ON CTV MINIMIZATION IN SINGLE MACHINE SCHEDULING

by

D. K. Manna
SQC & OR Unit
Indian Statistical Institute
110 Nelson Manickam Road
Aminjikarai
Madras 600 029

To my mother and wife

ACKNOWLEDGEMENTS

I wish to convey my thanks to all those who helped me in various ways in bringing out this thesis.

I express my deep sense of gratitude to Prof. T. S. Arthanari who primarily motivated and inspired me to take up research work in the field of Scheduling, besides being co-author in one of our papers.

I owe a special debt of gratitude to Dr. V. Rajendra Prasad, my thesis supervisor, for his critical comments, valuable guidance and continuous encouragement throughout the dissertation period.

I am indebted to Prof. A. R. D. Prasad and Dr. D. Mohan for their kind cooperation in extending the computing facility at ISI, Bangalore.

My sincere thanks are due to Prof. B. L. S. Prakasa Rao, Ex-Director, Prof. J. K. Ghosh, Ex-Director, Prof. B. K. Pal, Head, SQC & OR Division, and Mr. V. V. Iyer, Executive-in-Charge, SQC & OR Unit, Madras for extending all the necessary facilities to me. I am also thankful to Mr. V. Balachandran for his help in typing a substantial part of the manuscript.

I deem it a privilege to acknowledge the excellent association of all the staff members at SQC & OR Unit, ISI, Madras.

D. K. MANNA

January, 1996

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 SCHEDULING PROBLEM

Scheduling problems are quite common in real life. They arise whenever there is a need to plan execution of various tasks over time and therefore they play very important roles in commercial set-ups concerning manufacturing or service in the optimal use of resources and/or customer's satisfaction.

The theory of scheduling deals with the construction of suitable models and their analyses. Researchers' attention was drawn to the study of scheduling problems using mathematical modeling, probably for the first time when Johnson [1954] published his famous work on flowshop problem. Since then, the study of scheduling problem and its context has gradually attracted the researchers from various other fields. At present, a vast collection of research work on this area is available in the literature. For a comprehensive study of the subject, one may refer to Conway, Maxwell and Miller [1967], Baker [1974], Lawler, Lenstra, Rinnooy Kan and Shmoys [1990], Rinnooy Kan [1976], French [1982] etc.

The formulation and analysis of mathematical models representing scheduling problems involve the operations research techniques such as combinatorial analysis, dynamic programming, integer programming, network analysis etc.

Depending upon the nature of the problem, the scheduling problems are classified into several groups, namely, single machine, Flowshop, Jobshop, Parallel machines etc. Currently, the single machine scheduling problem has been of keen interest to many researchers. Each of the above classes of problems is further treated in two different ways, viz, using (*a*) deterministic model and (*b*) non-deterministic (stochastic) model.

A large number of deterministic models are designed to represent scheduling problems. The essential feature of these models is that they are combinatorial in nature, and unfortunately, the available mathematical tools are not adequate enough to cope with such problem efficiently. As a matter of fact, the majority of these problems are recognized as difficult ones. The very recent trend has been to develop pseudopolynomial procedures for the complex scheduling problems.

An extensive literature is available on the single machine scheduling starting from the work of Smith [1956]. (For reference, see the review papers of Sen and Gupta [1984], Emmons [1987], Gupta and Kyparisis [1987], Raghavachari [1988]. Cheng and Gupta [1989], Baker and Scudder [1990] etc.) A major part of the literature deals with the natural early-tardy problem with various cost structures as objective functions where the due dates of the jobs are either fixed or treated as decision variables.

In single machine scheduling, the common objectives considered individually or in combination are minimization of :

(1) Average Completion Time : $\bar{C} = \frac{1}{n}\sum_{j=1}^{n} C_j$
where C_j = completion time of job j,

(2a) Average Lateness : $\bar{L} = \frac{1}{n}\sum_{j=1}^{n} L_j$
(2b) Maximum Lateness : $L_{\max} = \max_{1\le j\le n} L_j$
where L_j = lateness of job j,

(3a) Maximum Tardiness : $T_{\max} = \max_{1\le j\le n} T_j$
(3b) Number of Tardy Jobs : $|\{j : T_j > 0\}|$
where T_j = tardiness of job j,

(4) Early-Tardy Penalty : $\sum_{C_j\le d} u_j f(C_j - d) + \sum_{C_j>d} v_j f(C_j - d)$
where C_j = completion time of job j,
d = common due date of the jobs,
u_j (v_j) = weightage associated with early (tardy) jobs,
$f(x) = |x|$ or x^2.

The above objective (4) has drawn great interest from the researchers due to the current emphasis on Just-in-time (JIT) production philosophy which espouses the notion that earliness as well as tardiness should be discouraged. (See for reference Bagchi, Sullivan and Chang [1989], Hall, Kubiak and Sethi [1991], Kahlbacher [1989], Panwalker, Smith and Siedmann [1982] etc..) An important special case of this objective is the variance of job completion times which is not a regular measure of performance.

In this thesis, we study the problem of scheduling jobs on a single machine so as to minimize the variance of job completion times. This problem is usually referred to as CTV problem. The problem arises in computer file management and is also applicable to many manufacturing as well as service facilities. It is very relevant to the current emphasis on Just-in-time production philosophy. The objective is especially important in situations where it is desirable to provide customers or jobs with approximately the same treatment. This problem was first considered by Merten and Muller [1972] in the context of organization of large computer data files in order to provide uniform response times to the users. Since then, it has been drawing the attention of several researchers. The importance of this problem is further highlighted due to its relationship with the so called *MSD problem* (a special case of objective (4)) as shown by Bagchi, Sullivan and Chang [1987], Raghavachari [1988], De, Ghosh and Wells [1989] and Weng and Ventura [1994].

1.2 DESCRIPTION OF CTV PROBLEM

Throughout this thesis, we consider only completion time variance (CTV) minimization problem in single machines scheduling with either deterministic or random processing times. The CTV problem is defined through the following assumptions :

A1 : There are n jobs to be processed on the single machine. The set of jobs is denoted by $N = \{1, 2, \ldots, n\}$.

A2 : Initially at time $t = 0$, all the n jobs are ready to be processed and the machine is available.

A3 : The machine is continuously available until all the n jobs are processed, that is, no breakdown or maintenance of the machine is considered.

A4 : The machine can process only one job at any instance of time.

A5 : Each job, once started, must be performed to completion before another job can start, that is, neither job cancellation nor preemption is allowed. However, idle time of the machine may be allowed.

A6 : There is no precedence relation among the jobs, that is, jobs can be processed in any order.

A7 : A job j, $j \in N$, requires a processing time, p_j, that includes setup time. These processing times are fixed and known. We assume, without loss of generality, that $p_1 \geq p_2 \geq \ldots \geq p_n$. In the stochastic case, the processing times P_j's (also including setup time) are random and independent, and each P_j has known finite mean (μ_j) and variance (σ_j^2). In this case, we assume that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$ and, $\sigma_i^2 \geq \sigma_j^2$ whenever $\mu_i = \mu_j$ for $i < j$.

For a scheduling problem with the above assumptions, a schedule can be represented by (a) a permutation (sequence) of the jobs, say, $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, and (b) start-time (or equivalently completion time) of each job.

Given any sequence π, let θ_r denote the machine idle time just before the processing of rth job $(1 \leq r \leq n)$ starts and let $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$. We need to know π and θ to describe any schedule. Therefore, we denote a schedule S by $S(\pi, \theta)$.

For a schedule $S(\pi, \theta)$, let $C_{[r]}(S(\pi, \theta))$ denote the completion time of the job in rth position of the sequence π for $r = 1, 2, \ldots, n$ and $\bar{C}(S(\pi, \theta))$ be the average of the job completion times. Therefore, for the schedule $S(\pi, \theta)$, the variance of the job completion times, denoted by $V(S(\pi,\theta))$, is given by

$$V(S(\pi,\theta)) = \frac{1}{n}\sum_{r=1}^{n}\left[C_{[r]}(S(\pi,\theta)) - \bar{C}(S(\pi,\theta))\right]^2. \qquad (1.2.1)$$

The CTV problem is said to be deterministic (stochastic) if the job processing times are fixed (random).

Hence, the deterministic CTV problem is to find a schedule $S(\pi^*,\theta^*)$ such that

$$V(S(\pi^*,\theta^*)) = \min_{\text{all possible } S(\pi,\theta)} V(S(\pi,\theta)) \qquad (1.2.2)$$

and the stochastic CTV problem is to find a schedule $S(\pi^*,\theta^*)$ such that the expected value of CTV is minimum, that is,

$$E[V(S(\pi^*,\theta^*))] = \min_{\text{all possible } S(\pi,\theta)} E[V(S(\pi,\theta))] \qquad (1.2.3)$$

where $E[.]$ stands for the expected value.

Definition 1.2.1 : A schedule $S(\pi,\theta)$ is called *no idletime* (NI) schedule if $\theta_r = 0$ for each $1 \le r \le n$.

Remark 1.2.2 : An NI schedule is completely specified by a sequence of the jobs. Such a schedule is also referred to as *permutation schedule* and is represented simply by a permutation, say π, of the jobs.

While dealing with the CTV problems (both deterministic and stochastic), all the researchers have so far restricted to the set of permutation schedules only. In the following, we formally establish that, indeed, there exists an optimal permutation schedule for any CTV problem.

Deterministic Case :

Lemma 1.2.3 : For any schedule $S(\pi,\theta)$,

$$n^2V(S(\pi,\theta)) = \sum_{r=1}^{n}(r-1)(n-r+1)q_{\pi_r}^2 + 2\sum_{r=1}^{n-1}(r-1)q_{\pi_r}\sum_{s=r+1}^{n}(n-s+1)q_{\pi_s} \qquad (1.2.4)$$

where $q_{\pi_r} = \theta_r + p_{\pi_r}$ for $r = 1, 2, \ldots, n$.

Proof : The proof is straightforward. ∎

Remark 1.2.4 : It is clear from the equation (1.2.4) that for any schedule $S(\pi,\theta)$, $V(S(\pi,\theta))$ is non-decreasing in each q_{π_r} for $r = 1, 2, \ldots, n$.

Corollary 1.2.5 : Let $S(\pi,\theta)$ and $S(\pi,\theta')$ be two schedules such that $\theta_r = \theta'_r$ for all $2 \le r \le n$. Then

$$V(S(\pi,\theta)) = V(S(\pi,\theta')).$$

Proof : It follows directly from the Lemma 1.2.3. ∎

Lemma 1.2.6 : Let $S(\pi,\theta)$ and $S(\pi,\theta')$ be two schedules such that $\theta'_r \ge \theta_r$ for all $2 \le r \le n$ and $\theta'_r > \theta_r$ for some $2 \le r \le n$. Then

$$V(S(\pi,\theta)) < V(S(\pi,\theta')).$$

Proof : Let $q'_{\pi_r} = \theta'_r + p_{\pi_r}$ for all $2 \le r \le n$. Then $q_{\pi_r} = \theta_r + p_{\pi_r} \le \theta'_r + p_{\pi_r} = q'_{\pi_r}$ for all $2 \le r \le n$ and for at least one r, $q_{\pi_r} < q'_{\pi_r}$. Therefore, using Remark 1.2.4 and Corollary 1.2.5, we complete the proof. ∎

Corollary 1.2.7 : Let $S(\pi, \theta^0)$ be a schedule with $\theta_r^0 = 0$ for all $2 \leq r \leq n$. Then for any schedule $S(\pi, \theta)$,

$$V(S(\pi, \theta^0)) \leq V(S(\pi, \theta)).$$

Proof : It follows directly from the Lemma 1.2.6. ∎

Remark 1.2.8 : By Corollary 1.2.5, CTV is invariant of the initial idle time of the machine. Whereas, by Corollary 1.2.7, inserted idletime (between execution of successive jobs) of the machine increases CTV value.

Theorem 1.2.9 : There exists an NI optimal schedule for the deterministic CTV problem.

Proof : The proof is simple using the Corollaries 1.2.5 and 1.2.7. ∎

Therefore, we denote a schedule by $\pi = (\pi_1, \ldots, \pi_n)$, a sequence (permutation) of the jobs and in view of this, our problem is to find a sequence π^* such that

$$V(\pi^*) = \min_{\pi \in \Pi} V(\pi) \tag{1.2.5}$$

where

$$\begin{aligned} \Pi &= \text{The set of all } n! \text{ permutations of the jobs,} \\ V(\pi) &= \text{Variance of the job completion times} \\ &\quad \text{for the sequence } \pi \\ &= \frac{1}{n}\sum_{r=1}^{n}\left[C_{[r]}(\pi) - \bar{C}(\pi)\right]^2, \qquad (1.2.6) \\ C_{[r]}(\pi) &= \text{Completion time of job } \pi_r, \text{ the } r\text{th job in } \pi \\ &= \sum_{i=1}^{r} p_{\pi_i} \end{aligned}$$

$$\begin{aligned} \text{and } \bar{C}(\pi) &= \text{Average of job completion times for } \pi \\ &= \frac{1}{n}\sum_{r=1}^{n} C_{[r]}(\pi) \qquad (1.2.7) \\ &= \frac{1}{n}\sum_{i=1}^{n}(n-r+1)p_{\pi_i}. \qquad (1.2.8) \end{aligned}$$

In the present context, it is important to mention the *MSD problem* which is defined as the problem of scheduling nonpreemptive, independent jobs with zero ready times on a single machine so as to minimize the mean squared deviation (abbreviated as MSD) of job completion times about a given fixed common due date, say, d. Bagchi, Sullivan and Chang [1987], De, Ghosh and Wells [1989], Weng and Ventura [1994] have discussed the relation between CTV and MSD problems.

Depending upon the value of the common due date d, the three versions of the MSD problem are identified by Bagchi, Sullivan and Chang [1987] and De, Ghosh and Wells [1989]. They are (*i*) *Unrestricted :* when d is sufficiently large, (*ii*) *Tightly restricted :* when d is sufficiently small, and (*iii*) *Restricted.* De, Ghosh and Wells [1989] have given a procedure to identify the version of a given MSD problem. An *unrestricted* MSD problem is equivalent to the corresponding CTV problem, that is, solution of such an MSD problem can be obtained on translation by solving the corresponding CTV problem and vice versa. The same may or may not be applicable for *restricted* MSD problem. However, there is no relationship between the *tightly restricted* MSD problem and the CTV problem.

Stochastic Case :

Lemma 1.2.10 : For any schedule $S(\pi,\theta)$,

$$E[V(S(\pi,\theta))] = V_\mu(S(\pi,\theta)) + \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\sigma^2_{\pi r} \qquad (1.2.9)$$

where $V_\mu(S(\pi,\theta))$

$$
\begin{aligned}
&= \text{Variance of the expected completion times} \\
&\quad \text{of the jobs for the schedule } S(\pi,\theta) \\
&= \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)(\mu_{\pi_r}+\theta_r)^2 \\
&\quad +\frac{2}{n^2}\sum_{r=1}^{n-1}(r-1)(\mu_{\pi_r}+\theta_r)\sum_{s=r+1}^{n}(n-s+1)(\mu_{\pi_s}+\theta_s). \qquad (1.2.10)
\end{aligned}
$$

Proof : Let $(p_{\pi_1}, p_{\pi_2}, \ldots, p_{\pi_n})$ be any realization of $(P_{\pi_1}, P_{\pi_2}, \ldots, P_{\pi_n})$. Using Lemma 1.2.3, we have

$$
\begin{aligned}
&V(S(\pi,\theta)|(P_{\pi_1},\ldots P_{\pi_n}) = (p_{\pi_1},\ldots p_{\pi_n})) \\
&\quad = \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)(p_{\pi_r}+\theta_r)^2 \\
&\qquad + \frac{2}{n^2}\sum_{r=1}^{n-1}(r-1)(p_{\pi_r}+\theta_r)\sum_{s=r+1}^{n}(n-s+1)(p_{\pi_s}+\theta_s). \qquad (1.2.11)
\end{aligned}
$$

It now follows from the equation (1.2.11) that

$$
\begin{aligned}
E[V(S(\pi,\theta))] &= \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\left\{\sigma_{\pi_r}^2+(\mu_{\pi_r}+\theta_r)^2\right\} \\
&\quad +\frac{2}{n^2}\sum_{r=1}^{n-1}(r-1)(\mu_{\pi_r}+\theta_r)\sum_{s=r+1}^{n}(n-s+1)(\mu_{\pi_s}+\theta_s) \\
&= \left[\frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)(\mu_{\pi_r}+\theta_r)^2\right. \\
&\quad \left.+\frac{2}{n^2}\sum_{r=1}^{n-1}(r-1)(\mu_{\pi_r}+\theta_r)\sum_{s=r+1}^{n}(n-s+1)(\mu_{\pi_s}+\theta_s)\right] \\
&\quad +\frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\sigma_{\pi_r}^2 \\
&= V_\mu(S(\pi,\theta))+\frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\sigma_{\pi_r}^2.
\end{aligned}
$$

Hence the lemma holds. ■

Corollary 1.2.11 : Let $S(\pi, \theta)$ and $S(\pi, \theta')$ be two schedules such that $\theta_r = \theta'_r$ for all $2 \leq r \leq n$. Then

$$E[V(S(\pi,\theta))] = E[V(S(\pi,\theta'))].$$

Proof : By Lemma 1.2.10,

$$\begin{aligned} & E[V(S(\pi,\theta))] - E[V(S(\pi,\theta'))] \\ = \; & V_\mu(S(\pi,\theta)) - V_\mu(S(\pi,\theta')) \\ & \qquad \text{(using the equation (1.2.9))} \\ = \; & 0 \qquad \text{(using the equation (1.2.10))}. \end{aligned}$$

■

Lemma 1.2.12 : Let $S(\pi, \theta)$ and $S(\pi, \theta')$ be two schedules such that $\theta'_r \geq \theta_r$ for all $2 \leq r \leq n$ and $\theta'_r > \theta_r$ for some $2 \leq r \leq n$. Then

$$E[V(S(\pi,\theta))] < E[V(S(\pi,\theta'))].$$

Proof : Using Lemma 1.2.10, we get

$$\begin{aligned} & E[V(S(\pi,\theta))] - E[V(S(\pi,\theta'))] \\ = \; & V_\mu(S(\pi,\theta)) - V_\mu(S(\pi,\theta')). \end{aligned}$$

The remaining part of the proof is analogous to that of Lemma 1.2.6. ■

Corollary 1.2.13 : Let $S(\pi, \theta^0)$ be a schedule with $\theta^0_r = 0$ for all $2 \leq r \leq n$. Then for any schedule $S(\pi, \theta)$,

$$E[V(S(\pi,\theta^0))] \leq E[V(S(\pi,\theta))].$$

Proof : It follows directly from the Lemma 1.2.12. ■

Remark 1.2.14 : By Corollary 1.2.11, the expected CTV is invariant of the initial idle time of the machine. From Corollary 1.2.13, inserted idle time (between execution of successive jobs) of the machine leads to increase in expected CTV value.

Theorem 1.2.15 : There exists an NI optimal schedule for the stochastic CTV problem.

Proof : It is simple using the Corollaries 1.2.11 and 1.2.13. ∎

Hence, we designate a schedule simply by $\pi = (\pi_1, \ldots, \pi_n)$, a sequence of the jobs and the stochastic CTV problem is to find a sequence of the jobs for which the expected CTV value is minimum.

For a sequence $\pi = (\pi_1, \ldots, \pi_n)$, we denote $E_r(\pi)$ as the expected completion time of the job at the rth $(1 \leq r \leq n)$ position in π and $\bar{E}(\pi)$ as their average, that is, $E_r(\pi) = E[C_{[r]}(\pi)]$ for $r = 1, 2, \ldots, n$ and $\bar{E}(\pi) = E[\bar{C}(\pi)]$. Finally, $E[V(\pi)]$ represents the expected value of the CTV for a sequence π.

1.3 DEFINITIONS AND NOTATION

Definition 1.3.1 : Given a sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, its *dual* is defined by $\pi^D = (\pi_1^D, \pi_2^D, \ldots, \pi_n^D)$ where $\pi_1^D = \pi_1$ and $\pi_r^D = \pi_{n-r+2}$ for $2 \leq r \leq n$.

Remark 1.3.2 : If π^D is the dual of π, then π is the dual of π^D, that is, $(\pi^D)^D = \pi$.

Notation 1.3.3 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an arbitrary permutation of the jobs in N and $Q \subseteq N$. The restriction of π to Q is a permutation of jobs in Q, denoted by π_Q and is obtained from π by dropping the jobs not in Q.

Definition 1.3.4 : A sequence $\alpha = (\alpha_1, \ldots, \alpha_q)$ of a subset of jobs in N is called a *partial* (*complete*) sequence if $q < (=)\ n$. Without ambiguity in usage, we shall denote the set of jobs in a partial sequence α by α itself.

Definition 1.3.5 : A partial or complete sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_q)$ is in *SPT* (*LPT*) order if $p_{\pi_1} \leq \cdots \leq p_{\pi_q}$ ($p_{\pi_1} \geq \cdots \geq p_{\pi_q}$).

Definition 1.3.6 : A partial or complete sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_q)$ is in *SEPT* (*LEPT*) order if $\mu_{\pi_1} \leq \cdots \leq \mu_{\pi_q}$ ($\mu_{\pi_1} \geq \cdots \geq \mu_{\pi_q}$).

Definition 1.3.7 : A partial or complete sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_q)$ is said to be *V-shaped* if there exists an index r ($1 \leq r \leq q$) such that $p_{\pi_1} \geq \cdots \geq p_{\pi_r} \leq \cdots \leq p_{\pi_q}$.

Definition 1.3.8 : A partial or complete sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_q)$ is said to be *V-shaped in mean* (*variance*) if there exists an index r, $1 \leq r \leq q$ such that $\mu_{\pi_1} \geq \cdots \geq \mu_{\pi_r} \leq \cdots \leq \mu_{\pi_q}$ ($\sigma^2_{\pi_1} \geq \cdots \geq \sigma^2_{\pi_r} \leq \cdots \leq \sigma^2_{\pi_q}$).

Definition 1.3.9 : A complete sequence π is referred to as *L-G-S in mean* if there exist three non-empty partial sequences α, γ and β satisfying (a) $\pi = (\alpha, \beta, \gamma)$, and (b) α (γ) is in LEPT (SEPT) order and β is a general partial permutation.

Definition 1.3.10 : Given two complete sequences π and π', we say that π *dominates* π' (denoted by $\pi \preceq \pi'$) if $V(\pi) \leq V(\pi')$ in deterministic case, and $E[V(\pi)] \leq E[V(\pi')]$ in stochastic case.

Definition 1.3.11 : Given a partial sequence α, we say that $(\beta_1, \alpha, \beta_2)$ is a *completion* of α if (β_1, β_2) is a permutation of the jobs in $\bar{\alpha} = N \setminus \alpha$.

If $(\beta_1, \alpha, \beta_2)$ is a completion of α, it is possible that either of β_1 and β_2 may be empty but not both.

Definition 1.3.12 : Let α and α' be two partial sequences of the jobs in $Q \subseteq N$. The completions $(\beta_1, \alpha, \beta_2)$ and $(\beta_1, \alpha', \beta_2)$ are called *identical completions* of α and α' respectively.

Definition 1.3.13 : We call $(\beta_1, \ldots, \beta_2)$ a *two-sided partial sequence* if β_1 and β_2 are partial sequences such that $\beta_1 \cup \beta_2 \subset N$ and $\beta_1 \cap \beta_2 = \phi$.

In a two-sided partial sequence $(\beta_1, \ldots, \beta_2)$, jobs in the first $|\beta_1|$ positions and the last $|\beta_2|$ positions are fixed, but jobs in the middle $(n - |\beta_1| - |\beta_2|)$ are yet to be decided.

Definition 1.3.14 : Given a two-sided partial sequence $(\beta_1, \ldots, \beta_2)$, the sequence $(\beta_1, \alpha, \beta_2)$ is said to be its *completion* if α is a permutation of jobs in $N \setminus (\beta_1 \cup \beta_2)$.

Definition 1.3.15 : Let $(\beta_1, \ldots, \beta_2)$ and $(\beta'_1, \ldots, \beta'_2)$ be the two-sided partial sequences such that $\beta_1 \cup \beta_2 = \beta'_1 \cup \beta'_2$. The complete sequences $(\beta_1, \alpha, \beta_2)$ and $(\beta'_1, \alpha, \beta'_2)$ are called *identical completions* of $(\beta_1, \ldots, \beta_2)$ and $(\beta'_1, \ldots, \beta'_2)$ respectively.

In the context of computational complexity of algorithms, we use the terminologies – NP-hard problem, polynomial time algorithm, pseudopolynomial algorithm, ϵ-approximate algorithm, fully polynomial-time approximation scheme etc. For rigorous definitions of these terms and concept, see Aho, Hopcroft and Ullman [1974], Garey and Johnson [1979], Horowitz and Sahni [1978], Lenstra, Rinnooy Kan and Van Emde Bas [1978], Papadimitriou and Steiglitz [1982], Rinnooy Kan [1976] and Ullman [1976].

1.4 ORGANIZATION OF THE THESIS

We give below a brief account of the work presented in this thesis.

In Chapter 2, we primarily address the exact procedures for solving the CTV problem with deterministic job processing times. Section 2.2 gives a detailed account of the mathematical results available in the literature on the deterministic CTV problem. Since the problem is shown to be NP-hard by Kubiak [1993], the recent approach has been towards the development of pseudopolynomial algorithm for the same. Sections 2.3 to 2.8 are mainly devoted to the pseudopolynomial algorithms. We derive here two dominance rules which are in turn used in the development of a new pseudopolynomial algorithm (**MP1**). The algorithm **MP1** involves binary branching making use of the V-shaped property (due to Eilon and Chowdhury [1977]). A node, in this algorithm, represents a two-sided partial sequence of the form $(\alpha, \ldots, \beta)$ containing the larger jobs. Based on extensive numerical investigation on the performances of **MP1** and **DGW** algorithm (of De, Ghosh and Wells [1992]), it is observed that (a) the performance of **DGW** is excellent when the processing times are homogeneous, and (b) **MP1** is very good for heterogeneous processing times.

By taking advantage of the contrasting merits of **DGW** and **MP1**, we then develop **MP2** algorithm (which is once again pseudopolynomial). The performance of **DGW** and **MP2** alongwith the algorithm of Kubiak [1995] is numerically investigated, and the results are reported in Section 2.8.

Next, we turn to the derivation of lower bound on the CTV which is used later in the development of another algorithm (**MP3**). De, Ghosh and Wells [1990, 1992] are the first to deal with it. Manna and Prasad [1994] and Mittenthal, Raghavachari and Rana [1994] observe the infeasibility of the solution given by De, Ghosh and Wells [1992]. We derive here the correct version of the solution, but note that the procedure of Mittenthal, Raghavachari and Rana [1994] is superior to it. We also study the performance of the lower bound given by Mittenthal, Raghavachari and Rana [1994] by numerical investigation.

Finally, we present the algorithm (MP3) which involves a dominance condition and a lower bounding procedure. The algorithm is an implicit enumeration method based on branch-and-bound approach. A node in this algorithm is again a two-sided partial sequence, similar to that of MP1. The lower bound on CTV at any node $(\alpha, \ldots, \beta)$ involves as a component the lower bound on CTV for a subproblem consisting of the jobs in $N \setminus (\alpha \cup \beta)$. The lower bound for the subproblem is computed using the procedure of Mittenthal, Raghavachari and Rana [1994]. The real advantage of MP3 is shown for non-integer processing times.

Chapter 3 deals with the CTV problem with random (stochastic) job processing times. In Section 3.2, we present the preliminary results which are used later in this chapter to derive the properties of optimal sequences and in the development of an algorithm. Like the deterministic version, this problem is also NP-hard.

In view of the difficulty of finding an optimal sequence, research has been directed towards the nature of optimal sequences. To be specific, it is of great interest to the researchers to know whether there exits a V-shaped optimal sequence (see Chakravarthy [1986] and Vani and Raghavachari [1987]). Existence of an optimal sequence in the set of all V-shaped sequences enables us to confine the search to 2^{n-1} sequences only.

Section 3.3 is devoted to the study on the properties of optimal sequences. In Subsection 3.3.1, we derive two sufficient conditions on the existence of V-shaped optimal sequence for the general stochastic CTV problem. Subsection 3.3.2 deals with the special case considered by Vani and Raghavachari [1987] and provides a simple but stronger sufficient criterion for V-shaped optimality. In Subsection 3.3.3, we introduce a special case in which the random processing times satisfy the following *order* property :

$$\mu_r < \mu_s \;\Rightarrow\; \sigma_r^2 < \sigma_s^2 \quad \text{for any } r \text{ and } s.$$

In this case, we first demonstrate that V-shaped property is not a necessity for optimality. Then we derive a sufficient condition for the same and obtain several results on the monotonic property of optimal sequence. We also prove the existence of an L-G-S (in mean) optimal sequence for this problem.

In Section 3.4, we present a procedure to derive a lower bound for the expected CTV and a dominance rule which are in turn effectively used to develop a branch-and-bound algorithm to solve the stochastic CTV problem with general processing times. Further, we discuss here the required modifications in the algorithm for ordered processing times.

The main contents of Chapter 4 are the heuristic procedures for the deterministic and stochastic CTV problems. Since both the problems are NP-hard, it is very important to develop heuristic procedures to derive near optimal solutions. Section 4.2 contains some preliminary results which are used in the development of the heuristic procedure (of Manna and Prasad [1993]) for the deterministic CTV problem. The results of this section are mainly concerned with the lower and upper bound on the position of the smallest job in an optimal sequence.

Section 4.3 deals with the heuristic procedures for the deterministic CTV problem. In Subsection 4.3.1, we present the following procedures : (H1) Eilon and Chowdhury [1977], (H2) Kanet [1981], (H3) Manna and Prasad [1993], (H4) Vani and Raghavachari [1987] and (H5) Gupta, Gupta and Bector [1990].

We note that H1, H2 and H3 are basic heuristics constructing near optimal sequences, whereas H4 and H5 are heuristics which improve upon a given sequence. In Subsection 4.3.2, we study the performances of the above heuristics by numerical investigation. Initially, we compare H1, H2 and H3. Later, we evaluate the effectiveness of H4 and H5 on the solutions generated by H1, H2 and H3. Here, the percentage of relative error, E_h, is taken as the performance index of heuristic h. The salient features arising out of the numerical investigation are as follows :

(a) Among the basic heuristics, H3 is the best. For all the problem instances, the percentage relative error $E_{H3} \leq 0.0085\%$ and the overall mean of E_{H3} is only 0.0003%. The performance of H3 improves as the problem size increases.

(b) H5 is more effective than H4.

(c) The combination (H3, H5) is found to be the best with the maximum and mean value of the performance index being 0.0012% and 0.0001% respectively.

Finally, in Section 4.4, we present a heuristic procedure for the stochastic CTV problem which makes use of H3 and H4, and we report on its performance.

In Chapter 5, we pose two conjectures based on the patterns observed while studying the deterministic CTV problem. We also provide here some results derived in an attempt to prove them.

The first conjecture is on the functional behavior of the CTV where the CTV function is defined as the minimum CTV value for a given position of the smallest job.

Let S_r $(1 \leq r \leq n)$ be the set of all V-shaped sequences in which the smallest job occurs at rth position. Obviously, $S = S_1 \cup S_2 \cup \cdots \cup S_n$ is the set of all V-shaped sequences. Also, let $T_r^* = \min_{\pi \in S_r} V(\pi)$ for $1 \leq r \leq n$. The first conjecture is : "$T_{k+1}^* \not> \min \left\{T_k^*, T_{k+2}^*\right\}$ for any $1 \leq k \leq m-2$ where $m = \lfloor \frac{n}{2} \rfloor + 1$". If this conjecture is true, it can be seen that $T_1^*, T_2^*, \cdots, T_n^*$ are either V-shaped or W-shaped.

We derive here some results involving C_n and $\bar{C}$ which throw some light on the above conjecture.

The second conjecture is on optimal sequence for the CTV problem with the processing times in arithmetic progression. It is as follows : "Let $p_j = a + (n - j + 1)b$ where $a \geq 0$ and $b > 0$. Only optimal sequences for this problem are $(1, 2, 5, 6, \ldots, 8, 7, 4, 3)$ and its dual $(1, 3, 4, 7, 8, \ldots, 6, 5, 2)$."

CHAPTER 2

EXACT ALGORITHMS FOR CTV MINIMIZATION

2.1 INTRODUCTION

The problem of minimizing job completion time variance (CTV) on a single machine arises, especially, when large computer data files are to be organized in on-line systems for providing uniform response times to the users. It was first considered by Merten and Muller [1972]. Since then, it has been drawing the attention of several researchers. Later, Eilon and Chowdhury [1977] derived some important results concerning the nature of optimal schedules.

