UNIVERSITY OF CALGARY

Scheduling to Optimize Due Date Performance under Uncertainty of Processing Times

by

Anubhuti Parajuli

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL AND MANUFACTURING ENGINEERING

CALGARY, ALBERTA

MARCH, 2010

Abstract

This thesis contributes to the theoretical and practical aspects of scheduling research. It is dedicated to the analysis of scheduling a set of jobs on a single machine when the jobs have uncertain processing times and conformance to the due date is the performance objective. The findings reveal that scheduling based on the point estimates of the processing times, when times are actually uncertain, may not lead to an optimal job sequence. As well, the decisions as to when each job should start on the machine may not be optimal. These decisions are important when costs are associated with both early and tardy completion of jobs. A stochastic scheduling methodology, based on sampling using simulation and optimization using evolutionary search, has been introduced. Results and behaviour have been evaluated and compared with single-machine deterministic scheduling, based on optimization using point estimates. Furthermore, the methodology has also been extended to the two-machine flow shop problem. Results confirm performance improvement using stochastic scheduling.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. S.T. Enns and Dr. Diane Bischak for their relentless support in my research and the guidance provided in authoring this thesis.

I would also like to thank Dr. Theodor Freiheit, Dr. Jaydeep Balakrishnan and Dr. Paul Tu of my examining committee for their efforts in examining this thesis.

Dedication

This thesis is dedicated to my twin daughters Ayushi and Arya who were missed all through my graduate study at the University of Calgary.

Table of Contents

Approval Page ii
Abstract iii
Acknowledgements iv
Dedication v
Table of Contents vi
List of Tables viii
List of Figures and Illustrations x

CHAPTER ONE: INTRODUCTION 1
1.1 The Problem of Job Scheduling 3
1.1.1 Basic problem variations 4
1.1.2 Due date setting 8
1.1.2.1 Due date performance measures 10
1.1.3 Models and Methodologies 12
1.2 Research Motivation 13

CHAPTER TWO: LITERATURE REVIEW 16
2.1 Brief Historical Overview 16
2.2 Single Machine Schedule – State of the Art 18
2.2.1 Overview on single machine scheduling approaches 24
2.3 Two-machine Flow shop – State of the Art 27

CHAPTER THREE: RESEARCH METHODOLOGY 29
3.1 Mathematical Modeling using Spreadsheets 29
3.2 Modeling the Research Problem 32
3.3 Performance Measures 42
3.3.1 Earliness-Tardiness weighting ratio 44

CHAPTER FOUR: SINGLE MACHINE PROBLEM ANALYSIS 49
4.1 Single-Machine, Single-Job Problem 50
4.1.1 Optimal idle time with linear penalties 50
4.2 Single-machine, Two-job problem 55
4.2.1 Schedule based on expected times (deterministic) 56
4.2.2 Scheduling based on stochastic times 60
4.2.3 Influence of processing time variability 65
4.2.3.1 Scheduling when expected values of processing times are identical 66
4.2.3.2 Scheduling when expected values of processing times are different 70
4.3 Analytical solution of a two-job stochastic problem 76
4.4 Summary 91

CHAPTER FIVE: TWO - MACHINE FLOW SHOP PROBLEM ANALYSIS 93
5.1 Description of the problem 94
5.2 Analysis of the Problem 97
5.2.1 Scheduling under identical task time variability 98
5.2.2 Scheduling under varying task time variability 107
5.3 Summary 112

CHAPTER SIX: CONCLUSION 114
6.1 Contributions 114
6.2 Further extensions 117

BIBLIOGRAPHY 119

APPENDIX A: MATHEMATICA CODE 123

APPENDIX B: SNAPSHOT OF A TWO-MACHINE SPREADSHEET MODEL 129

APPENDIX C: FORMULA PRINT OF A SINGLE MACHINE MODEL 130

List of Tables

Table 2.1 Spreadsheet compatible simulation-optimization packages 26

Table 4.1 Single-machine, single-job results at several values of α 52

Table 4.2 Expected loss using Unit Normal Distribution Table, $\alpha = 0.5$.......................... 54

Table 4.3 Input table for single-machine, two-job problem 56

Table 4.4 Deterministic solution with and without inserted idle times when $\alpha = 0.5$ 57

Table 4.5 Deterministic solution with and without inserted idle times, $\alpha = 1.5$ 59

Table 4.6 Stochastic solution with and without inserted idle times, $\alpha = 0.5$ 61

Table 4.7 Comparing deterministic and stochastic solution at several values of α using linear measure.. 62

Table 4.8 Comparing deterministic and stochastic solution at several values of α using quadratic measure.. 63

Table 4.9 Deterministic problem solution under identical processing times, $\alpha = 0.5$ 66

Table 4.10 Stochastic scheduling under identical expected processing times, $\alpha = 0.5$ 67

Table 4.11 Results demonstrating job variability effects on two sequences, $\alpha = 0.5$....... 71

Table 4.12 Inputs for predictive relationship analysis 73

Table 4.13 Early and tardy jobs scenario for a two job problem..................................... 76

Table 4.14 Total linear penalty and due date ranges for the two-job realizations............ 77

Table 4.15 Comparison of simulation and analytical results for a uniform distribution .. 90

Table 5.1 Example inputs with unidentical average machine loadings ($\rho_1 < \rho_2$) and identical task time variability 98

Table 5.2 Example inputs with identical average machine loadings ($\rho_1 = \rho_2$) and identical task time variability 104

Table 5.3 Example inputs with unidentical average machine loadings ($\rho_1 > \rho_2$) and identical task time variability 105

Table 5.4 Total quadratic penalty variation against machine loading 106

Table 5.5 Example input with different task time variability on second machine 107

Table 5.6 Results using the two sequences with different task time variability on second machine 109

Table 5.7 Results using the two sequences with different task time variability on first machine 110

List of Figures and Illustrations

Figure 1.1 Sequencing and scheduling in a decision hierarchy 4

Figure 1.2 Sequence dependent setup times 5

Figure 1.3 Basic scheduling problem variations 7

Figure 1.4 Regular and non-regular performance measures 11

Figure 1.5 Gantt chart illustration of a scheduling problem 12

Figure 3.1 Conceptual diagram of a mathematical model 30

Figure 3.2 Snapshot of a single-machine spreadsheet model 34

Figure 3.3 Snapshot of Evolutionary Solver dialog box 35

Figure 3.4 Model flowchart 37

Figure 3.5 Gantt chart illustration of initial and final solution 38

Figure 3.6 Verification of optimal idle time 40

Figure 3.7 Linear penalty function with different α 46

Figure 3.8 Quadratic penalty function with different α 46

Figure 3.9 Quadratic penalty function with normal distribution of processing time overlayed 47

Figure 4.1 Optimal idle time under deterministic processing time, all α 51

Figure 4.2 Optimal idle time under stochastic processing time, $\alpha = 0.5$ 51

Figure 4.3 Variations of total penalty with due dates in a deterministic problem 58

Figure 4.4 Variation of optimal idle time and total penalties for several α in a stochastic problem 65

Figure 4.5 Illustration of the propagation of uncertainty 68

Figure 4.6 Scheduling jobs under identical means but different variability 69

Figure 4.7 Illustration of job variability effect on the two sequences 71

Figure 4.8 Variation of optimal sequence switchover with difference on mean processing times (δ) of the jobs 73

Figure 4.9 Predictive relationship plot of variability against difference of mean processing times at switchover 75

Figure 4.10 Two job problem realizations 77

Figure 4.11 Representation of a two job problem with uniformly distributed processing times 79

Figure 5.1 Gantt chart illustration of a two machine two jobs flow shop 94

Figure 5.2 Deterministic scheduling example with unidentical average machine loading, $\rho_1 < \rho_2$ 99

Figure 5.3 Stochastic scheduling example with unidentical average machine loading, $\rho_1 < \rho_2$ 101

Figure 5.4 Illustration of job or machine waiting 103

Figure 5.5 Stochastic scheduling example with identical average machine loadings, $\rho_1 = \rho_2$ 105

Figure 5.6 Stochastic scheduling example with unidentical average machine loadings, $\rho_1 > \rho_2$ 106

Figure 5.7 Stochastic scheduling example under different task time variability 108

Figure 5.8 Illustration of optimal sequence switchover in a two machine problem against the task time variability on the second machine 109

Figure 5.9 Illustration of optimal sequence switchover in a two machine problem against the task time variability on the first machine 111

CHAPTER ONE: INTRODUCTION

The purpose of scheduling is to allocate scarce resources to activities with the objective of optimizing one or more performance measures. A good scheduling algorithm will enable a company to remain competitive by lowering costs. In a manufacturing process, the resources are usually the machines or operators available and the activities are the processing tasks that need to be performed on these resources. Three performance objectives that are prevalent in scheduling theory and practice (Baker and Trietsch, 2007) are a) Turnaround time b) Due-date performance and c) Throughput. The turnaround time is related to flowtimes and therefore affects holding costs. Due date performance relates to costs associated with tardy deliveries and the throughput relates to how much fixed costs needs to be allocated per unit of production.

When perfect problem information is available *a priori*, scheduling problems are called deterministic problems. With lack of perfect information, scheduling must be done based on expectations or probabilities. These are called stochastic problems. A class of stochastic problems where the processing times of the jobs are not known with certainty but the mean is assumed to be known is often solved in a deterministic manner. However, recent research has demonstrated this approach may not yield optimal schedules or even optimal sequences.

This research looks specifically at comparing stochastic scheduling performance with deterministic scheduling, given that the problem involves stochastic processing times.

Second, the focus is on due date performance as a scheduling objective, considering that both early and tardy completion is undesirable. It should be noted that minimization of earliness indirectly addresses the turnaround objective as well. Holding costs associated with turnaround times will be reduced as earliness is reduced. Third, this research examines the optimal scheduling of jobs as opposed to only the optimal sequencing. Therefore the optimal starting time of jobs is taken into account whenever due dates are relatively loose. This area of investigation has become increasingly important and emanates from the Just-In-Time (JIT) philosophy of manufacturing.

Past research has focussed mainly on tardiness as a performance criteria using single machine models. Little comparative evaluation using deterministic and stochastic scheduling approaches has been done. This study also uses the single machine model but undertakes comparative analysis of the two scheduling approaches, thus extending the current literature. As a further extension, a two machine flow shop model is also investigated. An experimental framework is used to investigate the scheduling behaviour. However, analytical techniques are also used for verification wherever possible. Although the methodology developed is flexible, the problems investigated primarily involve a common due date for all jobs. This approach, common in the literature, makes it easier to achieve fundamental insights.

In the following sections background information on scheduling and the motivation for this research is provided.

1.1 The Problem of Job Scheduling

Consider the following examples which could reflect a real situation. A custom bicycle manufacturer receives an order for several different models due on the same date. Each model requires assembly at the same station. Assembly times are stochastic. Late completion requires expedited delivery and loss of good will. Early completion results in increased storage and other holding costs. The scheduling problem is one of determining a sequence that will minimize earliness and tardiness costs. If the due dates are relatively loose, then additional decisions need to be made as to when to start the assembly operations.

As illustrated by this example, scheduling can be defined as a decision making process which involves the optimal allocation of limited resources to tasks over time so as to meet one or more performance objectives. In a typical decision-making hierarchy, illustrated in Figure 1.1, job sequencing and scheduling can be placed under the scope of operational planning. These are short term decisions. Tactical planning, which involves medium term decisions might involve determining the resources needed and the jobs to be scheduled. Strategic planning might involve the long term acquisition of facilities and resources. Decisions across all levels should be consistent.

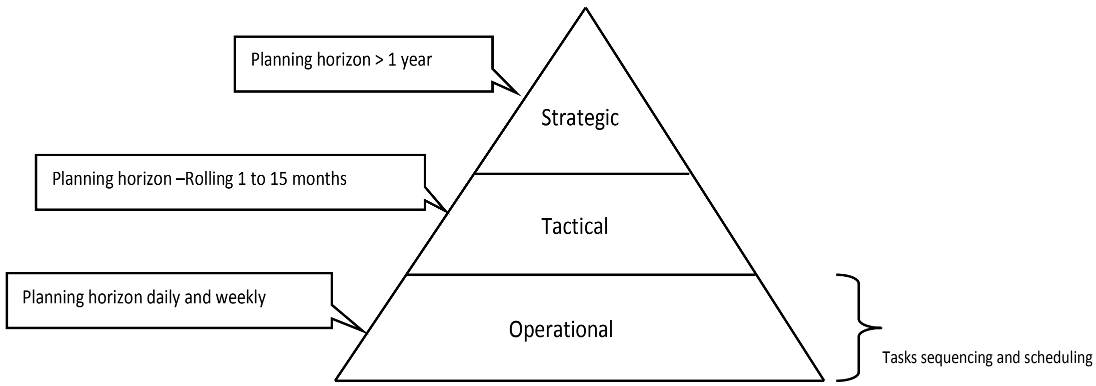


Figure 1.1 Sequencing and scheduling in a decision hierarchy

Information about the resources and the jobs to be processed is required to formally define the scheduling problem. A sequence refers only to the order in which jobs are to be processed on a given machine. A schedule considers the specific time interval over which and job will be processed and is thus broader, encompassing sequencing. An optimal solution needs to be derived from an allocation method which places each job to be processed at specific times on the resources, in a sequence that results in a best output for the chosen performance measure.

