DETERMINATION OF OPTIMAL SEQUENCE AND SCHEDULE IN A SINGLE - MACHINE JOB SHOP : A GOAL PROGRAMMING APPROACH

Mahesh Chander Gupta, M.Com.

A thesis presented to the University of Manitoba in partial fulfillment of the requirements for the degree of Master of Science in the Department of Actuarial and Management Sciences, Faculty of Management

September, 1986 ©

DETERMINATION OF OPTIMAL SEQUENCE AND SCHEDULE IN A SINGLE-MACHINE JOB SHOP: A GOAL PROGRAMMING APPROACH

BY

MAHESH CHANDER GUPTA

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

CERTIFICATE

Certified that the work embodied in this thesis entitled, "DETERMINATION OF OPTIMAL SEQUENCE AND SCHEDULE IN A SINGLE-MACHINE JOB-SHOP: A GOAL PROGRAMMING APPROACH", by Mashesh Chander Gupta has been carried out under our supervision and that this has not been submitted elsewhere for a degree.

(Dr. Y.P. Gupta, Ph.D., P.Eng.)   (Dr. C.R. Bector, F.I.A.M.)

Department of Actuarial and Management Sciences
Faculty of Management
University of Manitoba, Winnipeg

September 8, 1986

ABSTRACT

Sequencing and scheduling refers to the mere ordering of a collection of jobs or tasks to be performed and the assignment of points or times for each job or task that stipulate when it is to be performed, respectively. In a totally deterministic environment, sequencing and scheduling are often refered interchangeably (Salvador [1977]).

Job scheduling and sequencing can be examined from either shop that will process the job or the customer placing the order for the job perspective. If the shop viewpoint is taken, the objective of the schedule determination is likely to be related to the minimization of one or more cost factors. When approached from the customer viewpoint, the objective will be related to the due date (Panwalker et al.[1982]).

Since the early work of Johnson[1954] to determine an optimal solution to minimize make-span for the two-machine flow-shop problems, the scheduling and sequencing literature has increased exponentially (e.g. Bakshi and Arora[1969], Eilon and King[1967], Gupta[1971], Panwalker and Iskiander[1977] etc.).

Recently, Kanet[1981] has addressed the problem of scheduling n-jobs on a single machine so as to minimize the absolute lateness where all jobs have a common due date, d, (d > MS), where MS being the makespan of all jobs. The objective of this problem is a non-regular per-

formance measure which contrasts with the more general class of regular performance measure. Applications of non- regular performance measure include file organization problems. Although non-regular performance measure are extremely important in a job-shop scheduling, very little analytical work has been done in this area (Kanet[1981], Sundararaghavan and Ahmed[1984] etc.). In this thesis, we will present theoretical developments based on goal programming theory on scheduling and sequencing of n jobs in a single-machine situation. When all jobs have common due date. The objective in this problem is to minimize the total absolute deviations from the due date. Kanet[1981] provided a heuristic to determine an optimal sequence of n-jobs. We propose an algorithm which provides an optimal sequence when (i) a common due date is given and (ii) an optimal due date is needed to obtain as well.

Kanet[[1981] and others assumed the penalty for earliness and tardiness of jobs as equal. However, it is more realistic to assume that tardiness is more expensive for a manufacturer due to the fact that he may not only incur penalty due to late delivery but also may have loss of goodwill to future orders. Panwalker et al.[1981] considered problem of finding of an optimal sequence and due date by minimizing the total penalty based on the due date value and the earliness and tardiness of each job. In this thesis, we propose the application of ST algorithm for the above problem. Our algorithm reveals that an optimal sequence and optimal due date can be found for which total penalty is substantially less than that obtain by the algorithm provided by Panwalker et al.[11981].

Cheng[1985] and Panwalker et al. [1982] proved that an optimal due date for a given sequence of jobs will always lie on the one of the completion times of n jobs. Cheng[1985] provided the proof using duality approach where as Panwalker et al. [1981] proved their lemma by logical analogy. It will be shown that using goal programming the above results can be proved in more elegant and simplified way.

It has been argued that more realistic problem in sequencing of jobs is when a due date is provided [Sundararaghavan and Ahmed[1984] They developed a heuristic to determine a sequence when due date is given. However their heuristic does not provide optimal sequence in certain number of cases. It will be shown that ST algorithm can provide an optimal sequence.

Kanet[1981] provided a heuristic (SMV) to determine an optimal sequence by minimizing sum of completion times variances. We suggest another criterion i.e. minimization of variance around a common due date when due date is greater than or equal to make-span. Kanet[1981] also provided an algorithm to develop an optimal schedule when the due date, d > MS, is given. He used Gantt chart approach in developing an optimal schedule. When the number of jobs to be scheduled increases, the gantt chart approach tends to become cumbersome. We propose an goal programming approach to determine an optimal schedule.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor C.R. Beector and Dr. Y.P. Gupta, Head of Department, for their continuing interest, guidance and supervision throughout this work. Their ready availbility and many constructive suggestions were greatly appreciated.

The other members of my committee, Professor A. Kusiak and Professor S. K. Bhatt, provided invaluable support through their encouragement and their helpful comments.

I express my deepest regards for Yog Gupta and Kiran Gupta, my uncle and aunti, for their badly needed moral and financial support without which the present work would have been difficult.

Mahesh C. Gupta

Winnipeg, Manitoba

CONTENTS

Chapter I

INTRODUCTION

Sequencing and scheduling of jobs with limited resources is an important field of operations management. Sequencing problems are very common and are encountered whenever a choice in the order of performing operations (or jobs) exists. Some examples of such problems are (Conway et al.[1967]): jobs in a manufacturing plant, nurses in a hospital, aircrafts waiting for landing clearance, bank customers at a teller's window and computer programs waiting to be processed by a computing center.

The term 'sequencing' refers to the ordering of a collection of jobs to be performed. The term 'scheduling' refers to the assignment of times for each job that specify when it is to be performed. Conway et al. [1967] draws a distinction between the two terms by stating that sequencing is concerned with the ordering of operations on a single machine whereas scheduling is a simultaneous and synchronized sequence on several machines. However, the two terms are used synonymously in much of the literature reviewed in this study. Most of the research in scheduling theory has been directed towards deterministic scheduling which is simply a sequencing problem. Under deterministic conditions, the schedules are determined by the sequence of processing times of the jobs. Under probabilistic conditions a schedule cannot be determined but a scheduling methodology based on dispatching rules or priorities may be proposed.

A deterministic scheduling problem is one in which (a) all job characteristics such as number of jobs, their availability times and processing times, and (b) all shop characteristics, such as number of machines and their capacity availability, are fixed and known in advance. A probabilistic scheduling problem is one in which any of the job or shop characteristics vary stochastically i.e. according to some probability distribution function.

Scheduling problems have been classified into (i) single-machine cases and (ii) multiple-machine cases and the latter are further categorized into three classes as follows (Salvador[1977]) :-

(a) **Single-stage parallel machines class:** In this class all jobs require only one operation, which can be performed by any machine in the shop.

(b) **Multi-stage flow shop class:** In this class some jobs involve more than one operation but number and sequence of operations for each job to be processed are fixed.

(c) **Hybrid shop class:** All other cases are put under this category. An example of a hybrid shop to consider is the straightforward generalization of the previous two types of class i.e. multi-stage flow shops with parallel machines at each stage.

Baker[1974] has advocated the significance of scheduling decisions in the hierarchy of management decisions such as (i) type of product to be produced, and (ii) type of resource configuration available. The methodology developed for scheduling and sequencing problems attempt to provide solutions within the framework of managerial decisions. In other words, scheduling decisions are secondary to the fundamental managerial decisions.

The majority of the existing literature in sequencing and scheduling deals with job-shop scheduling problems. Conway et al. [1967] has discussed the basic definitions involved. These are summarized as follows :

Machine : A facility capable of performing some operation.

Job-shop : The set of all the machines that are identified with a given set of jobs.

Operation : An elemental task to be performed which will use the services of a machine for a certain period of time. Each operation must be 'processed' on a specific 'machine' for a specific duration unique to the job and machine.

Job-shop Process : This process consists of the machines, the operations, and a statement of the rule that restrict the manner in which an operation can be assigned to the corresponding machine.

Thus, a generalized **job-shop scheduling problem** consists of finding an order of the operations to be performed on each machine in a job-shop subject to the job-shop process specifications as well as any other constraints, and such that some measurable function of that ordering is optimized.

In the context of the job-shop process, it should be noted that job-shop scheduling problems are not only restricted to the manufacturing environment. Some examples of other areas in which job-shop scheduling concepts have been applied are as follows :

(i) Scheduling of classes(jobs) in an educational institution for the available lecture rooms(machines).

(ii) Scheduling of patients(jobs) in a hospital to be examined by doctors(machines).

(iii) Scheduling of computer programs(jobs) to be processed by processors(machines).

However, applying job-shop concepts in the above environments may require some restrictions (assumptions) on (a) the definitions of the job set, (b) the definitions of the machines, and (c) the manner in which the schedule is developed. These restrictions (assumptions) in a job-shop scheduling problem are discussed in the section 1.2. In section 1.1 we define the notation to be used throughout thesis.

1.1 **NOTATION**

We define the most commonly used notation in scheduling problems, which will be used in this thesis to formulate mathematical models :-

J_i = the th job, i = 1,2,3,....n

r_i = the ready time of the ith job, i.e. the time at which the ith job becomes available for processing.

p_i = the processing time of the ith job.

d_i = due date, i.e.the promised delivery date of the ith job.

a_i = the allowance for the ith job,the time interval allowed for processing the job between ready time and due date.

k = constant flow allowance to be used to determine due dates.

w_i = the waiting time of the ith job.

C_i = the completion time of the ith job.

F_i = the flow-time of the ith job.

L_i = the lateness of the ith job, i.e. the difference between its completion time and its due date.

[i] =the job at the ith position in the given sequence.

T_i = the tardiness of ith job: $T_i = \max(L_i, 0)$.

E_i = the earliness of ith job: $E_i = \max(-L_i, 0)$.

1.2 ASSUMPTIONS

French[1982] has advocated a comprehensive list of assumptions. For a single-machine deterministic problems, these are summarized as follows :

1.2.1 Assumptions on Jobs

(i) Each job is an entity and is processed to its completeness.

(ii) Pre-emption of jobs is not permitted. Each job, once started, must be completed before another operation may be started on the same machine.

(iii) All jobs are available for processing at the same time.

(iv) Job cancellation is not allowed. Each job must be processsed to completion.

(v) In-process inventory is allowed. The jobs may wait for the next machine to be free.

1.2.2 Assumptions on Machines

(i) Machine may be idle.

(ii) Machine break-down or repair does not occur.

1.2.3 Assumptions on Processing Times

(i) The processing times are independent of the schedule i.e. each set up time is sequence independent and is included in the processing time and the time to move jobs between machines is negligible.

(ii) The processing time of each operation is given at the outset and is constant regardless of its order of processing.

1.2.4 Other Assumptions

(i) The technological constraints are known in advance and are immutable.

(ii) There is no randomness i.e. the number of jobs, number of machines, and the ready times are known and fixed. All other quantities needed to define a particular problem are also known and fixed.

The above assumptions are undoubtedly restrictive and they limit the structure of job-shop problems. Nevertheless, these assumptions are important and usually some of them are relaxed while addressing a typical deterministic scheduling problem and developing a model to solve it. For example, Cheng[1985] and Panwalker et al.[1982].

1.3 COMMON SCHEDULE EVALUATION CRITERIA

The most critical factors in any scheduling problem are the schedule evaluation criteria. These may be either job related or shop related. The most commonly used criteria are described below.

1.3.1 JOB RELATED CRITERIA

1.3.1.1 Criteria Based Upon Completion Times

(i) Minimize the maximum flow-time: This implies that the schedule's cost is directly related to its longest job.

(ii) Minimize the maximum completion time: This implies that the schedule's cost depends on how long the processing system is devoted to the entire set of jobs. The total sum of processing times of all jobs is called 'makespan' (MS).

(iii) Minimize the total waiting time: This implies that the schedule's cost depends upon the portion of flow-time that the jobs spend waiting for their operations to be performed.

(iv) Minimize mean flow-time or mean completion time: This implies that the schedule's cost is directly related to the average amount of time it takes to process a single job. When ready time is equal for all jobs, minimizing the mean flow-time is equivalent to minimizing the mean completion time.

1.3.1.2 Criteria Based Upon Due-Dates

(i) Minimize either mean lateness or maximum lateness: This implies that a schedule's cost is related to how much we miss target dates and i~~s~~t is appropriate when there is a positive reward for completing a job earlier.

(ii) Minimize either mean tardiness or maximum tardiness: This implies that a schedule's cost is related to how much we miss target dates it is appropriate when early jobs bring no rewards.

(iii) Minimize the missed number of due dates: This implies that the schedule's cost is related to both the earliness or tardiness of the jobs. It is appropriate when the costs are incurred whether the job is early or tardy.

1.3.2 SHOP RELATED CRITERIA

(i) Minimize the mean number of jobs waiting for processing or the mean number of unfinished jobs : This implies that the schedule's cost is related to the in-process inventory costs.

(ii) Minimize mean idle time or maximum idle time: This implies that a schedule's cost is related to the efficient utilization of machines. Here mean and maximum are taken over the machines, and not the jobs.

These performance measures are classified as : (1) Regular Measures, and (2) Complex Measures.

1. **Regular Measures:** A regular measure is a value to be minimized that can be expressed as a function of job completion times. It increases only if at least one of the completion-times increases. It can be verified that mean completion-time, maximum completion-time, mean flow-time, maximum flow-time, mean tardiness and maximum tardiness satisfy these conditions.
2. **Complex Measures:** In these types of performance measures, weighted averages, combinations of average and maximum, or functions of fractiles are considered.

Recently, Eilon and Chowdhary[1977], Schrage[1975], Kanet[1981], Panwalker et al. [1982], Cheng[1985,1986] and many other have considered minimization of variance or average deviations as evaluation criteria. Variance of flow-time, completion-time, and also of waiting time, for example, have been addressed. Many of these studies are related to this thesis. The problem of determining optimal due date, sequence and optimal schedule have been addressed using different methods of assigning due dates to the set of jobs to be processed. Conway et al.[1967] ,and Baker and Bertrand [1981] have commented on due date assignment methods. The basic definitions of these methods are as follows :-

CON --> Jobs are given constant flow allowances, so that ($d_i = r_i + k$). Thus, when all jobs are available at the same-time, the due date is simply a constant k.

SLK --> Jobs are given flow-allowances that reflect equal waiting-times or equal slack, so that ($d_i = r_i + p_i + k$).

TWK --> Jobs are given flow-allowances that are proportional to the total work they require, so that ($d_i = r_i + k(p_i)$).

1.4 SCOPE OF THESIS

In this study we confine our discussion to the deterministic single-machine sequence schedule, and due date determination problems. The justification for studying the single- machine problems is provided by several authors, such as Conway et al. [1967], Elmaghraby[1968], and Day and Hottestein[1970]. They argue that i) several operational systems such as paint manufacturing and data processing can be modeled as a single-processor system and ii) there is always the possibility that the study of this single-machine case may lead to the solution of more general and complex scheduling problems. More specifically, this thesis deals with the following problems :-

(1) The determination of an optimal sequence when a common due date is provided. Goal Programming is used to minimize the total penalty function. The satisfactory achievement of due dates is perhaps the most significant measure of performance. To quote Conway et al. [1967, p. 229] :

" Of the various measures of performance that have been considered in research on sequencing, certainly the measure that arouses the most interest in those who face practical problems of sequencing is the satisfaction of preassigned job due dates. Equipment utilization, work-in process inventory, and job flow-time are all interesting and more or less important, but the ability to fulfill delivery promises on time undoubtedly dominates these other considerations."

