COMPARISON OF FLOWSHOP SCHEDULING ALGORITHMS

by 45

LAXMAN SINGH BENIWAL

B.E.(Mech.) Birla Institute of Technology & Science
Pilani, Rajasthan, India, 1965

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1969

Approved by:

Said Ashour

TABLE OF CONTENTS

ACKNOWLEDGMENT

I wish to extend my most profound gratitude to Dr. Said Ashour for his acting as my supervisor and giving his valuable guidance and inspiration throughout the period of this report. For this privilege and benevolence, not only, I remain indebted, but, I sincerely and fervently wish to place on record that but for his bestowing these favours, this little presentation could not have been possible.

My thanks are also due to Dr. Frank A. Tillman, Head Department of Industrial Engineering and Professor J. J. Smaltz for their kind patronage.

I also wish to extend my thanks, to the Director, Computing Center for providing me all the facilities in completing my work.

I greatly acknowledge and thank Mrs. Sharon Morrisey for her valuable assistance in typing.

My thanks are also due to Mr. Prem Chand and Mr. Promod Mehrotra for their assistance in the proof reading.

I shall be failing in my duty if I don't thank Dr. L. E. Grosh, who very kindly agreed to be a member of my committee at very short notice.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Scheduling is one of the critical problems in management. The scheduling problem may be classified into three categories: Production scheduling, Project scheduling and Job scheduling. This report is concerned with the job scheduling problem.

The job scheduling problem consists of processing J jobs on M machines such that a certain criterion is optimized. Some of the criteria are: (1) minimization of the total time required to process all jobs; (2) minimization of the total idle time on all the machines; (3) minimization of the time required for each machine, starting from the first job to the last job; and (4) maximization of the profit by meeting of due dates. However, the criterion considered in this report is that of minimizing the schedule time.

The job scheduling problem is of interest because of its diversity, complexity and magnitude. For example, consider a problem of six jobs to be processed on each of three different machines. The possible number of sequences is $(J!)^M$ or $(6!)^3 = 373{,}248{,}000$. A complete enumeration of these sequences would require years on a high speed computer. Many of these sequences are technologically non-feasible. An exhaustive enumeration must consider all sequences to eliminate the non-feasible and then select the optimal sequence.

1.1. Problem Formulation:*

Consider a job scheduling problem which consists of a certain number of jobs to be processed on various machines in a specified machine ordering. The optimal sequence of the jobs is to be determined so as to minimize the schedule time - the total time of processing all jobs.

The job scheduling problems are generally classified into: (1) Flowshop - where all jobs have the same machine ordering, and (2) jobshop - where each job has a different machine ordering.

The notations which have been used to formulate the problem mathematically are given below:

J	total number of jobs
M	total number of machines
j	job designation, $j = 1, 2, \ldots, J$
m	machine designation, $m = 1, 2, \ldots, M$
jm	operation designation
$j_x m$	sequence of jobs through machine m, $x = 1, 2, \ldots, J$
jm_y	order of machines for job j. $y - 1, 2, \ldots, M$
$j_x m_y$	a specific operation
t_{jm}	processing time of job j on machine m
T^*	processing time matrix of the original problem.
M_j^*	machine ordering vector for job j
M^*	machine ordering matrix of the original problem
S_m^*	job sequencing vector through machine m
S^*	job sequencing matrix of the original problem
T	schedule time

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

The numbering of jobs and machines is preconcieved and not necessarily correspond to the sequence in which jobs are processed on each machine or to the order in which the machines process each job. Thus, the sequence of the jobs will be designated as $j_1, j_2, \ldots, j_x, \ldots, j_J$ and the order of the machines as $m_1, m_2, \ldots, m_y, \ldots, M_M$. The term j_x indicates that job j in position x; and the term m_y means that machine m in position y.

The machine ordering for each job is designated as follows:

$$M_j^* = \{ jm_y \mid y \varepsilon M \quad \text{and} \quad (jm_y) < (jm_{y+1}) \} , \; j = 1, 2, \ldots, J$$

where,

$(jm_y) < (jm_{y+1})$ indicates that the operation of job j on machine m_y directly preceeds the operation of the same job on machine m_{y+1}. The above relation may be written as follows:

$$M_j^* = \{ jm_1 \; jm_2 \; \ldots \; jm_x \; \ldots \; jm_M \} , \quad j = 1, 2, \ldots, J.$$

The machine ordering vector for each job can be combined in a (J x M) matrix called the machine ordering matrix denoted by M*.

$$M^* = \begin{bmatrix} 1m_1 & 1m_2 & \ldots & 1m_y & \ldots & 1m_M \\ 2m_1 & 2m_2 & \ldots & 2m_y & \ldots & 2m_M \\ \vdots & \vdots & & \vdots & & \vdots \\ Jm_1 & Jm_2 & \ldots & Jm_y & \ldots & Jm_M \end{bmatrix}$$

The processing time of each job on each machine is designated by T*.

$$T^* = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1m} & \cdots & t_{1M} \\ t_{21} & 2_{22} & \cdots & t_{2m} & \cdots & t_{2M} \\ \vdots & \vdots & & \vdots & & \vdots \\ t_{j1} & t_{j2} & \cdots & t_{jm} & \cdots & t_{jM} \\ \vdots & \vdots & & \vdots & & \vdots \\ t_{J1} & t_{J2} & \cdots & t_{Jm} & \cdots & t_{JM} \end{bmatrix}$$

It should be pointed out that the processing time matrix, T*, does not imply any ordering of the operations. If a job is not to be processed on a particular machine, a zero processing time is placed in the processing time matrix.

The sequencing of jobs on each machine are designated by S_m*.

$$S_m{}^* = \{\, j_x m \mid x \varepsilon J \quad \text{and} \quad (j_x m) < (j_{x+1} m) \,\}\ , \qquad m = 1, 2, \ldots, M$$

where

$(j_x m) < (j_{x+1})m$ indicates that job j_x directly preceeds job j_{x+1} on machine m.

The above relation may be rewritten as follows:

$$S_m^* = \{ j_1m \; j_2m \; . \; . \; . \; j_xm \; . \; . \; . \; j_Jm \} , \quad m = 1, 2, . \; . \; ., M.$$

The above job sequencing vectors can be combined in a (M x J) matrix, known as the job sequencing matrix and denoted by S*.

$$S^* = \begin{bmatrix} j_1 1 & j_2 1 & \cdots & j_x 1 & \cdots & j_J 1 \\ j_1 2 & j_2 2 & \cdots & j_x 2 & \cdots & j_J 2 \\ \vdots & \vdots & & \vdots & & \vdots \\ j_1 m & j_2 m & \cdots & j_x m & \cdots & j_J m \\ \vdots & \vdots & & \vdots & & \vdots \\ j_1 M & j_2 M & \cdots & j_x M & \cdots & j_J M \end{bmatrix}$$

In summary, the job scheduling problem may be stated such that: given the machine ordering matrix M* and the processing time matrix T*, find the optimal job sequencing matrix S* which minimizes the schedule time T.

In the machine ordering matrix M*, the element jm_1, for example, indicates that job j must be processed on machine m first. The element jm_2 means that job j must be performed on machine m second, which is not the same as that in element jm_1. Also in the job sequencing matrix S, the element j_1m, for example, indicates that machine m processes job j first. The element j_2m means that machine m performs job j second, which is not the same job as that in the element j_1m. As mentioned earlier, the subscript here indicates the position of job j in the job sequencing or position of machine m in the machine ordering.

An example may clarify the above formulation. Consider a jobshop problem of four jobs and two machines. The machine ordering matrix is given below:

$$M^* = \begin{bmatrix} 1m_1 & 1m_2 \\ 2m_1 & 2m_2 \\ 3m_1 & 3m_2 \\ 4m_1 & 4m_2 \end{bmatrix} = \begin{bmatrix} 11 & 12 \\ 22 & 21 \\ 31 & 32 \\ 42 & 41 \end{bmatrix}$$

This indicates that jobs 1 and 3 are to be processed on machine 1 first and machine 2 last; however, jobs 2 and 4 are to be performed on machine 2 first and machine 1 last.

One of the feasible sequences which must be consistent with the above machine ordering matrix is shown below

$$S_o = \begin{bmatrix} j_1 1 & j_2 1 & j_3 1 & j_4 1 \\ j_1 2 & j_2 2 & j_3 2 & j_4 2 \end{bmatrix} = \begin{bmatrix} 11 & 31 & 21 & 41 \\ 12 & 22 & 32 & 42 \end{bmatrix}$$

This particular job sequencing matrix indicates that machine 1 processes the jobs in the sequence { 1 3 2 4 } and machine 2 performs the jobs in the sequence { 1 2 3 4 } .

A simple formulation is given below to the flowshop problem where all jobs have the same machine ordering.

Given the machine ordering matrix

$$M^* = [jm_1\ jm_2\ .\ .\ .\ jm_x\ .\ .\ .\ jm_M], \quad j = 1, 2, .\ .\ .\ ., J,$$

and the processing time matrix

$$T^* = [t_{j1}\ t_{j2}\ .\ .\ .\ t_{jm}\ .\ .\ .\ t_{jM}], \quad j = 1, 2, .\ .\ .\ ., J,$$

find the optimal sequence, represented by the job sequencing matrix

$$S_m^* = [j_1m\ j_2m\ .\ .\ .\ j_xm\ .\ .\ .\ j_Jm], \quad m = 1, 2, .\ .\ .\ ., M,$$

which gives the minimum schedule time T.

In the machine ordering matrix, M*, the element jm_1, for example, indicates that job j must be processed on machine m first. The element jm_2 means that job j must be performed on machine m second, which is not the same machine as that in the element jm_1. Also in the job sequencing matrix, S_m^* the element j_1m, for example, indicates that machine m processes job j first. The element j_2m means that machine m performs job j second, which is not the same job as that in the element j_1m.

The job scheduling problem as formulated above is restricted to the following assumptions:

1. Assumptions regarding jobs:

   1.1 A job may not be processed by more than one machine at a time.

   1.2 Each job must follow a specified machine ordering.

   1.3 A job is processed as soon as possible, subject to the machine ordering.

   1.4 All jobs are equally important; i.e., no priorities, due dates, or rush orders.

2. Assumptions regarding machines:

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

   2.2 Once started, each operation must be completed.

   2.3 There is only one machine from each type.

   2.4 No job is processed more than once by any machine.

3. Assumptions regarding processing times:

   3.1 The processing time of each job on each machine does not depend on the sequence in which the jobs are processed.

   3.2 The processing time of each job on each machine is determinate and integer.

   3.3 Transportation times between machines and set-up times, if any, are included in the processing times.

1.2. Literature Review:

Considerable research has been done in the field of job scheduling problem. However, an optimal solution for practical problems have not yet been obtained except for small size problems. The approaches available to solve the job scheduling problem are: (1) Combinatorial Analysis, (2) Integar-Linear Programming, (3) Graphical, (4) Graphical-Dynamic Programming, (5) Schedule Algebras, (6) Simulation.

Some of the researchers have concentrated on generating the feasible sequences only, for the job scheduling problem. A Boolean Algebra approach has been developed by Akers and Friedman (4) which eliminate initially a great number of sequences. Certain decision rules have been developed for minimizing the schedule time. An illustration has been made by solving two-job and M-machine problem.

Giffler and Thompson (55) have developed an algorithm in which they suggested to search for optimal sequences over only feasible schedules referred to as active feasible schedules. They have defined an active feasible schedule as a feasible schedule having the property that no operation can be made to start sooner by permissible left shifting. In their paper, it has been proved that the subset of active feasible schedule contains a subset of optimal schedules; and every optimal schedule is equivalent to an active optimal solution. Equivalence means that one schedule can be obtained from the other by a sequence of permissible shifts. The main reasons for concentrating on the active schedules are: (1) they contain a subset of optimal sequences; (2) they are superior to the inactive schedules; and (3) their number is usually very small compared to the number of all schedules. For smaller problems, it is possible to generate the complete set of active schedules and then pick-up the optimal. However, in large problems, it is impractical to generate all active schedules. Therefore, a random sample is generated from the set of all active schedules and the optimal schedule(s) may be obtained with some probability close to one as desired.

Using the linear graphs, Heller and Logemann (68) have developed an algorithm to generate feasible schedules and compute the corresponding schedule time. An operation of processing job j on machine m for the return i is referred to as a node (mji). These appropriate nodes are linked as per the machine ordering. One of the available nodes is chosen and scheduled. This process is continued until all operations are scheduled. The result is one feasible schedule and the corresponding schedule time.

Furthermore, there exists several scheduling algorithms for obtaining a solution to the scheduling problem.

For J-jobs and two-machine flowshop problem, Johnson (80) has developed a simple algorithm to determine an optimal schedule for minimizing the schedule time. He also extended his algorithm to cover a special case of the three-machine problems. For jobshop problems, however, Jackson (71) extended Johnson's results.

Dudek and Teuton (38) have extended Johnson's algorithm to solve the flowshop problems of J-job and M-machine. The algorithm involves the minimization of the cumulative idle time on the last machine. Although they have claimed that optimality is guaranteed, Karush (82) has formulated a counter example of three-job and 3-machine. Smith and Dudek (156) have revised the above algorithm to obtain the optimal solution.

Another approach which gives optimal or near optimal solution after the generation of only a small subset of the possible sequences. This is referred to as branch-and-bound technique which has been developed by Little et. al. (92) to solve the travelling salesman problem. Ignall and Scharge (70) have applied the above technique to the two-and three-machine flowshop problem. Their computational experience involves up to nine jobs. Brown and Lomnicki (27) have generalized the branch-and-bound algorithm developed by Lomnicki (93) for three machines to arbitrary number of machines. McMahon and Burton (99) have applied branch-and-bound technique for the three-machine problem, giving a new method of obtaining the bound and utilized the fact that scheduling problems are symmetrical and with respect to time-reversal. Their computational experience on CDC 3600 computer involves up to 45 jobs and three-machines. They concluded that the use of the composite bound (machine based bound and job based bound) decision rule is more efficient.

Brooks and White (26) have modified Giffler and Thompson's (54) algorithm using the lower bound as a decision rule for developing an optimal or near optimal solution. Since practical problems are much larger than could be solved economically by this procedure, they used lower bound as a decision rule for developing a near optimal solution.

Integar-Linear programming approach has been utilized for job scheduling problem. Up to this time, three different formulations developed by Bowman (25), Wagner (165) and Manne (95) are available. However, the computations are not feasible. Dantzig (35, 36) formulated the job scheduling problem as an ordinary linear programming problem by neglecting the integar constraints. The drawback of this method is that it may give fractional optimal solution. To overcome this problem, Giglio and Wagner (57) have solved 100 flowshop problems of six-job and three-machine using an ordinary linear programming and rounding off the final solution. However, computations were not encouraging.

In his investigation, Heller (53) has shown that the limit distribution of the schedule times is asymptotically normal as the number of jobs increases.

Ashour (6) has developed a decomposition approach to the job scheduling problem. In this approach, the problem is decomposed to a smaller, more manageable problems which requires less computational effort. In his investigation, Ashour has found that the schedule time distribution obtained by decomposition shifts toward the minimum value. Furthermore, the shift increases as the number of jobs in each subgroup increases. His computational experiment consists of six to 40 jobs and three to ten machines.

Hardgrave and Nemhauser (62) and Akers (3) have developed a graphical approach for two-job and M-machine problem. Graphical-Dynamic programming approach has been presented by Held and Karp (64) for J-job and one-machine problem. Szwarc (160) has given a solution for a problem of two-job and M-machine by a combination of dynamic programming and graphical methods. Szwarc has also developed a technique for the job-shop problem, but it does not guarantee optimality.

In practice, the job scheduling problem is dynamic in nature. The break-down of machines, efficiency of the labor, quality of the products make the scheduling problems quite complicated. For such complex problems, a computer simulation is used. Eilon and Hodgson (40) have developed a simulation model for jobshop scheduling problems consisting of two identical machines operating in parallel. Conway et. al. (33) have simulated jobshop problem of five-machine and 100 arriving jobs to test the priority rules. Gere (47) has studied the performance of a number of combinations of priority rules and heuristics for several criteria such as due dates, minimization of the sum of latness.

1.3. Proposed Research:

Several techniques have been suggested for the solution of job scheduling problems. In general, these techniques are practical for small size problems. A little progress has been made toward the development of efficient algorithms.

Among the promising techniques, which are used to solve the flowshop problem are: (1) Direct technique reported by Smith and Dudek (156); (2) Branch-and-bound technique developed by Brown and Lomnicki (27); and (3) lower bound modified by Brooks and White (26). In reviewing the literature, it appears that no comparison among these procedures has been made. The purpose of this report is to investigate the solution obtained by different procedures. Thus the techniques are compared considering basically the following: (1) the relative efficiency of the different solutions to the optimal value; (2) the statistical characteristics of the distinct schedule times; and (3) the computational efficiency. To obtain adequate comparison among these techniques and to minimize the variations, considerable experiments were conducted. In this report, the solution by Branch-and-bound and Direct algorithms are compared to the complete enumeration.

In the next chapter, the scheduling techniques are discussed and illustrated by a sample problem. Chapter III is devoted to the experimental investigation designed to compare the performance of the scheduling techniques. The results of these computational experiments and the conclusions are reported in Chapter III and IV.

CHAPTER II

COMPUTATIONAL SCHEDULING ALGORITHMS

This chapter includes clear presentation of various algorithms for the solution of the flow shop scheduling problem, such as (1) Linear graph algorithm by Ashour (6) which is used in this report to obtain the complete enumeration; (2) Direct algorithm by Smith and Dudek (156); and (3) Branch-and-bound algorithm by Brown and Lomnicki (27). A sample problem of flow shop having six jobs and three machines is solved by the above various techniques to illustrate these algorithm step-by-step.

2.1. Complete Enumeration Technique *

The linear graph algorithm which has been developed by Ashour (6) to generage feasible sequences and compute the schedule time for flow shop problems is used to obtain the complete enumeration solutions. This algorithm is based on linear graph theory presented by Heller (67). An operation of processing job j on machine m is represented by node (jm). Important features of the algorithm are: (1) it can permute a complete set of sequences or generate, at random, a subset of all sequences. Thus the algorithm is flexible in the sense that any number of sequences can be constructed and evaluated; (2) it constructs and evaluates a sequence of J jobs in exactly J iterations regardless of the number of machines involved. As a result, the algorithm schedules m nodes in one iteration and a sequence is eval-

*Adapted from Ashour, S., "A Decomposition Approach for the Machine Scheduling Problem," Ph.D. Thesis, University of Iowa, 1967.

uated immediately; and (3) the machine ordering can arbitrarily be assigned but the same for all jobs. For example, in the case of three machines, the machine ordering could be that all jobs are processed on machine 3 first, machine 1 second and machine 2 last. These features enables the algorithm to compute quite easily and quickly the schedule time as well as the construction of a sequence.

Consider the processing time and machine ordering matrices appeared on page 16 of this report, the initial schedule table is constructed. This table includes the following columns

Q_1: node designation, (jm),

Q_2: processing time, t_{jm},

Q_3: index of the sequence of job j on machine m

Q_4: starting time of node (jm)

Q_5: finishing time of node (jm)

The following algorithm evaluates the schedule time for the sequence $\{j_1\ j_2\ .\ .\ .\ j_J\}$

Step 1: Construct the initial scheduling table, by setting:

1.1. the nodes (jm) in column 1 of the machine ordering matrix under Q_1, first, those in column 2 second, . . , and those in column M last,

1.2. the corresponding processing times t_{jm} under Q_2, and by leaving

1.3. Q_3 (jm), Q_4 (jm) and Q_5 (jm) blank.

Step 2: Let U = 1, and X = 1

Step 3: Set j = jx

Step 4: Index the job sequence in all machines by setting:

Q_3 (jm_y) = U, y = 1, 2, . . ., M

Step 5: Check U

5.1. if U = 1, go to step 6.

5.2. if U > 1, let jo = j_{x-1} and replace Q_4 (jm_4)

by

max [Q_4 (jmy), Q_5 (j_om_y)], = 1, 2, . . ., M

Step 6: Compute the finishing time:

6.1. Compute Q_5 (jm_1) = Q_4 (jm_1) + Q_2 (jm_1)

6.2. replace Q_5 (jm_y) by max [Q_4 (jm_y), Q_5 (jm_{y-1})]

and compute

Q_5 (jm_y) = Q_r (jm_y) + Q_2 (jm_y)

y = 2, 3, . . ., M.

Step 7: Increase both x and U by one. Then repeat step 3 through 6 until all jobs are scheduled.

Step 8: Find the schedule time such that T* = max [Q_5 (jm)]

The following is a sample problem, which has been solved. The processing time and machine ordering matrices are

$$T^* = \begin{bmatrix} 4 & 5 & 5 \\ 2 & 17 & 7 \\ 2 & 10 & 4 \\ 10 & 8 & 2 \\ 7 & 15 & 6 \\ 9 & 4 & 11 \end{bmatrix} \qquad M^* = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \\ 51 & 52 & 53 \\ 61 & 62 & 63 \end{bmatrix}$$

Let the sequence {3 6 2 5 1 4} is there and by applying the above algorithm, the schedule time can be found.

According to step 1.1. at the algorithm, initial scheduling table is constructed by writing nodes (jm) in the first column as shown in the table I and according to step 1.2. corresponding processing times are filled in the second column. It is a point to be noted in this table that nodes under Q_1 are arranged in three groups such that those in column 1 of the machine ordering matrix are set first, those in column 2 second and those in column 3 last. In table I, nodes (j1) are set in the first group, nodes (j2) in the second, and nodes (j3) in the third, where j = 1, 2, . . ., 6.