1.1.1 Basic problem variations

Two ways of viewing the job processing time, which is the time that each job spends at a resource, are common. When the job processing time is certain and a known constant, the processing time is said to be deterministic. The total processing time of a job, may however, be influenced by several factors. For example, in batch manufacturing, the job could be a lot of parts. In this case lot sizes determine the processing time of the job, or batch. Each item within the batch involves an elementary operation contributing to the total batch (job) processing times. If the time to perform each of these elementary operation is certain and known, then the total processing time for the batch is also deterministic and certain.

In batch production problems, it is also customary to lump the machine setup times in with the processing times. This is appropriate when the setup times are constant and independent of job sequences. When the setup times are sequence-dependent, as shown in Figure 1.2, it becomes necessary to consider these setup times separately.

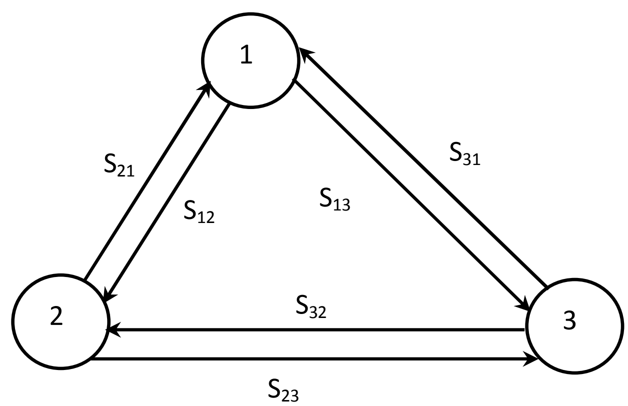


Figure 1.2 Sequence dependent setup times

The bi-directional arcs in Figure 1.2 represent the set up times going from one job to the other and the nodes represent three available jobs. S_{ij} is the setup time going from job i to job j. If for a given job j the S_{ij} values are all identical, the setups are sequence independent and can be added to the job processing times during schedule construction.

The total processing time (job processing time + set-up time) is a constant (deterministic) if both of these quantities are known.

However, processing times or setup times cannot always be determined with certainty. Uncertainties in total processing times are common in manufacturing and could arise from different sources. For example, random variations in the performance or speed of a machinist, non-homogeneous batch sizes or unreliable machines can result in uncertainty. Although exact processing times may be unknown, it is however often possible to estimate the distribution of these times. Scheduling can then be based on the laws of probability governing these random distributions. Furthermore, even if exact form of the distributions are not known, use of the first two moments of the distribution (mean and variance) can usually be estimated and used to model uncertainty. In fact, rapid modeling software packages, which are based on queuing heuristics, make use of the mean and the variance without any assumptions on the shape of the distribution.

The arrival time, or the release time, is the time the job becomes available for processing. It is the earliest time that a job can begin its operation on a machine. If all the jobs to be scheduled are available at the time of scheduling, the scheduling problem is said to be a static problem. In a static problem, no rescheduling is done even if more jobs arrive after processing has been initiated. A more complex problem is a dynamic situation in which jobs arrive continuously over time and the schedule is re-examined every time new information is available, such as a new job arrival. Knowledge of the job interarrival time pattern is therefore beneficial for a dynamic scheduling. However, it should be noted that in dynamic scheduling the arrival time and the release time are not necessarily the same. For example, if work load balancing is an issue, jobs may be held in a ‘job pool’ after arrival before releasing them to the floor. This type of input control is another dimension of the dynamic scheduling problem (Ramasesh, 1990).

Based on the processing time and job arrival characteristics, scheduling problems can therefore be classified into four basic types. These are shown in the four quadrants of Figure 1.3.

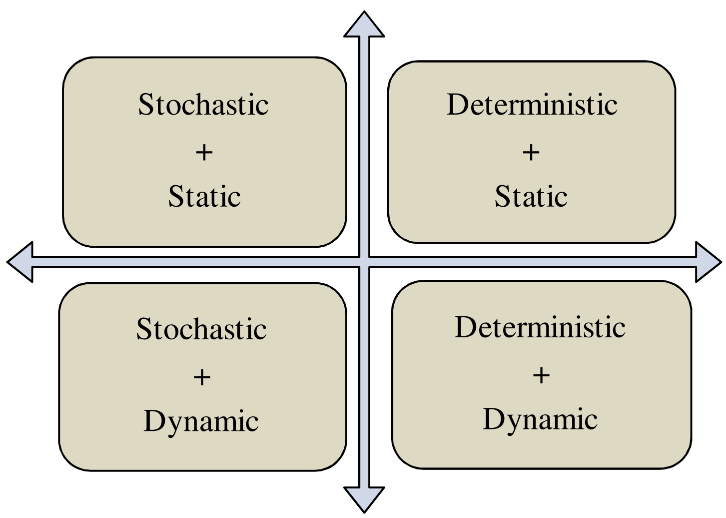


Figure 1.3 Basic scheduling problem variations

The focus of this research is on the class of problem identified in the upper left quadrant of this matrix. These are problems with static arrivals and stochastic processing times. Understanding these problems thoroughly is a prerequisite to dealing the problems identified in the lower left quadrant, with stochastic processing times and dynamic arrivals.

1.1.2 Due date setting

Two due date setting practices, based on the manufacturing environment that a company operates in, are common. In the make-to-stock (MTS) environments, the due dates are a discretionary decision of the manufacturer. They are set so as to maintain a balance between the levels of inventory against the level of delivery service to the customer. In the make-to-order (MTO) environment, the due dates are negotiable with the customer and hence need to be quoted.

Although in general scheduling models involving due date performance assume that the due dates are given, in practice companies operating in a MTO environment need an effective approach for quoting due dates. The following are among the three due date setting rules found in the literature. These rules are based on job characteristics and can only take the shop workload status into account indirectly through the selection of the parameters used in the rule:

i) Total work content (TWK) rule: In the TWK approach the due date of a job i, d_i, is estimated by adding a multiple b times the sum of the processing time p_i and set up time s_i to the job arrival time a_i. The value of b determines the degree of tightness of the due date. This can be mathematically expressed as in Equation 1.1

$$d_i\,(TWK) \;=\; a_i + b.(p_i + s_i) \tag{1.1}$$

ii) Slack allowance (SLK) rule: If b is set equal to 1 in Equation 1.1, and equal slack allowance γ is introduced for all the jobs instead, then the due date will be based on the slack allowance (SLK). The due date tightness is then dependent on the values of the slack factor γ, as is given by Equation 1.2.

$$d_i\ (SLK) = a_i + \gamma + p_i + s_i \tag{1.2}$$

iii) Constant flow allowance (CON) rule: If due dates are based on a constant flow allowance, β, irrespective of job processing times, then the due date can be expressed as Equation 1.3.

$$d_i\ (CON) = a_i + \beta \tag{1.3}$$

When several jobs need to be completed at the same time then the scheduling models are said to have a common due date. The common due date model applies in many production environments. For instance, one customer order may involve more than one product to be delivered at the same time. As well, in assembly operations multiple component types may need to be ready for assembly at the same time. The common due date model is therefore practically relevant. As well, it is simpler analytically and results in due date behaviour that is easier to understand. Therefore, common due date problems are often studied separately from distinct due date problems.

1.1.2.1 Due date performance measures

The performance measures used to evaluate a schedule are quantitative measures based on decision goals. In considering due date performance, the objective to be minimized is always a function of the actual completion times of the jobs relative to the due dates. If the completion time of a job i is denoted by C_i, then the job lateness L_i is defined as Equation 1.4.

$$L_i = C_i - d_i \tag{1.4}$$

The tardiness of a job is the positive lateness, given as Equation 1.5.

$$T_i = \max(L_i, 0) = \max(C_i - d_i, 0) \tag{1.5}$$

The earliness, is then expressed as Equation 1.6.

$$E_i = \max(0, d_i - C_i) \tag{1.6}$$

Traditional measures of performance focus on single measures which are non-decreasing functions of job completion times. These belong to a class of performance measures called regular measures. The relationship of Tardiness, T_i and the number of tardy jobs, U_i, which are the regular performance measures to the completion times, C_i is illustrated in Figures 1.4 (a) and (b).

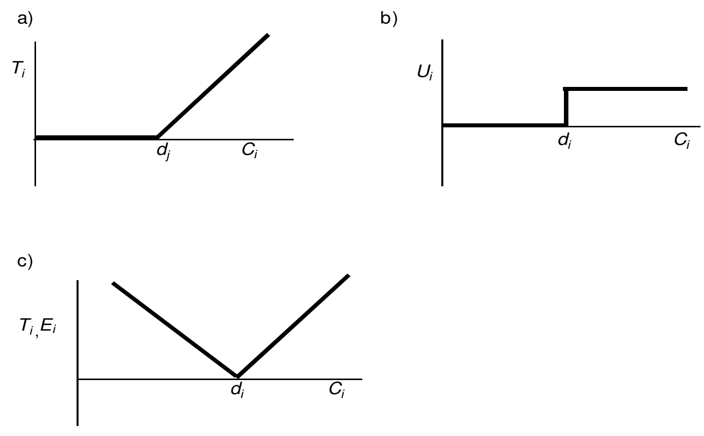


Figure 1.4 Regular and non-regular performance measures

Regular measures of performance are monotone in job completion times. In other words, they are non-decreasing as completion time increases. In contrast, a different class of performance measure, called non-regular measures, is illustrated in Figure 1.4 (c). Non-regular measures are dual measures which may either increase or decrease with increasing completion times. Non-regular measures are more difficult to deal with than regular measures.

When optimizing non-regular due date performance, penalties (costs) are associated with any deviation of completion times from the due dates. With dual measures (non-regular), such as the earliness-tardiness measure, penalties may be applied at different rates for earliness and tardiness. This is achieved by having different weights for early or late job deviations from the due date. An earliness-tardiness weighting ratio, or multiplier, can be used for weighting one relative to the other. Details on the performance measures used in this research, involve weightings, are dealt with in Chapter 3.

1.1.3 Models and Methodologies

Scheduling models help to analyze and compare the different schedules. These models essentially capture the resource and the tasks specifications and any technological constraints in the problem. The Gantt chart developed by Henry L. Gantt in the early 1900s is one of the simplest and most widely used models for sequencing jobs on the available resources. Figure 1.5 shows the Gantt chart relationships for a 4-job, single task, 2-machine scheduling problem. The Gantt chart is a useful visual aid to understand the necessary relationships to consider in scheduling jobs. Optimality or the goodness of the solution can be gleaned by considering different rearrangements of the jobs. Obviously, optimization using Gantt charts alone is limited to small problems.

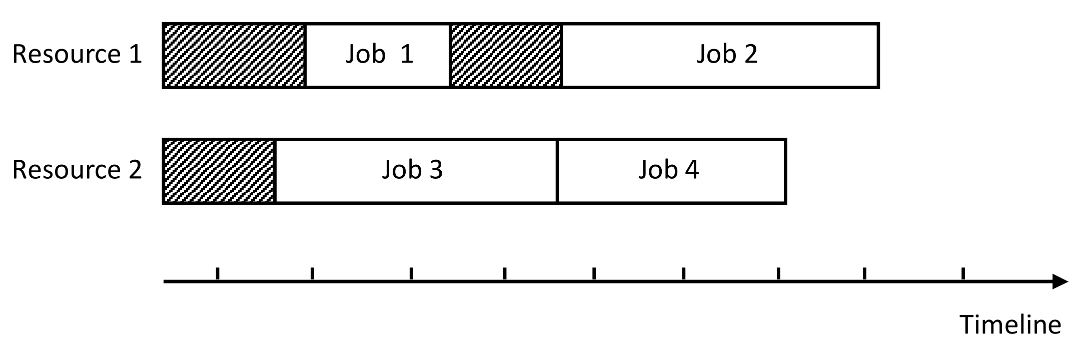


Figure 1.5 Gantt chart illustration of a scheduling problem

Although they do not give the same visual clarity as with the graphical models, mathematical models form the essence of scheduling theory. When building a mathematical model, scheduling is viewed as an optimization problem subject to constraints. The mathematical model requires translation of the decision goal into an explicit objective function as well as a full description of the resources and jobs.

