8 DO 9 I=1,NJOB
      IPS(I)=MATRIX(LM1,MJOB+I)
    9 CONTINUE
      DO 10 I=1,NJOB
      IT=IPS(1)
      IPS(1)=IPS(I)
      IPS(I)=IT
      MARK=0
      CALL BOUND
```

```
      IF(MARK.EQ.0)GO TO 10
      IF(NAG.NE.0)JONSON=ITEMP
      CALL INTFO
   10 CONTINUE
      LOWBON(LM1)=100000
      RETURN
      END
```

```
      SUBROUTINE BOUND
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IPS(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITEMP,MARK
      INTEGER PROCTI,TIME
      ITEMP=0
      KTIME(1)=TIME(1)+PROCTI(IPS(1),1)
      DO 2 I=2,MACHIN
      IF(TIME(I).GT.KTIME(I-1))GO TO 1
      KTIME(I)=KTIME(I-1)+PROCTI(IPS(1),I)
      GO TO 2
    1 KTIME(I)=TIME(I)+PROCTI(IPS(1),I)
    2 CONTINUE
      DO 3 I=1,MACHIN
      LBND(I)=0
      IF(NJOB.GT.1)GO TO 6
      ILOVAL=0
      GO TO 4
    6 ILOVAL=100000
      DO 4 IKR=2,NJOB
      LBND(I)=LBND(I)+ PROCTI(IPS(IKR),I)
      IVAL=0
      I1=I+1
      IF(I1.GT.MACHIN)GO TO 5
      DO 5 IM=I1,MACHIN
      IVAL=IVAL+ PROCTI(IPS(IKR),IM)
    5 CONTINUE
      IF(ILOVAL.GT.IVAL) ILOVAL=IVAL
    4 CONTINUE
      ITEPOJ=KTIME(I)+LBND(I)+ILOVAL
      IF(ITEPOJ.GT.JONSON)GO TO 25
      IF(ITEPOJ.GT.ITEMP) ITEMP=ITEPOJ
    3 CONTINUE
      MARK=1
   25 RETURN
      END
```

```
      SUBROUTINE INTRO
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IRS(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITEMP,MARK
      INTEGER PROCTI,TIME
      LIST=LIST+1
      LOWBON(LIST)=ITEMP
      DO 2 I=1,JOB
      IF(I.GT.MJOB)GO TO 1
      MATRIX(LIST,I)=MATRIX(LM1,I)
      GO TO 2
    1 MATRIX(LIST,I)=IRS(I-MJOB)
    2 CONTINUE
      JOBSCH(LIST)=MJOB+1
      IF(LIST.LE.700)GO TO 4
      ISTOP=2
    4 RETURN
      END
```

```
      SUBROUTINE DESTRO
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IRS(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITEMP,MARK
      INTEGER PROCTI,TIME
      IREMOV=0
      DO 3 I=1,LIST
      IF(JONSON.GT.LOWBON(I))GO TO 1
      IREMOV=IREMOV+1
      GO TO 3
    1 IF(IREMOV.EQ.0)GO TO 3
      LOWBON(I-IREMOV)=LOWBON(I)
      JOBSCH(I-IREMOV)=JOBSCH(I)
      DO 2 J=1,JOB
      MATRIX(I-IREMOV,J)=MATRIX(I,J)
    2 CONTINUE
    3 CONTINUE
      LIST=LIST-IREMOV
      IF(LIST.GT.0)GO TO 4
      ISTOP=1
    4 RETURN
      END
```

```
      SUBROUTINE CHOOSE
      COMMON JOBSCH(720),MATRIX(720,20),LOWBON(720),PROCTI(
     120,10),IPS(20),LBND(20),KTIME(10),TIME(10),LIST,LM1,NJOB
     2,MJOB,JONSON,ISTOP,NAG,JOB,MACHIN,ITEMP,MARK
      INTEGER PROCTI,TIME
      M=0
    1 ITEMP=100000
      DO 4 I=1,LIST
      IF(ITEMP-LOWBON(I)) 4,3,2
    2 ITEMP=LOWBON(I)
      K=0
    3 K=K+1
    4 CONTINUE
      DO 5 I=1,LIST
      IF(ITEMP.NE.LOWBON(I)) GO TO 5
      WRITE(6,201)I,LOWBON(I),JOBSCH(I),(MATRIX(I,J),J=1,JOB)
      WRITE(7,201)I,LOWBON(I),JOBSCH(I),(MATRIX(I,J),J=1,JOB)
      LOWBON(I)=100000
    5 CONTINUE
      M=M+K
      IF(M.LT.LIST)GO TO 1
  201 FORMAT(4I5,20I3)
      RETURN
      END
```

# APPENDIX C