In order to compute the schedule time of the sequence {3 6 2 5 1 4} one first schedules job 3 on all machines by setting Q_3 (31), Q_3 (32) and Q_3 (33) equal to one in the schedule table. Since Q_4 (31) = 0, the completion time of the node 31 in the unit of time is

$$Q_5\ (31) = Q_4\ (31) + Q_2\ (31) = 0 + 2 = 2$$

Job 3 cannot be processed on machine 2 unless it is has been processed on machine 1. Therefore, Q_4 (32) is replaced by

$$\max [Q_4 (32), Q_5 (31)] = \max [0, 2] = 2.$$

The finishing time of this node is

$$Q_5 (32) = Q_4 (32) + Q_2 (32) = 2 + 10 = 12.$$

In order to process job 3 on machine 3, replace Q_4 (33) by

$$\max [Q_4 (33), Q_5 (32)] = \max [0, 12] = 12,$$

and compute the completion time as

$$Q_5 (33) = Q_4 (33) + Q_2 (33) = 12 + 4 = 16$$

Now job 3, has been scheduled on machines 1, 2 and 3. The updated values are in Table 2.1.

Table 2.1

Initial Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
11	4			
21	2			
31	2			
41	10			
51	7			
61	9			
12	5			
22	17			
32	10			
42	8			
52	15			
62	4			
13	5			
23	7			
33	4			
43	2			
53	6			
63	11			

Table 2.2

Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
11	4			
21	2			
31	2	1	0	2
41	10			
51	7			
61	9			
12	5			
22	17			
32	10	1	2	12
42	8			
52	15			
62	4			
13	5			
23	7			
33	4	1	12	16
43	2			
53	6			
63	11			

Next, job 6 is scheduled on all machines by setting Q_3 (61), Q_3 (62) and Q_3 (63) equal to two. This indicates that job 6 is the second job to be scheduled on all machines. Obviously, job 6 cannot be processed on machine 1 before job 3 is completed on that machine. Therefore, to avoid conflict or overlapping, Q_4(61) is replaced by

$$\max\ [Q_4\ (61),\ Q_5\ (31)] = \max\ [0,\ 2] = 2$$

For the same reason, also replace Q_4 (62) with

$$\max\ [Q_4\ (62),\ Q_5\ (32)] = \max\ [0,\ 12] = 12,$$

and Q_4 (63) with

$$\max [Q_4 (63), Q_4 (33)] = \max [0, 16] = 16.$$

The completion time of node 61 is

$$Q_5 (61) = Q_4 (61) + Q_2 (61) = 2 + 9 = 11.$$

Again, job 6 cannot be processed on machine 2 before job 3 is completed on that machine. Therefore, $Q_4 (62)$ is replaced by

$$\max [Q_4 (62), Q_5 (61)] = \max [12, 11] = 11$$

Table 2.3

Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
11	4			
21	2			
31	2	1	0	2
41	10			
51	7			
61	9	2	2	11
12	5			
22	17			
32	10	1	2	12
42	8			
52	15			
62	4	2	12	16
13	5			
23	7			
33	4	1	12	16
43	2			
53	6			
63	11	2	16	27

Table 2.4

Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
11	4			
21	2	3	11	13
31	2	1	0	2
41	10			
51	7			
61	9	2	2	11
12	5			
22	17	3	16	33
32	10	1	2	12
42	8			
52	15			
62	4	2	12	16
13	5			
23	7	3	33	40
33	4	1	12	16
43	2			
53	6			
63	11	2	16	27

Table 2.5

Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
11	4	5	20	24
21	2	3	11	13
31	2	1	0	2
41	10	6	24	34
51	7	4	13	20
61	9	2	2	11
12	5	5	48	53
22	17	3	16	33
32	10	1	2	12
42	8	6	53	61
52	15	4	33	48
62	4	2	12	16
13	5	5	54	59
23	7	3	33	40
33	4	1	12	16
43	2	6	61	63
53	6	4	48	54
63	11	2	16	27

Then the finishing time of job 5 on machine 3 is

$$Q_5\ (62) = Q_4\ (62) + Q_2\ (62) = 12 + 4 = 16$$

Further, replace Q_4 (63) by

$$\max\ [Q_4\ (63),\ Q_5\ (62)] = \max\ [16,\ 16] = 16$$

and the completion time of node {63} is

$$Q_5\ (63) = Q_4\ (63) + Q_2\ (63) = 16 + 11 = 27$$

The updated schedule table shows that jobs 3 and 6 are scheduled on all machines and appear as Table 2.3.

Now job 2 is the third job to be scheduled according to the sequence of {3 6 2 5 1 4}. Table 2.4 shows the updated scheduling table by proceeding as before. Finally, when jobs 5, 1 and 4 are scheduled in this sequence, the final scheduling table of the six jobs is obtained as shown in Table 2.5.

For convenience, Table 2.6 shows the arranged scheduling table. This table is the final scheduling table after its row in each group have been arranged in ascending order according to Q_3 (jm). The schedule time for the sequence {3 6 2 5 1 4} is the maximum entry under Q_5, which is 63.

Table 2.6

Scheduling Table

Q_1	Q_2	Q_3	Q_4	Q_5
31	2	1	0	2
61	9	2	2	11
21	2	3	11	13
51	7	4	13	20
11	4	5	20	24
41	10	6	24	34
32	10	1	2	12
62	4	2	12	16
22	17	3	16	33
52	15	4	33	48
12	5	5	48	53
42	8	6	53	61
33	4	1	12	16
63	11	2	16	27
23	7	3	33	40
53	6	4	48	54
13	5	5	54	59
43	2	6	61	63

2.2. Direct Technique:

The basic concept of the direct technique for the flowshop scheduling problem involves decision rules to help fill each sequence - position by one of the candidate jobs. This procedure is directed to find the complete sequence which minimize the idle time. Johnson (80) has developed an algorithm for two-machine flow shop problem based on the above concept. The algorithm minimizes the accumulated idle time on the last machine, in processing each job which is similar to minimize the schedule time. Dudek and Teuton (38) have devised an algorithm based on Johnson's approach for the M-machine problem; however, counter-example has been given by Karush (82). The drawback is that the possible partial sequences have not been checked for dominance, while constructing the feasible sequences. Smith and Dudek (156) have modified the above algorithm, which overcomes the drawbacks to guarantee an optimal for the flow shop problem with arbitrary number of jobs and machines.

In order to discuss the basic idea of this algorithm, the following definitions and notations are used.

1. Candidate sequences are those partial sequences, which are generated through filling sequence position x except the last.
2. Dominated jobs are those jobs which are eliminated from further consideration as a possible candidate for a sequence-position x.
3. Dominated sequences are those partial sequences eliminated from further consideration.

4. A candidate set of jobs are those jobs which: (1) are not in the pre-sequence being considered; (2) have not been dominated; and (3) have not been used yet for dominance check.

5. Two or more partial sequences are considered equivalent when they have the same combinations of jobs but with different permutations.

The notations considered are

S_i Complete sequence, i, consisting of J jobs, $j_1\, j_2 \cdot \cdot \cdot j_J$

s pre-sequence consisting of x-1 scheduled jobs, $\{j_1\, j_2 \cdot \cdot \cdot j_{x-1}\}$

s_i partial sequence, i = 1, 2, . . ., n

$\bar{s}$ set of unscheduled jobs, $j_x, j_{x+1}, \ldots, j_J$

a_1, a_2 jobs included in $\bar{s}$ and competing for the sequence position x

$\bar{s}_1, \bar{s}_2$ represent all possible exclusive subsets of $\bar{s}$ excluding jobs a_1 and a_2

I(m, s) total idle time resulting from the processing of the pre-sequence s on machine m

k(m, sa) total processing time resulting from the processing of the pre-sequence s on machine m.

To clarify some of the above notations consider an example of nine jobs 1, 2, . . ., 9. Let s = {2, 5} , then $\bar{s}$ consists of the jobs 1, 3, 4, 6, 7, 8 and 9. Let jobs a_1 and a_2 be 1 and 9 respectively. The remaining jobs in the set $\bar{s}$ are then 3, 4, 6, 7 and 8. Consequently, one may consider two subsets, $\bar{s}_1$ and $\bar{s}_2$ such that

$\bar{s}_1$ consists of jobs 3 and 6,

$\bar{s}_2$ consists of jobs 4 and 8.

According to the above notation, s_1 consisting of the same jobs in s may have the sequence {5, 2}, thus s_1 and s_2 are equivalent.

This algorithm is based on two dominance checks:

1. Job dominance: consider the two sequences

$$S_1 = sa_1a_2\bar{s}_1\bar{s}_2 \qquad \text{and} \qquad S_2 = sa_2\bar{s}_1a_1\bar{s}_2$$

Since the presequence s consists of x - 1 jobs, jobs a_1 and a_2 are competing for the sequence position x. Job a_2 is eliminated from further consideration for this position in favor of job a_1 if the M-1 conditions

$$k\ (m,\ sa_2) \geq \max\ [k(m,\ sa_1\ a_2),\ k(m,\ sa_1)],\ \ldots\ldots\ldots\ldots \quad (1)$$

$$m = 2,\ 3,\ \ldots,\ M.$$

are satisfied. If any of the above conditions is not satisfied, jobs a_1 and a_2 are retained for further consideration. The job dominance will ensure that no set of sequences $\{sa_2\}$ is discarded unless one equally good or better $\{sa_1a_2\}$ is retained.

2. Sequence dominance: Consider the two sequences

$$S_1 = s_1\bar{s} \qquad \text{and} \qquad S_2 = s_2\bar{s},$$

where the partial sequences s_1 and s_2 are equivalent. The partial sequence s_2 is eliminated from further consideration in favor of the partial sequence s_1, if the M - 1 conditions

$$I\ (m,\ s_1)\ \leq\ I\ (m,\ s_2), \quad \dots \dots \quad (2)$$

$$m = 2, 3, \dots, M$$

are satisfied. If any of the above conditions is not satisfied, both partial sequences are retained for further consideration. The sequence dominance will ensure that no partial sequence $\{s_2\}$ will be discarded unless an equally good or better sequence $\{s_1\}$ is retained. It should be pointed out that this sequence dominance check has been also used by Ignall and Schrage (68).

For the dominance checks, the terms, I (m, s), K (m, sa), and R(m, sa) are presented below in mathematical forms. The idle time I (m, s), $m = 1, 2, \dots, M$ is

$$I\ (1,\ s) = 0,$$

$$I\ (2,\ s) = \max_{1 \leq u \leq x-1} \left[\sum_{k=1}^{u} t_{j_k 1} - \sum_{k=1}^{u-1} t_{j_k 2} \right],$$

$$I\ (3,\ s) = \max_{1 \leq u \leq x-1} \left\{ \sum_{k=1}^{u} t_{j_k 2} - \sum_{k=1}^{u-1} t_{j_k 3} + \right.$$

$$\left. \max_{1 \leq v \leq v-1} \left[\sum_{k=1}^{v} t_{j_k 1} - \sum_{k=1}^{v-1} t_{j_k 2} \right] \right\} ,$$

$$I\ (4,\ s) = \max_{1 \leq u \leq x-1} \left\{ \sum_{k=1}^{u} t_{j_k 3} - \sum_{k=1}^{u-1} t_{j_k 4} + \right.$$

$$\max_{1\leq v\leq -1} \left[\sum_{k-1}^{v} t_{j_k 2} - \sum_{k=1}^{v-1} t_{j_k 3} + \max_{1\leq w\leq -1} \left[\sum_{k=1}^{w} t_{j_k 1} - \sum_{k=1}^{w-1} t_{j_k 2} \right]\right]\}$$

.

.

.

$$I(M,s) = \max_{1\leq u\leq x-1} \{ \sum_{k=1}^{u} t_{j_k M-1} - \sum_{k=1}^{u-1} t_{j_k M} +$$

$$\max_{1\leq v\leq u-1} \left[\sum_{k=1}^{u} t_{j_k, M-2} - \sum_{k=1}^{v-1} t_{j_k, M-3} + \ldots \right.$$

$$\ldots + \max_{1\leq w\leq t} \left[\sum_{k=1}^{w} t j_k 1 - \sum_{k=1}^{w-1} t_{j_k 2} \right] \ldots]\} \; \ldots \ldots \quad (3)$$

Consequently, the terms k(m, sa) and R(m, sa) which will be used for job dominance checks are

$$K(m, sa) = \sum_{j\varepsilon sa} t_{j,m-1} - \sum_{j\varepsilon s} t_{j_m} +$$

$$\max\left[I(m-1, s),\ K(m-1, sa)\right], \ldots \ldots \ldots \quad (4)$$

$$m = 2, 3, \ldots, M.$$

where

$$K(0, sa) = 0,$$

$$I(0, s) = 0,$$

and

$$R(m, sa) = I\ (m, sa) + \sum_{j\varepsilon s} t_{j_m} - \sum_{j\varepsilon s} t_{j_1}, \qquad (5)$$

In considering the job dominance check mentioned in (1), the M-1 conditions may be simplified to the following:

$$H\ (m, sa_2) \geq \max\ [H\ (m, sa_1), H\ (m, sa_1\ a_2)]\ . \qquad (6)$$

$$m = 2, 3, \ldots, M.$$

where

$$H\ (m, sa_2) = K\ (m, sa_2) + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j,m-1} + t_{a_2,m-1} - \sum_{j\varepsilon s} t_{jm} +$$

$$\max\ [I\ (m-1, s), K(m-1, sa_2)] +$$

$$\sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j,m-1} + t_{a_2,m-1} - \sum_{j\varepsilon s} t_{jm} +$$

$$\max\ [R(m-1, s), H(m-1, sa_2)] - \sum_{j\varepsilon s} tj,m-1 +$$

$$\sum_{j\varepsilon s} t_{j1} + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= t_{a_2,m-1} + \max\ [R(m-1, s), H(m-1, sa_2)]\ . \qquad (7)$$

$$m = 2, 3, \ldots, M.$$

$$H(m, sa_1) = K(m, sa_1) + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j,m-1} + t_{a_1,m-1} - \sum_{j\varepsilon s} t_{jm} +$$

$$\max\,[I\,(m-1, s),\ K(m-1, sa_1)] + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j,m-1} + t_{a_1,m-1} - \sum_{j\varepsilon s} t_{jm} +$$

$$\max\,[R(m-1, s),\ H\,(m-1, sa_1)] - \sum_{j\varepsilon s} t_{jm} +$$

$$\sum_{j\varepsilon s} t_{j1} + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= t_{a_1,m-1} + \max\,[R\,(m-1, s),\ H\,(m-1, sa_1)] \quad . \ . \ . \ . \ (8)$$

$$m = 2, 3, \ldots, M.$$

and

$$H\,(m, sa_1\,a_2) = K\,(m, sa_1\,a_2) + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon sa_1a_2} t_{j,m-1} - \sum_{j\varepsilon sa_1} t_{jm} +$$

$$\max\,[I\,(m-1, s),\ K(m-1, sa_1),\ K(m-1, sa_1a_2)]$$

$$+ \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j,m-1} + t_{s_1,m-1} + t_{a_2,m-1} - \sum_{j\varepsilon s} t_{jm}$$

$$- t_{a_1 m} + \max\,[R\ (m-1,\ s),\ H\ (m-1,\ sa_1),\ H\ (m-1,\ sa_1\ a_2)]$$

$$+ \sum_{j\varepsilon s} t_{j1} - \sum_{j\varepsilon s} t_{j,m-1} + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= t_{a_1,m-1} + t_{a_2,m-1} - t_{a_1 m} +$$

$$\max\,[R\ (m-1,\ s),\ H(m-1,\ sa_1),\ H\ (m-1,\ sa_1\ a_2)] \quad . \ . \ . \ (9)$$

$$m = 2,\ 3,\ .\ .\ .,\ M.$$

Accordingly, the following M-1 conditions are stated with those simplifications achieved above:

Condition 1

For condition 1, Formula (6) becomes

$$H\ (2,\ sa_2) \geq \max\,[H\ (2,\ sa_1),\ H\ (2,\ sa_1\ a_2)]$$

where

$$H\ (2,\ sa_2) = K\ (2,\ sa_2) + \sum_{j\varepsilon s} t_{j2} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j1} + t_{a_2 1} - \sum_{j\varepsilon s} t_{j2} + \sum_{j\varepsilon s} t_{j2} - \sum_{j\varepsilon s} t_{j1}$$

$$= ta_2 1,$$

$$H\ (2,\ sa_1) = K\ (2,\ sa_1) + \sum_{j\varepsilon s} t_{j2} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j1} + t_{a_1 1} - \sum_{j\varepsilon s} t_{j2} + \sum_{j\varepsilon s} j_{j2} - \sum_{j\varepsilon s} t_{j1}$$

$$= ta_1 1,$$

and

$$H\ (2,\ sa_1\ a_2) = K\ (2,\ sa_1\ a_2) + \sum_{j\varepsilon s} t_{jm} - \sum_{j\varepsilon s} t_{j1}$$

$$= \sum_{j\varepsilon s} t_{j1} + t_{a_1 1} + t_{a_2 1} - \sum_{j\varepsilon s} - t_{a_1 2}$$

$$+ \sum_{j\varepsilon s} t_{j2} - \sum_{j\varepsilon s} t_{j1},$$

$$= ta_1 1 + t_{a_2 1} - t_{a_1 2}.$$

Thus

$$ta_2 1 \geq \max\ [t_{a_1 1},\ (t_{a_1 1} + t_{a_2 1} - t_{a_1 2}] \ldots\ldots\ldots\ldots (10)$$

Condition 2

For condition 2, Formula (6) becomes

$$H(3,\ sa_2) \geq \max\ [H\ (3,\ sa_1),\ H\ (3,\ sa_1\ a_2)],$$

which may be simplified as done condition 1 to

$$t_{a_2 2} + \max\ [R(2,\ s),\ \ H\ (2,\ sa_2)] \geq \max\ t_{a_1 2} + \max\ [R\ (2,\ s),$$

$$H\ (2,\ sa_1)],\ ta_1 2 + t_{a_2 2} - t_{a_1 3} + \max\ [R\ (2,\ s),\ H\ (2,\ sa_1),\ H(2,\ sa_1\ a_2)],$$

.
.
.
. . . . (11)
.
.

Condition M-1

For condition M-1, Formula (6) becomes

$$t_{a_2, M-1} + \max\ [R\ (M-1,\ s),\ H(M-1,\ s),\ H(M-1,\ sa_2)] \geq$$

$$\max \left[t_{a_1,M-1} + \max \left[R\ (M-1,\ s),\ H\ (M-1,\ sa_1)\right], \right.$$

$$t_{a_1,\ M-1} + t_{a_2 M-1} - t_{a_1 M} + \max \left[R\ (M-1,\ s), \right.$$

$$\left. H\ (M-1,\ sa_1),\ H\ (M-1,\ sa_1 a_2)\right] \right] \quad \ldots\ldots\ldots\ldots \quad (12)$$

The following algorithm, which will generate an optimal solution is stated step by step:

Step 1: Let the sequence-position $x = 1$

Step 2: Check the presequence s, $s = \{j_1\ j_2\ \ldots\ j_{x-1}\}$

2.1. if the presequence is empty, let

$R\ (m,\ s) = 0$, and go to step 3

2.2. if the presequence consists of one or more jobs, compute

$$R\ (m,\ s) = I\ (m,\ s) + \sum_{j \varepsilon s} t_{jm} - \sum_{j \varepsilon s} t_{j1}$$

$$m = 2,\ 3,\ \ldots,\ M.$$

Step 3: Select a job, a_1 and check condition 1 as follows:

3.1. select a job to be referred to as a_1, from the set of the unscheduled jobs, $\bar{s}$ such that

$$t_{a_1 1} = \min_{j \varepsilon s} [t_{j1}]$$

where j is not previously selected for sequence position x.

3.2. check condition 1 such that

$$t_{a2} > t_{a1}$$

3.2.1. if condition 1 is satisfied, go to step 4.

3.2.2. if condition 1 is not satisfied, return to step 3.

Step 4: Select a job a_2 and check conditions 2 through M-1 as follows:

4.1. select a job, to be referred to as a_2, from the set of unscheduled jobs, $\bar{s}$ such that $a_2 \neq a_1$

4.2. check all the M-2 conditions, such that

$$H(m, sa_2) \geq \max [H(m, sa_1), H(m, sa_1 a_2)],$$
$$m = 3, 4, \ldots, M.$$

4.2.1. if any one of the M-2 conditions is not satisfied, retain jobs a_1 and a_2 as candidates for the sequence-position x. Go to step 5.

4.2.2. if all the M-2 conditions are satisfied, eliminate job a_2 from further consideration for the sequence-position x. Go to step 5.

Step 5: Repeat steps 3 and 4 for all possible combinations of a_1 and a_2 in order to fill the sequence-position x.

Step 6: Check x

6.1. if $x = 1$, go to step 10

6.2. if $x > 1$, develop a number of candidate sequences $\{j_1 \, j_2 \, . \, . \, . \, j_x\}$ by repeating steps 2 through 5 for all obtained pre-sequences having x-1 sequence-positions.

Step 7: Select a candidate sequence, to be referred to as s_1.

Step 8: Select a candidate sequence s_2 and check conditions 2 through M-1 as follows:

8.1. Select a candidate sequence, to be referred to as s_2, from the set of all candidate sequence such that s_1 and s_2 are equivalent:

8.1.1. if such sequence exist, go to step 8.2.