(2) (a) Given a common due date ($d \geq MS$), determine an optimal sequence.

(2) (b) Given an optimal sequence and a common due date, ($d \geq MS$), determine an optimal schedule.

(3) Given an n jobs along with their processing times, determine an optimal sequence and an optimal due date when the objective is to minimize the penalty function. The penalty for tardy jobs is more severe than that for early jobs.

Chapter II

LITERATURE REVIEW

Job-shop scheduling literature is richly endowned with bibliographical and review works. Some of the noteworthy attempts include : Bakshi and Arora[1969], Day and Hottestein[1970], Eilon and King[1967], Eilon[1979], Elmaghraby [1968], Godin[1978], Graham et al.[1979], Graves[1981], Gupta[1971], Jeremiah and Schrage[1964], Mellor[1966],Moore and Willson[1967], Panwalker and Iskiander[1971] and Salvador[1978]. Several textbooks contain extensive lists of references dealing with scheduling. These include Aho, Hopcroft and Ullman[1974], Baker[1974], Coffman[1976], Conway et al. [1967], French[1983] and Miller and Thatcher[1972].

Since the scope of this thesis is limited to sequencing scheduling and due-date determination of a static, single-machine job-shop problems, and the vast amount of literature dealing with scheduling and sequencing problems exists, it is not practical to review the entire literature here. Therefore, only deterministic single-machine problems of the classification scheme shown in Figure # 2.1 will be reviewed in depth.

A CLASSIFICATION SCHEME

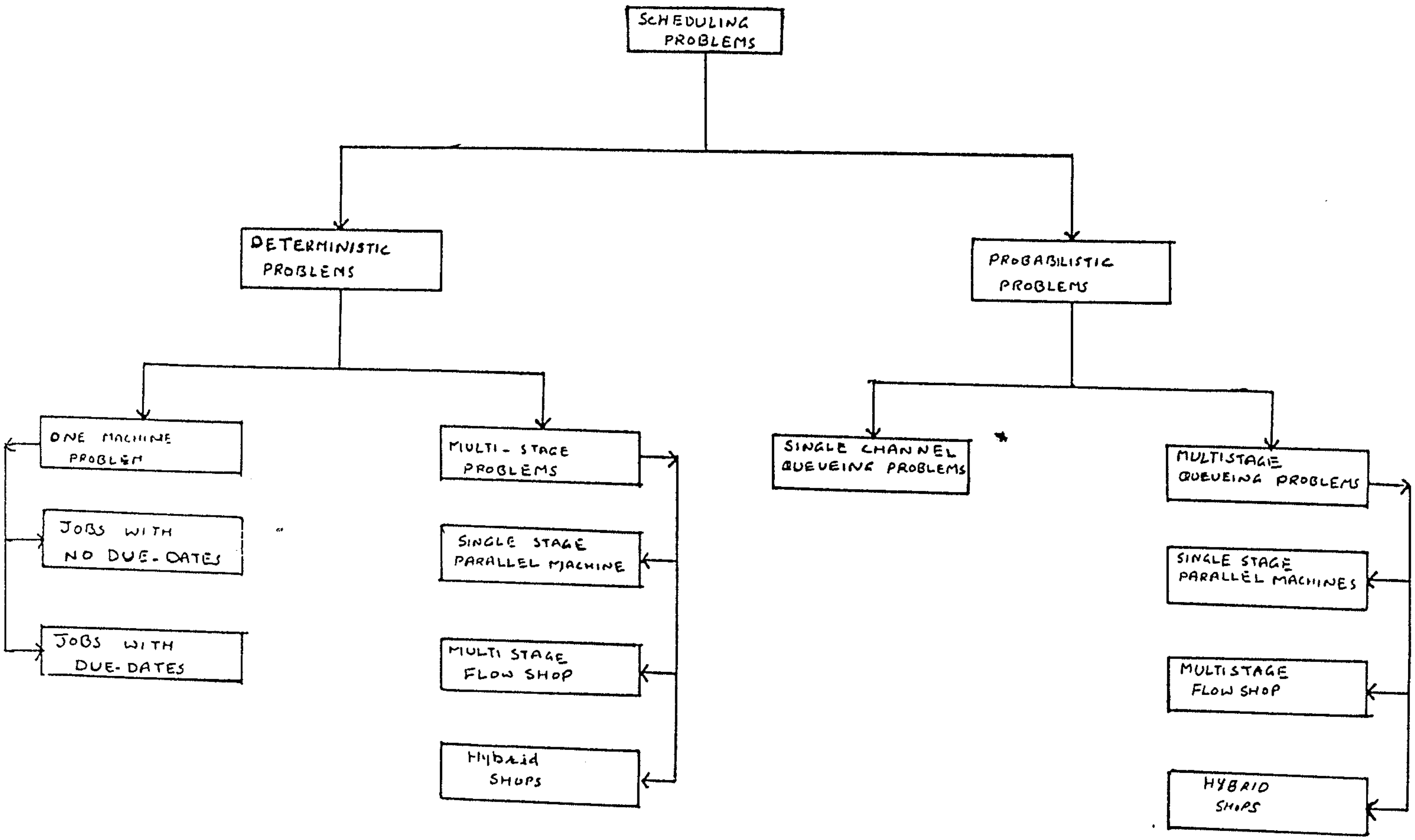


TABLE 2.1 (CHAPTER 2, p. 12)

2.1 BASIS OF LITERATURE REVIEW

In the literature several schedule, sequence and due-date criteria are used which form the basis of this review.

1. **Number of late jobs :** Some of the earliest work in minimizing the number of late jobs on one machine can be found in Moore[1968]. He considered this objective function under the constraint that all jobs have different due-dates. Maxwell[1970] presented an integer-programming formulation for the problem addressed by Moore[1968]. Hodgson[1977] provided an optimizing algorithm for minimizing the number of tardy jobs. Hodgson states that, if the sequence is arranged in the order of non-decreasing due-dates and yields at most one tardy job, then the current sequence is optimal for the problem. Sturm[1970] developed a simple proof of optimality for Moore's sequencing algorithm.

2. **Total tardiness cost :** The minimization of total tardiness cost is criterion related to the number of late jobs. McNaughton[1959] provided a simple procedure for finding the minimum cost schedule when the deferal costs are all linear. The deferal costs may be associated with 'relative' and 'absolute' deadlines. He advocated that the n jobs should be ordered according to P_i / p_i(i.e. the ratio of the penalty of th job to its processing-time) with the job having the largest ratio being performed first. Schild and Fredman[1961] generalized McNaughton's results by relaxing the restrictive assumptions. Lawler[1964] applied dynamic programming and linear programming

techniques to the single-processor as well as multiple-processors when the deferal costs are non-linear. Elmaghraby[1968] used the same criterion for sequencing n jobs which are related to each other in such a manner that regardless of which job is done first, its utility is hampered until all other jobs in the subset of related jobs are also completed. He provided an algorithm using dynamic programming. Shwimer[1972] proposed a branch and bound ('BAB') algorithm for solving n jobs sequence-independent problems. On the other hand, Emmons[1969] minimized the total tardiness. The criterion of minimization of mean tardiness or maximum tardiness is only appropriate when the jobs processed earlier than their due-date bring no rewards. On the other hand, if positive rewards are associated with completing a job earlier, then minimizing the mean lateness or the maximum lateness are considered to be appropriate criteria.

3. **Total penalty** : Sidney[1977] advocated that the costs arising from both earliness and tardiness of individual jobs must be considered in the scheduling process. He presented an algorithm to minimize the total penalty subject to restrictive assumptions on the target start times, the due-dates, and the penalty function. Lakshminarayan et al. [1978] proposed an improved algorithm which reduced the number of computations by (n^2 - n log n). Panwalker et al. [1982] proposed a generalized penalty function which also includes the cost of achieving the due-date and an algorithm to find an optimal sequence and an optimal due-date with an objective of minimization of the total penalty.

Gilmore and Gomory[1964] presented another version of cost related criteria, namely minimization of total processing costs. The algorithm suggested by them is a special case of the travelling-salesman problem.

4. **Variance :** Minimization of the variance was the criterion originally suggested by Mertern and Mellor[1972]. They argued that in situations such as computing systems with large data files, often the response time to user's request is strongly dependent on the time required to access or retrieve the data files referenced by the users. It is often desirable to provide uniform response to user's request which means minimizing the variance of the response time. They further argued that, in the job-shop scheduling problems, the variance performance function will lead to some insight into non-linear performance functions. In their paper, it was shown that a sequence that minimizes the variance of flow-time is an anti-thetical to a sequence that minimizes the variance of waiting time. Scharge[1975] uses the same minimization criterion and proved the following : a) The variance minimizing schedule for a finite job-set has the longest-job first, and b) the time-in-system variance for the schedule (1,2,3,...n-1,n) is the same as that for the schedule (1,n,n-1,...,3,2) i.e. the order of the last n-1 jobs can be reversed without affecting the variance. Eilon and Chowdhary[1977] presented a pioneering proof which states that the optimal sequence must be V-shaped, thus strengthening the results of Scharge[1975]. They minimized the variance of job-waiting time and presented four comparatively

simple heuristics to determine an optimal sequence. Kanet[1981] proposed another procedure to minimize the completion-time variance. He compared the performance of his heuristic, SMV, with that of Eilon and Chowdhary[1977] and concluded that SMV has good properties in that it yields optimal solutions for n less than or equal to 5 and compares favorably to the optimal solution for large values of n.

5. **Lateness:** Apart from the problem of minimizing the variances of waiting and completion times there are those problems in which non-regular measure is a function of job-lateness. This lateness is measured on the basis of i) the sum of absolute lateness and ii) the sum of the squared lateness. Kanet[1981] determined an optimal schedule when all jobs have a common due-date which is greater than or equal to the sum of the processing times of all jobs. He minimized the average deviations of job completion times using a heuristic which is essentially one developed by Eilon and Chowdhary[1977]. Sundararaghavan and Ahmed[1984] extended the results of Kanet[1981] to the problem of scheduling n jobs on m parallel identical machines in order to minimize the sum of absolute lateness. Also they proposed an algorithm which is more general than Kanet's algorithm and has no restriction on the common due-date i.e.due-date may be less than MS. Gupta and Sen[1983] minimized a quadratic function of job lateness and presented an algorithm based on branch and bound technique. They also presented a heuristic rule which gives a near optimal solution

6. **Mean flow-time :** Smith[1956] showed that the average flow-time is minimized by sequencing the jobs in order of non-decreasing processing-times, i.e. $p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]}$.This result has been proved in different by a number researchers in different ways (French[1982]). Baker[1974] has shown that the weighted mean flow-time is minimized by Weighted Shortest Processing Time (WSPT) sequencing, i.e. ($p_{[1]}/w_{[1]} \leq p_{[2]}/w_{[2]} \leq .. \leq p_{[n]}/w_{[n]}$).
7. **Total cost of production and inventory :** Recently, some research has been done on single-machine scheduling problems which primarily focused on the performance criteria such as minimization of flow-time, tardiness or total tardiness. It has been argued that a so called optimum with respect to one objective, could perform extremely poorly with respect to other criteria. Therefore, a 'non-optimal' solution with satisfactory performance on other measures might be considered as a better alternative. This point was partly recognized by some researchers who studied scheduling problems with secondary criteria. Some examples are provided by Smith[1956], Emmons[1975] and Wassenhove and Gelders[1978]. These studies identify the best sequence for the secondary measure from among the set of alternative optima with respect to the primary measures. Recent studies such as Wassenhove and Gelders[1980]] Fry and Leong[1986] have argued that any relevant cost measure should include both a tardiness measure, to represent customer-related performance, and a flow-time measure to represent inventory performance. Smith[1956] presented an algorithm minimizing weighted completion times subject to the constraint that

all jobs were completed by their due-dates. Heck and Roberts[1972] presented another algorithm for minimizing the sum of completion times subject to 'not increasing the maximum tardiness calculated by the due-date sequence'. They claimed that their result could be extended to the problem of weighted completion-times in a manner similar to Smith's algorithm. However, Burns[1976] presented a counter example to Smith's algorithm and suggested an improved algorithm that converges to a local optimum for both problems. Wassenhove and Gelders[1978] addressed the same bi-criterion scheduling problem and presented an algorithm to solve it. Recently, Fry and Leong[1986] has presented a mixed-integer programming model where the objective function is a combination of total tardiness and earliness. However, it is not efficient to solve a large problem and a branch and bound method may be developed.

In scheduling literature, due-date determination is one of the most popular area which has received enormous attention among scheduling researchers and for which abundant fruitful results have been derived. Recently, Sen and Gupta[1984] have made an excellent synthesizational effort. They reviewed important theoretical developments, computational experience and applications of scheduling problems involving due-dates. Conway[1965] was the first to compare the effectiveness of various due-date assignment methods with respect to a number of performance measures. Conway et al.[1967], Jackson[1955], and Smith[1956] proved that the maximum tardiness, is minimized by processing jobs in the order of non-decreasing due-dates, if all jobs have the same ready times. While Eilon and Chowdhary[1976], Weeks and Fryer[1977], and Weeks[1977] have approached the problem of optimal due-date setting by applying computer simulation techniques, Baker and Bertrand[1980], Seidman and Smith [1981], Seidman et al.[1981], Panwalker et al.[1982] and Cheng[1984,1985] have carried out research on analytical determination of optimal due-dates.

2.2 APPROACHES TO DETERMINISTIC SINGLE-MACHINE PROBLEMS

Deterministic scheduling problems have been dealt with using different methodologies. Elmaghraby[1968] has suggested four approaches in which the scheduling literature can be categorized. The basic concepts underlying these methods are summarized as follows along with some examples :-

1. **Combinatorial Approaches.** This refers to techniques that involve 'switching ' pairs of jobs of a given sequence in a controlled fashion until the resulting sequence is optimal. The basis of this approach is generalized by Smith[1956].

   McNaughton[1959], Gilmore and Gomory[1964], and many others have

2. **Mathematical Programming Approach.** This refers to linear, dynamic, integer, quadratic, and goal programming techniques as well as some applications of network theory and lagrangian methods. The integer programming models (e.g. Maxwell[1970], Fry and Leong[1986]) and dynamic programming models (e.g.Lawler[1964],Elmaghraby[1968]) have been extensively manipulated for scheduling problems. Recently, bi-criterion

scheduling has also been addressed. Wassenhove and Gelders[1980] and Fry and Leong[1986] are examples of single machine scheduling problems.

3. **HEURISTICS.** This is also known as 'combinatorial programming' or 'controlled enumeration'. Elmaghraby[1968] has stated that these are problem-solving procedures developed on the basis of two principles: i) the use of a controlled enumeration technique for considering all potential solutions which are known from dominance and bounding, and ii) feasibility considerations should be acceptable.

   Branch and bound ('BAB') is another name for such approaches and has been extensively used to determine optimal solutions for scheduling problems. Moore[1968], MaNaughton[1959], Shwimer[1972], Kanet[1981], Emmons[1975] and Gupta and Sen[1980] are a few examples.

4. **Simulation Approach** : A final alternative is the random search for optimal sequence via Monte carlo sampling. In his paper, Heller[1960] spawned the technique by generating 3000 random sequences for a flow-shop of 10 machines. However, recently this technique has been widely used for due-date determination problems of scheduling. Eilon and Chowdhary[1977], Weeks[77], Weeks and Fryer[77] are some of the examples.

Chapter III

A GOAL PROGRAMMING MODEL APPROACH

One-machine Sequencing & Scheduling Problem

- SOME THEORETICAL DEVELOPMENTS

In the present chapter we consider an n-job one machine scheduling problem in which all jobs have a common due date and penalties are incurred when a job is completed before or after the due date. We have either of the following two objectives.