```
      DIMENSION T(5,4),F(5,4),A(5),B(5),JY(5),IFACT(5),
     1IQ(150),IM(150),JTOT(120),KTOT(120)
      DIMENSION PLAN(4,3),TIME(5,16),IDIFF(500),IB(80)
      DIMENSION JOB(6),JOBTT(150,4),MI2U(150),MYTOT(10),KR(80)
      DIMENSION ISMLUI(150),NIU(150),ISMU(150),IFLAGU(150),
     1 AVTOT(10),ISML1(500),ILRG1(500),TOT1(500),IA(80)
      INTEGER A,B,T,F,TS,PLAN,TIME
      DATA IFACT/1,2,6,24,120/
      DATA LM1,LM2/001,500/
      DATA JY/6,2,1,1,1/
      NNN=4
      JRN=0
      ITOT2=0
      IF1=IFACT(NNN)
      DO 1 M=1,IF1
      JTOT(M)=0
    1 KTOT(M)=0
      DO  2 I=1,10
      MYTOT(I)=0
    2 CONTINUE
      DO 4 I=1,4
      DO 3 J=1,3
      PLAN(I,J)=0
    3 CONTINUE
    4 CONTINUE
      DO 40 MN=LM1,LM2
      READ(5,101)((T(I,J),J=1,4),I=1,NNN)
      ISML1(MN)=100000
      ILRG1(MN)=0
      TOT1(MN)=0
      DO 19 M=1,IF1
      CALL SCHED(M,A,JY,NNN)
      CALL FCALC1(A,T,F,NNN)
      CALL FCALC2(A,T,F,NNN)
      TS=F(A(NNN),4)
      TOT1(MN)=TOT1(MN)+TS
      IF(TS.LE.ILRG1(MN)) GO TO 16
      ILRG1(MN)=TS
   16 IF(TS-ISML1(MN)) 17,18,19
   17 ISML1(MN)=TS
      N=0
   18 N=N+1
      IQ(N)=M
   19 CONTINUE
      TOT1(MN)=TOT1(MN)/IF1
```

```
      ITOT2=ITOT2+N
      WRITE(6,204) MN,N,ISML1(MN),TOT1(MN),ILPS1(MN)
     1,(IQ(N1),N1=1,N)
      KTOT(N)=KTOT(N)+1
      DO 20 N1=1,N
      JTOT(IQ(N1))=JTOT(IQ(N1))+1
   20 CONTINUE
      DO 21 J=1,6
      JOE(J)=ISML1(MN)
   21 CONTINUE
      MARK=0
      N1=N
      DO 28 M=1,IF1
      CALL SCHED(M,A,JY,NNN)
      CALL FCAIC1(A,T,F,NNN)
      DO 27 N=1,IF1
      IF(M.EQ.N) GO TO 27
      CALL SCHED(N,B,JY,NNN)
      CALL FCAIC2(B,T,F,NNN)
      ISM=F(F(NNN),4)
      IF(ISM.GE.ISML1(MN)) GO TO 27
      CALL SHIFT(A,B,NSHIFT,NNN)
      WRITE(6,203) ISM,M,N,NSHIFT
      MARK=MARK+1
      IF(JOE(1).LE.ISM) GO TO 22
      JOE(1)=ISM
   22 IF(NSHIFT.NE.1) GO TO 23
      IF(JOE(2).LE.ISM)GO TO 23
      JOE(2)=ISM
   23 DO 26 NS=1,N1
      IF(IQ(NS).EQ.M) GO TO 24
      IF(IQ(NS).EQ.N)GO TO 24
      GO TO 26
   24 IF(JOE(3).LE.ISM)GO TO 25
      JOE(3)=ISM
   25 IF(NSHIFT.NE.1)GO TO 26
      IF(JOE(4).LE.ISM) GO TO 26
      JOE(4)=ISM
   26 CONTINUE
   27 CONTINUE
   28 CONTINUE
      IF(MARK.EQ.0)GO TO 40
      JRN=JRN+1
      MYTOT(1)=MYTOT(1)+ISML1(MN)
      MYTOT(2)=MYTOT(2)+N1
```