8.1.2. if such sequence does not exist, retain candidate sequence s_1. Go to step 9.

8.2. Check all the M-1 conditions such that

$I\ (m,\ s_1) \leq I\ (m,\ s_2),\ m = 2, 3, \ldots, M.$

8.2.1. if any of the conditions is not satisfied, retain candidate sequences, s_1 and s_2. Go to step 9.

8.2.2. if all conditions are satisfied, candidate sequence s_2 is dominated. Retain candidate sequence s_1. Go to step 9.

8.2.3. if none of the conditions are satisfied, candidate sequence s_1 is dominated. Retain sequence s_2. Go to step 9.

Step 9: Repeat steps 7 and 8 for all possible combinations of the equivalent candidate sequences s_1 and s_2.

Step 10: Check x

10.1 if $x < J - 2$, let $x = x + 1$. Go to step 6.

10.2 if $x \geq J - 2$, go to step 11.

Step 11: Compare the candidate sequences for the condition:

$I\ (M,\ sa_1\ a_2) \leq I\ (M,\ sa_2\ a_1)$

11.1. If this condition is satisfied, the sequence $\{sa_2\ a_1\}$

is eliminated.

11.2. If this condition is not satisfied, the sequence

$sa_1 a_2$ is eliminated.

Step 12: Find the optimal sequence(s) by the following:

12.1. Evaluate each of the sequences retained in step 11

such that

$$T = \sum_{\substack{x=1 \\ j \in s_x}}^{J} t_{j_x m} + \max [I (M, sa_1), K (M, sa_2), K (M, sa_1 a_2)]$$

12.2. Selecting the sequence(s) which has the minimum

schedule time.

The same flowshop problem of six jobs and three machines discussed in section 2.1, is solved to illustrate the direct technique. For convenience, the processing time and machine ordering matrices are reproduced below:

$$T^* = \begin{bmatrix} 4 & 5 & 5 \\ 2 & 17 & 7 \\ 2 & 10 & 4 \\ 10 & 8 & 2 \\ 7 & 15 & 6 \\ 9 & 4 & 11 \end{bmatrix}, \qquad M^* = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \\ 51 & 52 & 53 \\ 61 & 62 & 63 \end{bmatrix}$$

The computation is carried out step-by-step to illustrate the above algorithm.

For sequence - position 1, there is no pre-sequence. In other words the pre-sequence is empty. Therefore

$$R\ (m, s) = 0, \qquad m = 2, 3.$$

Following the step 3, job 2 is selected as a_1, since

$$t_{21} = \min\ [t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}]$$

$$= \min\ [4, 2, 2, 10, 7, 9]$$

$$= 2$$

Thus, condition 1 is satisfied.

$$t_{22} > t_{21}$$

Consider job 1 as a_2 according to step 5 and check job 2 versus job 1 for condition 2 such that

$$H\ (2, sa_2) \geq \max\ [H\ (2, sa_1), H\ (2, sa_1\ a_2)]$$

where

$$H\ (2, sa_2) = t_{a_2 2} + \max\ [R(2, s), t_{a_2 1}],$$

$$H\ (2, sa_1) = t_{a_1 2} + \max\ [R(2, s), t_{a_1 1}],$$

and

$$H\ (2, sa_1\ a_2) = t_{a_1 2} + t_{a_1 2} - t_{a_1 3} +$$

$$\max\ [R(2, s), t_{a_1 1}, t_{a_1 1} + t_{a_2 1} - t_{a_1 2}],$$

or

$$H\ (2,\ sa_2) = 5 + \max\ [0,\ 4] = 9,$$

$$H\ (2,\ sa_1) = 17 + \max\ [0,\ 2] = 19,$$

and

$$H\ (2,\ sa_1\ a_2) = 17 + 10 - 7 + \max\ [0,\ 2,\ 2 + 2 - 17] = 22$$

Accordingly, $9 \not\geq [19, 22]$. Since condition 2 is not satisfied, job 1 is retained for further consideration. Similarily, job 2 is checked versus the remaining jobs. The results are summarized below

Sequence - position 1			Pre-sequence {-}
a_1 versus	a_2	Condition 2	Results
2	1	$9 \not\geq \max\ [19, 17]$	not satisfied
2	3	$12 \not\geq \max\ [19, 22]$	not satisfied
2	4	$18 \not\geq \max\ [19, 20]$	not satisfied
2	5	$22 \not\geq \max\ [19, 32]$	not satisfied
2	6	$13 \not\geq \max\ [19, 16]$	not satisfied

Upon examining the above results, job 2 does not dominate any of the remaining jobs for the first position, and, thus any of the jobs 1, 2, 3, 4, 5 or 6 may fill the pre-sequence position. Following step 5, consider job 3 as a_1, since $t_{32} > t_{31}$. Therefore, condition 1 is satisfied. Similarily, job 3 is checked versus the remaining jobs. The results are summarized below:

Sequence-position 1			Pre-sequence {-}
a_1	versus a_2	Condition 2	Results
3	1	$9 \not\geq \max\,[12, 13]$	not satisfied
3	2	$19 \not\geq \max\,[12, 25]$	not satisfied
3	4	$18 \geq \max\,[12, 16]$	satisfied
3	5	$22 \not\geq \max\,[12, 23]$	not satisfied
3	6	$13 \geq \max\,[12, 12]$	satisfied

From the above table, job 3 dominates jobs 4 and 6, and, therefore they are eliminated from further consideration for the first sequence-position. Accordingly, jobs 1, 2, 3 or 5 may fill the first sequence position. Repeating the above computation for the each of the remaining jobs as a_1, the results are summarized in the following tables.

Sequence-position 1			Pre-sequence {-}
a_1	versus a_2	Condition 2	Results
1	2	$19 \not\geq \max\,[9, 21]$	not satisfied
1	3	$12 \not\geq \max\,[9, 14]$	not satisfied
1	4	$18 \geq \max\,[9, 12]$	satisfied
1	5	$22 \geq \max\,[9, 21]$	satisfied
1	6	$13 \geq \max\,[9, 8]$	satisfied

This table shows that jobs 1, 2 or 3 may fill the first sequence position.

Sequence-position 1		pre-sequence {-}
a_1 versus a_2	Condition 2	Results
5 1	$9 \not\geq \max[22, 21]$	not satisfied
5 2	$19 \not\geq \max[22, 26]$	not satisfied
5 3	$12 \not\geq \max[22, 26]$	not satisfied
5 4	$18 \not\geq \max[22, 22]$	not satisfied
5 6	$13 \not\geq \max[22, 20]$	not satisfied

This table shows that jobs 1, 2, 3, 4, 5 or 6 may fill the first sequence position. In checking the job dominance for all possible combinations of a_1 and a_2, it is concluded that jobs 1, 2, or 3, may fill the first sequence-position. This is because these jobs dominate the others. The sequence position is 1, x becomes 2, according to step 10. The resulting pre-sequences are then {1}, {2}, and {3}.

For sequence position 2, the pre-sequences exist, and, therefore step 2.2 is computed for each of the above pre-sequences. In considering the pre-sequences {1},

$$R(2, 1) = \max_{1 \leq k \leq x-1} \left[\sum_{x=1}^{k} t_{j_x 1} - \sum_{x=1}^{k-1} t_{j_x 2} \right] + \sum_{j \varepsilon s} t_{j2} - \sum_{j \varepsilon s} t_{j1}$$

$$= t_{11} + t_{12} - t_{11}$$

$$= 4 + 5 - 4$$

$$= 5$$

Now repeating steps 3-4, consider job 2 as a_1,

since

$$t_{22} > t_{21},$$

condition 1 is satisfied. Consider job 3 as a_2 and check condition 2 such that

$$H(2, sa_2) \geq \max [H(2, sa_1), H(2, sa_1 a_2)]$$

where

$$H(2, sa_2) = t_{a_2 2} + \max [R(2, s), t_{a_2 1}],$$

$$H(2, sa_1) = t_{a_1 2} + \max [R(s, s), t_{a_1 1}],$$

and

$$H(2, sa_1 a_2) = t_{a_1 2} + t_{a_2 2} - t_{a_1 3} +$$

$$\max [R(2, s), t_{a_1 1}, t_{a_1 1}, + t_{a_2 1} - t_{a_1 2}]$$

or

$$H(2, 13) = 10 + \max [5, 2] = 15,$$

$$H(2, 12) = 17 + \max [5, 2] = 22,$$

and

$$H(2, 123) = 17 + 10 - 7 + \max [5, 2, 2 + 2 - 17]$$

$$= 17 + 10 - 7 + 5$$

$$= 25$$

Accordingly, $15 \ngeq \max [22, 25]$. Since condition 2 is not satisfied, job 3 is retained for further consideration. Similarly, job 2 is checked versus the remaining jobs. The results are summarized below

Sequence-position 2			Pre-sequence {1-}
a_1 versus	a_2	Condition 2	Results
2	3	15 $\ngeq$ max [22, 25]	not satisfied
2	4	18 $\ngeq$ max [22, 23]	not satisfied
2	5	22 $\ngeq$ max [22, 30]	not satisfied
2	6	13 $\ngeq$ max [22, 19]	not satisfied

Upon examining the above results, job 2 does not dominate any of the remaining jobs for the second sequence position, and, thus any of the jobs 2, 3, 4, 5 or 6 may fill the pre-sequence position. Following step 5, consider job 3 as a_1, since $t_{32} > t_{31}$, therefore, condition 1 is satisfied. Similarly, job 3 is checked versus the remaining jobs. The results are summarized below:

Sequence-position			Pre-sequence {1-}
a_1 versus	a_2	Condition 2	Results
3	2	22 $\ngeq$ max [15, 28]	not satisfied
3	4	18 $\ngeq$ max [15, 19]	not satisfied
3	5	22 $\ngeq$ max [15, 26]	not satisfied
3	6	13 $\ngeq$ max [15, 15]	not satisfied

From the above table, job 3 also does not dominate any of the jobs for the second sequence-position. Accordingly, jobs 2, 3, 4, 5 or 6 may fill the second sequence-position. Repeating the above computation for the each of the remaining jobs as a_1, the results are summarized in the following table.

Sequence-position 2			Pre-sequence {1-}
a_1 versus	a_2	Condition 2	Results
5	2	$22 \nsucceq \max [22, 33]$	not satisfied
5	3	$15 \nsucceq \max [22, 26]$	not satisfied
5	4	$18 \nsucceq \max [22, 24]$	not satisfied
5	6	$14 \nsucceq \max [22, 20]$	not satisfied

This table shows that job 5 does not dominate any of the jobs for the second sequence position. Since condition 1 is not satisfied for jobs 4 and 6, so they cannot be taken as a_1. In checking the job dominance for all possible combinations of a_1 and a_2, it is concluded that jobs 2, 3, 4, 5 or 6 may fill the second sequence-position. The resulting possible candidate sequences are {12}, {13}, {14}, {15} and {16}.

Accordingly, steps 2 through 5 are executed for the other pre-sequences {2} and {3}, the candidate sequences obtained for the pre-sequences {2} and {3} are {21}, {26} and {31}, {36}, respectively. Finally the candidate sequences are

{12}, {13}, {14}, {15}, {16},

{21}, {26}, {31} and {36}

Before filling the sequence-position 3, the sequence dominance is checked.

Following step 7, consider sequence {12} as s_1. Since sequence {21} is equivalent to {12}, the former sequence is considered as s_2. For both s_1 and s_2, the M-1 conditions are checked according to the step 8.2.

$$I(2, 12) = \max_{1 \le k \le 2} \left[\sum_{x=1}^{k} t_{j_x 1} - \sum_{x=1}^{k-1} t_{j_x 2} \right]$$

$$= \max \begin{bmatrix} 4 - 0 \\ 6 - 17 \end{bmatrix}$$

$$= \max \begin{bmatrix} 4 \\ -11 \end{bmatrix}$$

$$= 4.$$

Similarly,

$$I(2, 21) = 2;$$

$$I(3, 12) = \max_{1 \le k \le 2} \left[\sum_{x=1}^{k} t_{j_x 2} - \sum_{x=1}^{k-1} t_{j_x 3} + \max_{1 \le h \le k} \left\{ \sum_{x=1}^{h} t_{j_x 1} - \sum_{x=1}^{h-1} t_{j_x 2} \right\} \right]$$

$$= \max \begin{bmatrix} t_{12} + t_{11} \\ t_{12} + t_{22} - t_{13} + \max \begin{Bmatrix} t_{11} \\ t_{11} + t_{21} - t_{12} \end{Bmatrix} \end{bmatrix}$$

$$= \max \begin{bmatrix} 5 + 4 \\ 5 + 17 - 5 + \max \begin{Bmatrix} 4 \\ 6 - 1 \end{Bmatrix} \end{bmatrix}$$

$$= \max \begin{bmatrix} 9 \\ 22 \end{bmatrix}$$

$$= 22$$

Similarly,

$$I(3, 21) = 17.$$

Then, condition 1

$$I(2, 12) < I(2, 21)$$

or

$$4 \nless 2$$

and condition 2

$$I(3, 12) < I(3, 21)$$

or

$$22 \nless 17$$

are not satisfied. As a result, the candidate sequence {12} is eliminated from further consideration. Similarly, candidate sequences {13} as s_1 and {31} as s_2 are checked for the sequence dominance. Conditions 1 and 2 are not satisfied since

$$I(2, 13) = 4,$$
$$I(2, 31) = 2,$$
$$I(3, 13) = 14,$$
$$I(3, 31) = 13$$

for condition 1,

$$4 \nless 2$$

and for condition 2,

$14 \nprec 13.$

Therefore, the candidate sequence {13} is eliminated from further consideration. Since these are the only equivalent candidate sequences, the remaining ones are retained. Consequently the pre-sequences retained for further consideration are,

{14}, {15}, {16}, {21},

{26}, {31}, and {36}

Since, sequence-position 2 is less than J-2 or 4, according to step 10, steps 2 through 9 are repeated to fill the sequence-position 3.

For sequence-position 3, the pre-sequences exist, and, therefore step 22 is computed for each of the above pre-sequences. In considering the pre-sequence {14}

$$R(2, 14) = I(2, 14) + \sum_{x=1}^{2} t_{j_x 2} - \sum_{x=1}^{2} t_{j_x 1}$$

$$= \max \begin{bmatrix} t_{11} \\ t_{11} + t_{41} - t_{12} \end{bmatrix} + t_{12} + t_{42} - t_{11} - t_{41}$$

$$= \max \begin{bmatrix} 4 \\ 14 - 5 \end{bmatrix} + 5 + 8 - 4 - 10$$

$$= \max \begin{bmatrix} 4 \\ 9 \end{bmatrix} + 5 + 8 - 4 - 10$$

$$= 9 + 5 + 8 - 4 - 10$$

$$= 8$$

Now repeating steps 3 through 4, consider job 2 as a_1, since

$$t_{22} > t_{21},$$

Condition 1 is satisfied. Consider job 3 as a_2 and check the condition 2 such that

$$H(2, sa_2) \geq \max [H(2, sa_1), H(2, sa_1 a_2)]$$

where

$$H(2, sa_2) = t_{a_2 2} + \max [R(2, s), t_{a_2 1}],$$

$$H(2, sa_1) = t_{a_1 2} + \max [R(2, s), t_{a_1 1}],$$

and

$$H(2, sa_1 a_2) = t_{a_1 2} + t_{a_2 2} - t_{a_1 3} +$$

$$\max [R(2, s), t_{a_1 1}, t_{a_1 1} + t_{a_2 1} - t_{a_1 2}]$$

or

$$H(2, 143) = 10 + \max [8, 2] = 18,$$

$$H(2, 142) = 17 + \max [8, 2] = 25,$$

and

$$H(2, 1423) = 17 + 10 - 7 + \max [8, 2, 2 + 2 - 17]$$

$$= 17 + 10 - 7 + 8$$

$$= 28$$

Accordingly, $18 \ngeq \max [25, 28]$. Since condition 2 is not satisfied, job 3 is retained for further consideration. Similarly, job 2 is checked versus the remaining jobs. The results are summarized below.

Sequence-position 3			Pre-sequence {14-}
a_1 versus	a_2	Condition 2	Results
2	3	$18 \ngeq \max\ [25, 28]$	not satisfied
2	5	$25 \ngeq \max\ [25, 33]$	not satisfied
2	6	$13 \ngeq \max\ [25, 22]$	not satisfied

upon examining the above results, job 2 does not dominate any of the remaining jobs for the second sequence position, and, thus any of the jobs 2, 3, 5 or 6 may fill the empty pre-sequence position. Repeating the above combination for each of the remaining jobs as a_1, the results are summarized in the following table.

Sequence-position 3			Pre-sequence {14-}
a_1 versus	a_2	Condition 2	Results
3	2	$25 \ngeq \max\ [18, 31]$	not satisfied
3	5	$23 \ngeq \max\ [18, 29]$	not satisfied
3	6	$13 \ngeq \max\ [18, 18]$	not satisfied

This table shows that job 3 does not dominate any of the jobs for the third sequence-position.

Sequence-position 3			Pre-sequence {14-}
a_1 versus	a_2	Condition 2	Results
5	2	$25 \not\geq \max [23, 35]$	not satisfied
5	3	$18 \not\geq \max [23, 28]$	not satisfied
5	6	$13 \not\geq \max [23, 22]$	not satisfied

This table shows that job 5 does not dominate any of the remaining jobs for the sequence-position 3, and, thus any of the jobs 2, 3, 5 or 6 may fill the pre-sequence. Jobs 4 and 6 cannot be taken as a_1, since condition 1 is not satisfied for them. The resulting possible candidate sequences are {142}, {143}, {145} and {156}.

Accordingly, steps 2 through 5 are executed for the other pre-sequences {15}, {16}, {21}, {26}, {31}, and {36}, the results of these computations shows for pre-sequence {15}, the candidate sequence are {152}, {153}, {154} and {156}, for the presequence 16 , the candidate sequence is {163}, for the pre-sequence {21}, the candidate sequences are {213}, {214}, {215} and {216}, for the pre-sequence, {26}, the candidate sequence is {261}, for the pre-sequence, {31}, the candidate sequences are {312}, {314}, {315} and {316} and for the pre-sequence 36 , the candidate sequence obtained is {361}. All the candidate sequences are

{142}, {143}, {145}, {146},

{152}, {153}, {154}, {156},

{213}, {214}, {215}, {216},

{261}, {312}, {314}, {315},

{316}, {361}, {163}.

Before filling the sequence-position 4, the sequence dominance is checked. Following step 7, consider sequence {142} as s_1. Since sequence {214} is equivalent to {142}, the former sequence is considered as s_2. For both s_1 and s_2, the M-1 conditions are checked according to the step 8.2.

$$I(2, 142) = \max_{1 \leq k \leq 3} \left[\sum_{x=1}^{k} t_{j_x 1} - \sum_{x=1}^{k-1} t_{j_x 2} \right]$$

$$= \max \begin{bmatrix} 4 \\ 14 - 13 \\ 16 - 13 \end{bmatrix}$$

$$= \max \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix} .$$

$$= 4$$

Similarly,

$$I(2, 214) = 2$$

$$I(3, 142) = \max_{1 \leq k \leq 3} \left[\sum_{x=1}^{k} t_{j_x 2} - \sum_{x=1}^{k-1} t_{j_x 3} + \max_{1 \leq h \leq k} \left\{ \sum_{x=1}^{h} t_{j_x 1} - \sum_{x=1}^{h-1} t_{j_x 2} \right\} \right]$$

$$= \max \begin{bmatrix} 5 + 4 \\ 13 - 5 + \max \begin{pmatrix} 4 \\ 14 - 5 \end{pmatrix} \\ 30 - 7 + \max \begin{pmatrix} 4 \\ 14 - 5 \\ 16 - 13 \end{pmatrix} \end{bmatrix}$$

$$= \max \begin{bmatrix} 9 \\ 17 \\ 32 \end{bmatrix}$$

$$= 32$$

Similarly,

$$I(3, 214) = 32$$

Then, condition 1

$$I(2, 142) < I(2; 214)$$

or

$$4 \nless 2$$

and condition 2

$$I(3, 142) < I(3, 214)$$

or

$$32 \nless 22$$

are not satisfied. As a result, the candidate sequence {142} is eliminated from further consideration. Similarly considering {143} as s_1, and its equivalent candidate sequence {314} as s_2, the candidate sequence {143} is eliminated from further consideration, candidate sequence {145} is eliminated when checked against its equivalent candidate sequence {154}, candidate sequence {152} is eliminated when checked against its equivalent candidate sequence {215}, candidate sequence {153} is eliminated in favor of its equivalent candidate sequence {315}, the candidate sequence {163} is eliminated when checked against its equivalent sequence {316}, however, {316} is eliminated when checked against its equivalent candidate sequence {361}, candidate sequence {261} its eliminated when checked against {216} however, for the equivalent candidate sequences {213} and {312} one condition is satisfied and other is not, therefore, both the sequences are retained. Since these are the only equivalent candidate sequences, the remaining ones are retained. Consequently, the pre-sequences retained for further consideration are:

{146}, {154}, {156}, {213},

{214}, {215}, {216}, {312},

{314}, {315}, {316}.

Proceeding as in the algorithm, the first J-2 sequence-positions are filled. Once this is done, the last two positions are filled, and then the complete sequences are checked according to step 11. Finally the retained sequences are evaluated. The following are the complete sequences generated by the algorithm and the corresponding schedule times.