1. To determine the optimal value of this common due date and an optimal sequence to minimize a total penalty function, or
2. When some predetermined value of the common due date is given, to determine a corresponding optimal sequence to minimize a total penalty function.

In the existing literature different penalty functions have been considered e.g. Sidney[1977], Kanet[1981], Panwalker et al.[1982], and Cheng[1985]. The penalty function suggested in Panwalker et al.[1982] and Cheng[1985] is based on the due date value and the earliness and tardiness of each job. The per unit costs involved in their papers are all linear. It is this linear cost function associated with earliness and tardiness that in fact suggests the use of a goal programming model approach to solve such a problem.

GOAL PROGRAMMING MODEL FORMULATION : Let

$N = \{1,2,...,n\}$, the set of n jobs,

p_i = the processing time of job $i \in N$,

π = the set of n! sequences generated out of n jobs in set N,

σ = an arbitrary sequence in π,

R = the set of real numbers.

ASSUMPTIONS :

(i) the job labelling is such that

$$p_1 \leq p_2 \leq \cdots\cdots \leq p_n$$

i.e. sequence 1,2,....,n represents the **SPT** sequence,

(ii) all jobs have a common due date k.

Let

$J_i(\sigma)$ = the job in position i in σ,

C_i = the completion time of $J_i(\sigma)$,

d_{i1} = the tardiness of $J_i(\sigma)$,

d_{i2} = the earliness of $J_i(\sigma)$,

w = unit penalty associated with due date,

w_{i1} = unit penalty associated with tardiness of $J_i(\sigma)$,

w_{i2} = unit penalty associated with earliness of $J_i(\sigma)$,

We now associate the following **Linear Goal Programming Problem (LGP),** with our main problem.

(LGP) Minimize $Z = wk + \sum_{J_i(\sigma)} w_{i1}\, d_{i1} + \sum_{J_i(\sigma)} w_{i2}\, d_{i2}$ (3.1)

subject to

$$k + d_{i1} - d_{i2} = C_i \quad \text{for all } J_i(\sigma) \quad (3.2)$$

$$k, d_{i1}, d_{i2} \geq 0 \quad \text{for all } J_i(\sigma) \quad (3.3)$$

$$k,\ d_{i1}, d_{i2} \in R.$$

We now prove the following results using **(LGP).**

LEMMA 1 (Panwalker et al[1982], Cheng[1985]) : For any specified sequence σ, there exists an optimal value of **k** which coincides with the completion time of one of the jobs in σ.

Proof We will give two alternative proofs for this lemma.

1. From **Linear Goal Programming** theory we know that at most one of d and d is positive for each J. Also using linear programming theory in **(LGP)** it is known that the optimal solution, if it exists, will be taken on at one of the extreme points of the **(LGP)** constraints. This implies that for at least one constraint (say the rth constraint) in **(LGP)** both d_{r1} and d_{r2} must be zero. This implies $d_{r1} = 0$ and $d_{r2} = 0$. This in turn yields $k = C_r$.
2. We know that in n-job single machine scheduling problem the total number of possible sequences is n! and one of them is an optimal sequence, which can be achieved (by numeration, say). If possible, let $k \neq C_i$ for all J_i (σ)

$$\begin{aligned} \Rightarrow \quad k &> C_i \quad \text{for all } J_i \in S_1 \\ k &< C \quad \text{for all } J_i \in S_2 \end{aligned} \qquad (3.4)$$

where $S_1 \cup S_2 = \sigma$.

This implies that the constraint set of **(LGP)** is an open set. Hence the optimal will not be achieved, which is a contradiction.

Relating (LGP) to the Problem Of Minimization of the Weighted Average of the Absolute Value of Job Lateness : Cheng[1985] considered the problem of minimization of average missed due dates as objective. For this purpose he adopted the weighted average of absolute lateness as the objective function to be minimized. According to Cheng[1985], for a given σ ,let $L_{[i]}$,$C_{[i]}$($C_{[i]} = \sum_{j=1}^{i} p_j$) and $d_{[i]}$ denote the lateness, completion time and due date respectively of the job in position [i]. The objective function is expressed as (Cheng[1985]) :

$$
\begin{aligned}
f(k) &= \sum_{i=1}^{n} w_{[i]} \left| L_{[i]} \right| \\
&= \sum_{i=1}^{n} w_{[i]} \left| C_{[i]} - d \right| \\
&= \sum_{i=1}^{n} w_{[i]} \left| C_{[i]} - k \right| \qquad (3.5)
\end{aligned}
$$

where w ($0 < w_i < 1; \sum_{i=1}^{n} w_i = 1$) is a weighting factor for $J_i(\sigma)$.

let

$$d_{i1} = \begin{cases} C_{[i]} - k & \text{if } C_{[i]} > k \text{ for all } J_i(\sigma) \\ 0 & \text{otherwise} \end{cases}$$

$$d_{i2} = \begin{cases} k - C_{[i]} & \text{if } k > C_{[i]} \text{ for all } J_i(\sigma) \\ 0 & \text{otherwise} \end{cases}$$

$$w_{[i]} = \begin{cases} w_{i1} & \text{if } C_{[i]} > k \text{ for all } J_i(\sigma) \\ 0 & \text{otherwise} \end{cases}$$

$$w_{[i]} = \begin{cases} w_{i2} & \text{if } k > C_{[i]} \text{ for all } J_i(\sigma) \\ 0 & \text{otherwise} \end{cases}$$

The problem of minimizing (3.5) now becomes

$$\text{MIN } Z = \sum_{J_i(\sigma)} w_{i1} d_{i1} + \sum_{J_i(\sigma)} w_{i2} d_{i2}$$

subject to

$$k + d_{i1} + d_{i2} = C_i \qquad \text{for all } J_i(\sigma)$$

$$k, d_{i1}, d_{i2} \geq 0 \qquad \text{for all } J_i(\sigma)$$

which is same as **(LGP)** for $w = 0$.

3.1 SOME THEORETICAL DEVELOPMENTS

3.1.1 Optimal Tableau Of (LGP) of a Givan Sequence

To do analysis on the **slack and surplus** variables, we first change the model (LGP) to the problem (LGP1) below by replacing each equality constraint by two inequalities of opposite signs, and then to the problem (LGP2) by introducing the slack variables, y , and surplus variables, y .

(LGP1) Minimize $Z = wk + \sum_{J_i(\sigma)} w_{i1} d_{i1} + \sum_{J_i(\sigma)} w_{i2} d_{i2}$

subject to

$k + d_{i1} - d_{i2} > C_i$ for all $J_i(\sigma)$

$k + d_{i1} - d_{i2} < C_i$ for all $J_i(\sigma)$

$k, d_{i1}, d_{i2} \geq 0$ for all $J_i(\sigma)$

$k, d_{i1}, d_{i2} \in R$ for all $J_i(\sigma)$

(LGP2) Minimize $Z = wk + \sum_{J_i(\sigma)} w_{i1} d_{i1} + \sum_{J_i(\sigma)} w_{i2} d_{i2}$

subject to

$k + d_{i1} - d_{i2} - y_{i1} = C_i$ for all $J_i(\sigma)$

$k + d_{i1} - d_{i2} + y_{i2} = C_i$ for all $J_i(\sigma)$

$k, d_{i1}, d_{i2}, y_{i1}, y_{i2} > 0$ for all $J_i(\sigma)$

$k, d_{ij}, y_{ij} \in R$ for all $J_i(\sigma)$, $j = 1, 2$

Thus, to obtain optimal table for σ, each constraint i for $J_i(\sigma)$ [(LGP) model] is replaced by two inequality constraints [(LGP1) model] with y_{i1} as **slack** variables for '$\leq$' constrains and y_{i2} as **surplus** variable for '$\geq$' constraint for $J_i(\sigma)$ [(LGP2) model]. This process results in-

the total number of constraints = 2n

and the total number of variables = 4n + 1,

since the number of constraints is 2n,

the total number of basic variables $= 2n$

For the sequence σ_{SPT}, which is in SPT (assumption 1, p. 21), let the position of the optimal due-date in the optimal tableau (3.1) be the rth. This implies that jobs completed before the optimal due date for σ_{SPT}, ODD(σ_{SPT}), are early and jobs completed after ODD(σ_{SPT}) are tardy. Hence the ODD(σ_{SPT}) at the rth position divides the SPT into two subsets E and T where

E denotes the set of early jobs with E_i as the earliness of $J_i(\sigma_{SPT})$ job, and

T denotes the set of tardy jobs with T_j as the tardiness of $J_j(\sigma_{SPT})$ job.

3.1.2 BASIC VARIABLES BEFORE ODD(σ)

The basic variables in the optimal tableau before ODD(σ) are:

$$= 2r - 1$$

And they are as:

$$d_{12}, d_{22}, d_{32}, \ldots, d_{r-1,2}$$

$$y_{11}, y_{21}, y_{31}, \ldots, y_{r1}$$

so there are $2r - 1$ of them.

3.1.3 BASIC VARIABLES AFTER OOD(σ)

The basic variables in the OT(3.1) after ODD(σ) are :

$$d_{r+1,1}, d_{r+2,1}, \ldots, d_{n1}$$

$$y_{r+1,2}, y_{r+2,2}, \ldots, y_{n2}$$

hence there are $2n - 2r$ of them.

OPTIMAL TABLEAU FOR n-JOBS PROBLEMS [OT(3·1)]

		w	w_1	w_{12}	w_{21}	w_{22}		$w_{r-1,1}$	$w_{r-1,2}$	w_{r1}	w_{r2}	$w_{r+1,1}$	$w_{r+1,2}$		w_{n1}	w_{n2}	0	0	0	0		0	0	0	0	0	0		0	0	0	0	
C_B	X_B	K	d_{11}	d_{12}	d_{21}	d_{22}	…	$d_{r-1,1}$	$d_{r-1,2}$	d_{r1}	d_{r2}	$d_{r+1,1}$	$d_{r+1,2}$		d_{n1}	d_{n2}	y_{11}	y_{12}	y_{21}	y_{22}	…	$y_{r-1,1}$	$y_{r-1,2}$	y_{r1}	y_{r2}	$y_{r+1,1}$	$y_{r+1,2}$	…	$y_{n-1,1}$	$y_{n-1,2}$	y_{n1}	y_{n2}	RHS
0	y_{11}	0	0	0	0	0	--	0	0	0	0	0	0	----	0	0	1	1	0		---	0	0	0	0	0	0	---	0	0	0	0	0
w_{12}	d_{12}	0	-1	1	0	0	--	0	0	1	-1	0	0	----	0	0	0	-1	0		---	0	0	0	1	0	0	---	0	0	0	0	b_1
0	y_{21}	0	0	0	0	0	--	0	0	0	0	0	0	----	0	0	0	0	1	1	---	0	0	0	0	0	0	---	0	0	0	0	0
w_{22}	d_{22}	0	0	0	-1	1	--	0	0	1	-1	0	0	----	0	0	0	0	0	-1	---	0	0	0	1	0	0	---	0	0	0	0	b_2
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0	$y_{r-1,1}$	0	0	0	0	0	--	0	0	0	0	0	0	----	0	0	0	0	0	0	----	1	1	0	0	0	0	---	0	0	0	0	0
$w_{r-1,2}$	$d_{r-1,2}$	0	0	0	0	0	--	-1	1	1	-1	0	0	----	0	0	0	0	0	0	----	0	0	0	1	0	0	---	0	0	0	0	b_{r-1}
0	y_{r1}	0	0	0	0	0	--	0	0	0	0	0	0	----	0	0	0	0	0	0	----	0	0	1	1	0	0	---	0	0	0	0	0
w	k	1	0	0	0	0	--	0	0	1	-1	1	-1	----	0	0	0	0	0	0	----	0	0	0	1	0	0	---	0	0	0	0	b_r
$w_{r+1,1}$	$d_{r+1,1}$	0	0	0	0	0	--	0	0	-1	1	0	0	----	0	0	0	0	0	0	---	0	0	0	-1	-1	0	---	0	0	0	0	b_{r+1}
0	$y_{r+1,2}$	0	0	0	0	0		0	0	0	0	0	0		0	0	0	0	0	0		0	0	0	0	1	1	---	0	0	0	0	0
⋮		⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	$y_{n-1,2}$	0	0	0	0	0	—	0	0	0	0	0	0	---	0	0	0	0	0	0	---	0	0	0	0	0	0	---	1	1	0	0	0
w_{n1}	d_{n1}	0	0	0	0	0	--	0	0	-1	1	0	0	---	1	-1	0	0	0	0	---	0	0	0	-1	0	0	---	0	0	1	0	b_n
0	y_{n2}	0	0	0	0	0	--	0	0	0	0	0	0	---	0	0	0	0	0	0	---	0	0	0	0	0	0	---	0	0	1	1	0
$C_j - Z_j$		0	$I_{d_{11}}$	$I_{d_{12}}$	$I_{d_{21}}$	$I_{d_{22}}$	--	$I_{d_{r-1,1}}$	$I_{d_{r-1,2}}$	$I_{d_{r1}}$	$I_{d_{r2}}$	$I_{d_{r+1,1}}$	$I_{d_{r+1,2}}$	---	$I_{d_{n1}}$	$I_{d_{n2}}$	$I_{y_{11}}$	$I_{y_{12}}$	$I_{y_{21}}$	$I_{y_{22}}$	---	$I_{y_{r-1,1}}$	$I_{y_{r-1,2}}$	$I_{y_{r1}}$	$I_{y_{r2}}$	$I_{y_{r+1,1}}$	$I_{y_{r+1,2}}$	---	$I_{y_{n-1,1}}$	$I_{y_{n-1,2}}$	$I_{y_{n1}}$	$I_{y_{n2}}$	Z^*

All the basic y_{ij} 's are at zero level in the OT(σ) because each constraint must hold as the equation. (see Optimal Tableau 3.1, OT(3.1))

NOTE :

1. The total number of basic variables

   = (2r -1) + (2n - 2r) + 1 variable, k

   = 2n

2. For early jobs, the E_i, d_{i1} ,d_{i2} , ..and $d_{r-1,1}$ are non-basic and hence are equal to zero (which in fact should be the case because tardiness for early jobs is zero). Similarly, for late jobs, the T_i , $d_{r+1,2}$,$d_{r+2,2}$,...and d_{n2} are non-basic variables, also equal to zero.