Bagchi, Sullivan and Chang [1987] have pointed out the importance of the CTV problem in the context of current emphasis on Just-in-time production philosophy. Bagchi, Sullivan and Chang [1987], Raghavachari [1988], De, Ghosh and Wells [1989] and Weng and Ventura [1994] have found relationship between the CTV problem and the MSD problem (discussed in Section 1.2 of Chapter 1).

From mathematical point of view, the problem is combinatorial in nature. Recently, Kubiak [1993] has shown that the CTV problem is, in fact, NP-hard (also see Cheng and Cai [1993]). So, it is very unlikely that an efficient (polynomial time) algorithm can be developed for the CTV problem. Thus, the recent approach to obtain an exact optimal solution is towards the development of pseudopolynomial algorithms.

Since the CTV problem is equivalent to the unconstrained version of the MSD problem, we may take sufficiently large due date and adopt an MSD procedure (for example, Bagchi, Sullivan and Chang [1987], De, Ghosh and Wells [1990], Federgruen and Mosheiov [1993] etc.) to generate a solution for the CTV problem. Alternatively, the CTV problem can also be viewed as a tightly constrained MSD problem with unknown due date. By enumerating over all possible values of the due date (see De, Ghosh and Wells [1992]) and solving the resulting instances of the MSD problem through an adaptation of the algorithm given by

Hall, Kubiak and Sethi [1991], we can once again generate a solution for the CTV problem.

However, some specialized procedures are available for the CTV problem in Bagchi, Sullivan and Chang [1987], De, Ghosh and Wells [1990,1992], Manna and Prasad [1994, 1995], and Kubiak [1995]. All these algorithms are primarily based on the V-shaped property of optimal sequence.

In this chapter, we primarily address the exact procedures for solving the CTV problem with deterministic job processing times. Section 2.2 contains the mathematical results available in the literature on the deterministic CTV problem. In Sections 2.3 through 2.8, we present the pseudopolynomial algorithms of De, Ghosh and Wells [1992], Manna and Prasad [1995] and Kubiak [1995], and study their merits with the help of extensive numerical investigation. We describe, in Section 2.9, the implicit enumeration method of Manna and Prasad [1994] which is based on branch-and-bound approach, and is of particular importance for non-integer processing times. In this context, we study on the derivation of lower bound for CTV in Subsection 2.9.1.

2.2 PRELIMINARY RESULTS

In this section, we present some important mathematical results that are available in the literature on the deterministic CTV problem.

Lemma 2.2.1 (Schrage [1975]): For any sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, we have

$$\begin{aligned} n^2V(\pi) &= \sum_{r=1}^{n}(r-1)(n-r+1)p_{\pi_r}^2 \\ &\quad +2\sum_{r=1}^{n-1}(r-1)p_{\pi_r}\sum_{s=r+1}^{n}(n-s+1)p_{\pi_s} \end{aligned} \tag{2.2.1}$$

It can be seen from the equation (2.2.1) that the expression $V(\pi)$ is independent of the first job in π.

The following result decides the position of the largest job (in terms of processing time) in an optimal sequence.

Theorem 2.2.2 (Schrage [1975]): Any optimal sequence is of the form $(1, \ldots)$.

This result follows directly from Lemma 2.2.1 by observing that given any sequence π,

i) each p_i has non-negative contribution to $V(\pi)$,

ii) $V(\pi)$ is independent of the first job in π.

■

Remark 2.2.3 : *In the case of multiple largest jobs, that is,* $p_1 = p_2 = \cdots = p_r$ for $r \geq 2$, the jobs, one may relabel the jobs, without loss of generality, in order to note that Theorem 2.2.2 holds good.

Merten and Muller [1972] have proved the existence of at least two optimal sequences by the following result.

Theorem 2.2.4 (Merten and Muller [1972]): Let π be any sequence and π^D be its dual. Then $V(\pi) = V(\pi^D)$.

Proof : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ and $\pi^D = (\pi_1^D, \pi_2^D, \ldots, \pi_n^D)$. Since π^D is the dual of π, we have $C_{[r]}(\pi^D) = C_{[n]}(\pi) + C_{[1]}(\pi) - C_{[n-r+1]}(\pi)$ for $1 \leq r \leq n$. Therefore,

i) $\bar{C}(\pi^D) = C_{[n]}(\pi) + C_{[1]}(\pi) - \bar{C}(\pi)$,

ii) $C_{[r]}(\pi^D) - \bar{C}(\pi^D) = \bar{C}(\pi) - C_{[n-r+1]}(\pi)$ for $1 \leq r \leq n$

and,

$$\begin{aligned} V(\pi^D) &= \frac{1}{n}\sum_{r=1}^{n}\left[C_{[r]}(\pi^D) - \bar{C}(\pi^D)\right]^2 \\ &= \frac{1}{n}\sum_{r=1}^{n}\left[C_{[r]}(\pi) - \bar{C}(\pi)\right]^2 \\ &= V(\pi). \end{aligned}$$

Hence the theorem holds. ■

Theorem 2.2.5 (Eilon and Chowdhury [1977]) : Any optimal sequence is V-shaped.

Remark 2.2.6 : According to the above result, V-shapedness is a necessary condition for optimality of a sequence. Thus, it is enough to confine the search for optimal sequence to the set of V-shaped sequences only. It may be noted that the total number of V-shaped sequences is 2^{n-1}.

Theorem 2.2.7 : There exists an optimal sequence of the form $(1, \ldots, 2)$.

Proof : The result follows from the Theorems 2.2.2, 2.2.5 and 2.2.4. ■

Schrage [1975] conjectured that there exists an optimal sequence of the form $(1, 3, 4, \ldots, 2)$. This was later shown by Kanet [1981] to be partially incorrect. Specifically, Kanet [1981] gave numerical example to demonstrate that the fourth largest job need not be at third position in the best sequence among the sequences of the form $(1, 3, \ldots, 2)$. However, the question on the existence of V-shaped optimal sequence of the form $(1, 3, \ldots, 2)$ remained open for a long time. Later, Vani and Raghavachari [1987] showed that such an optimal sequence does exist for problems with 18 or fewer jobs. Finally, the issue is completely settled by Hall and Kubiak [1993].

Theorem 2.2.8 (Hall and Kubiak [1993]) **:** There exists a V-shaped optimal sequence of the form (1, 3, ..., 2).

Remark 2.2.9 : Consequent to Theorem 2.2.8, one may restrict search for optimal sequence to the set of V-shaped sequences of the form (1, 3, ..., 2).

2.3 THE ALGORITHM OF DE, GHOSH AND WELLS [1992]

The algorithm builds on V-shaped partial sequences by successively scheduling the larger jobs one at a time in n stages. In the first stage, the partial sequence contains the only job n and at the final stage, the algorithm generates the complete sequences. From a partial sequence α of stage k, two partial (or complete if $k = n - 1$) sequences, namely, $(\alpha, \overline{n-k})$ and $(\overline{n-k}, \alpha)$ are generated. The maximum number of partial (or complete) sequences at stage k is 2^{k-1}. But, after generation of all the partial sequences, at any stage, the algorithm compares them to fathom, if possible some of these partial sequences, using a dominance criterion and an elimination rule. Only the active partial sequences are used for further generation new nodes at the immediate next stage.

Let α and β be two partial sequences at stage k. Let $\bar{C}(\alpha)$ and $V(\alpha)$ be the average and the variance of the job completion times for the partial sequence α. Similarly, $\bar{C}(\beta)$ and $V(\beta)$ are defined.

Lemma 2.3.1 : Let π and π' be the identical completions of α and β respectively, that is, $\pi = (\alpha, \gamma)$ and $\pi' = (\beta, \gamma)$ where γ is an arbitrary V-shaped partial sequence. Then

$$V(\pi) \leq V(\pi')$$

$$\text{if } \bar{C}(\alpha) = \bar{C}(\beta) \text{ and } V(\alpha) \leq V(\beta) \tag{2.3.2}$$

Proof : Refer to De, Ghosh and Wells [1992]. ■

Corollary 2.3.2 : Let G (G') be the set of all complete sequences obtained from the partial sequence α (β). If the condition (2.3.2) holds, all the sequences in G' are dominated by those in G with respect to CTV.

Thus, if the condition (2.3.2) holds, the node β is fathomed at stage k.

The lemma below describes the lower and upper bounds on the average of job completion times for any given partial sequence to check whether it is potential enough to yield an optimal sequence.

Lemma 2.3.3 : Let α be a V-shaped partial sequence of the jobs $n-k+1$, $n-k+2, \ldots, n$. Fathom α if any of the following holds :

$$(i)\ \bar{C}(\alpha) < \frac{1}{k}\left[\frac{n}{2}MS - \sum_{j=1}^{n-k}(n-j+1)p_j\right] \qquad (2.3.3)$$

$$(ii)\ \bar{C}(\alpha) > \frac{1}{k}\left[\frac{n}{2}(MS-p_1) - (n-k-1)\sum_{j=n-k+1}^{n} p_j - \sum_{j=2}^{n-k}(j-1)p_j\right] \qquad (2.3.4)$$

where $MS = \sum_{j=1}^{n} p_j$.

Proof : Refer to De, Ghosh and Wells [1992]. ∎

Now, we describe the algorithm of De, Ghosh and Wells [1992], denoted by DGW, using the dominance criterion given in Lemma 2.3.1 and the elimination rule given in Lemma 2.3.3.

Algorithm DGW :

Step 1 : Set $W_1 = \{(n)\}$, $W_2 = \phi$, $k = 2$.

Step 2 : Take any partial sequence $\alpha \in W_1$. Update $W_1 \leftarrow W_1 \setminus \{\alpha\}$ and $W_2 \leftarrow$

$W_2 \cup \left\{(\overline{n-k+1},\alpha),(\alpha,\overline{n-k+1})\right\}$. If $W_1 = \phi$, go to Step 3. Otherwise, repeat Step 2.

Step 3 : If $k = n$, go to Step 4. Else, for every partial sequence $\alpha \in W_2$, compute $\tilde{C}(\alpha)$ and $V(\alpha)$. If either of the conditions (2.3.3) and (2.3.4) holds for any partial sequence $\alpha \in W_2$, remove α from W_2. Using Lemma 2.3.1, delete all dominated partial sequences from W_2. Set $k \leftarrow k+1$, $W_1 \leftarrow W_2$ and go to Step 2.

Step 4 : Compute and compare the CTV values of all complete sequences in W_2. Let $V(\pi^*) = \min_{\pi \in W_2} V(\pi)$. Return π^* as optimal sequence with $V(\pi^*)$ as the corresponding optimal CTV value.

Remark 2.3.4 (Complexity) : When p_j's are integers, the complexity of this algorithm is $o(n^2 \sum_{j=1}^{n} p_j)$, because the number of distinct values of $k\tilde{C}(\alpha)$ (refer to Lemma 2.3.1) is bounded by $n \sum_{j=1}^{n} p_j$, at any stage k $(1 \leq k \leq n)$, in the algorithm.

Remark 2.3.5 : *It is evident from Remark 2.3.4 that the complexity of the* above algorithm depends on the particular problem instance. Let $p_j \in U$ for $j = 1, 2, \ldots, n$ where U is a discrete sample space. For any given U, but sufficiently large n, p_j's are quite homogeneous. In such situations, it can be easily observed that the dominance rule, given in Lemma 2.3.1, becomes highly effective, that is, the algorithm performs very efficiently.

2.4 A NEW PSEUDOPOLYNOMIAL ALGORITHM

We know from Remark 2.3.5 that when the job processing times are homogeneous, the performance of the algorithm of De, Ghosh and Wells [1992] is expected to be very good. However, we can guess that the dominance rule involved in that algorithm will not be much effective in fathoming the nodes for heterogeneous processing times.

In this section, we present a new pseudopolynomial algorithm that exploits primarily the heterogeneity present among the processing times.

The algorithm involves, at any stage, generation of two nodes from an existing node. A node is a two-sided partial sequence $(\alpha, \ldots, \beta)$ where α, $\beta \subseteq N$ such that (i) $\alpha \cap \beta = \phi$, (ii) $\alpha \cup \beta = \{1, 2, \ldots, r\}$ for some r, $1 \leq r < n$, and (iii) the sequence (α, β) is V-shaped. For example, $(1, 3, 6, \ldots, 5, 4, 2)$ is a node with $\alpha = \{1, 3, 6\}$, $\beta = \{5, 4, 2\}$. From a node $(\alpha, \ldots, \beta)$ with $|\alpha \cup \beta| = k - 1$, two nodes - $(\alpha, k, \ldots, \beta)$ and $(\alpha, \ldots, k, \beta)$ are generated.

Consider a node $(\alpha, \ldots, \beta)$ with $|\alpha \cup \beta| = k - 1$ and let $\pi_{(k)} = (\alpha, k, \beta)$. Note that $\pi_{(k)}$ is a V-shaped partial (complete) sequence of the jobs $1, 2, \ldots, k$ if $k < n$ $(k = n)$.

Let G_1 be the set of all complete sequences generated from the node $(\alpha, k, \ldots, \beta)$ and G_2 the same for the node $(\alpha, \ldots, k, \beta)$. It means that any sequence in G_1 (G_2) is of the form $(\alpha, k, \pi_L, \beta)$ $((\alpha, \pi_L, k, \beta))$ where π_L is a V-shaped partial sequence of the jobs $k + 1, k + 2, \ldots, n$.

Let $C_j(\pi_{(k)})$ and $\bar{C}(\pi_{(k)})$ denote the completion time of job j and average completion time respectively for the partial sequence $\pi_{(k)}$. Also, let

$M_L = \sum_{j \in \pi_L} p_j$ (= makespan of π_L)

$\delta = \frac{1}{n} [(n - k)p_k - M_L]$

$b = |\beta|$, and

$$Y = 2k\delta \left[\bar{C}(\pi_{(k)}) - C_k(\pi_{(k)})\right] + 2bM_L\delta + (k-1)\delta^2 + (M_L + \delta)^2 + (n-k)(p_k - \delta)^2.$$

We now derive two important results that are used to develop the algorithm.

Lemma 2.4.1 : Let π_L be an arbitrary V-shaped partial sequence of jobs $k+1$, $k + 2$, $\ldots, n$. Let $\pi = (\alpha, k, \pi_L, \beta)$ and $\pi' = (\alpha, \pi_L, k, \beta)$ be two complete sequences generated from the nodes $(\alpha, k, \ldots, \beta)$ and $(\alpha, \ldots, k, \beta)$ respectively.

Then

$$(i)\quad V(\pi) \geq V(\pi') \quad \text{if} \quad D_{min}(\pi_L) \geq \frac{nY}{2(M_L + kp_k)}, \qquad (2.4.5)$$

$$(ii)\quad V(\pi) \leq V(\pi') \quad \text{if} \quad D_{max}(\pi_L) \leq \frac{nY}{2(M_L + kp_k)} \qquad (2.4.6)$$

where $D_{min}(\pi_L)$ is the minimum total flowtime of jobs in π_L which is given by SPT sequence of the jobs in π_L, and $D_{max}(\pi_L)$ is the maximum total flowtime of jobs in π_L given by LPT sequence of the jobs in π_L.

Proof : Note that

$$C_j(\pi') = \begin{cases} C_j(\pi) & \text{for } j \in \alpha \cup \beta \\ C_j(\pi) + M_L & \text{for } j \in \{k\} \\ C_j(\pi) - p_k & \text{for } j \in \pi_L \end{cases}$$

and therefore $\bar{C}(\pi') = \bar{C}(\pi) - \delta$. We have

$$\begin{aligned} nV(\pi') &= \sum_{j\in\alpha\cup\beta} \left[C_j(\pi') - \bar{C}(\pi')\right]^2 + \left[C_k(\pi') - \bar{C}(\pi')\right]^2 \\ &\qquad + \sum_{j\in\pi_L} \left[C_j(\pi') - \bar{C}(\pi')\right]^2 \\ &= \sum_{j\in\alpha\cup\beta} \left[C_j(\pi) - \bar{C}(\pi) + \delta\right]^2 + \left[C_k(\pi) - \bar{C}(\pi) + (M_L + \delta)\right]^2 \\ &\quad + \sum_{j\in\pi_L} \left[C_j(\pi) - \bar{C}(\pi) - (p_k - \delta)\right]^2 . \end{aligned}$$

Let $X = (k-1)\delta^2 + (M_L + \delta)^2 + (n-k)(p_k - \delta)^2$. Rearranging the terms in the above equation, we can write

$$\begin{aligned} &n[V(\pi') - V(\pi)] \\ &\quad = 2\delta \sum_{j\in\alpha\cup\beta} \left[C_j(\pi) - \bar{C}(\pi)\right] + 2(M_L + \delta)\left[C_k(\pi) - \bar{C}(\pi)\right] \\ &\qquad\qquad - 2(p_k - \delta) \sum_{j\in\pi_L} \left[C_j(\pi) - \bar{C}(\pi)\right] + X \end{aligned}$$

$$= 2p_k \sum_{j\in\alpha\cup\{k\}\cup\beta} \left[C_j(\pi) - \bar{C}(\pi)\right] + 2M_L\left[C_k(\pi) - \bar{C}(\pi)\right] + X$$

$$\text{(since } \sum_{j\in\pi_L}\left[C_j(\pi) - \bar{C}(\pi)\right] = -\sum_{j\in\alpha\cup\{k\}\cup\beta}[C_j(\pi) - \bar{C}(\pi)])$$

$$= 2p_k \sum_{j\in\alpha\cup\{k\}\cup\beta} C_j(\pi) - 2(M_L + kp_k)\bar{C}(\pi) + 2M_LC_k(\pi) + X.$$

We also have $C_j(\pi) = \begin{cases} C_j(\pi_{(k)}) & \text{for } j \in \alpha\cup\{k\} \\ C_j(\pi_{(k)}) + M_L & \text{for } j \in \beta \end{cases}$

which imply that $n\bar{C}(\pi) = k\bar{C}(\pi_{(k)}) + bM_L + \sum_{j\in\pi_L} C_j(\pi)$.

Now we can write

$$\begin{aligned}
&n[V(\pi') - V(\pi)] \\
&= 2p_k\left[\sum_{j\in\alpha\cup\{k\}} C_j(\pi_{(k)}) + \sum_{j\in\beta} C_j(\pi_{(k)}) + bM_L\right] \\
&\quad -2(M_L + kp_k)\bar{C}(\pi) + 2M_LC_k(\pi_{(k)}) + X \\
&= 2kp_k\bar{C}(\pi_{(k)}) + 2bM_Lp_k - 2(M_L + kp_k)\frac{1}{n}\left[k\bar{C}(\pi_{(k)}) + bM_L\right. \\
&\quad \left.+ \sum_{j\in\pi_L} C_j(\pi)\right] + 2M_LC_k(\pi_{(k)}) + X \\
&= \left[2k\delta\bar{C}(\pi_{(k)}) + 2bM_L\delta + 2M_LC_k(\pi_{(k)}) + X\right] - \frac{2(M_L + kp_k)}{n}\sum_{j\in\pi_L} C_j(\pi) \\
&= Z - \frac{2(M_L + kp_k)}{n}\sum_{j\in\pi_L} C_j(\pi)
\end{aligned}$$

where $Z = 2k\delta\bar{C}(\pi_{(k)}) + 2bM_L\delta + 2M_LC_k(\pi_{(k)}) + X$.

Therefore,

$$V(\pi) \geq V(\pi') \quad \text{if} \quad \sum_{j\in\pi_L} C_j(\pi) \geq \frac{nZ}{2(M_L + kp_k)}$$

holds for any arbitrary V-shaped partial sequence π_L containing the jobs $k+1$, $k+2, \ldots, n$.

Since $\sum_{j\in\pi_L} C_j(\pi) \geq (n-k)C_k(\pi_{(k)}) + D_{min}(\pi_L)$, it now follows that

$$V(\pi) \geq V(\pi') \quad \text{if} \quad D_{min}(\pi_L) \geq \frac{nZ}{2(M_L + kp_k)} - (n-k)C_k(\pi_{(k)})$$
$$i.e., V(\pi) \geq V(\pi') \quad \text{if} \quad D_{min}(\pi_L) \geq \frac{nY}{2(M_L + kp_k)}.$$

Using similar arguments, we can easily see that

$$V(\pi') \geq V(\pi) \quad \text{if} \quad D_{max}(\pi_L) \leq \frac{nY}{2(M_L + kp_k)}.$$

■

Corollary 2.4.2 : If the condition (2.4.5) ((2.4.6)) holds, all the sequences in G_1 (G_2) are dominated by those in G_2 (G_1) with respect to CTV.

Thus, if the condition (2.4.5) ((2.4.6)) holds, then the node $(\alpha, k, \ldots, \beta)$ $((\alpha, \ldots, k, \beta))$ is fathomed.

Remark 2.4.3 : While generating two new nodes from an existing node, the conditions (2.4.5) and (2.4.6) help us to fathom, if possible, one of the new nodes.

Lemma 2.4.4 : Let $(\alpha, \ldots, \beta)$ and $(\alpha', \ldots, \beta')$ be two nodes such that $\alpha \cup \beta$ $= \alpha' \cup \beta' = \{1, 2, \ldots, k\} = E$ (say). Further, let $\pi = (\alpha, \pi_L, \beta)$ and $\pi' =$ (α', π_L, β') be two complete sequences where π_L is an arbitrary V-shaped partial sequence of the jobs $k+1, k+2, \ldots, n$. Then

$$V(\pi) \leq V(\pi')$$

$$\text{if} \quad \left. \begin{aligned} \bar{C}_E(\pi) - \textstyle\sum_{j\in\alpha} p_j &= \bar{C}_E(\pi') - \textstyle\sum_{j\in\alpha'} p_j \\ \text{and} \quad h(\pi) &\leq h(\pi') \end{aligned} \right\} \qquad (2.4.7)$$

where $\bar{C}_E(\pi) = \frac{1}{k}\sum_{j\in E} C_j(\pi)$ and $h(\pi) = \sum_{j\in E}\left[C_j(\pi) - \bar{C}_E(\pi)\right]^2$. Similarly, $\bar{C}_E(\pi')$ and $h(\pi')$ are also defined.

Proof : We know that

$$
\begin{aligned}
nV(\pi) &= \sum_{j=1}^{n}\left[C_j(\pi) - \bar{C}(\pi)\right]^2 \\
&= \sum_{j\in E}\left[C_j(\pi) - \bar{C}_E(\pi)\right]^2 + \sum_{j\in L}\left[C_j(\pi) - \bar{C}_L(\pi)\right]^2 \\
&\qquad + \frac{k(n-k)}{n}\left[\bar{C}_E(\pi) - \bar{C}_L(\pi)\right]^2 \qquad (2.4.8)
\end{aligned}
$$

(using variance partition formula)

where $L = N \setminus E = \{k+1,\ldots,n\}$, $\bar{C}_L(\pi) = \frac{1}{n-k}\sum_{j\in L} C_j(\pi)$ and $\bar{C}(\pi) = \frac{1}{n}\left[k\bar{C}_E(\pi) + (n-k)\bar{C}_L(\pi)\right]$.

Let $\bar{C}(\pi_L)$ ($V(\pi_L)$) denote the average (variance) of the job completion times for subproblem with jobset as L with the sequence π_L. It may be noted that $C_j(\pi) = \sum_{j\in\alpha} p_j + C_j(\pi_L)$ for all $j \in L$ and hence $\bar{C}_L(\pi) = \sum_{j\in\alpha} p_j + \bar{C}(\pi_L)$. Therefore, we have, from the equation (2.4.8),

$$
\begin{aligned}
nV(\pi) &= h(\pi) + (n-k)V(\pi_L) \\
&\qquad + \frac{k(n-k)}{n}\left[\bar{C}_E(\pi) - \sum_{j\in\alpha} p_j - \bar{C}(\pi_L)\right]^2. \qquad (2.4.9)
\end{aligned}
$$

Similarly, we can get

$$
\begin{aligned}
nV(\pi') &= h(\pi') + (n-k)V(\pi_L) \\
&\qquad + \frac{k(n-k)}{n}\left[\bar{C}_E(\pi') - \sum_{j\in\alpha'} p_j - \bar{C}(\pi_L)\right]^2. \qquad (2.4.10)
\end{aligned}
$$

Now, the lemma follows from the equations (2.4.9) and (2.4.10). ■

Corollary 2.4.5 : Let H_1 (H_2) be the set of all complete sequences generated from the node $(\alpha,\ldots,\beta)$ $((\alpha',\ldots,\beta'))$ where $(\alpha,\ldots,\beta)$ $((\alpha',\ldots,\beta'))$ is as defined in Lemma 2.4.4. If the condition (2.4.7) holds, all the sequences in H_2 are dominated by those in H_1 with respect to CTV.

Remark 2.4.6 : The dominance condition (2.4.7) is very similar to the dominance condition (2.3.2), and due to this condition, the complexity of the following algorithm becomes pseudopolynomial.

We now present a pseudopolynomial algorithm, denoted by **MP1**, which involves the dominance rules presented in Lemmas 2.4.1 and 2.4.4. Due to Theorem 2.2.8, the algorithm restricts the search to the set of V-shaped sequences of the form $(1,3,\ldots,2)$ and therefore it starts with the node $(1,3,\ldots,2)$.

Algorithm **MP1** :

Step 1 : Initialize $W_1 = \{(1,3,\ldots,2)\}$, $W_2 = \phi$, $k = 3$.

Step 2 : Set $k \leftarrow k+1$. If $k = n-1$, go to Step 6. Else, go to Step 3.

Step 3 : If $W_1 = \phi$, go to Step 5. Otherwise, take a node $(\alpha,\ldots,\beta) \in W_1$ and update $W_1 \leftarrow W_1 \setminus \{(\alpha,\ldots,\beta)\}$. Generate two new nodes $(\alpha,k,\ldots,\beta)$ and $(\alpha,\ldots,k,\beta)$.

Step 4 : If the condition (2.4.5) holds, update $W_2 \leftarrow W_2 \cup \{(\alpha,\ldots,k,\beta)\}$. If condition (2.4.6) holds, update $W_2 \leftarrow W_2 \cup \{(\alpha,k,\ldots,\beta)\}$. If neither of theses conditions holds, update $W_2 \leftarrow W_2 \cup \{(\alpha,k,\ldots,\beta),(\alpha,\ldots,k,\beta)\}$. Go to Step 3.

Step 5 : Invoke the dominance rule given in Lemma 2.4.4 and delete all the dominated nodes from W_2. Set $W_1 \leftarrow W_2$ and go to Step 2.

Step 6 : From every node $(\alpha,\ldots,\beta) \in W_1$, obtain a complete sequence $\pi = (\alpha,n,\beta)$ and compute its CTV value, $V(\pi)$. Let $V(\pi^*) = \min_\pi V(\pi)$. Return π^* as optimal sequence with $V(\pi^*)$ as the corresponding optimal CTV value.

The algorithm starts with $k = 3$ and the initial node $(1,3,\ldots,2)$. At any stage k $(3 \le k \le n-1)$, we have a set of nodes of the form $(\alpha,\ldots,\beta)$ with $\alpha \cup \beta = \{1,2,\ldots,k\}$. From each existing node $(\alpha,\ldots,\beta)$ with $|\alpha \cup \beta| = k < n-1$, the algorithm generates two nodes — $(\alpha, \overline{k+1}, \ldots, \beta)$ and $(\alpha, \ldots, \overline{k+1}, \beta)$. Conditions (2.4.5) and (2.4.6) are used to fathom (if possible) one of these two new nodes. This process is repeated for all nodes with $|\alpha \cup \beta| = k$. The condition (2.4.7) is then applied to fathom some of these newly generated nodes. From every node $(\alpha, \ldots, \beta)$ at stage $n-1$, a complete sequence (α, n, β) is obtained. Finally, the CTV values of all such complete sequences are compared to find an optimal solution.

Remark 2.4.7 (Complexity) : When p_j's are integers, the complexity of this algorithm is $o\left(n^2\{(p_2 - p_3) + \sum_{j=4}^{n}(j-2)p_j\}\right)$, because $k\left[\bar{C}_E(\pi) - \sum_{j \in \alpha} p_j\right]$ (refer to Lemma 2.4.4) takes, at any stage k, the integral values in the interval $\left[(p_2 - p_3) - \sum_{j=4}^{n}(j-2)p_j,\ (p_2 - p_3) + \sum_{j=4}^{n}(j-2)p_j\right]$.

2.5 COMPARISON OF DGW AND MP1

The algorithm **MP1** takes good advantage of the heterogeneity present among the job processing times (p_j's) using the dominance condition given in Lemma 2.4.1. In fact, the performance of this algorithm is expected to be highly satisfactory if the p_j's are too heterogeneous. To illustrate it, we consider the following six-job problem.

Example 2.5.1 :

j	1	2	3	4	5	6
p_j	1000	996	4	3	2	1

While solving this problem using **MP1**, it generates only four nodes — $(1,3,\ldots,2)$, $(1,3,4,\ldots,2)$, $(1,3,5,\ldots,2)$ and $(1,3,4,5,6,2)$, one at each stage.

In order to compare the performances of DGW and **MP1**, we have carried out extensive numerical investigation on both of them for a large number of problems with various sizes (n) and several sample spaces (the set of all possible values of p_j's).

In this investigation, the (discrete) sample spaces considered are $U_r = \{1, 2, \ldots, r\}$ for r = 10, 50, 100, 200, 300 and 500, and the problem size ranges from 20 to 100. Given a combination of n and U_r, generation of a problem instance involves drawing n random values from U_r as the job processing times. For each possible combination of n and U_r, ten problem instances are generated. Both the algorithms (DGW and MP1) are applied on each of these problem instances and the execution times are recorded.

This computational work has been carried out on the system DEC300/AXP 700 (in UNIX Operating System) and the computer programs are coded in FORTRAN. The summarized computational results for both the algorithms are presented in Tables 2.5.1(a) and 2.5.1(b).

Table 2.5.1(a) : Average Execution Time (in seconds) for DGW and MP1

	U_{10}		U_{50}		U_{100}	
n	DGW	MP1	DGW	MP1	DGW	MP1
20	0.02	*	0.70	0.04	2.13	0.11
30	0.25	0.07	7.34	2.03	27.77	9.19
40	0.96	0.44	34.12	24.26	133.70	81.27
50	3.34	1.94	109.37	84.23	434.32	373.14
60	9.12	6.61	281.57	250.23		
70	19.83	20.44	608.43	742.48		
80	37.99	34.95				
90	75.09	84.38				
100	119.66	145.39				

Note : * indicates negligible.

Table 2.5.1(b) : Average Execution Time (in seconds) for DGW and MP1

	U_{200}		U_{300}		U_{500}	
n	DGW	MP1	DGW	MP1	DGW	MP1
20	9.11	0.34	15.90	0.44	39.14	0.44
30	120.38	24.78	260.45	67.06	634.49	122.91
40	510.72	389.48	1044.46	577.94	3673.24	1730.36

For each combination of n and U_r, the above table contains the average execution time (of ten randomly generated problem instances) taken by the algorithms DGW and MP1.

Remark 2.5.1 : On comparison of the performances of the two algorithms, we note that

(a) *for any fixed problem size,* MP1 *becomes more and more superior compared to* DGW *as sample space increases,*

(b) for any fixed sample space, MP1 is better than DGW for smaller problems, but DGW becomes more and more superior compared to MP1 as the problem size increases.

We can explain the above phenomena as follows. The algorithm MP1 is designed to take advantage of the heterogeneity in the processing times. Whereas, DGW exploits very efficiently the homogeneity and equalities among the processing times using the dominance condition (2.3.2) given in Lemma 2.3.1. For example, in a problem with 100 jobs and processing times generated from the sample space {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, the number of equalities among the processing times is very high. For such a problem, the dominance condition (2.3.2) will effectively fathom a large number of nodes and thereby enables the algorithm DGW to obtain optimal solution involving less computational effort. The following numerical example illustrates this point.

Example 2.5.2 : Consider the problem data given below for $n = 60$.

j	p_j	j	p_j	j	p_j	j	p_j	j	p_j	j	p_j
1	9	11	8	21	6	31	5	41	4	51	2
2	9	12	7	22	6	32	5	42	3	52	2
3	8	13	7	23	6	33	5	43	3	53	2
4	8	14	7	24	6	34	4	44	3	54	2
5	8	15	7	25	5	35	4	45	3	55	2
6	8	16	7	26	5	36	4	46	3	56	2
7	8	17	7	27	5	37	4	47	3	57	1
8	8	18	6	28	5	38	4	48	3	58	1
9	8	19	6	29	5	39	4	49	2	59	1
10	8	20	6	30	5	40	4	50	2	60	1

In order to solve the above problem, the number of nodes created by DGW algorithm at different stages are given below.