Sequence No.	Sequence	Schedule Time
1	{3 1 5 6 2 4}	63*
2	1 5 6 4 2 3	69
3	3 6 2 5 1 4	63*
4	2 1 3 4 6 5	67
5	3 1 4 6 2 5	67
6	3 6 1 5 2 4	64
7	2 1 5 3 6 4	66
8	2 1 6 5 3 4	63*
9	3 1 5 4 6 2	68
10	3 6 1 4 2 5	71
11	2 1 5 4 6 3	66
12	2 1 6 4 5 3	65

*The optimal schedule time

Upon examining the above schedule times, sequences 1, 3 and 8 have the minimum schedule time 63. It is interested to note that 28 sequences gave the optimal when solved by complete enumeration.

2.3. Branch-and-bound technique:

Branch-and-bound technique provides a systematic search for a subset of feasible sequences, from which one or more may yield the optimal. This procedure may best be illustrated by a scheduling tree which consists of nodes, each representing a partial sequence $\{j_1\ j_2\ .\ .\ .\ j_k\}$, where $k < J$. At level 1, the scheduling tree is initialized by J nodes, each of which consists of a partial sequence having one job, $\{j_1\}$. Each of these nodes may further be branched into a number of node, at level 2, each consisting of a partial sequence $\{j_1\ j_2\}$. As one moves down the tree, the number of nodes branched from a node is decreased by one than that of the preceding level; and the number of jobs in each node (partial sequence) is increased by one.

Reduction in the generation of nodes at each level can be achieved through a bounding approach. In this approach, a lower bound for each node is computed. A particular branch is then considered from a node which has the minimum lower bound on the schedule time. Obviously, the power of the branching procedure depends heavily on the quality of the lower bound, particularily those used in the early stages of branching. Various bounding procedures have been proposed by several investigators. The following notation is considered to discuss those lower bounds:

L level of the scheduling tree; $L = 1, 2, . . ., J-1$

n node consisting of a partial sequence of scheduled jobs, $\{j_1\ j_2\ .\ .\ .\ j_k\}$

$\bar{n}$ set of unscheduled jobs, $j_{k+1}, j_{k+2}, \ldots, j_J$

$c_m^{L,n}$ completion time, at level L, for node n on machine m. It is also the earliest possible start time for the first unscheduled job.

$d_m^{L,n}$ same as above but computed differently

$G_m^{L,n}$ bound on the schedule time, at level L, for node n, on machine m

$G^{L,n}$ lower bound on the schedule time, at level L, for the node n

G^L minimum lower bound on the schedule time at level L

The following lower bounds developed by various investigators may be considered:

1. Machine-based bound reported by Brown and Lomnicki (27), and McMahon and Burton (99) is computed as follows:

At level L, the bound for each node n on machine m is

$$G_m^{L,n} = c_m^{L,n} + \sum_{\substack{x=1 \\ j_x \varepsilon \bar{n}}}^{J} t_{j_x m} + \min_{\substack{j_x \\ j_x \varepsilon \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'} , \quad \ldots\ldots\ldots\ldots \quad (1)$$

$$x = 1, 2, \ldots, J ,$$

$$m = 1, 2, \ldots, M ,$$

and

$$\min_{\substack{j_x \\ j_x \varepsilon \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'} = 0 \quad \text{for } m = M,$$

where for node n which consists of a partial sequence $\{j_1\, j_2\, \ldots\, j_L\}$, the completion time on machine m is

$$c_m^{L,n} = \max\,[c_{m-1}^{L,\, j_1\, j_2\, \ldots\, j_L},\ c_m^{L-1,\, j_1\, j_2\, \ldots\, j_{L-1}}] + t_{j_L}^{m},$$

$$m = 1, 2, \ldots, M$$

where

$$c_{m-1}^{L,\, j_1\, j_2\, \ldots\, j_L} = 0 \quad \text{for } m = 1$$

and

$$c_m^{L-1,\, j_1\, j_2\, \ldots\, j_{L-1}} = 0 \quad \text{for } L = 1$$

Formula (1) states that at level L, the schedule time of each node n, machine m is bounded from below by the sum of three terms: the total processing times of the scheduled jobs on machine m (those jobs which are included in the partial sequence of node n) plus the total processing times of the unscheduled jobs on machine m (those jobs which are not included in the partial sequence of node n) plus the minimum of the total processing times required to perform the unscheduled jobs on machines m+1, m+2, . . ., M. It should be noted that the last term becomes zero when the bound is computed on machine M.

Thus the lower bound at level L and for each node n is such that

$$G^{L,n} = \max_m\,[G_m^{L,n}]$$

2. Machine-based bound reported by Ignall and Schrage (70) is similar to that of Brown and Lomniki (27) except the first term. This bound is computed such that

$$G_m^{L,n} = d_m^{L,n} + \sum_{\substack{x=1 \\ j_x \in \bar{n}}}^{J} t_{j_x m} + \min_{\substack{j_x \\ j_x \in \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'}, \quad \ldots \ldots \ldots \quad (2)$$

$$x = 1, 2, \ldots, J$$

$$m = 1, 2, \ldots, M$$

and

$$\min_{\substack{j_x \\ j_x \in \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'} = 0 \quad \text{for } m = M$$

where for each node n, the completion time is

$$d_m^{L,n} = \max_k \left[c_m^{L,n},\ c_{m-k}^{L,n} + \min_{j_x \in \bar{n}} \left[\sum_{m'=m-k}^{m-1} t_{j_x m'}\right]\right],$$

$$x = 1, 2, \ldots, J,$$

$$m = 1, 2, \ldots, M,$$

$$k = 1, 2, \ldots, m-1,$$

where

$$\min_{j \in \bar{n}} \left[\sum_{m'=m-k}^{m-1} t_{jm'}\right] = 0 \quad \text{for } m = 1$$

Formula (2) states that at level L, the schedule time of each node n, machine m is bounded from below by the sum of three terms: the total processing times of the scheduled jobs on machine m plus the total processing times of the unscheduled jobs on machine m plus the minimum of the total processing times required to perform the unscheduled jobs on machines m+1, m+2, . . ., M. As in Formula (1), the last term becomes zero when the bound is computed on machine M.

Thus the lower bound at level L and for each node is such that

$$G^{L,n} = \max_{m} [G_m^{L,n}].$$

3. Job-based bound reported by McMahon and Burton (99) expresses the fact that the schedule time may be determined by the total processing time for a job, rather than by the total processing time on one machine. These bounds are computed such that

$$G_m^{L,n} = c_m^{L,n} + \max_{j_x \varepsilon \bar{n}} \Big[\sum_{m'=m}^{M} t_{j_x m'} + \sum_{\substack{k=\underline{1} \\ j_k \varepsilon \bar{n} \\ j_k \neq j_x}}^{J} \min (t_{j_k m}, t_{j_k M}) \Big], \quad . \; . \; . \quad (3)$$

$$x = 1, 2, . . ., J,$$

$$m = 1, 2, . . ., M,$$

where

$$\sum_{\substack{k=\underline{1} \\ j_k \varepsilon \bar{n} \\ j_k \neq j_x}}^{J} \min (t_{j_k m}, t_{j_k M}) = 0 \quad \text{for } m = M.$$

Formula (3) states that, at level L, the schedule time of each node n, on machine m, is bounded from below by the sum of the following terms: the total processing times of the scheduled jobs on machine m plus the maximum of the values, each of which is computed for each unscheduled job. Each of these values is expressed as the total processing time for an unscheduled job, say $\hat{j}_x$, on machines m, m+1, . . ., M plus the sum of the minimum of the processing times required to perform the unscheduled jobs excluding $\hat{j}_x$ on machine m and that on machine M. Thus the lower bound, at level L, for each node n is such that

$$G^{L,n} = \max_m \, [G_m^{L,n}].$$

4. Composite-bound reported by McMohon and Burton (99), is defined as the maximum of that of machine-based bound and job-based bound. It has been claimed that the composite-bound is more efficient than either machine-based bound or the job-based bound. The number of nodes to be explored is considerably reduced.

5. A lower bound reported by Ashour (10) is defined as the time required to process all jobs on the last machine without conflict or overlapping of jobs on that machine. This lower bound is a simplified version of that developed by Brooks and White (26). To obtain this lower bound, at level L and for each node n, the completion time $c_{j_k}^{L,n}$ is computed such that

$$c_{j_1}^{L,n} = \sum_{m=1}^{M-1} t_{j_1 m}$$

$$C_{j_k}^{L,n} = \sum_{x=1}^{k-1} t_{j_x 1} + \sum_{m=1}^{M-1} t_{j_k m}, \quad k = 2, 3, \ldots, L.$$

$$= \sum_{x=1}^{L} t_{j_x 1} + \sum_{m=1}^{M-1} t_{j_k m}, \quad k = L+1, L+2, \ldots, J.$$

These J values are then arranged in an ascending order and placed in a vector u such that

$$U = [U_1\ U_2\ \ldots\ U_j\ \ldots\ U_J]$$

consequently, the corresponding processing times on machine M are placed in a vector v such that

$$V = [V_1\ V_2\ \ldots\ V_j\ \ldots\ V_J]$$

Thus, the lower bound $G^{L,n}$ is computed such that

$$G^{L,n} = D_J$$

where

$$D_J = \max\ [D_{J-1}, U_J] + V_J$$

$$D_{J-1} = \max\ [D_{J-2}, U_{J-1}] + V_{J-1}$$

$$\vdots$$

$$D_2 = \max\ [D_1, U_2] + V_2$$

$$D_1 = U_1 + V_1$$

In applying the branch-and-bound technique to the flowshop problem, Brown and Lomnicki (27) have considered arbitrary number of machines. However, Ignall and Schrage (70), and McMohon and Burton (99) have considered problems having up to three machines following theorems 5-1 and 5-2 appeared in Conway et. al. (32, pp. 81).

As mentioned above, the branching is considered from the node which has the minimum lower bound. However, sometimes there is a tie. This tie may be broken by: (1) branching off all the nodes labeled with the minimum lower bound; (2) selecting a node by random to be branched; or (3) branching off at the dominating node.

It should be pointed out that lower bounds never decrease as one moves down the scheduling tree; however, it may or may not increase. Therefore at each level L, except the first one, the minimum lower bound is compared with that of the preceding level. If it is greater, the branching process is taken place from the node, at the preceding level, which has the second minimum lower bound. Although this is in contradiction to the fact that no further branching is done from a node, which has a higher value than the minimum lower bound at first level.

Using the machine-based lower bound reported by Brown and Lomnicki (27) and breaking a tie (if any) by random, the basic steps of the branch-and-bound algorithm are stated below as they are implemented on the computer. However, it should be pointed out that this algorithm is general in the sense that step 2 may be replaced if other lower bound is considered. Step 6.2 may also be replaced according to the procedures considered for breaking a tie (if any).

The algorithm may now be stated as follows:

Step 1: Set L = 1

Step 2: At level L, compute the lower bound for each node n as follows

2.1. compute the completion time $c_m^{L,n}$ on each machine such that

$$c_m^{L,\, j_1 j_2 \cdots j_L} = \max\left[c_{m-1}^{L,\, j_1 j_2 \cdots j_L},\ c_m^{L-1,\, j_1 j_2 \cdots j_{L-1}}\right] + t_{j_L m},$$

$$m = 1, 2, \ldots, M,$$

where

$$c_{m-1}^{L,\, j_1 j_2 \cdots j_L} = 0 \quad \text{for } m = 1,$$

and

$$c_m^{L-1,\, j_1 j_2 \cdots j_{L-1}} = 0 \quad \text{for } L = 1.$$

2.2. Compute the bound, $G_m^{L,n}$ on each machine such that

$$G_m^{L,n} = c_m^{L,n} + \sum_{\substack{j_x = 1 \\ j_x \varepsilon \bar{n}}}^{J} t_{j_x m} + \min_{\substack{j_x \\ j_x \varepsilon \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'},$$

$$x = 1, 2, \ldots, J,$$

$$m = 1, 2, \ldots, M,$$

and

$$\min_{\substack{j_x \\ j_x \epsilon \bar{n}}} \sum_{m'=m+1}^{M} t_{j_x m'} = 0 \quad \text{for } m = M.$$

2.3. Find the lower bound $G^{L,n}$ such that

$$G^{L,n} = \max_{m} [G_m^{L,n}]$$

Step 3: At level L, find the node(s) which has the minimum lower bound G^L such that

$$G^L = \min_{n} [G^{L,n}]$$

Step 4: Check L

4.1. If $L = 1$, go to step 6

4.2. If $L > 1$, go to step 5

Step 5: Check the minimum lower bound

5.1. If $G^L = G^{L-1}$, go to step 6.

5.2. If $G^L > G^{L-1}$, terminate the search in this direction. Branch the node, at level L-1, having the second minimum, go to step 2.

Step 6: Determine the node(s) having the minimum lower bound:

6.1. If a tie does not exist, branch that node, go to step 7.

6.2. If a tie exists, resolve it by random and branch the selected node.

Step 7: Check L

7.1. If $L < J-1$, set $L = L+1$ and go to step 2.

7.2. If $L = J-1$, go to step 8.

Step 8: Select the node which represents the minimum lower bound. This is the schedule time of the sequence shown in this node.

A sample flowshop problem of six jobs and three machines is solved to illustrate the above algorithm. The processing time and machine ordering matrices are given below:

$$T^* = \begin{bmatrix} 4 & 5 & 5 \\ 2 & 17 & 7 \\ 2 & 10 & 4 \\ 10 & 8 & 2 \\ 7 & 15 & 6 \\ 9 & 4 & 11 \end{bmatrix} \qquad M^* = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \\ 51 & 52 & 53 \\ 61 & 62 & 63 \end{bmatrix}$$

The computation is carried out step-by-step to illustrate the branch-and-bound algorithm.

At level 1 and for the node n which consists of $\{j_1\}$ or $\{1\}$, the completion time on each machine is computed such that

$$c_1^{1,1} = \max\,[c_0^{1,1},\ c_1^{0,0}] + t_{11},$$

$$c_2^{1,1} = \max\,[c_1^{1,1},\ c_2^{0,0}] + t_{12},$$

and

$$c_3^{1,1} = \max\ [c_2^{1,1},\ c_3^{0,0}] + t_{13},$$

or

$$c_1^{1,1} = \max\ [0\ ,\ 0] + 4 = 4,$$

$$c_2^{1,1} = \max\ [4,\ 0] + 5 = 9,$$

and

$$c_3^{1,1} = \max\ [9,\ 0] + 5 = 14.$$

The bound for the above node is then computed on each machine such that

$$G_1^{1,1} = c_1^{1,1} + \sum_{j=2}^{6} t_{j1} + \min_{\substack{j \\ j\neq 1}} \sum_{m=2}^{3} t_{jm},$$

$$G_2^{1,1} = c_2^{1,1} + \sum_{j=2}^{6} t_{j2} + \min_{\substack{j \\ j\neq 1}} t_{j3},$$

and

$$G_3^{1,1} = c_3^{1,1} + \sum_{j=2}^{6} t_{j3},$$

or

$$G_1^{1,1} = 4 + (2 + 2 + 10 + 7 + 9) + \min [24, 14, 10, 21, 15]$$

$$= 4 + 30 + 10$$

$$= 44,$$

$$G_2^{1,1} = 9 + (17 + 10 + 8 + 15 + 4) + \min [7, 4, 2, 6, 11]$$

$$= 9 + 54 + 2$$

$$= 65,$$

and

$$G_3^{1,1} = 14 + (7 + 4 + 2 + 6 + 11)$$

$$= 14 + 30$$

$$= 44$$

Thus, the lower bound for this node is

$$G^{1,1} = \max [44, 65, 44]$$

$$= 65$$

Similarily, the lower bounds, at level 1, for the nodes having the partial sequences {2}, {3}, {4}, {5} and {6} are found to be 63, 63, 73, 68, and 70, respectively, see Table 2.7.

Following step 3, the minimum lower bound at this level is such that

$$G^1 = \min_n [G^{1,n}]$$

Table 2.7
Table of Lower bounds

Level	Node Including Partial Sequence	First Term 1	2	3	Second Term 1	2	3	Third Term 1	2	3	Bounds on Machine $G_1^{L,n}$	$G_2^{L,n}$	$G_3^{L,n}$	lower Bound $G^{L,n}$
1	1	4	9	14	30	54	30	10	2	0	44	65	44	65
	2	2	19	26	32	42	28	10	2	0	44	63	54	63*
	3	2	12	16	32	49	31	10	2	0	44	63	47	63
	4	10	18	20	24	51	33	10	4	0	44	73	53	73
	5	7	22	28	27	44	29	10	2	0	44	68	57	68
	6	9	13	24	25	55	24	10	2	0	44	70	48	70
2	21	6	24	31	28	37	23	10	2	0	44	63	54	63
	23	4	29	33	30	32	24	10	2	0	44	63	57	63*
	24	12	27	29	22	34	26	10	4	0	44	65	55	65
	25	9	34	40	25	27	22	10	2	0	44	63	62	63
	26	11	23	37	23	38	17	10	2	0	44	63	54	63
3	231	8	34	39	26	27	19	10	2	0	44	63	58	63*
	234	14	37	39	20	24	22	10	5	0	44	66	61	66
	235	11	44	50	23	17	18	10	2	0	44	63	68	68
	236	13	33	44	21	28	13	10	2	0	44	63	57	63*
4	2314	18	42	44	16	19	17	15	6	0	49	67	61	67
	2315	15	49	55	19	12	13	10	2	0	44	63	68	68
	2316	17	38	50	17	23	8	10	2	0	44	63	58	63*
5	23164	27	46	52	7	15	6	21	6	0	55	67	58	67
	23165	24	53	59	10	8	2	10	2	0	44	63	61	63*
6	231654	34	61	63	0	0	0	0	0	0	34	61	63	63

*indicates the node at which branching is done to obtain the optimal sequence.

or

$$G^1 = \min [65, 63, 63, 73, 68, 70]$$

$$= 63.$$

Since a tie exists between the nodes having the partial sequence {2} and {3}, the {2} is selected by random. Consequently, the selected node is branched at level 2 to the nodes having the partial sequences {31}, {32}, {34}, {35} and {36}.

At level 2 and for the node n which consists of $\{j_1\ j_2\}$ or {21}, the completion time on each machine is computed such that

$$c_1^{2,21} = \max [c_0^{2,21}, c_1^{1,2}] + t_{11},$$

$$c_2^{2,21} = \max [c_1^{2,21}, c_2^{1,2}] + t_{12},$$

and

$$c_3^{2,21} = \max [c_2^{2,21}, c_3^{1,2}] + t_{13},$$

or

$$c_1^{2,21} = \max [0, 2] + 4 = 6,$$

$$c_2^{2,21} = \max [6, 19] + 5 = 24,$$

and

$$c_3^{2,21} = \max\ [24,\ 2,6] + 5 = 31.$$

The bounds for the above node is then computed on each machine such that

$$G_1^{2,21} = c_1^{2,21} + \sum_{\substack{j=1 \\ j\neq 1,2}}^{6} t_{j1} + \min_{\substack{j \\ j\neq 1,2}} \sum_{m=2}^{3} t_{jm},$$

$$G_2^{2,21} = c_2^{2,21} + \sum_{\substack{j=1 \\ j\neq 1,2}}^{6} t_{j2} + \min_{\substack{j \\ j\neq 1,2}} t_{j3},$$

and

$$G_3^{2,21} = c_3^{2,21} + \sum_{\substack{j=1 \\ j\neq 1,2}}^{6} t_{j3},$$

or

$$G_1^{2,\ 21} = 6 + (2 + 10 + 7 + 9) + \min\ [14,\ 10,\ 21,\ 15]$$

$$= 6 + 28 + 10$$

$$= 44,$$

$$G_2^{2,21} = 24 + (10 + 8 + 15 + 4) + \min\ [4,\ 2,\ 6,\ 11]$$

$$= 24 + 37 + 2$$

$$= 63,$$

and

$$G_3^{2,21} = 31 + (4 + 2 + 6 + 11)$$

$$= 31 + 23$$

$$= 54.$$

Thus the lower bound for this node is

$$G^{2,21} = \max\ [44,\ 63,\ 54]$$

$$= 63$$

Similarily, the lower bounds, at level 2, for the nodes having the partial sequence {23}, {24}, {25} and {26} are found to be 63, 65, 63 and 63, respectively, see Table 2.7.

Following step 3, the minimum lower bound at this level is such that

$$G^2 = \min_n\ [G^{2,n}]$$

$$= \min\ [63,\ 63,\ 65,\ 63,\ 63]$$

$$= 63.$$

Since a tie exists among the nodes having the partial sequence {21}, {23}, {25} and {26}, the node {23} is randomly selected. Consequently, the selected node is branched at level 3 to the nodes having the partial sequence {231}, {234}, {235} and {236}.