3. For the optimal due date constraint both d_{r1} and d_{r2} are non-basic and hence at zero level.(see Lemma 1)

3.1.4 INDICATORS

The indicators in the OT(3.1) for all basic variables are at zero level as is always is case in (LPP); the indicators for non-basic variables are determined as follows :

Let $I_{d_{ij}}$,and $I_{y_{ij}}$ denote the indicators for the d_{ij} and y_{ij} respectively. Then,

$I_{d_{11}} \quad = w_{11} + w_{12}$

$I_{d_{21}} \quad = w_{21} + w_{22}$

:

:

:

$I_{d_{r-1,1}} = w_{r-1,1} + w_{r-1,2}$

$$I_{d_{r1}} = w_{r1} - \{ w_{12} + w_{22} + \ldots + w_{r-1,2} + w - (w_{r+1,1} + w_{r+2,1} + \ldots + w_{n1})\}$$

$$= w_{r1} - \sum_{i=1}^{r-1} w_{i2} - w + \sum_{i=r+1}^{n} w_{i1}$$

$$= \boxed{\sum_{i=r}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} - w} \qquad (3.6)$$

$$I_{d_{r2}} = w_{r2} - \{- (w_{12} + w_{22} + \ldots + w_{r-1,2} + w) + (w_{r+1,1} + w_{r+2,1} + \ldots + w_{n1})\}$$

$$= w_{r2} + \sum_{i=1}^{r} w_{i2} + w - \sum_{i=r+1}^{n} w_{i1}$$

$$= \boxed{\sum_{i=1}^{r} w_{i2} - \sum_{i=r+1}^{n} w_{i1} + w} \qquad (3.7)$$

$$I_{d_{r+1,2}} = w_{r+1,1} + w_{r+1,2}$$

$$\vdots$$

$$I_{d_{n2}} = w_{n2} + w_{n1}$$

$$I_{y_{12}} = w_{12}$$

$$I_{y_{22}} = w_{22}$$

$$\vdots$$

$$I_{y_{r-1,2}} = w_{r-1,2}$$

$$I_{y_{r2}} = 0 - \{ (w_{12} + w_{22} + \ldots + w_{r-1,2} + w) - (w_{r+1,1} + w_{r+2,2} + \ldots + w_{n,1}) \}$$

$$= \boxed{-\sum_{i=1}^{r-1} w_{i2} - w + \sum_{i=r+1}^{n} w_{i1}} \qquad (3.8)$$

$$I_{y_{r+1,1}} = w_{r+1,1}$$

$$I_{y_{r+2,1}} = w_{r+2,1}$$

$$\vdots$$

$$\vdots$$

$$I_{y_{n1}} = w_{n1}$$

Since the tableau is optimal, all indicators must be non-negative. We see that the indicators for non-basic variables other than $I_{d_{r1}}$, $I_{d_{r2}}$, and $I_{y_{r2}}$ are evidently positive (because we assumed w_{ij} to be strictly positive).

From (3.6) we have

$$I = \sum_{i=r}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} - w \geq 0$$

$$\Rightarrow \boxed{\sum_{i=r}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} \geq w} \qquad (3.9)$$

From (3.7) we have

$$I_{d_{r2}} = \sum_{i=1}^{r} w_{i2} - \sum_{i=r+1}^{n} w_{i1} + w > 0$$

$$\Rightarrow \boxed{\sum_{i=r+1}^{n} w_{i1} - \sum_{i=1}^{r} w_{i2} \leq w} \qquad (3.10)$$

From (3.8) we have

$$I_{y_{r2}} = \sum_{i=r+1}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} - w \geq 0$$

$$\Rightarrow \boxed{\sum_{i=r+1}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} \geq w} \qquad (3.11)$$

Comparing (3.9) and (3.11) we have the following important result.

$$\boxed{\sum_{i=r}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} > w} \qquad (3.12)$$

Theorem 3.1 : The value of r, the position of the optimal due date for σ is obtained from :

$$\boxed{\sum_{i=r}^{n} w_{i1} - \sum_{i=1}^{r-1} w_{i2} > w} \qquad (3.12)$$

and

$$\boxed{\sum_{i=r+1}^{n} w_{i1} - \sum_{i=1}^{r} w_{i2} \leq w} \qquad (3.10)$$

Proof : The proof follows from (3.9), (3.11) and (3.10).

COROLLARY 3.1 :

If $w = P1$, $w_{i2} = P2$ and $w_{i1} = P3$ for all $J_i(\sigma)$,

then r, the position of ODD(σ) is obtained by using

$$\frac{nP3 - P1}{P2 + P3} \leq r < \frac{nP3 - P1}{P2 + P3} + 1 \qquad (3.13)$$

COROLLARY 3.2 :

If $w = 0$, $w_{i1} = w_{i2} = w$ for all $J_i(\sigma)$, then the position of ODD(σ) is obtained by using

$$\frac{n}{2} \leq r < \frac{n}{2} + 1 \qquad (3.14)$$

REMARKS

1. It may be pointed out here that the result of Cheng[1985, Theorem; p. 393] follows as a special case from our Theorem 3.1 by setting w = 0.
2. It may be interesting to compare the result (3.13) of Corollary 3.1 with the result of Panwalker et al.[1982, Lemma 2; p. 393]

Marginal Values (MV's) Corresponding to the Completion-Times Of (LGP) Associated with the Sequence : From the optimal tableau we obtain the marginal values , u , of the right hand side (RHS) of the (LGP) constraints :

$$\boxed{u_i = I_{y_{i1}} - I_{y_{i2}} \quad \text{for all } J_i(\sigma)} \qquad (3.15)$$

3.2 DEVELOPMENT OF THE METHOD

We will make use of the following observations in the development of the method to find the optimal sequence.

Observations based on (3.15), we have :

(O1) : the MV's corresponding to the set E (the set of early jobs) are negative; therefore, by increasing completion times of early jobs we decrease the value of the penalty function.

(<u>O2</u>): the MV's corresponding to the set T (the set of tardy jobs) are positive, therefore, by decreasing the completion times of tardy jobs, we decrease the value of the penalty function.

(<u>O3</u>): the increase(decrease) in the completion times of an an early(tardy) job in the set E (T) in observation O1(O2) is not arbitrary and is strictly constrained by the processing times p_1, p_2, p_3, ... , p_n of the n jobs of the given sequence (σ). However, we try to increase(decrease) the completion times of early(tardy) jobs by as large an amount as possible to cause the maximum decrease in the penalty function at every step. This creates the possibility of making more than one interchange of jobs in each step and decreasing the penalty by as much as possible at each step.

(<u>O4</u>): the increase(decrease) of completion times of early(tardy) jobs (as described in Observations (O1) through (O3)) by exchanging their positions in the sequence is continued so long as the penalty function keeps decreasing. Since the penalty function is a **monotonic function,** we stop as soon as it starts increasing and identify the optimal sequence σ and, along with it the optimal common due date $k(\sigma)$.

REMARK (R3) We shall base our method on four observations (O1) - (O4). We now address the following problem(P1) :-

3.2.1 <u>PROBLEM</u> (<u>P1</u>)

Given

i) a specified sequence σ with n jobs, along with their processing times p_1, p_2, p_3, ... , p_n , ii)w, such that nw = 1 ($0 < w < 1$), as the unit penalty for early and tardy jobs,

to find an optimal sequence, σ^* , and optimal due date d^* such that the total penalty (sum of early and tardy penalties) is a minimum.

Before we give an algorithm to solve (P1) we introduce some some definitions:

Initial Feasible Sequence : Given a sequence with n jobs, arrange it in SPT form. Then obtain an initial due date, and its position, r, by applying the (LGP) model to the SPT sequence. Arrange the jobs to the left of ODD(SPT) including the job at ODD(SPT) in LPT form and jobs to the right of ODD(SPT) in SPT form (which may already be in SPT form) to obtain a V-shaped sequence, σ_0 , of n jobs. We call ,the initial feasible sequence.

Optimal Sequence : Starting with σ_0,we obtain a sequence σ_1 by exchanging jobs in the set $E(\sigma_0)$ with the jobs in the set $T(\sigma_0)$ such that :

(i) σ_1 is a V-shape (i.e. a feasible) sequence,

(ii) completion times of the jobs in $E(\sigma_1)$ does not decrease and in $T(\sigma_1)$) does not increase,

(iii) the increase (decrease) in the processing times is restricted (observation (O3)) to p_i's of jobs in σ_1 .

(i) and (ii) along with (O1) and (O2) decrease (or do not increase) the value of the penalty function Z. Therefore,

$Z(\sigma_1) \leq Z(\sigma_0)$.

Similarly, we obtain $Z(\sigma_2)$, $Z(\sigma_3)$ and so on such that:

$Z(\sigma_s) \leq Z(\sigma_{s-1}) \leq Z(\sigma_{s-2}) \leq \ldots \leq Z(\sigma_1) \leq Z(\sigma_0)$ where s is finite.

If $Z(\sigma_{s+1}) > Z(\sigma_s)$,

we define σ_s as an optimal sequence σ^* with $Z(\sigma^*)$ as the optimal penalty $Z(\sigma_s)$ and $k(\sigma_s)$ as the optimal due date.

ALGORITHM TO SOLVE (P1) :

STEP 1. From the sequence σ_{SPT}, obtain the initial feasible sequence σ_0, $k(\sigma_0)$, $C_i(\sigma_0)$, $Z(\sigma_0)$, $d_{ij}(\sigma_0)$ and r the position of $k(\sigma_0)$.

STEP 2. From σ_0 obtain σ_1 in the manner described in the definition of 'optimal sequence'.

STEP 3. Find $Z(\sigma_1)$

(i) If $Z(\sigma_1) > Z(\sigma_0)$, then σ_0 is the optimal sequence σ^* with $k(\sigma_0)$ as the optimal common due date $k(\sigma^*)$ and $Z(\sigma_0)$ is the minimum penalty $Z(\sigma^*)$.

(ii) If $Z(\sigma_1) \leq Z(\sigma_0)$, go to step 4.

STEP 4 Repeat step 2 and 3 on $\sigma_1, \sigma_2, \sigma_3, \ldots. \sigma_s, \sigma_{s+1}$, where s is finite such that

$$Z(\sigma_s) \leq Z(\sigma_{s-1}) \leq \ldots. \leq Z(\sigma_2) \leq Z(\sigma_1) \leq Z(\sigma_0)$$

and $$Z(\sigma_{s+1}) > Z(\sigma_s).$$

Then σ_s is the optimal sequence σ^* with $k(\sigma^*)$ as the optimal common due date and $Z(\sigma_s)$ as the minimum value of the penalty function.

NUMERICAL EXAMPLE : We now give a numerical example to apply and illustrate the method developed for solving a problem of type (P1).

Suppose we have a single machine 10 jobs problem as follows:-

Jobs	: J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
p_i's	:18	13	16	19	1	2	5	8	9	10

σ_{SPT}	: 1	2	5	8	9	10	13	16	18	19

To Obtain σ_0 **:**

σ_0	9	8	5	2	1	10	13	16	18	19
$C_i(\sigma_0)$	9	17	22	24	25	35	48	64	82	101
$d_{ij}(\sigma_0)$	16	8	3	1	0	10	23	39	57	76

By solving LGP(σ_0) we obtain

$r = 5$ (the position of ODD( )).

$k(\sigma_0) = 25$ (the optimal common due-date).

$Z(\sigma_0) = 232$ (the value of the objective function).

To Obtain σ_1 **:**

(i) We rewrite σ_0 by showing the position r of ODD(σ_0) by an arrow and marking the sets E(σ_0) and T(σ_0).

$$\sigma_0 : (9 \quad 8 \quad 5 \quad 2 \quad \overset{\downarrow}{1} \quad 10 \quad 13 \quad 16 \quad 18 \quad 19)$$

$$E(\sigma_0) : \{9 \quad 8 \quad 5 \quad 2 \quad 1\}$$

$$T(\sigma_0) : \qquad\qquad\qquad \{10 \quad 13 \quad 16 \quad 18 \quad 19\}$$

(ii) Now we know that, by increasing(decreasing) the completion times of jobs in the set E(T), we obtain a decrease in the penalty function (see O1 and O2). However, we want the maximum possible decrease in the penalty function so we increase(decrease) the completion times of the jobs in set E(T) as much as possible subject to the restrictions on their processing times and the feasibility of the sequence to be obtained.

(iii)Starting with the job at position 1 in set E(T) we identify the maximum possible increase(decrease) in each of the jobs' completion time such that :

a) the increase(decrease) is restricted by the processing times of the jobs in σ_0.

b) σ_1 is feasible .

We conclude that we can increase the processing time of the job in the first position in set E to the processing time of the job in the first position of set T, and decrease the processing time in set T to the processing time of the job in the first position of set E.

Exchange Process For the sake of simplicity we will call this process an 'exchange process' because we can now say that, to obtain σ_1 from σ_0 ,we simply exchange the job in position 1 (to be general say position i) with the job in position 6 (to be general say position j) in the sequence .

$$\sigma_1 = (10 \quad 8 \quad 5 \quad 2 \quad 1 \quad 9 \quad 13 \quad 16 \quad 18 \quad 19)$$

$$C_i(\sigma_1) = 10 \quad 18 \quad 23 \quad 25 \quad 26 \quad 35 \quad 48 \quad 64 \quad 82 \quad 101$$

$r = 5$,

$k(\sigma_1) = 26$,

$Z(\sigma_1) = 228$.

Since

$$Z(\sigma_1) < Z(\sigma_0) ,$$

we obtain σ_2 from σ_1 in the manner described in Step 2.

To Obtain :

(i) We rewrite σ_1 by showing the position r of ODD(σ_1) by the arrow and marking the sets and E(σ_1) AND T(σ_1) as follows :-

$$= (10 \quad 8 \quad 5 \quad 2 \quad \overset{\downarrow}{1} \quad 9 \quad 13 \quad 16 \quad 18 \quad 19)$$

$E(\sigma_1) = \{ 10 \quad 8 \quad 5 \quad 2 \quad 1 \}$

$T(\sigma_1) = \qquad\qquad\qquad\qquad \{ 9 \quad 13 \quad 16 \quad 18 \quad 19 \}$

Exchange Process

(i) Exchange $J_1(\sigma_1)$ with $J_7(\sigma_1)$,

(ii) Exchange $J_2(\sigma_1)$ with $J_6(\sigma_1)$.

$\sigma_2 \quad = (13 \quad 9 \quad 5 \quad 2 \quad 1 \quad 8 \quad 10 \quad 16 \quad 18 \quad 19)$

$C_i(\sigma_2) = (13 \quad 22 \quad 27 \quad 29 \quad 30 \quad 38 \quad 48 \quad 64 \quad 82 \quad 101)$

$d_{ij}(\sigma_2) = (17 \quad 8 \quad 3 \quad 1 \quad 0 \quad 8 \quad 18 \quad 34 \quad 52 \quad 71)$

$r = 5,$

$k(\sigma_2) = 30$

$Z(\sigma_2) = 212.$

Since

$$Z(\sigma_2) < Z(\sigma_1) < Z(\sigma_0),$$

we obtain σ_3 from σ_2 in the manner described in Step 2 of the algorithm.