The methodologies in use for scheduling problems can be broadly categorized as follows: a) combinatorial optimization, b) heuristics and c) simulation-optimization. The relevance of these techniques in the context of mathematical scheduling problems is discussed in Chapter 2.

1.2 Research Motivation

This research primarily investigates the optimal scheduling of a set of jobs at a single workstation when the processing times are uncertain. An extension considering a two-workstation flow-shop follows.

The vast majority of scheduling literature has been analytical, often resulting in defining some general scheduling algorithm (e.g, Seo, Klein and Jang 2005, Cai and Zhou 1996). A common approach to scheduling under uncertainty is to use a point estimates for the processing time, based on the expected values. The problem is then treated as deterministic by substituting the point estimate for the uncertain distribution of processing times. Schedule optimization based on these deterministic assumptions can often be readily performed.

However, Baker and Trietsch (2007) concluded that taking the stochastic nature of the processing times into account will lead to a different optimal schedule and, in some cases, even a different sequence. In order to produce optimal schedules based on stochastic expectations Baker and Trietsch developed a unique approach in which numerous problem instances were simulated, based on assumed processing time distributions. The Evolutionary Solver®, available as a spreadsheet software add-in, was then used to determine the sequence most likely to provide the best results, given that the actual processing times were unknown at the time of scheduling.

This research extends the work of Baker and Trietsch. Like their previous work, this research primarily uses experimental techniques. As well, it similarly explores theoretical behaviour and thus uses relatively small problems. This makes different scenarios easier to explore, interpret and compare. However, the intent is to gain insights relevant to understanding issues related to large scheduling problems as well.

Baker and Trietsch (2007) investigated stochastic versus deterministic scheduling where the objective was to minimize some regular measure of performance involving tardiness, such as the proportion of tardy jobs. The probability distributions for processing times were considered to be discrete. In this research continuous distributions are used and non-regular performance measures are examined, such as those including earliness as well as tardiness penalties.

As well, this research considers inserted, or forced, idle times as decision variables. Therefore, it examines schedule optimization and not only sequence optimization. Considering earliness as well as tardiness penalties is consistent with a rapidly developing line of research in the scheduling field. In this case it may be advantageous to have the work station be idle for certain durations even if there is work in queue. Therefore, an important area of investigation concerns finding when and how much difference it makes to have inserted idle time in the schedule.

In addition to studying stochastic scheduling behaviour, additional major thrusts of this research are directed toward the comparative analysis of the deterministic and stochastic scheduling approaches and toward investigating the influence of variability on the optimal schedule.

CHAPTER TWO: LITERATURE REVIEW

The literature review in this chapter looks specifically into state-of-the-art stochastic scheduling research, primarily concerning non-regular performance measures. These are studied primarily in the context of single-machine scheduling (SMS) problems but a brief review is also provided on the two-machine flow shop problem. The major focus is on those problems which involve a common due date. Owing to the many variations of the problem, few papers address the issues of both common due date and non-regular measures. Hence this review looks at the literature on both dimensions.

In the following section a brief historical perspective on the evolution and directions of scheduling research is presented. In Section 2.2 the state-of-the-art stochastic, single-machine scheduling is dealt with. Section 2.3 examines the two-machine, flow shop research.

2.1 Brief Historical Overview

Study on scheduling began as early as the 1950's with what have become classical scheduling problems. These problems were deterministic in nature, characterized by n jobs to be processed on m different machines. Each job was assumed to have sequence-independent, fixed and known processing times on the machines. Jobs were required to visit each machine exactly once and technology constraints specified a complete, distinct routing which was also fixed and known in advance. Machines were considered to be continuously available from time zero and operations were processed without pre-emption.

Classical scheduling theory considered five major objective functions: makespan, maximum lateness, total weighted completion time, total weighted number of late jobs and total weighted tardiness. A number of efficient (exact) algorithms have been developed to provide optimal solutions. Most notable are the work by Jackson (1956), Johnson (1954) and Smith (1956).

As scheduling problems became more sophisticated, researchers were unable to develop efficient algorithms for them. The problem of optimal sequencing proved especially difficult due to the discrete nature of the problem whereby sequence variables need to be discrete quantities. Most early researchers tried developing efficient branch-and-bound methods to find an optimal solution. When complexity theory was invented, it was realized that many of these problems may be inherently difficult to solve. In the 1970's, many scheduling problems were shown to be NP-hard (Pinedo, 1995).

Appreciating the inherent difficulty in solving scheduling problems with the exact optimization algorithms, research in the 1980's started to pursue development and analysis of heuristic (approximation) algorithms. Increasing attention was also paid to the stochastic scheduling problems and non-regular due date performance measures in this era.

2.2 Single Machine Schedule – State of the Art

Single machine scheduling (SMS) models have been emphasized by many researchers as being the building blocks for a comprehensive understanding of scheduling concepts. Most of the available literature on single-machine scheduling is concerned with static, deterministic problems. Deterministic SMS models involving due date related performance measures have traditionally concentrated on regular measures like tardiness. Gupta and Kyparisis (1987) present a very good review of deterministic-static single-machine scheduling spanning about three decades.

It is only recently that an increasing number of single-machine scheduling researchers have concentrated on non-regular performance measures, which consider earliness as well as tardiness. Baker and Scudder (1990) provide a review of the literature involving earliness and tardiness penalties. The earliness-tardiness problem has been studied in different variations in the recent years. Most notable among these are studies which base analysis on common due date and linear penalties. Examples are studies by De, Ghosh and Wells (1991), Cai et al (1997), Mondal and Sen (2001), Hino, Ronconi and Mendes (2005). Studies considering common due dates and quadratic penalties include Weng and Ventura (1996) and Mondal (2002).

Sarin, Erel and Steiner (1991) considered the problem of sequencing n jobs on a single-machine with jobs having a common due date and stochastic processing times. They considered minimization of the expected tardiness (incompletion) cost, which is the sum of the weighted tardiness probabilities expressed as Equation 2.1. The job processing times are assumed to be normally distributed with known means and variances.

$$\min E[TC] = \min \sum_{i=1}^{n} k_i \Pr[C_i > d] \tag{2.1}$$

The incompletion cost, k_i, and the job variances were considered proportional to the mean processing times of the job. With these assumptions they showed that the optimal sequence is V-shaped where jobs are arranged first in non-increasing order of mean processing times and then in the non-decreasing order. They also proposed an efficient heuristic based on this optimality property.

Soroush and Fredendall (1994) studied the static single-machine scheduling problem with earliness and tardiness costs, where job processing times are random variables and due dates are distinct and deterministic. The objective was to identify an optimal sequence which minimizes the total expected weighted linear penalty (cost). They explored the case where processing times are normally distributed and performed analysis to demonstrate the effects of variations in processing times on sequencing decisions. Their model implicitly assumed that no idle time is desirable before or between the jobs and was therefore a sequence problem. They proposed three heuristics for finding an optimal sequence. Mixed linear integer programming (MIP) and the sequencing rules based on these heuristics were applied to achieve near optimal solutions. The proposed heuristic procedures were evaluated based on experiments with job sizes $n \leq 7$. The expected mean and variance parameters for the normally distributed processing times were randomly generated from a uniform distribution between 5 and 20. Also each job due date was selected randomly from the interval defined by the expected completion time ± 1, 2 or 3 standards deviations of the completion time. The expected completion time was estimated from the three heuristic candidate sequences. The unit earliness and tardiness costs were generated from a uniform distribution between 1 and 20.

Their results showed that the optimal sequences and their expected costs are significantly different from those provided by the classical deterministic, single-machine models. The expected earliness-tardiness (E/T) cost is highly influenced by the variances as well as the means of the processing times. The authors also pointed out the need to further investigate how the job characteristics, such as the mean and variance of processing times, earliness and tardiness costs, and due dates affect sequencing decisions.

Hussain and Sastry (1999) also considered the single-machine, stochastic scheduling problem with earliness and tardiness penalty costs. Following Soroush and Fredendall, they used a normal distribution of processing times with known means and variances. These distribution parameters were again generated randomly between 5 and 20 using a uniform distribution. The due dates and the unit costs of earliness and tardiness were also generated in a similar fashion. However, they used an evolutionary algorithm for solution. In particular this was a genetic algorithm (GA) coded using the ‘C’ programming language. Problems of sizes from $n = 5$ to 25 were considered and 500 samples of problems were generated for each value of n. Comparison of their results with the results of Soroush, et al. confirmed that the genetic algorithm could yield better solutions for these problems and that solution time was reasonable.

Cai and Zhou (1997) considered the minimization of expected value for a weighted combination of earliness, tardiness and flowtime penalties. The processing times were considered to be independent and normally distributed random variables with known means and known variances proportional to the means. The due dates were generated randomly using a common probability distribution. Their approach was to first analyze the optimality property and based on the analysis propose an algorithm that can be implemented to solve the problem. Two (dynamic programming) algorithms which can generate optimal or near-optimal solution in pseudo-polynomial time are proposed. Polynomial means the running time of the algorithm is polynomial relative to the numerical inputs, or problem size (Pinedo, 1995). Their results show that the optimal sequence for the problem is V-shaped with respect to the mean processing times.

Comparing their stochastic results with similar common due date deterministic E/T problems, they have demonstrated that the V-shaped algorithm and the solvability with these algorithms hold for both class of these problems, although the interpretation of the stochastic results is much more complicated than the deterministic counterpart. In addition, the proposed algorithms were extended to problems with different probability distributions. For symmetric distributions like uniform or normal their algorithms found solutions close to the optimal. However, for exponential processing times, which are nonsymmetric, they failed to obtain a good solution. In addition, they extended the work of Soroush and Fredendall by also addressing the issue of idle time insertion. They established a property that satisfies the possible range of optimal machine starting times.

Seo, Klein and Jang (2005) used a mathematical programming approach to solve a single-machine scheduling problem with an objective of minimizing the expected number of tardy jobs. The jobs had normally distributed processing times and deterministic common due dates. Unlike the study by Sarin, et al. (1991) no relationship was assumed between the mean and variance of the processing time. The original stochastic optimization problem was transformed to the equivalent non-linear deterministic problem. Four non-linear integer programming models were proposed by relaxing the problem. In other words, an approximation model was created that could be solved using LINGO®. Numerical results were obtained using the job sizes of 5, 10 and 20, mean processing times (μ_i) randomly drawn from a uniform distribution U (10, 20), and common due dates between $0.4 \sum \mu_i$ and $\sum \mu_i$. Their first model resulted in the optimal values comparable to using complete enumeration. The other three models also yielded good solutions in reasonable computation time.

Soroush (1999) considered the stochastic single-machine, earliness-tardiness problem in which due dates were also the decision variables. He presents an analytical approach to determine the optimal due dates. However, owing to the combinatorial nature of the proposed model, determining the optimal sequence was difficult. Therefore two efficient heuristics were presented to find candidates for the optimal sequence. The case with independent, normally distributed processing times was explored to demonstrate that variations in the processing times increase cost, and affect sequencing and due date decisions. He also demonstrated that if the completion time variances are ignored in the stochastic problem and the problem treated as a deterministic one, the total costs tend to increase and due dates are inaccurately assigned.

Xia, Chen and Yue (2005) studied the stochastic SMS problem with a quadratic penalty function. Processing times were assumed to be uncertain, characterized by a finite mean and a variance but no knowledge of the entire distribution. They considered minimizing the quadratic earliness-tardiness penalty, along a quadratic penalty for loose due date assignment. Earliness and tardiness were penalized at different levels, but the same level of penalty was applied to all the jobs. A heuristic solution technique was again adopted. To numerically test the performance of their heuristics a job shop with 50 jobs was considered. The parameters for the distribution means and variances were generated randomly using a uniform distribution. The actual processing times were then generated based on four probability distributions: uniform, normal, gamma and exponential. Their results indicated that the proposed heuristics were fairly robust with respect to the job processing time probability distribution. This result has practical implications since it means determining the specific shape of a distribution may not be necessary.

Another paper by Cai and Tu (1996) again deals with a quadratic penalty but also considers unreliable machines with stochastic breakdowns, along with random job processing times. Two cases of common due date are considered. The first uses a known due date while the second one treats the due date as a decision variable. Although their model considers having initial idle time in the first case, it is shown that the optimal solution for the second case does not require idle time. The machine uptime during which the machine is busy, is considered to follow a Poisson distribution. For each problem case, a general form of the deterministic equivalent problem is obtained. A sufficient condition for the optimal sequence being V-shaped with respect to the mean processing times is derived and algorithms to find optimal solution are formulated. Although their paper is the first one to consider randomness of both jobs processing and machines uptimes, stochastic machine breakdowns have also been considered by Birge, et al. (1990), Tang, et al. (2008) and others.