At level 3 and for the node n which consists of $\{j_1\ j_2\ j_3\}$ or {231}, the completion time on each machine is computed such that

$$c_1^{3,\ 231} = \max\ [c_0^{3,231},\ c_1^{2,23}] + t_{11},$$

$$c_2^{3,\ 231} = \max\ [c_1^{3,231},\ c_2^{2,31}] + t_{12},$$

and

$$c_3^{3,\ 231} = \max\ [c_2^{3,231},\ c_3^{2,23}] + t_{13},$$

or

$$c_1^{2,\ 231} = \max\ [0,\ 4] + 4 = 8,$$

$$c_2^{3,\ 231} = \max\ [8,\ 29] + 5 = 34,$$

and

$$c_3^{3,\ 231} = \max\ [34,\ 33] + 5 = 39.$$

The bounds for the above node is then computed for each machine such that

$$G_1^{3,231} = c_1^{3,231} + \sum_{\substack{j=1 \\ j \neq 1,2,3}}^{6} t_{j1} \pm \min_{\substack{j \\ j \neq 1,2,3}} \sum_{m=2}^{3} t_{jm},$$

$$G_2^{3,231} = c_2^{3,231} + \sum_{\substack{j=1 \\ j \neq 1,2,3}}^{6} t_{j2} + \min_{\substack{j \\ j \neq 1,2,3}} t_{j3},$$

and

$$G_3^{3,231} = C_2^{3,361} + \sum_{\substack{j=1 \\ j\neq 1,2,3}}^{6} t_{j3},$$

or

$$G_1^{3,231} = 8 + (10 + 7 + 9) + \min (10, 21, 15)$$

$$= 8 + 26 + 10$$

$$= 44,$$

$$G_2^{3,231} = 34 + (8 + 15 + 4) + \min (2, 6, 11)$$

$$= 34 + 27 + 2$$

$$= 63,$$

and

$$G_3^{3,231} = 39 + (2 + 6 + 11)$$

$$= 39 + 19$$

$$= 58.$$

Thus the lower bound for this node is

$$G^{3,231} = \max [44, 63, 58]$$

$$= 63$$

Similarily, the lower bounds, at level 3, for the nodes having the partial sequences {234}, {235} and {236} are found to be 66, 68 and 63 respectively; see Table 2.7.

The minimum lower bound at this level is such that

$$G^3 = \min_n \,[G^{3,n}]$$
$$= \min\,[63,\ 66,\ 68,\ 63]$$
$$= 63.$$

Since a tie exists between the nodes having the partial sequences {231} and {236} select {231} randomly and branch it, at level 4, to the nodes having the partial sequences {2314}, {2315} and {2316}.

At level 4 and for the node n which consists of $\{j_1\ j_2\ j_3\ j_4\}$ or {2 3 14}, the completion time on each machine is computed such that

$$c_1^{4,2314} = \max\,[c_0^{4,2314},\ c_1^{3,\ 231}] + t_{41},$$

$$c_2^{4,2314} = \max\,[c_1^{4,2314},\ c_2^{3,231}] + t_{42},$$

and

$$c_3^{4,2314} = \max\,[c_2^{4,2314},\ c_3^{3,231}] + t_{43},$$

or

$$G_2^{4,2314} = 42 + (15 + 4) + \min\,[6,\ 11]$$

$$= 42 + 19 + 6$$

$$= 67,$$

and

$$G_3^{4,2314} = 44 + (6 + 11)$$

$$= 61.$$

Thus the lower bound for this node is

$$G^{4,2314} = \max\,[49,\ 67,\ 61]$$

$$= 67.$$

Similarily, the lower bounds, at level 4, for the nodes having the partial sequences {2315} and {2316} are found to be 68 and 63 respectively, see Table 2.7. The minimum lower bound at this level is

$$G^4 = \min_n\,[G^{4,n}]$$

$$= \min\,[67,\ 68,\ 63]$$

$$= 63.$$

Since, the node having the partial sequence {2316} has the minimum lower bound; therefore, it is branched at level 5 to the nodes having partial sequences {23164}, and {23165}.

$$c_1^{4,2314} = \max\,[0,\ 8] + 10 = 18,$$

$$c_2^{4,\ 2314} = \max\,[18,\ 34] + 8 = 42,$$

and

$$c_3^{4,2314} = \max\,[42,\ 39] + 2 = 44.$$

The bounds for the above node is then computed on each machine such that

$$G_1^{4,2314} = c_1^{4,2314} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4}}^{6} t_{j1} + \min_{\substack{j \\ j\neq 1,2,3,4}} \sum_{m=2}^{3} t_{jm},$$

$$G_2^{4,2314} = c_2^{4,2314} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4}}^{6} t_{j2} + \min_{\substack{j \\ j\neq 1,2,3,4}} t_{j3}$$

and

$$G_3^{4,2314} = c_3^{4,2314} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4}}^{6} t_{j3},$$

or

$$\begin{aligned} G_1^{4,2314} &= 18 + (7 + 9) + \min\,[21,\ 15] \\ &= 18 + 16 + 15 \\ &= 49, \end{aligned}$$

At level 5 and for the node n which consists of $\{j_1\ j_2\ j_3\ j_4\ j_5\}$ or {23164}, the completion time on each machine is computed such that

$$c_1^{5,23164} = \max\ [c_0^{5,23164},\ c_1^{4,2316}] + t_{41},$$

$$c_2^{5,23164} = \max\ [c_1^{5,23164},\ c_2^{4,2316}] + t_{42},$$

and

$$c_3^{5,23164} = \max\ [c_2^{5,23164},\ c_3^{4,2316}] + t_{43}.$$

or

$$c_1^{5,23164} = \max\ [0,\ 17] + 10 = 27,$$

$$c_2^{5,23164} = \max\ [27,\ 38] + 8 = 46,$$

and

$$c_3^{5,23164} = \max\ [46,\ 50] + 2 = 52.$$

The bounds for the above node is then computed on each machine such that

$$G_1^{5,23164} = c_1^{5,23164} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4,6}}^{6} t_{j1} + \min_{\substack{j \\ j\neq 1,2,3,4,6}} \sum_{m=2}^{3} t_{jm},$$

$$G_2^{5,23164} = c_2^{5,23164} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4,6}}^{6} t_{j2} + \min_{\substack{j \\ j\neq 1,2,3,4,6}} t_{j3},$$

and

$$G_3^{5,23164} = c_3^{5,23164} + \sum_{\substack{j=1 \\ j\neq 1,2,3,4,6}}^{6} t_{j3}$$

or

$$G_1^{5,23164} = 27 + 7 + 21 = 55,$$

$$G_2^{5,23164} = 46 + 15 + 6 = 67,$$

and

$$G_3^{5,23164} = 52 + 6 = 58.$$

Thus the lower bound for this node is

$$G^{5,36251} = \max\ [55,\ 67,\ 58]$$

$$= 67$$

Similarily, the lower bound at level 5 for the other node having the partial sequence {23165} is 63. The minimum lower bound at this level is

$$G^5 = \min_n\ [G^{5,n}]$$

= min [67, 63]

= 63

and the node is {2 3 1 6 5}. The unscheduled job for this node is job 4. Thus the feasible sequence is {2 3 1 6 5 4} with a schedule time of 63. It is of interest to note that this problem has been solved by complete enumeration and direct technique and the optimal solution is 63.

CHAPTER III

COMPUTATIONAL EXPERIMENTS

This chapter deals with the computational experiments conducted to establish a fair comparison among Complete Enumeration, Direct method and Branch-and-Bound method.

Experiment I consists of 100 flowshop problems each consisting of six jobs and three machines. Experiment II consists of ten flowshop problems, each of which has eight jobs and three machines. Experiment III consists of ten flowshop problems. Each of these problems has six jobs and five machines. Experiment IV consists of one problem having 12 jobs and three machines.

For comparison, several statistics such as the efficiency of each technique, the frequency of the minimum schedule time and the execution time spent by the computer to obtain the various solutions as computed.

3.1 Experiment I

This experiment was designed to compare results with those obtained from a similar experiment by Decomposition (6) and Rounded linear programming (57). The experiment consists of 100 Flowshop problems, each consisting of six jobs and three machines.

The entries of the processing time matrices were generated uniformly at random between one and 30, inclusive. For each problem, the solution was obtained by enumerating all the 6! or 720 sequences, using the subroutine ENUMER and the execution time spent to obtain the solution was recorded by the computer. Out of these 720 sequences, the minimum schedule time was printed out with the corresponding sequence(s). Furthermore, by calling the STAT subroutine, various statistics were computed. Similarly, all the problems were solved by the Direct method and the Branch-and-bound technique according to the algorithms discussed in section 2.2 and 2.3, respectively. Various statistics similar to those obtained from the complete enumeration solution were computed for the Direct Method only, since one sequence was generated by the Branch-and-Bound technique.

The objective was to observe the statistical distribution of the minimum schedule times and the mean schedule times obtained from the 100 problems by different approaches. Thus in the case of complete enumeration, Direct method and Branch-and-bound method, the frequencies were tabulated as shown in Table A.1. The observed relative cumulative frequencies, appeared in Table A.2, were then plotted on normal probability paper as shown in Figure 3.1. The plot of the Complete Enumeration and Direct method are the same, since they have the same optimal values.

In this report, the Branch-and-Bound algorithm in which all nodes at all levels are generated is referred to as Branch-and-Bound I; and that in which only one node at each level is generated is referred to as Branch-and-Bound II.

The sample average and the standard deviation of the optimal schedule times and the mean schedule times by various approaches appear in Table 3.1. Upon examining this table, the mean and standard deviation computed for the minimum and mean schedule times obtained by these investigators are approximately the same. Therefore, it would be somewhat safe to compare the other approaches to each other. Furthermore, the Direct method and Branch-and-Bound I generate optimal. It should be noted that the later

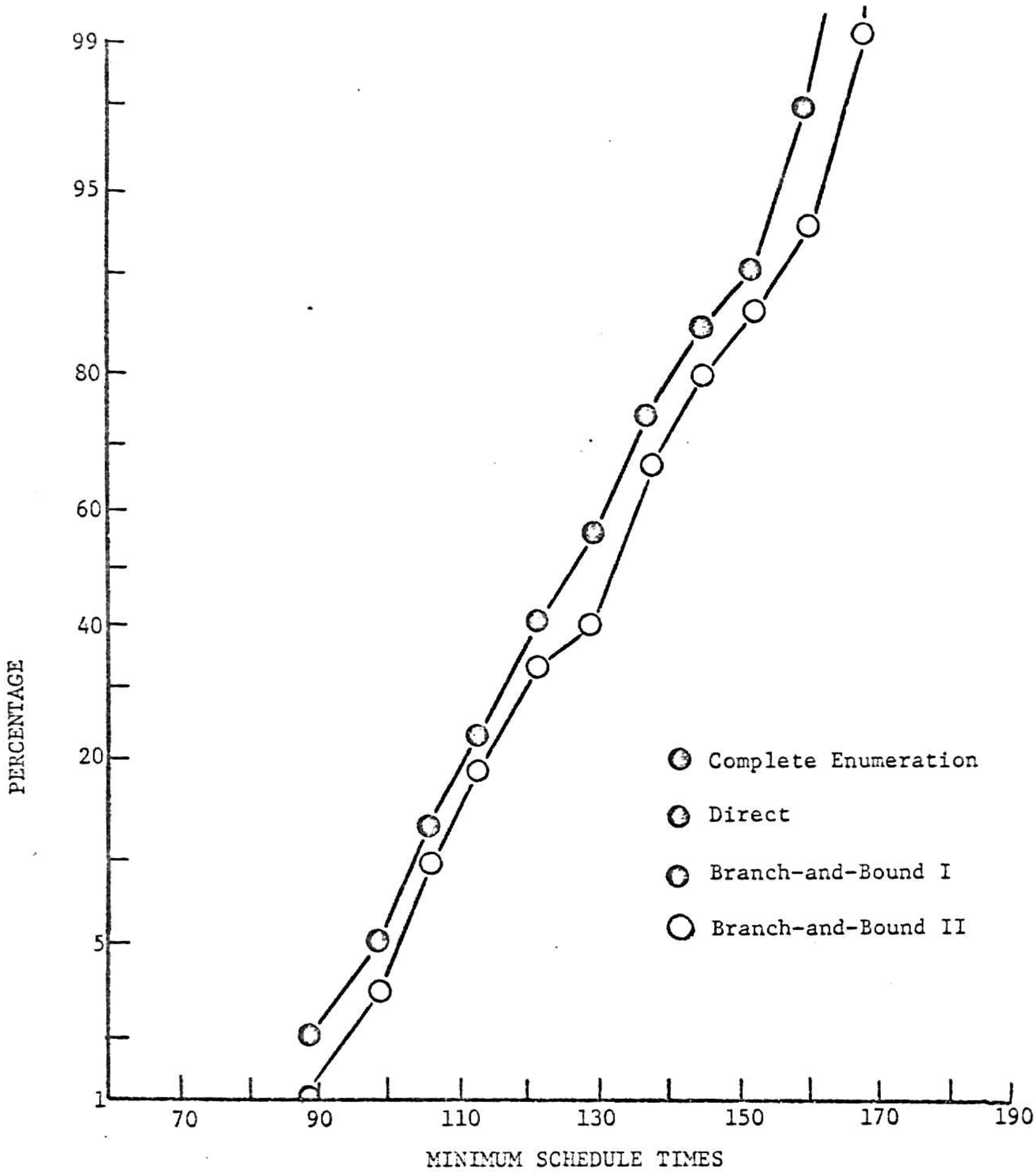

Fig. 3.1. Relative Cumulative Frequencies of Minimum Schedule Times Obtained by Different Approaches

generate all nodes regardless to the lower bound computed. In this experiment, these approaches are ranked in superiority from the point of view of the optimal schedule time as follows: Direct, Branch-and-Bound I, Decomposition, Branch-and-Bound II and Rounded Linear Programming.

For further comparison, the efficiencies of various approaches are computed. The results of these efficiencies are tabulated in Tables 3.2, 3.3. This shows that the number of problems of which the optimal values obtained by Direct, Branch-and-Bound I and II, Decomposition and Rounded Linear Programming are 100, 100, 45, 43 and 8 respectively. However, the number of those having efficiencies greater than or equal to 0.95 is 100, 100, 73, 92 and 27.

The general results observed from experiment I tabulated in various tables are:

1. The subset of the optimal sequences is very small in most of the problems. In 50 percent of the problems solved, the number of optimal sequences is less than or equal to 5. Furthermore, as the number of optimal sequences in each problem increases, the number of problems decreases exponentially, see Table 3.4. This decreases the probability of obtaining the optimal sequences by the Branch-and-Bound algorithm as programmed in this report.

2. The combinatorial techniques reduce the problem of constructing and evaluating feasible sequences to a limited number of sequences. For example, in the sample problem solved in chapter 2, the number of sequences generated and evaluated by complete Enumeration, Direct, Decomposition and Branch-and-Bound II is 720, 12, 20 and 1, respectively. See Table 3.6.

Table 3.1

Comparison of Minimum and Mean Schedule Times Obtained by Different Approaches.

		This Report			Ashour (6)		Giglio & Wagner (57)		Brown & Lomnicki(27)	
	Statistics	Complete Enumeration	Direct* Method	Branch-and-Bound II	Complete Enumeration	Decomposition K=3	Complete Enumeration	Rounded Linear Programming	Complete Enumeration	Branch-* and-Bound I
Minimum Schedule Time	Sample Average	126.58	126.58	131.08	127.14	129.54	125.95	142.06	127.17	127.17
	Standard Deviation	18.38	18.38	18.85	17.02	16.76	18.46	20.18	17.46	17.46
Mean Schedule Time	Sample Average	149.70	134.97	**	151.22	143.98	148.80	**	**	**
	Standard Deviation	18.59	19.24	**	16.97	16.91	18.12	**	**	**

*Optimal Schedule times are obtained for the 100 problem.

**These values are not available.

Table 3.2

Efficiencies of Different Approaches in Experiment I

Efficiency	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K = 3
1.00	100	100	48	43
0.99			8	19
0.98			7	9
0.97			4	10
0.96			7	6
0.95			5	5
0.94			4	2
0.93			4	1
0.92			3	1
0.91			2	1
0.90			2	1
0.89			2	
0.88			2	1
0.87			1	
0.86			1	
0.85			1	1
0.84				
0.83				
0.82			1	
0.81				
0.80				
0.79				
	100	100	100	100

3. The number of distinct schedule times in the scheduling problem is much less than the number of sequences generated. For example, Table 3.6 shows that the number of distinct schedule times obtained by complete enumeration, Direct, Decomposition and Branch-and-Bound II is 21, 8, 10 and 1 respectively.

Table 3.3

Efficiencies and Corresponding Number of Problems
Solved by Different Approaches

Efficiency	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K=3	Rounded Linear Programming
1.00	100	100	45	43	8
0.95			31*	49	19
0.90			15	6	22
0.85			7	2	26
0.80			1		11
0.75			1		8
0.70					4
0.65					1
0.60					1
	100	100	100	100	100

*Illustration: The number of problems solved by Branch-and-Bound method having efficiency such that $0.95 \leq$ efficiency < 1.00 is 31.

4. The distinct schedule times obtained by the various combinatorial methods have lower frequencies of occurances than those from complete enumeration. Therefore, these procedures reduce the redundancies of the sequences. See Table 3.6 as an example.

5. As in all problems solved, Table 3.6 shows that these combinatorial techniques produces schedules times very close to the optimal schedule time.

Table 3.4

Frequencies of Optimal Sequences Obtained By Complete Enumeration Technique and Direct Method

No. of Problems	Range of the no. of Optimal Sequences Complete Enumeration	Direct
50	1 - 5	1 - 3
16	6 - 12	1 - 4
8	13 - 19	2 - 3
8	20 - 25	2 - 6
6	26 - 41	3 - 6
5	43 - 60	1 - 7
6	68 - 110	3 - 10
1	120	8

Table 3.5

Frequencies of Generated Sequences Obtained by Direct Method

No. of Problems	Range of the no. of Generated Sequences	Range of the no. of Optimal Sequences
28	3 - 10	1 - 10
49	11 - 20	1 - 8
16	21 - 27	1 - 7
4	31 - 37	1 - 4
3	40 - 47	1 - 9

Table 3.6

Frequency Table of Distinct Schedule Times Obtained by Different Approaches

Distinct Schedule	Complete Enumeration	Direct Method	Decomposition K=3	Branch-and-Bound II
63	28	3	1	1
64	2	1		
64	26	1	1	
66	19	2		
67	62	2	3	
68	52	1	2	
69	26	1	4	
70	44		2	
71	26	1		
72	54		4	
73	67		1	
74	71		1	
75	56			
76	51			
77	37		1	
78	19			
79	13			
80	33			
81	12			
82	12			
83	10			
	720	12	20	1

6. The number of optimal sequences generated by the Direct algorithm increases as the subset of the optimal sequences become larger, See Table A.3.

7. The Direct algorithm and the Branch-and-Bound algorithm II generates optimal solution for all the problems. See Table 3.3.

8. The number of sequences generated by the Direct algorithm is independent of the size of the subset of optimal sequences. See Table 3.7.

9. The efficiency of the Branch-and-Bound II method increases as the subset of the optimal sequences becomes larger. See Table 3.7.

10. Since the branching in the Branch-and-Bound technique II, is

Table 3.7

Average Values of Computational Results of Experiment I

Number of problems	Complete Enumeration: No. of optimal sequences	Complete Enumeration: Time in seconds	Complete Enumeration: No. of sequences generated	Direct: No. of optimal sequences	Direct: Time in seconds	Branch-and-Bound II: Efficiency Range	Branch-and-Bound II: Efficiency Average	Branch-and-Bound II: Time in seconds
13	1	55.37	21.84	1.00	2.81	0.87-1.00	0.91	2.29
12	2	55.79	23.00	1.20	2.18	0.82-1.00	0.93	2.30
9	3	55.42	15.66	1.33	1.06	0.92-1.00	0.97	2.35
14	4	55.92	14.79	1.55	1.66	0.90-1.00	0.96	2.12
2	5	54.73	14.00	3.00	1.31	0.93-0.94	0.93	2.25
5	6	56.20	13.80	2.60	1.67	0.98-1.00	0.99	2.06
2	7	54.93	13.50	2.00	1.81	0.96-1.00	0.98	2.67
2	8	56.36	10.00	2.50	0.79	0.96-1.00	0.98	2.24
2	10	56.06	13.50	2.00	1.50	0.90-1.00	0.95	1.33
5	12	56.23	16.20	3.40	2.62	0.95-1.00	0.99	1.85
2	13	54.95	14.00	2.50	1.37		1.00	1.91
2	14	57.29	11.00	3.50	0.86	0.96-1.00	0.98	1.94
2	16	55.20	13.50	2.50	1.80	0.92-0.95	0.93	2.69
1	18	55.03	5.00	3.00	0.28		1.00	1.52
1	19	56.56	15.00	3.00	2.35		0.99	1.89
1	20	58.12	12.00	3.00	1.43		1.00	2.13
2	22	59.87	11.00	3.50	1.30	0.98-0.99	0.98	2.49
1	23	58.06	10.00	6.00	0.85		1.00	1.61
3	24	60.71	16.30	3.33	2.12	0.96-1.00	0.98	2.41
1	25	57.69	7.00	4.00	0.78		1.00	2.50
1	26	55.51	25.00	6.00	2.83		1.00	2.57
2	28	56.78	15.00	3.50	1.25		1.00	1.84
1	32	55.96	15.00	5.00	1.37		1.00	1.51
1	34	55.78	12.00	7.00	1.64		1.00	1.97
1	41	61.14	8.00	4.00	0.51		1.00	2.51

Table 3.7 (continued)

Number of problems	Complete Enumeration		Direct			Branch-and-Bound		
	No. of optimal sequences	Time in seconds	No. of sequences generated	No. of optimal sequences	Time in seconds	Efficiency		Time in seconds
						Range	Average	
1	43	58.06	21.00	7.00	2.56		1.00	1.62
1	44	58.50	14.00	7.00	1.62		1.00	1.65
1	48	58.37	14.00	7.00	2.68		1.00	1.60
1	54	56.25	14.00	4.00	1.20		1.00	1.52
1	60	56.38	5.00	1.00	0.61		1.00	1.54
1	68	56.75	11.00	8.00	1.83		0.89	3.25
1	72	57.12	40.00	9.00	8.17		1.00	1.51
2	80	61.68	9.00	4.56	0.85		1.00	1.85
1	87	57.66	5.00	3.00	6.40		0.88	2.66
1	110	61.32	10.00	10.00	1.08		1.00	1.99
1	120	61.11	17.00	8.00	1.81		0.97	2.43

done from only one node at each level, the execution time spent to obtain the solution remains nearly the same. See Table 3.7.