To Obtain :

(i) We rewrite σ_2 by showing r by an arrow and marking the sets $E(\sigma_2)$ & $T(\sigma_2)$.

$$\begin{array}{lllllllllll} & & & & & \downarrow & & & & & \\ \sigma_2 = & (13 & 9 & 5 & 2 & 1 & 8 & 10 & 16 & 18 & 19\ ) \\ E(\sigma_2) = & \{13 & 9 & 5 & 2 & 1\} & & & & & \\ T(\sigma_2) = & & & & & & \{ 8 & 10 & 16 & 18 & 19\ \} \end{array}$$

Exchange Process

(i) Exchange $J_1(\sigma_2)$ with $J_8(\sigma_2)$

(ii) Exchange $J_2(\sigma_2)$ with $J_7(\sigma_2)$,

(iii) Exchange $J_3(\sigma_2)$ with $J_6(\sigma_2)$.

$$\sigma_3 = (\ 16 \quad 10 \quad 8 \quad 2 \quad 1 \quad 5 \quad 9 \quad 13 \quad 18 \quad 19\ )$$

$$C_i(\sigma_3) = (\ 16 \quad 26 \quad 34 \quad 36 \quad 37 \quad 42 \quad 51 \quad 64 \quad 82 \quad 101\ )$$

$r = 5$

$k(\sigma_3) = 37$

$Z(\sigma_3) = 191$

Since

$$Z(\sigma_3) < Z(\sigma_2) < Z(\sigma_1) < Z(\sigma_0)$$

the optimal sequence sequence is not obtained yet so we will obtain σ_4, σ_5, σ_6, and σ_7 by following the exchange process as :-

$$\sigma_4 = (\ 18 \quad 13 \quad 9 \quad 5 \quad 1 \quad 2 \quad 8 \quad 10 \quad 16 \quad 19\ )$$

$$C_i(\sigma_4) = (\ 18 \quad 31 \quad 40 \quad 45 \quad 46 \quad 48 \quad 56 \quad 66 \quad 82 \quad 101\ )$$

$$d_{ij}(\sigma_4) = (\ 28 \quad 15 \quad 6 \quad 1 \quad 0 \quad 2 \quad 10 \quad 20 \quad 36 \quad 55\ )$$

$r = 5$

$k(\sigma_4) = 46$

$Z(\sigma_4) = 173$

$$\sigma_5 = (\ 19 \quad 16 \quad 10 \quad 8 \quad 2 \quad 1 \quad 5 \quad 9 \quad 13 \quad 18)$$

$C_i(\sigma_5) = (\ 19 \quad 35 \quad 45 \quad 53 \quad 55 \quad 56 \quad 61 \quad 70 \quad 83 \quad 101)$

$d_{ij}(\sigma_5) = (\ 36 \quad 20 \quad 10 \quad 2 \quad 0 \quad 1 \quad 6 \quad 15 \quad 28 \quad 46)$

$r = 5$

$k(\sigma_5) = 55$

$Z(\sigma_5) = 164$

$\sigma_6 = (\ 19 \quad 18 \quad 13 \quad 9 \quad 5 \quad 1 \quad 2 \quad 8 \quad 10 \quad 16)$

$C_i(\sigma_6) = (\ 19 \quad 37 \quad 50 \quad 59 \quad 64 \quad 65 \quad 67 \quad 75 \quad 85 \quad 101)$

$d_{ij}(\sigma_6) = (\ 45 \quad 27 \quad 14 \quad 5 \quad 0 \quad 1 \quad 3 \quad 11 \quad 21 \quad 37)$

$r = 5$

$k(\sigma_6) = 64$

$Z(\sigma_6) = 164$

$\sigma_7 = (\ 19 \quad 18 \quad 16 \quad 10 \quad 5 \quad 1 \quad 2 \quad 8 \quad 9 \quad 13\ )$

$C_i(\sigma_7) = (\ 19 \quad 37 \quad 53 \quad 63 \quad 68 \quad 69 \quad 71 \quad 79 \quad 88 \quad 101)$

$d_{ij}(\sigma_7) = (\ 49 \quad 31 \quad 15 \quad 5 \quad 0 \quad 1 \quad 3 \quad 11 \quad 20 \quad 33)$

$r = 5$

$k(\sigma_7) = 68$

$Z(\sigma_7) = 168$

Since,

$Z(\sigma_7) > Z(\sigma_6)$

therefore, the σ_6 is an optimal sequence σ^*. Hence the final solution is :

σ^* = (19 18 13 9 5 1 2 8 10 16)

$k(\sigma^*)$ = 64

$Z(\sigma^*)$ = 164

REMARK : In the above example we see that

$Z(\sigma_5) = Z(\sigma_6)$

therefore,

σ_5 is an alternative optimal sequence with σ^*

σ^* = (19 16 10 8 2 1 5 9 13 18)

$k(\sigma^*)$ = 55

$Z(\sigma^*)$ = 164

3.2.2 PROBLEM(P2)

We now address the following problem (P2) : Consider an n job single machine scheduling problem in which all jobs have a common due date, k, processing times, p_1, p_2, ..., p_n, and a penalty to be incurred when a job is completed before or after the due-date. When some pre-determined value of common due date is known, to determine a corresponding optimal sequence that minimizes the penalty function.

Either exactly on the predetermined value of common due date,($P_2(a)$)

$P_1(c)$

or in the neighborhood of the predetermined due date. Before we give the algorithm to solve (P2) we modify the definition of optimal sequence :

a)corresponding to predetermined due date ; and

b)around the predetermined due date .

Modified Feasible Sequence : Given a sequence σ_0 with n jobs. Arrange it in SPT form and call it a σ_{SPT} sequence. Obtain an optimal due date and its position, r, by applying (LGP2) model to σ_{SPT}. Arrange the jobs to the left of ODD(σ_{SPT}), including the job at ODD(σ_{SPT}), in LPT form and the jobs to the right of ODD(SPT) in SPT form to obtain a V-shaped sequence, σ_0, of n jobs. We call an initial feasible sequence if

$$k(\sigma_{SPT}) \leq d$$

Modified Optimal sequence : Starting with σ_0, we obtain a sequence σ_1 by exchanging jobs in the set E(σ_0) with the jobs in the set T(σ_0) such that :

(i) σ_1 is a V-shape (i.e. σ_1 is a feasible) sequence,

(ii) Completion-time of jobs, in E(σ_1) does not decrease and in T(σ_1) does not increase.

(iii)Increase(decrease) in completion times is restricted by (O3) and by the predetermined due date d .

(i) and (ii) along with (O1) and (O2) decrease the value of the penalty function Z. Therefore,

$$Z(\sigma_1) \leq Z(\sigma_0)$$

and $$d > k(\sigma_1) \geq k(\sigma_0)$$

Similarly, we obtain $Z(\sigma_2)$, $Z(\sigma_3)$, ...$Z(\sigma_S)$, $Z(\sigma_{S+1})$ such that

$$Z(\sigma_S) \leq Z(\sigma_{S-1}) \leq \ldots \leq Z(\sigma_2) \leq Z(\sigma_1) \leq Z(\sigma_0)$$

and $$d^* = k(\sigma_S) \geq k(\sigma_{S-1}) \geq \ldots \geq k(\sigma_1) \geq k(\sigma_0)$$

Now, we define σ_S as the optimal sequence σ^* with $Z(\sigma_S)$ as the optimal penalty $Z(\sigma^*)$ and

$$k(\sigma_S) = d^*$$

ALGORITHM TO SOLVE (P2)

Step 1 From the sequence σ_{SPT}, obtain the initial feasible sequence σ_0, the completion times, $C(\sigma_0)$, the due date value, $k(\sigma_0)$, the deviational values, $d_{ij}(\sigma_0)$, the total penalty value, $Z(\sigma_0)$, and r, the position of $k(\sigma_0)$. If

(i) $k(\sigma_0) = d$; σ_0 is optimal sequence for d .

(ii) $k(\sigma_0) > d$; the sequence is infeasible, and

(iii) $k(\sigma_0) < d$; go to the step 2.

Step 2 : From σ_0 obtain σ_1 , $C_i(\sigma_1)$, $k(\sigma_1)$, $d_{ij}(\sigma_1)$, $Z(\sigma_1)$ and r, the position of due date according to the observations (O1) to (O3).

Step 3 : If

(i) $k(\sigma_1) = d$; σ_1 is the optimal sequence for $P_2(a)$.

(ii) $k(\sigma_1) < d$; go to Step 4.

Step 4 : Repeat the Steps 2 and 3 on $\sigma_2, \sigma_3, \sigma_4, \ldots \sigma_s$ where s is finite such that :

CASE 1 If

$$Z(\sigma_s) \leq Z(\sigma_{s-1}) \leq \ldots\ldots \leq Z(\sigma_2) \leq Z(\sigma_1) \leq Z(\sigma_0)$$

and $$d^* = k(\sigma_s) \geq k(\sigma_{s-1}) \geq \ldots\ldots \geq k(\sigma_1) \geq k(\sigma_0)$$

then, σ_s is the desired optimal sequence σ^* associated with d^*.

else

If $k(\sigma_s) > d$ then go to case 2.

CASE 2

(i) Compute the difference $[k(\sigma_{s-1}) - d]$.

(ii) Investigate the individual increases(decreases) in the completion-times of jobs that may be done to minimize the penalty function.

(iii) Select which combination, if any, of some of the jobs' processing times that results in the desired increase. If there is none, select a group of one or more jobs whose processing times sum up very close to the desired increase.

(iv) form the alternative sequences giving

a) $k(\sigma_s) = d$ or

b) $k(\sigma_s) = d + \varepsilon$, where ε is a small integer value.

c) compute $Z(\sigma_s^{(i)}, d^*)$ for the above values of ε , where

$Z(\sigma_s^{(i)}, d^*)$ are the penalty functions associated with the deviation of the sequences, $\sigma_s^{(i)}$, around d^*.

Suppose $$Z(\bar{\sigma_s}, d^*) = \text{Min } Z(\sigma_s^{(i)}, d^*) \quad ,$$

then $\bar{\sigma_s}$ is the desired optimal sequence for $P_2(b)$.

NUMERICAL EXAMPLE : We will take the problem from the problem set tabulated by Sundararaghavan and Ahmed [1982] and solve it for both Cases discussed above in (P2) when the due date is given i.e. d = 290.

Jobs:	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14
p_j's:	1	2	4	5	8	23	31	53	55	65	68	69	90	92

To obtain σ_0 :

Exchange Process

Simply obtain the V-shaped sequence as required by the modified feasible sequence.

σ_0 :	(31	23	8	5	4	2	1	53	55	65	68	69	90	92)
C_i :	(31	54	62	67	71	73	74	127	182	247	315	384	474	566)
d_{ij} :	(43	20	12	7	3	1	0	53	108	173	241	310	400	492)

$k(\sigma_0) = 74$

$Z(\sigma_0) = 1863$

To Obtain σ_1 :

Exchange Process :

(i) Exchange $J_1(\sigma_0)$ with $J_8(\sigma_0)$

σ_1 :	(53	23	8	5	4	2	1	31	55	65	68	69	90	92)
C_i :	(53	76	84	89	93	95	96	127	182	247	315	384	474	566)
d_{ij} :	(43	20	12	7	3	1	0	31	86	151	219	288	378	470)

$k(\sigma_1) = 96$

$Z(\sigma_1) = 1707$

To Obtain σ_2:

Exchange Process

(i) Exchange $J_1(\sigma_1)$ with $J_9(\sigma_1)$,

(ii) Exchange $J_2(\sigma_1)$ with $J_8(\sigma_1)$,

σ_2 :	(55	31	8	5	4	2	1	23	53	65	68	69	90	92)
C_i' :	(55	86	94	99	103	105	106	129	182	247	315	384	474	566)
d_{ij}' ;	(51	20	12	7	3	1	0	23	76	141	209	278	368	460)

$k(\sigma_2) = 106$

$z(\sigma_2) = 1649$

To Obtain σ_3 :

Exchange Process

(i) Exchange $J_1(\sigma_2)$ with $J_{10}(\sigma_2)$,

(ii) Exchange $J_2(\sigma_2)$ with $J_9(\sigma_2)$,

(iii) Exchange $J_3(\sigma_2)$ with $J_8(\sigma_2)$.

σ_3 :	(65	53	23	5	4	2	1	8	31	55	68	69	90	92)
C_i' :	(65	118	141	146	150	152	153	161	192	247	315	384	474	566)
d_{ij}' :	(88	35	12	7	3	1	0	8	39	94	162	231	321	413)

$k(\sigma_3) = 153$

$Z(\sigma_3) = 1414$

To Obtain σ_4 :

Exchange Process

(i) exchange $J_1(\sigma_3)$ with $J_{11}(\sigma_3)$,

(ii) exchange $J_2(\sigma_3)$ with $J_{10}(\sigma_3)$,

(iii) exchange $J_3(\sigma_3)$ with $J_4(\sigma_3)$,

(iv) exchange $J_4(\sigma_3)$ with $J_8(\sigma_3)$.

σ_4 :	(68	55	31	8	4	2	1	5	23	55	65	69	90	92)
C_i :	(68	123	154	162	166	168	169	174	197	250	315	384	474	566)
d_{ij} :	(101	46	15	7	3	1	0	5	28	81	146	215	305	397)

$k(\sigma_4) = 169$

$Z(\sigma_4) = 1350$

To Obtain σ_5 **:**

Exchange Process

(i) exchange $J_1(\sigma_4)$ with $J_{12}(\sigma_4)$,

(ii) exchange $J_2(\sigma_4)$ with $J_{11}(\sigma_4)$,

(iii) exchange $J_3(\sigma_4)$ with $J_{10}(\sigma_4)$,

(iv) exchange $J_4(\sigma_4)$ with $J_9(\sigma_4)$,

(v) exchange $J_5(\sigma_4)$ with $J_8(\sigma_4)$.

σ_5 :	(69	65	53	23	5	2	1	4	8	31	55	68	90	92)
C_i :	(69	134	187	210	215	217	218	222	230	261	316	384	474	566)
d_{ij} :	(149	84	31	8	3	1	0	4	12	43	98	166	256	348)

$k(\sigma_5) = 218$

$Z(\sigma_5) = 1303$

To Obtain σ_6 **:**

Exchange Process

(i) exchange $J_1(\sigma_5)$ with $J_{13}(\sigma_5)$,

(ii) exchange $J_2(\sigma_5)$ with $J_{12}(\sigma_5)$,

(iii) exchange $J_3(\sigma_5)$ with $J_{11}(\sigma_5)$,

(iv) exchange $J_4(\sigma_5)$ with $J_{10}(\sigma_5)$,

(v) exchange $J_5(\sigma_5)$ with $J_9(\sigma_5)$,

(vi) exchange $J_6(\sigma_5)$ with $J_8(\sigma_5)$.

σ_6 :	(90	68	55	31	8	4	1	2	5	23	53	65	69	92)
C_i :	(90	158	213	244	252	256	257	259	264	287	340	405	474	566)
d_{ij} :	(167	99	44	13	5	1	0	2	7	30	83	148	217	309)

$k(\sigma_6) = 257,$

$Z(\sigma_6) = 1215.$

Δ's: (2 1 10 22 15 1 1 -1 -1 -15 -22 -10 -1 -2)

DESIRED INCREASE : 290 - 257 = 33

ALTERNATIVE SEQUENCES :

By investigating the possible combinations of jobs, we obtain a set of possible alternative sequences. We will now compute the penalty from these alternative sequences around the due date as follows:- the due date as :

Alt(1)

σ_{A1} :	(90	69	65	53	8	4	1	2	5	23	31	55	68	92)
C_i :	(90	159	224	277	285	289	290	292	297	320	351	406	474	566)
d_{ij} :	(200	131	66	13	5	1	0	2	7	30	61	116	184	276)

$k(\sigma_{A1}) = 290$

$z(\sigma_{A1}) = 1092$

Alt(2)

σ_{A2} : (90 68 65 53 8 5 1 2 4 23 31 55 69 92)

C_i' : (90 158 223 276 284 289 290 292 296 319 350 405 474 566)

d_{ij} : (200 132 67 14 6 1 0 2 6 29 60 115 184 276)

$k(\sigma_{A2}) = 290$

$Z(\sigma_{A2}) = 1092$

Alt(3)

σ_{A3} :(90 68 65 53 8 4 2 1 5 23 31 55 69 92)

C_i :(90 158 223 276 284 288 290 291 296 319 350 405 474 566)

d_{ij} :(200 132 67 14 6 2 0 1 6 29 60 115 184 276)

$k(\sigma_{A3}) = 290$

$Z(\sigma_{A3}) = 1092$

Alt(4)

σ_{A4} :(90 69 65 53 8 5 1 2 4 23 31 55 68 92)

C_i :(90 159 224 277 285 290 291 293 297 320 351 406 474 566)

d_j :(200 131 66 13 5 0 1 3 7 30 61 116 184 276)

$k(\sigma_{A4}) = 290$

$Z(\sigma_{A4}) = 1093$

Alt(5)

σ_{AJ} :(	90	69	65	53	8	4	2	1	5	23	31	55	68	92)
C_j :(	90	159	224	277	285	289	291	292	297	320	351	406	474	566)
d_j :(	200	131	66	13	5	1	1	2	7	30	61	116	184	276)

$$k(\sigma_{AJ}) = 290$$

$$Z(\sigma_{AJ}) = 1093$$

Thus, we see that our algorithm is able to recognize the alternative optimal sequences and has obtained the optimal value of objective function. It shows that our algorithm is an improvement on the heuristic provided in Sundararaghavan and Ahmed [1982]. All of the problems tabulated by them for which they were not able to obtained the exact values has been solved for exact value by our algorithm.