```
      MYTOT(3)=MYTCT(3)+JCE(1)
      MYTOT(4)=MYTCT(4)+MARK
      DO 35 J=1,4
      JOETT(JRN,J)=ISML1(MN)-JOE(J)
      MYTOT(4+J )=MYTOT(4+J )+JCETT(JRN,J )
   35 CONTINUE
      DO 36 K=2,4
      JLM=1
      IF(JCETT(JRN,K).LT.JCETT(JRN,1))JLM=2
      IF(JCETT(JRN,K).EQ.0)JLM=3
      PLAN(K-1,JLM)=PLAN(K-1,JLM)+1
   36 CONTINUE
      MI2U(JRN)=MN
      ISMLUI(JRN)=ISML1(MN)
      NIU(JRN)=N1
      ISMU(JRN)=JCE(1)
      IFLAGU(JRN)=MARK
   40 CONTINUE
   41 CONTINUE
      DO 62 IJKL=1,6
      DO 51 JRU=1,JRN
      JJR=JRU/25
      JJR=JJR*25+1
      IF(JJR.NE.JRU)GO TO 51
      WRITE(6,206)
      WRITE(6,207)
      WRITE(6,208)
      WRITE(6,209)
      WRITE(6,210)
   51 WRITE(6,203)JRU,MI2U(JRU),ISMLUI(JRU),NIU(JRU),ISMU(JRU),
     1IFLAGU(JRU),(JCETT(JRU,JI),JI=1,4)
      DO 52 I=1,8
      AVTOT(I)=FLOAT(MYTOT(I))/FLOAT(JRN)
   52 CONTINUE
      WRITE(6,211)(MYTOT(I),I=1,8 )
      WRITE(6,212)(AVTOT(I),I=1,8)
      N=LM2-LM1+1
      ISTART=161
      INTER=20
      CALL     FREQ(ISML1,LM1,N,ISTART,INTER,KR)
      IA(1)=ISTART
      IB(1)=IA(1)-1+INTER
      WRITE(6,214)
      DO 55 I=1,19
      WRITE(6,215)IA(I),IB(I),KR(I)
```

```
      IA(I+1)=IA(I)+INTER
      IB(I+1)=IB(I)+INTER
   55 CONTINUE
      CALL     XMSQ(ISML1,LM1,N,XMEAN,XSQ)
      WRITE(6,212) XMEAN
      WRITE(6,213) XSQ
      ISTART=241
      INTER=20
      CALL     FREQ(ILRG1,LM1,N,ISTART,INTER,KR)
      IA(1)=ISTART
      IB(1)=IA(1)-1+INTER
      WRITE(6,214)
      DO 56 I=1,18
      WRITE(6,215) IA(I),IB(I),KR(I)
      IA(I+1)=IA(I)+INTER
      IB(I+1)=IB(I)+INTER
   56 CONTINUE
      CALL     XMSQ(ILRG1,LM1,N,XMEAN,XSQ)
      WRITE(6,212) XMEAN
      WRITE(6,213) XSQ
      ISTART=1
      INTER=10
      DO 58 I=LM1,LM2
       IDIFF(I)= ILRG1(I)-ISML1(I)
   58 CONTINUE
      CALL     FREQ(IDIFF,LM1,N,ISTART,INTER,KR)
      IA(1)=ISTART
      IB(1)=IA(1)-1+INTER
      WRITE(6,214)
      DO 60 I=1,22
      WRITE(6,215) IA(I),IB(I),KR(I)
      IA(I+1)=IA(I)+INTER
      IB(I+1)=IB(I)+INTER
   60 CONTINUE
      CALL     XMSQ(IDIFF,LM1,N,XMEAN,XSQ)
      WRITE(6,212) XMEAN
      WRITE(6,213) XSQ
   62 CONTINUE
      WRITE(6,201) ITCT2
      WRITE(6,205) (JTCT(I),I=1,IF1)
      WRITE(6,205) (KTCT(L),L=1,IF1)
      DO 65 I=1,3
      WRITE(6,203) (PLAN(I,J),J=1,3)
   65 CONTINUE
  101 FORMAT(4I4,2X,4I4,2X,4I4,2X,4I4,2X)
```

```
201 FORMAT(1H1,30X,I5,9X,20I4)
202 FORMAT(10I5)
203 FORMAT(1H0,I4,2I8,I7,1X,3I8,3I6)
204 FORMAT(1X,3I5,5X,F5.1,11I5)
205 FORMAT(1H0,5(24I5,//))
206 FORMAT(1H1,//)
207 FORMAT(1H0, 8X,23H--------PASSING--------, 1X
   1,24H-------NC PASSING-------)
208 FORMAT(1H0,8X,23HPROBLEM MINIMUM NUMBER ,1X
   1,24H TOTAL   NUMBER MAXIMUM ,18H--SAVING BY---     )
209 FORMAT(17X,15H TOTAL  OF OPT.,1X,11X,2HOF,16X,4HPLAN)
210 FORMAT(32H  NUMBER  NUMBER MAKESPAN SEQ.   ,1X
   1,24HMAKESPAN  SEQ.    SAVING ,14H 1     2      3)
211 FORMAT(//,6HOTOTAL,7X,I8,I7,1X,3I8,3I6)
212 FORMAT(8H0AVERAGE, 5X, F8.2,F8.2,1X,3F8.2,3F6.2)
213 FORMAT(9H0STD.DEV.,4X,2F8.2)
214 FORMAT(1H1)
215 FORMAT(1H0,I6,'-',I3,I10)
    STOP
    END
```