3.2 Experiment II

This experiment included ten problems taken from Ashour (6). Each problem has eight jobs and three machines. The processing times were generated at random from a uniform distribution between 1 and 30, inclusive.

The solution of these ten problems were obtained by partial enumeration. In sampling with replacement, which is more convenient on the computer, the probability of obtaining a minimum schedule time in a sample of size n is

$$1 - (1 - P)^n$$

with a selection of p as 0.001 and a probability of 0.95 of obtaining the minimum schedule time, the sample size, n is about 3000 sequences. The partial enumeration was done by ENUMER and the execution time spent to obtain the solution was recorded. Out of the 3000 sequences, the minimum schedule time was printed out with the corresponding sequence(s). Similarly, the above problems were solved by the Direct method and the Branch-and-Bound II method. The result obtained by the above technique and Decomposition are summarized in Table 3.8. For comparison among these procedures and the Decomposition approach, Table 3.9 shows the minimum schedule times obtained by the various methods for all the problems. The values between parentheses show the efficiencies of the corresponding minimum schedule time which are not optimal. Thus the number of problems for which the optimal solutions were obtained are 10, 3, 9 and 8 for

Table 3.8

Results Obtained from Experiment II

Problem No.		Partial Enumeration	Direct Method	Branch-and-Bound Method	Decomposition with K=4
1	Number of Sequences generated	3000	26	1	70
	Number of Distinct Schedule Times	69	14	1	32
	Number of Optimal Sequences	9	3	1	7
	Execution Time*	339	7.82	5.31	
2	Number of Sequences Generated	3000	37	1	70
	Number of Distinct Schedule Times	58	13	1	24
	Number of Optimal Sequences	3	8	1	2
	Execution Time*	339	30.55	5.28	
3	Number of Sequences Generated	3000	67	1	70
	Number of Distinct Schedule Times	33	8	1	17
	Number of Optimal Sequences	24	22	1	8
	Execution Time*	339.00	44.48	5.95	
4	Number of Sequences Generated	3000	47	1	70
	Number of Distinct Schedule Times	52	18	1	24
	Number of Optimal Sequences	3	2		3
	Execution Time	339.00	60.37	5.25	
5	Number of Sequences Generated	3000	15	1	70
	Number of Distinct Schedule Times	79	13	1	30
	Number of Optimal Sequences	3	2		
	Execution Time *	339.00	4.56	8.02	
6	Number of Sequences Generated	3000	15	1	70
	Number of Distinct Schedule Times	60	8	1	20
	Number of Optimal Sequences	30	1		9
	Execution Time*	339.00	3.99	5.56	
7	Number of Sequences Generated	3000	28	1	70
	Number of Distinct Schedule Times	46	13	1	15
	Number of Optimal Sequences	27	3	1	18
	Execution Time*	339.00	30.28	4.86	
8	Number of Sequences Generated	3000	12	1	70
	Number of Distinct Schedule Times	50	4	1	33
	Number of Optimal Sequences		5		2
	Execution Time*	339.00	2.28	5.41	
9	Number of Sequences Generated	3000	23	1	70
	Number of Distinct Schedule Times	76	10	1	39
	Number of Optimal Sequences		1		9
	Execution Time	339.00	5.33	6.93	
10	Number of Sequences Generated	3000	58	1	70
	Number of Distinct Schedule Times	53	18	1	23
	Number of Optimal Sequences	30	7		20
	Execution Time*	339.00	91.79	5.93	

* Execution Time in seconds

Table 3.9

The Schedule Time Solutions and their Efficiencies Obtained by Different Techniques in Experiment II

Problem No.	Direct Method	Branch-and-Bound II	Decomposition With K=3	Partial Enumeration
1	157	157*	157	157
2	130	138 (0.95)	130	130
3	195	195	195	195
4	145	148 (0.98)	145	145
5	157	162 (0.97)	159 (0.99)	159
6	160	174 (0.92)	160	160
7	173	173	173	173
8	191	196 (0.98)	191	196 (0.98)
9	122	136 (0.96)	122	126 (0.98)
10	172	185 (0.93)	172	172

*Those values without efficiencies indicated between parantheses are optimal

Table 3.10

Results Obtained From Experiment III

Problem	Complete Enumeration			Direct	Branch-and-Bound II		
	No. of Optimal Sequences	Execution Time*	No. of sequences Generated	No. of Optimal Sequences	Execution Time *	Efficiency	Execution Time*
1	2	96.70	49	2	7.18	0.90	2.07
2	2	96.31	52	1	8.30	0.93	2.30
3	22	97.61	43	7	8.99	0.80	2.12
4	1	96.86	50	1	10.67	0.98	1.75
5	2	96.85	58	1	8.23	0.97	1.32
6	8	96.84	23	1	2.86	1.00	1.26
7	22	97.33	15	1	2.83	0.92	1.81
8	4	96.41	67	2	16.30	0.93	2.46
9	12	96.33	31	3	5.51	1.00	1.01
10	3	95.84	58	5	13.43	0.99	1.09

*Execution Time in Seconds

Direct, Branch-and-Bound II, Decomposition and Partial Enumeration, respectively. The execution time spent to obtain these solutions are 28.14, 5.85, 50.12, 339.00 seconds, respectively. In general, the results obtained from Experiment II are as follows:

(1) In the case of Direct method, the optimal sequence is obtained for all the problems, but the execution time spent is considerably increased.

(2) It is observed that in some problems the execution time is increased regardless to the number of the sequences generated by the Direct algorithm. The obvious reason is that considerable number of pre-sequences are checked for dominance. For example, in problems 3, 4 and 10, the number of sequences generated are 67, 47 and 58, respectively and the corresponding execution time is 44.48, 60.37 and 91.79 seconds respectively.

(3) The execution time spent by the Branch-and-Bound method II is considerably low, but it generated optimal sequences in only three problems out of ten. The solutions of the other seven problems have efficiencies ranged from 0.92 to 0.98.

(4) The Decomposition technique produces nine optimal schedule times out of ten. The solution of problem 5 has efficiency 0.99.

3.3 Experiment III

In order to investigate the effects of the changes in the number of machines, Experiment III has been conducted. This experiment consists of ten flowshop problems, each having six jobs and five machines. The processing times were generated at random from a uniform distribution between 1 and 30, inclusive. This experiment has been conducted in the same manner as Experiment I. The results are tabulated in Table 3.10.

The general results observed from Experiment III are:

1. The subset of the optimal sequences is very small in most of the problems. In six problems out of 10, the number of optimal sequences is less than or equal to 4. This reduces the probability of obtaining the optimal solution by the Branch-and-Bound algorithm II. On examining Table 3.10, it is clear that Branch-and-Bound II generated optimal solution in only two problems.

2. The Direct technique generated optimal sequences for all problems.

3. The exectuion time spent by the Direct method has high variability because of the different number of sequences generated.

4. Since the branching process in the Branch-and-Bound technique II is done from only one node at each level, the execution time spent to obtain the solution remains nearly the same. See Table 3.10.

3.4 Experiment IV

Further experiment has been conducted with 12 jobs and three machines. One problem is solved by Partial Enumeration, Direct and Branch-and-Bound II. The execution time spent by the above techniques are found to be 594.81, 311.81 and 38.59 seconds and the efficiencies are 1.00, 1.00 and 0.96, respectively. These results are summarized in Table 3.11.

Table 3.11

Efficiency and Execution Time of Experiment IV

	Complete Enumeration	Direct	Branch-and-Bound II
Efficiency	1.00	1.00	0.96
Execution Time*	594.81	311.81	38.59

*Execution time in seconds.

Another problem of the same size was generated by random from a uniform distribution between 1 and 30, inclusively, the Direct method could not generate the solution because of excessive dimension requirement. This is because the number of pre-sequences increases very rapidly through the process of filling the sequence-positions.

CHAPTER IV

RESULTS AND CONCLUSIONS

The basic objective of this report is to compare various algorithms for the solution of flowshop scheduling problems in which the minimum schedule time is considered as a criterion. The various algorithms considered are Complete or Partial Enumeration, Direct, and Branch-and-Bound II. The results obtained from these techniques are compared with those obtained by Decomposition (6), Branch-and-Bound I (27); and Rounded Linear Programming (57).

In order to study the merits of these techniques, several computational experiments have been performed and various results such as the efficiencies, the statistical characteristics, and the execution time spent to obtain solutions have been compared.

The most significant results of the computational experiments are summarized as follows:

1. From all experiments, it is revealed that the subset of the optimal sequences is very small with respect to the set of feasible sequences. Furthermore, the number of the distinct schedule times is very small compared to the number of sequences generated and evaluated.
2. The feasible sequences generated and evaluated by Direct, Branch-and-Bound I, Branch-and-Bound II, Decomposition and K equaling J/2 appeared to be closer to the minimum schedule times. In other words, generally speaking, the feasible sequences which have larger schedule times, i.e. close to the maximum schedule time, are discarded. For example, see Table 3.6.

3. The solutions obtained in the experiments conducted in this report and obtained by Direct and Branch-and-Bound I are optimal. However, not all the solutions obtained by Branch-and-Bound II, Decomposition, and Rounded Linear Programming are optimal. The number of optimal solution varies with the various techniques employed.

4. In analyzing the procedure of obtaining the solutions by Direct method, the effects of the change in the number of machines and jobs are as follow:

4.1. The number of conditions computed to be used in job and sequence dominance increases as the number of machines increases. However, the number of partial sequences may or may not increase as the number of machine increases. For example, the average execution time for solving a problem of six jobs and three machines in Experiment I was 1.92 seconds. However, the computer spent on the average 8.43 seconds for solving a problem of six jobs and five machines in Experiment III. In other words, the execution time spent increases approximately five times as the number of machines increases from three to five with the same number of jobs. See Tables 4.2, ~~and~~ 4.6. and 4.10.

4.2. The numbers of job dominance check and sequence dominance check increase as the number of jobs increases. For example, the average execution time for solving a problem of six jobs and three machines in Experiment I was 1.92 seconds. However, the computer spent on the average 28.42 seconds for solving a problem of eight jobs and three machines in Experiment II, i.e. the execution time spent increases approximately 14 times as the number of jobs increases from six to eight jobs with the same number of machines. See Tables 4.2, ~~and~~ 4.4. and 4.9.

Table 4.1

Ranges of Efficiencies and Execution Time Obtained in Experiment I.

	Complete Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K=3	Rounded Linear Programming
Efficiency	1.00	1.00	1.00	0.79 - 1.00	0.86 - 1.00	0.60 - 1.00
Execution Time†	55.85	0.28 - 8.17	1.70 - 31.88	1.33 - 2.66	12.53*	**

Table 4.2

Average Values of Efficiencies and Execution Time Obtained in Experiment I.

	Complete Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K=3	Rounded Linear Programming
Efficiency	1.00	1.00	1.00	0.95	0.97	0.87
Execution Time†	55.85	1.95	5.69	2.12	12.53*	**

*The computer program for the Decomposition Algorithm has been written for the IBM 7040. Due to storage limitations, two parts of this program are not efficient. The first part is the construction of the complete set of arrangements of J jobs involving two subgroubs each having J/2 jobs. The second part is the evaluation of these R arrangements. The computer time can be reduced immensely by finding a scheme to enumerate these arrangements directly, and by storing the schedule time, and the starting and finishing times on each machine of each subgroub in the first R/2 arrangements.

**These values are not available.

†Execution time in seconds

Table 4.3

Ranges of Efficiencies and Execution Time Obtained in Experiment II.

	Partial Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K=4
Efficiency	0.98 - 1.00	1.00	1.00	0.92 - 1.00	0.99 - 1.00
Execution† Time	339.00	3.99 - 91.79	4.90 - 630.00	4.86 - 8.02	50.12*

Table 4.4

Average Values of Efficiencies and Execution Time obtained in Experiment II.

	Partial Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II	Decomposition with K=4
Efficiency	0.99	1.00	1.00	0.97	1.00
Execution† Time	339.00	28.14	57.26	5.85	50.12*

*See the footnote under Tables 4.1 and 4.2

†Execution time in seconds

4.3. The number of feasible sequences increases as the number of machines increases. The results of Experiments I and III shows that the average number of feasible sequences generated for problems of three and five machines with the same number of jobs are 15 and 44.6, respectively. These averages are computed from Tables 3.5 and 3.10.

4.4. The number of feasible sequences increases as the number of jobs increases. From Experiments I and II, it appears that the average number of feasible sequences generated from problems of six and eight jobs with the same number of machines are 15 and 35.8, respectively. These averages are computed from Tables 3.5 and 3.8.

4.5. The number of partial sequences depends upon the processing times of the jobs on the machines. Therefore, the complete sequences generated for the same size problems varies as the processing times change. Consequently, there is a high fluctuation in the execution times spent for the various problems in each of the experiments. This is shown in Tables 4.1, 4.3, and 4.5, where the ranges of the execution times are 0.28 - 8.17 seconds in Experiment I, 3.99 - 91.79 seconds in Experiment II; and 2.83 - 13.43 seconds in Experiment III, respectively.

5. In the Direct method, the number of optimal sequences generated decreases as the number of those obtained by Complete Enumeration decreases. Furthermore, the Direct method produces very few optimal sequences. Table 4.7 shows the frequencies of the optimal sequences obtained by Direct method in Experiment I.

6. In Branch-and-Bound method II, the minimum number of nodes are explored. In other words, only one node is explored at each level of the scheduling

Table 4.5

Ranges of Efficiencies and Execution Time Obtained in Experiment III.

	Complete Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II
Efficiency	1.00	1.00	1.00	0.30 - 1.00
Execution* Time	96.57	2.83 - 13.43	1.30-22.30	1.01 - 2.30

Table 4.6

Average Values of Efficiencies and Execution Time Obtained in Experiment III

	Complete Enumeration	Direct	Branch-and-Bound I	Branch-and-Bound II
Efficiency	1.00	1.00	1.00	0.94
Execution* Time	96.57	8.43	6.50	1.71

*Execution time in seconds.

Table 4.7

Frequencies of Optimal Sequences Obtained by Direct Method

No. of Optimal Sequences	No. of Problems	No. of Optimal Sequences	No. of Problems
1	40	6	2
2	16	7	4
3	19	8	2
4	13	9	1
5	2	10	1

tree. This method does not produce the optimal solution in all problems, since the bounds, computed for each node, upon which the branching is based, are not powerful enough to guarantee optimality. However, to obtain the optimal solution, more nodes should be explored. For example, Table 4.8 shows the size of the scheduling trees for problems having six and eight jobs, with three machines and for problems of six jobs with five machines. The number of nodes explored in the experiments conducted by Brown and Lomnicki (27) using Branch-and-Bound I are shown in the same table.

Table 4.8

Number of Nodes in Various Problems Solved by Branch-and-Bound I

Size of the Problem (J x M)	(6 x 3)	(6 x 5)	(8 x 3)
Total no. of nodes (Size of Scheduling Tree)	1236	1236	69,280
Maximum no. of nodes explored	375	343	4,500
Minimum no. of nodes explored	20	20	35
Average no. of nodes explored to get one solution	67	100.5	408.7
Average no. of nodes explored to get all solutions	143.8	-	-

From the above table, it appears that the execution time spent for solving the problems by Branch-and-Bound technique I vary according to the number of nodes explored. However, in Branch-and-Bound technique II, the execution times remain approximately the same, since only the minimum number of nodes are explored.

7. On examining the results of Branch-and-Bound technique I, the effects of changes in the number of machines and jobs are as follows:

7.1. The execution time spent should increase as the number of machines increases. It should be pointed out that the lower bound at each node is selected as the maximum value of the bounds of all machines. Therefore, the execution time spent to compute the lower bound at each node depend on the number of machines. For example, the average execution time for solving a problem of six jobs and three machines is 5.69 seconds as shown in Table 4.1; and on the average 6.5 seconds are required for solving a problem of six jobs and five machines as shown in Table 4.6.

7.2. The execution time spent increases as the number of jobs increases. For example the average execution time for solving a problem of six jobs and three machines is 5.69 seconds, see Table 4.1. However, on the average 57.26 seconds are spent for solving a problem of eight jobs and three machines as tabulated in Table 4.4.

8. It has been reported by Brown and Lomnicki (27) that for the same number of jobs with increasing the number of machines, the additional execution time in obtaining all the optimal sequences instead of one should decrease. Although, they have not stated any reason for this conclusion, it seems, from the logical point of view, that the number of optimal

sequences decreases as the number of machines increases. Tables 3.10 and A.1 may support this reasoning.

9. In analyzing the results obtained by the Branch-and-Bound technique II, the effects of the change in the number of machines and jobs are as follows:

9.1. Table 4.2 shows that the average execution time spent to obtain the solution for a problem of six jobs and three machines in Experiment I is 2.12 seconds. However, the computer spent on the average in Experiment III 1.71 seconds for solving a problem with six jobs and five machines, see Table 4.6. The reason might be that in Experiment I, more than one node at each level were explored due to step 5 of the Branch-and-Bound II, section 2.3.

9.2. The execution time spent increases approximately twice as the number of jobs increases from six to eight jobs with the same number of machines.

10. In comparing the various methods used in all experiments conducted in this report and those reported by other investigators whose results appear in Tables 4.1 - 4.6, the conclusions may be as follows:

10.1. The Direct method guarantee optimality; however the execution time required increases rapidly with the increase of problem size.

10.2. In general, the Branch-and-Bound technique can produce the optimal solutions depending on the computed time available. This technique is flexible, since any number of nodes can be explored. Many variations can also be embodied in the algorithm.

10.3. The decomposition approach with k equaling J/2 increases the

Table 4.9

Effect of Changing the number of jobs on Efficiency and Execution Time

Problem Size	Complete Enumeration		Direct		Branch-and-Bound I		Branch-and-Bound II		Decomposition	
	EF	ET†	EF	ET†	EF	ET†	EF	ET†	EF	ET†
6x3	1.00	55.85	1.00	1.95	1.00	5.69	0.95	2.12	0.97	12.53
8x3	0.99	339.00	1.00	28.14	1.00	57.26	0.97	5.85	1.00	50.12
12x3	1.00	594.81	1.00	311.81	*	*	0.96	38.59	*	*

Table 4.10

Effect of changing the number of machines on Efficiency and Execution Time

Problem Size	Complete Enumeration		Direct		Branch-and-Bound I		Branch-and-Bound II		Decomposition	
	EF	ET†	EF	ET†	EF	ET†	EF	ET†	EF	ET†
6x3	1.00	55.85	1.00	1.95	1.00	5.69	0.95	2.12	0.97	12.53
6x5	1.00	96.57	1.00	8.43	1.00	6.50	0.94	1.71	*	*

EF Efficiency

ET Execution Time

* These values are not available

† Execution time in seconds

capability of the Direct and Branch-and-Bound methods for solving larger or more practical scheduling problems.

10.4. Although the linear programming technique has a potentiality in the future, the resulting solutions do not seem sufficiently close to the optimal values to warrant the amount of the computational effort involved, as reported by Giglio and Wagner (57).

APPENDIX A

TABULATED RESULTS

This appendix includes various results of Experiment I. Table A.1 shows the frequency of the minimum schedule time obtained by different approaches. The relative cumulative frequency of the minimum schedule times obtained by various approaches appear in Table A.2. A summary of the computational results is shown in Table A.3.