Chapter IV

DISCUSSION OF SPECIAL CASES

This chapter deals with the comparison of existing heuristics and algorithms for the determination of an optimal sequence :

a) when the penalties for earliness and tardiness are equal, and

b) when a generalized penalty function is provided.

There is extensive literature available which deals with the above mentioned problems. Merten and Muller[1972], Schrage[1975], Eilon and Chowdhary[1977] and Kanet[1981] minimized the completion times variance in order to develop an optimal sequence. Merten and Muller[1972] demonstrated that the optimal sequence obtained by minimizing the flow-time variance could lead to the determination of another optimal sequence which minimizes the waiting time variance by simply obtaining its antithesis. Schrage[1975] has examined scheduling for minimum completion times variance when there are up to 5 jobs to be scheduled. Eilon and Chowdhary[1977] have shown that, for a schedule to have minimum completion times variance, its sequence must be V-shaped. They proposed a number of heuristics which result in V-shaped curves. Kanet[1981] proposed another procedure to minimize the completion times variance (SMV) and showed that his procedure provides improved results when compared with the heuristics proposed by Eilon and Chowdhary[1977]. He also suggested a criterion of minimization of total absolute difference from completion times (TADC) as a viable alternative to the criterion of minimization of completion times. He presented a procedure to find an optimal sequence using the TADC.

Sidney[1977] suggested an algorithm to minimize the maximum penalty subject to restrictive assumptions on the target start times, the due dates and the nature of the given penalty functions.He defined the penalty function mathematically as follows :-

$$\mathbf{COST(S) = max[\ g\{max(E_i)\},\ h\{max(T_i)\}]}$$

where g(.) and h(.) are the earliness and tardiness functions. Panwalker et al.[1982] presented an algorithm to find an optimal sequence and an optimal due date by minimizing the penalty function of the type given below :

$$\mathbf{Z} = \sum_{i=1}^{n} [\ \mathbf{P1(d) + P2(E_i) + P3(T_i)}\ ]$$

where P1, P2, and P3 are the penalties associated with the due date achieved, earliness of all jobs and tardiness of all jobs. Several other types of penalty functions have been considered in the sequencing literature(e.g. McNaughton[1959], Lawler[1964])

To find an optimal schedule in a deterministic single machine jobshop, Kanet[1981] suggested a Gantt chart approach. French[1982] further demonstrated the use of such a technique in scheduling.

In this chapter, we propose -

(1) The TSDD heuristic and the ST procedure which minimizes absolute deviations about the due date to determine an optimal sequence. Then, the TSDD and the ST procedures are compared with the SMV and the TADC heuristics.

(2) The ST procedure when a given generalized penalty function for the determination of an optimal sequence and an optimal due date. The St procedure minimizes the penalty function to find the optimal sequence and due date. Then this is compared with the algorithm proposed by Panwalker et al.[1981].

(3) The procedure to find an optimal sequence minimizing the penalty around the given due date. The algorithm presented is then compared with the one developed by Sundararghavan and Ahmed[1984]. This comparison shows that our algorithm is better.

(4) Finally, a goal programming approach to find an optimal schedule. It is also compared with the Gantt chart approach suggested by Kanet[1981].

In the next section we will illustrate the various heuristics and procedures with numerical examples and then make comparisons among them.

4.1 HEURISTICS AND PROCEDURES

TADC Procedure :- This procedure minimizes the total absolute difference from completion times in order to obtain an optimal sequence. Mathematically, it can be written as follows :-

$$\sum_{i=1}^{n} \sum_{j=1}^{n} | C_j - C_i |$$

The steps necessary to obtain an optimal sequence can be summarized below :

Procedure TADC
Let U be the set of unscheduled jobs;
Let S be the final schedule;
B,A be empty sets; and
n be the number of jobs to be scheduled.

```
For i= 1 to n do
Begin
 Remove the largest job from U and label it job i.
 if i is odd then
 place job i in the last position of set B.
 else place job i in the first position of set A.
end;
S:= { B,A }
end TADC.
```

<u>EXAMPLE</u> :

U = { 16, 13, 10, 9, 6, 4, 3, 2, 1 }

B = { 16, 10, 6, 3, 1 }

A = { 2, 4, 9, 13 }

S = { 16, 10, 6, 3, 1, 2, 4, 9, 13 }

SMV PROCEDURE :

This procedure minimizes the completion times variance in order to find an optimal sequence. Mathematically, it can be written as follows :

$$\sum_{j=1}^{n} \sum_{i=1}^{n} \{ C_j - C_i \}^2$$

The steps necessary to obtain an optimal sequence can be summarized as follows :-

Let V(S) denotes the variance of completion times for some schedule S and U be the set of unscheduled jobs. Then, procedure SMV

```
Begin
    For i = 1 to n do
       begin
         Let K be the smallest job in S.
         Remove the largest job from U and label it job i.
         Insert job i to the immediate left of K and call
         the result S', compute V(S').
         Insert job i to the immediate right of K & call
         the result S", compute V(S").
         If V(S') < V(S") then S = S'
         else S = S".
       end;
End SMV.
```

EXAMPLE :

U = { 16, 10, 9, 8, 7, 6, 4, 2 }

S = 16

S' = {10, 16}; V(S') = 64

S"= {16, 10}; V(S")= 25 *

S={16,10}

S'={16,9,10}; V(S') = 60.22 *

S"={16,10,9}; V(S") = 60.22

S ={16,9,10}

S'={16,8,9,10}; V(S') = 101.5

S"={16,9,8,10}; V(S") = 99.18 *

S ={16,9,8,10}

S'={16,9,7,8,10}; V(S') =138.24

S"={16,9,8,7,10}; V(S") = 138.16 *

S ={16,9,8,7,10};

S'={16,9,8,6,7,10}; V(S') = 173.138 *

S"={16,9,8,7,6,10}; V(S") = 174.33

S ={16,9,8,6,7,10};

S'={16,9,8,4,6,7,10}; V(S') = 190.20 *

S"={16,9,8,6,4,7,10}; V(S") = 190.28

S ={16,9,8,4,6,7,10};

S'={16,9,8,2,4,6,7,10}; V(S') = 188.484

S"={16,9,8,4,2,6,7,10}; V(S") = 187.23 *

S = {16,9,8,4,2,6,7,10}

TSDD PROCEDURE : In this heuristic, minimization of total variance about a common due date ($d \geq MS$) is used as a criterion for obtaining an optimal sequence. This is based on the premise that generally a manufacturer in job-shop environment is required to provide a due date to the clients prior to even determining the sequence in which jobs will be processed. It should be in the interest of the manufacturer to minimize the variance around this due date, because if the jobs are performed earlier, he incurs inventory carrying costs and on the other hand if they are completed late then the penalty is incurred. The procedure used for finding an optimal sequence can be presented as follows:-

Let U be a set of unscheduled jobs and A,B,C' and C" be the empty set. Also let V(S) denote the variance of completion times about the due date for some schedule S.

```
procedure TSDD Begin
For i = 1 to n do
begin
 Let K be the smallest job in S.
 Remove the largest job from U and label it job i.
 Insert job i to the immediate left of K and call the
 result S'.
 Calculate,C' ,the set of completion times for set S'.
 Calculate the absolute deviations from due date, d, for
 set C'.
 Call the new set A.
 Compute V(A);
 Insert the job i to the immediate right of K and call the
 result S".
 Calculate,C', the set of completion times for set S".
 Calculate, B, the set of absolute deviations from due date, d.
 Compute V(B).
 If V(A) ≤ V(B) then S = S'
 else S = S";
 end;
end TSDD.
```

EXAMPLE:

U = {16, 10, 9, 8, 7, 6, 4, 2} ; $d^* = 63$;

S = {16}

S'= {10,16};C'={10,26};

A = {53,37};V(A) =64.

S"= {16,10};C"={16,26};

B = {47,37);V(B) =25 *

S = {16,10}

S'=(16,9,10};C'={16,25,35};

A =(47,38,28);V(A) = 60.22 *

S"=(16,10,9);C"=(16,26,35);

B =(47,37,28);V(B) = 60.22

S =(16,9,10)

S'=(16,8,9,10);C'=(16,24,33,43);

A =(47,39,30,20);V(A) =101.5

S"=(16,9,8,10);C"=(16,25,33,43);

B =(47,38,30,20);V(B) = 99.18 *

S = (16,9,8,10)

S'=(16,9,7,8,10);C'=(16,25,32,40,50);

A =(47,38,31,23,13);V(A) = 138.24

S"=(16,9,8,7,10);C'=(16,25,33,40,10);

B =(47,38,30,23,13);V(B) = 138.16 *

S =(16,9,8,7,10)

S'=(16,9,8,6,7,10);C'=(16,25,33,39,46,56);

A =(47,38,30,24,17,7);V(A) = 173.138 *

S"=(16,9,8,7,6,10);C"=(16,25,33,40,46,56);

B =(47,38,30,23,17,7);V(B) = 174.33

S = (16,9,8,6,7,10)

S'=(16,9,8,4,6,7,10);C'=(16,25,33,37,43,50,60);

A =(47,38,30,26,20,13,3);V(A) =190.20 *

S"=(16,9,8,6,4,7,10);C"=(16.25,33,39,43,50,60);

B =(47,38,30,24,20,13,3);V(B) = 190.28

S = (16,9,8,4,6,7,10)

S' =(16,9,8,2,4,6,7,10);C'=(16,25,33,35,39,45,52,62);

A =(47,38,30,28,24,11,1);V(A)=188.484

S"=(16,9,8,4,2,6,7,10);C"=(16,25,33,37,39,45,52,62);

B =(47,38,30,26,24,11,1);V(B) =187.23 *

S = (16,9,8,4,2,6,7,10)

ST PROCEDURE : This procedure minimizes the absolute deviations from the due date in order to find an optimal sequence. It is based on the theoretical developments and algorithm from Chapter 3. The procedure is as follows :

Procedure ST
Let U be the SPT sequence of n jobs and r be the location of optimal due date which is calculated using the Equations (3.10) and (3.12) of the Chapter 3. Then,

arrange 1 to rth jobs of set U in LPT form ;
n := 0;
call it σ_n and compute $Z(\sigma_n)$;
$Z(\sigma_*) := Z(\sigma_n)$;
while $Z(\sigma_n) \geq Z(\sigma_*)$ do
begin
    n := n + 1 ;
  perform 'exchange process' such that:
    (i) feasibility is maintained;
    (ii) $p_i < p_j$ (for all $J_i(\sigma_n)$) ;
  compute $Z(\sigma_n)$;
  if $Z(\sigma_n) \leq Z(\sigma_*)$ then
    $Z(\sigma_*) := Z(\sigma_n)$;
end; (* while loop *)
end ST

EXAMPLE This procedure is illustratd by a numerical problem of 10 jobs as follows :

Jobs :	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
p_i's :	12	9	13	1	4	2	10	3	16	6

U	:(1	2	3	4	6	9	10	13	16)

r = 5 (i.e. 9/2 < r < (9/2 + 1))

INITIALIZATION

σ_0	:(6	4	3	2	1	9	10	13	16)
$C_i(\sigma_0)$	:(6	10	13	15	16	25	35	48	64)
d_{ij}	:(10	6	3	1	0	9	19	32	48)

ITERATION 1

exchange $J_1(\sigma_0)$ with $J_6(\sigma_0)$

σ_1 :	(9	4	3	2	1	6	10	13	16)
$c_i(\sigma_1)$:	(9	13	16	18	19	25	35	48	64)
$d_{ij}(\sigma_1)$:	(10	6	3	1	0	6	16	29	45)

$$(Z(\sigma_1) = 116) \leq Z(\sigma_*) = 128)$$

$$\Rightarrow Z(\sigma_*) = 116$$

ITERATION #2

exchange $J_1(\sigma_1)$ with $J_8(\sigma_1)$;

exchange $J_2(\sigma_1)$ with $J_7(\sigma_1)$;

σ_2 :	(10	6	3	2	1	4	9	13	16)
$c_i(\sigma_2)$:	(10	16	19	21	22	26	35	48	64)
$d_{ij}(\sigma)$:	(12	6	3	1	0	4	13	26	42)

$$(Z(\sigma_2) = 107) \leq (Z(\sigma_*) = 116)$$

$$\Rightarrow Z(\sigma_*) := 107$$

ITERATION #3

exchange $J_1(\sigma_2)$ with $J_8(\sigma_2)$

exchange $J_2(\sigma_2)$ with $J_7(\sigma_2)$

exchange $J_3(\sigma_2)$ with $J_6(\sigma_2)$

σ_3 :	(13	9	4	2	1	3	6	10	16)
$C_i(\sigma_3)$:	(13	22	26	28	29	32	38	48	64)
$d_{ij}(\sigma_3)$:	(16	7	3	1	0	3	9	19	35)

$$(Z(\sigma_3) = 93) \leq (Z(\sigma_*) = 106)$$

$$\Rightarrow Z(\sigma_*) := 93$$

ITERATION #4

exchange $j_1(\sigma_3)$ with $J_9(\sigma_3)$

$J_2(\sigma_3)$ with $J_8(\sigma_3)$

$J_3(\sigma_3)$ with $J_7(\sigma_3)$

$J_4(\sigma_3)$ with $J_6(\sigma_3)$

σ_4	:(16	10	6	3	1	2	4	9	13)
$C_i(\sigma_4)$:	(16	26	32	35	36	38	42	51	64)
$d_{ij}(\sigma_4)$:	(20	10	4	1	0	2	6	15	28)

$(Z(\sigma_4) = 86) \leq (Z(\sigma_*) = 93)$

$\Rightarrow Z(\sigma_*) := 86$

ITERATION #5

exchange $J_2(\sigma_4)$ with $J_9(\sigma_4)$

$J_3(\sigma_4)$ with $J_8(\sigma_4)$

$J_4(\sigma_4)$ with $J_7(\sigma_4)$

$J_5(\sigma_4)$ with $J_6(\sigma_4)$

σ_5	:(16	13	9	4	2	1	3	6	10)
$C_i(\sigma_5)$:	(16	29	38	42	44	45	48	54	64)
$d_{ij}(\sigma_5)$:	(28	15	6	2	0	1	4	10	20)

$(Z(\sigma_5) = 86) \leq (Z(\sigma_*) = 86)$

$\Rightarrow Z(\sigma_*) := 86$

ITERATION #6

exchange $J_3(\sigma_5)$ with $J_9(\sigma_5)$

$J_4(\sigma_5)$ with $J_8(\sigma_5)$

$J_5(\sigma_5)$ with $J_7(\sigma_5)$

σ_6	:(16	13	10	6	3	1	2	4	9)

$C_i(\sigma_6)$:(16 29 39 45 48 49 51 55 64)

$d_{ij}(\sigma_6)$:(32 19 9 3 0 1 3 7 16)

$$(Z(\sigma_7) = 90) > (Z(\sigma_*) = 86)$$

=> OPTIMALITY IS ACHIEVED

$Z(\sigma_*) = 86$ and $d(\sigma_*) = 44$

COMPARISON In this section, TADC, SMV, TSDD and ST are compared using seven sequencing problems considered by Eilon and Chowdhary[1977] (see Table 4.1).