2.2.1 Overview on single machine scheduling approaches

Several observations can be made from the scheduling literature reviewed. One observation is that the size and type of problems that can be solved analytically using exact methods is small. The second observation is that the computer time to solve the problem grows exponentially with the number of jobs to be scheduled if analytical methods are used. As a result heuristics have often been developed but these do not guarantee an exact solution. Third, the most of the algorithms mentioned in the literature, whether exact (optimization/analytical) or approximate (heuristics), require specialized computer code for implementation. These codes are not easily available to the user.

An alternative is to use general search software that can deal with discrete optimization problems. Spreadsheet software, such as Excel®, has long been recognized as a great platform for executing computations, sometimes involving fairly sophisticated algorithms if accompanied by the use of Visual Basic for Applications® (VBA). Baker and Trietsch (2007) have recently taken spreadsheet-based methodologies a step further by showing how stochastic SMS problems can be formulated and solved using the Evolutionary Solver®, embedded in the Premium Solver® spreadsheet add-in. By considering a discrete probability distribution of the processing times they demonstrated a methodology of optimizing stochastic problem with objectives such as minimizing the expected number of tardy jobs, E[U_j], or the expected maximum tardiness E[T_{max}]. This approach uses sample based estimates of the expected values in finding the optimal solution. In other words, simulation is used to generate the expected performance as decision variables are changed during the evolutionary search for the optimum. The Evolutionary Solver is used to determine the best schedule based on the expected processing times, due dates and objectives. Optimization is based on numerical methods. While an optimal solution is not guaranteed, the methodology has been found to perform well on small problems. This methodology, described in the next chapter, marks a significant shift from the historical optimization approaches in the scheduling theory. The methodology developed by Baker and Trietsch (2007) is also the foundation for the methodology used in this research.

As far as can be ascertained, the simulation and optimization techniques described have not yet been used elsewhere in static scheduling. While simulation-optimization is becoming increasingly used in dynamic scheduling (Ramasesh, 1990), many of the problems to which it has been effectively applied do not involve discrete optimization. For example, the use of optimization tools such as OptQuest® linked into simulation tools such as Arena® still have trouble finding optimal sequences.

From review of the literature, three representative solution approaches have been identified for use in solving the scheduling problem. These are:

i) optimization (exact) algorithm
ii) heuristic (approximate) algorithm
iii) simulation-optimization (approximate)

Although there is a vast body of static scheduling literature dealing with the first two approaches, much less work has been done using the simulation-optimization approach, the main approach of interest in this research. Apart from Excel's own simulation-optimization capability with the Evolutionary Solver®, there are at least three other proprietary packages that can work well in a spreadsheet environment. Table 2.1 presents a list of these.

Table 2.1 Spreadsheet compatible simulation-optimization packages

Simulation software	*Optimizer*	*Primary search strategy*
Crystal Ball	OptQuest (built in)	Tabu search, Neural network
@RISK	Evolver (companion)	Genetic algorithm
Risk Solver	PSPS (companion)	Genetic algorithm

2.3 Two-machine Flow shop – State of the Art

There are very limited studies examining stochastic scheduling in a flow shop. Therefore several studies that have considered deterministic schedules are also identified here. Sarper (1995) considered minimization of the sum of absolute completion time deviations from a common due date for the two-machine flow shop problem. His paper is the first considering both earliness and tardiness criterion in a two-machine environment. The relative weights on earliness and tardiness for all jobs are considered identical. The problem is mathematically modeled using a mixed integer programming formulation and solved in LINDO for small sizes (5 and 6 jobs). For bigger problems three heuristic solutions are developed.

Sakuraba, Ronconi and Sourd (2009) also addressed the minimization of mean absolute deviation of completion times in a two-machine flowshop scheduling problem with a permutation sequence. A restrictive common due date was considered. This is a constant due date defined to be tight enough to influence the optimal schedule. Idle time insertion between jobs on both machines was allowed. They also modeled the problem mathematically using mixed integer programming and proposed a heuristic solution which provided better results over Sarper's heuristics.

Deterministic two-machine flow shop problems have also been studied by some other authors recently. Sung and Min (2001) considered the two machine flow shop with batch processing using the earliness-tardiness measure and common due dates. Moslehi et al.

(2009) considered the minimization of the sum of maximum earliness and tardiness using the branch-and-bound technique. At about the same time Yang (2009) studied the two-machine flow shop scheduling problem using weighted work-in-process (WIP) inventory costs. More references can be found in Lauff and Werner (2004), who present a survey of scheduling with common due dates and earliness and tardiness penalties for multi-machine problems.

Very few studies have looked at the stochastic two-machine flow shop problem. Most of these have considered flowtime related performance, most notably makespan, C_{max}, minimization. A few papers have considered total completion time $\sum C_i$ minimization in a stochastic two-machine flowshop environment. Most notable among these is Soroush and Allahverdi (2005). They studied a two-machine flow shop where the processing times have discrete distributions and determined the schedule that minimized the expected total completion times, $\sum C_i$. Independent, normally distributed processing times were considered heuristic solution methods were proposed and tested.

There do not appear to be flow shop studies that have used due date related measures in static scheduling problems with stochastic processing times. In fact, no literature dealing with static, stochastic two-machine (or more) flow shop problems with non-regular due date performance measures could be identified.

CHAPTER THREE: RESEARCH METHODOLOGY

In Chapter 1 the motivations for this research, along with the extensions to be pursued, were discussed. In Chapter 2, the methodology to be used was briefly mentioned. In this chapter, the research methodology is presented in detail. In Section 3.1 the conceptual information for mathematical modeling in spreadsheets is presented. Section 3.2 explains the spreadsheet formulation and mathematical modeling. The performance measures of relevance and a key parameter influencing these measures, the earliness-tardiness weighting ratio, is discussed in Section 3.3.

3.1 Mathematical Modeling using Spreadsheets

In the quantitative approach to decision making, mathematical models are used to represent the problem of interest. Computer spreadsheets such as Excel® are ideal for building such mathematical models, since the relationships of interest are transparent. As well, the computations are quick and the output is easy to evaluate.

A model that is built in the spreadsheet is essentially a mathematical formulation of the relationships of interest. It takes as inputs the necessary controllable (decision maker has control over or can influence) or uncontrollable (decision maker has no control over) problem variables and gives as an output the desired performance measure. Uncontrollable problem inputs can either be known with certainty (deterministic) or subject to random variation (stochastic). The conceptual diagram for this mathematical modeling relationship is illustrated in Figure 3.1.

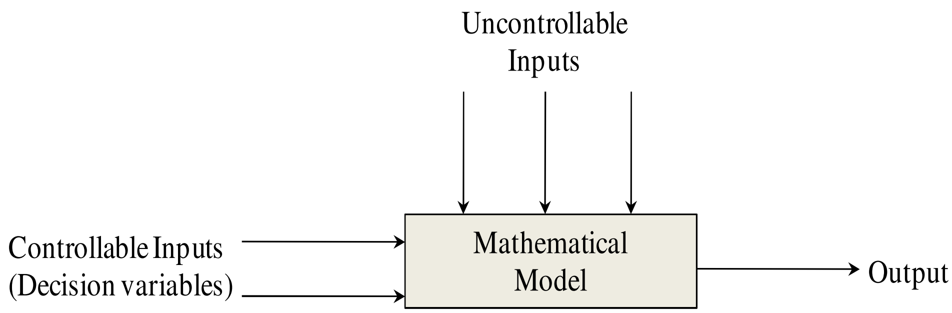


Figure 3.1 Conceptual diagram of a mathematical model

When at least one of the uncontrollable inputs in the model is probabilistic, it is possible to randomly generate this input in the spreadsheet. The spreadsheet model then provides a method to predict the output for some chosen values of the controllable inputs and the randomly generated inputs. Additionally, by embedding an appropriate optimization tool in this simulation environment, the modeller is facilitated in the search for optimal decision variables which minimize (or maximize) the chosen objective function which is the output performance measure.

The idea of maximizing or minimizing a function subject to some constraints finds its roots in mathematical programming. If the objective function and the constraints in the problem can be expressed as linear equalities or inequalities, an optimal solution can be found by using the simplex algorithm. This approach is known as linear programming (LP). However, if some or all of the decision variables must be integers, the problem is said to be an Integer Programming (IP) problem. Integer variables are required where sequence optimization is required. IP problems have been proved to be much more difficult to solve than the LP models, since the simplex method alone cannot be used. IP problems can be formulated in spreadsheets readily by specifying integer decision variables in the Solver.

Many optimization problems have an objective or constraints which are non-linear functions of the decision variables. The Standard GRG (Generalized Reduced Gradient) non-linear Solver® add-in can be used to solve these problems. If the objective function or the constraint in the spreadsheet model, directly or indirectly, points to decision cells using spreadsheet functions like @MAX, @MIN, @ABS or @IF, the model is said to have a “non-smooth” characteristics. This makes the problem difficult to solve using the standard non-linear solver.

Evolutionary Solver®, which comes as a component of the Premium Solver® add-in, is useful in solving problems that are difficult to handle using traditional non-linear solvers. The Evolutionary Solver® uses a genetic algorithm to intelligently search the entire feasible region for an optimal solution and is less likely to get trapped in local minima/maxima. It is particularly suited to solving discrete combinatorial optimization problems, such as the sequencing/scheduling problem, which requires one to choose the best among many different possible combinations. The “all different” constraint available in the Evolutionary Solver® facilitates shuffling through the valid permutation sequences during the evolutionary search for the optimal schedule.

3.2 Modeling the Research Problem

The scheduling problem in this research is defined to be that of finding the optimal sequence in which to process a set of jobs and the optimal idle time to insert prior to the starting of each job. In other words, the sequence and the inserted idle times are the model decision variables.

The spreadsheet is made flexible enough to consider various performance measures as the model objective function. These include the proportion of tardy jobs, mean tardiness, the mean weighted completion time deviations from the due date and the mean weighted squared completion time deviations from the due date. However, the focus of this research is on the last two of these measures. Schedules are created using these two measures in the objective functions.

In the stochastic approach, the job processing times are subject to random variations based on a known probability distribution and its first two moments, the mean and the variance. In constructing the optimal schedule, the actual distribution of processing times is considered by using randomly generated times. In the deterministic approach the processing times are assumed to be known exactly. Any uncertainty in the processing times is suppressed and the scheduling is done on the basis of the expected values of the processing times as the actual processing times.

Figure 3.2 gives a snapshot of the spreadsheet model for a single-machine, two-job problem. This model is illustrative of the size of the problem addressed in this research. However, equivalent spreadsheets for up to 8 jobs were also experimented. Although the same methodology as the one illustrated here works with these bigger problems, the Solver does not always find an optimal solution, plus evaluating whether the optimal is found is more difficult. As well the behaviour is harder to understand and obtain insights from. Therefore the study is restricted to two job problems. The two-machine model is conceptually similar and a snapshot is provided in Appendix B.

All model inputs related to the set of jobs to be scheduled, such as processing time distributions and due dates, are defined in this Excel® worksheet (cells B9:D22). As well, the @HLOOKUP function in Excel facilitates setting up references between the decision variables (cells H21:I22) and job characteristics during re-sequencing. Performance characteristics can be determined for different schedules by using cell formulas to determine the number of tardy jobs, average earliness and average tardiness (columns N through X). The performance measure of interest can then be referenced by the objective function input area in the Evolutionary Solver®.

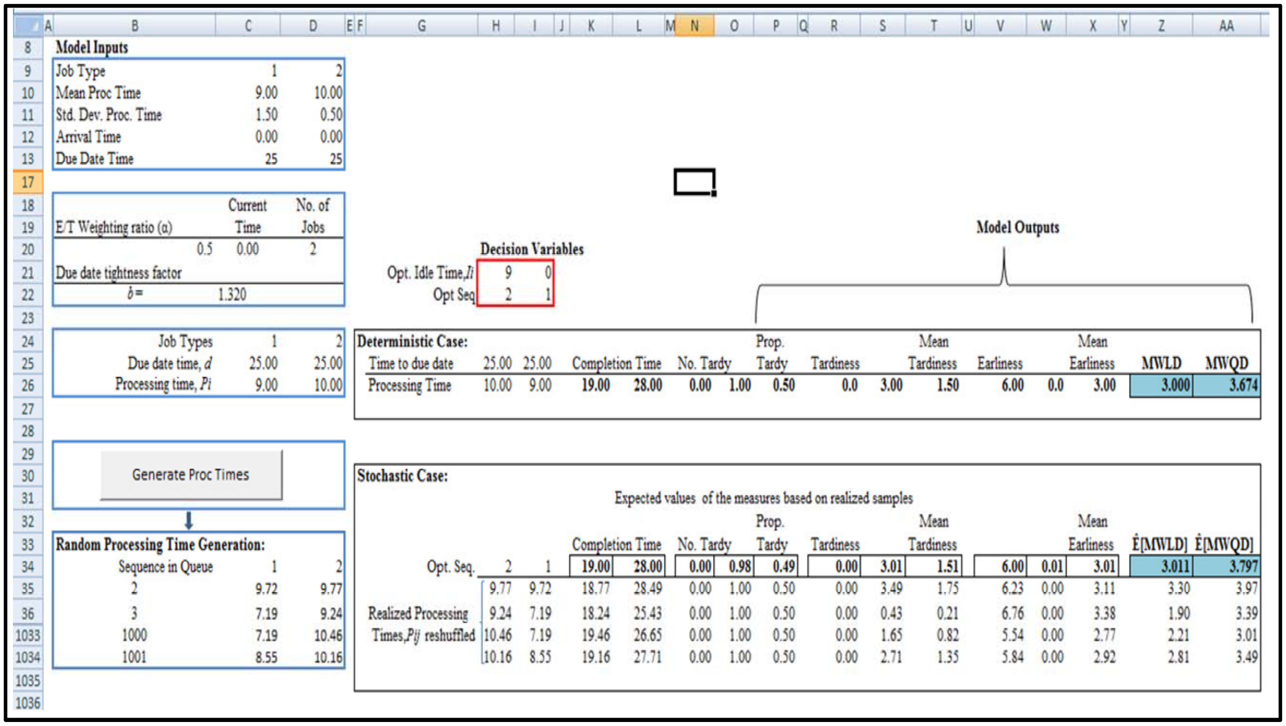


Figure 3.2 Snapshot of a single-machine spreadsheet model

The Evolutionary Solver® is invoked by clicking the Add-Ins tab. Activation of the Premium Solver® opens a Solver Parameters window, as shown in Figure 3.3.