Table A.1

Frequency Table of the Minimum Schedule Times Obtained by Different Approaches

Minimum Schedule Time	Complete Enumeration	Direct Method	Branch-and Bound II
85	1	1	
92	1	1	1
93	1	1	1
94	1	1	
95	1	1	1
98	2	2	
99	1	1	1
100			1
101	2	2	
102	1	1	2
103	1	1	
104	1	1	
105			2
106	2	2	2
107	3	3	1
109	1	1	1
110	1	1	1
111	2	2	1
112	1	1	3
113	1	1	3
114	4	4	2
115	3	3	3
116			2
118	1	1	1
119	2	2	2
120	3	3	1
121	3	3	1
122	2	2	1
123	2	2	1
124	1	1	
125	3	3	2
126	3	3	2
127			2
128	1	1	1
129	3	3	1

Table A.1 (continued)

Minimum Schedule Time	Complete Enumeration	Direct Method	Branch-and-Bound II
130	3	3	3
131	3	3	3
132	3	3	6
133	1	1	
134	4	4	3
135	3	3	2
136			1
137	2	2	1
138			3
139			2
140	1	1	1
141	2	2	1
142	3	3	3
143	3	3	3
144	2	2	2
145			3
148			1
149	1	1	
150	2	2	3
151	1	1	1
152	1	1	
153			1
154	1	1	3
155	2	2	1
157	2	2	1
159	2	2	1
160			1
161	1	1	
162			1
163			1
165	1	1	1
166	1	1	2
169			1
174			1
	100	100	100

Table A.2

Relative Cumulative Frequency of the Minimum Schedule Times Obtained by Different Approaches

Minimum Schedule Time	Complete Enumeration	Direct Method	Branch-and-Bound II
< 89.5	0.01	0.01	0.00
97.5	0.05	0.05	0.03
105.5	0.13	0.13	0.09
113.5	0.24	0.24	0.21
121.5	0.40	0.40	0.33
129.5	0.55	0.55	0.40
137.5	0.74	0.74	0.62
145.5	0.85	0.85	0.80
153.5	0.90	0.90	0.86
161.5	0.98	0.98	0.93
169.5	1.00	1.00	0.99
≥ 169.5	1.00	1.00	1.00

Table A.3

Computational Results of Experiment I

Problem No.	Complete Enumeration: No. of Optimal Sequences	Complete Enumeration: Execution Time†	Direct: No. of Sequences Generated	Direct: No. of Optimal Sequences	Direct: Execution Time†	Branch-and-bound: Efficiency	Branch-and-bound: Execution Time†
2	1	54.68	27	1	2.68	0.79	2.47
17	1	54.67	26	1	3.20	0.87	2.86
30	1	54.56	6	1	0.57	1.00	1.56
38	1	54.74	37	1	5.83	0.92	2.29
39	1	54.69	21	1	2.65	1.00	1.82
46	1	54.60	15	1	1.52	0.85	2.45
48	1	54.64	42	1	8.24	0.96	2.90
49	1	54.61	17	1	1.61	0.88	2.30
55	1	54.61	31	1	4.64	0.99	1.92
70	1	57.14	11	1	0.83	0.94	2.81
96	1	56.96	19	1	1.85	0.96	2.15
97	1	56.98	16	1	1.42	0.89	2.58
100	1	57.00	17	1	1.56	1.00	1.63
40	2	54.75	16	2	1.52	0.98	2.09
44	2	54.77	21	1	1.83	0.99	1.79
45	2	54.73	22	1	3.05	0.85	2.45
47	2	54.64	18	1	1.93	0.95	2.90
54	2	54.66	21	2	1.73	1.00	1.80
57	2	54.71	26	1	2.67	0.97	1.76
61	2	55.71	47	2	6.87	0.93	2.03
74	2	54.87	23	1	3.74	0.82	2.62
79	2	57.03	21	1	2.18	0.86	2.38
80	2	57.31	20	1	1.93	0.86	3.27
86	2	59.36	24	1	2.13	1.00	2.08
94	2	56.99	17	1	1.64	0.94	2.51
1	3	54.80	4	1	0.40	0.98	2.13
8	3	58.11	20	2	1.93	0.99	3.04
14	3	54.66	9	1	0.64	1.00	2.39
19	3	54.78	24	2	2.42	0.92	2.84
32	3	54.77	18	1	2.03	1.00	2.61
53	3	54.73	23	2	3.51	0.99	1.98
59	3	54.77	21	1	3.33	0.98	2.33
67	3	54.85	4	1	0.40	1.00	1.81
89	3	57.38	18	1	2.30	0.95	2.00

†Execution time in seconds.

Table A.3 (Continued)

Problem No.	Complete Enumeration No. of Optimal Sequences	Complete Enumeration Execution Time	Direct No. of Sequences Generated	Direct No. of Optimal Sequences	Direct Execution Time	Branch-and-bound Efficiency	Branch-and-bound Execution Time
3	4	57.62	7	1	0.43	0.91	2.20
4	4	59.65	8	2	0.54	0.99	2.44
23	4	54.80	22	1	1.95	0.95	1.76
24	4	54.76	19	3	2.23	0.94	3.02
34	4	54.79	14	1	1.46	1.00	1.53
37	4	54.81	7	1	0.70	0.93	2.29
43	4	54.66	6	3	0.41	1.00	1.52
60	4	54.71	3	2	0.20	1.00	2.05
62	4	56.96	21	1	2.43	1.00	2.38
64	4	54.78	18	2	3.76	1.00	1.53
76	4	54.98	17	2	1.42	0.96	1.74
77	4	54.86	36	1	4.57	0.97	2.48
81	4	58.36	20	1	2.45	0.90	2.53
90	4	57.26	9	1	0.68	0.91	2.21
42	5	54.78	10	3	0.67	0.93	2.12
52	5	54.68	18	3	1.94	0.94	2.38
15	6	54.73	10	3	0.99	1.00	1.53
25	6	54.81	15	2	1.43	0.98	2.81
63	6	56.44	18	2	3.76	0.99	1.31
72	6	57.44	9	3	0.67	1.00	1.53
88	6	57.66	17	3	1.53	0.98	3.16
20	7	54.81	16	3	2.56	1.00	2.29
65	7	55.06	11	1	1.07	0.96	3.00
11	8	57.83	11	3	0.96	0.96	2.96
41	8	54.90	9	2	0.63	1.00	1.52
29	10	54.83	5	3	0.50	1.00	1.56
99	10	57.29	22	1	2.49	0.90	2.10
21	12	55.01	12	4	1.02	1.00	1.51
33	12	55.12	32	4	5.90	0.95	2.10
35	12	54.95	15	4	1.24	1.00	2.42
92	12	57.54	13	4	1.24	1.00	1.52
93	12	57.94	9	1	0.70	1.00	1.65
22	13	54.97	15	2	1.33	1.00	1.52
51	13	54.92	13	3	1.40	1.00	2.31
12	14	57.15	15	4	1.08	0.96	2.35
73	14	57.43	7	3	0.63	1.00	1.53
16	16	55.16	21	3	3.15	0.92	3.11
26	16	55.24	6	2	0.45	0.95	2.28

Table A.3 (continued)

Problem No.	Complete Enumeration No. of Optimal Sequences	Execution Time	No. of Sequences Generated	Direct No. of Optimal Sequences	Execution Time	Branch-and-Bound Efficiency	Execution Time
31	18	55.03	5	3	0.28	1.00	1.52
78	19	56.56	15	3	2.35	0.99	1.89
82	20	58.12	12	3	1.43	1.00	2.13
5	22	60.29	16	4	2.13	0.99	2.70
7	22	59.46	6	3	0.47	0.98	2.27
87	23	58.06	10	6	0.95	1.00	1.61
6	24	61.87	20	4	2.03	0.96	3.23
68	24	59.66	10	2	1.14	0.97	2.20
84	24	60.60	17	4	3.19	1.00	1.80
71	25	57.69	7	4	0.78	1.00	2.50
50	26	55.51	25	6	2.83	1.00	2.57
58	28	55.56	22	3	1.85	1.00	2.06
95	28	58.01	8	4	0.65	1.00	1.63
75	32	55.96	15	5	1.37	1.00	1.51
56	34	55.78	12	7	1.64	1.00	1.97
83	41	61.14	8	4	0.51	1.00	2.51
69	43	58.06	21	7	2.56	1.00	1.62
91	44	58.56	14	7	1.62	1.00	1.65
13	48	58.37	14	7	2.68	1.00	1.60
28	54	56.25	14	4	1.20	1.00	1.52
36	60	56.38	5	1	0.61	1.00	1.54
18	68	56.75	11	8	1.83	0.89	3.25
27	72	57.12	40	9	8.17	1.00	1.51
10	80	61.72	4	4	0.31	1.00	1.99
25	80	61.65	14	5	1.39	1.00	1.76
66	87	57.66	5	3	0.40	0.88	2.66
9	110	61.32	10	10	1.08	1.00	1.99
98	120	61.11	17	8	1.81	0.97	2.43

APPENDIX B

COMPUTER PROGRAM

This appendix contains a brief description of the subroutines that have been used and also the computer program. This consists of the main routine and five subroutines: ENUMER, DIRECT, BRANCH, STAT, ESTMAT. ENUMER, STAT and ESTMAT subroutines are taken from Ashour (6). ENUMER is same as the FLOSHP, except that the output is slightly changed. For more detail about STAT and ESTMAT, see Ashour (6). All subroutines are written in FORTRAN IV language for the IBM 360/50 system.

MAIN Routine

The main routine acts as a control program. This routine may read or randomly generate the processing time matrix. Then, it calls various subroutines to solve the problem by different methods.

The processing times of each problem are randomly generated as integar values from a uniform distribution with a finite interval, [a, b].

In the main routine, TIME subroutine, which is embodied in the 360/50 system, is being called before and after the subroutines ENUMER, DIRECT and BRANCH to determine the time spent in obtaining the solution by various algorithms.

ENUMER Subroutine

This subroutine constructs and evaluates the feasible schedules for the flowshop problem. Since the number of the permutations increases very rapidly as the number of jobs increases, the ENUMER routine contains two phases. The phase I procedure is to enumerate a complete set of permutations. This phase is used for problems with maximum of six jobs. The phase II procedure is to randomly sample a subset from the set of all permutations. This phase is applied to problems having more than six jobs.

DIRECT Subroutine

This subroutine is a computer program written by Teuton (164) and modified by Smith (155) to solve the flowshop scheduling problem by the direct technique presented in section 2.2.

It consists of the main phases. The first phase checks for job dominance, the second phase checks for sequence dominance. This is done until J-2 sequence-positions are filled. The last two sequences are then filled by switching the remaining two jobs and then the total idle time is checked. Those retained are evaluated to find the corresponding schedule time.

BRANCH Subroutine

This subroutine is written to solve the flowshop scheduling problem by the branch-and-bound algorithm developed by Brown and Lomnicki (27). In selecting a node, at a certain level, which has the minimum lower bound, a tie may exist. This subroutine breaks the tie by random. Furthermore, according to Brown and Lomnicki, the minimum lower bound at each level is checked against that in the previous level. If it is greater, a node which has the second minimum lower bound at the previous level is considered for branching. However, if it is equal; the branching will take place from this node to the following level.

Input Specifications:

The input data deck has two sets of cards for each job scheduling problems: (1) Control card, (2) Original Data Cards.

The set of the original data cards is omitted when the data of the original problem is to be generated randomly. The description of the input data cards is as follows:

1. Control Data Cards: The FORTRAN format for this card is FORMAT(14I5)

Columns	Variable	Definition
1 - 5	JOBS	Total no. of jobs
6 - 10	MACH	Total no. of machines
11 - 15	ITYPE	Type of Problem 1 = Flowshop; 0 = Jobshop.
16 - 20	IREAD	Data origination 1 = Generate randomly; 0 = Read from cards.
21 - 25	IX	The starting point for the random no generator
26 - 30	LIMIT1	Used only when IREAD = 1; smallest value in the interval [a, b]
31 - 35	LIMIT2	Used only when IREAD = 1; largest value in the interval [a, b].
36 - 40	NQ	No. of sequences to be constructed and evaluated for obtaining a complete or partial enumeration solution
41 - 45	IAI	Maximum no. of jobs to be permuted through phase I in the ENUMER subroutine.
46 - 50	ISWICH	Conditional print out, 0 = Printout each schedule in detail; 1 = supress conditional printout.

Columns	Variable	Definition
51 - 55	IPRINT	Conditional printout 0 = Printout; 1 = Supress the conditional printout.
56 - 60	ICARD	Conditional punch out 0 = Punch the no. of schedules and schedule times on cards; 1 = Supress the conditional punch out.

2. Original Data Cards: When IREAD equals zero, the set of cards follows the above control card. The processing time matrix is read such that for each job, the processing times on various machines are given on one card. The FORTRAN statement for the above data is FORMAT (16 I5).

Output Specification:

The most summarized output obtained by each method includes: (1) a list of the optimal sequences (or the sequences generated) and their schedule times; (2) Execution time spent to obtain the solution; and (3) Various statistics (except in Branch-and-Bound II).

BIBLIOGRAPHY

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

2. Agin, N., "Optimum Seeking by Branch-and-Bound", Management Science, Vol. 13, No. 4, 1966, pp. 176-185.

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

4. Akers, Jr., S., and J. Friedman, "A Non-Numerical Approach to Production Scheduling Problems," Operations Research, Vol. 3, No. 4, 1955, pp. 429-442.

5. Alcalay, J., E. Buffa, "A Proposal for a General Model of Production System," International Journal of Production Research, Vol. 2, No. 1, 1963, pp. 73-88.

6. Ashour, S., "A Decomposition Approach for the Machine Scheduling Problem," Ph.D. Thesis, University of Iowa, Iowa City, Iowa, 1967.

7. Ashour, S., "A Decomposition Approach for the Machine Scheduling Problem," The International Journal of Production Research, Vol. 6, No. 2, 1967, pp. 109-122.

8. Ashour, S., "A Lower-bound for Flowshop Scheduling Problem," Submitted to the Journal of Canadian Operational Research Society for Publication, 1968.

9. Ashour, S., "Experimental Investigation of the Effects of Suboptimization in Sequencing Problems," Submitted to Management Science for publication, 1968.

10. Ashour, S., "Flowshop Scheduling with Linear Graphs Algorithm," submitted to the Journal of the Operations Research Society of Japan for publication, 1968.

11. Baker, C. and B. Dzielinski, "Simulation of a Simplified Job Shop," IBM Business Systems Research Memorandum, August 1, 1958; also Management Science, Vol. 6, No. 3, 1960, pp. 311-323.

12. Baker, C., B. Dzielinski, and A. Manne, "Simulation Tests of Lot Size Programming," Management Science, Vol. 9, No. 2, 1963, pp. 229-258.

13. Banerjee, B., "Single Facility Sequencing with Random Execution Times," Operations Research, Vol. 13, No. 3, 1965, pp. 358-364.

14. Banerjee, B., "The Shop decomposition problem," The Journal of Operations Research Society of India, Vol. 4, No. 1, 1967, pp. 9-19.

15. Barankin, E., "The Scheduling Problem as an Algebraic Generalization or Ordinary Linear Programming," Discussion Paper, No. 9, University of California, Los Angeles, California, 1952.

16. Barnes, W., "The Application of Computer Simulation to Production Scheduling Research," 16th National Meeting of the Operations Research Society of America, 1962.

17. Beenhakker, H., "Development of Alternate Criteria for Optimality in the Machine Sequencing Problem," Ph.D. Thesis, Purdue University, Lafayette, Indiana, 1963.

18. Beenhakker, H., "Mathematical Analysis of Facility-commodity Scheduling Problems," The International Journal of Production Research, Vol. 2, No. 4, 1963, pp. 313-321.

19. Beenhakker, H., "Optimization versus Suboptimization," The International Journal of Production Research, Vol. 3, No. 4, 1964, pp. 317-325.

20. Bellman, R., "Combinatorial Processes and Dynamic Programming," The RAND Corporation, P-1284, Feb., 1958.

21. Bellman, R., "Some Mathematical Aspects of Scheduling Theory," Journal of the Society for Industrial and Applied Mathematics, Vol. 4, No. 3, 1956, pp. 168-205.

22. Bellman, R. and S. Dreyfus, Applied Dynamic Programming, Princeton University, Princeton, New Jersey, 1962.

23. Bellman, R. and O. Gross, "Some Combinatorial Problems Arising in the Theory of Multi Stage Processes," Journal of the Society for Industrial and Applied Mathematics, Vol. 2, No. 3, 1954, pp. 175-183.

24. Blake, K. and W. Stopakis, "Some Theoretical Results on the Job Shop Scheduling Problem," Report M-1533-1, United Aircraft Corp. Research Dept., East Hartford, Connecticut, July, 1959.

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

26. Brooks, G., and C. White, "An Algorithm for Finding Optimal or Near-Optimal Solutions to the Production Scheduling Problem," The Journal of Industrial Engineering, Vol. 16, No. 1, 1965, pp. 34-40.

27. Brown, A. and Z. Lomnicki, "Some Applications of the Branch-and-Bound Algorithm to the Machine Scheduling Problem," Operational Research Quarterly, Vol. 17, No. 2, 1966, pp. 173-186.

28. Burstall, R., "A Heuristic Method for a Job-Scheduling Problem," Operational Research Quarterly, Vol. 17, No. 3, 1966, pp. 291-304.

29. Carlson, R. and G. Nemhauser, "Scheduling to Minimize Interaction Cost," Operations Research, Vol. 14, No. 1, 1966, pp. 52-58.

30. Churchman, C., R. Ackoff, and E. Arnoff, Introduction to Operations Research, John Wiley & Sons, New York, 1961, pp. 450-476.

31. Clark, W., The Gantt Chart, Pitman and Sons, London, 1952.

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

33. Conway, R., W. Maxwell and J. Oldziey, "Sequencing Against Due-Dates," Proceedings of IFORS Conference, Cambridge, Massachusetts, September, 1966.

34. Crabill, T., "A Lower-Bound Approach to the Scheduling Problem," Research report, Department of Industrial Engineering, Cornell University, Ithaca, New York, 1964.

35. Dantzig, G., "A Machine-Job Scheduling Model," Management Science, Vol. 6, No. 2, January 1960, pp. 187-190.

36. Dantzig, G., "On the Shortest Route Through a Network", Management Science, Vol. 6, No. 2, 1960, pp. 187-190.

37. Dudek, R. and P. Ghare, "Make-Span Sequencing on M-Machines," The Journal of Industrial Engineering, Vol. 18, No. 1, 1967, pp. 131-134.

38. Dudek, R., and O. Teuton, Jr., "Development of M-Stage Decision Rule for Scheduling n Jobs Through m Machines," Operations Research, Vol. 12, No. 3, 1964, pp. 471-497.

39. Eastman, W., S. Even, and I. Isaacs, "Bounds for the Optimal Scheduling of n Jobs on m Processors," Management Science, Vol. 11, No. 2, 1964, pp. 1-13.

40. Eilon, S. and R. Hodgson, "Job Shops Scheduling With Due Dates," The International Journal of Production Research, Vol. 6, No. 1, 1967, pp. 1-13

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

42. Elmaghraby, S., and R. Cole, "On the Control of Production in Small Job Shops," Journal of Industrial Engineering, Vol. 14, No. 4, 1963, pp. 180-196.

43. Elmaghraby, S. and A. Ginsberg, "A Dynamic Model of the Optimal Loading of Linear Multi-Operation Shops," Management Technology, Vol. 4, No. 1, 1964.

44. Fischer, H., and G. Thompson, "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules," Chapter 15, in Reference 108.

45. Gapp, W., P. Mankekar, and L. Mitten, "Sequencing Operations to Minimize In-Process Inventory Costs," Management Science, Vol. 11, No. 3, 1965, pp. 476-484.

46. Gavett, J., "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility," Management Science, Vol. 11, No. 8, 1965, pp. B-166--B-176.

47. Gere, W., "A Heuristic Approach to Job-Shop Scheduling," Ph.D. Thesis, Carnegie Institute of Technology, 1962.

48. Gere, W., "Heuristics in Job Shop Scheduling," Management Science, Vol. 13, No. 3, 1966, pp. 167-190.

49. Giffler, B., "Mathematical Solution of Explosion and Scheduling Problems," IBM Research Report RC-118, Yorktown Heights, New York, June 1959.

50. Giffler, B., "Mathematical Solution of Production Planning and Scheduling Problems, IBM ASDD Technical Report 09.026, White Plains, New York, Oct. 1960.

51. Giffler, B., SIMPRO 1: An IBM 704-7090 "Simulation Program for Planning Scheduling and Monitoring Production Systems," IBM ASDD Technical Report 17-053, White Plains, New York, Dec. 1961.

52. Giffler, B., "Schedule Algebras and Their Use in Formulating General Systems Simulations," Chapter 4, in Reference 108.

53. Giffler, B., "Scheduling General Production Systems Using Schedule Algebra," Naval Research Logistics Quarterly, Vol. 10, No. 3, 1963, pp. 237-255.

54. Giffler, B., and G. Thompson, "Algorithms for Solving Production Scheduling Problems," IBM Research Report RC-118, Yorktown Heights, New York, June 1959.

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

56. Giffler, B., G. Thompson, and V. Van Ness, "Numerical Experience with the Linear and Monte Carlo Algorithms for Solving Production Scheduling Problems," Chapter 3, in Reference 108.

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

58. Gilmore, P. and R. Gomory, "Sequencing a One State-Variable Machine: A Solvable Case of the Traveling-Salesman Problem," Operations Research, Vol. 12, No. 5, 1964, pp. 655-679.

59. Gotterer, M., "Scheduling with Deadlines, Priorities, and Non-linear Loss Functions," Research report, Department of Industrial Engineering, Georgia Institute of Technology, Atlanta, Georgia.

60. Greenberg, H., "A Branch-Bound Solution to the General Scheduling Problem," Operations Research, Vol. 16, No. 2, 1968, pp. 353-361.

61. Gupta, J., "Combinatorial Search Algorithms for Scheduling Problems," Abstract of Thesis, The Journal of Operations Research Society of India, Vol. 4, No. 3, 1967, pp. 133-134.

62. Hardgrave, W., and G. Nemhauser, "A Geometric Model and Graphical Algorithm for a Sequencing Problem," Operations Research, Vol. 11, No. 6, 1963, pp. 889-900.

63. Held, M. and R. Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of the Society for Industrial and Applied Mathematics, Vol. 10, No. 2, 1962, pp. 196-209.

64. Held, M., R. Karp, and R. Shareshian, "Scheduling with Arbitrary Profit Functions", Data Systems Division, Mathematics and Applications Dept., IBM Corporation, New York.

65. Heller, J., "Combinatorial Properties of Machine Shop Scheduling", Report NYO-2879, Atomic Energy Commission Computing and Applied Mathematics Center, Institute of Mathematical Science, New York University, New York, July 1959.

66. Heller, J., "Some Numerical Experiments for an M x J Flow Shop and its Decision-Theoretical Aspects," Operations Research, Vol. 8, No. 2, 1960, pp. 178-184.

67. Heller, J. "Some Problems in Linear Graph Theory that Arise in the Analysis of the Sequencing of Jobs through Machines," Report NYO-9487, Atomic Energy Computing and Applied Mathematics Center, New York, 1960.