Stage	No. of Nodes	Stage	No. of Nodes	Stage	No. of Nodes	Stage	No. of Nodes
1	1	16	25	31	579	46	1998
2	1	17	29	32	641	47	2139
3	1	18	33	33	703	48	2190
4	1	19	37	34	765	49	2188
5	2	20	74	35	827	50	2039
6	3	21	111	36	889	51	1849
7	4	22	148	37	987	52	1659
8	5	23	183	38	1085	53	1469
9	6	24	218	39	1183	54	1279
10	7	25	253	40	1281	55	1089
11	8	26	288	41	1379	56	899
12	9	27	323	42	1477	57	709
13	13	28	393	43	1575	58	520
14	17	29	455	44	1716	59	272
15	21	30	517	45	1857	60	544

We shall now describe graphically the performances of **DGW** and **MP1**, in terms of number of nodes created, for the following numerical example. This description gives a better insight into the nature of the algorithms.

Example 2.5.3 : Let $n = 50$ and the processing times be as given in the table below.

j	p_j	j	p_j	j	p_j	j	p_j	j	p_j
1	49	11	42	21	31	31	19	41	6
2	48	12	42	22	30	32	18	42	6
3	48	13	42	23	29	33	15	43	6
4	47	14	36	24	27	34	15	44	5
5	46	15	35	25	26	35	15	45	5
6	45	16	34	26	24	36	13	46	5
7	45	17	34	27	23	37	10	47	5
8	44	18	33	28	23	38	9	48	4
9	44	19	33	29	21	39	9	49	4
10	43	20	31	30	20	40	8	50	4

Figure 2.5.1 gives the number of nodes created by **DGW** and **MP1** at every stage. It is to be noted that the stages for **DGW** are numbered from right to left. For either algorithm, the area under the curve representing the number of nodes is proportional (approximately) to the total number of nodes created.

It may be seen from Figure 2.5.1 that as the algorithm **MP1** progresses through stages $3, 4, \ldots, n-1$, the number of nodes gradually increases upto some stage and then decreases steadily. It is true for **DGW** also. This phenomenon is observed in all the problems investigated by us for this purpose. However, the locations of the peaks of both the curves vary with problem instances.

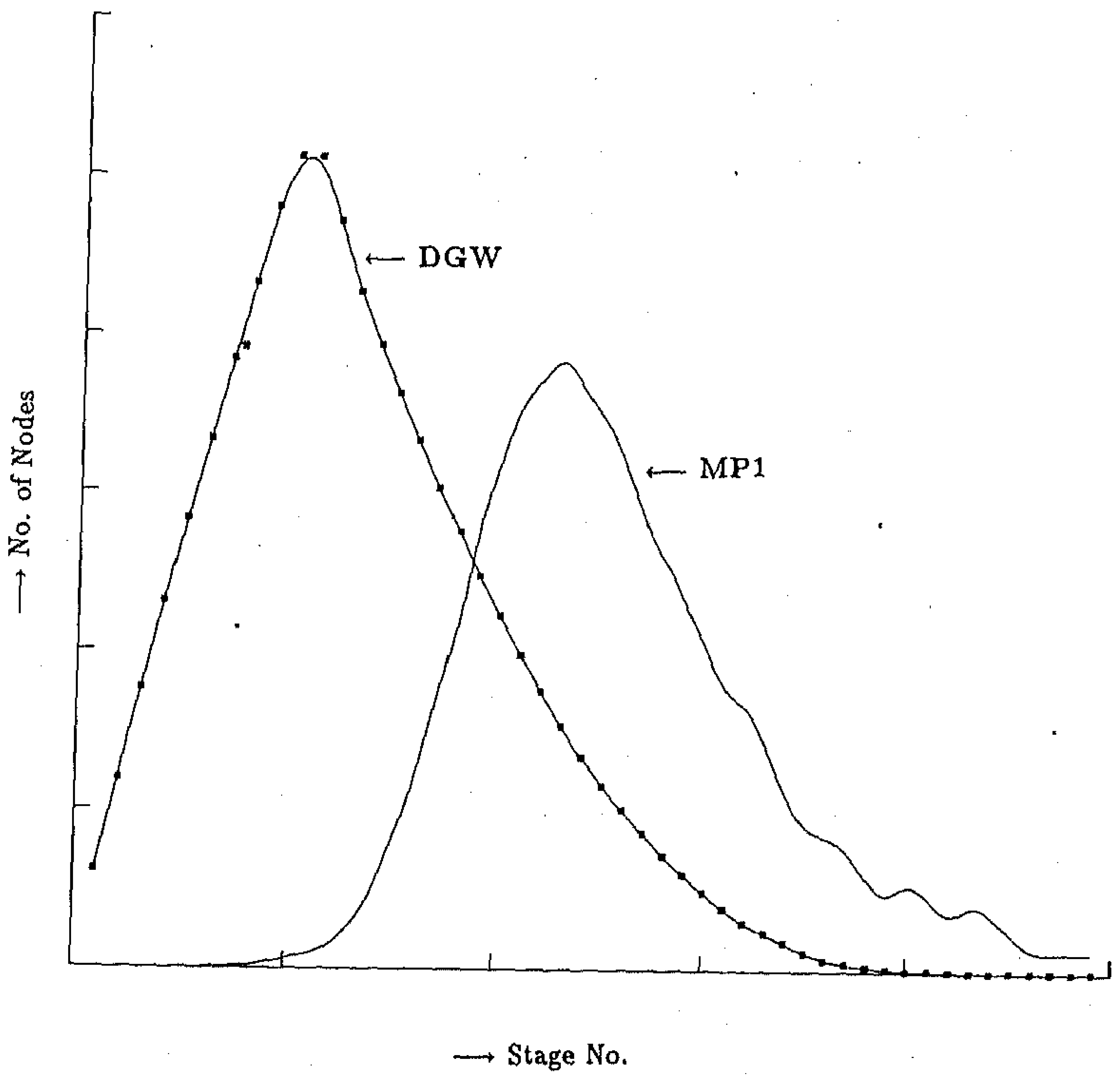

Figure 2.5.1 : Nodes Created at different stages by DGW and MP1

2.6 A HYBRID ALGORITHM

In order to take advantage of the contrasting merits of the algorithms **DGW** and **MP1**, we have developed a new algorithm which consists of two phases, the first phase involving **MP1** and the second involving **DGW**.

This algorithm, in the first phase using **MP1**, obtains all the nondominated two-sided partial sequences of the form $(\alpha, \ldots, \beta)$ where $\alpha \cup \beta = \{1, 2, \ldots, g\}$ for some predetermined integer g $(3 \leq g \leq n-1)$. In the next phase, by applying **DGW** algorithm, it decides on all nondominated partial sequences γ's where γ is a permutation of the jobs $g+1, g+2, \ldots, n$. Then, from every combination of $(\alpha, \ldots, \beta)$ and γ, a complete sequence $\pi = (\alpha, \gamma, \beta)$ is derived. The CTV values of all such complete sequences are compared to find an optimal solution. The algorithm, denoted by **MP2**, is presented below.

Algorithm MP2 :

Step 1 : Choose an integer g, $3 \leq g \leq n-1$.

Step 2 : Apply algorithm MP1 for $k = 3$ to g. Let $\mathcal{L} = \{(\alpha, \ldots, \beta) : \alpha \cup \beta = \{1, 2, \ldots, g\}\}$ be the set of all nondominated nodes obtained from **MP1**.

Step 3 : Apply algorithm DGW for $k = 2$ to $n-g$. Let $\mathcal{S} = \{\gamma : \gamma$ is a partial sequence of the jobs $g+1, \ldots, n\}$ be the set of all nondominated nodes obtained from **DGW**.

Step 4 : For each complete sequence $\pi = (\alpha, \gamma, \beta)$ where $(\alpha, \ldots, \beta) \in \mathcal{L}$ and $\gamma \in \mathcal{S}$, compute the CTV value, $V(\pi)$. Let $V(\pi^*) = \min_{\pi} V(\pi)$. Return π^* as optimal sequence with $V(\pi^*)$ as the corresponding optimal CTV value.

Proof of Correctness of MP2 : Let $\mathcal{V}$ be the set of all V-shaped sequences of the jobs $1, 2, \ldots, n$. Let $\mathcal{V}_1 = \{(\alpha, \ldots, \beta) : \alpha \cup \beta = \{1, 2, \ldots, g\}$ and (α, β) is a V-shaped partial sequence $\}$ and $\mathcal{V}_2 = \{\gamma : \gamma$ is a V-shaped partial sequence of the jobs $g+1,\dots,n\}$. Obviously, $\mathcal{L} \subseteq \mathcal{V}_1$ and $\mathcal{S} \subseteq \mathcal{V}_2$. Also, let $\mathcal{W} = \{(\alpha,\gamma,\beta) : (\alpha,\dots,\beta) \in \mathcal{L}$ and $\gamma \in \mathcal{S}\}$.

It is enough to show that if $\mathcal{W} \neq \phi$,

$$\text{for any } (\alpha',\gamma',\beta') \in \mathcal{V} \setminus \mathcal{W}, \text{ there exists } (\alpha,\gamma,\beta) \in \mathcal{W} \text{ such that } (\alpha,\gamma,\beta) \preceq (\alpha',\gamma',\beta') \tag{2.6.11}$$

We know that

(i) for every $(\alpha',\dots,\beta') \in \mathcal{V}_1$, there exists $(\alpha,\dots,\beta) \in \mathcal{L}$ such that $(\alpha,\gamma',\beta) \preceq (\alpha',\gamma',\beta')$ for all $\gamma' \in \mathcal{V}_2$ (2.6.12)

and (ii) for every $\gamma' \in \mathcal{V}_2$, there exists $\gamma \in \mathcal{S}$ such that $(\alpha',\gamma,\beta') \preceq (\alpha',\gamma',\beta')$ for all $(\alpha',\dots,\beta') \in \mathcal{V}_1$ (2.6.13)

Suppose $\mathcal{W} \neq \phi$ and let $(\alpha',\gamma',\beta') \in \mathcal{V} \setminus \mathcal{W}$. We pick $(\alpha,\dots,\beta) \in \mathcal{L}$ and $\gamma \in \mathcal{S}$ such that

$$(\alpha,\gamma',\beta) \preceq (\alpha',\gamma',\beta') \tag{2.6.14}$$
$$(\alpha',\gamma,\beta') \preceq (\alpha',\gamma',\beta') \tag{2.6.15}$$
$$(\alpha,\gamma,\beta) \preceq (\alpha',\gamma,\beta') \tag{2.6.16}$$
$$\text{and } (\alpha,\gamma,\beta) \preceq (\alpha,\gamma',\beta) \tag{2.6.17}$$

It is possible due to (2.6.12) and (2.6.13). Now, using (2.6.14) and (2.6.17) we get,

$$(\alpha,\gamma,\beta) \preceq (\alpha,\gamma',\beta) \preceq (\alpha',\gamma',\beta') \tag{2.6.18}$$

Note that the dominance relation under consideration is transitive. Hence it follows from the relation (2.6.18) that (α,γ,β) $(\in \mathcal{W})$ dominates (α',γ',β'), that is (2.6.11) holds good. This completes the proof. ■

Remark 2.6.1 : It is evident that the algorithm MP2 is also pseudopolynomial in complexity.

Remark 2.6.2 : MP2 involves the parameter g ($3 \leq g \leq n-1$). It may be noted that (i) MP2 coincides with MP1 for $g = n-1$, and (ii) when $g = 3$, MP2 is same as DGW except for the positions of the three largest jobs. The performance of MP2 depends on the value of the parameter g. Consider the following numerical example to see the effect of the value of g on the computational time required by MP2.

Example 2.6.1 : Let $n = 40$ and the processing times be given in the following table.

j	p_j	j	p_j	j	p_j	j	p_j
1	49	11	38	21	24	31	14
2	49	12	36	22	23	32	12
3	48	13	35	23	19	33	11
4	47	14	35	24	19	34	7
5	47	15	31	25	19	35	7
6	47	16	30	26	18	36	6
7	46	17	28	27	18	37	3
8	45	18	28	28	17	38	3
9	43	19	27	29	16	39	3
10	42	20	24	30	15	40	2

To solve the above problem, the execution times required by MP2 for different values of g are given below.

Value of g	5	10	15	20	25	30	35
Execution time (in seconds)	29.9	10.6	5.4	13.1	24.1	28.0	28.5

It can be noticed that as the value of g increases, the required computational time initially decreases upto some point, and then increases. We observe that optimal value of g depends on the problem dimension and the magnitudes of the processing times.

Remark 2.6.3 : In order to explore the optimal value of g, we have carried out extensive numerical experimentation, and we have the following observations :

(a) The computational time required by MP2 describes a convex function of g.

(b) For homogeneous (heterogeneous) processing times, the performance of MP2 is better with smaller (larger) value of g.

This computational study also strongly indicates the prospect of improving the performance of MP2 by proper selection of the value of g. However, the problem of optimal selection of the value of g remains open.

2.7 THE ALGORITHM OF KUBIAK [1995]

In a very recent publication, Kubiak [1995] has formulated the CTV problem as a problem of maximizing a zero-one quadratic function which is submodular with special cost structure, and has developed a pair of dynamic programs in order to maximize this function.

We make the following departure (only in this Section) from our standard notation to describe the submodular function and the proposed pair of dynamic programming (DP) algorithms. Let $N = \{1, 2, \ldots, n, n+1\}$ be the jobset with $p_1 \leq p_2 \leq \cdots \leq p_n \leq p_{n+1}$.

Kubiak [1995] also confines to the set of V-shaped sequences with the largest job in the first position (refer to Theorems 2.2.2 and 2.2.5) and has shown that the CTV problem is equivalent to the following problem :

$$\max_{y \in \{0,1\}^n} \{f(y)\} \tag{2.7.19}$$

where

$$
\begin{aligned}
f(y) &= \sum_{2\leq j<i\leq n} d_{ij}\,(y_i \oplus y_j)\,, \\
d_{ij} &= \beta_i\gamma_j, \\
\beta_i &= (n-i+2)p_i + \sum_{1\leq j\leq i-1} p_j, \\
\gamma_j &= (j-1)p_j + \sum_{1\leq r\leq j-1} p_r
\end{aligned}
$$

and corresponding to a sequence π, $y = (y_1, \ldots, y_n)$ with

$$
\begin{aligned}
y_i &= \begin{cases} 0 & \text{if job } i \text{ appears before job 1} \\ 1 & \text{otherwise,} \end{cases} \\
y_i \oplus y_j &= \bar{y}_i y_j + y_i \bar{y}_j, \\
\bar{y}_i &= 1 - y_i.
\end{aligned}
$$

In order to solve the problem given by (2.7.19), Kubiak has proposed the following pair of DP algorithms.

(I) Backward DP :

Solve for $h(3,0) = \max_{\mathbf{y}\in\{0,1\}^n}\{f(\mathbf{y})\}$ using the recursive relation,

$$
\begin{aligned}
h(k,\gamma) = \max_{y_{k,\gamma}\in\{0,1\}} &\{(1-y_{k,\gamma})\,[h(k+1,\gamma)+\beta_k\gamma]\,, \\
&y_{k,\gamma}\left[h(k+1,\gamma+\gamma_k)+\beta_k(\gamma^*_{k-1}-\gamma)\right]\}
\end{aligned} \tag{2.7.20}
$$

with $h(n+1,\gamma) = 0$ for all γ, $\gamma^*_k = \sum_{2\leq j\leq k}\gamma_j$ for $2 \leq k \leq n-2$, $\gamma^*_{n-1} = \gamma^* = \sum_{1\leq j\leq n-1}(2j-n)p_j$ and, for any γ, $y_{k,\gamma}$ has the same meaning as y_k.

(II) Forward DP :

Solve for $g(n-1,0) = \max_{\mathbf{y}\in\{0,1\}^n}\{f(\mathbf{y})\}$ using the recursive relation,

$$
\begin{aligned}
g(k,\beta) = \max_{y_{k,\beta}\in\{0,1\}} &\{(1-y_{k,\beta})\,[g(k-1,\beta)+\gamma_k\beta]\,, \\
&y_{k,\beta}\,[g(k-1,\beta+\beta_k)+\gamma_k(\beta^*_k-\beta)]\}
\end{aligned} \tag{2.7.21}
$$

with $g(1,\beta) = 0$ for all β, $\beta_k^* = \sum_{k+1\leq j\leq n} \beta_j$ for $3 \leq k \leq n$, $\beta_2^* = \beta^* = 2\sum_{3\leq i\leq n}(n-i+1)p_i + (n-2)(p_1+p_2)$ and, for any β, $y_{k,\beta}$ has the same meaning as y_k.

Remark 2.7.1 : Kubiak [1995] has shown that the time complexity of the above algorithms are $o(n\gamma^*)$ and $o(n\beta^*)$ respectively. The criterion to choose among the algorithms is based on the minimum of γ^* and β^* for any given problem instance. ■

2.8 COMPARISON OF ALGORITHMS

We now compare **DGW** and **MP2** alongwith the very recently published algorithm of Kubiak [1995] (described in Section 2.7) by extensive numerical investigation. We denote the algorithm of Kubiak [1995] by **KBK**.

For the purpose of comparison, we have taken the same problem set as considered for the comparison of **DGW** and **MP1**, and the investigation is carried out under identical computing environment. The computer programs (coded in FORTRAN) for all the three algorithms give both the optimal sequence and its CTV value.

It is already noticed that the performance of **MP2** is dependent on the value of the parameter g. We fix $g = \lfloor 0.30n \rfloor$ in this investigation.

The summarized computational results are presented in the Tables 2.8.1(a), 2.8.1(b) and 2.8.1(c). For each combination of n and U_r, the tables contain the average execution time for ten problem instances as required by **DGW**, **MP2** and **KBK**.

Table 2.8.1(a) : Average Execution Time (in seconds)

	U_{10}			U_{50}		
n	DGW	MP2	KBK	DGW	MP2	KBK
20	0.02	*	*	0.70	0.06	0.02
30	0.25	0.03	0.02	7.34	1.23	0.17
40	0.96	0.21	0.04	34.12	6.74	0.55
50	3.34	0.66	0.09	109.37	21.64	1.24
60	9.12	1.56	0.18	281.57	72.98	2.58
70	19.83	4.08	0.35	608.43	200.33	4.97
80	37.99	7.46	0.59			
90	75.09	12.02	0.74			
100	119.66	28.71	1.07			

Note : * indicates negligible.

Table 2.8.1(b) : Average Execution Time (in seconds)

	U_{100}			U_{200}		
n	DGW	MP2	KBK	DGW	MP2	KBK
20	2.13	0.14	0.05	9.11	0.33	0.12
30	27.77	3.42	0.41	120.38	17.83	0.63
40	133.70	21.71	1.34	510.72	103.26	2.74
50	434.32	110.68	2.97			

Table 2.8.1(c) : Average Execution Time (in seconds)

	U_{300}			U_{500}		
n	DGW	MP2	KBK	DGW	MP2	KBK
20	15.90	0.42	0.40	39.14	0.55	0.90
30	260.45	41.78	3.75	634.49	78.20	**
40	1044.46	191.14	6.36	3673.24	287.90	**

Note : ** indicates that the algorithm failed to produce solution because of insufficient computer memory.

Remark 2.8.1 : Based on this numerical investigation, we infer the following :

(a) KBK is extremely good in terms of the required execution time.

(b) MP2 is superior to DGW.

(c) As the sample space increases for fixed number of jobs, MP2 becomes better than KBK.

(d) The computer memory required by KBK is higher than that required by DGW or MP2. As a matter of fact, for $n = 30$ and 40 with the sample space U_{500}, KBK failed to produce solution because of this reason.

2.9 LOWER BOUND ON CTV AND AN IMPLICIT ENUMERATION METHOD

In this section, we deal with the derivation of lower bound on CTV, and then, use this lower bound in the development of an implicit enumeration method to solve the CTV problem with heterogeneous non-integer processing times.

2.9.1 Lower Bound on CTV

Derivation of lower bound is important to develop good algorithms. For example, approximation scheme of De, Ghosh and Wells [1992] and implicit enumeration method of Manna and Prasad [1994] make use of lower bound on CTV, and the performance of these algorithms can be enhanced considerably by using sharper lower bounds. De, Ghosh and Wells [1992] have presented a fully polynomial approximation scheme to obtain a solution with relative error bounded by a specified ϵ (> 0). The scheme involves lower bound on CTV in such a way that the computational effort required by the scheme for a specified ϵ reduces as the lower bound becomes sharper.

In this subsection, we discuss the derivation of lower bound on CTV. De, Ghosh and Wells [1990,1992] are the first to deal with this problem. Later

Manna and Prasad [1994] and Mittenthal, Raghavachari and Rana [1994] observe the infeasibility of the solution given by De, Ghosh and Wells [1992]. Prasad, Manna and Arthanari [1994] give the correct version of the solution. Whereas Mittenthal, Raghavachari and Rana [1994] propose an improved lower bounding scheme.

In order to find a lower bound for the CTV, De, Ghosh and Wells [1992] consider a relaxation of CTV in which processing times of the jobs are assumed to be variables, and the objective is to find the processing times which minimizes CTV subject to the restrictions - (i) sum total of these processing time variables (say, x_j's) equals the makespan ($= \sum_{j=1}^{n} p_j$) of the given problem instance, (ii) upper limit on x_j's using V-shaped property of optimal sequence, and (iii) lower limit on x_j's as p_n.

Let x_j ($1 \leq j \leq n$) be the unknown processing time of a job in the jth position of a sequence. Then, using Lemma 2.2.1, the CTV is given by,

$$\begin{aligned} \text{CTV} &= \sum_{j=1}^{n} \frac{(j-1)(n-j+1)}{n^2} x_j^2 + 2 \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} \frac{(j-1)(n-i+1)}{n^2} x_j x_i \\ &= \frac{1}{n^2} \boldsymbol{x}^T D \boldsymbol{x} \end{aligned}$$

where $\boldsymbol{x}^T = (x_2, x_3, \ldots, x_n)$, $D = ((d_{ij}))$ is the symmetric matrix of order $(n-1) \times (n-1)$ with $d_{ii} = i(n-i)$, $d_{ij} = i(n-j)$ for $i < j$.

Thus, the objective function is given by $\boldsymbol{x}^T D \boldsymbol{x}$ that it is to be minimized subject to the constraints :

(i) $\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} p_j$. Using Theorem 2.2.2, we take $x_1 = p_1$ so that $\sum_{j=2}^{n} x_j = \sum_{j=2}^{n} p_j$.

(ii) By V-shaped property of optimal sequence (refer to Theorem 2.2.5), $x_j \leq p_j$ and $x_{n-j+2} \leq p_j$ for $j = 2, 3, \ldots, \lceil \frac{n+1}{2} \rceil$.

(iii) $x_j \geq p_n$ for $j = 2, 3, \ldots, n$.

Using the above, De, Ghosh and Wells [1992] have proposed as lower bound the optimal objective value of the following quadratic programming problem.

$$\text{(QP-1)} \qquad \begin{array}{lrcl} \text{Minimize} & x^T D x & & \\ \text{subject to} & e^T x & = & MS - p_1, \\ & p_n & \leq & x_j \ \leq \ p_j \\ \text{and} & p_n & \leq & x_{n-j+2} \ \leq \ p_j \\ \text{for} & j & = & 2, 3, \ldots, \lceil \frac{n+1}{2} \rceil \end{array}$$

where e is $(n-1)$-component column vector of 1's and $MS = \sum_{j=1}^{n} p_j$.

They have also given a solution to this problem in algebraic form. As observed by Manna and Prasad [1994], this solution is not always feasible. For example, their solution $x_2 = x_5 = p_2$, $x_3 = x_4 = p_5$ with $0 < p_3 + (p_4 - p_5) - p_2 < 2p_5$ for $n = 5$ is not feasible. Moreover, De, Ghosh and Wells [1992] have given an approach based on Karush-Kuhn-Tucker conditions to justify this solution. Following similar approach, we derive a correct and simple form of optimal solution.

Derivation of Lower Bound :

Let $p'_j = p_j - p_n$ for $j = 2, \ldots, n$ and $W = MS - p_1 - (n-1)p_n$. Consider the following quadratic programming problem which is equivalent to **QP-1**.

$$\text{(QP-2)} \qquad \begin{array}{lr} \text{Minimize} & c^T y + \frac{1}{2} y^T (2D) y \\ \text{subject to} & Ay \leq b \\ \text{and} & y \geq 0 \end{array}$$

where

$$A_{(n+1)\times(n-1)} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ . & . & \cdots & . \\ . & . & \cdots & . \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

$c = \frac{1}{n}p_n(n-1, 2(n-2), \ldots, j(n-j), \ldots, (n-2)2, n-1)^T$, $y = (y_2, y_3, \ldots, y_n)^T$, $b^T = (b_1, b_2, \ldots, b_{n+1})$ with $b_1 = -b_2 = W$, and $b_j = b_{n-j+4} = p'_{j-1}$ for $j = 3, \ldots, \lfloor\frac{n+4}{2}\rfloor$.

It can be easily verified that if $x = (x_2, \ldots, x_n)^T$ is a feasible solution to the problem **QP-1**, then $y = (y_2, \ldots, y_n)^T = (x_2 - p_n, \ldots, x_n - p_n)^T$ is feasible solution of **QP-2** and $x^T D x = c^T y + y^T D y + p_n{}^2(n^2-1)/12$. Since $x^T D x$ is known to be convex, the function $c^T y + y^T D y$ is also convex. In fact both of them are strictly convex for $n > 2$. Thus, Karush-Kuhn-Tucker (KKT) conditions for **QP-2** give the unique optimal solution (refer to Bazaraa and Shetty [1979]). KKT conditions for **QP-2** are

$$
\begin{aligned}
c + 2Dy + A^T\lambda &\geq 0, && (2.9.22)\\
Ay &\leq b, && (2.9.23)\\
\lambda^T(Ay - b) &= 0, && (2.9.24)\\
y^T(c + 2Dy + A^T\lambda) &= 0, && (2.9.25)\\
y \geq 0, \quad \lambda \geq 0. & && (2.9.26)
\end{aligned}
$$

Let us denote the vector λ by $(\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_n)^T$.

Theorem 2.9.1 : If $p'_2 \geq \frac{1}{2}W$, the solution (y, λ) with

i) $y_2 = y_n = \frac{1}{2}W$, $y_j = 0$ for $j = 3, \ldots, n-1$,

ii) $\lambda_0, \lambda_1 \geq 0$ such that $(\lambda_1 - \lambda_0) = [(n-1)p_n + W]/n$, and

iii) $\lambda_j = 0$ for $j = 2, \ldots, n$

satisfies conditions (2.9.22) to (2.9.26).

This can be proved by verifying that the solution (y, λ) is a KKT point. Therefore, the solution $y_2 = y_n = \frac{1}{2}W$ and $y_j = 0$ for $j = 3, \ldots, n-1$ is the optimal solution of **QP-2**.

Note that if $p'_2 < \frac{1}{2}W$, then there exists an integer k, $2 \le k \le \lfloor \frac{n-1}{2} \rfloor$ such that

$$\sum_{j=1}^{k-1} p'_{j+1} < \frac{1}{2}W \le \sum_{j=1}^{k} p'_{j+1}. \tag{2.9.27}$$

Theorem 2.9.2 : Let $p'_2 < \frac{1}{2}W$ and an integer k satisfy (2.9.27). Then the solution,

$$\begin{aligned} y_j = \; & y_{n-j+2} = p'_j \quad \text{for } j = 2, \ldots, k, \\ y_{k+1} = \; & y_{n-k+1} = \frac{1}{2}\left[W - 2\sum_{j=1}^{k-1} p'_{j+1})\right] \\ \text{and} \quad y_j = \; & 0 \quad \text{for } j = k+2, \ldots, n-k, \end{aligned}$$

is uniquely optimal to QP-2.

Proof : Let $\lambda = (\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_n)^T$ be a non-negative vector such that

$$\begin{aligned} \lambda_1 - \lambda_0 &= \left[k(n-k)p_n + 2\sum_{j=1}^{k} j y_{j+1}\right]/n, \\ \lambda_j = \lambda_{n-j+2} &= [(\lambda_1 - \lambda_0) - (j-1)(n-j+1)p_n \\ &\quad -2(y_2 + 2y_3 + \ldots + (j-1)(y_j + \ldots + y_{k+1})]/n \\ &\quad \text{for } j = 2, 3, \ldots, k \\ \text{and} \quad \lambda_j &= 0 \;\text{ for } j = k+1, \ldots, n-k+1. \end{aligned}$$

It can be easily seen that (y, λ) satisfies KKT conditions (2.9.22) to (2.9.26) and therefore the solution y is uniquely optimal to QP-2. ■

A Superior Lower Bound :

Later, we come to know about a lower bounding procedure developed by Mittenthal, Raghavachari and Rana [1994]. The basic approach is similar to that of De, Ghosh and Wells [1992] by solving a quadratic programming problem where the job processing times are again assumed to be decision variables x_j's defined earlier in this subsection. However, Mittenthal, Raghavachari and Rana [1994] exploit the V-shaped property of optimal sequence in a better way by considering the following constraints :

$$\begin{aligned} x_2 + x_n &\leq p_2 + p_3 \\ x_2 + x_3 + x_{n-1} + x_n &\leq p_2 + p_3 + p_4 + p_5 \\ &\vdots \\ &\vdots \\ \text{and so on} & \end{aligned}$$

Using the above set of constraints, in addition to $\sum_{j=2}^{n} x_j = \sum_{j=2}^{n} p_j$, Mittenthal, Raghavachari and Rana [1994] have minimized the $\boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}$ and obtained optimal solution $\boldsymbol{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ as

$$\begin{aligned} x_1^* &= p_1 \\ x_2^* = x_n^* &= (p_2 + p_3)/2 \\ x_3^* = x_{n-1}^* &= (p_4 + p_5)/2 \\ &\vdots \\ &\vdots \\ x_{\frac{n}{2}+1}^* &= p_n \qquad \text{if } n \text{ is even} \\ x_{\frac{n+1}{2}}^* = x_{\frac{n+1}{2}+1}^* &= (p_{n-1} + p_n)/2 \qquad \text{if } n \text{ is odd} \end{aligned}$$

Remark 2.9.3 : It can be easily verified that the above lower bounding procedure is at least as good as the one derived from Theorems 2.9.1 and 2.9.2.

Remark 2.9.4 : We observe from the structure of the solution $\boldsymbol{x}^*$ given above that the procedure is likely to perform well when the p_j's are homogeneous.

Computational Results :

We have carried out numerical investigation to assess the merit of the above lower bounding procedure developed by Mittenthal, Raghavachari and Rana [1994]. For each value of n (n = 10, 20, ..., 70), ten problem-instances are randomly generated with p_j's from discrete uniform distribution $U\{1, 2, \ldots, 50\}$. For every problem, the percentage relative error of the lower bound (LB) is measured as $\frac{V_0 - LB}{V_0} \times 100$ where V_0 is the minimum (optimal) CTV value. The minimum CTV value, V_0, is obtained with the help of MP2 algorithm (presented in Section 2.6). Table 2.9.1 gives the minimum, average and maximum relative error for each value of n.

Table 2.9.1 : Percentage Relative Errors

Number	Percentage Relative Error		
of jobs (n)	Minimum	Average	Maximum
10	0.0061	0.0903	0.2235
20	0.0046	0.0121	0.0219
30	0.0012	0.0029	0.0065
40	0.0005	0.0010	0.0023
50	0.0002	0.0003	0.0005
60	0.0000	0.0001	0.0003
70	0.0000	0.0001	0.0002

Remark 2.9.5 : It must be observed from Table 2.9.1 that the overall performance of the lower bounding procedure developed by Mittenthal, Raghavachari and Rana [1994] is highly satisfactory. We also note that the performance becomes superior for larger problem sizes. This is due to the fact that the job processing times become more homogeneous for larger n.

Remark 2.9.6 : If p_j's are very heterogeneous, the performance of the above procedure may not be so well. For example, consider a six-job problem with

$p_1 = 1000$, $p_2 = 996$, $p_3 = 4$, $p_4 = 3$, $p_5 = 2$ and $p_6 = 1$. The lower bound on the CTV, for this problem calculated by the procedure of Mittenthal, Raghavachari and Rana [1994] is 84339.4167. However, the minimum (optimal) CTV value for the problem is 138899.8889. It means that the percentage relative error can be as high as 39.28%.

2.9.2 An Implicit Enumeration Method

In majority of the real-life situations, we come across non-integer processing times. With non-integer, but rational processing times, one can always adopt the algorithms **DGW**, **MP1**, **MP2** or **KBK** (described in the earlier Sections) to solve CTV problem. However, they may not be very effective if the number of decimal places in processing times are large (two or more decimal places), since they are of pseudopolynomial complexity.