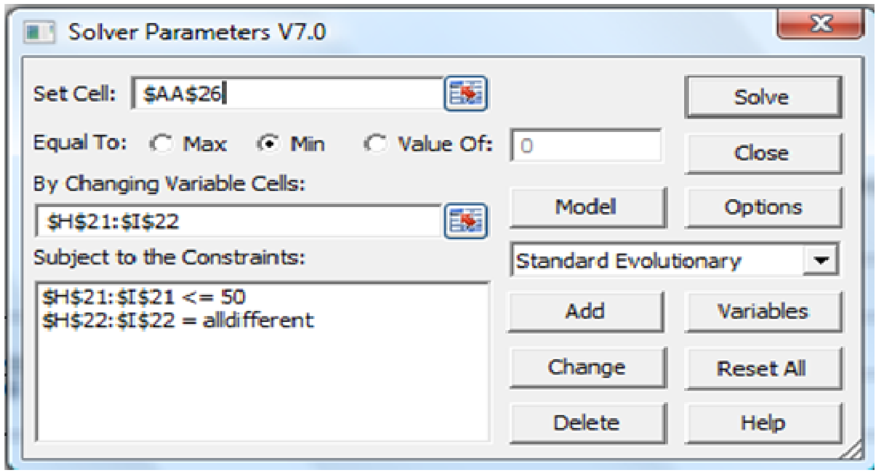


Figure 3.3 Snapshot of Evolutionary Solver dialog box

The objective function is referenced as cell AA26 (MWQD) in this example. The variable cells (H21:I22) are for the decision variables, which are the idle times before the start of each jobs (H21:I21) and the sequences of the jobs (H22:I22). The constraints on the idle time (H21:I21<=50) ensures that the problem is not restricted from having a sufficient amount of idle time before jobs while the search is on during optimization. The "all different" constraint (H22:I22=alldifferent) ensures that the sequence is feasible permutation sequence of the jobs.

In the deterministic approach, the Evolutionary Solver® is used to find the optimal sequence and inserted idle time for a set of jobs assumed to have deterministic processing times. These times are equal to the means (cells C10:D10) of the true distribution of processing times for each job.

In the stochastic approach, the solution procedure used is similar to that identified by Baker and Trietsch (2007). An initial sequence of jobs is assumed (C34:D34) and then 1000 problem instances were generated (C35:D1034) by drawing processing times from the assumed job processing time distributions. In this spreadsheet the Normal distribution is assumed when the processing times are generated. This simulation is facilitated by using the Random Number Generation tool in the Data Analysis add-in for Excel. As well, a Visual Basic for Applications® (VBA) macro is implemented to facilitate generating and writing these values out to the spreadsheet. This macro is activated by a button in the spreadsheet. For example, a two-job problem resulted in two columns and 1000 rows of processing times being written to the spreadsheet. Another two columns with 1000 rows (cells H35:I1034) are created with @HLOOKUP functions (see Appendix C for formula) to reference the processing times for each problem instance with any alternative sequence specified. In other words, these columns allow the jobs to be re-shuffled according to the different sequences being evaluated by the Evolutionary Solver®. Columns K through X are used to evaluate earliness and tardiness measures through the use of cell formulas, taking the given due dates into account. Finally, averages of the results for the 1000 problem instances are computed within a single cell (cell Z34 for linear penalty or AA34 for quadratic penalty). This cell is then referenced by the objective function in the Evolutionary Solver®. A flowchart representation of a stochastic schedule optimization problem described here is provided in Figure 3.4.

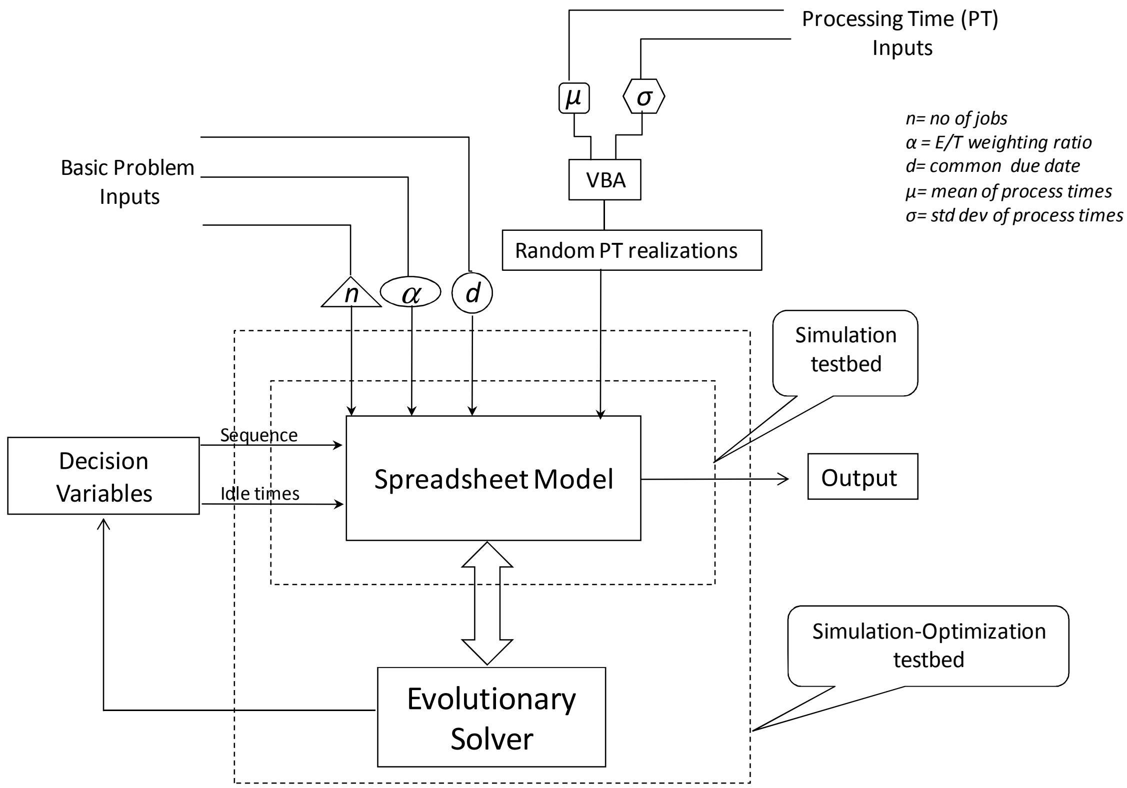


Figure 3.4 Model flowchart

When deterministic job processing times are used the Evolutionary Solver searches for the best solution in a far less time than if the optimization is done using a stochastically generated processing times. This is because the stochastic problem requires generating more number of sub-problems or the trial solutions than its deterministic counterpart before an optimal is found. For example, in the example problem of Figure 3.2, Evolutionary Solver typically generated 756 sub-problems using deterministic problem approach. A stochastic approach to the same problem, required about 4000 sub-problems to be generated before the optimal was found. Several user defined parameters available by clicking the Options tab in the solver dialog box can be defined as criteria for the stopping rules; however these should be large enough not to impede the search for a good (or an optimal) solution.

The spreadsheet optimization of the decision variables or the minimization of the objective function is a two step process. In the first step an initial solution is chosen, whereby both the sequencing decision variables (H22:I22) and the idle time decision variables (H21:I21) are assigned arbitrarily. The initial idle times are generally set at zero and the initial sequence can be any feasible permutation. Choosing these initial decision variables the performance measures are output in the respective cells, which are not optimal. For example, the initial solution with 1-2 sequence and no idle times before jobs in the example problem of Figure 3.2 yields a mean weighted quadratic deviation, *MWQD*= 8.544. A Gantt chart illustration of this problem is given in Figure 3.5

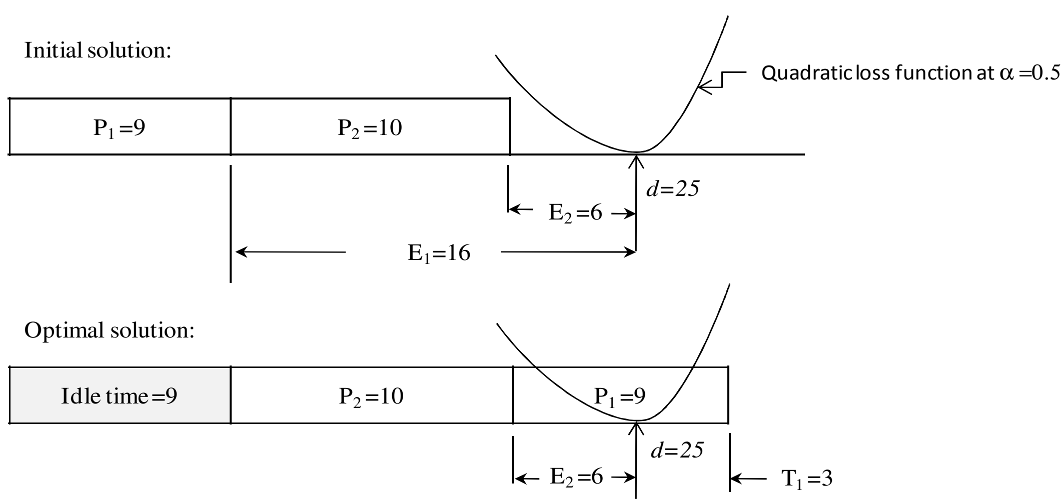


Figure 3.5 Gantt chart illustration of initial and final solution

In the initial solution, job 1 completes 16 units ahead and job 2 completes 6 units ahead of the due date. These are not the optimal completion times with respect to the given penalty function, the due date and the earliness tardiness weighting ratio. When the Evolutionary Solver optimizes for this problem, it tries to find the best idle times to be placed before the jobs and the best sequence to order the jobs. This is achieved by having a sequence 2-1 and an idle time of 9 units before the first job in the sequence as shown in the Figure 3.4. This will make job 2, which is ordered first in the sequence, to complete 6 units ahead and job 1, which is ordered second in the sequence, to complete 3 units late than the due date. This will result in a *MWQD*=3.674 for this problem which turns out to be the optimal (minimum) value in this problem.

Once the Evolutionary Solver arrives at a solution, further experimental verification on the results is generally performed to ascertain that the results obtained are optimal. In one verification, the idle time decision variable is changed incrementally to make sure the objective function value deteriorates. For example, referring to the problem in Figure 3.2, when idle time before the first job in the optimal sequence is varied from the obtained value of 9, the result shown in Figure 3.6 is obtained.

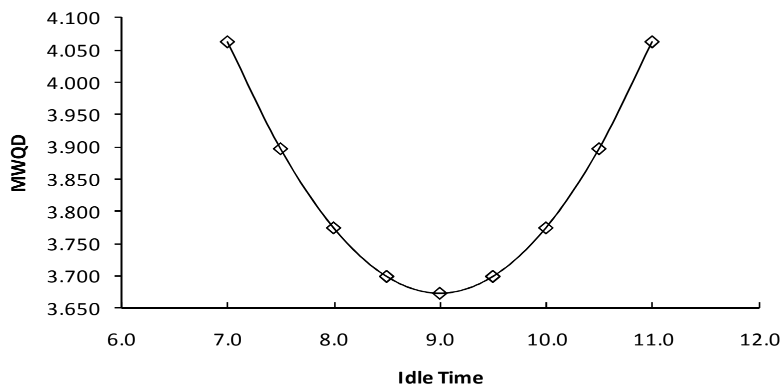


Figure 3.6 Verification of optimal idle time

In the other verification, a different starting sequence is forced by including only the idle times as a decision variable, the problem is re-optimized and checked whether the objective function has deteriorated.