TABLE 4.1

Problem Set from Eilon and Chowdhary[1977]

****

S.NO	JOB NO.	PROCESSING TIMES
P1	(J1,J2,J3,J4,J5,J6)	(21, 19, 12, 9, 6, 2)
P2	(J1,J2,J3,J4,J5,J6,J7)	(82, 65, 21, 9, 6, 3, 2)
P3	(J1,J2,J3,J4,J5,J6, J7,J8)	(16, 10, 9, 8, 7, 6, 4, 2)
P4	(J1,J2,J3,J4,J5,J6, J7,J8)	(16, 13, 12, 10, 9, 8, 2, 1)
P5	(J1,J2,J3,J4,J5, J6,J7,J8,J9,J10)	(19, 18, 16, 13, 10, 9, 8, 5, 2, 1)
P6	(J1,J2,J3,J4,J5, J6,J7,J8,J9,J10)	(92, 82, 65, 34, 23, 21, 9, 6, 3, 2)
P7	(J1,J2,J3,J4,J5, J6,J7,J8,J9,J10)	(100, 41, 25, 21, 13, 10, 9, 8, 7, 5)

TABLE 4.2
COMPARISON AMONG DIFFERENT PROCEDURES

P. No.	TADC	SMV	TSDD	ST
P1	(21,12,6,2,9,19)	(21,12,9,2,6,19)	(21,12,9,2,6,19	(21,12,6,2,9,19)
P2	(82,21,6,2,3,9,65)	(82,21,9,6,3,2,65)	(82,21,9,6,3,2,65)	(82,21,6,2,3,9,65)
P3	(16,9,7,4,2,6,8,10)	(16,9,8,4,2,6,7,10)	(16,9,8,4,2,6,7,10)	(16,9,7,4,2,6,8,10)
P4	(16,12,9,2, 1,8,10,13)	(16,13,9,8, 1,2,10,12)	(16,13,9,8, 1,2,10,12)	(16,12,9,2, 1,8,10,13)
P5	(19,16,10,8,2, 1,5,9,13,18)	(19,16,13,8,2, 1,5,9,10,18)	(19,16,13,8,2, 1,5,9,10,18)	(19,16,10,8,2, 1,5,9,13,18)
P6	(92,65,23,9,3 2,6,21,34,82)	(92,65,34,9,6, 3,2,21,23,82)	(92,65,34,9,6, 3,2,21,23,82)	(92,65,34,9,6, 2,6,21,34,82)
P7	(100,25,13,9,7, 5,8,10,21,41)	(100,25,21,9,8, 5,7,10,13,41)	(100,25,21,9,8, 5,7,10,13,41)	(100,25,13,9,7, 5,8,10,21,41)

This comparison is shown in Table 4.2. It can be seen from the Table 4.2 that the sequences obtained using TSDD and the SMV are identical. Kanet[1981] proved that the SMV performed better than the variance minimization heuristics suggested by Eilon and Chowdhary[1977]. Therefore one can conclude that the TSDD is also an improvement on E & C 's heuristics. Further, Table 4.2 shows that the TADC heuristic and the ST algorithm provide identical sequences when the earliness and tardiness carry equal weights.

4.2 PANWALKER ET AL.[1981]'S STUDY

Panwalker et al. [1981] considered the problem of minimizing a penalty function based on an optimal due date and optimal sequence. They considered a linear penalty function which includes cost incurred due to a) meeting a due date,b) earliness, and c) tardiness of jobs. They proved the following two lemmas :-

(i)for any specified sequence σ,there exists an optimal value of d which coincides with the completion times of one of the jobs in that sequence

(ii)for any specified sequence ,there exists an optimal due date equal to $C_{[k]}$, where k is the smallest integral value greater then or equal to the following function :

n(P3 - P1)/(P2 + P3)

Panwalker et al.[1981] also provided the following results :

Result 1 : $d^* < C_{[i]} = \sum_{i=1}^{n} P_i$, $i \in N$, where d^* denotes the optimal due date.

Result 2 : If P1 $\geq$ P3, $d^* = 0$ and SPT is optimal.

Result 3 : The $\sum_{j=1}^{n} \gamma_j \, p_{[j]}$ is minimized by matching the smallest value of with the largest value of p ,the next larger value γ with the next smaller value of p, and so on . Based on above lemmas and results, they developed an algorithm to find an optimal sequence and optimal due date. The algorithm is as follows :

ALGORITHM

PHASE 1.

Step 1.1 set k' = n(P3 - P1)/(P2 + P3).

Step 1.2 check if k' > 0.
If YES : go to step 1.3
If NO : set d = 0.
SPT sequence is Optimal.
STOP.

Step 1.3 Check if k' is an integer.
If YES : set k = k'.
proceed to PHASE 2.
If NO : set k equal to the smallest integer value greater than k'.

PHASE 2

Step 2.1 label position j (1 < j < n) as

$$\gamma_j = \begin{cases} nP1 + (j-1)P2, & 1 < j < k \\ (n+1-j)P3, & k+1 < j < n \end{cases}$$

Step 2.2 rank the positional labels γ_j in descending order of magnitude such that the largest γ_j is ranked 1 and the smallest γ_j is ranked n. Break ties arbitrarily.

Step 2.3 Obtain the optimal sequence such that job i is scheduled in position j corresponding to γ_j ranked in position i.

Step 2.4 Set $d^* = \{ p_1 + p_2 + \ldots + p_k \}$. STOP

EXAMPLE :

Given seven jobs with $p_1 = 3$, $p_2 = 4$, $p_3 = 6$, $p_4 = 9$, $p_5 = 14$, $p_6 = 18$, $p_7 = 20$. The penalties are P1 = 5, P2 = 11, and P3 = 18.

From Phase 1 of the algorithm, we get k' = 3.13 and thus k = 4. The seven positional labels and their ranks are as follows :-

Position j :	1	2	3	4	5	6	7
γ_j :	35	46	57	68	54	36	18
Rank i :	6	4	2	1	3	5	7

Optimal sequence : 6, 4, 2, 1, 3, 5, 7.

$$d^{*} : p_6 + p_4 + p_2 + p_1 = 34.$$

Total penalty = 1644.

ST Procedure With Penalty Function : The ST procedure described in the previous section can be applied while a more general penalty function is provided as an objective function to be minimized. The only modifications required are as follows :

1. Computation of the Location of Optimal Due-Date : The location of the optimal due date is determined by using equation (3.13) of Chapter 3. We compute the location r of the optimal due date as follows :

$$\frac{nP3 - P1}{P3 + P2} \leq r < \frac{nP3 - P1}{P3 + P2} + 1$$

where r is an integer value.

2. Computation of Total Penalty : The objective function value i.e. $Z(\sigma)$ is computed in a way similar to the algorithm of Panwalker et al[1981]. The equation of the objective function is as follows :

$$Z = P1k + P3\sum_{i=1}^{n} d_{i1} + P2\sum_{i=1}^{n} d_{i2}$$

where i = 1,2,3, ..., n.

NUMERICAL EXAMPLE : We will solve a numerical example to illustrate the procedure A problem of 7 jobs is considered which has already been considered in Panwalker et al.[1981]. The solution is as follows :-

Jobs :	J1	J2	J3	J4	J5	J6	J7
p_i's :	3	4	6	9	14	18	20

where P1 = 5, P2 =11, and P3 = 18 are the costs associated with the due date, earliness and tardiness of the jobs respectively.

U :(3 4 6 9 14 18 20)

The following equation :(see chapter 3, (3.10) and (3.12)

$[(7)(18) - 11]/(18 + 11) < r < [(7)(18) - 11]/(18 + 11)$

gives us the location of optimal due date, r = 5.

INITIALIZATION

σ_0 :	(14	9	6	4	3	18	20)
$C_i(\sigma_0)$:	(14	23	29	33	36	54	74)
$d_{ij}(\sigma_0)$:	(22	13	7	3	0	18	38)

$Z(\sigma_0) = 1683$ and $Z(\sigma_*) := 1683$

ITERATION #1

exchange : $J_1(\sigma_0)$ with $J_6(\sigma_0)$

σ_1 :	(18	9	6	4	3	14	20)
$C_i(\sigma_1)$:	(18	27	33	37	40	54	74)
$d_i(\sigma_1)$:	(22	13	7	3	0	14	30)

$(Z(\sigma_1) = 1487) \leq (Z(\sigma_*) = 1683)$

$\Rightarrow Z(\sigma_*) := 1487$

ITERATION #2

exchange : $J_1(\sigma_1)$ with $J_7(\sigma_1)$

$J_2(\sigma_1)$ with $J_6(\sigma_1)$

σ_2 :	(20	14	6	4	3	9	18)
$C_i(\sigma_2)$:	(20	34	40	44	47	56	74)
$d_{ij}(\sigma_2)$:	(27	13	7	3	0	9	27)

$(Z(\sigma_2) = 1433 \leq Z(\sigma_*) = 1487)$

$\Rightarrow \quad Z(\sigma_*) := 1433$

ITERATION #3

exchange : $J_2(\sigma_2)$ with $J_7(\sigma_2)$

J ( ) WITH J ( )

σ_3 :	(20	18	9	4	3	6	14)
$C_i(\sigma_3)$:	(20	38	47	51	54	60	74)
$d_{ij}(\sigma_3)$:	(34	16	7	3	0	6	20)

$(Z(\sigma_3) = 1398 \leq Z(\sigma_*))$

$\Rightarrow \quad Z(\sigma_*) := 1398$

ITERATION #4

exchange : $J_3(\sigma_3)$ with $J_7(\sigma_3)$

$J_4(\sigma_3)$ with $J_6(\sigma_3)$

σ_4 :	(20	18	14	6	3	4	9)
$C_i(\sigma_4)$:	(20	38	52	58	61	65	74)
$d_{ij}(\sigma_4)$:	(41	23	9	3	0	4	13)

$(Z(\sigma_4) = 1447 > Z(\sigma_*) = 1398$

$\Rightarrow$ OPTIMALITY IS ACHIEVED

$Z(\sigma_*) = 1398$ and $d(\sigma_*) = 54$

COMPARISON : Here the algorithm of Panwalker et al[1981] is compared with the procedure developed on the basis of theory discussed in Chapter 3. The comparison is performed on the same set of seven problems tabulated in Table 4.1 from Eilon and Chowdhary[1977].

Table 4.3

Comparative Studies

	PANWALKER ET AL.'s STUDY				ST PROCEDURE's RESULTS			
	Z	d	E'	T'	Z	d	E'	T'
P1	1287	36	27	45	1085	58	13	61
P2	3182	79	21	142	2499	161	110	33
P3	1483	25	16	71	1412	42	65	35
P4	1860	31	29	77	1841	32	72	35
P5	3215	40	45	140	2824	86	23	150
P6	9488	105	67	457	6913	200	189	213
P7	6885	60	75	320	4494	200	213	189)

From Section 4.2, it can be seen that the the ST procedure provides a different approach for the determination of the parameter r, the location of the optimal due date. Consequently our algorithm provides an optimal sequence for which the total penalty is lower than the sequence obtained using Panwalker et al.'s algorithm. However, the optimal due date obtained using the ST procedure is higher than those achieved by Panwalker et al. See Table 4.3. The only argument is that the objective function consists of three major components and the objective is to minimize the total penalty function and not the due date, it appears that our algorithm provides better results than Panwalker et al.'s algorithm.

4.3 SUNDARARAGHAVAN AND AHMED[1984]'S STUDY

So far in this Chapter we have considered the situations in which we are required to determine an optimal sequence and optimal due date. Generally due dates are given parameters and are not the decision-variables. These dates are normally specified through a process which is beyond the control of the scheduler. For example, due dates may be specified by the marketing department or derived from assembly schedules the production department. Sundararaghavan and Ahmed[1984] provided an algorithm which determines a sequence when a common due date is given. This algorithm is as follows :

Algorithm

The various steps are as follows :

Step 1 : $MS = \sum_{i=1}^{n} p_i$, $i = n$, $R = MS - d$, $L = d$, $A = \emptyset$ and $B = \emptyset$.

Step 2:

If $R \geq L$ then $A \leftarrow A \cup i$.

If $R < L$ then $B \leftarrow B \cup i$.

If $i = 1$, go to step 4.

Step 3 :

$R \leftarrow R - p_i$ if $i \in A$.

$L \leftarrow L - p_i$ if $i \in B$.

$i \leftarrow i - 1$.

COMPARISON BETWEEN SUNDARARAGHAVAN AND AHMED (1984)

AND ST ALGORITHM (TABLE 4.4)

Problem No.	No. of jobs	Processing times	Due date	Optimal sequence	Objective function value	Heuristic sequence (Algorithm 3)	Objective function value	Due date	ST ALGORITHM Optimal sequence
1	6	(1,10,11,48,50,53)	90	(5,4,1,2,3,6)	189	(6,4,1,2,3,5)	198	90	(5,4,1,2,3,6)
2	6	(22,25,36,65,73,84)	150	(6,3,2,1,4,5)	355	(5,4,1,2,3,6)	360	150	(6,3,2,1,4,5)
3	6	(3,37,57,75,81,99)	180	(6,4,1,2,3,5)	387	(6,3,2,1,4,5)	397	180	(6,4,1,2,3,5)
4	6	(7,10,11,69,77,92)	140	(5,4,1,2,3,6)	265	(6,3,2,1,4,5)	307	140	(5,4,1,2,3,6)
5	9	(3,6,7,8,14,30,45, 48,72)	130	(9,7,3,2,1,4,5,6,8)	274	(9,7,4,2,1,3,5,6,8)	275	130	(9,7,3,2,1,4,5,6,8)
6	14	(1,2,4,5,8,23,31, 53,55,65,68,69, 90,92)	290	(13,11,10,8,5,4,1, 2,3,6,7,9,12,14)	1092	(14,12,10,8,5,2,1, 3,4,6,7,9,11,13)	1094	290	(13,11,10,8,5,4,1, 2,3,6,7,9,12,14)
7	14	(4,13,14,21,33,37, 38,48,63,78,80, 85,93,94)	360	(14,12,10,8,6,3,1, 2,4,5,7,9,11,13)	1603	(14,12,10,7,6,4,1, 2,3,5,8,9,11,13)	1606	360	(14,12,10,8,6,3,1, 2,4,5,7,9,11,13)
8	14	(8,21,23,29,35,36, 44,51,52,62,69, 81,84,85)	320	(13,11,9,7,5,4,1, 2,3,6,8,10,12,14)	1742	(13,11,9,7,5,3,1, 2,4,6,8,10,12,14)	1746	320	(13,11,9,7,5,4,1, 2,3,6,8,10,12,14)
9	14	(1,2,4,5,8,23,31, 53,55,65,68,69, 90,92)	340	(14,13,11,9,6,5,3, 1,2,4,7,8,10,12)	1073	(14,13,11,8,6,5,3, 2,1,4,7,9,10,12)	1074	340	(14,13,11,9,6,5,3, 1,2,4,7,8,10,12)
10	14	(4,13,14,21,33,37, 38,48,63,78,80, 85,93,94)	250	(12,9,7,6,3,2,1,4, 5,8,10,11,13,14)	1820	(11,9,8,5,3,2,1,4, 6,7,10,12,13,14)	1821	250	(12,9,7,6,3,2,1,4, 5,8,10,11,13,14)
11	14	(4,13,14,21,33,37, 38,48,63,78,80, 85,93,94)	425	(14,13,11,9,7,5,4, 1,2,3,6,8,10,12)	1594	(14,13,11,9,7,5,3, 2,1,4,6,8,10,12)	1596	425	(14,13,11,9,7,5,4, 1,2,3,6,8,10,12)

Step 4 : Obtain a schedule by concatenating elements of B and A where elements of B are arranged in first-come first-serve order and the starting time is zero.