68. Heller, J., and G. Logemann, "An Algorithm for the Construction and Evaluation of Feasible Schedules," Management Science, Vol. 8, No. 3, January 1962, pp. 168-181.

69. IBM Mathematics and Applications Department, The Job Shop Simulator, 1271 Avenue of the Americas, New York, 1960.

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

71. Jackson, J., "An Extension of Johnson's Results on Job-Lot Scheduling," Naval Research Logistics Quarterly, Vol. 3, No. 3, 1956, pp. 201-203.

72. Jackson, J., "Machine Shop Simulation Using SWAC: A Progress Report," Management Sciences Research Project, Discussion Paper No. 67, University of California, Los Angeles, California, April 1958.

73. Jackson, J., "Notes on Some Scheduling Problems," Research Report No. 35, Management Sciences Research Project, University of California, Los Angeles, California, 1954.

74. Jackson, J., "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report 43, Management Sciences Research Project, University of California, Los Angeles, California, 1955.

75. Jackson, J., "Simulation Research on Job-Shop Production," Naval Research Logistics Quarterly, Vol. 4, No. 4, 1957, pp. 287-295.

76. Jackson, J., and Y. Kurantani, "Production Scheduling Research: A Monte Carlo Approach," Management Sciences Research Project, Research Paper No. 61, University of California, Los Angeles, California, May 1957.

77. Jackson, J., and R. Nelson, "SWAC Computations for Some m x n Scheduling Problems," Journal of the Association of Computing Machinery, Vol. 4, No. 4, 1957, pp. 438-441.

78. Jaeschke, G., "Branching and Bounding: Eine algemeine Methode zur Loesung kombinatorischer Probleme,"

79. Jeremiah, B., A. Lalchandani, and L. Schrage, "Heuristic Rules Toward Optimal Scheduling," Research report, Department of Industrial Engineering, Cornell University, Ithaca, New York, 1964.

80. Johnson, S., "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Naval Research Logistics Quarterly Vol. 1, No. 1, 1954, pp. 61-68; Also, Chapter 2, in Reference 108.

81. Johnson, S., "Discussion: Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol. 5, No. 3, 1959, pp. 299-303.

82. Karush, W., "A counter example to a Proposed Algorithm for Optimal Sequencing of Jobs," Operations Research, Vol. 13, No. 2, 1965, pp. 323-325.

83. Karush, W., and L. Moody, "Determination of Feasible Shipping Schedules for a Job Shop," Operations Research, Vol. 6, No. 1, 1958, pp. 35-55.

84. Karush, W. and A. Vazsonui, "Mathematical Programming and Service Scheduling," Management Science, Vol. 3, No. 2, 1957, pp. 140-148.

85. Kisi, T., "On the Optimal Searching Schedule," Journal of the Operations Research Society of Japan, Vol. 8, No. 2, 1966, pp. 53-65.

86. Kurantani, Y. and R. Nelson, "A Pre-Computational Report on Job-Shop Simulation Research," Journal of the Operations Research Society of Japan, Vol. 2, 1960, pp. 145-183.

87. Kuratani, Y., and J. McKenney, "A Preliminary Report on Job-Shop Simulation Research," Research Report 65, Management Sciences Research Project, University of California, Los Angeles, California, 1958.

88. Lawler, E., "On Scheduling Problems with Deferral Costs," Management Science, Vol. 11, No. 2, 1964, pp. 280-288.

89. Lawler, E. and D. Wood, "Branch-and-Bound Methods: A Survey," Operations Research, Vol. 14, No. 4, 1966, pp. 699-719.

90. LeGrande, E., "The Development of a Factory Simulation Using Actual Operating Data," Management Technology, Vol. 3, No. 1, 1963, pp. 1-19.

91. Ligtenberg, E., "Minimal Cost Sequencing of n Grouped and Ordered Jobs on m Machines," The Journal of Industrial Engineering, Vol. 17, No. 4, 1966, pp. 217-223.

92. Little, J., K. Murty, D. Sweeny, and C. Karel, "An Algorithm for the Traveling-Salesman Problem," Operations Research, Vol. 11, No. 6, 1963, pp. 972-989.

93. Lomnicki, Z., "A Branch-and-Bound Algorithm for the Exact Solution of the Three Machine Scheduling Problem," Operational Research Quarterly, Vol. 16, No. 1, 1965, pp. 89-107.

94. Makino, J., "On a Scheduling Problem," Journal of Operations Research Society of Japan, Vol. 8, No. 1, 1965, 32-44.

95. Mankekar, P., and L. Mitten, "The Constrained Least-Cost Testing Sequence Problem," The Journal of Industrial Engineering, Vol. 16, No. 2, 1965, pp. 146-149.

96. Manne, A., "On the Job-Shop Scheduling Problem," Operations Research, Vol. 8, No. 2, 1960, pp. 219-223; Also, Chapter 12, in Reference 108.

97. Mayhugh, J., "On the Mathematical Theory of Schedules, Management Science, Vol. 11, No. 2, 1964, pp. 289-307.

98. Maxwell, W., "The Scheduling of Single Machine Systems: A Review," The International Journal of Production Research, Vol. 3, No. 3, 1964, pp. 177-199.

99. McMahon, G. and P. Burton, "Flowshop Scheduling with Branch-and-Bound Method", Operations Research, Vol. 15, No. 3, 1967, pp. 473-481.

100. McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 6, No. 1, 1959, pp. 1-12.

101. Mehra, M., "An Experimental Investigation of Job-Shop Scheduling with Assembly Constraints," M.S. Thesis, Cornell University, Ithaca, New York, 1967.

102. Mellor, P., "A Review of Job Shop Scheduling", Operational Research Quarterly, Vol. 17, No. 2, 1966, pp. 161-171.

103. Mikhalerich, V. and V. Shkurba, "Sequential Optimization Methods in Problems of Operation Scheduling," (in Russian), Kibernetika, Vol. 2, No. 3, 1966, pp. 34-40. (in English), Cybernetics, (The Faraday Press Inc.), Vol. 2, No. 3, 1966, pp. 28-33.

104. Mitten, L., "A SCheduling Problem," The Journal of Industrial Engineering, Vol. 10, No. 2, 1959, pp. 131-135.

105. Mitten, L., "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, Vol. 5, No. 3, 1959, pp. 293-298.

106. Mueller-Merbach, M., "A Method For Solving Sequencing Problems of Industrial Production," Zeitchrift Fur wirtschaftliche Fertigung, Germany, Vol. 61, No. 3, 1966, pp. 147-152.

107. Muth, J., "The Effect of Uncertainty in Job Times on Optimal Schedules," Chapter 18, in Reference 108.

108. Muth, J., and G. Thompson, eds., Industrial Scheduling, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

109. Nabeshima, I., "Computational Solution to the M-machine Scheduling Problem," Journal of the Operations Research Society of Japan," Vol. 7, No. 3, 1965, pp. 93-103.

110. Nabeshima, I., "On the Bound of Makespans and its applications in M Machine Scheduling Problem," Journal of the Operations Research Society of Japan, Vol. 9, No. 3&4, 1967, pp. 6-44.

111. Nabeshima, I., "The order of n items processed on m machines," Journal of the Operations Research Society of Japan, Vol. 3, No. 4, 1961, pp. 170-175.

112. Nabeshima, I., "Sequencing on Two Machines with Start Lag and Stop Lag," Journal of the Operations Research Society of Japan, Vol. 5, No. 3, 1963, pp. 97-101.

113. Nabeshima, I., "Some Extensions of the M-machine Scheduling Problem,: Journal of the Operations Research Society of Japan, Vol. 10, No. 1 & 2, 1967, pp. 1-17.

114. Nabeshima, I., "The Order of n Items Processed on m Machines (II)," Journal of Operations Research Society of Japan, Vol. 4, No. 1, 1961, pp. 1-8.

115. Nabeshima, I., "Sequencing on Two Machines with Start Lag and Stop Lag," Journal of Operations Research Society of Japan, Vol. 5, No. 3, 1963, pp. 97-101.

116. Naik, M. D., "m by n Job-Shop Scheduling," M.S. Thesis, Cornell University, Ithaca, New York, 1967.

117. Neimeier, H. A., "An Investigation of Alternative Routing in a Job Shop," M.S. Thesis, Cornell University, Ithaca, New York, 1967.

118. Nelson, R., "Enumeration of a Three Job, Three Machine Scheduling Problem on SWAC," Discussion Paper No. 50, Management Sciences Research Project, University of California, Los Angeles, California, 1955.

119. Nelson, R., "Priority Function Methods for Job-Lot Scheduling," Research Report 51, Management Sciences Research Project, University of California, Los Angeles, California, 1955.

120. Nugent, C., "On Sampling Approaches to the Solution of the n-by-m Static Sequencing Problem," Ph.D. Thesis, Cornell University, Ithaca, New York, 1964.

121. Oldfather, P., A. Ginsberg, and H. Markowitz, Programming by Questionaire, IM 5129PR and IM 4460PR, The Rand Corporation, Santa Monica, California, 1966.

122. Oldziey, J., "Dispatching Rules and Job Tardiness in a Simulated Job Shop," M.S. Thesis, Cornell University, Ithaca, New York, 1966.

123. Orkin, G., "An Experimental Investigation of Shop Loading for Setting Operation Due-Dates," M.S. Thesis, Cornell University, Ithaca, New York, 1960.

124. Page, E., "An approach to the Scheduling of Jobs on a Computer," Journal of the Royal Statistical Society, Series B, Vol. 23, No. 2, 1961, pp. 484-492.

125. Page, E., "On Monte Carlo Methods in Congestion Problems: I. Searching For an Optimum in Discrete Situation," Operations Research, Vol. 13, No. 2, 1965, pp. 291-299.

126. Page, E., "On the Scheduling of Jobs by Computer," Computer Journal, Vol. 5, 1962, pp. 214-221.

127. Palmer, D., "Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time--A Quick Method of Obtaining a Near Optimum," Operational Research Quarterly, Vol. 16, No. 1, 1965.

128. Parekh, A., "The Machine Scheduling Problem," M.S. Report," Kansas State University, Manhattan, Kansas, 1968.

129. Presby, J. and M. Wolfson, "An Algorithm for Solving Job Sequencing Problems," Management Science, Vol. 13, No. 8, 1967, pp. 454-464.

130. Reep, H., "Here's Easy-way Job Scheduling," Factory, Vol. 117, No. 12, 1959, pp. 67.

131. Reinitz, R., "An Integrated Job-Shop Scheduling Problem," Ph.D. Thesis, Case Institute of Technology, Cleveland, Ohio, 1961.

132. Reintz, C., "On the Job-Shop Scheduling Problem," Chapter 5, Reference 108.

133. Root, J., "Scheduling with Deadlines and Loss Functions on k Parallel Machines," Management Science, Vol. 11, No. 3, 1965, pp. 460-475.

134. Roy, B., "Cheminement et Connexite dans les graphes - Applications aux problemes d' ordonnacement," METRA, Serie Speciale No. 1, Societe d'economie et de mathematiques appliquees, Paris, 1962.

135. Rowe, A., "Toward a Theory of Scheduling," The Journal of Industrial Engineering, Vol. 11, No. 2, 1960, pp. 125-136.

136. Rowe, A., "Sequential Decision Rules in Production Scheduling," Ph.D. Thesis, University of California, Los Angeles, California, 1958.

137. Rowe, A., "Sequential Decision Rules in Production Scheduling General Electric Company, Oct. 1958.

138. Rowe, A., "Toward a Theory of Scheduling," The Journal of Industrial Engineering, Vol. 11, No. 2, 1960, pp. 125-136.

139. Rowe, A., and J. Jackson, "Research Problems in Production Routing and Scheduling," Research Report No. 46, University of California, Los Angeles, California, Oct. 1956.

140. Russo, F., "A Heuristic Approach to Alternate Routing in a Job Shop," M.S. Thesis, Massachusetts Institute of Technology, 1965.

141. Salveson, M., "A Problem in Optimal Machine Loading," Management Science, Vol. 2, No. 3, 1956, pp. 232-260.

142. Salveson, M., "A Computational Technique for the Scheduling Problem, The Journal of Industrial Engineering, Vol. 13, No. 1, 1962, pp. 30-41.

143. Sasieni, M., A. Yaspan, and L. Friedman, Operations Research: Methods and Problems, Chapter 9, John Wiley, New York, 1959.

144. Schild, A., and I. Fredman, "Scheduling Tasks with Deadlines and Nonlinear Loss Functions," Management Science, Vol. 9, No. 1, 1962, pp. 73-81.

145. Schild, A. and I. Fredman, "Scheduling Tasks with Linear Loss Functions," Management Science, Vol. 7, No. 3, 1961, pp. 280-285.

146. Schultz, C., A. Brooks, and P. Schwartz, "Scheduling Meeting with a Computer," Communications of the Association for Computing Machinery, Vol. 7, No. 9, 1964, pp. 534-540.

147. Sherman, G., "The Use of a Computer for Scheduling Students," Presented before the annual meeting of the Operations Research Society of America, Washington, D. C., May 14-15, 1959.

148. Shkubra, V., "Computing Schemes For Solution of Problems in Scheduling Theory," (in Russian), Kibernetika, Vol. 1, No. 3, 1965, pp. 72-75. (in English), Cybernetics, (The Faraday Press, Inc.), Vol. 1, No. 3, pp. 74-78.

149. Shkubra, V., "Scheduling Theory I, Combinatory Problems," Seminar on Economic Cybernetics and Operations Research (in Russian), KDNTP, Kiev, 1964.

150. Shkubra, V., "Scheduling Theory II, General Approaches and Simulation Methods," Seminar on Economic Cybernetics and Operations Research (in Russian), KDNTP, Kiev, 1964.

151. Simon, H. and A. Newell, "Heuristic Problem Solving: The Next Advance in Operations Research," Operations Research, Vol. 6, No. 1, 1958, pp. 1-10.

152. Sisson, R., "Machine Shop Simulation Using SWAC: Part II of a Proposal," Management Sciences Research Project, Research Paper No. 58, Number 71, University of California, Los Angeles, California, 1956.

153. Sisson, R., "Methods of Sequencing in Job Shops - A Review," Operations Research, Vol. 7, No. 1, 1959, pp. 10-29.

154. Sisson, R., "Sequencing Theory," Chapter 7, Progress Operations Research, Vol. 1, Ackoff, R., ed., John Wiley, New York, 1961.

155. Smith, R., "A General Algorithm For Solution of the n-Job, M- Machine Sequencing Problem of the Flow Shop," M.S. Thesis, Texas Technological College, Lubbock, Texas, 1967.

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

157. Smith, W., "Various Optimizers for Single-Stage Production," Naval Research Logistics Quarterly, Vol. 3, No. 1, 1956, pp. 59-66.

158. Spradlin, B., D. Pierce, "Production Scheduling under a learning effect by Dynamic Programming," Journal of Industrial Engineering, Vol. 18, No. 3, 1967, pp. 219-222.

159. Story, A, and H. Wagner, "Computational Experience with Integer Programming for Job-Shop Scheduling,"; Also, Chapter 14, in Reference 108.

160. Szwarc, W., "Solution of the Akers-Friedman Scheduling Problem," Operations Research, Vol. 8, No. 6, 1960, pp. 782-788.

161. Talwar, P., "A Note on Sequencing Problem with Uncertain Job Time," Journal of the Operations Research Society of Japan, Vol. 9, No. 3&4, 1967, pp. 1-5.

162. Taft, M., and A. Reisman, "A Proposed Generalized Heuristic Algorithm for Scheduling with Respect to n- Integrated Criterion Functions," The International Journal of Production Research, Vol. 5, No. 2, 1966, pp. 155-162.

163. Thompson, G., "Recent Developments in the Job-Shop Scheduling Problem," Naval Research Logistics Quarterly, Vol. 7, No. 4, 1960, pp. 585-589.

164. Teuton, Jr., O., "Optimal M-Stage Production Schedules When No Passing is Permitted," M.S. Thesis, Texas Technological College, Lubbock, Texas, 1963.

165. Wagner, H., "An Integer Linear-Programming Model for Machine Scheduling," Naval Research Logistics Quarterly, Vol. 6, No. 2, 1959, pp. 131-140.

The following references have come to the author's attention since the completion of this bibliography:

166. Balinski, M., and R. Quandt, "On an Integar Program for a Delivery Problem," Operations Research, Vol. 12, No. 2, 1964, pp. 300-304.

167. Brown, R., "Simulation to Explore Alternative Sequencing Rules," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 281-286.

168. Bomberger, E., "A Dynamic Programming Approach to a Lot Size Scheduling Problem," Management Science, Vol. 12, No. 11, 1966, pp. 778-784.

169. Carroll, D., "Heuristic Sequencing of Single and Multiple Component Jobs," Ph.D. Thesis, Massachusetts Institute of Technology, 1965.

170. Charnes, A., W. Cooper, and D. Farr, "Linear Programming and Profit Preference Scheduling for a Manufacturing Firm," Operations Research, Vol. 1, No. 3, 1953, pp. 114-128.

171. Conway, R., "An Experimental Investigation of Priority Assignment in a Job Shop," Memorandum RM - 3789 - PR, Rand Corporation, 1964.

172. Dzielinki, B., C. Baker and A. Manne, "Simulation Tests of Lot Size Programming," Management Science, Vol. 9, No. 2, 1963, pp. 229-258.

173. Eilon, S., "Economic Bath Size Determination of Multi-Product Scheduling," Operations Research Quarterly, Vol.10, No. 4, 1959, pp. 217-227.

174. Eilon, S., and J. King, "Industrial Scheduling Abstracts - 1950 - 1966," Oliver & Boyd, Edinburgh and London, 1967.

175. Elmaghraby, S., "The Design of Production Systems, Reinhold Publication Corporation, New York, 1966, pp. 12.

176. Elmaghraby, S., "The sequencing of 'Related' Jobs," Naval Research Logistics Quarterly, Vol. 15, No. 1, 1968.

177. Elmaghraby, S., "The sequencing of n Jobs on m Parallel Processors," Research Memorandum, North Carolina State University, 1968.

178. Elmaghraby, S., "The Machine Sequencing Problem - Review and Extension," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 205-232.

179. Giffler, B., "Schedule Algebra: A Progress Report," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 255-280.

180. Glassey, C., "Scheduling Several Products on One Machine to Minimize Changeovers," Operations Research, Vol. 16, No. 2, 1968, pp. 342-352.

181. Kortanek, D., D. Sodaro, and A. Soyster, "Multi-Product Production Scheduling via Extreme Point Properties of Linear Programming," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 287-300.

182. Maxwell, W., "The Scheduling of Economic Lot Sizes," Naval Research Logistics Quarterly, Vol. 11, No. 1, 1964, pp. 89-124.

183. Maxwell, W., and M. Mehra, "Multiple-Factor-Rules for sequencing with Assembly Constraints," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 241-254.

184. Pierce, J. and D. Hartfield, "Production Sequencing by Combinatorial Programming," Operations Research and Management Information Systems (Editor, J. F. Pierce) (TAPPI, Technical Association of Pulp and Paper Industry, 1966) Chapter 17.

185. Porter, O., "The Gantt Chart as Applied to Production Scheduling and Control," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 311-318.

186. Pounds, W., "The Scheduling Environment," Chapter 7 of Reference 108.

187. Rogo, L., "Sequencing, Modeling, and Gantt Charting Repetitive Manufacturing," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1968, pp. 301-310.

188. Rogers, J., "A Computational Approach to the Lot Scheduling Problem," Management Science, Vol. 4, No. 3, 1958, pp. 284-291, pp. 437-447.

189. Rothkopf, M., "Scheduling Independent Tasks on Parallel Processors," Management Science, Vol. 12, No. 5, 1966, pp. 437-447.

190. Rothkopf, M., "Scheduling with Random Service Times," Management Science, Vol. 12, No. 9, 1966, pp. 707-713.

191. Spinner, A., "Sequencing Theory - Development to Date," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1966, pp. 319-324.

192. Soyster, A., "Multi-Product, Multi-Period Production Scheduling via Extreme Point Properties of Linear Programming and Some Horizon Posture Properties Overtime," M.S. Thesis, Cornell University, 1967.

193. Szwarc, W., "On Some Sequencing Problems," Naval Research Logistics Quarterly, Vol. 15, No. 2, 1966, pp. 127-156.

COMPARISON OF FLOWSHOP SCHEDULING ALGORITHMS

by

LAXMAN SINGH BENIWAL

B.E. (Mech.), Birla Institute of Technology & Science
Pilani, Rajasthan, India, 1965

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1969

The purpose of this report is to compare various algorithms for solving flowshop scheduling problems. The Enumeration, Direct and Branch-and-Bound II techniques have been considered. The computational algorithms of these techniques are discussed and illustrated by a sample problem. Several computational experiments have been conducted. The results obtained by the above techniques have been compared with those of Decomposition technique developed by Ashour, the Rounded Linear Programming reported by Giglio and Wagner and Branch-and-Bound I devised by Brown and Lomnicki. Direct algorithm generated optimal solution in all the problems; however the execution time spent increased rapidly with the increase in the size of the problem. For a problem involving twelve jobs and three machines, the storage location required exceeds the capacity of the computer, IBM 360/50.

The Branch-and-Bound method may or may not produce the optimal solution depending upon the number of the nodes to be explored. This technique is more flexible. The Decomposition approach with K equalling J/2 increases the capability of the Direct and Branch-and-Bound methods for solving the larger or more practical scheduling problems. Although the Rounded Linear Programming technique has a potentiality in the future, the resulting solutions do not seem sufficiently close to the optimal values to warrant the amount of the computational effort involved.