Verification using the analytical solution in Mathematica® is also extensively used in this research. Comparing the simulation (experimental) results from the Evolutionary Solver with the analytical results is facilitated by running multiple optimizations to obtain confidence intervals on the means. This requires several independent replications of an experiment which is achieved by generating the 1000 instances of random processing times and running the Evolutionary Solver each time to find an optimal solution. When comparing the different results of simulation experiment, however, common random numbers are used.

For the performance measures which are linear functions of the decision variables, for example the *MWLD* measure, a deterministic class of the problem can also be modeled using the linear programming formulation in the spreadsheet. The linear programming formulation for the two-job problem like the one shown in Figure 3.2 is illustrated here.

<u>LP formulation of a two job deterministic problem with linear measure</u>

Let x_{jk} represent the job-position variable, and let t_k and e_k be the tardiness and earliness for jobs in positions k. The problem will consist of four job-position variables, three sets of constraints and one objective function. The problem can be written as:

Minimize $\sum_{k=1}^{2} t_k + e_k$,

subject to the constraints:

i) All jobs must be scheduled in only one position

$$x_{11} + x_{12} = 1$$
$$x_{21} + x_{22} = 1$$

ii) All positions must have only one job

$$x_{11} + x_{21} = 1$$
$$x_{12} + x_{22} = 1$$

iii) Earliness - Tardiness constraint

For the job in first position:

$$I_1 + [p_1(x_{11}) + p_2(x_{21})] - [d(x_{11}) + d(x_{21})] + e_1 - t_1 = 0$$

$$e_1 \geq 0, t_1 \geq 0$$

For the job in second position:

$$I_1 + I_2 + [p_1(x_{11} + x_{12}) + p_2(x_{21} + x_{22})] - [d(x_{12}) + d(x_{22})] + e_2 - t_2 = 0$$

$$e_2 \geq 0,\ t_2 \geq 0$$

When the processing times p_1 and p_2 are deterministic, this linear problem can be solved using the standard linear programming solver (LP solver) in Excel®. When the problem involves objective functions that are non-linear, such as quadratic deviations in the *MWQD* measure, the traditional non-linear programming solver (NLP solver) may still be used. However, constraining the problem to get a permutation sequence is difficult as the problem size grows larger. In addition, discrete optimization using non-linear solvers presents problems, as discussed in Section 3.1. These factors both make it unsuitable to deal with sequencing problems, in particular those involving stochastic times. The Evolutionary Solver®, on the other hand, is good at solving sequencing problems involving both deterministic or stochastic times, and linear or non-linear measures. The strength of this solver results from its genetic search algorithm and the discrete "all different" constraint, which facilitates sequencing.

3.3 Performance Measures

In this research two particular non-regular performance measures are of interest. These are the weighted earliness-tardiness measures given by Equations (3.1) and (3.2).

$$MWLD\ (n,\alpha,d) \equiv \sum_{i=1}^{n}\left[\max(0,C_i-d)+\alpha\{\max(0,d-C_i)\}\right]\Big/n \tag{3.1}$$

$$MWQD(n,\alpha,d) \equiv \sqrt{\left(\sum_{i=1}^{n}\left[\{\max(0,C_i-d)\}^2+\alpha\{\max(0,d-C_i)\}^2\right]\Big/n\right)} \tag{3.2}$$

where:

i	- job index
n	- number of jobs in the job set
C_i	- completion time for job i
d	- common due date (time) for jobs
α	- multiplier for weighting earliness relative to tardiness

Equation (3.1) gives a mean weighted linear deviation measure (*MWLD*) while Equation (3.2) gives a mean weighted quadratic deviation (*MWQD*) measure. Tardiness, defined as positive lateness, and earliness, defined as negative lateness, are respectively represented as $T_i = \max(0, C_i - d)$ and $E_i = \max(0, d - C_i)$ in the above equations.

If the problem is deterministic, the processing time of a job is constant (not a random variable). Then the completion time of the i^{th} job in the sequence, $C_{(i)}$ is also a constant, given by Equation 3.3. Note here that the parentheses are used to differentiate the sequence position from the job index.

$$C_{(i)} = \sum_{w=1}^{i} I_{(w)} + \sum_{w=1}^{i} P_{(w)} \tag{3.3}$$

where,

$I_{(w)}$ - idle time prior to the w^{th} job in sequence

$P_{(w)}$ - processing time for the w^{th} job in sequence

The deterministic approach is therefore based on minimizing the objective functions represented by Equations (3.1) and (3.2) by choosing the best sequence and the idle times. It is also customary to refer to these objective functions as the total weighted linear/quadratic penalty function or the total linear/quadratic loss function.

In the stochastic problem, the completion times appear as random variables because the processing times, P_w, are now random variables. If we consider C_{ij} as the j^{th} realization of C_i for a given sequence of jobs and if m problem instances are simulated, the expected values of the mean weighted deviation measures can be estimated as an average over the m problem instances. This is expressed algebraically in Equations (3.4) and (3.5) for linear and quadratic deviations respectively.

$$\hat{E}\left[MWLD\,(n,\alpha,d)\right] \equiv \frac{\sum_{j=1}^{m}\left[\left(\sum_{i=1}^{n}\left[\max(0,C_{ij}-d)+\alpha\left\{\max(0,d-C_{ij})\right\}\right]\right)\Big/n\right]}{m} \tag{3.4}$$

$$\hat{E}\left[MWQD\,(n,\alpha,d)\right] \equiv \frac{\sum_{j=1}^{m}\left[\sqrt{\left(\sum_{i=1}^{n}\left[\left\{\max(0,C_{ij}-d)\right\}^2+\alpha\left\{\max(0,d-C_{ij})\right\}^2\right]\Big/n\right)}\right]}{m} \tag{3.5}$$

The stochastic problem approach in this research is therefore based on the minimization of the functions represented by Equations (3.4) and (3.5). The decision variables are the sequence of jobs and the inserted idle times between jobs. In this research the number of problem instances, m, simulated any set of decision variables was 1000.

3.3.1 Earliness-Tardiness weighting ratio

The earliness-tardiness weighting ratio, α, is used in these equations to penalize earliness relative to tardiness. It is assumed that the unit penalties do not vary for job types. That is to say, α is the same across all jobs. An alternative approach would be to place different weights on the jobs by assigning different α values to each of them.

An α value of 1 indicates that the unit penalty on a job being early is equal to the unit penalty on it being tardy. If $\alpha < 1$ the tardiness penalty is greater than the earliness penalty, and vice versa for $\alpha > 1$. In practice $\alpha \neq 1$ is more common than $\alpha = 1$ since the unit cost of being tardy is an exogenous factor affecting customer service, whereas the unit cost of being early is an endogenous factor affecting holding costs. These could be significantly different (Baker and Scudder, 1990). It is important to consider each of these cases if the scheduling behaviour were to be explored completely. In the spreadsheet this can be done by changing the value for α (cell B20) and re-running the optimizer. This can be automated for experimenting with several values of α by using a VBA macro.

Figure 3.7 and 3.8 show how the penalty function varies for different values of α with linear and quadratic measures. For α =1, the penalty functions are symmetric about d . i.e., jobs with equal absolute deviations from a common due date receive the same penalty. But for $\alpha \neq 1$, the penalty functions are asymmetric about d, so that the jobs with equal absolute deviations are penalized differently.

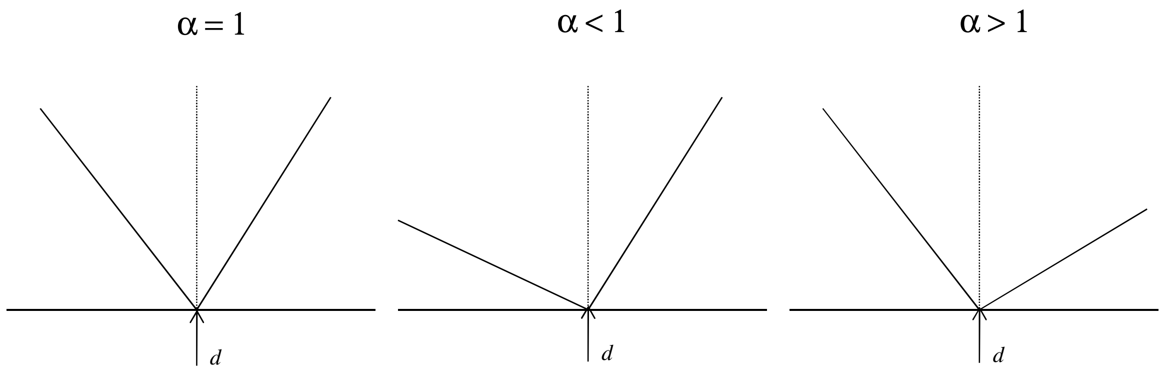


Figure 3.7 Linear penalty function with different α

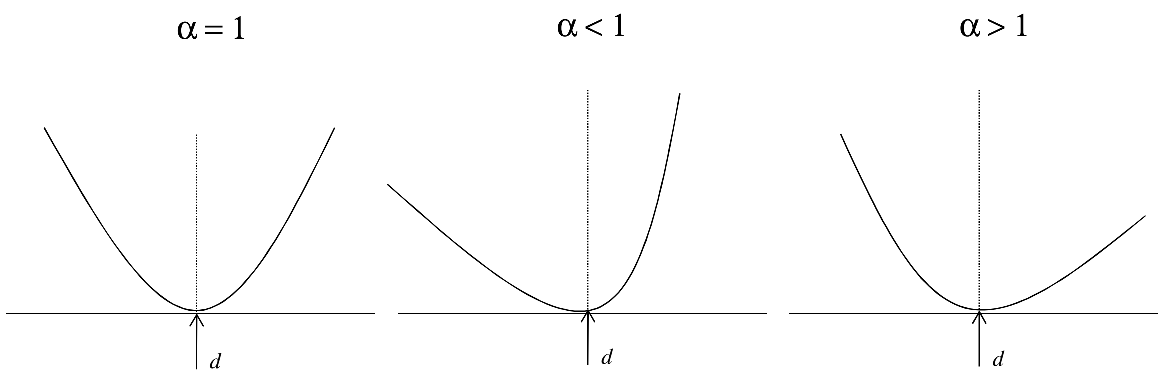


Figure 3.8 Quadratic penalty function with different *α*

In a deterministic problem, where there is no random variability in the processing times, or for that reason the completion times C_i, the total weighted penalties can be predicted from the penalty functions as shown in Figure 3.7 and 3.8 .

But in a stochastic problem, because the completion times C_i are the random variables, the knowledge of the probability density function of the completion time distribution also becomes necessary to predict the expected total penalty of each job. It is more appropriate to overlay the density function with the loss (penalty) function as the one shown in Figure 3.9 for a quadratic penalty with $\alpha < 1$ and normal distribution of completion time.

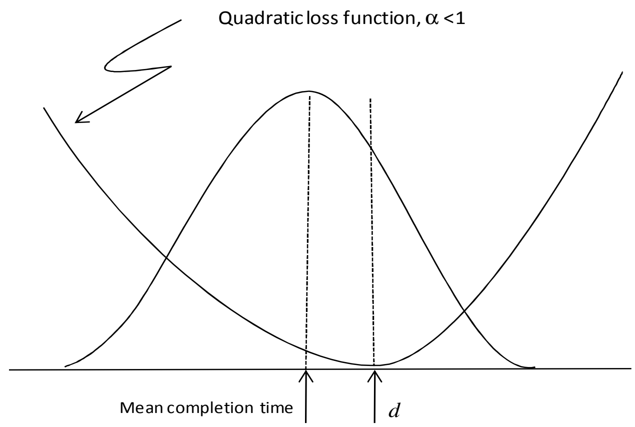


Figure 3.9 Quadratic penalty function with normal distribution of processing time overlayed

The expected weighted quadratic penalty for a job can then be expressed as Equation 3.6.

$$E[WQD] = \int_{c_i=0}^{\infty} f(c_i).\ g(c_i)\ dc_i \tag{3.6}$$

where,

$f(C_i) = \left(\{\max(0, C_i - d)\}^2 + \alpha\{\max(0, d - C_i)\}^2 \right)$

$g(C_i)$ - probability density function of the completion time distribution

To analytically calculate the expected mean weighted quadratic deviation over all the jobs, the expression similar to Equation 3.6 needs to be calculated for all the jobs in the sequence and then averaged over the number of jobs. Analytical computations are however difficult with unbounded distributions like the normal distribution even for a two job problem. Simpler distributions such as the uniform distribution are therefore used to solve the stochastic problems analytically. Such an analytical model for a two job problem case is presented in chapter 4.