```
      SUBROUTINE SCHED(M,JA,JY,NNN)
      DIMENSION JA(5),JY(5),K(5),IT(5)
      DO 1 I=1,NNN
    1 IT(I)=I
      IY=M
      DO 2 I=1,NNN
      K(I)=((IY-1)/JY(I))+1
      K1=1
      DO 3 J=1,NNN
      IF(IT(J).EQ.0) GO TO 3
      IF(K(I).EQ.K1)GO TO 4
      K1=K1+1
    3 CONTINUE
    4 JA(I)=IT(J)
      IT(J)=0
      IY=IY-(K(I)-1)*JY(I)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHIFT(A,B,NSHIFT,NNN)
      DIMENSION A(5),B(5),A1(5)
      INTEGER A,B,A1
      NSHIFT=0
      LM1=1
      DO 1 LM=1,NNN
    2 IF(A(LM).EQ.B(LM1)) GO TO 5
      IF(NSHIFT.EQ.0)GO TO 6
      DO 3 NN=1,NSHIFT
      IF(A1(NN).EQ.B(LM1)) GO TO 4
    3 CONTINUE
    6 NSHIFT=NSHIFT+1
      A1(NSHIFT)=A(LM)
      GO TO 1
    4 LM1=LM1+1
      GO TO 2
    5 LM1=LM1+1
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FCALC1(A,T,F,NNN)
      DIMENSION A(5),T(5,4),F(5,4)
      INTEGER A,T,F
      F(A(1),1)=T(A(1),1)
      DO 2 I=2,NNN
    2 F(A(I),1)=F(A(I-1),1)+T(A(I),1)
      F(A(1),2)=F(A(1),1)+T(A(1),2)
      DO 3 I=2,NNN
      IF(F(A(I-1),2)-F(A(I),1))5,5,4
    4 F(A(I),2)=F(A(I-1),2)+T(A(I),2)
      GO TO 3
    5 F(A(I),2)=F(A(I),1)+T(A(I),2)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FCALCZ(B,T,F,NNN)
      DIMENSION B(5),T(5,4),F(5,4)
      INTEGER B,T,F
      F(B(1),3)=F(B(1),2)+T(B(1),3)
      F(B(1),4)=F(B(1),3)+T(B(1),4)
      DO 7 J=3,4
      DO 8 I=2,NNN
      IF(F(B(I-1),J)-F(B(I),(J-1)))9,9,10
    9 F(B(I),J)=F(B(I),(J-1))+T(B(I),J)
      GO TO 8
   10 F(B(I),J)=F(B(I-1),J)+T(B(I),J)
    8 CONTINUE
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XMSQ(IM,IM1,N,XMEAN,XSQ)
      DIMENSION IM(500)
      XM=0.0
      DO 1 I=1,N
      XM=XM+IM(I+LM1-1)
    1 CONTINUE
      XMEAN=XM/N
      XM=0.0
      DO 2 I=1,N
      XM=XM+(IM(I+LM1-1)-XMEAN)**2
    2 CONTINUE
      XSQ=(XM/N)**0.5
      RETURN
      END
```

# APPENDIX D

```
      DIMENSION IP(16)
      IX=111111
      A1=0.5
      B1=0.5
      D1=5.0
      IA=A1*2
      IB=B1*2
      DO 1 I=1,1800
      DO 2 K=1,16
    3 CALL RANDU(IX,IY,X1)
      IX=IY
      CALL RANDU(IX,IY,X2)
      IX=IY
      CALL BDTR(X1,A1,B1,P,D,IER)
      DD1=D/D1
      IF(X2.GT.DD1) GO TO 3
      IP(K)=X1*1000.+0.5
    2 CONTINUE
      WRITE(6,201) (IP(K),K=1,16),IA,IB,I
      WRITE(7,201) (IP(K),K=1,16),IA,IB,I
    1 CONTINUE
  201 FORMAT (4I4,2X,4I4,2X,4I4,2X,4I4,3X,2I1,I5)
      STOP
      END
```