To solve the CTV problem with heterogeneous non-integer processing times, we now present an implicit enumeration algorithm based on branch-and-bound approach. The algorithm is very similar to **MP1** except for the following modifications :

(a) Selection of an existing node for the purpose of branching is based on minimum lower bound for CTV associated with the nodes.

(b) Dominance rule, given in Lemma 2.4.4, is not applied.

The notation used in **MP1** algorithm is also followed here.

The following lemma provides a lower bound on CTV for any node $(\alpha, \ldots, \beta)$.

Lemma 2.9.7 : Let π_L be an arbitrary V-shaped partial sequence of jobs $k+1, k+2, \ldots, n$. Let $\pi = (\alpha, k, \pi_L, \beta)$ and $\pi' = (\alpha, \pi_L, k, \beta)$ be two complete sequences generated from the nodes $(\alpha, k, \ldots, \beta)$ and $(\alpha, \ldots, k, \beta)$ respectively.

Then

$$
\begin{aligned}
(\ i)\ \ nV(\pi) \ &\geq \ kV(\pi_{(k)}) + 2M_L \sum_{j\in\beta} \left[C_j(\pi_{(k)}) - \bar{C}(\pi_{(k)})\right] \\
&\quad + \frac{b(k-b)}{k} M_L^2 + (n-k)\Upsilon(\pi_L), \qquad (2.9.28) \\
(ii)\ \ nV(\pi') \ &\geq \ kV(\pi_{(k)}) + 2M_L \sum_{j\in\beta\cup\{k\}} \left[C_j(\pi_{(k)}) - \bar{C}(\pi_{(k)})\right] \\
&\quad + \frac{(b+1)(k-b-1)}{k} M_L^2 + (n-k)\Upsilon(\pi_L) \qquad (2.9.29)
\end{aligned}
$$

where $\pi_{(k)} = (\alpha, k, \beta)$, $b = |\beta|$, $M_L = \sum_{j\in\pi_L} p_j$ and, $\Upsilon(\pi_L)$ is a lower bound on the CTV for the subproblem with jobset as $L = \{k+1, k+2, \ldots, n\}$.

Proof : Let $E = \alpha \cup \{k\} \cup \beta$. Using variance partition formula we can write

$$
\begin{aligned}
nV(\pi) \ &= \ \sum_{j\in E} \left[C_j(\pi) - \bar{C}_E(\pi)\right]^2 + \sum_{j\in\pi_L} \left[C_j(\pi) - \bar{C}_{\pi_L}(\pi)\right]^2 \\
&\quad + \frac{k(n-k)}{n} \left[\bar{C}_E(\pi) - \bar{C}_{\pi_L}(\pi)\right]^2 \\
&\qquad (\text{where } \ \bar{C}_E(\pi) = \frac{1}{k}\sum_{j\in E} C_j(\pi), \text{ and } \bar{C}_{\pi_L}(\pi) = \frac{1}{n-k}\sum_{j\in\pi_L} C_j(\pi)) \\
&= \ T_1 + T_2 + T_3, \quad (\text{say})
\end{aligned}
$$

It is obvious that $C_j(\pi) = \begin{cases} C_j(\pi_{(k)}), & j \in \alpha \cup \{k\} \\ C_j(\pi_{(k)}) + M_L, & j \in \beta \end{cases}$

and $\bar{C}_E(\pi) = \bar{C}(\pi_{(k)}) + \frac{1}{k} b M_L$

Therefore,

$$
\begin{aligned}
T_1 \ &= \ \sum_{j\in\alpha\cup\{k\}} \left[C_j(\pi_{(k)}) - \bar{C}(\pi_{(k)}) - \frac{bM_L}{k}\right]^2 \\
&\quad + \sum_{j\in\beta} \left[C_j(\pi_{(k)}) + M_L - \bar{C}(\pi_{(k)}) - \frac{bM_L}{k}\right]^2 \\
&= \ kV(\pi_{(k)}) - \frac{2bM_L}{k} \sum_{j\in\alpha\cup\{k\}} \left[C_j(\pi_{(k)}) - \bar{C}(\pi_{(k)})\right] + (k-b)\frac{b^2M_L^2}{k^2}
\end{aligned}
$$

$$+\frac{2(k-b)M_L}{k}\sum_{j\in\beta}\left[C_j(\pi_{(k)})-\bar{C}(\pi_{(k)})\right]+b\frac{(k-b)^2M_L^2}{k^2}$$

$$= kV(\pi_{(k)})+2M_L\sum_{j\in\beta}\left[C_j(\pi_{(k)})-\bar{C}(\pi_{(k)})\right]+\frac{b(k-b){M_L}^2}{k}.$$

Next, let $\underline{V}(\pi_L)$ be a lower bound for CTV subproblem with jobs only in π_L. Then $T_2 \geq (n-k)\underline{V}(\pi_L)$, and obviously $T_3 \geq 0$. Hence part (i) follows.

The part (ii) can be shown in similar fashion. ■

Remark 2.9.8 : (i) In order to compute $\underline{V}(\pi_L)$ for any jobset L, we use the results of Mittenthal, Raghavachari and Rana [1994] described in the Subsection 2.9.1. (ii) The right hand side of inequality (2.9.28) ((2.9.29)) can be taken as a lower bound on CTV associated with the node $(\alpha, k, \ldots, \beta)$ $((\alpha, \ldots, k, \beta))$ respectively.

We now present the algorithm, denoted by **MP3**, which involves both the dominance criterion (in Lemma 2.4.1) and the lower bound on CTV (given by Lemma 2.9.7). Due to Theorem 2.2.8, the algorithm restricts the search to the set of V-shaped sequences of the form $(1, 3, \ldots, 2)$ and therefore it starts with the node $(1, 3, \ldots, 2)$.

Algorithm MP3 :

Step 0 : Initialize $W = \{(1, 3, \ldots, 2)\}$. Compute lower bound for $(1, 3, \ldots, 2)$ and set $MCTV = (MS)^2$.

Step 1 : If $W = \phi$, STOP. Otherwise, take a node $(\alpha, \ldots, \beta)$ with the least lower bound from W and set $W = W\setminus\{(\alpha, \ldots, \beta)\}$. If $|\alpha \cup \beta| = n-1$ go to Step 3. Otherwise, generate two nodes, say, $(\alpha, k, \ldots, \beta)$ and $(\alpha, \ldots, k, \beta)$ and go to Step 2.

Step 2 : If condition (2.4.5) holds, compute lower bound for $(\alpha, \ldots, k, \beta)$ and set $W = W \cup \{(\alpha, \ldots, k, \beta)\}$. If condition (2.4.6) holds, compute lower bound for $(\alpha, k, \ldots, \beta)$ and set $W = W \cup \{(\alpha, k, \ldots, \beta)\}$. If neither condition holds, compute lower bounds for $(\alpha, \ldots, k, \beta)$ and $(\alpha, k, \ldots, \beta)$ and set $W = W \cup \{(\alpha, k, \ldots, \beta), (\alpha, \ldots, k, \beta)\}$. Go to Step 1.

Step 3 : Compute $V(\pi)$ for the complete sequence $\pi = (\alpha, n, \beta)$. If $MCTV > V(\pi)$, set $MCTV = V(\pi)$, $\pi^* = \pi$ and delete all the nodes in W with lower bound not less than $V(\pi)$. Go to Step 1.

From a node $(\alpha, \ldots, \beta)$ with $|\alpha \cup \beta| < n - 1$, the algorithm generates two nodes $(\alpha, k, \ldots, \beta)$ and $(\alpha, \ldots, k, \beta)$ and includes nondominated ones of these two nodes in the collection W of active nodes after computing their lower bounds. The node $(\alpha, \ldots, \beta)$ becomes inactive and gets deleted from W. If $|\alpha \cup \beta| = n-1$, the variance of the single complete sequence (α, n, β) generated from $(\alpha, \ldots, \beta)$ is evaluated. The nodes in W with lower bound not less than this variance are deleted from W along with $(\alpha, \ldots, \beta)$. Always, the node with the least lower bound in W is chosen for branching.

It can be easily seen that the sequence π^* given by the algorithm is optimal with MCTV as its CTV value.

We now consider the following numerical example to demonstrate the effectiveness of **MP3** algorithm for non-integer processing times.

Example 2.9.1 : Let $n = 25$ and the processing times be as given in the table below.

j	p_j	j	p_j	j	p_j	j	p_j	j	p_j
1	9.23	6	7.60	11	6.59	16	5.30	21	3.50
2	8.54	7	7.45	12	6.28	17	5.25	22	3.49
3	8.23	8	7.20	13	6.14	18	4.86	23	2.85
4	8.19	9	6.91	14	5.84	19	4.41	24	2.81
5	7.95	10	6.78	15	5.39	20	4.04	25	2.62

It should be noted that the above problem instance could not be solved by KBK algorithm. This is due to the higher computer memory requirement by this algorithm. However, the execution times required by DGW, MP1, MP2 and MP3, to solve the above problem, are given below.

Algorithm	DGW	MP1	MP2 (with $g = \lfloor 0.30n \rfloor$)	MP3
Execution time (in seconds)	415.0	75.5	36.5	1.2

It must be noticed that MP3 is by far superior to DGW, MP1 and MP2 (with $g = \lfloor 0.30n \rfloor$). We also evaluate the performance (given below in terms of execution time in seconds) of MP2 for all possible values of g, and observe that it is best with $g = 13$.

g	Time	g	Time	g	Time	g	Time	g	Time
3	265.8	8	16.8	13	0.7	18	9.4	23	70.5
4	191.8	9	6.9	14	0.8	19	18.2	24	75.5
5	122.8	10	2.7	15	1.1	20	32.3		
6	69.3	11	1.3	16	2.3	21	51.4		
7	36.5	12	0.9	17	5.0	22	62.0		

Since the optimal selection of the value g is not completely resolved, MP3 algorithm is likely to be very useful to solve CTV problem when the processing times are non-integers.

2.10 DISCUSSION

This chapter is devoted to the exact procedures for solving the deterministic CTV problem. Since the problem is NP-hard, the recent approach has been towards the development of pseudopolynomial algorithm for the same. We have derived here two dominance rules which are in turn used in the development of a new pseudopolynomial algorithm (MP1). The algorithm MP1 involves binary branching making use of the V-shaped property. A node, in this algorithm, represents a two-sided partial sequence of the form $(\alpha, \ldots, \beta)$ containing the larger jobs. The MP1 algorithm is intended to take good advantage of the heterogeneity present among the processing times. In fact, it is highly effective for smaller number of jobs with very heterogeneous processing times.

However, it is no better than the DGW algorithm (of De, Ghosh and Wells [1992]) when the processing times are homogeneous which can happen if the sample space is small and the number of jobs is large.

The hybrid algorithm MP2 is developed in order to exploit the the contrasting merits of DGW and MP1. The complexity of this algorithm is also pseudopolynomial.

Based on extensive numerical investigation, it is found that the recent algorithm KBK (of Kubiak [1995]) is superior to both DGW as well MP2 in general. However, we have noticed the following drawbacks of KBK algorithm :

(a) As the heterogeneity among the processing times increases for fixed n, KBK becomes worse than MP2.

(b) The computer memory requirement of KBK is high in general, and it does pose some problems even with reasonable computing facility.

The algorithm MP3 is an implicit enumeration method involving a dominance rule and a lower bounding on CTV.

Observing an error in the lower bound given by De, Ghosh and Wells [1992] for CTV, Prasad, Manna and Arthanari [1994] and Mittenthal, Raghavachari and Rana [1994] have independently derived lower bounds for the same. It has been noted that the lower bound of Mittenthal, Raghavachari and Rana [1994] is sharper. We have incorporated it in our algorithm MP3. This algorithm is of particular importance for non-integer processing times. It is demonstrated by a numerical problem for which the KBK algorithm failed to obtain optimal solution due to large memory requirement.

CHAPTER 3

CTV PROBLEM WITH RANDOM PROCESSING TIMES

3.1 INTRODUCTION

In many real-life situations, the job processing times are not known at the time of scheduling the entire processing. In fact, the processing time of a job is known only after the job is processed. This is due to several uncertain elements in the processing requirement of jobs. For example, repair times, project execution times etc. are quite often random. If the variation in the processing time is known to be very small compared to its mean, from practical point of view, such variation can be ignored in scheduling. Otherwise, it must be taken into account while scheduling the processing with some objective.

In this chapter, we consider the problem of minimizing the variation among job completion times with partial or complete information. The objective under consideration is, to be precise, minimization of expected value of the completion time variance (CTV). It may be noted that in this case, all job completion times are random in nature and their probabilistic law varies with the schedule.

The stochastic version of the single machine CTV problem is studied by Chakravarthy [1986], Vani and Raghavachari [1987] and Prasad and Manna [1994]. Until now, no efficient algorithm (with polynomial time complexity) is available to solve the problem. As a matter of fact, this problem is more difficult compared to its deterministic version which is known to be NP-hard (see Kubiak [1993] and Cheng and Cai [1993]).

Chakravarthy [1986] and Vani and Raghavachari [1987] have shown independently that an important result derived by Merten Muller [1972] for the deterministic case holds good for the stochastic version also. Chakravarthy [1986] has shown that if all the P_j's have same means (variances) an optimal sequence is V-shaped in means (variances). Also, he has established that the property of

V-shapedness is a necessary condition for optimality when P_j's are exponentially distributed. Vani and Raghavachari [1987] have dealt a more general case assuming that for each P_j, the second (raw) moment can be expressed as a quadratic function of the first moment (mean). Under certain conditions, in this case, they have shown that optimal sequence is V-shaped.

In section 3.2, we present the preliminary results which are used later in this chapter to derive properties of optimal sequence and a lower bound on the expected CTV. Section 3.3 is devoted to the study on the properties of optimal sequences. In Subsection 3.3.1, we derive two sufficient conditions on the existence of V-shaped optimal sequence for the general stochastic CTV problem. Subsection 3.3.2 deals with the special case considered by Vani and Raghavachari [1987] and provides a simple but stronger sufficient criterion for V-shaped optimality. In Subsection 3.3.3, we introduce a special case in which the random processing times satisfy the following *order* property :

$$\mu_r < \mu_s \Rightarrow \sigma_r^2 < \sigma_s^2 \qquad \text{for any } r \text{ and } s.$$

In this case, we first demonstrate that V-shaped property is not a necessity for optimality. Then we derive a sufficient condition for the same and obtain several results on the monotonic property of optimal sequence. We also prove the existence of an L-G-S optimal sequence for this problem. In Section 3.4, we present a procedure to derive a lower bound for the expected CTV and a dominance rule which are in turn effectively used to develop a branch-and-bound algorithm to solve the stochastic CTV problem with general processing times. Further, we discuss here the required modifications in the algorithm for ordered processing times.

3.2 PRELIMINARY RESULTS

In this section, we present some preliminary results required for deriving the main results of this chapter.

Lemma 3.2.1 (Vani and Raghavachari [1987]): For a sequence $\boldsymbol{\pi} = (\pi_1, \pi_2,$

$\ldots, \pi_n)$, the expected variance of completion times under π is given by,

$$E[V(\pi)] = V_\mu(\pi) + \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\sigma^2_{\pi_r} \qquad (3.2.1)$$

where

$$\begin{aligned} V_\mu(\pi) &= \frac{1}{n}\sum_{r=1}^{n}\left[E_r(\pi) - \bar{E}(\pi)\right]^2 \\ &= \frac{1}{n^2}\sum_{r=1}^{n}(r-1)(n-r+1)\mu^2_{\pi_r} \\ &\quad + \frac{2}{n^2}\sum_{r=1}^{n-1}(r-1)\mu_{\pi_r}\sum_{s=r+1}^{n}(n-s+1)\mu_{\pi_s}, \qquad (3.2.2) \\ E_r(\pi) &= \sum_{i=1}^{r}\mu_{\pi_i} \\ \text{and } \bar{E}(\pi) &= \frac{1}{n}\sum_{r=1}^{n}(n-r+1)\mu_{\pi_r}. \end{aligned}$$

Remark 3.2.2 : It can be seen that the expression $E[V(\pi)]$ is independent of the parameters of the first job in π, but depends on the other jobs through their first two moments only.

Remark 3.2.3 : It is clear from the equation (3.2.1) that the special case $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_n^2$ becomes the deterministic version of the problem, and hence the stochastic problem under consideration is also NP-hard.

Theorem 3.2.4 : Let π be any sequence and π^D be its dual. Then

$$E[V(\pi)] = E[V(\pi^D)].$$

Remark 3.2.5 : The above theorem has been proved independently by Chakravarthy [1986] and Vani and Raghavachari [1987]. Note that this is a generalization of the result of Merten and Muller [1972] for the deterministic case. The theorem proves the existence of at least two optimal sequences.

Theorem 3.2.6 (Chakravarthy [1986]): If $\sigma_1^2 = \max_{1 \le j \le n} \sigma_j^2$, then there exists an optimal sequence of the form $(1, \ldots\ldots)$.

Proof : It can be proved directly from Lemma 3.2.1 by observing that given any sequence π,

i) each P_i has non-negative contribution to $E[V(\pi)]$ through its mean (μ_i) and variance (σ_i^2),

ii) $E[V(\pi)]$ is independent of the parameters of the first job in π.

■

The following result gives the change in the expected CTV due to interchange of two jobs in a sequence.

Lemma 3.2.7 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be any sequence. Let π' be a sequence obtained from π by interchanging the two jobs π_s and π_t $(s < t)$ only. Then

$$n\{E[V(\pi')] - E[V(\pi)]\}$$

$$= 2(\mu_{\pi_t} - \mu_{\pi_s}) \left[\sum_{r=s}^{t-1} \{E_r(\pi) - \bar{E}(\pi)\} + \frac{(t-s)(n-t+s)}{2n}(\mu_{\pi_t} - \mu_{\pi_s}) \right] - \frac{(t-s)(n-t-s+2)}{n}(\sigma_{\pi_t}^2 - \sigma_{\pi_s}^2). \quad (3.2.3)$$

Proof : Let $D = n\{E[V(\pi')] - E[V(\pi)]\}$. Using Lemma 3.2.1, we can write

$$D = n\{V_\mu(\pi') - V_\mu(\pi)\} + \sum_{r=1}^{n} \frac{(r-1)(n-r+1)}{n}(\sigma_{\pi'_r}^2 - \sigma_{\pi_r}^2). \quad (3.2.4)$$

where $V_\mu(\pi')$ $(V_\mu(\pi))$ is the variance of expected job completion times for the sequence π' (π).

We have

$$\begin{aligned} E_r(\pi') &= E_r(\pi) \quad \text{for } r = 1, \ldots, s-1, t, t+1, \ldots, n && (3.2.5) \\ \text{and } E_r(\pi') &= E_r(\pi) + (\mu_{\pi_t} - \mu_{\pi_s}) \quad \text{for } r = s, s+1, \ldots, t-1 && (3.2.6) \end{aligned}$$

and therefore

$$\bar{E}(\pi') = \bar{E}(\pi) + \frac{t-s}{n}(\mu_{\pi_t} - \mu_{\pi_s}). \qquad (3.2.7)$$

Now, we can write

$$\begin{aligned} n\left[V_\mu(\pi') - V_\mu(\pi)\right] &= \sum_{r=s}^{t-1}\left[E_r^2(\pi') - E_r^2(\pi)\right] - n\left[\bar{E}^2(\pi') - \bar{E}^2(\pi)\right] \\ &= (\mu_{\pi_t} - \mu_{\pi_s})\sum_{r=s}^{t-1}\left[2E_r(\pi) + (\mu_{\pi_t} - \mu_{\pi_s})\right] \\ &\quad -(t-s)(\mu_{\pi_t} - \mu_{\pi_s})\left[2\bar{E}(\pi) + \frac{t-s}{n}(\mu_{\pi_t} - \mu_{\pi_s})\right] \\ &\quad \text{(using the equations (3.2.5), (3.2.6) and (3.2.7)).} \end{aligned}$$

By simplifying the above terms of the right-hand-side, it can be seen that $n[V_\mu(\pi') - V_\mu(\pi)]$ is same as the first term on right-hand-side of the equation (3.2.3). Since $\sigma^2_{\pi'_r} = \sigma^2_{\pi_r}$ for all r except $r = s$ and t, it can be easily verified that the second term on right-hand-side of the equation (3.2.4) is same as that of the equation (3.2.3).

Hence the Lemma holds. ■

3.3 PROPERTIES OF OPTIMAL SEQUENCE

In this section, we derive first sufficient conditions for an optimal sequence to be V-shaped for general random processing times. Next, we deal with two special cases : one based on quadratic relation between mean and variance of processing times, and another one involving an ordered property of the jobs.

3.3.1 Results With General Processing Times

Chakravarthy [1986] and Vani and Raghavachari [1987] have derived, under some assumptions, sufficient conditions for the existence of an optimal sequence which is V-shaped in mean.

We now obtain sufficient conditions for the same under a general assumption on the job processing times, that is, they can have any arbitrary means and variances.

If $\mu_1 = \mu_2 = \cdots = \mu_n$, that is, all the jobs have the mean processing time, it is easily seen from Lemma 3.2.1 that optimal sequence is V-shaped in variance (also see Chakravarthy [1986]. In fact, using a result (page 261) of Hardy, Littlewood and Polya [1952], one can verify that an optimal sequence is

$$(\tau_1, \tau_3, \tau_5, \ldots, \tau_{n-3}, \tau_{n-1}, \tau_n, \tau_{n-2}, \ldots, \tau_4, \tau_2) \quad \text{if } n \text{ is even,}$$

$$(\tau_1, \tau_3, \tau_5, \ldots, \tau_{n-2}, \tau_n, \tau_{n-1}\tau_{n-3}, \ldots, \tau_4, \tau_2) \quad \text{if } n \text{ is odd}$$

where $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_n)$ is a sequence such that $\sigma^2_{\tau_1} \geq \sigma^2_{\tau_2} \geq \cdots \geq \sigma^2_{\tau_n}$.

Hence, we assume that the values of all the μ_j's are not same, and define

$$\Delta = \max\left\{0, \max_{\mu_i \neq \mu_j} \frac{\sigma_i^2 - \sigma_j^2}{\mu_i - \mu_j}\right\}. \tag{3.3.8}$$

Theorem 3.3.1 : If for any three jobs r, s and t with $\mu_r > \max\{\mu_s, \mu_t\}$, the condition

$$2\mu_r + (n-1)(\mu_s + \mu_t) > 2(n-2)\Delta \tag{3.3.9}$$

holds, any optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ satisfies

(i) V-shaped property in mean, and

(ii) if $\mu_{\pi_i} = \mu_{\pi_{i+1}}$ then $\sigma^2_{\pi_i} \geq \sigma^2_{\pi_{i+1}}$ ($\sigma^2_{\pi_i} \leq \sigma^2_{\pi_{i+1}}$) for $i \leq \frac{n}{2}$ ($i \geq \frac{n}{2} + 1$).

Proof : Suppose the condition (3.3.9) holds for any three jobs r, s and t. Let $\pi = (\pi_1, \ldots, \pi_n)$ be an optimal sequence which is not V-shaped in mean. Then, there exist three successive jobs π_i, π_{i+1}, π_{i+2} such that $\mu_{\pi_{i+1}} > \max\{\mu_{\pi_i}, \mu_{\pi_{i+2}}\}$. Let $\pi^{(1)}$ ($\pi^{(2)}$) be a sequence obtained from π by interchanging the two jobs π_i and π_{i+1} (π_{i+1} and π_{i+2}).

Let $D^{(1)} = n\{E[V(\pi^{(1)})] - E[V(\pi)]\}$ and $D^{(2)} = n\{E[V(\pi^{(2)})] - E[V(\pi)]\}$. The values of $D^{(1)}$ and $D^{(2)}$ are nonnegative since π is optimal. By Lemma 3.2.7, we have

$$\begin{aligned} D^{(1)} &= 2(\mu_{\pi_{i+1}} - \mu_{\pi_i})\left[E_i(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_{i+1}} - \mu_{\pi_i})\right] \\ &\quad - \frac{(n-2i+1)}{n}(\sigma^2_{\pi_{i+1}} - \sigma^2_{\pi_i}) \\ \text{and } D^{(2)} &= 2(\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\left[E_{i+1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\right] \\ &\quad - \frac{n-2i-1)}{n}(\sigma^2_{\pi_{i+2}} - \sigma^2_{\pi_{i+1}}). \end{aligned}$$

Let

$$Q = D^{(2)}/\{2(\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\} - D^{(1)}/\{2(\mu_{\pi_{i+1}} - \mu_{\pi_i})\}. \tag{3.3.10}$$

Then, we can write

$$\begin{aligned} Q &= \mu_{\pi_{i+1}} + \frac{n-1}{2n}(\mu_{\pi_i} + \mu_{\pi_{i+2}} - 2\mu_{\pi_{i+1}}) \\ &\quad - \frac{n-2i-1}{2n}\{(\sigma^2_{\pi_{i+2}} - \sigma^2_{\pi_{i+1}}) \,/\, (\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\} \\ &\quad + \frac{n-2i+1}{2n}\{(\sigma^2_{\pi_{i+1}} - \sigma^2_{\pi_i}) \,/\, (\mu_{\pi_{i+1}} - \mu_{\pi_i})\} \\ \text{i.e., } 2nQ &= 2\mu_{\pi_{i+1}} + (n-1)(\mu_{\pi_i} + \mu_{\pi_{i+2}}) \\ &\quad -(n-2i-1)\{(\sigma^2_{\pi_{i+2}} - \sigma^2_{\pi_{i+1}}) \,/\, (\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\} \\ &\quad +(n-2i+1)\{(\sigma^2_{\pi_{i+1}} - \sigma^2_{\pi_i}) \,/\, (\mu_{\pi_{i+1}} - \mu_{\pi_i})\} \\ &\geq 2\mu_{\pi_{i+1}} + (n-1)(\mu_{\pi_i} + \mu_{\pi_{i+2}}) - (n-3)\Delta - (n-1)\Delta \\ &\qquad (\text{ since } 1 \leq i \leq n-2 \text{ }) \\ &= 2\mu_{\pi_{i+1}} + (n-1)(\mu_{\pi_i} + \mu_{\pi_{i+2}}) - 2(n-2)\Delta. \end{aligned}$$

Using inequality (3.3.9), we can see that $Q > 0$, that is,

$$\begin{aligned} D^{(2)}/\{2(\mu_{\pi_{i+2}} - \mu_{\pi_{i+1}})\} &> D^{(1)}/\{2(\mu_{\pi_{i+1}} - \mu_{\pi_i})\} \\ &\geq 0 \end{aligned}$$

since $D^{(1)} \geq 0$. It means $D^{(2)} < 0$ which leads to the contradiction that π is not optimal. Therefore the optimal sequence π is V-shaped.

Next, it can be seen from second term on the right-hand-side of the equation (3.2.1) that any optimal sequence π must have the property (*ii*).

Hence the theorem holds. ■

Remark 3.3.2 : From the condition (3.3.9), it can be seen that if the variances (σ_j^2's) are homogeneous or small when compared to the means (μ_j's), then optimal sequence is more likely to be V-shaped in mean. However, if the variances (σ_j^2's) are very large in relation to the means (μ_j's), optimal sequence tends to be V-shaped in variance (refer to Lemma 3.2.1).

The following result shows that when the coefficient of variation of each job processing time does not exceed a limit determined by the means of processing times, any optimal sequence is V-shaped in mean.

Theorem 3.3.3 : An optimal sequence is V-shaped in mean, if

$$\sigma_j/\mu_j \in \left[0, \sqrt{\mu_n/\{(\mu_1+\mu_2)+\gamma\mu_1^2\}}\right] \tag{3.3.11}$$

for all j where $\gamma^{-1} = \min_{\mu_i \neq \mu_k} |\mu_i - \mu_k|$.

Proof : Let $c = \max_j \sigma_j/\mu_j$ and $\sigma_i^2 = c^2\mu_i^2 - \epsilon_i$, $\epsilon_i \geq 0$ for $i = 1, 2, \ldots, n$. For any i and j with $\mu_i \neq \mu_j$, we have

$$\frac{\sigma_i^2 - \sigma_j^2}{\mu_i - \mu_j} = \frac{c^2(\mu_i^2 - \mu_j^2) + (\epsilon_j - \epsilon_i)}{\mu_i - \mu_j}$$

$$
\begin{aligned}
&= c^2(\mu_i + \mu_j) + \left| \frac{\epsilon_j - \epsilon_i}{\mu_i - \mu_j} \right| \\
&\leq c^2(\mu_1 + \mu_2) + \gamma \max_{1 \leq k \leq n} \epsilon_k \\
&\leq c^2(\mu_1 + \mu_2) + c^2 \gamma \mu_1^2.
\end{aligned}
$$

Thus

$$
\begin{aligned}
\Delta &\leq c^2 \left[(\mu_1 + \mu_2) + \gamma \mu_1^2 \right] \\
&\leq \mu_n
\end{aligned}
$$

due to (3.3.11).

For any r, s and t, we have

$$
\begin{aligned}
& 2\mu_r + (n-1)(\mu_s + \mu_t) \\
\geq\ & 2n\mu_n \\
>\ & 2(n-2)\Delta.
\end{aligned}
$$

Now the result follows from Theorem 3.3.1. ■

In view of the fact that the general problem is NP-hard, it also becomes important to study special cases, develop efficient exact and approximate algorithms. From Theorem's 3.3.1 and 3.3.3, it is clear that complete characterization of optimal sequence looks very difficult, unless suitable assumption is made on the underlying random processing times. It may be recalled that expected CTV depends only on the first two moments of any job processing time (refer to Lemma 3.2.1). And, because of this reason, Chakravarthy [1986] and Vani and Raghavachari [1987] have considered special cases with different structures involving the first two moments only.

In the following, we discuss two special cases on the structural properties of optimal sequence.

3.3.2 Special Case I : Quadratic Relation Between Mean and Variance

Vani and Raghavachari [1987] have considered a special case based on a quadratic relation between mean and variance of each processing time. They have assumed that $g_j = A\mu_j^2 + B\mu_j$, $1 \leq j \leq n$, for fixed non-negative values of A and B, where g_j is the second (raw) moment of P_j, and observed that quite a few standard probability distributions have this property. For example, the distributions (i) Uniform (with interval starting from zero), (ii) Gamma (with fixed shape parameter), (iii) Chi-square, (iv) Poisson and (v) Binomial (with $A = (n-1)/n$ and $B = 1$) satisfy this property. They have proved that optimal sequence is V-shaped in mean if

$$\delta_i = \frac{A(n+1) - 2(A-1)i - 2}{2n} \geq 0, \quad i = 1, 2, \ldots, n-2 \qquad (3.3.12)$$

$$\alpha_j = \frac{(2-A)n + 2(A-1)j - A}{2n} \geq 0, \quad j = 2, 3, \ldots, n-1 \qquad (3.3.13)$$

In order to know that there exists an optimal sequence which is V-shaped in mean, we need to verify $2n - 4$ constraints.

We simplify this result and show that these constraints are either redundant or can be weakened depending upon the value of the parameter A.

Theorem 3.3.4 :

(i) For $0 \leq A \leq 2$ and $B \geq 0$, optimal sequence is V-shaped in mean.

(ii) For $A > 2$ and $B \geq 0$, optimal sequence is V-shaped in mean when $n < \frac{5(A-1)}{A-2}$.

Proof : Suppose there is an optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ that is not V-shaped in mean. Then there exists three successive jobs in π, say, π_i, π_{i+1} and π_{i+2} such that $\mu_{\pi_{i+1}} > \max\{\mu_{\pi_i}, \mu_{\pi_{i+2}}\}$.

We show that by interchanging π_{i+1} with either of the other two jobs yields a better sequence than π.

We use the same arguments as in the proof of Theorem 3.3.1. Here we can write

$$\begin{aligned} Q &= \delta_i \mu_{\pi_i} + \frac{A}{n}\mu_{\pi_{i+1}} + \alpha_{i+1}\mu_{\pi_{i+2}} + \frac{B}{n} \\ &= \delta_i \mu_{\pi_i} + \left(\frac{1}{2n} + \frac{2A-1}{2n}\right)\mu_{\pi_{i+1}} + \alpha_{i+1}\mu_{\pi_{i+2}} + \frac{B}{n} \end{aligned}$$

where α_i and δ_j are given in the systems (3.3.12) and (3.3.13).