CHAPTER FOUR: SINGLE MACHINE PROBLEM ANALYSIS

The importance of the single machine scheduling (*SMS*) problem to theoretical scheduling research was mentioned in Chapter 2. This chapter presents a detailed analysis of a static *SMS* problem which is characterized by n independent jobs available to be processed on a single resource without pre-emption. The minimization of linear and quadratic deviations from a common due date are considered as the performance objectives. The deterministic class of problems are denoted as [$n/m(=1)/MWLD/d$] and [$n/(m=1)/MWQD/d$] for linear and quadratic measures respectively. In the stochastic problem deviations are minimized in an expectation sense. The problems are equivalently identified as [$n/m(=1)/\hat{E}[MWLD]/d$] and [$n/m(=1)/\hat{E}[MWQD]/d$].

Simple models of the *SMS* problems are created using only one and two jobs (n = 1, 2). Considering several example problems, insights are developed which help evaluate the deterministic and stochastic scheduling approaches that were described in Chapter 3. A preliminary study of idle time scheduling using a single job problem is first presented in Section 4.1. Then optimal scheduling with two jobs and a common due date is reviewed in detail in Section 4.2 using the experimental or simulation-optimization technique discussed in Chapter 3. The analytical approach to solving this type of problem is presented in Section 4.3. A summary of important insights and key observations is finally provided in Section 4.4.

4.1 Single-Machine, Single-Job Problem

The single-machine, single-job problem is intrinsically not a sequencing problem. However basic insight into idle time behaviour can be acquired using this problem. In this study, we consider that the job processing time P_1 is a random variable, drawn from a Normal distribution, with a mean $\mu = 14$ and a standard deviation of $\sigma = 1.0$. The due date time is assumed arbitrarily set at $d = 40$. It is assumed that no additional setup times are incurred.

4.1.1 Optimal idle time with linear penalties

A deterministic problem is first considered by suppressing the randomness in the job processing time ($\sigma = 0$) and taking the assumed mean of 14 as the actual processing time P_1. The due date deviation (either *MWLD* or *MWQD*) is zero if the job completes exactly at its due date. If the due date is sufficiently loose, ($d > P_1$) it is therefore desirable to delay the starting time of the job so that it completes exactly at the due date.

Referring to Figure 4.1, the optimal amount of inserted idle time I_1 before the job for a single-job problem is therefore equal to the difference between the due date and the processing time (d - P_1), which for this problem is 26. In other words, optimal idle time may be intuitively deduced by setting $d - (P_1 + I_1) = 0$ for a deterministic problem. The optimal idle time for a single-job deterministic problem is then also independent of earliness tardiness weighting ratio, α.

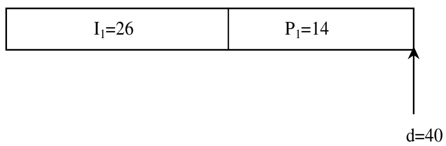


Figure 4.1 Optimal idle time under deterministic processing time, all *α*

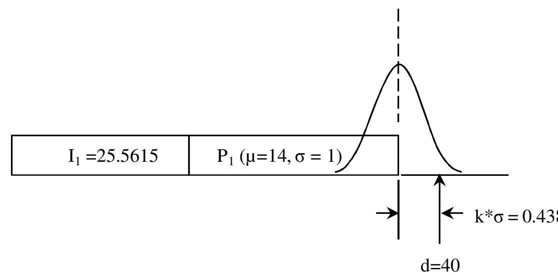


Figure 4.2 Optimal idle time under stochastic processing time, *α* = 0.5

When the processing time P_1 is uncertain (random variable), the expected value of the weighted linear deviation can be analytically obtained as the weighted sum of the expected earliness and expected tardiness given by Equation 4.1.

$$E[MWLD] = \int_0^d \alpha\left(d - p_1 - I_1\right) f_{P_1}(p_1)\,dp_1 + \int_d^\infty \left(p_1 + I_1 - d\right) f_{P_1}(p_1)\,dp_1 \qquad (4.1)$$

where:

$f_{P_1}(p_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p_1-\mu)^2}{2\sigma^2}\right)$ is the probability density function of a normal processing time (P_1) distribution with mean μ and standard deviation σ.

The first integral in Equation 4.1 gives the expected weighted earliness and the second integral gives the expected weighted tardiness.

Equation 4.1 can be numerically solved for the minimum value with respect to the idle time I_1 using the numerical minimization function in Mathematica®. Table 4.1 summarizes the results with different values of α.

The optimal idle time I_1 in the stochastic problem is observed to be dependent on the chosen values of α. When the penalties for earliness or tardiness are the same ($\alpha = 1$), then the linear penalty is minimized when the expected values for earliness and tardiness balance. This occurs at an idle time of 26, as in the deterministic case. If $\alpha < 1$, tardiness needs to be avoided more, resulting in an initial idle time which is less than if the penalties were the same for both. Similarly, if $\alpha > 1$, the optimal idle time is greater than with having equal penalties.

Table 4.1 Single-machine, single-job results at several values of α

α	Numerical Minimization Result		Evolutionary Solver Result	
	I_1	$E[MWLD]$	I_1	$\hat{E}[MWLD]$
0.5	25.5693	0.5454	25.5615 ± 0.018	0.5467 ± 0.005
1	26.0000	0.7979	26.0035 ± 0.021	0.7994 ± 0.007
1.5	26.2533	0.9659	26.2607 ± 0.040	0.9679 ± 0.007

The Evolutionary Solver (EV solver) was also run for this problem. The EV solver results are tabulated (Table 4.1) as the average values with a 95% confidence interval (CI) half-width based on ten independent replications of an experiment using different random numbers. However, common random numbers were used for each α. In other words, each of the experimental runs was statistically identical and independent and uncorrelated. The 95 % confidence interval half-widths in Table 4.1 are pretty low compared to the means (<1% in all the cases). Therefore the precision of the results based on using the ten independent replications can be relied on. The analytical (Mathematica®) and experimental results (Evolutionary Solver®) correspond closely, verifying the optimality of the idle times obtained.

If the penalty functions for earliness and tardiness are linear (i.e, when $\hat{E}[MWLD]$ is the measure) it is also possible to use the partial expectations, or loss functions, to determine the total earliness and tardiness penalties analytically. This approach is similar to that used to determine the expected number of stock-outs per order cycle (ESPRC) in inventory problems (Silver et al., p. 279, or Vollmann, et al., p. 149).

Assuming an inserted idle time of 25.5615, obtained experimentally with α of 0.5, Figure 4.2 shows the problem in more detail. The minimum linear penalty or the experimental loss of 0.5467 obtained for this problem (Table 4.1) corresponds to a deviation of $k * \sigma$ (= 40-39.5615 = 0.438) from the mean job completion time ($\hat{x}$ =39.5615) using the ESPRC approach. This results in a value k (experimental) = 0.438 for a standard deviation of σ = 1.

Now theoretically, the expected loss, $L(x)$ (or $G_u(k)$ according to Silver et. al), due to the tardiness of the job, using Unit Normal Distribution tables, will be 0.2169. The expected loss, $L(-x)$ (or $G_u(-k)$ according to Silver et. al), due to earliness of the job will be 0.6569.

The total weighted expected theoretical loss is therefore, $L(x) + \alpha * L(-x) = 0.2169 + 0.5 * 0.6569 = 0.54535$. This is close to the experimental value 0.5467, which further proves the consistency of the experimental result with the theoretical (analytical) result.

In order to verify whether this theoretical loss is the minimum loss, the Unit Normal Distribution tables (Silver et al., p 724) is again utilized to calculate the weighted total expected theoretical loss at several values of *k*. The results are provided in Table 4.2, which shows that the expected theoretical loss is minimum for $k = 0.44$.

Table 4.2 Expected loss using Unit Normal Distribution Table, α = 0.5

k	$G_u(k)$	$G_u(-k)$	Expected Theoretical Loss $[G_u(k) + \alpha \times G_u(-k)]$	Average Experimental Loss
0.40	0.2304	0.6304	0.545600	0.547241
0.42	0.2236	0.6436	0.545400	0.546965
0.44	**0.2169**	**0.6569**	**0.545350**	**0.546917**
0.46	0.2104	0.6704	0.545600	0.547084
0.48	0.2040	0.6840	0.546000	0.547431

To further verify the optimality, the idle time in the spreadsheet were incrementally changed while observing the loss. It was found that any other value of the idle times that result in *k* values higher or lower than 0.44 increased the average experimental loss as observed in the results in the last column of Table 4.2. This suggests that the inserted idle time, I_1 (= 25.5165) found using the Evolutionary Solver® is at or very near optimal for this problem.

4.2 Single-machine, Two-job problem

A common due date (d) based on the average total work content (TWK) is initially used to investigate the single-machine, two-job problem. For a deterministic problem the job processing times, P_i's, are the expected values or the means (μ_i) of the assumed processing time distribution. In the stochastic problem, multiple instances of P_i are randomly drawn from a known distribution. The common due date is thus defined to be some multiple (b) of the cumulative processing time of the jobs given by Equation 4.2.

$$d = b \sum_{i=1}^{n} \mu_i \tag{4.2}$$

where μ_i is the deterministic processing time of the job i. Further all jobs are assumed to be available at the start of the scheduling period and no set up times are incurred.

Three different levels of the common due dates are considered. When the due date is sufficiently large that all of the jobs can fit in (complete) before the due date, it is said to be "loose". In this case the due date is greater than the cumulative sum of the (deterministic) processing times, $d > \sum_{i=1}^{n} \mu_i$. Due dates less than half of the cumulative processing time, $d < \frac{1}{2} \sum_{i=1}^{n} \mu_i$, of the jobs are considered to be "tight". Due dates anywhere in between are said to be of "medium" tightness, which satisfy $\frac{1}{2} \sum_{i=1}^{n} \mu_i \leq d \leq \sum_{i=1}^{n} \mu_i$. The idea of defining the due date tightness in this manner is in some sense similar to having tightly restricted, restricted and unrestricted versions of the common due date. Restricted problems are those where the due date influences the optimal sequence. The values of *b* are varied accordingly to construct the different problem scenarios. The problem inputs are shown in Table 4.3.

Table 4.3 Input table for single-machine, two-job problem

Job Type	A	B
Mean Processing Time, μ	14.0	18.0
Std. Dev. of Proc. Time, σ	1.0	1.0
Arrival Time	0.0	0.0

4.2.1 Schedule based on expected times (deterministic)

While considering earliness as well as tardiness penalties, it is of interest to understand the benefits of having the inserted or forced idle time in the schedule. In this section, both the IIT (inserted idle time) schedule and the Non-IIT schedule are considered. The IIT schedule is one in which the idle times prior to starting any job are considered as a decision variable. In the Non-IIT schedule forced idle times are not allowed. In other words, in a Non-IIT schedule, the idle times are not the decision variables (only the sequences are) and are set at zero.

Experiments are run at varying degrees of due date tightness. The results for the deterministic problem using α =0.5 are shown in Table 4.4. The linear and the quadratic deviation measures given in this Table were obtained using the Equations 3.1 and 3.2 respectively, which are defined in chapter 3.

Table 4.4 Deterministic solution with and without inserted idle times when α = 0.5

Linear Measure, deterministic approach, α =0.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	MWLD	Opt. Seq	I_1	I_2	MWLD	
Tight	8	A-B	15.0	A-B	0	0	15.0	-
	14	A-B	9.0	A-B	0	0	9.0	-
Medium	16	A-B	8.5	A-B	0	0	8.5	-
	18	B-A	7.0	B-A	0	0	7.0	-
	20	B-A	6.5	B-A	0	0	6.5	-
	32	B-A	3.5	B-A	0	0	3.5	-
Loose	40	B-A	7.5	B-A	8	0	3.5	53.33%

Quadratic Measure, deterministic approach, α =0.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	MWQD	Opt. Seq	I_1	I_2	MWQD	
Tight	8	A-B	17.49	A-B	0	0	17.49	-
	14	A-B	12.73	A-B	0	0	12.73	-
Medium	16	A-B	11.36	A-B	0	0	11.36	-
	18	B-A	9.89	B-A	0	0	9.89	-
	20	B-A	8.54	B-A	0	0	8.54	-
	32	B-A	7.00	B-A	4.67	0	5.72	18.30%
Loose	40	B-A	11.70	B-A	12.67	0	5.72	51.11%

These results here indicate that the benefit in having idle times accrues only when the due date is sufficiently loose. It should be noted that the role of idle time is to delay the job finish time. If the due date tightness is such that no jobs can be early then IIT scheduling is meaningless. Idle time schedules can be meaningful with less tighter due dates, which allow some jobs to complete early. A significant improvement in the penalty is guaranteed with a loose due date whereby all jobs can complete before the due date. Idle time helps to push the early jobs closer to the due date.