NUMERICAL EXAMPLE : We will present a numerical example to illustrate the algorithm. This is a problem of 14 jobs with varying processing-times and a due date of 290 . This problem has already been solved by our algorithm for (P2) in chapter 3. Sundararaghavan and Ahmed[1984]'s algorithm is applied as follows :

Jobs	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12
p_i's	1	2	4	5	8	23	31	53	55	65	68	69

J13	J14
90	92.

and $d = 290$.

Step 1 : MS = 566 ; i = 14, R = 566 - 290 = 276,

L = 290; A = $\emptyset$ and B = $\emptyset$.

Step 2 : A = {J1, J2, J4, J5, J6, J8, J9, J11, J13}

B = {J14, J12, J10, J7, J3 }

STEP 3 : The values taken on are

R = 276; 186; 118; 63; 32; 9; 4; 0

L = 290; 198; 129; 64; 11; 3; 2; 1

i = 14; 13; 12; 11; 10; 9 ; 8; 7; 6; 5; 4; 3; 2; 1; 0

Step 4 : Schedule j = (B,A)

(J14, J13, J12, J11, J10, J9, J8, J7, J6, J5, J4, J3, J2,J1) is the schedule found which gives objective function value

Z = 1094 whereas

optimal Z = 1092.

Thus it is not possible to obtain the optimal schedule exactly whereas the optimal solution has been achieved by the using goal programming approach suggested in Chapter 3. In fact, our algorithm is also capable of recoganizing the multiple optimal schedules, if any. Table 4.4 compares the results obtained by using our algorithm and those obtained by Sundararaghavan and Ahmed[1984] and shows that, in all problems for which their algorithm could not determine an optimal sequence, our algorithm does indeed provide the optimal sequence along with alternatives, if any.

4.4 SCHEDULING

In this section, the determination of an optimal schedule when due date is greater than or equal to makespan is discussed. Kanet[1981] suggested the development of an optimal schedule using a Gantt chart. This method is extremely simple and easy to understand for a small number of jobs. However, as the number of jobs increases, the method becomes cumbersome. Therefore, we propose the use of goal programming in determining an optimal schedule. First of all, Kanet's approach is presented and then, the goal programming approach is discussed.

Kanet's Approach : In order to develop an optimal schedule, Kanet[1981] proposed the following procedure which will be illustrated by the Gantt chart for an example.

Procedure Sched

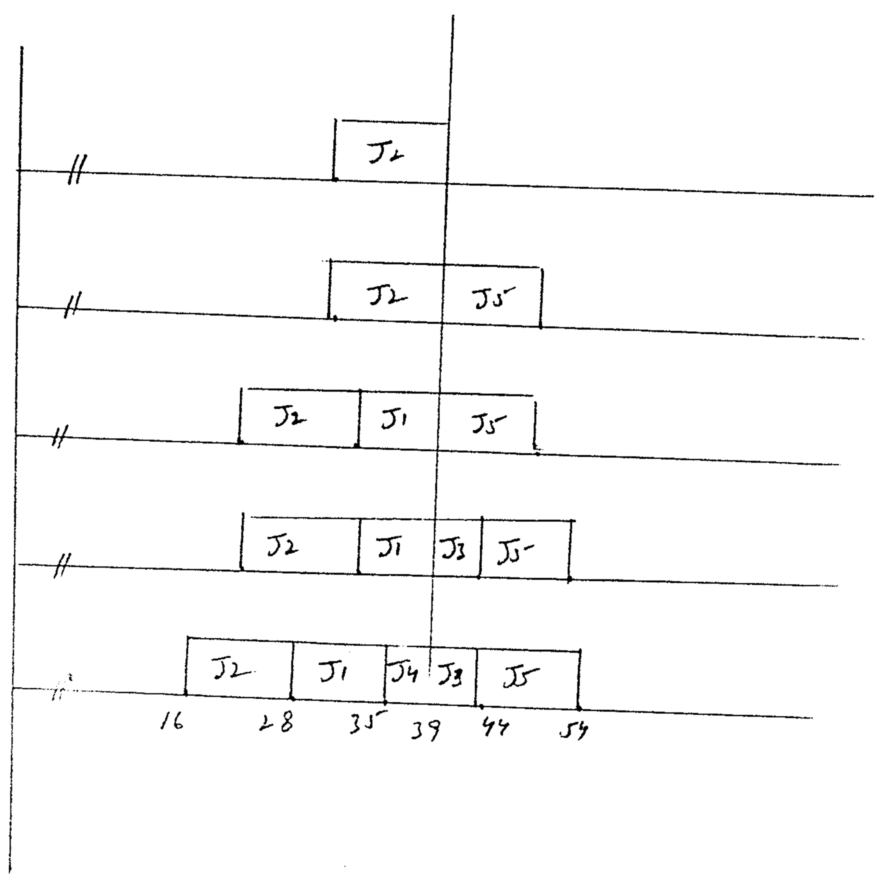
```
B <- A <- 0 ;
```

```
  While (U ≠ 0) do
  begin
   remove a job K from U such that p_K = max { p_[K]};
   insert job K into the last position in B        ;
   if { U ≠ 0 } do
     remove a job K from U such that p_K = max { p_[K]};
     insert job K into the first position in A     ;
     end;
  end;
  s <- {B,A};
end Sched.
```

NUMERICAL EXAMPLE : We will solve the following problem of 5 jobs with the help of a Gantt chart to illustrate the procedure.

Jobs;	J1	J2	J3	J4	J5	
p_i's :	7	12	5	4	10	; $d = 39$

We will develop the following Gantt chart following the procedure described above in order to find the optimal schedule as follows:



Goal Programming Approach :

Kanet's approach is essentially the the TADC heuristic which has been used to draw the Gantt chart. In the previous section, (4.1), we see that the ST algorithm and the TADC heuristic result in identical sequences. So it is suggested that the optimal sequence is determined using the ST procedure or the TADC heuristic and then it could be employed with the following goal programming model to find the optimal schedule. Goal programming model can be expressed as below:

G.P. Model :

$$
\begin{aligned}
\text{MIN} \quad & Z(S) = \sum_{i=1}^{n} \{ D_{i1} + D_{i2} \} \qquad \forall\, i = 1,2,\ldots,n. \\
\text{S.T.} \quad & \\
& K - T + D_{11} - D_{12} = C_1 \\
& K - T + D_{21} - D_{22} = C_2 \\
& \vdots \\
& K - T + D_{n1} - D_{n2} = C_n \\
& K \geq MS \\
& K, T, D_{i1}, D_{i2} \geq 0
\end{aligned}
$$

for all i = 1,2,3,...,n.

Procedure :

Step 1 : Find the optimal sequence using the TADC heuristic or the ST procedure.

Step 2 : Formulate the goal programming model as discussed earlier and from that find the value of decision variable T which represents the starting time of the schedule. Once the value of T is determined by solving the model, the optimal schedule is easily obtained by processing the jobs starting at time T in the order of optimal sequence.

To sum up, in this chapter various heuristics and procedures available in the literature were evaluated and compared with respect to find optimal due date, sequence and schedule. Then these were compared with the algorithm developed in Chapter 3, showing that our algorithm is more efficient and performs better.

Chapter V

FURTER EXTENSIONS

In the present thesis we have considered a deterministic single-machine job-shop sequencing and scheduling problems. Although there are a number of real life systems which can be modeled as single-machine systems but majority of job-shop operations have several machines. It is, therefore, would be of interest to make the study more realistic by attempting to generalized linear programming theory developed in Chapter 3 for the case of multiple machines case.

Another interesting possibility would be to extend the algorithms and heuristics developed in this thesis for the determination of optimal sequence when each job is provided with its own due-date under single and multiple machine job-shop situations.

The models presented in this thesis can further be developed to take into consideration the stochastic nature of job-shop which may include availability and reliability of machines.

The linear goal programming approach suggested in this work, currently minimizes the absolute completion times deviations. It should be of interest to generalize this approach to include the criterion of minimization of completion times variance.

The literature dealing with sequencing and scheduling problems under different priority rules is sparse. It would be useful to determine the optimal sequence schedule under priority rules by extending the algorithms developed in this thesis.

BIBLIOGRAPHY

Aho, A.V., Hopcraft, J.E. and Ullman, J.D., "The Design and Analysis of Computer Algorithms". Addison-Wesley, Reading, Massachusetts, 1974.

Baker, K.R.,"Introduction to Sequencing and Scheduling". John Willey, New York, 1974.

Baker, K.R. and Bertrand, J.W.M.,"A Comparison of Due-date Selection Rules", AIIE Transactions, Vol. 13, no. 2, 1981.

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

Burns, R.N.,"Scheduling to Minimize the Weighted Sum of Completion-Times with Secondary Criteria", Naval Research Logistics Quartly, Vol. 21, 1974.

Cheng, T.C.E.,"Optimal Due-Date Determination and Sequencing of n Jobs on a Single Machine", Journal of Operations Research Society, Vol. 35, no. 5, 1984.

-------------,"A Duality Approach to Optimal Due-Date Determination", Engineering Optimization, Vol. 9, 1985.

Coffman, E.G.(Editor), Computer and Job-shop Scheduling Theory, John Willey, New York, 1976.

Conway, R.W.,"Priority Dispatching and Job Lateness in a Job-shop", Journal of Industrial Engineering, Vol. 16, 1965.

-----------, William, L.M. and Miller, W.M., " Theory of Scheduling ", Addison Wesley Publishing Company, Reading, Massachusetts, 1967.

Day, J.E. and Hottenstein, M.P.,"Review of Sequencing Research", Naval Research Logistics Quartly, Vol. 17, no. 1, 1970.

Eilon, S.,"Production Scheduling". in OR' 78 (editor Hayley, K.B.), pp 237-266, North-Holland, Amesterdam, 1979.

---------and King, J.R.," Production Scheduling Abstracts, Oliver and Boyd, Edinburgh, 1967.

-------- and Chowdhary, I.S.,"Due-dates in Job-shop Scheduling", International Journal of Production Research, Vol. 14, no. 2, 1976.

________ and ______________,"Minimizing Waiting-time Variance in Single Machine Problem", Management Science, Vol. 23, no. 6, 1977.

Elmaghraby, S.E.,"Review of Sequencing Research and Extension", Naval Research Logistics Quartly, Vol. 15, No. 2, 1968.

----------------,"The One machine Sequencing Problem with Delay Costs", Journal of Industrial engineering, Vol. 19, NO. 2, 1968.

Emmons, H.,"One Machine Sequencing to Minimize certain functions of Job Lateness", Operations Research, Vol. 17, No. 4, 1969.

---------," One Machine Sequencing to Minimize Mean Flow-Time With Minimum Number Tardy", Naval Research Logisticss Quartly, Vol. 22, No. 3, 1975.

French, S., "Sequencing and Scheduling", John Willey, New York, 1982.

Fry, T.D. and Leong, G.K.,"Bi-Criterion Single Machine Scheduling with Forbidden Early Shipments", Engineering costs and Production Economics, Vol. 10, 1986.

Gilmore, P.C. and Gomory, R.E.,"Sequencing A One State-Variable Machine: A solvable case of the travelling salesman problem", Operations Research, Vol. 12, No. 5, 1964.

Godin, V.B.,"Interactive Scheduling: Historical Survey and State of Art, AIIE Transactions, Vol. 10, 1978.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinooy Kan A.H.G.," Optimization and Approximation in Deterministic Sequencing and Scheduling: a survey", Anal Discrete Mathematics, Vol. 5, 1979.

Graves, S.C., "A Review of Production Scheduling", Operations Research, Vol. 29, No. 4, 1981.

Gupta, J.N.D.," M-stage Scheduling Problem- a critical appraisal", International Journal of Production Research, Vol. 9, no. 2, 1971.

Gupta, S.K. and Sen, T.,"Minimizing a Quadratic Function Of Job Lateness on a Single Machine", Engineering Costs and Production Economics, Vol. 7, 1983.

___________ and _______,"A State-of-art Survey of Static Scheduling Research Involving Due-Dates", OMEGA, Vol. 12, No. 1, 1984.

Heck, H. and Roberts, S.,"A note on The Extensions of A Result on Scheduling with Secondary Criteria", Naval Research Logistics Quartly, Vol. 19, 1972.

Hellor, J.,"Some Numerical Experiments for M*J Flow-Shops and Its Discussion- Theoretical Aspects", Operations Research, Vol. 8, No. 12, 1960.

Hodgson, T.J.,"A note on Single Machine Sequencing with Random Processing Times", Management Science, Vol. 23, NO. 10, 1977.

Jackson, J.R.,"Scheduling A Production Line to Minimize Maximum Tardiness", Management Science Research Project, Ucla, 1975.

Jeremiah, B., Lalchandani,A. and Schrage, L.,"Heuristic Rules for Towards Optimal Scheduling", Research Report, Department of Industrial Engineering, Cornell University, 1964.

Kanet, J.J.,"Minimizing the Average Deviation of Completion Times about a Common Due-Date", Naval Research Logistics Quartly, Vol. 28, No. 4, 1981.

-----------,"Minimizing Variation of Flow_time in Single-Machine Systems", Management Science, Vol. 27, no. 12, 1981.

Lakshminarayan, L., Lakshmanan, L, Papineau, R.L. and Rochette,R.," Optimal Single-Machine scheduling With Earliness and Tardiness Penalties", OPerations Research. Vol. 26, No. 6, 1978.

Lawler, E.L.,"On Scheduling Problems with Deferal Costs", Management Science, Vol. 11, No. 2, 1964

Maxwell, W.L.,"On Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs", Management Science, Vol. 16, No. 1, 1970.

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

Meller, P.,"A Review Of Job-shop Scheduling", Operations Research Quartly, Vol. 17, No. 2, 1966.

Merten, A.G. and Muller, M.E.,"Variance Minimization in Single Machine Sequencing Problems", Management Science, Vol. 18, No. 9, 1972.

Miller, R.E. and Thacher, J.W.(editors), "Complexity of Computer Computations", Plenum Press, New York, 1972.

Moore, J.M.,"An n-Jobs, One-Machine Sequencing Algorithm for Minimizing the Number of Late Jobs", Management Science, Vol. 18, No. 9, 1972.

----------- and Wilson, R.C.,"A Review Of Simulation Research in Job-shop Scheduling", Production Inventory Management, Vol. 8, 1967

Panwalker, S.S., Seidman, A. and Smith, M.L.,"Common Due-Date Assignment to minimize Total Penality for the one Machine Scheduling Problem", Operations Research, Vol. 30, No. 2, 1982.

---------------and Iskiander, W.,"A Survey of Scheduling Rules", Operations Research, Vol. 25, No.1, 1977.

Salvador, M.S.," Sequencing And Scheduling". In 'Handbook of Operations Research: Models and Applications', (editors Moder, J. and Elmaghraby S.E.), Vol. 2, Van Nostrand, Florence, Ky.

Schild, A. and Fredman, I.J., "Scheduling Tasks with Deadlines and Non-linear Loss Functions", Management Science, Vol. 9, No. 1, 1962.

Schrage, L., "Minimizing the Time_in-system Variance for a Finite Job Set", Management Science, Vol. 21, No. 5, 1975.

Seidman, A., Panwalker, S.S. and Smith, M.L.,"Optimal Assignment of Due-Date for a Single Processor Scheduling Problem", International Journal Of Production Research, Vol. 19, 1981.

Sturm, L.B.J.M.,"A Simple Optimality Proof of Moore's Sequencing Problem", Management Science, Vol. 17, No. 1, 1970.

Wassenhove, L.N.V. and Gelders, L.F.,"Solving a Bi-Criterion Scheduling Problem",' European Journal of Operations Research, Vol. 4, 1980

Weeks, J.K., "A Simulation Study of Predictable Due-Dates", Management Science, Vol. 25,1979.

_________ and Fryer, J.S.,"A Methodology for Assigning Minimum cost Due-Dates", Management Science, Vol. 23, 1979.