Since $\mu_{\pi_{i+1}} > \max\{\mu_{\pi_i}, \mu_{\pi_{i+2}}\}$, we can write

$$\begin{aligned} Q &> \left(\delta_i + \frac{1}{2n}\right)\mu_{\pi_i} + \left(\alpha_{i+1} + \frac{2A-1}{2n}\right)\mu_{\pi_{i+2}} + \frac{B}{n} \\ &= \delta_i^* \mu_{\pi_i} + \alpha_{i+1}^* \mu_{\pi_{i+2}} + \frac{B}{n} \end{aligned}$$

where $\delta_i^* = \delta_i + \frac{1}{2n}$ and $\alpha_{i+1}^* = \alpha_{i+1} + \frac{2A-1}{2n}$. It is obvious that $\delta_i^* = \alpha_{n-i}^*$, for $i = 1, 2, \ldots, n-2$. We observe that

$$\begin{array}{ccccccc} \delta_{n-2}^* & \geq & \cdots & \geq & \delta_1^* & = & \frac{1}{2n}[A(n-1)+1] \\ \alpha_2^* & \geq & \cdots & \geq & \alpha_{n-1}^* & = & \delta_1^* \end{array}$$

for $0 \leq A \leq 1$ and

$$\begin{array}{ccccccc} \delta_1^* & \geq & \cdots & \geq & \delta_{n-2}^* & = & \frac{1}{2n}[(2-A)n + 5(A-1)] \\ \alpha_{n-1}^* & \geq & \cdots & \geq & \alpha_2^* & = & \delta_{n-2}^* \end{array}$$

for $A > 1$.

It implies that $Q > 0$ for $0 \leq A \leq 2$. Moreover, $Q > 0$ for $A > 2$ provided $n < \frac{5(A-1)}{A-2}$.

Hence the theorem holds. ■

3.3.3 Special Case II : Ordered Processing Times

In real-life situations, job processing time having larger mean is generally expected to have larger variance also. For this reason, we now assume that

$$\mu_r < \mu_s \quad \Rightarrow \quad \sigma_r^2 < \sigma_s^2 \quad \text{for any } r \text{ and } s. \tag{3.3.14}$$

The processing times are said to be *ordered* if the condition (3.3.14) is satisfied.

There may not exist a V-shaped optimal sequence even if the processing times are ordered. For instance, consider the following numerical example of nine-job problem with ordered processing times.

Job (j)	1	2	3	4	5	6	7	8	9
μ_j	200	200	7	6	5	4	2	2	1
σ_j	50	50	42	41	40	36	35	3	2

The best among V-shaped sequences is (1, 3, 4, 5, 8, 9, 7, 6, 2) which gives 6020.9136 as expected CTV (variance of completion times). However, the sequence (1, 3, 4, 5, 8, 9, 6, 7, 2) is the best among all 9! sequences but not V-shaped. It gives 6017.4568 as the minimum expected CTV.

Theorem 3.3.5 : There exists an optimal sequence of the form $(1, \ldots, 2)$.

Proof : It can be seen from Theorem 3.2.6 that there exists an optimal sequence of the form $(1, \ldots)$. Suppose $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is an optimal sequence with $\pi_1 = 1$. Let $\pi_n \neq 2$. Then $\pi_k = 2$ for some $2 \leq k \leq n-1$.

For $k = 2$, we can see by Theorem 3.2.4 that the sequence $(\pi_1, \pi_n, \pi_{n-1}, \ldots, \pi_3, 2)$ is optimal.

Let $3 \leq k \leq n-1$.

Suppose $\mu_2 > \max\{\mu_{\pi_2}, \mu_{\pi_n}\}$. Obtain $\pi^{(1)}$ ($\pi^{(2)}$) from π by interchanging only two jobs π_2 and π_k (π_k and π_n). Let $D^{(1)} = n\{E[V(\pi^{(1)})] - E[V(\pi)]\}$ and

$D^{(2)} = n\{E[V(\pi^{(2)})] - E[V(\pi)]\}$. Then by Lemma 3.2.7, we have

$$\begin{aligned} D^{(1)} &= 2(\mu_2 - \mu_{\pi_2})\, X - \frac{(k-2)(n-k)}{n}(\sigma_2^2 - \sigma_{\pi_2}^2), \\ D^{(2)} &= 2(\mu_{\pi_n} - \mu_2)\, Y - \frac{(k-2)(n-k)}{n}(\sigma_2^2 - \sigma_{\pi_n}^2) \end{aligned}$$

where

$$\begin{aligned} X &= \sum_{r=2}^{k-1} \left[E_r(\pi) - \bar{E}(\pi)\right] + \frac{(k-2)(n-k+2)}{2n}(\mu_2 - \mu_{\pi_2}), \\ Y &= \sum_{r=k}^{n-1} \left[E_r(\pi) - \bar{E}(\pi)\right] + \frac{k(n-k)}{2n}(\mu_{\pi_n} - \mu_2). \end{aligned}$$

Since, $\mu_2 > \max\{\mu_{\pi_2}, \mu_{\pi_n}\}$, we have $\sigma_2^2 > \max\{\sigma_{\pi_2}^2, \sigma_{\pi_n}^2\}$. We also have $D^{(1)} \geq 0$, $D^{(2)} \geq 0$ as π is optimal. It now follows that $Y < 0 < X$.

If $\bar{E}(\pi) \leq E_{k-1}(\pi)$, then

$$\begin{aligned} Y &\geq (n-k)\mu_2 + \frac{k(n-k)}{2n}(\mu_{\pi_n} - \mu_2) \\ &= (n-k)\left[1 - \frac{k}{2n}\right]\mu_2 + \frac{k(n-k)}{2n}\mu_{\pi_n} \\ &> 0 \end{aligned}$$

which is a contradiction.

Similarly, if we assume $\bar{E}(\pi) \geq E_k(\pi)$, we again arrive at a contradiction that $X < 0$.

Therefore $E_{k-1}(\pi) < \bar{E}(\pi) < E_k(\pi)$. Let $\bar{E}(\pi) = E_k(\pi) - \epsilon$, for some $\epsilon > 0$. Now, we have

$$Y \geq (n-k)\left[\epsilon - \frac{k}{2n}\mu_2\right] \quad \text{and} \quad X \leq (k-2)\left[\epsilon - \frac{n+k-2}{2n}\mu_2\right].$$

Then, $Y < 0 \Rightarrow \epsilon < \frac{k}{2n}\mu_2 \Rightarrow X < 0$ which is a contradiction.

Therefore, if π is optimal with $\pi_k = 2$ and $3 \leq k \leq n-1$, then $\mu_2 \not> \max\{\mu_{\pi_2}, \mu_{\pi_n}\}$.

Suppose $\mu_{\pi_2} = \mu_{\pi_k}$. If jobs π_2 and π_k are interchanged, the value of first term on the right-hand-side in the equation (3.2.1) remains same whereas the second term does not increase. That is, interchange of jobs π_2 and π_k does not increase expected variance. Now it follows from Theorem 3.2.4 that $(\pi_1, \pi_n, \pi_{n-1}, \ldots, \pi_{k+1}, \pi_2, \pi_{k-1}, \ldots, \pi_3, 2)$ is at least as good as π.

Similarly, if $\mu_{\pi_n} = \mu_{\pi_k}$, we can show that $(\pi_1, \pi_2, \ldots, \pi_{k-1}, \pi_n, \pi_{k+1}, \ldots, \pi_{n-1}, 2)$ is at least as good as π. Hence the theorem holds. ∎

In the following, we present some results dealing with monotonic property of optimal sequence.

Lemma 3.3.6 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence. Then

(i) for any g, $2 \leq g \leq n$,

$$\mu_{\pi_g} < \mu_{\pi_{g-1}} \Rightarrow \bar{E}(\pi) \geq E_{g-1}(\pi) + \frac{n-1}{2n}(\mu_{\pi_g} - \mu_{\pi_{g-1}}) - \frac{n-2g+3}{2n} \cdot \frac{\sigma^2_{\pi_g} - \sigma^2_{\pi_{g-1}}}{\mu_{\pi_g} - \mu_{\pi_{g-1}}}, \qquad (3.3.15)$$

(ii) for any h, $1 \leq h < n$,

$$\mu_{\pi_h} < \mu_{\pi_{h+1}} \Rightarrow \bar{E}(\pi) \leq E_h(\pi) + \frac{n-1}{2n}(\mu_{\pi_{h+1}} - \mu_{\pi_h}) - \frac{n-2h+1}{2n} \cdot \frac{\sigma_{\pi_{h+1}}2 - \sigma^2_{\pi_h}}{\mu_{\pi_{h+1}} - \mu_{\pi_h}}. \qquad (3.3.16)$$

Proof : Suppose $\mu_{\pi_g} < \mu_{\pi_{g-1}}$ for some g, $2 \leq g \leq n$. Obtain the sequence π' from π by interchanging the two jobs π_{g-1} and π_g only. Using Lemma 3.2.7, we have

$$\begin{aligned} &\frac{n}{2}\{E[V(\pi')] - E[V(\pi)]\} \\ = \;& (\mu_{\pi_g} - \mu_{\pi_{g-1}}) \left[E_{g-1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_g} - \mu_{\pi_{g-1}})\right] \\ & - \frac{n-2g+3}{2n}(\sigma^2_{\pi_g} - \sigma^2_{\pi_{g-1}}). \end{aligned} \qquad (3.3.17)$$

Since π is optimal, $E[V(\pi')] - E[V(\pi)] \geq 0$. Therefore, using the equation (3.3.17), we get

$$E_{g-1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_g} - \mu_{\pi_{g-1}}) - \frac{n-2g+3}{2n} \cdot \frac{\sigma^2_{\pi_g} - \sigma^2_{\pi_{g-1}}}{\mu_{\pi_g} - \mu_{\pi_{g-1}}} \leq 0$$

$$\text{or, } \bar{E}(\pi) \geq E_{g-1}(\pi) + \frac{n-1}{2n}(\mu_{\pi_g} - \mu_{\pi_{g-1}}) - \frac{n-2g+3}{2n} \cdot \frac{\sigma^2_{\pi_g} - \sigma^2_{\pi_{g-1}}}{\mu_{\pi_g} - \mu_{\pi_{g-1}}}.$$

Hence the part (i) of the lemma holds.

The part (ii) can be proved by similar arguments. ∎

Theorem 3.3.7 : If $\mu_{\pi_h} < \mu_{\pi_{h+1}}$ for some $1 \leq h \leq \lfloor\frac{n+1}{2}\rfloor$ in an optimal sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, then

$$\mu_{\pi_{\lfloor\frac{n+1}{2}\rfloor}} \leq \mu_{\pi_{\lfloor\frac{n+1}{2}\rfloor+1}} \leq \ldots \leq \mu_{\pi_n}. \tag{3.3.18}$$

Proof : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence with $\mu_{\pi_h} < \mu_{\pi_{h+1}}$ for some $1 \leq h \leq \lfloor\frac{n+1}{2}\rfloor$. Assume, if possible, the inequality (3.3.18) does not hold good for π. Then there exists two adjacent jobs in π, say, π_{g-1} and π_g, $\lfloor\frac{n+1}{2}\rfloor + 1 \leq g \leq n$, such that $\mu_{\pi_{g-1}} > \mu_{\pi_g}$. Therefore, by Lemma 3.3.6(i), we have

$$\begin{aligned}\bar{E}(\pi) \;\geq\; & E_{g-1}(\pi) + \frac{n-1}{2n}(\mu_{\pi_g} - \mu_{\pi_{g-1}}) \\ & - \frac{n-2g+3}{2n} \cdot \frac{\sigma^2_{\pi_g} - \sigma^2_{\pi_{g-1}}}{\mu_{\pi_g} - \mu_{\pi_{g-1}}}.\end{aligned} \tag{3.3.19}$$

Because $\mu_{\pi_h} < \mu_{\pi_{h+1}}$, we also get from Lemma 3.3.6(ii),

$$\begin{aligned}\bar{E}(\pi) \;\leq\; & E_h(\pi) + \frac{n-1}{2n}(\mu_{\pi_{h+1}} - \mu_{\pi_h}) \\ & - \frac{n-2h+1}{2n} \cdot \frac{\sigma_{\pi_{h+1}}2 - \sigma^2_{\pi_h}}{\mu_{\pi_{h+1}} - \mu_{\pi_h}}.\end{aligned} \tag{3.3.20}$$

Combining the inequalities (3.3.19) and (3.3.20),

$$[E_h(\pi) - E_{g-1}(\pi)] + \frac{n-1}{2n}\left[\mu_{\pi_{h+1}} - \mu_{\pi_h} - \mu_{\pi_g} + \mu_{\pi_{g-1}}\right]$$

$$-\frac{n-2h+1}{2n}\cdot\frac{\sigma^2_{\pi_{h+1}}-\sigma^2_{\pi_h}}{\mu_{\pi_{h+1}}-\mu_{\pi_h}}+\frac{n-2g+3}{2n}\cdot\frac{\sigma^2_{\pi_g}-\sigma^2_{\pi_{g-1}}}{\mu_{\pi_g}-\mu_{\pi_{g-1}}}\geq 0 \quad (3.3.21)$$

$$\text{or, } -\left[\mu_{\pi_{h+1}}+\cdots+\mu_{\pi_{g-1}}\right]+\frac{n-1}{2n}\left[\mu_{\pi_{h+1}}+\mu_{\pi_{g-1}}-(\mu_{\pi_h}+\mu_{\pi_g})\right]$$

$$-\frac{n-2h+1}{2n}\cdot\frac{\sigma^2_{\pi_{h+1}}-\sigma^2_{\pi_h}}{\mu_{\pi_{h+1}}-\mu_{\pi_h}}+\frac{n-2g+3}{2n}\cdot\frac{\sigma^2_{\pi_g}-\sigma^2_{\pi_{g-1}}}{\mu_{\pi_g}-\mu_{\pi_{g-1}}}\geq 0. \quad (3.3.22)$$

Since $1 \leq h \leq \lfloor\frac{n+1}{2}\rfloor$ and $\lfloor\frac{n+1}{2}\rfloor + 1 \leq g \leq n$, we note that

$$(i) \quad \frac{n-2h+1}{2n}\cdot\frac{\sigma^2_{\pi_{h+1}}-\sigma^2_{\pi_h}}{\mu_{\pi_{h+1}}-\mu_{\pi_h}}\geq 0,$$

$$(ii) \quad \frac{n-2g+3}{2n}\cdot\frac{\sigma^2_{\pi_g}-\sigma^2_{\pi_{g-1}}}{\mu_{\pi_g}-\mu_{\pi_{g-1}}}\leq 0.$$

Therefore, the left-hand-side of the inequality (3.3.22) is strictly negative, which is a contradiction.

Hence the theorem holds. ∎

Corollary 3.3.8 : If $\mu_{\pi_{g-1}} > \mu_{\pi_g}$, for some $\lfloor\frac{n+1}{2}\rfloor + 1 \leq g \leq n$ in an optimal sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, then

$$\mu_{\pi_2} \geq \mu_{\pi_3} \geq \ldots \geq \mu_{\pi_{\lfloor\frac{n+1}{2}\rfloor+1}}. \quad (3.3.23)$$

Proof : It follows from Theorems 3.3.7 and 3.2.4. ∎

Theorem 3.3.9 : In every optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$, (i) the first job is the largest job, and (ii) *either* $\mu_{\pi_2} \geq \ldots \geq \mu_{\pi_{\lfloor\frac{n+1}{2}\rfloor+1}}$ *or* $\mu_{\pi_{\lfloor\frac{n+1}{2}\rfloor}} \leq \ldots \leq \mu_{\pi_n}$.

Proof : Using Theorem 3.3.7 and Corollary 3.3.8, it can be easily seen that for any optimal sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$, the relation (ii) holds.

Suppose the job π_1 is not the largest, that is,

$$\text{either} \quad \mu_1 > \mu_{\pi_1} \quad (\Rightarrow \sigma_1^2 > \sigma^2_{\pi_1}),$$

$$\text{or} \quad \mu_1 = \mu_{\pi_1} \text{ and } \sigma_1^2 > \sigma^2_{\pi_1}.$$

Let π_k be the largest job for some $2 \leq k \leq n$.

Case (a) $k = 2$: It can be easily observed (using Lemma 3.2.7) that the sequence, obtained from π by interchanging π_1 and π_2, is strictly better than π. This is a contradiction to the optimality of π.

Case (b) $k = n$: Consider π^D, the dual of π. We know from Theorem 3.2.4 that π^D is as good as π. Note that the largest job in π^D lies at the second position and hence the Case (a) applies to π^D.

Case (c) $3 \leq k \leq n-1$: Let $\pi^{(1)}$ ($\pi^{(2)}$) be the sequences obtained from π by interchanging the jobs π_2 and π_k (π_k and π_n) only. We can use the similar arguments as in the proof of Theorem 3.3.5 and show that either $\pi^{(1)}$ or $\pi^{(2)}$ is at least as good as π. We note that Case (a) (Case (b)) is applicable to $\pi^{(1)}$ ($\pi^{(2)}$), and hence, we once again arrive at a contradiction.

This completes the proof of the theorem. ∎

Corollary 3.3.10 : There exists an optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ which satisfies (i) the first job is the largest job, and (ii) $\mu_{\pi_2} \geq \ldots \geq \mu_{\pi_{\lfloor \frac{n+1}{2} \rfloor + 1}}$.

Proof : It follows immediately from Theorems 3.3.9 and 3.2.4.

Theorem 3.3.11 : There exists an optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ satisfying the following :

(i) the first job is the largest job,

(ii) $\mu_{\pi_2} \geq \ldots \geq \mu_{\pi_{\lfloor \frac{n+1}{2} \rfloor + 1}}$, and

(iii) $\mu_{\pi_{n-1}} \leq \mu_{\pi_n}$.

Proof : We know from Corollary 3.3.10 that $\exists$ an optimal sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ satisfying (i) and (ii).

We shall now show that $\mu_{\pi_n} \geq \mu_{\pi_{n-1}}$.

Suppose $\mu_{\pi_n} = \mu_2$. Because of (i), it follows that $\mu_{\pi_n} \geq \mu_{\pi_{n-1}}$.

Let $\mu_{\pi_n} < \mu_2$. Using the arguments in the proof of Theorem 3.3.5, we can show that $\mu_{\pi_2} = \mu_2$. Now, obtain a sequence π' from π by interchanging the jobs π_{n-1} and π_n. By Lemma 3.2.7, we have

$$\begin{aligned}
&\frac{n}{2}\{E[V(\pi')] - E[V(\pi)]\} \\
&= (\mu_{\pi_n} - \mu_{\pi_{n-1}})\left[E_{n-1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_n} - \mu_{\pi_{n-1}})\right] \\
&\quad + \frac{n-3}{2n}\cdot(\sigma^2_{\pi_n} - \sigma^2_{\pi_{n-1}}). \qquad (3.3.24)
\end{aligned}$$

Since (i) holds and $\mu_{\pi_2} = \mu_2$, it can be easily seen that

$$\left[E_{n-1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_n} - \mu_{\pi_{n-1}})\right] \;>\; 0. \qquad (3.3.25)$$

If $\mu_{\pi_n} < \mu_{\pi_{n-1}}$, because the *ordered* property (3.3.14), we have $\sigma^2_{\pi_n} < \sigma^2_{\pi_{n-1}}$. It then follows from the equation (3.3.24) using the inequality (3.3.25) that $E[V(\pi')] < E[V(\pi)]$, which is a contradiction to the optimality of π. Therefore, $\mu_{\pi_n} \geq \mu_{\pi_{n-1}}$.

Hence the theorem holds. ∎

Remark 3.3.12 : Using the above results, to obtain an optimal sequence, we may confine to $\binom{n-1}{k_0}\binom{n-k_0-1}{2}(n-k_0-3)!$ sequences only, where $k_0 = \lfloor\frac{n+1}{2}\rfloor$.

The following theorem restricts the position of job n in any optimal sequence.

Theorem 3.3.13 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be any optimal sequence with $(\pi_1, \pi_n) = (1, 2)$. Then $\mu_{\pi_2} \geq \mu_{\pi_3} \geq \mu_{\pi_4}$ for $n > 5$.

Proof : Suppose $\mu_{\pi_4} > \mu_{\pi_3}$. Consider the sequence π' obtained from π by interchanging the jobs π_3 and π_4. By Lemma 3.2.7, we have

$$\begin{aligned} n\{E[V(\pi')] - E[V(\pi)]\} &= 2(\mu_{\pi_4} - \mu_{\pi_3})\left[E_3(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_4} - \mu_{\pi_3})\right] \\ &\quad - \frac{(n-5)}{n}(\sigma^2_{\pi_4} - \sigma^2_{\pi_3}). \end{aligned} \quad (3.3.26)$$

Note that

$$\begin{aligned} & E_3(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_4} - \mu_{\pi_3}) \\ = & \frac{1}{n}\left[(\mu_{\pi_2} - \mu_2) - \left\{\frac{n-5}{2}(\mu_{\pi_3} + \mu_{\pi_4}) + \sum_{r=5}^{n-1}(n-r+1)\mu_{\pi_r}\right\}\right] \\ < & 0. \end{aligned}$$

Now, from the equation (3.3.26) it is obvious that $E[V(\pi')] < E[V(\pi)]$, that is, π' is better than π which contradicts the optimality of π. Hence $\mu_{\pi_3} \geq \mu_{\pi_4}$.

We can argue similarly that $\mu_{\pi_2} \geq \mu_{\pi_3}$. ∎

Corollary 3.3.14 : There exists an optimal sequence of the form $(1, \ldots, 2)$ with job n in one of the positions $4, 5, \ldots, n-1$.

Corollary 3.3.15 : If $\pi = (\pi_1, \ldots, \pi_n)$ is an optimal sequence with $(\pi_1, \pi_2) = (1, 2)$, then $\mu_{\pi_n} \geq \mu_{\pi_{n-1}} \geq \mu_{\pi_{n-2}}$ for $n > 5$.

Proof : This follows from Theorems 3.3.13 and 3.2.4. ∎

We now present some results on the V-shaped property for some small size problems.

Corollary 3.3.16 : For $n = 5$, the sequence $(1, 3, 4, 5, 2)$ is optimal.

Proof : This can proved following the arguments given in Theorems 3.3.5 and 3.2.4. ∎

Lemma 3.3.17 : In any optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$, $\mu_{\pi_k} \leq \max\{\mu_{\pi_{k-1}}, \mu_{\pi_{k+1}}\}$ for $k = \lfloor \frac{n}{2} \rfloor + 1$ when n is even and for $k = \lfloor \frac{n}{2} \rfloor + 1$ and $\lfloor \frac{n}{2} \rfloor + 2$ when n is odd.

Proof : Suppose $\mu_{\pi_k} > \max\{\mu_{\pi_{k-1}}, \mu_{\pi_{k+1}}\}$. Obtain $\pi^{(1)}$ ($\pi^{(2)}$) from π by interchanging the jobs π_{k-1} and π_k (π_k and π_{k+1}). We shall show that $\pi^{(1)}$ is better than π.

Let $D^{(i)} = n\{E[V(\pi^{(i)})] - E[V(\pi)]\}$, $i = 1,2$. From Lemma 3.2.7, we have

$$
\begin{aligned}
D^{(1)} &= 2(\mu_{\pi_k} - \mu_{\pi_{k-1}})\, X - \frac{n-2k+3}{n}(\sigma^2_{\pi_k} - \sigma^2_{\pi_{k-1}}), \\
D^{(2)} &= 2(\mu_{\pi_{k+1}} - \mu_{\pi_k})\, Y - \frac{n-2k+1}{n}(\sigma^2_{\pi_{k+1}} - \sigma^2_{\pi_k})
\end{aligned}
$$

where

$$
\begin{aligned}
X &= E_{k-1}(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_k} - \mu_{\pi_{k-1}}) \\
\text{and } Y &= E_k(\pi) - \bar{E}(\pi) + \frac{n-1}{2n}(\mu_{\pi_{k+1}} - \mu_{\pi_k}).
\end{aligned}
$$

It is obvious that $Y > X$.

Note that for $k = \lfloor \frac{n}{2} \rfloor + 1$ and $\lfloor \frac{n}{2} \rfloor + 2$ with odd n and $k = \lfloor \frac{n}{2} \rfloor + 1$ with even n, we have $n - 2k + 1 \leq 0 \leq n - 2k + 3$. Since π is optimal, we have $D^{(2)} \geq 0$, that is, $2(\mu_{\pi_{k+1}} - \mu_{\pi_k})\, Y \geq 0$ which implies $Y \leq 0$. As $X < Y$, it now follows that

$$
\begin{aligned}
D^{(1)} &< 2(\mu_{\pi_k} - \mu_{\pi_{k-1}})\, Y - \frac{n-2k+3}{n}(\sigma^2_{\pi_k} - \sigma^2_{\pi_{k-1}}) \\
&\leq 0
\end{aligned}
$$

since $n - 2k + 3 \geq 0$. It means that the sequence $\pi^{(1)}$ is better than π which contradicts the optimality of π. Hence the Lemma holds. ∎

Corollary 3.3.18 : For $n = 6$ and 7, there exists a V-shaped (in mean) optimal sequence of the form $(1, \ldots, 2)$.

Proof : If $n = 6$, it follows immediately from Theorem 3.3.13 that any optimal sequence of the form $(1, \ldots, 2)$ is V-shaped in mean. For $n = 7$, one can easily verify that any optimal sequence of the form $(1, \ldots, 2)$ is V-shaped in mean due to Theorem 3.3.13 and Lemma 3.3.17.

Remark 3.3.19 : Further, it can be seen from the result of Vani and Raghavachari [1987] concerning the position of the third largest job (for deterministic case) and Lemma 3.3.17 that for $n = 6$ and 7, there exists a V-shaped (in mean) optimal sequence of the form $(1, 3, \ldots, 2)$.

We now derive a sufficient condition for V-shapedness of optimal sequence for ordered processing times.

Theorem 3.3.20 : If for any three jobs r, s and t with $\mu_r > \max\{\mu_s, \mu_t\}$,

$$2\mu_r + (n-1)(\mu_s + \mu_t) > (n-5)\Delta \tag{3.3.27}$$

where Δ is as defined in (3.3.8), there exists an optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ which satisfies

(*i*) $\pi_1 = 1$,

(*ii*) V-shaped property in mean, and

(*iii*) if $\mu_{\pi_i} = \mu_{\pi_{i+1}}$ then $\sigma^2_{\pi_i} \geq \sigma^2_{\pi_{i+1}}$ ($\sigma^2_{\pi_i} \leq \sigma^2_{\pi_{i+1}}$) for $i \leq \frac{n}{2}$ ($i \geq \frac{n}{2} + 1$).

Proof : (*i*) follows directly from Theorem 3.2.6. (*ii*) and (*iii*) can be proved using the same arguments as in the proof of Theorem 3.3.1. ■

Remark 3.3.21 : Note that condition (3.3.27) is weaker than the condition (3.3.9).

3.4 LOWER BOUND, DOMINANCE CRITERION AND AN ALGORITHM

Here we first discuss on the derivation of lower bound for the expected CTV which is used in Subsection 3.4.3 to develop a branch-and-bound algorithm for the stochastic CTV problem with general job processing times. In Subsection 3.4.2, we derive a dominance rule for fathoming partial sequences in the proposed algorithm to reduce the computational effort required by the algorithm. Improved branching procedure for *ordered* processing times based on the results of Section 3.3.3 is also discussed.

For a partial or complete sequence α, let

$E_j(\alpha)$ = Expected completion time of job j for α (Note that $E_j(\alpha)$ is used in the earlier sections to denote the expected completion time of jth job in α),

$\bar{E}(\alpha)$ = Average of expected job completion times for α,

$V_\mu(\alpha)$ = Variance of the expected job completion times for α,

$E[V(\alpha)]$ = Expectation of the variance of job completion times for α,

$\bar{E}_{min}(\alpha)$ = Minimum average of the expected job completion times for the jobset α,

Also, for a complete sequence π of the form $\pi = (\alpha, \beta)$ $(\alpha, \beta \neq \phi)$, we define

$\bar{E}_\alpha(\pi) = \frac{1}{|\alpha|}\sum_{j\in\alpha} E_j(\pi)$ and $\bar{E}_\beta(\pi) = \frac{1}{|\beta|}\sum_{j\in\beta} E_j(\pi)$.

3.4.1 Lower Bound on Expected CTV

Derivation of lower bound plays an important role in the development of algorithm. In the deterministic CTV problem, lower bound on CTV has been studied by De, Ghosh and Wells [1992], Prasad, Manna and Arthanari [1994] and Mittenthal, Raghavachari and Rana [1994] and it is effectively used to devise approximate and exact algorithms. As done in the deterministic case, we derive lower bound on the minimum expected CTV and use it in the development of a branch-and-bound procedure.

Recall (refer to Lemma 3.2.1) that for any sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$,

$$\begin{aligned} E[V(\pi)] &= V_\mu(\pi) + \frac{1}{n}\sum_{r=1}^{n} w_r \sigma_{\pi_r}^2 \\ &= V_\mu(\pi) + \frac{1}{n} T(\sigma, \pi) \quad (\text{ say }) \end{aligned} \tag{3.4.28}$$

where $V_\mu(\pi)$ is the variance of expected job completion times for the sequence π and $w_r = \frac{1}{n}(r-1)(n-r+1)$ for $r = 1, 2, \ldots, n$.

In order to find a lower bound for the expected CTV, we treat the terms $V_\mu(\pi)$ and $T(\sigma, \pi)$ separately, that is, we derive lower bounds separately for $V_\mu(\pi)$ as well as $T(\sigma, \pi)$.

It can be seen from the structure of $T(\sigma, \pi)$ that the positional weights w_r's (coefficients of $\sigma_{\pi_r}^2$'s) are independent of the jobs at different positions. Therefore, one can easily obtain (using a result (page 261) of Hardy, Littlewood and Polya [1952]) a lower bound for $T(\sigma, \pi)$ by minimizing the same. Whereas, in order to derive a lower bound on $V_\mu(\pi)$, we use directly the results of the deterministic CTV problem and its lower bounding procedure (refer to Subsection 2.9.1 of Chapter 2) by simply replacing the fixed job processing times by the mean job processing times.

The lower bounds for $V_\mu(\pi)$ and $T(\sigma, \pi)$, so obtained, together gives a lower bound for the expected CTV, $E[V(\pi)]$.

Lemma 3.4.1 : For a sequence $\pi = (\alpha, \beta)$, with $|\alpha| = k$,

$$\begin{aligned} nV_\mu(\pi) &= kV_\mu(\alpha) + (n-k)V_\mu(\beta) \\ &\quad + \frac{k(n-k)}{n}\left[\bar{E}(\alpha) - \sum_{j\in\alpha}\mu_j - \bar{E}(\beta)\right]^2 . \end{aligned} \tag{3.4.29}$$

Proof : Using variance partition formula, we can write

$$nV_\mu(\pi) = \sum_{j\in\alpha}\left[E_j(\pi)-\bar{E}_\alpha(\pi)\right] - \sum_{j\in\beta}\left[E_j(\pi)-\bar{E}_\beta(\pi)\right]^2 + \frac{k(n-k)}{n}\left[\bar{E}_\alpha(\pi)-\bar{E}_\beta(\pi)\right]^2. \quad (3.4.30)$$

Observe that

$$E_i(\pi) = E_i(\alpha) \quad \text{for } i\in\alpha \quad (3.4.31)$$

$$= \sum_{j\in\alpha}\mu_j + E_i(\beta) \quad \text{for } i\in\beta. \quad (3.4.32)$$

Consequently,

$$\bar{E}_\alpha(\pi) = \bar{E}(\alpha) \quad (3.4.33)$$

$$\text{and } \bar{E}_\beta(\pi) = \sum_{j\in\alpha}\mu_j + \bar{E}(\beta). \quad (3.4.34)$$

Therefore, we get, using equations (3.4.31) to (3.4.34),

$$\sum_{j\in\alpha}\left[E_j(\pi)-\bar{E}_\alpha(\pi)\right]^2 = kV_\mu(\alpha) \quad (3.4.35)$$

$$\text{and } \sum_{j\in\beta}\left[E_j(\pi)-\bar{E}_\beta(\pi)\right]^2 = (n-k)V_\mu(\beta). \quad (3.4.36)$$

Finally, we get (3.4.29) by combining the equations (3.4.30), (3.4.33) to (3.4.36).

Hence the Lemma holds. ■

The following result gives a lower bound on the expected CTV for an arbitrary completion of a given partial sequence.

Lemma 3.4.2 : Let α be a given partial sequence with $|\alpha| = k$ and $\pi = (\alpha,\beta)$ be an arbitrary completion of α. Then

$$nE[V(\pi)] \geq kV_\mu(\alpha) + \sum_{r=1}^{k} w_r\sigma_{\pi_r}^2 + (n-k)\underline{V_\mu}(\beta)$$

$$
+\frac{k(n-k)}{n}\left[\bar{E}(\alpha)-\sum_{j\in\alpha}\mu_j-\bar{E}_{min}(\beta)\right]^2
$$
$$
+\sum_{r=k+1}^{n} w_r\sigma_{\tau_r}^2 \qquad (3.4.37)
$$

where $\pi =$ $(\pi_1, \pi_2, \ldots, \pi_n)$,

$\underline{V_\mu}(\beta) =$ a lower bound for the variance of expected completion times for the subproblem with jobset as β,

$(\tau_{k+1}, \ldots, \tau_n) =$ a permutation of jobs in β satisfying the property that $w_r > w_s \Rightarrow \sigma_{\tau_r}^2 \leq \sigma_{\tau_s}^2$ for all $r \neq s$ and $k+1 \leq r, s \leq n$.

Proof : For the sequence π, we have from (3.4.28),

$$
nE[V(\pi)] \;=\; nV_\mu(\pi) + T(\sigma, \pi). \qquad (3.4.38)
$$

Using Lemma 3.4.1 and following the definitions of $\underline{V_\mu}(\beta)$ and $\bar{E}_{min}(\beta)$, we get

$$
nV_\mu(\pi) \;\geq\; kV_\mu(\alpha) + (n-k)\underline{V_\mu}(\beta)
$$
$$
+\frac{k(n-k)}{n}\left[\bar{E}(\alpha)-\sum_{j\in\alpha}\mu_j-\bar{E}_{min}(\beta)\right]^2. \qquad (3.4.39)
$$

We also have,

$$
T(\sigma, \pi) \;=\; \sum_{r=1}^{k} w_r\sigma_{\pi_r}^2 + \sum_{r=k+1}^{n} w_r\sigma_{\pi_r}^{2}. \qquad (3.4.40)
$$

Note that for given α, the first term on the right-hand-side of the equation (3.4.40) is completely determined. The second term on the right-hand-side of the same equation involves the jobs in β only and is minimized (refer to page 261 of Hardy, Littlewood and Polya [1952]) by allocating the jobs as per $(\tau_{k+1}, \ldots, \tau_n)$ (a permutation of the jobs in β) satisfying the property that $w_r > w_s \Rightarrow \sigma_{\tau_r}^2 \leq \sigma_{\tau_s}^2$ for all $r \neq s$ and $k+1 \leq r, s \leq n$.

Thus,

$$T(\sigma,\pi) \geq \sum_{r=1}^{k} w_r \sigma_{\pi_r}^2 + \sum_{r=k+1}^{n} w_r \sigma_{\tau_r}^2. \qquad (3.4.41)$$

Finally, combining the inequalities (3.4.39) and (3.4.41), we get the inequality in (3.4.37). ■

Remark 3.4.3 : Note that $\underline{V_\mu}(\beta)$ is a lower bound on the variance of the expected completion times for the subproblem with jobset as β. This can easily be obtained using the lower bounding scheme for the deterministic CTV problem detailed in Subsection 2.9.1 of Chapter 2. We take the scheme of Mittenthal, Raghavachari and Rana [1994] for the purpose.

Remark 3.4.4 : It is well known (see Smith [1956], Conway, Maxwell and Muller [1967]) that for a given jobset, the average of (expected) completion times is minimized by SPT (SEPT) sequence. Thus $\bar{E}_{min}(\beta)$ is simple to workout.

Remark 3.4.5 : The lower bound given by (3.4.37) corresponding to a partial sequence α with $|\alpha| = n-1$, is the expected variance of the completion times for $\pi = (\alpha, j)$ where $N \setminus \alpha = \{j\}$.

3.4.2 Dominance Criterion

The following result gives a dominance rule for fathoming partial sequences which is used in the development of an algorithm presented later in Subsection 3.4.3.

Lemma 3.4.6 : Let α_1 and α_2 be two given partial sequences containing the same jobs. Also, let $\pi = (\alpha_1, \beta)$ and $\pi' = (\alpha_2, \beta)$ be the identical completions of α_1 and α_2 respectively for any arbitrary partial sequence β. Then

$$E[V(\pi)] \leq E[V(\pi')]$$

$$\text{if} \quad k\{E[V(\alpha_1)] - E[V(\alpha_2)]\} + \frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2(\sigma_{\pi_r}^2 - \sigma_{\pi'_r}^2)$$

$$\leq \frac{k(n-k)}{n}\left[\bar{E}(\alpha_1) - \bar{E}(\alpha_2)\right]$$

$$\cdot\left[2\sum_{j\in\alpha_1}\mu_j + 2\bar{E}_{min}(\beta) - \bar{E}(\alpha_1) - \bar{E}(\alpha_2)\right] \qquad (3.4.42)$$

where

$$\pi = (\pi_1, \pi_2, \ldots, \pi_n),$$
$$\pi' = (\pi'_1, \pi'_2, \ldots, \pi'_n)$$
$$\text{and } k = |\alpha_1| = |\alpha_2|.$$

Proof : For any sequence π, we know from the relation (3.4.28) that

$$nE[V(\pi)] = nV_\mu(\pi) + T(\sigma, \pi). \qquad (3.4.43)$$

By Lemma 3.4.1, we know that

$$nV_\mu(\pi) = kV_\mu(\alpha_1) + (n-k)V_\mu(\beta) + \frac{k(n-k)}{n}\left[\bar{E}(\alpha_1) - \sum_{j\in\alpha_1}\mu_j - \bar{E}(\beta)\right]^2. \qquad (3.4.44)$$

Next,

$$T(\sigma,\pi) = \sum_{r=1}^{k}\frac{(r-1)(n-r+1)}{n}\sigma_{\pi_r}^2 + \sum_{r=k+1}^{n}\frac{(r-1)(n-r+1)}{n}\sigma_{\pi_r}^2$$

$$= \sum_{r=1}^{k}\frac{(r-1)(k-r+1)}{k}\sigma_{\pi_r}^2 + \frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2\sigma_{\pi_r}^2$$

$$+ \sum_{r=k+1}^{n}\frac{(r-1)(n-r+1)}{n}\sigma_{\pi_r}^2. \qquad (3.4.45)$$

Using the relations (3.4.44), (3.4.45) and (3.4.43), we get

$$nE[V(\pi)] = kE[V(\alpha_1)] + (n-k)V_\mu(\beta) + \frac{k(n-k)}{n}\left[\bar{E}(\alpha_1) - \sum_{j\in\alpha_1}\mu_j - \bar{E}(\beta)\right]^2$$

$$
\begin{aligned}
&+\frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2\sigma^2_{\pi_r} \\
&+\sum_{r=k+1}^{n}\frac{(r-1)(n-r+1)}{n}\sigma^2_{\pi_r}. \qquad (3.4.46)
\end{aligned}
$$

Similarly, we can get

$$
\begin{aligned}
nE[V(\pi')] &= kE[V(\alpha_2)] + (n-k)V_\mu(\beta) \\
&+\frac{k(n-k)}{n}\left[\bar{E}(\alpha_2) - \sum_{j\in\alpha_2}\mu_j - \bar{E}(\beta)\right]^2 \\
&+\frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2\sigma^2_{\pi'_r} \\
&+\sum_{r=k+1}^{n}\frac{(r-1)(n-r+1)}{n}\sigma^2_{\pi'_r}. \qquad (3.4.47)
\end{aligned}
$$

Note that $\sum_{j\in\alpha_1}\mu_j = \sum_{j\in\alpha_2}\mu_j$ and $\sigma^2_{\pi_r} = \sigma^2_{\pi'_r}$ for $r = k+1, k+2, \ldots, n$. Therefore, using the equations (3.4.46) and (3.4.47), we have

$$
\begin{aligned}
&n\{E[V(\pi)] - E[V(\pi')]\} \\
&= k\{E[V(\alpha_1)] - E[V(\alpha_2)]\} + \frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2(\sigma^2_{\pi_r} - \sigma^2_{\pi'_r}) \\
&+\frac{k(n-k)}{n}\left[\bar{E}(\alpha_1) - \bar{E}(\alpha_2)\right] \\
&\quad\cdot\left[\bar{E}(\alpha_1) + \bar{E}(\alpha_2) - 2\sum_{j\in\alpha_1}\mu_j - 2\bar{E}(\beta)\right]. \qquad (3.4.48)
\end{aligned}
$$

Thus,

$$
E[V(\pi)] \le E[V(\pi')]
$$

$$
\begin{aligned}
\text{if}\quad & k\{E[V(\alpha_1)] - E[V(\alpha_2)]\} + \frac{n-k}{nk}\sum_{r=1}^{k}(r-1)^2(\sigma^2_{\pi_r} - \sigma^2_{\pi'_r}) \\
&\le \frac{k(n-k)}{n}\left[\bar{E}(\alpha_1) - \bar{E}(\alpha_2)\right] \\
&\quad\cdot\left[2\sum_{j\in\alpha_1}\mu_j + 2\bar{E}(\beta) - \bar{E}(\alpha_1) - \bar{E}(\alpha_2)\right] \qquad (3.4.49)
\end{aligned}
$$

holds any arbitrary partial sequence β. It may be easily observed that

$$2\sum_{j\in\alpha_1}\mu_j+2\bar{E}(\beta)-\bar{E}(\alpha_1)-\bar{E}(\alpha_2)\ \geq\ 0. \qquad (3.4.50)$$

· Hence, the inequality (3.4.42) follows directly from the equation (3.4.49). ■

Corollary 3.4.7 : Let α_1, α_2 be defined as in Lemma 3.4.6 and G_1 (G_2) be the set of all completions of α_1 (α_2). If the condition (3.4.42) holds, all the *sequences in G_2 are dominated by those in G_1 with respect to expected* CTV.

3.4.3 A Branch-and-Bound Algorithm

At the present, there is no exact procedure available to solve the stochastic CTV problem. In the following, we propose a branch-and-bound algorithm with general job processing times using (i) the lower bound and (ii) the dominance rule derived in Subsections 3.4.1 and 3.4.2 respectively.

We represent a node in this algorithm by a partial sequence. A node is said to be of order k if the cardinality of the corresponding partial node is k. The node of order zero is denoted by (.). Every node of order $k\geq 1$ is denoted by its corresponding partial sequence, say, α. A descendant from a node α of order k $(0\leq k\leq n-1)$ is (α,j) for some $j\in N\setminus\alpha$ and there are in all $n-k$ descendants from the node α. A node α of order k $(0\leq k\leq n-1)$ at any stage is said to be active, if it has no descendant till then. At any stage of computation, let W be the set of all active nodes.

Step 1: Set $W=\{(.)\}$, $\alpha=(.)$, $MECTV=\left(\sum_{j\in N}\mu_j\right)^2+\max_{j\in N}\sigma_j^2$ and $k=0$. Go to Step 2.

Step 2: Generate $n-k$ descendants of α of the form (α,j) for $j\in N\setminus\alpha$. Let $\gamma_1,\gamma_2,\ldots,\gamma_{n-k}$ be the nodes thus generated. Update $W\leftarrow(W\setminus\alpha)\cup\{\gamma_1,\gamma_2,\ldots,\gamma_{n-k}\}$. For each γ_i $(1\leq i\leq n-k)$, compute lower bound given by Lemma 3.4.2. Go to Step 3.

Step 3: (*a*) If $k = n - 2$, let L_0 be the minimum of the lower bounds corresponding to the nodes γ_1 and γ_2. If $MECTV \geq L_0$, set $MECTV = L_0$ and delete all the nodes in W with lower bound not less than L_0. (*b*) Invoke the dominance rule given in Lemma 3.4.6 and delete all the dominated nodes from W. Choose a node $\alpha \in W$ having the smallest lower bound and let $k = |\alpha|$. If $k \leq n - 1$, go to Step 2. Otherwise go to Step 4.

Step 4: The sequence α is an optimal sequence and $MECTV$ is the corresponding expected CTV value.

Remark 3.4.8 : If any problem instance satisfies the sufficient condition for V-shapedness (refer to Theorems 3.3.1 and 3.3.3 for general processing times and Theorem 3.3.20 for ordered processing times), an algorithm analogous to that of Manna and Prasad [1994] (for the deterministic CTV problem, detailed in Subsection 2.9.2 of Chapter 2) will be more appropriate.

Remark 3.4.9 : If $\sigma_1^2 = \max_j \sigma_j^2$, the algorithm starts with $k = 1$ and the initial node as (1) (refer to Theorem 3.2.6).

Remark 3.4.10 : For the problems with ordered processing times, the algorithm can be suitably modified to get substantial reduction in computation using the Theorems 3.3.11 and 3.3.5 as follows :

i) Initialize the algorithm (in Step 1) with $k = \lfloor \frac{n+1}{2} \rfloor + 1$ and W as the set of all LEPT partial sequences (of order k) with largest job at the first position, that is, $W = \{\alpha : \alpha = (j_1, j_2, j_3, \ldots, j_k)$ satisfying $j_1 = 1$ and $\mu_{j_2} \geq \mu_{j_3} \geq \cdots \geq \mu_{j_k}\}$.

ii) In order to generate the descendants (in Step 2) of a node $\alpha = (j_1, j_2, \ldots, j_k)$ such that $j_2 \neq 2$ and $|\alpha| < n - 2$, we make use of the Theorem 3.3.5. For such an α, it is enough to consider the $(n - k - 1)$ descendants — (α, j) for $j \in N \setminus (\alpha \cup \{2\})$.

iii) From any node α with $|\alpha| = n-2$, we generate (using Theorem 3.3.11(*iii*)) the only descendant (α, j) where $\mu_j \leq \mu_i$ and $N \setminus \alpha = \{i, j\}$.

3.5 DISCUSSION

In this chapter have studied the CTV problem with random (stochastic) job processing times with the objective as minimization of the expected value of the completion time variance. Like the deterministic version, this problem is also NP-hard.

We have derived several sufficient conditions (which are quite likely to hold in real life) for V-shaped property of optimal sequences for general random processing times as well as some special cases of the problem. A simple but stronger sufficient criterion for V-shaped optimality has been obtained for the special case studied by Vani and Raghavachari [1987]. We have introduced a new special case with the realistic assumption that the job processing times are ordered, that is, if $\mu_i > \mu_j$ for any i and j, then $\sigma_i^2 > \sigma_j^2$. A numerical example has been provided to show that even under this assumption, there may not exist a V-shaped optimal sequence. However, we obtain several results on the properties of optimal sequence and prove the existence of an L-G-S optimal sequence $\pi = (\pi_1, \ldots, \pi_n)$ satisfying (*i*) the first job is the largest job, (*ii*) $\mu_{\pi_2} \geq \ldots \geq \mu_{\pi_{\lfloor \frac{n+1}{2} \rfloor +1}}$, and (*iii*) $\mu_{\pi_{n-1}} \leq \mu_{\pi_n}$. It is shown that V-shaped optimality holds good for small problems ($n \leq 7$) if the processing times are ordered.

Finally, we have presented a procedure to derive a lower bound on the expected CTV and a dominance rule (applicable to partial sequences) which are in turn used in the development of a branch-and-bound algorithm to solve the stochastic CTV problem with arbitrary job processing times. We have also discussed here the required modifications in the algorithm for ordered processing times in order to avail substantial reduction in the computation involved.

CHAPTER 4

HEURISTIC METHODS : DETERMINISTIC AND STOCHASTIC CTV PROBLEMS

4.1 INTRODUCTION.

We have already discussed the procedures for exact solution of the deterministic and the stochastic CTV problems in Chapter 2 and Chapter 3 respectively.

Since both the problems are NP-hard, it is unlikely that we can have efficient (polynomial time complexity) algorithm in order to derive exact optimal solution for the same. Therefore, it becomes essential to develop heuristic methods to derive near optimal solutions.

For the deterministic CTV problem, Eilon and Chowdhury [1977] are the first to propose heuristic methods. Subsequently, many researchers have contributed towards this. For reference, see Kanet [1981], Vani and Raghavachari [1987], Gupta, Gupta and Bector [1990], Mittenthal, Raghavachari and Rana [1993], Gupta, Gupta and Kumar [1993], Manna and Prasad [1994] etc.

However, until now, there is no heuristic method available for the stochastic CTV problem.

In section 4.2, we present the preliminary results concerning the position of the smallest job in an optimal sequence for the deterministic CTV problem. These results are used later in this chapter to describe the heuristic of Manna and Prasad [1994]. Section 4.3 is devoted to the heuristic methods for deterministic CTV problem. Here, we present the available heuristic methods and compare their performance by numerical experimentation. In section 4.4, we propose a heuristic procedure for the stochastic CTV problem with general job processing times and assess its performance again based on numerical investigation.

4.2 PRELIMINARY RESULTS

In this section, we present some preliminary results to derive lower and upper bounds for the position of the smallest job in a V-shaped optimal sequence for the deterministic CTV problem.

Lemma 4.2.1 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_{i-1}, \pi_i, \pi_{i+1}, \ldots, \pi_n)$ be a sequence and π' the sequence obtained from π by interchanging the two jobs π_i and π_{i+1}. Then

$$\frac{n}{2}[V(\pi) - V(\pi')] = (p_{\pi_i} - p_{\pi_{i+1}})\left[C_{[i]}(\pi) - \bar{C}(\pi) - \frac{n-1}{2n}(p_{\pi_i} - p_{\pi_{i+1}})\right].$$

This can be easily proved by observing that

(*i*) $C_{[r]}(\pi') = C_{[r]}(\pi)$ for $r \neq i$,

(*ii*) $C_{[i]}(\pi') = C_{[i]}(\pi) + (p_{\pi_{i+1}} - p_{\pi_i})$,

(*iii*) $\bar{C}(\pi') = \bar{C}(\pi) + \frac{1}{n}(p_{\pi_{i+1}} - p_{\pi_i})$.

■

The following result gives a necessary condition on the position of the smallest job in an optimal sequence.

Theorem 4.2.2 : For $n > 5$, in any optimal sequence of the form $(1, \ldots, 2)$, the smallest job does not lie in $2nd$ or $3rd$ positions when it is unique.

Proof : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence with $\pi_1 = 1$ and $\pi_n = 2$. Suppose $\pi_3 = n$, that is, the smallest job takes third position in π. Now consider the sequence π' obtained from π by interchanging the jobs π_3 and π_4. We have from Lemma 4.2.1,

$$\frac{n}{2}[V(\pi) - V(\pi')] = (p_{\pi_3} - p_{\pi_4})\left[C_{[3]}(\pi) - \bar{C}(\pi) - \frac{n-1}{2n}(p_{\pi_3} - p_{\pi_4})\right].$$

It can be easily verified that

$$n\left[C_{[3]}(\pi) - \bar{C}(\pi) - \tfrac{n-1}{2n}(p_{\pi_3} - p_{\pi_4})\right]$$

$$= (p_{\pi_2} - p_2) - \{\tfrac{n-5}{2}(p_{\pi_3} + p_{\pi_4}) + \textstyle\sum_{j=5}^{n-1}(n-j+1)p_{\pi_j}\}$$

$$< 0.$$

Since $p_{\pi_3} - p_{\pi_4} < 0$, it now follows that $V(\pi) - V(\pi') > 0$, that is, π' is strictly better than π which contradicts the optimality of π. Therefore $\pi_3 \neq n$. Similarly, we can argue that $\pi_2 \neq n$. ■

Remark 4.2.3 : For $n \leq 5$, Schrage [1975] has derived optimal sequences.

We now introduce two sets of indices which are used in deriving lower and upper bounds for the position of the smallest job in an optimal V-shaped sequence. We follow the usual convention that $\sum_{r=a}^{b} = 0$ if $a > b$.

Define

$$u_k = \left[p_3 + \textstyle\sum_{r=4}^{k}(r-2)p_r + (k-1)p_n\right] - \left[p_2 + \textstyle\sum_{r=k+1}^{n-1}(r-k+1)p_r\right] + \tfrac{n-1}{2}(p_{n-k+2} - p_n)$$

$$\text{for } 4 \leq k \leq n-2$$

and

$$v_k = \left[p_3 + \textstyle\sum_{r=n-k+3}^{n-1}(r - \overline{n-k+1})p_r\right] - \left[p_2 + (n-k+1)p_n + \textstyle\sum_{r=4}^{n-k+2}(r-2)p_r\right] - \tfrac{n-1}{2}(p_k - p_n)$$

$$\text{for } 5 \leq k \leq n-1.$$

Lemma 4.2.4 : Both u_k and v_k increase with k.

Proof : For any k, $4 \leq k \leq n-3$, consider

$$\begin{aligned}
& u_{k+1} - u_k \\
= \; & \left[\sum_{r=4}^{k+1}(r-2)p_r - \sum_{r=4}^{k}(r-2)p_r\right] + p_n \\
& \quad - \left[\sum_{r=k+2}^{n-1}(r-k)p_r - \sum_{r=k+1}^{n-1}(r-k+1)p_r\right] + \frac{n-1}{2}(p_{n-k+1} - p_{n-k+2}) \\
= \; & (k-1)p_{k+1} + 2p_{k+1} + \sum_{r=k+2}^{n} p_r + p_r + \frac{n-1}{2}(p_{n-k+1} - p_{n-k+2}) \\
= \; & (k+1)p_{k+1} + \sum_{r=k+2}^{n} p_r + p_r + \frac{n-1}{2}(p_{n-k+1} - p_{n-k+2}) \\
> \; & 0.
\end{aligned}$$

$$\therefore \quad u_k \uparrow k.$$

Next, for any k, $5 \leq k \leq n-2$, it can be similarly seen that

$$\begin{aligned}
v_{k+1} - u_k &= (n-k+2)p_{n-k+2} + \sum_{r=n-k+3}^{n} p_r + p_r + \frac{n-1}{2}(p_k - p_{k+1}) \\
&> 0.
\end{aligned}$$

$$\therefore \quad v_k \uparrow k.$$

This completes the proof. ∎

Lemma 4.2.5 : Let

$$L = \max_{4 \leq k \leq n-2}\{k : u_k \leq 0\}$$

and

$$U = \min_{5 \leq k \leq n-1}\{k : v_k \geq 0\}.$$

Then $U - L \geq 2$.

Proof : We can write $v_{k+1} - u_k$

$$= \left[\sum_{r=n-k+2}^{n-1}(r - \overline{n-k})p_r - \sum_{r=4}^{k}(r-2)p_r\right]$$
$$- \left[\sum_{r=4}^{n-k+1}(r-2)p_r - \sum_{r=k+1}^{n-1}(r-k+1)p_r\right]$$
$$- \tfrac{n-1}{2}(p_{k+1} + p_{n-k+2}).$$

Rearranging the terms on the right hand side, we get

$v_{k+1} - u_k$

$$= \left[\sum_{r=n-k+2}^{n-2}(r-2)p_r - \sum_{r=4}^{k}(r-2)p_r\right]$$
$$- \left[\sum_{r=4}^{n-k+1}(r-2)p_r - \sum_{r=k+1}^{n-2}(r-k+1)p_r\right]$$
$$-(n-k-2)\sum_{r=n-k+2}^{n-2} p_r + (n-1)\left[p_{n-1} - (p_{k+1} + p_{n-k+2})/2\right].$$

It is obvious that the first and last terms are non-positive, whereas the second and third terms are non-negative.

Thus $v_{k+1} \leq u_k$. In fact, it can be seen that $v_{k+1} < u_k$. It implies that $v_{L+1} < 0$ and consequently $U > L+1$. Hence the Lemma holds. ∎

The result below gives lower and bounds for the position of smallest job in an optimal sequence.

Theorem 4.2.6 : To minimize the CTV, it is enough to consider only V-shaped sequences of the form $(1, 3, \ldots, 2)$ with the smallest job in one of the positions $L+1, L+2, \ldots, U-1$.

Proof : Consider an arbitrary V-shaped sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ with $\pi_1 = 1$, $\pi_2 = 3$, $\pi_n = 2$ and $\pi_k = n$.

Case (i) $k = 3$: Suppose the smallest job is unique. Then we have, by Theorem 4.2.2, a better sequence with the smallest job in fourth position so that it can be treated as case (ii). If the smallest job is not unique, then π_4 is also one of the smallest jobs and hence case (*ii*) applies here.

Case (*ii*) $4 \leq k \leq L$: Obtain the sequence π' from π by interchanging the jobs π_k and π_{k+1}. We will show that $V(\pi) \geq V(\pi')$. From Lemma 4.2.1, we have

$$V(\pi) - V(\pi') = \frac{2}{n}(p_{\pi_k} - p_{\pi_{k+1}})\left[C_{[k]}(\pi) - \bar{C}(\pi) - \frac{n-1}{2n}(p_{\pi_k} - p_{\pi_{k+1}})\right].$$

Now, $n\left[C_{[k]}(\pi) - \bar{C}(\pi) - \frac{n-1}{2n}(p_{\pi_k} - p_{\pi_{k+1}})\right]$

$$= \left[p_3 + \sum_{r=3}^{k-1}(r-1)p_{\pi_r} + (k-1)p_n\right] - \left[\sum_{r=k+1}^{n-1}(n-r+1)p_{\pi_r} + p_2\right]$$
$$-\tfrac{n-1}{2}(p_n - p_{\pi_{k+1}})$$
$$\leq \left[p_3 + \sum_{r=4}^{k}(r-2)p_r + (k-1)p_n\right] - \left[p_2 + \sum_{r=k+1}^{n-1}(r-k+1)p_r\right]$$
$$+\tfrac{n-1}{2}(p_{n-k+2} - p_n)$$
$$= u_k.$$

The above inequality holds since p_r's are in non-decreasing order, the sequence π is V-shaped with smallest job in k-th position, and $p_{n-1} \leq p_{\pi_{k+1}} \leq p_{n-k+2}$. Next, since $k \leq L$, $u_k \leq u_L \leq 0$.

$$\therefore \quad V(\pi) - V(\pi') \geq 0.$$

So that π' is at least as good as π. This interchanging procedure may be repeated until smallest job is shifted to the $(L+1)$-th position.

Case (*iii*) $k \geq U$: Since $k \geq L$, $v_k \geq v_L \geq 0$. Following similar arguments as in case (ii), we can arrive at a V-shaped sequence η with smallest at $(U-1)$-th position such that η is as good as π.

This completes the proof of the theorem. ■

Remark 4.2.7 : *The above theorem enables us to confine the search for optimal sequence to a set of* $\sum_{r=L+1}^{U-1}\binom{n-4}{r-3}$ *V-shaped sequences of the form* $(1, 3, \ldots, 2)$ *only.*

Corollary 4.2.8 : If p_2 is sufficiently large, then $(1,3,4,5,\ldots,n-1,n,2)$ is an optimal sequence.

Proof : If $u_{n-2} \leq 0$, that is,

$$p_2 \geq p_3 + \frac{n+3}{2}p_4 + \sum_{r=5}^{n-2}(r-2)p_r - 2p_{n-1} + \frac{n-5}{2}p_n$$

then, by performing pairwise interchanges repeatedly, the smallest job can be shifted to the $(n-1)$-th position. Now the result follows from V-shapedness of optimal sequence. ∎

Remark 4.2.9 : It is evident from the above corollary that magnitude of the larger jobs play vital role in the determination of optimal sequence.

4.3 HEURISTICS FOR DETERMINISTIC CTV PROBLEM

The heuristic methods, available in the literature, for the deterministic CTV problem can be categorized into two classes, namely, 'Basic heuristic' and 'Improvement heuristic'. The essential difference between these two is as follows. Unlike a 'basic heuristic', an 'improvement heuristic' asks for an input (a sequence of all the jobs for a given problem instance) to initialize the method and gives an improved sequence as output.

In this section, we present these procedures and investigate their performances by extensive numerical experimentation. Subsequently, we also study combinations of these procedures for possible improvement in solutions.

4.3.1 Description of Heuristics

We present now the heuristic procedures available in the literature and for the convenience of reference we denote them by **H1**, **H2**, ... etc.

H1 : 'Method 1.2' of Eilon and Chowdhury [1977]

Eilon and Chowdhury [1977] have developed five heuristics for the purpose and found that the 'Method 1.2' is best among them. Here we consider the 'Method 1.2' only.

This method uses V-shaped property and the conjecture of Schrage [1977] on the position of the largest four jobs. It starts with a two-sided partial sequence $(1, 3, 4, \ldots, 2)$ and subsequently assigns the largest unscheduled jobs to the unassigned extreme right and left positions alternately until all the jobs are allocated. Therefore, the method produces a sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ as follows :

$\pi_1 \leftarrow 1,\ \pi_2 \leftarrow 3,\ \pi_3 \leftarrow 4,$ and $\pi_n \leftarrow 2.$

$J \leftarrow 4.$

For $I = 4$ to $\lfloor \frac{n}{2} \rfloor + 1$ Do

 $J \leftarrow J + 1.$

 $\pi_{n-I+3} \leftarrow J.$

 $J \leftarrow J + 1.$

 $\pi_I \leftarrow J.$

End { of I }.

If n is odd then $\pi_{\lfloor \frac{n}{2} \rfloor + 2} \leftarrow n.$

Remark 4.3.1 : This method, **H1**, considers only the order (by processing times) of the jobs and does not take into account the actual magnitudes of the processing times.

Remark 4.3.2 : The solution (sequence) of H1 will have the position of the smallest job, n, almost at the middle of the sequence and the larger jobs are evenly distributed on both the sides of job n following the V-shaped property. In an optimal sequence, if the smallest job occurs closer to either end of the sequence, for example, when p_2 is very large (see Corollary 4.2.8), the performance of

H1 will not be satisfactory. However, the method appears quite appealing for homogeneous job processing times.

H2 : 'SMV Procedure' of Kanet [1981]

The procedure involves n stages and builds on V-shaped partial sequence by successively scheduling the smaller jobs one at a time in the following manner. In the first stage, it starts with the partial sequence α containing the largest job only, that is, $\alpha = (1)$. At any stage k $(1 \leq k < n)$, a V-shaped partial sequence α is represented by $\alpha = (\alpha^{(1)}, k, \alpha^{(2)})$ where $(\alpha^{(1)}, \alpha^{(2)})$ is a permutation of the jobs in $\{1, 2, \ldots, k-1\}$ and either $\alpha^{(1)}$ or $\alpha^{(2)}$ can be empty. From α, the procedure generates two partial sequences β_1 and β_2 where $\beta_1 = (\alpha^{(1)}, k+1, k, \alpha^{(2)})$ and $\beta_2 = (\alpha^{(1)}, k, k+1, \alpha^{(2)})$. Note that β_1 and β_2 are V-shaped partial (complete if $k = n-1$) sequences of the jobs $\{1, 2, \ldots, k+1\}$. Between β_1 and β_2, the one having smaller CTV value (defined for the jobset $\{1, 2, \ldots, k+1\}$) is retained to determine the position of the next larger job as above until a complete sequence having smaller CTV value is obtained.

Let $V(\alpha)$ denote the CTV value corresponding to the jobset α for the partial (or complete) sequence α. Then the procedure is as follows :

$\alpha = (1)$.

For $k = 2$ to n Do

  Let $\alpha = (\alpha^{(1)}, k-1, \alpha^{(2)})$.

  $\beta_1 \leftarrow (\alpha^{(1)}, k, k-1, \alpha^{(2)})$.

  $\beta_2 \leftarrow (\alpha^{(1)}, k-1, k, \alpha^{(2)})$.

  If $V(\beta_1) \leq V(\beta_2)$ then $\alpha \leftarrow \beta_1$,

   Else $\alpha \leftarrow \beta_2$.

End { of k }.

Remark 4.3.3 : Besides the ordering of the jobs, the procedure, at any stage k, decides the position of job k given α (a sequence of the jobs in $\{1, 2, \ldots, k-1\}$) by making use of the actual magnitudes of the largest k jobs. However, it fails to utilize the magnitudes of the remaining smaller jobs.

Remark 4.3.4 : Kanet [1981] has claimed that the 'SMV Procedure' (**H2**) produces optimal solution for $n \leq 5$. The following numerical example contradicts the claim.

j	1	2	3	4	5
p_j	10	10	10	10	1

The procedure (H2) involves the following steps :

$$
\begin{array}{lcl}
 & & \alpha = (1) \\
k = 2 & : & \beta_1 = (2,1) \\
 & & \beta_2 = (1,2) \\
 & & V(\beta_1) = V(\beta_2) \Rightarrow \alpha = (2,1) \\
k = 3 & : & \beta_1 = (3,2,1) \\
 & & \beta_2 = (2,3,1) \\
 & & V(\beta_1) = V(\beta_2) \Rightarrow \alpha = (3,2,1) \\
k = 4 & : & \beta_1 = (4,3,2,1) \\
 & & \beta_2 = (3,4,2,1) \\
 & & V(\beta_1) = V(\beta_2) \Rightarrow \alpha = (4,3,2,1) \\
k = 5 & : & \beta_1 = (5,4,3,2,1) \\
 & & \beta_2 = (4,5,3,2,1) \\
 & & V(\beta_1) = 200 > 140.96 = V(\beta_2) \Rightarrow \alpha = (4,5,3,2,1)
\end{array}
$$

Thus, **H2** yields the best sequence as (4, 5, 3, 2, 1) and 140.96 as its CTV value. But, an optimal sequence (1, 3, 4, 5, 2) (see Schrage [1975]) has the CTV value as 111.44. Hence, we get a contradiction to the claim.

H3 : Heuristic Method of Manna and Prasad [1994]

It is based on the results derived in Section 4.2 of this Chapter. The heuristic generates $(U - L - 1)$ V-shaped sequences with the smallest job taking the positions $L+1, L+2, \ldots, U-1$ and, it selects the best among these sequences.

This heuristic method considers the sequences only of the form $(1, 3, \ldots, 2)$. For each k, $L+1 \leq k \leq U-1$, it gives heuristically best sequence say $\pi^{(k)}$ with the smallest job in k-th position. For the algebraic purpose, hypothetical jobs with processing time p_{n-1} are created.

For any particular position k, $L+1 \leq k \leq U-1$, the sequence $(1, 3, \ldots, n, \ldots, 2)$ with job n in the k-th position and $(n-4)$ hypothetical jobs in all the positions except 1, 2, k and n is first considered. The hypothetical jobs are replaced by jobs 4, 5, ..., $n-1$.

The best sequence among these $(U - L - 1)$ sequences $\pi^{(k)}$'s is taken as the heuristic solution.

Let us consider an hypothetical job labeled as $(n+1)$ such that $p_{n+1} = p_{n-1}$. Let L and U (from Lemma 4.2.5) be known. We call a position in a sequence as 'unscheduled' if it is occupied by a hypothetical job. By 'putting an actual job in an unscheduled position', we mean replacing a hypothetical job by an actual job. Now we present the heuristic.

For $k = L+1$ to $U-1$ Do
    $\pi^{(k)} = (\pi_1, \ldots, \pi_n)$ with $\pi_1 = 1, \pi_2 = 3, \pi_k = n, \pi_n = 2$
        and $\pi_r = n+1$ for $r \neq 1, k, n$.
    For $I = 4$ to $N-1$ Do
        If no unscheduled position on the left of position k,
            put job I on the last unscheduled position in $\pi^{(k)}$.
        If no unscheduled position on the right of position k,
            put job I on the first unscheduled position in $\pi^{(k)}$.
        Otherwise
            Obtain π' from $\pi^{(k)}$ by putting job I on the first

unscheduled position in $\pi^{(k)}$.

Obtain π'' from $\pi^{(k)}$ by putting job I on the last unscheduled position in $\pi^{(k)}$.

If $V(\pi') \leq V(\pi'')$ then $\pi^{(k)} = \pi'$

Else $\pi^{(k)} = \pi''$.

End {of I }.

End {of k }.

Select π^* among the $\pi^{(k)}$'s such that $V(\pi^*) = \min_{\pi^{(k)}} V(\pi^{(k)})$.

Numerical Example : We illustrate the above heuristic by the numerical example (Problem No. 3, Kanet[5], pp1457).

Jobs	1	2	3	4	5	6	7	8
Processing times	16	10	9	8	7	6	4	2

It can be seen that $L = 4$ and $U = 6$. This implies that in an optimal V-shaped sequence, smallest job must lie at 5-th position, that is, k can take only value 5. Hence the heuristic starts with $\pi^{(5)} = (1,3,9,9,8,9,9,2)$

$$I = 4 : \pi' = (1,3,4,9,8,9,9,2) \qquad V(\pi') = 142.609375$$
$$\pi'' = (1,3,9,9,8,9,4,2) \qquad V(\pi'') = 143.109375$$
$$\therefore \quad \pi^{(5)} = (1,3,4,9,8,9,9,2)$$

$$I = 5 : \pi' = (1,3,4,5,8,9,9,2) \qquad V(\pi') = 172.75$$
$$\pi'' = (1,3,4,9,8,9,5,2) \qquad V(\pi'') = 166.609375$$
$$\therefore \quad \pi^{(5)} = (1,3,4,9,8,9,5,2)$$

$$I = 6 : \pi' = (1,3,4,6,8,9,5,2) \qquad V(\pi') = 187.359375$$
$$\pi'' = (1,3,4,9,8,6,5,2) \qquad V(\pi'') = 187.239375$$
$$\therefore \quad \pi^{(5)} = (1,3,4,9,8,6,5,2)$$

$I = 7 : \pi^{(5)} = (1,3,4,7,8,6,5,2)$

$$\therefore \quad \pi^* = (1,3,4,7,8,6,5,2)$$

We observe that this solution, π^*, is optimal for the problem.

Remark 4.3.5 : It may be noted that H3 takes into account not only the order but also the *magnitudes of all the processing times.*

H4 : Heuristic Method of Vani and Raghavachari [1987]

This method takes a V-shaped (complete) sequence as input and improves upon it in $(n-3)$ iterations. In each iteration, it checks sequentially the desirability (with respect to CTV value) for interchange of two jobs chosen suitably.

The method is given below :

$\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a V-shaped sequence.
with $\pi_1 = 1$ and $\pi_n = 2$.
$k \leftarrow 1,\ q \leftarrow 1,\ m \leftarrow 3.$
Repeat
Obtain a sequence π' from π
by interchanging the jobs π_{k+1} and π_{n-q}.
If $V(\pi) \leq V(\pi')$ then $k \leftarrow k+1,\ m \leftarrow m+1,$
Else $\pi \leftarrow \pi',\ q \leftarrow q+1,\ m \leftarrow m+1.$
Until $m = n$.

Remark 4.3.6 : The performance of the method is likely to depend on the input sequence. Vani and Raghavachari [1987] suggests this input be taken as a 'good' V-shaped sequence.

Remark 4.3.7 : We note that given a V-shaped input sequence, the resultant sequence (obtained from H4) need not be V-shaped, although it may have objective value very close to that of optimal one. Consider the example (problem No. 8, Eilon and Chowdhury [1977], pp571). Here $n = 10$ and $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}) = (100, 41, 25, 21, 13, 10, 9, 8, 7, 5)$. If one starts with the V-shaped sequence (1, 3, 4, 5, 6, 7, 8, 9, 10, 2) as the input, this method gives output sequence as (1, 3, 4, 10, 7, 8, 9, 6, 5, 2) which is not V-shaped. However, it's CTV value is 1348.01 against the optimal CTV value of 1336.

H5 : Heuristic Method of Gupta, Gupta and Bector [1990]

The present method, again, at Step 1, asks for a V-shaped sequence to initialize the procedure, and subsequently exploits the principle of feasible complementary pairs (FCP) (refer to Bector, Gupta and Gupta [1988]) to effect improvement.

The description of the method involves the following terminology :

Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be a V-shaped sequence with $C_{[r]}(\pi) \leq \bar{C}(\pi) < C_{[r+1]}(\pi)$ for some $1 \leq r \leq n - 1$. Also, let $\alpha = (\pi_1, \pi_2, \ldots, \pi_r)$ and $\beta = (\pi_{r+1}, \pi_{r+2}, \ldots, \pi_n)$.

V-feasible Sequence : The sequence π is called V-feasible sequence if α (β) is in LPT (SPT) order.

Feasible Complementary Pair : Let $\pi_i \in \alpha$ and $\pi_j \in \beta$ and π' be a sequence obtained from π by interchanging the jobs π_i and π_j. A pair (π_i, π_j) is said to be Feasible Complementary Pair (FCP) if (i) $p_{\pi_i} < p_{\pi_j}$, (ii) π' is V-shaped and (iii) $V(\pi') < V(\pi)$.

We now present the heuristic method below :

1. Find an initial V-shaped sequence π.
2. Is π V-feasible ? If not, go to 8.
3. Is there any FCP available in π ? If yes, go to 7.
4. Find its dual π^D and $\pi \leftarrow \pi^D$.
5. Is π V-feasible ? If not, go to 13.
6. Is there any FCP available in π ? If not, go to 13.
7. Determine the sequence iteratively by interchanging the jobs in a FCP (select the pair with most negative value) until such time there is no further FCP available, and go to 4.
8. Find its dual π^D and $\pi \leftarrow \pi^D$.
9. Is π V-feasible ? If not, go to 12.
10. Is there any FCP available in π ? If not, go to 13.
11. Determine the sequence iteratively by interchanging the jobs in a FCP (select the pair with most negative value) until such time there is no further FCP available. Go to 4.
12. Obtain V-shaped sequence, π' from π by arranging the jobs preceding $\tilde{C}(\pi)$ in LPT order and the remaining set of jobs in SPT order. $\pi \leftarrow \pi'$. If π V-feasible ? If yes, go to 3.
13. STOP.

Remark 4.3.8 : Gupta, Gupta and Bector [1990] have concluded that the performance of this heuristic, **H5** varies with input sequence. Specifically, they have recommended the use of the heuristic of Kanet [1981], namely **H2**, to obtain an input sequence for this method.

Apart from the above methods, one may refer to Gupta, Gupta and Kumar [1993] and Mittenthal, Raghavachari and Rana [1993] for heuristic solutions of the deterministic CTV problem. The procedure of Gupta, Gupta and Kumar [1993] is based on *genetic algorithms* (refer to Goldberg [1989]) and that of Mittenthal, Raghavachari and Rana [1993] makes use of the concept of *simulated annealing* (refer to Ackley [1987]). Both of these procedures have potential to address more general problem involving non-regular penalty functions. However, successful implementation of these procedures depend upon careful determination of the parameters involved therein. We omit these heuristic methods from evaluation or comparison of performances here.

4.3.2 Comparison of Heuristics

We shall now compare the performance of the heuristics **H1** to **H5** by numerical investigation. It may be noted that **H1**, **H2** and **H3** are 'basic heuristics', and **H4** and **H5** are 'improvement heuristics'. Initially, we consider only **H1**, **H2** and **H3** for comparison. Later, in view of the Remarks 4.3.6 and 4.3.8, we utilize **H4** and **H5**, for improvement of solutions generated by **H1**, **H2** and **H3**, to assess and compare their effectiveness. The basis of comparison is the deviation of the objective value given by a heuristic from the optimal one.

The investigation involves 70 numerical problems generated randomly with the number (n) of jobs varying from 10 to 70. Ten problems are generated for each value of n. The processing times are drawn from the discrete uniform distribution with sample space as $\{1, 2, \ldots, 50\}$. For each problem, the heuristics **H1**, **H2** and **H3** are applied, and an exact optimal sequence is derived using the algorithm of Manna and Prasad [1995] (refer to Chapter 2).

For a given problem, let V_0 denote the optimal objective value and V_h, the objective value given by a heuristic h. The percentage deviation of V_h from V_0 is

$$E_h = \frac{V_h - V_0}{V_0} \times 100.$$

We take E_h as the performance index of the heuristic h and it is used for comparison of the heuristics. The performance index is evaluated for the heuristics **H1**, **H2** and **H3** in all 70 problems, and the values are summarized in the Tables 4.3.1 and 4.3.2.

Table 4.3.1 : Average of Performance Index Values

n	Average of E_h for heuristic		
	H1	H2	H3
10	0.0657	0.0043	0.0000
20	0.0459	0.0154	0.0014
30	0.0156	0.0954	0.0005
40	0.0139	0.0164	0.0004
50	0.0088	0.0162	0.0001
60	0.0090	0.1134	0.0001
70	0.0046	0.2063	< 0.0001

Table 4.3.2 : Distribution of Performance Index Values

Heuristic	Number of Problems Solved with E_h						Max	Mean
(h)	$= 0$	≤ 0.01	≤ 0.05	≤ 0.10	≤ 0.50	≤ 1.00	E_h	E_h
H1	36	42	61	65	70	70	0.2816	0.0233
H2	39	52	55	58	65	69	1.5282	0.0866
H3	40	70	70	70	70	70	0.0085	0.0003

Remark 4.3.9 : On comparison of the heuristics **H1**, **H2** and **H3**, we have the *following points :*

(a) **H3** is the best under any basis of comparison possible from the Tables 4.2.1 and 4.2.2,

(b) **H1** and **H3** exhibit decreasing trends (refer to Table 4.2.1) in average performance index values, that is, the average performance of these heuristics become better for larger problem sizes,

(c) It is interesting to note that **H3** solves over 57% of the problems optimally. For all the problem instances, the percentage relative error $E_{H3} \leq 0.0085\%$ and the overall mean of E_{H3} is only 0.0003%,

(d) All the three heuristics perform almost equally well if 1% deviation is allowed,

(e) When no deviation is allowed, both **H2** as well as **H3** are marginally better than H1,

(f) Let ν_h denote the conditional mean of the distribution of E_h given $E_h > 0$. ν_h provides an idea on the performance of the heuristic h when the heuristic does not give an optimal solution for a problem. It is of interest to note that $\nu_{H3} = 0.0008\% < \nu_{H1} = 0.0479\% < \nu_{H2} = 0.1956\%$.

We now evaluate and compare the effectiveness of the improvement heuristics **H4** and **H5** as follows. Given any problem instance, we apply a basic heuristic h ($h =$ H1, H2, H3) to obtain a solution which, in turn, is taken as input for an improvement heuristic h' ($h' =$ H4, H5) to derive a final solution for the original problem. Subsequently, we measure the extent of deviation of this solution from the optimal one by $E_{h'}$ for $h' =$ H4, H5.

For this purpose, we have taken the same problem set used for the comparison of the basic heuristics. The results are summarized separately for H4 and **H5** in the Tables 4.3.3 and 4.3.4 respectively.

Table 4.3.3 : Distribution of E_{H4}

Basic	Number of Problems Solved with E_{H4}						Max	Mean
Heuristic	$= 0$	≤ 0.01	≤ 0.05	≤ 0.10	≤ 0.50	≤ 1.00	E_{H4}	E_{H4}
H1	37	46	68	70	70	70	0.0750	0.0105
H2	40	53	55	58	65	69	1.4157	0.0798
H3	40	70	70	70	70	70	0.0085	0.0003

Table 4.3.4 : Distribution of E_{H5}

Basic Heuristic	Number of Problems Solved with E_{H5} = 0	≤ 0.01	≤ 0.05	≤ 0.10	≤ 0.50	≤ 1.00	Max E_{H5}	Mean E_{H5}
H1	40	61	70	70	70	70	0.0172	0.0032
H2	40	57	63	64	69	70	0.9055	0.0351
H3	40	70	70	70	70	70	0.0012	0.0001

Remark 4.3.10 : In order to study the performances of **H4** and **H5**, we refer to the Tables 4.3.2, 4.3.3 and 4.3.4 and observe that :

(a) **H4** is effective when the input sequence is obtained from **H1** or **H2**,

(b) **H5** can effect improvement on input sequence obtained from **H1**, **H2** or **H3**,

(c) **H5** is superior to **H4**,

(d) The effectiveness of **H5** is low for input from **H2** when compared that from **H1** or **H3**.

4.4 HEURISTIC FOR STOCHASTIC CTV PROBLEM

We know that the stochastic CTV problem is also NP-hard (see Remark 3.2.3 of Chapter 3). In fact, it is more difficult compared to its deterministic version. Thus, it becomes all the more important to study special cases, and develop heuristic procedures which work satisfactorily. In Chapter 3, we have presented several results to characterize some special cases of the problem.

In this section, we propose a heuristic procedure for the stochastic CTV problem with general processing times and study its performance by numerical experimentation.

From the Lemma 3.2.1 of Chapter 3, it is known that, for any sequence $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$,

$$E[V(\pi)] = V_\mu(\pi) + \frac{1}{n}T(\sigma, \pi)$$

where

$$V_\mu(\pi) = \text{Variance of the expected job completion times for } \pi$$

$$\text{and } T(\sigma, \pi) = \frac{1}{n}\sum_{r=1}^{n}(r-1)(n-r+1)\sigma_{\pi_r}^2.$$

The heuristic, initially, makes use of the Remark 3.3.2 (of Chapter 3) in order to derive an approximate solution, and then attempts to effect improvement on the same by a straightforward adaptation of the heuristic H4 developed by Vani and Raghavachari [1987] (refer to Section 4.3).

In order to describe the heuristic, we consider the two problems **P1** and **P2** defined as follows :

P1 : Find a sequence $\pi \ni V_\mu(\pi)$ is minimum.

P2 : Find a sequence $\pi \ni T(\sigma, \pi)$ is minimum.

We know that optimal sequence for **P1** (**P2**) is V-shaped in mean (variance). It is also known that although **P2** is very easy to solve, **P1** is NP-hard. However, we can generate good heuristic solution for **P1** using the methods for the deterministic CTV problem which are discussed in Section 4.3. We take the heuristic **H3** of Manna and Prasad [1994] for this purpose.

We propose below the heuristic procedure for the stochastic CTV problem with general processing times.

Obtain a heuristic solution (sequence) $\pi^{(1)}$ for **P1** using **H3**.
Obtain an optimal solution (sequence) $\pi^{(2)}$ for **P2**.
If $E[V(\pi^{(1)})] \leq E[V(\pi^{(2)})]$ then $\pi \leftarrow \pi^{(1)}$,

Table 4.4.1 : Performance of the Heuristic

Problem No.	Data j :	1	2	3	4	5	6	7	8	9	10	Value of Expected CTV: Optimal	Value of Expected CTV: Heuristic
1	μ_j :	90	82	80	66	63	58					13588.43	13588.43
	σ_j :	2	14	26	1	10	20						
2	μ_j :	28	23	21	17	17	14					931.72	931.72
	σ_j :	4	7	3	5	1	4						
3	μ_j :	93	71	69	43	42	17					5884.38	5884.38
	σ_j :	15	20	10	13	5	1						
4	μ_j :	84	81	61	53	48	41					8690.25	8690.25
	σ_j :	3	18	14	8	6	12						
5	μ_j :	97	97	88	87	76	64	60	57			27447.18	27447.18
	σ_j :	30	11	18	26	5	17	3	15				
6	μ_j :	99	87	84	81	73	63	57	53			24498.13	24498.13
	σ_j :	19	10	1	19	23	12	5	14				
7	μ_j :	76	69	56	43	42	17	12	10			5008.01	5008.01
	σ_j :	10	10	5	13	5	1	7	3				
8	μ_j :	99	91	84	81	61	53	41	34			18288.64	18288.64
	σ_j :	33	30	3	18	14	8	12	2				
9	μ_j :	95	83	68	57	53	50	44	44	26	19	16426.96	16428.63
	σ_j :	9	23	13	4	6	11	13	8	8	7		
10	μ_j :	70	68	68	65	63	62	59	46	41	36	23943.82	23943.82
	σ_j :	3	6	1	31	19	14	9	12	5	6		
11	μ_j :	46	31	27	26	23	22	17	12	12	11	2776.83	2776.83
	σ_j :	10	4	10	5	6	5	1	7	3	1		
12	μ_j :	99	87	85	84	82	63	56	46	24	10	23605.22	23606.02
	σ_j :	19	30	11	3	18	14	8	5	2	2		

Else $\pi \leftarrow \pi^{(2)}$.

$k \leftarrow 0,\ q \leftarrow 0,\ m \leftarrow 1$.

Repeat

Obtain a sequence π' from π

by interchanging the jobs π_{k+1} and π_{n-q}.

If $E[V(\pi')] \leq E[V(\pi)]$ then $k \leftarrow k+1,\ m \leftarrow m+1$,

Else $\pi \leftarrow \pi',\ q \leftarrow q+1,\ m \leftarrow m+1$.

Until $m = n$.

We now present the results of a limited numerical experimentation on the performance of the above heuristic procedure.

Four random problem instances are generated for each n, $n = 6, 8, 10$ (see Table 4.4.1). The heuristic is applied to each of the twelve problem instances and the solutions are obtained. Besides the heuristic expected CTV values, the Table 4.4.1, also, contains the values of the optimal expected CTV for all the problem in order to facilitate comparisons.

Remark 4.4.1 : Using the Table 4.4.1, we note that the proposed heuristic (a) has produced optimal solution in 10 out of 12 problems, and (b) the solutions for the other problems (No. 9 and 12) are near-optimal.

CHAPTER 5

SOME CONJECTURES RELATED TO DETERMINISTIC CTV PROBLEM

5.1 INTRODUCTION

In this chapter, we pose two conjectures based on the behaviour of the deterministic CTV problem observed during a numerical experimentation, and provide some results derived in an attempt to prove them.

The organization of this chapter is as follows. Section 5.2 contains the preliminary results. We derive here a necessary condition for optimality involving the completion time of the smallest job (C_n) and the average of job completion times ($\bar{C}$). This result shows that C_n and $\bar{C}$ must be close enough for any optimal sequence. We deduce the V-shaped property (refer to Theorem 2.2.5 of Chapter 2) from this result. Some new definitions and concepts are given in Section 5.3. The conjectures are described in Section 5.4. In Subsection 5.4.1, we attempt to build a framework to prove a conjecture on the functional behaviour of the CTV where the CTV function is defined as the minimum CTV value for a given position of the smallest job. The second conjecture, given in Subsection 5.4.2, is on optimal sequence for the CTV problem when the processing times are in arithmetic progression.

5.2 PRELIMINARY RESULTS

The following result evaluates the change in CTV value due to interchange of any two jobs in a sequence.

Lemma 5.2.1 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be any sequence. Let π' be a sequence obtained from π by interchanging the two jobs π_s and π_t ($s < t$) only. Then

$$\frac{n}{2}[V(\pi') - V(\pi)] = (p_{\pi_t} - p_{\pi_s})\left[\sum_{r=s}^{t-1}\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\} + \frac{(t-s)(n-t+s)}{2n}(p_{\pi_t} - p_{\pi_s})\right]. \qquad (5.2.1)$$

Proof : We have

$$C_{[r]}(\pi') = C_{[r]}(\pi) \ \text{ for } r = 1, 2, \ldots, s-1, t, t+1, \ldots, n \qquad (5.2.2)$$

$$= C_{[r]}(\pi) + (p_{\pi_t} - p_{\pi_s}) \ \text{ for } r = s, s+1, \ldots, t-1 \qquad (5.2.3)$$

and therefore

$$\bar{C}(\pi') = \bar{C}(\pi) + \frac{t-s}{n}(p_{\pi_t} - p_{\pi_s}). \qquad (5.2.4)$$

Now, we can write

$$n[V(\pi') - V(\pi)]$$

$$= \sum_{r=1}^{n}\left[C_{[r]}(\pi') - \bar{C}(\pi')\right]^2 - \sum_{r=1}^{n}\left[C_{[r]}(\pi) - \bar{C}(\pi)\right]^2$$

$$= \sum_{r=1}^{n}\left[C^2_{[r]}(\pi') - C^2_{[r]}(\pi)\right] - n\left[\bar{C}^2(\pi') - \bar{C}^2(\pi)\right]$$

$$= \sum_{r=s}^{t-1}\left[C^2_{[r]}(\pi') - C^2_{[r]}(\pi)\right] - n\left[\bar{C}^2(\pi') - \bar{C}^2(\pi)\right]$$

(using the equation (5.2.2))

$$= (p_{\pi_t} - p_{\pi_s})\sum_{r=s}^{t-1}\left[2C_{[r]}(\pi) + (p_{\pi_t} - p_{\pi_s})\right]$$

$$-(t-s)(p_{\pi_t} - p_{\pi_s})\left[2\bar{C}(\pi) + \frac{t-s}{n}(p_{\pi_t} - p_{\pi_s})\right]$$

(using the equations (5.2.3) and (5.2.4))

$$= (p_{\pi_t} - p_{\pi_s})\left[2\sum_{r=s}^{t-1}\{C_{[r]}(\pi) - \bar{C}(\pi)\} + \frac{(t-s)(n-t+s)}{n}(p_{\pi_t} - p_{\pi_s})\right]. \qquad (5.2.5)$$

Hence the lemma follows from the equation (5.2.5). ■

Corollary 5.2.2 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be any sequence. Let π' be a sequence obtained from π by interchanging the two jobs π_i and π_{i+1} $(1 \le i < n)$ only. Then

$$\frac{n}{2}[V(\pi') - V(\pi)] = (p_{\pi_{i+1}} - p_{\pi_i})\Big[C_{[i]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_{i+1}} - p_{\pi_i})\Big]. \quad (5.2.6)$$

Proof : The proof is immediate from Lemma 5.2.1. ■

Theorem 5.2.3 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence. Then

(i) for any g, $2 \le g \le n$,

$$p_{\pi_g} < p_{\pi_{g-1}} \Rightarrow \bar{C}(\pi) \ \ge\ C_{[g-1]}(\pi) + \frac{n-1}{2n}(p_{\pi_g} - p_{\pi_{g-1}}), \quad (5.2.7)$$

(ii) for any h, $1 \le h < n$,

$$p_{\pi_h} < p_{\pi_{h+1}} \Rightarrow \bar{C}(\pi) \ \le\ C_{[h]}(\pi) + \frac{n-1}{2n}(p_{\pi_{h+1}} - p_{\pi_h}). \quad (5.2.8)$$

Proof : Let $p_{\pi_g} < p_{\pi_{g-1}}$ for some g, $2 \le g \le n$. Obtain the sequence π' from π by interchanging the two jobs π_{g-1} and π_g only. Using Corollary 5.2.2, we get

$$\frac{n}{2}[V(\pi') - V(\pi)]$$
$$= (p_{\pi_g} - p_{\pi_{g-1}})\Big[C_{[g-1]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_g} - p_{\pi_{g-1}})\Big]. \quad (5.2.9)$$

Since π is optimal, $V(\pi') - V(\pi) \ge 0$. Therefore, from the equation (5.2.9), we have

$$C_{[g-1]}(\pi) - \bar{C}(\pi) + \tfrac{n-1}{2n}(p_{\pi_g} - p_{\pi_{g-1}}) \le 0$$

or, $\bar{C}(\pi) \ge C_{[g-1]}(\pi) + \frac{n-1}{2n}(p_{\pi_g} - p_{\pi_{g-1}})$.

Hence the part (i) of the theorem holds.

The part (ii) can be proved by similar arguments. ∎

We have the jobset $N = \{1, 2, \ldots, n\}$ and $p_1 \geq p_2 \geq \cdots \geq p_n$. Let k be the index such that

$$p_1 \geq \cdots \geq p_{n-k} > p_{n-k+1} = \cdots = p_n.$$

Define Q $(Q \subseteq N)$ by $Q = \{n-k+1, n-k+2, \ldots, n\}$. Obviously, $|Q| = k \geq 1$. Let $\bar{Q} = N \backslash Q = \{1, 2, \ldots, n-k\}$. Note that $\bar{Q}$ can be empty, that is, all the p_j's have the same magnitude. However, in such case the CTV problem is trivially solved. We assume that $\bar{Q}$ is non-empty. Note that for $i \in Q$ and $j \in \bar{Q}$, we have $p_i < p_j$.

If $k = 1$, we say that the smallest job is unique. For $k \geq 2$, smallest job is said to be of multiplicity k.

The following result describes the positions of the jobs in Q in any optimal sequence.

Lemma 5.2.4 : Let π be an optimal sequence. Then

$$\left.\begin{array}{rcl} \pi &=& (\alpha, \pi_Q, \beta) \\ \\ \text{where} \quad \pi_Q &=& \text{a permutation of jobs in } Q \\ \text{and} \quad (\alpha, \beta) &=& \text{a permutation of jobs in } \bar{Q}. \end{array}\right\} \qquad (5.2.10)$$

Proof : If $k = 1$, the result is obvious.

Let $k > 1$ and $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$. Suppose, if possible, π is not of the form given by (5.2.10). Then there exist two indices r and s satisfying the following :

$$(i)\quad s - r \geq 2,$$
$$(ii)\quad \pi_r, \pi_s \in Q$$
$$\text{and } (iii)\quad \pi_{r+1}, \ldots, \pi_{s-1} \in \bar{Q}.$$

With such r and s, we have $p_{\pi_r} < p_{\pi_{r+1}}$ and $p_{\pi_s} < p_{\pi_{s-1}}$.

Now, using Theorem 5.2.3, we have

$$\bar{C}(\pi) \leq C_{[r]}(\pi) + \frac{n-1}{2n}(p_{\pi_{r+1}} - p_{\pi_r}) \tag{5.2.11}$$

$$\text{and} \quad \bar{C}(\pi) \geq C_{[s-1]}(\pi) + \frac{n-1}{2n}(p_{\pi_s} - p_{\pi_{s-1}}). \tag{5.2.12}$$

Combining the inequalities (5.2.11) and (5.2.12), we get

$$C_{[s-1]}(\pi) + \frac{n-1}{2n}(p_{\pi_s} - p_{\pi_{s-1}}) \leq C_{[r]}(\pi) + \frac{n-1}{2n}(p_{\pi_{r+1}} - p_{\pi_r})$$

$$\text{or,} \quad \left[p_{\pi_{r+1}} + \cdots + p_{\pi_{s-1}}\right] - \frac{n-1}{2n}p_{\pi_{r+1}} - \frac{n-1}{2n}p_{\pi_{s-1}} + \frac{n-1}{n}p_n \leq 0. \tag{5.2.13}$$

Since $s - r \geq 2$ and $p_j > p_n$ for $j \in \{\pi_{r+1}, \ldots, \pi_{s-1}\}$, the left-hand-side of the inequality (5.2.13) is strictly positive. Thus, we arrive at a contradiction.

Hence the theorem holds. ∎

Remark 5.2.5 : The above result shows that in any optimal sequence, all the smallest jobs must be adjacent. The same also follows from the V-shaped property of optimal sequence (refer to Theorem 2.2.5 of Chapter 2).

We use Lemma 5.2.4 to state the following result.

Theorem 5.2.6 : Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence with $\pi_i \in Q$ where $i \in \{s+1, \ldots, s+k\}$ for some s, $1 \leq s < n-k$. Then

$$(i)\ \bar{C}(\pi) \geq C_{[s]}(\pi) + \frac{n-k}{2n}(p_n - p_{\pi_s}) + \frac{k-1}{2}p_n,$$

$$(ii)\ \bar{C}(\pi) \leq C_{[s+k]}(\pi) + \frac{n-k}{2n}(p_{\pi_{s+k+1}} - p_n) - \frac{k-1}{2}p_n.$$

Proof : Let π' be the sequence obtained from π by interchanging the two jobs π_s and π_{s+k} only. Using Lemma 5.2.1, we have

$$\frac{n}{2}[V(\pi') - V(\pi)] = (p_{\pi_{s+k}} - p_{\pi_s})\left[\sum_{r=s}^{s+k-1}\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\} + \frac{k(n-k)}{2n}(p_{\pi_{s+k}} - p_{\pi_s})\right]. \tag{5.2.14}$$

Note that $p_{\pi_{s+k}} = p_n < p_{\pi_s}$, and because π is optimal, $V(\pi') - V(\pi) \geq 0$. Therefore, we can get from the equation (5.2.14),

$$\sum_{r=s}^{s+k-1}\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\} + \frac{k(n-k)}{2n}(p_n - p_{\pi_s}) \leq 0$$

$$\text{or,}\quad k\left[C_{[s]}(\pi) + \frac{k-1}{2}p_n\right] - k\bar{C}(\pi) + \frac{k(n-k)}{2n}(p_n - p_{\pi_s}) \leq 0$$

$$\text{or,}\quad C_{[s]}(\pi) + \frac{n-k}{2n}(p_n - p_{\pi_s}) + \frac{k-1}{2}p_n \leq \bar{C}(\pi).$$

Hence the part (i) of the theorem holds.

Next, let π'' be the sequence obtained from π by interchanging the two jobs π_{s+1} and π_{s+k+1} only. Again note that $p_{\pi_{s+1}} = p_n < p_{\pi_{s+k+1}}$ and $V(\pi'') - V(\pi) \geq 0$. We can argue similarly as in the case of part (i) to prove the part (ii). ∎

Corollary 5.2.7 : Suppose the smallest job is unique. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be an optimal sequence with $\pi_s = n$. Then

$$(i)\ \bar{C}(\pi) \geq C_{[s-1]}(\pi) - \frac{n-1}{2n}(p_{\pi_{s-1}} - p_n)$$

$$(ii)\ \bar{C}(\pi) \leq C_{[s]}(\pi) + \frac{n-1}{2n}(p_{\pi_{s+1}} - p_n).$$

Proof : The proof is immediate from Theorem 5.2.6. ∎

Remark 5.2.8 : From Theorem 5.2.6 or Corollary 5.2.7, we know that for any optimal sequence π, the average of job completion times, $\bar{C}(\pi)$, must be located in the neighborhood of the completion time of the smallest job, $C_n(\pi)$.

In the following, we prove the V-shaped property of optimal sequence (refer to Theorem 2.2.5 of Chapter 2) using the Lemma 5.2.4 and Theorem 5.2.6.

Proof of V-shaped Property : Let π be an optimal sequence. We know, from Lemma 5.2.4, that π must have the form $\pi = (\alpha, \pi_Q, \beta)$. In order to prove that π is V-shaped, it is enough to show that α (β) is in LPT (SPT) order.

Let $\alpha = (\pi_1, \pi_2, \ldots, \pi_s)$, $\pi_Q = (\pi_{s+1}, \pi_{s+2}, \ldots, \pi_{s+k})$ and $\beta = (\pi_{s+k+1}, \pi_{s+k+2}, \ldots, \pi_n)$.

Suppose α is not in LPT order. Then there exists an index i, $1 \leq i < s$, such that $p_{\pi_i} < p_{\pi_{i+1}}$. With such an i, obtain a sequence π' from π by interchanging the two jobs π_i and π_{i+1} only. Using Corollary 5.2.2, we have

$$\begin{aligned} \frac{n}{2}[V(\pi') - V(\pi)] &= (p_{\pi_{i+1}} - p_{\pi_i}) \Big[C_{[i]}(\pi) - \bar{C}(\pi) \\ &\quad + \frac{n-1}{2n}(p_{\pi_{i+1}} - p_{\pi_i}) \Big]. \end{aligned} \tag{5.2.15}$$

Now,

$$\begin{aligned} & C_{[i]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_{i+1}} - p_{\pi_i}) \\ \leq\ & C_{[i]}(\pi) - C_{[s]}(\pi) - \frac{n-k}{2n}(p_n - p_{\pi_s}) - \frac{k-1}{2}p_n + \frac{n-1}{2n}(p_{\pi_{i+1}} - p_{\pi_i}) \\ & \qquad \text{(using Theorem 5.2.6}(i)) \\ <\ & -\left[p_{\pi_{i+1}} + \cdots + p_{\pi_s}\right] + \frac{n-k}{2n}p_{\pi_s} + \frac{n-1}{2n}p_{\pi_{i+1}} \\ <\ & 0. \end{aligned}$$

Therefore, we get from the equation (5.2.15) that $V(\pi') < V(\pi)$ which is a contradiction to the optimality of π. Hence α must be in LPT order.

Next, suppose, if possible, β is not in SPT order. This implies that there are two jobs π_j and π_{j+1}, $s + k + 1 \leq j < n$ with $p_{\pi_j} > p_{\pi_{j+1}}$. By interchanging π_j and π_{j+1} in π, we get a new sequence π''.

Using Theorem 5.2.6(ii) and by similar arguments as in the previous case, we arrive at the contradiction that π'' is strictly better than the optimal sequence π. Hence β must be in SPT order.

This completes the proof. ∎

Remark 5.2.9 : Note that the necessary condition for optimality of a sequence given by Theorem 5.2.6 is stronger than the V-shaped property given by Eilon and Chowdhury [1977].

We now evaluate the change in CTV value for a sequence when a job is moved to a new position, but the relative order of the remaining jobs is not disturbed.

Lemma 5.2.10 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be any sequence and $\pi' = (\pi'_1, \ldots, \pi'_n)$ be the sequence obtained from π such that (i) $\pi'_r = \pi_r$ for $r = 1, \ldots, s-1, t+1, \ldots, n$, (ii) $\pi'_s = \pi_t$ and (iii) $\pi'_r = \pi_{r-1}$ for $r = s+1, \ldots, t$. Then

$$\frac{n}{2}\{V(\pi') - V(\pi)\} = \sum_{j=s}^{t-1}(p_{\pi_t} - p_{\pi_j})\Big[C_{[j]}(\pi) - \bar{C}(\pi) + \frac{1}{2}(p_{\pi_t} - p_{\pi_j}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r})\Big]. \quad (5.2.16)$$

Proof : It can be observed that

$$\begin{aligned} C_{[r]}(\pi') &= C_{[r]}(\pi) \quad \text{for } r = 1, \ldots, s-1, t, t+1, \ldots, n & (5.2.17)\\ &= C_{[r]}(\pi) + (p_{\pi_t} - p_{\pi_r}) \quad \text{for } r = s, s+1, \ldots, t-1 & (5.2.18)\end{aligned}$$

and therefore

$$\bar{C}(\pi') = \bar{C}(\pi) + \frac{1}{n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}) \quad (5.2.19)$$

Now, we have

$$
\begin{aligned}
& n\{V(\pi') - V(\pi)\} \\
&= \sum_{j=1}^{n}\left[C_{[j]}^2(\pi') - C_{[j]}^2(\pi)\right] - n\left[\bar{C}^2(\pi') - \bar{C}^2(\pi)\right] \\
&= \sum_{j=s}^{t-1}(p_{\pi_t} - p_{\pi_j})\left[2C_{[j]}(\pi) + (p_{\pi_t} - p_{\pi_j})\right] \\
&\qquad - \sum_{j=s}^{t-1}(p_{\pi_t} - p_{\pi_j})\left[2\bar{C}(\pi) + \frac{1}{n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r})\right] \\
&\qquad \text{(using the equations (5.2.17), (5.2.18) and (5.2.19))} \\
&= \sum_{j=s}^{t-1}(p_{\pi_t} - p_{\pi_j})\left[2C_{[j]}(\pi) - 2\bar{C}(\pi) + (p_{\pi_t} - p_{\pi_j})\right. \\
&\qquad\qquad \left. - \frac{1}{n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r})\right] \qquad (5.2.20)
\end{aligned}
$$

and hence the lemma follows from the equation (5.2.20). ∎

Lemma 5.2.11 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be a V-shaped sequence with $\pi_s = n$ for some $1 \leq s < n$ and

$$\bar{C}(\pi) \quad < \quad C_{[s]}(\pi) + \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s}), \qquad (5.2.21)$$

and $\pi' = (\pi'_1, \ldots, \pi'_n)$ be the sequence derived from π such that (i) $\pi'_r = \pi_r$ for $r = 1, \ldots, s-1, t+1, \ldots, n$, (ii) $\pi'_s = \pi_t$ and (iii) $\pi'_r = \pi_{r-1}$ for $r = s+1, \ldots, t$. Then $V(\pi') \geq V(\pi)$. Strict inequality holds if $p_{\pi_t} > p_{\pi_{t-1}}$.

Proof : Using Lemma 5.2.10, we get

$$\frac{n}{2}\{V(\pi') - V(\pi)\} \quad = \quad \sum_{j=s}^{t-1}(p_{\pi_t} - p_{\pi_j})X_j \qquad (5.2.22)$$

$$\text{where } X_j \quad = \quad C_{[j]}(\pi) - \bar{C}(\pi) + \frac{1}{2}(p_{\pi_t} - p_{\pi_j}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}). \quad (5.2.23)$$

Since π is V-shaped and $\pi_s = n$, we know that $p_{\pi_t} \geq p_{\pi_j}$ for all $j = s, \ldots, t-1$. Therefore, it is enough to prove that $X_j > 0$ for all $j = s, \ldots, t-1$. Now,

$$\begin{aligned}
X_s &= C_{[s]}(\pi) - \bar{C}(\pi) + \frac{1}{2}(p_{\pi_t} - p_{\pi_s}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}) \\
&= \left[C_{[s]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s})\right] - \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s}) \\
&\qquad + \frac{1}{2}(p_{\pi_t} - p_{\pi_{s+1}}) + \frac{1}{2}(p_{\pi_{s+1}} - p_{\pi_s}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}) \\
&> \frac{1}{2n}(p_{\pi_{s+1}} - p_{\pi_s}) + \frac{1}{2}(p_{\pi_t} - p_{\pi_{s+1}}) - \frac{t-s}{2n}p_{\pi_t} \\
&\qquad + \frac{1}{2n}p_{\pi_s} + \frac{1}{2n}\sum_{r=s+1}^{t-1} p_{\pi_r} \\
&\qquad \text{(using the equation (5.2.21))} \\
&= \frac{n - \overline{t-s}}{2n}p_{\pi_t} - \frac{n-1}{2n}p_{\pi_{s+1}} + \frac{1}{2n}\sum_{r=s+1}^{t-1} p_{\pi_r} \\
&\geq \frac{n - \overline{t-s}}{2n}p_{\pi_t} - \frac{n-1}{2n}p_{\pi_{s+1}} + \frac{t-s-1}{2n}p_{\pi_{s+1}} \\
&= \frac{n - \overline{t-s}}{2n}(p_{\pi_t} - p_{\pi_{s+1}}) \\
&\geq 0,
\end{aligned}$$

$$\begin{aligned}
X_{s+1} &= C_{[s+1]}(\pi) - \bar{C}(\pi) + \frac{1}{2}(p_{\pi_t} - p_{\pi_{s+1}}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}) \\
&= \left[C_{[s]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s})\right] + p_{\pi_{s+1}} - \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s}) \\
&\qquad + \frac{1}{2}(p_{\pi_t} - p_{\pi_{s+1}}) - \frac{1}{2n}\sum_{r=s}^{t-1}(p_{\pi_t} - p_{\pi_r}) \\
&> \frac{1}{n}p_{\pi_{s+1}} + \frac{n-1}{2n}p_{\pi_s} + \frac{n - \overline{t-s}}{2n}p_{\pi_t} + \frac{1}{2n}\sum_{r=s}^{t-1} p_{\pi_r} \\
&\qquad \text{(using the equation (5.2.21))} \\
&> 0
\end{aligned}$$

and for $s + 2 \leq j \leq t - 1$,

$$X_j = C_{[j]}(\pi) - \bar{C}(\pi) + \frac{1}{2}(p_{\pi_t} - p_{\pi_j}) - \frac{t-s}{2n}p_{\pi_t} + \frac{1}{2n}\sum_{r=s}^{t-1} p_{\pi_r}$$

$$= \left[C_{[s]}(\pi) - \bar{C}(\pi) + \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s})\right] + \sum_{r=s+1}^{j} p_{\pi_r} - \frac{n-1}{2n}(p_{\pi_{s+1}} - p_{\pi_s})$$
$$+ \frac{n-\overline{t-s}}{2n} p_{\pi_t} - \frac{1}{2} p_{\pi_j} + \frac{1}{2n} \sum_{r=s}^{t-1} p_{\pi_r}$$
$$> 0 \quad \text{(using the equation (5.2.21) and on simplification).}$$

This completes the proof. ∎

5.3 DEFINITIONS

Definition 5.3.1 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be a V-shaped sequence with $\pi_k = n$. The sequence π is said to be *saturated* if $\bar{C}(\pi) < C_{[k]}(\pi) + \frac{n-1}{2n}(p_{\pi_{k+1}} - p_{\pi_k})$. We call a sequence *unsaturated* if it is not saturated.

Definition 5.3.2 : Let $\pi = (\pi_1, \ldots, \pi_n)$ be a V-shaped sequence with $\pi_k = n$. The sequence $\pi' = (\pi_1, \ldots, \pi_{k-1}, \pi_j, \pi_k, \ldots, \pi_{j-1}, \pi_j, \ldots, \pi_n)$ is called a *first-order descendant* (FOD) of π if π' is V-shaped. We denote this relationship by $\pi \xrightarrow{d} \pi'$.

For V-shaped sequences $\pi^{(0)} \xrightarrow{d} \pi^{(1)} \xrightarrow{d} \pi^{(2)} \xrightarrow{d} \cdots \xrightarrow{d} \pi^{(r)}$, that is, $\pi^{(j)}$ is an FOD of $\pi^{(j-1)}$ for $j = 1, \ldots, r$, then $\pi^{(j)}$ is said to a descendant of $\pi^{(0)}$ for $j = 1, \ldots, r$.

Definition 5.3.3 : A V-shaped sequence $\pi = (\pi_1, \ldots, \pi_n)$ with $\pi_k = n$ is called an *end sequence* if $\pi_{k-1} = n - 1$.

An end sequence does not have any descendant.

Definition 5.3.4 : Let $\pi = (\pi_1, \ldots, \pi_n)$ and $\pi' = (\pi'_1, \ldots, \pi'_n)$ be two V-shaped sequences with $\pi_k = \pi'_k = n$ for some $k < \lfloor \frac{n}{2} \rfloor + 1$. The sequence π' is a *descendant-dual* (DD) of π if $\left\{\pi'_2, \pi'_3, \ldots, \pi'_{k-1}\right\} \subseteq \left\{\pi_{k+1}, \pi_{k+2}, \ldots, \pi_n\right\}$.

A DD of π is dual of a descendant of π. For example, with $n = 10$ the sequence $\alpha = (1, 3, 5, 7, 10, 9, 8, 6, 4, 2)$ is a DD of $\pi = (1, 2, 4, 6, 10, 9, 8, 7,$

5, 3). It can be seen that $\alpha^D = (1, 2, 4, 6, 8, 9, 10, 7, 5, 3)$ is dual of α and a descendant of π.

5.4 CONJECTURES

5.4.1 A Conjecture on New V-shaped Property

We now present some results which are derived in an attempt to prove a conjecture regarding the property of CTV function.

Lemma 5.4.1 : If π is saturated and π' is a descendant of π, then

$$(i) \;\; V(\pi') \geq V(\pi)$$

and (ii) π' is also saturated.

Proof : Let π' be obtained from π through the following chain of FODs

$$\pi \xrightarrow{d} \pi^{(1)} \xrightarrow{d} \pi^{(2)} \xrightarrow{d} \cdots \xrightarrow{d} \pi^{(r)} \xrightarrow{d} \pi'.$$

Using Lemma 5.2.11, we know that $V(\pi) \leq V(\pi^{(1)}) \leq V(\pi^{(2)}) \leq \cdots \leq V(\pi^{(r)}) \leq V(\pi')$. Thus, we have $V(\pi') \geq V(\pi)$.

It can be algebraically verified that if π is saturated, any FOD of π is also saturated. This is because, the increase in completion time of job n is larger than the increase in average completion time. It implies that $\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(r)}$ and π' are also saturated. ■

Theorem 5.4.2 : If $\pi = (\pi_1, \ldots, \pi_n)$ is unsaturated with $\pi_k = n$, then for the sequence $\pi' = (\pi_1, \ldots, \pi_{k-1}, \pi_{k+1}, \pi_k, \pi_{k+2}, \ldots, \pi_n)$, $V(\pi') \leq V(\pi)$.

Proof : The proof is immediate from Corollary 5.2.2. ■

Lemma 5.4.3 : Let π be saturated and α be a descendant-dual of π. Then $V(\alpha) \geq V(\pi)$.

Proof : Since α is a descendant-dual of π, its dual α^D is a descendant of π. We have $V(\alpha) = V(\alpha^D)$ from Theorem 2.2.4 (of Chapter 2) and $V(\pi) \leq V(\alpha^D)$ using Lemma 5.4.1. It means that $V(\pi) \leq V(\alpha)$. ∎

$$
\begin{aligned}
\text{Let } S_k &= \text{the set of all V-shaped sequences with} \\
&\quad\ \text{smallest job } (n) \text{ in } k\text{th position,} \\
A_k &= \text{the set of all saturated sequences in } S_k, \\
E_k &= \text{the set of all end sequences in } S_k \\
\text{and } T_k^* &= \min_{\pi \in S_k} V(\pi) \\
&\qquad\qquad\qquad \text{for } k = 1, 2, \ldots, n.
\end{aligned}
$$

Theorem 5.4.4 : $T_k^* = T_{n-k+2}^*$ for $k \geq 2$.

Proof : It follows directly from the duality theorem (see Theorem 2.2.4 of Chapter 2). ∎

By this result, we can confine ourself to the set $S_1 \cup S_2 \cup \cdots \cup S_m$ only where $m = \lfloor \frac{n}{2} \rfloor + 1$ in order to find an optimal sequence.

Theorem 5.4.5 : For any $1 \leq k < n$,

$$S_{k+1} = \cup_{\pi \in S_k \setminus E_k} D(\pi)$$

where $D(\pi)$ is the set of all FODs of π.

Proof : Consider an arbitrary V-shaped sequence $\pi = (\pi_1, \ldots, \pi_n)$ in S_{k+1}. If $\pi_k < \min\{\pi_{k+2}, \ldots, \pi_n\}$, let $\pi' = (\pi_1, \ldots, \pi_{k-1}, \pi_{k+1}, \ldots, \pi_n, \pi_k)$, otherwise find j such that $\pi_j > \pi_k > \pi_{j+1}$ and let $\pi' = (\pi_1, \ldots, \pi_{k-1}, \pi_{k+1}, \ldots, \pi_j, \pi_k, \pi_{j+1}, \ldots, \pi_n)$.

It is obvious that π' belongs to $S_k \setminus E_k$ and π is an FOD of π'. Therefore, $S_{k+1} \subseteq \cup_{\pi \in S_k \setminus E_k} D(\pi)$. Now, the result holds because any FOD of a sequence in $S_k \setminus E_k$ belongs to S_{k+1}. ∎

Theorem 5.4.6 : For each $\pi \in S_k \backslash (A_k \cup E_k)$, if every FOD of π is a descendant-dual of some $\alpha \in A_k$, then $T_k^* \leq T_{k+1}^*$.

Proof : Let $T_{k+1}^* = V(w^*)$, that is, w^* is optimal. By Lemma 5.4.5, w^* is an FOD of some sequence in $S_k \setminus E_k$. If w^* is an FOD of some sequence in A_k, we know from Lemma 5.4.1 that $T_{k+1}^* \geq T_k^*$. Suppose w^* is an FOD of some $\pi \in S_k \setminus (A_k \cup E_k)$. Therefore, π is a descendant-dual of some $\alpha \in A_k$. By applying Lemma 5.4.3, we note again that $T_k^* \leq T_{k+1}^*$. ∎

Theorem 5.4.7 : Let $T_k^* = V(w^*)$. If w^* is unsaturated, $T_k^* \geq T_{k+1}^*$.

Proof : Since w^* is unsaturated, we have $V(w') \leq V(w^*)$ by Theorem 5.4.2, where $w' = (w_1^*, \ldots, w_{k-1}^*, w_{k+1}^*, w_k^*, w_{k+2}^*, \ldots, w_n^*) \in S_{k+1}$, that is, $T_{k+1}^* \leq T_k^*$. ∎

We now present the conjecture which involves T_k^*'s, the minimum value of CTV for the fixed position of the smallest job.

Conjecture 5.4.8 : $T_{k+1}^* \not> \min\left\{T_k^*, T_{k+2}^*\right\}$ for any $1 \leq k \leq m-2$.

The above conjecture implies that

$$T_1^* \geq T_2^* \geq \cdots \geq T_m^* \tag{5.4.24}$$

or

$$T_1^* \geq T_2^* \geq \cdots \geq T_r^* \leq T_{r+1}^* \leq \cdots \leq T_m^* \quad \text{for some } r,\ 1 \leq r \leq m-1 \tag{5.4.25}$$

Using Theorem 5.4.4, we note that (a) if (5.4.24) holds, $T_m^* \leq T_{m+1}^* \leq \cdots \leq T_n^*$, (b) if (5.4.25) holds, $T_m^* \geq \cdots \geq T_{n-r+2}^* \leq \cdots \leq T_n^*$. It means that T_j^*'s are either V-shaped or W-shaped.

5.4.2 A Conjecture on Optimal Sequence

In this subsection, we describe a conjecture on optimal sequence for the CTV problem when the job processing times are in arithmetic progression (AP).

It can be seen that an optimal sequence remains optimal when all the p_j's are multiplied by a positive constant. However, it is not true for the addition of positive number to all the p_j's.

Let us denote the CTV problem (with p_j's as the processing times) by **P1** and consider the CTV problem (**P2**) with processing times as $q_j = a + bp_j$ for $j = 1, \ldots, n$, where $a \geq 0$, $b > 0$. Note that, if p_j's are the natural numbers, q_j's are in AP.

For any sequence π, let $V'(\pi)$ be the CTV value of **P2**.

The following result gives the relationship between the CTV values of **P1** and **P2**.

Theorem 5.4.9 : For any sequence $\pi = (\pi_1, \ldots, \pi_n)$,

$$V'(\pi) = \frac{1}{12}a^2(n^2-1) + b^2V(\pi) + \frac{ab}{n}\sum_{r=1}^{n}(r-1)(n-r+1)p_{\pi_r}. \quad (5.4.26)$$

Proof : Let $C'_{[r]}(\pi)$ and $\bar{C}'(\pi)$ be the completion time of rth job in π and the average of the completion times respectively for **P2**. Then, we have

$$\begin{aligned} C'_{[r]}(\pi) &= ra + bC_{[r]}(\pi) \quad \text{for } r = 1, \ldots, n \\ \text{and } \bar{C}'(\pi) &= \frac{n+1}{2}a + b\bar{C}(\pi). \end{aligned}$$

Therefore,

$$\begin{aligned} V'(\pi) &= \frac{1}{n}\sum_{r=1}^{n}\left[C'_{[r]}(\pi) - \bar{C}'(\pi)\right]^2 \\ &= \frac{1}{n}\sum_{r=1}^{n}\left[\left(r - \frac{n+1}{2}\right)a + b\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\}\right]^2 \end{aligned}$$

$$= \frac{1}{12}a^2(n^2-1) + b^2V(\pi) + \frac{2ab}{n}\sum_{r=1}^{n} r\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\}. \quad (5.4.27)$$

Now,

$$\begin{aligned}
&\sum_{r=1}^{n} r\left\{C_{[r]}(\pi) - \bar{C}(\pi)\right\} \\
&= \sum_{r=1}^{n} rC_{[r]}(\pi) - \frac{1}{2}n(n+1)\bar{C}(\pi) \\
&= \frac{1}{2}\sum_{r=1}^{n}(n+r)(n-r+1)p_{\pi_r} - \frac{1}{2}\sum_{r=1}^{n}(n+1)(n-r+1)p_{\pi_r} \\
&= \frac{1}{2}\sum_{r=1}^{n}(r-1)(n-r+1)p_{\pi_r}. \qquad (5.4.28)
\end{aligned}$$

Hence, using the equations (5.4.27) and (5.4.28), we get the relationship given by the equation (5.4.26). ■

Conjecture 5.4.10 : For $p_j = (n-j+1)$ for $j = 1, \ldots, n$, only optimal sequences for this problem are $\pi^* = (1, 2, 5, 6, \ldots, 8, 7, 4, 3)$ and its dual $(1, 3, 4, 7, 8, \ldots, 6, 5, 2)$.

That is, optimal sequence for different values of n are given as follows :

$$\begin{array}{ll}
(1,2,\ldots,n-3,n-2,n,n-1,\ldots,4,3) & \text{if } n \bmod 4 = 0 \\
(1,2,\ldots,n-4,n-3,n,n-1,n-2\ldots,4,3) & \text{if } n \bmod 4 = 1 \\
(1,2,\ldots,n-5,n-4,n-1,n,n-2,n-3,\ldots,4,3) & \text{if } n \bmod 4 = 2 \\
(1,2,\ldots,n-2,n-1,n,n-3,n-4,\ldots,4,3) & \text{if } n \bmod 4 = 3.
\end{array}$$

We can observe that π^* (as given in Conjecture 5.4.10) minimizes the last term on the right-hand-side of the equation (5.4.26). It then follows from Lemma 5.4.9 that if the above conjecture is true, π^* is optimal for **P2**.

REFERENCES

Ackley D. H. [1987] *A connectionist Machine for Genetic Hillclimbing*, Bluwer, Boston, Mass.

Aho A. V., Hopcroft J. E. and Ullman J. D. [1974] *The Design and Analysis of Computer Algorithms*, Addison, Wesley, Reading, Mass.

Bagchi U., Sullivan R.S. and Chang Y. [1987] "Minimizing squared deviation of completion times about a common due date", *Management Science*, 33, 894-900.

Bagchi U., Sullivan R.S. and Chang Y. [1989] "Simultaneous minimization of mean and variance of flowtime and waiting time in single machine systems", *Operations Research*, 37, 118-125.

Baker K. R. [1974] *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York.

Baker K. R. and Scudder G. D. [1990] "Sequencing with earliness and tardiness penalties : A review", *Operations Research*, 38, 22-38.

Bazaraa M. S. and Shetty C. M. [1979] *Non-linear Programming – Theory and Applications*, John Wiley, New York.

Bector C. R., Gupta Y. P. and Gupta M. C. [1988] "Determination of optimal common due date and optimal sequence in a single machine job shop", *International Journal for Production Research*, 26, 613-628.

Chakravarthy S. [1986] "A single machine scheduling problem with random processing times", *Naval Research Logistics Quarterly*, 33, 391-397.

Cheng T. C. E. and Cai X. [1993] "On the complexity of completion time variance minimization problem", *Asia Pacific Journal of Operational Research*, 10, 109-120.

Cheng T. and Gupta M. [1989] "Survey of scheduling research involving due date determinations decisions", *European Journal of Operational Research*, 38, 156-166.

Conway R. W., Maxwell W. L. and Miller L. W. [1967] *Theory of Scheduling*, Addison-Wesley, Reading, Mass.

De P., Ghosh J. B. and Wells C. E. [1989] "A note on the minimization of mean squared deviation of completion times about a common due date", it Management Science, 35, 1143-1147.

De P., Ghosh J. B. and Wells C. E. [1990] "Scheduling about a common due date with earliness and tardiness penalties", *Computers and Operations Research*, 17, 231-241.

De P., Ghosh J. B. and Wells C. E. [1992] "On the minimization of completion time variance with a bicriteria extension", *Operations Research*, 40, 1148-1155.

Eilon S. and Chowdhury I. G. [1977] "Minimizing waiting time variance in single machine problem", *Management Science*, 23, 567-575.

Emmons H. [1987] "Scheduling to a common due date on parallel common processors", *Naval Research Logistics Quarterly*, 34, 803-810.

Federgruen A. and Mosheiov G. [1993] "Simultaneous optimization of efficiency and performance balance measures in single machine scheduling problems", *Naval Research Logistics*, 40, 951-970.

French S. [1982] *Sequencing and Scheduling : An introduction to the mathematics of the jobshop*, Ellis Horwood Ltd., West Sussex, England.

Garey M. R. and Johnson D. S. [1979] *Computers and Intractability : A guide to the Theory of NP-Completeness*, Freeman.

Goldberg D. E. [1989] *Genetic algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass.

Gupta M. C., Gupta Y. P. and Bector C. R. [1990] "Minimizing the flow-time variance in single machine systems", *Journal of Operational Research Society*, 41, 767-779.

Gupta M. C., Gupta Y. P. and Kumar A. [1993] "Minimizing the flow-time variance in a single machine system using genetic algorithms", *European Journal of Operational Research*, 70, 289-303.

Gupta S. and Kyparisis J. [1987] "Single machine scheduling research", *OMEGA*, 15, 207-227.

Hall N. G. and Kubiak W. [1993] "Proof of a conjecture of Schrage about the completion time variance problem", *Operations Research Letters*, 10, 467-472.

Hall N. G., Kubiak W. and Sethi S. P. [1991] "Earliness-Tardiness scheduling problems – II : Deviation of completion times about a restrictive common due date", *Operations Research*, 39, 847-856.

Hardy G. H., Littlewood J. E. and Polya G. [1952] *Inequalities*, Second edition, New York : Cambridge University Press.

Horowitz E. and Sahni S. [1978] *Fundamentals of Computer Algorithms*, Computer Science Press, Potomac, Md.

Johnson S. M. [1954] "Optimal two- and three- stage production schedules with setup times included", *Naval Research Logistics Quarterly*, 1, 61-68.

Kahlbacher H. G. [1989] "SWEAT – A program for scheduling problems with earliness and tardiness penalties", *European Journal of Operational Research*, 43, 111-112.

Kanet J. J. [1981] "Minimizing variance of flowtime in single machine systems", *Management Science*, 27, 1453-1459.

Kubiak W. [1993] "Completion time variance minimization on a single machine is difficult", *Operations Research Letters*, 14, 49-59.

Kubiak W. [1995] "New results on the completion time variance minimization", *Discrete Applied Mathematics*, 58, 157-168.

Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G. and Shmoys D. B. [1990] *Sequencing and Scheduling : Algorithms and Complexity – a handbook of operations research.*

Lenstra J. K., Rinnooy Kan A. H. G. and Van Emde Bas P. (eds.) [1978] *Interface between Computer Science and Operations Research*, Mathematich Centrum, Amsterdam.

Manna D. K. and Prasad V.R. [1993] "Scheduling on single machine to minimize variance of completion times", Technical Report No. 15, Studies in Quality and Optimization, Indian Statistical Institute, Madras, India.

Manna D. K. and Prasad V.R. [1994] "Minimization of completion time variance using a dominance criterion and a lower bound", Technical Report No. 21, Studies in Quality and Optimization, Indian Statistical Institute, Madras, India.

Manna D. K. and Prasad V.R. [1995] "Pseudopolynomial algorithms for CTV minimization in single machine scheduling", Technical Report No. 27, Studies in Quality and Optimization, Indian Statistical Institute, Madras, India.

Merten A. G. and Muller M. E. [1972] "Variance minimization in single machine sequencing problems", *Management Science*, 18, 518-528.

Mittenthal J., Raghavachari M. and Rana A. [1993] "A hybrid simulated annealing approach for single machine scheduling problems with non-regular penalty functions", *Computers and Operations Research*, 20, 103-111.

Mittenthal J., Raghavachari M. and Rana A. [1994] "On modelling scheduling problems that simultaneously minimize mean and variance of completion times" , Submitted for publication.

Panwalker S., Smith M. and Siedmann A. [1982] "Common due date assignment to minimize total penalty for the one machine scheduling problem", *Operations Research*, 30, 391-399.

Papadimitriou C. H. and Steiglitz K. [1982] *Combinatorial Optimization : Algorithms and Complexity*, Prentice Hall Inc., Eaglewood Cliffs, New Jersey.

Prasad V. R. and Manna D. K. [1993] "Minimizing variance of completion times with stochastic jobs", Technical Report No. 17, Studies in Quality and Optimization, Indian Statistical Institute, Madras, India.

Prasad V. R., Manna D. K. and Arthanari T. S. [1994] "A note on lower bound for completion time variance in single machine scheduling", Technical Report No. 22, Studies in Quality and Optimization, Indian Statistical Institute, Madras, India.

Raghavachari M. [1988] "Scheduling problems with non-regular penalty functions : A review", *Opsearch*, 25, 144-164.

Rinnooy Kan A. H. G. [1976] *Machine scheduling problems : classification, complexity and computation*, Nijhoff, The Hague, Netherlands.

Schrage L. [1975] "Minimizing the time-in-system variance for a finite jobset", *Management Science*, 21, 540-543.

Sen T. and Gupta S. [1984] "A state-of-the art survey of static scheduling research involving due dates", *OMEGA*, 12, 62-76.

Smith W. E. [1956] "Various optimizers for single-stage production", *Naval Research Logistics Quarterly*, 3, 59-66.

Ullman J. D. [1976] *Complexity of sequencing problem*, Computer and job shop scheduling theory, 139-164, Coffman Jr E. G. (ed.), John Wiley & Sons, New York.

Vani V. and Raghavachari M. [1987] "Deterministic and random single machine sequencing with variance minimization", *Operations Research*, 35, 111-120.

Weng M. X. and Ventura J. A. [1994] "A quadratic integer programming method for minimizing the mean squared deviation of completion times", *Operations Research Letters*, 15, 205-211.

Some Corrections on Typographical Mistakes in the thesis entitled "ON CTV MINIMIZATION IN SINGLE MACHINE SCHEDULING" of D. K. Manna

---

1. Page 2 Line ↓ 7 : Replace " [1988]. " by " [1988], ".
2. Page 8 Line ↓ 3 : Replace " $(n - r + 1)$ " by " $(n - i + 1)$ ".
3. Page 22 Line ↑ 4 : Replace " $\pi = (\alpha, \gamma) \cdots$ sequence. " by " $\pi = (\gamma_1, \alpha, \gamma_2)$ and $\pi' = (\gamma_1, \beta, \gamma_2)$ where γ_1 (γ_2) is in LPT (SPT) order. ".
4. Page 24 Line ↓ 6 : Replace " $W_1 \leftarrow W_2$ " by " $W_1 \leftarrow W_2; W_2 \leftarrow \phi$ ".
5. Page 30 Line ↑ 8 : Replace " $W_1 \leftarrow W_2$ " by " $W_1 \leftarrow W_2, W_2 \leftarrow \phi$ ".
6. Page 31 Line ↓ 9 : Replace " $o\left(n^2\{(p_2 - p_3) + \sum_{j=4}^{n}(j-2)p_j\}\right)$ " by " $o\left(n\{(p_2 - p_3) + \sum_{j=4}^{n}(j-2)p_j\}\right)$ ".
7. Page 45 Line ↑ 7 : Replace " $\cdots \boldsymbol{x}^T \boldsymbol{D}\boldsymbol{x}$ " by " $\cdots \boldsymbol{x}^T \boldsymbol{D}\boldsymbol{x}$ (which is equivalent to $\frac{1}{n^2}x^T Dx$) ".
8. Page 92 Line ↑ 10 : Replace " $\cdots$ lower and " by " $\cdots$ lower and upper ".
9. Page 99 Line ↓ 10 : Replace " Kanet [5] " by " Kanet [1981] ".
10. Page 122 Line ↑ 9 : Replace " **Theorem** " by " **Lemma** ".
11. Page 124 Line ↑ 9 : Replace " **Theorem** " by " **Lemma** ".