As observed from Table 4.4, idle time was not required until a due date of $d = 40$ when the linear penalty was considered. Improvement of about 53 % was obtained in the linear penalty by having an idle time I_1=8 before the first job. Similar experiments with a quadratic penalty measure resulted in the improvement starting with due date $d = 32$. Improvements increased as due dates were loosened further.

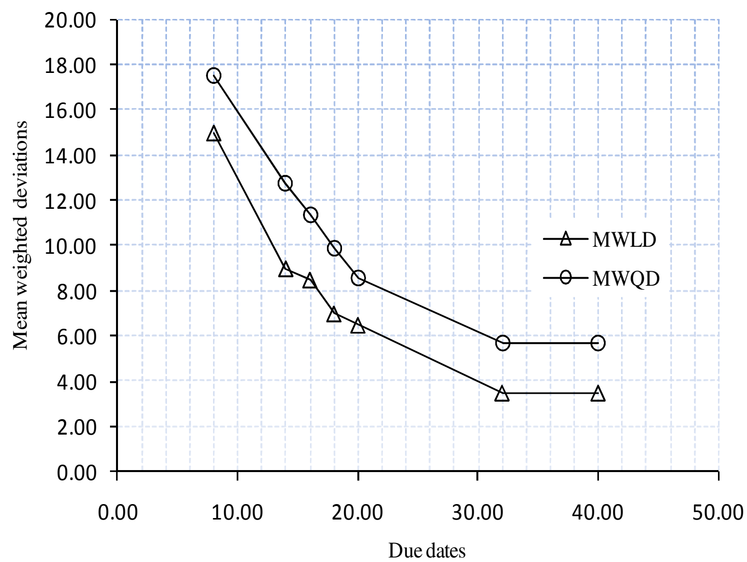


Figure 4.3 Variations of total penalty with due dates in a deterministic problem

It is also observed from Table 4.4 that the amount of idle time to insert increases as the due date tightness decreases, although the total penalty remains constant. Referring to Figure 4.3, at a due date $d = 32$, where improvements start to be realized using the IIT schedule, the penalties are minimum for both the linear and quadratic measures (*MWLD*=3.5 and *MWQD*=5.72) and remain constant for all $d > 32$. This illustrates a fundamental characteristic of optimal idle time insertion in a deterministic problem.

Effect of α:

If there can be no early jobs within a schedule, variations in α have no effect on the performance measure. This is due to the fact that α is a multiplier for penalizing earliness (linear penalty = Tardiness + α.Earliness, quadratic penalty = Tardiness^2+ α.Earliness$^{2)}$. However if at least one of the jobs in the set is early, variations in α will significantly affect the total penalty measure. As α is increased the total penalty tends to increase under this situation. Therefore the IIT scheduling is more beneficial with a larger α than with a smaller α. Table 4.5 illustrates the same problem as in Table 4.4 but with α =1.5. Improvements using IIT scheduling are noticeably larger than those obtained with α = 0.5.

Table 4.5 Deterministic solution with and without inserted idle times, α = 1.5

Linear Measure, deterministic approach, α =1.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	MWLD	Opt. Seq	I_1	I_2	MWLD	
Tight	8	A-B	15	A-B	0	0	15	-
	14	A-B	9	A-B	0	0	9	-
Medium	16	B-A	9	B-A	0	0	9	-
	18	B-A	7	B-A	0	0	7	-
	20	B-A	7.5	B-A	2	0	7	6%
	32	B-A	10.5	B-A	14	0	7	33.33%
Loose	40	B-A	22.5	B-A	22	0	7	68.89%

Quadratic Measure, deterministic approach, α =1.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	MWQD	Opt. Seq	I_1	I_2	MWQD	
Tight	8	A-B	17.49	A-B	0	0	17.49	-
	14	A-B	12.73	A-B	0	0	12.73	-
Medium	16	B-A	11.40	B-A	0	0	11.40	-
	18	B-A	9.89	B-A	0	0	9.89	-
	20	B-A	8.66	B-A	0	0	8.66	-
	32	B-A	12.12	B-A	8.40	0	7.67	36.72%
Loose	40	B-A	20.27	B-A	16.40	0	7.67	62.16%

All of the above results suggest that the optimal sequence will change depending on the degree of due date tightness. However, where this shift occurs will also depend on the value of α. For example, looking at the results of Table 4.4 and 4.5, sequence A-B was optimal until a due date $d = 14$ when $\alpha = 1.5$ was used. However with $\alpha = 0.5$, sequence A-B was optimal until $d = 16$. This suggests that a switchover occurs at a less tighter due date with lower values of α. These results establish that with deterministic processing times, the optimal sequence is a function of both the common due date d and the earliness tardiness weighting ratio α.

4.2.2 Scheduling based on stochastic times

The job processing times are now randomly drawn from a normal distribution with the mean and standard deviations given in Table 4.3. Common due dates of 40 are assumed. The same random numbers are maintained when experimenting with each performance measure and with different values of d and α.

Table 4.6 gives the results with α = 0.5. Similar to the deterministic results, these results show that significant performance improvements result when IIT schedules are used, especially with loose due dates. The optimal sequences with different due dates were also observed to be identical for the stochastic and deterministic approaches. However, differences were observed in the value of the objective function (performance measure) and the optimal idle times using the two approaches.

Table 4.6 Stochastic solution with and without inserted idle times, α = 0.5

Linear Measure, Stochastic approach, α =0.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	Ê[MWLD]	Opt. Seq	I_1	I_2	Ê[MWLD]	
Tight	8	A-B	15	A-B	0	0	15	-
	14	A-B	9.3	A-B	0	0	9.3	-
Medium	16	A-B	8.51	A-B	0	0	8.51	-
	18	B-A	7.3	B-A	0	0	7.3	-
	20	B-A	6.51	B-A	0	0	6.51	-
	32	B-A	3.93	B-A	0.63	0	3.89	1.01%
Loose	40	B-A	7.5	B-A	8.63	0	3.89	48.13%

Quadratic Measure, Stochastic approach, α =0.5

d		Non-IIT Schedule		IIT Schedule				% improvement in performance with IIT schedule
		Opt. Seq	Ê[MWQD]	Opt. Seq	I_1	I_2	Ê[MWQD]	
Tight	8	A-B	17.50	A-B	0	0	17.50	-
	14	A-B	12.74	A-B	0	0	12.74	-
Medium	16	A-B	11.37	A-B	0	0	11.37	-
	18	B-A	9.92	B-A	0	0	9.92	-
	20	B-A	8.57	B-A	0	0	8.57	-
	32	B-A	7.06	B-A	4.71	0	5.81	17.7%
Loose	40	B-A	11.72	B-A	12.71	0	5.81	50.43%

To understand the behaviour of optimal idle time, experiments were run for several values of α and a common due date of $d = 40$. Table 4.7 gives the results for the linear measure using both the stochastic and deterministic approaches. The optimal sequence is B-A in each of these cases.

Table 4.7 Comparing deterministic and stochastic solution at several values of α using linear measure

Linear measure, d=40

α	Deterministic solution		Stochastic solution	
	I_1	MWLD	I_1	Ê[MWLD]
0.5	8	3.920	8.576	3.887
0.7	8	5.376	9.315	5.215
0.9	8	6.832	10.298	6.447
1.1	22	7.422	20.313	7.106
1.3	22	7.462	20.826	7.248
1.5	22	7.503	21.155	7.359

For all $\alpha < 1$, as observed in Table 4.7, the optimal amount of idle time I_1 to be inserted before the first job in the sequence is larger if stochastic times are used than if the deterministic times are used. However, for all $\alpha > 1$, I_1 is smaller for a stochastic problem than a deterministic one. The linear penalty using the stochastic scheduling is always smaller than using deterministic scheduling however. These results demonstrate that scheduling in a deterministic manner when the problem is stochastic, may increase the linear penalty, as well as under or over estimate the best idle times.

Another key feature of deterministic IIT scheduling with two jobs and linear measures should also be noted. With a sufficiently loose due date, the shorter job is always positioned second (sequence B-A). The start or the finish time of this job will always be placed on the due date. Whether the start or the finish date is on the due date, depends on the relative weighting of earliness and tardiness, α. When $\alpha > 1$, it is the start time of the shorter job that falls on the due date. With $\alpha < 1$, it is the finish time of this job that falls on the due date. When $\alpha = 1$, the job straddles the due date (i.e. this job starts prior to the due date and completes after the due date).

When the quadratic measure was considered, both the optimal idle times and the total penalty were found to be greater for a stochastic problem than for a deterministic problem. The optimal sequence again places the longer job first (i.e. sequence B-A). As well, the penalty using deterministic scheduling approach to this problem is greater than using stochastic scheduling approach, although the differences in performance are small and less than those observed using the linear penalty function. Table 4.8 illustrates the results. These results were also consistent with those using several other due dates, where idle time was beneficial to improve performance.

Table 4.8 Comparing deterministic and stochastic solution at several values of α using quadratic measure

Quadratic measure, d=40

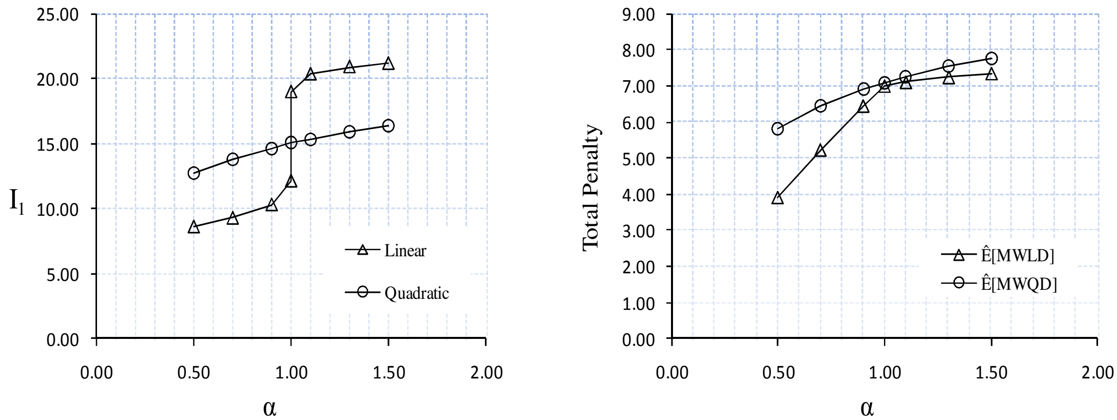
α	Deterministic solution		Stochastic solution	
	I_1	MWQD	I_1	$\hat{E}$[MWQD]
0.5	12.670	5.81084	12.712	5.81073
0.7	13.765	6.44322	13.806	6.44311
0.9	14.632	6.90310	14.669	6.90301
1.1	15.333	7.25535	15.367	7.25527
1.3	15.913	7.53505	15.951	7.53498
1.5	16.400	7.76322	16.429	7.76315

These results have demonstrated that scheduling in a deterministic manner for a stochastic problem can increase the quadratic penalty.

When the relative weightings on the earliness and tardiness are different $\alpha \neq 1$, the results in Table 4.7 and 4.8 also demonstrate that the minimum penalties (linear or quadratic) obtained are unique to using the listed optimal inserted idle times I_1. However, when both earliness and tardiness are equally penalized, $\alpha = 1$, the linear penalty is not unique to a particular idle time, although it is unique for the quadratic measure. For example, using $\alpha = 1$ in the above stochastic problem means the optimal idle time could range anywhere from 12.14 to 18.97 but the total linear penalty would still be 7. For a deterministic problem the range of optimal idle time with $\alpha = 1$ varied from 8 to 22.

Using the quadratic measure, an optimal idle time of 15.035 resulted in a minimum penalty of 7.089 in a stochastic problem and an optimal idle time of 15 with a minimum penalty of 7 in a deterministic problem. The minimum penalties were unique to the specific optimal idle times obtained.

The stochastic scheduling results showing optimal idle times and total penalties with different α values are given in Figure 4.4



(a) (b)

Figure 4.4 Variation of optimal idle time and total penalties for several α in a stochastic problem

Figure 4.4 (a) results show that although the optimal idle times I_1 increase with increasing α, these relationships are not linear. Further a single unique optimal idle time is obtained for all measures of α using a quadratic penalty. However, multiple optimal idle times are obtained when $\alpha = 1$ using a linear penalty. In Figure 4.4 (b) it can be observed that the total penalty increases when α increases. At $\alpha = 1$ the linear and quadratic penalties agree quite closely. This is a function of using the root mean square of the quadratic penalty function as the performance measure.

4.2.3 Influence of processing time variability

This section concentrates on investigating how the variability in processing times influences the optimal schedule.

4.2.3.1 Scheduling when expected values of processing times are identical

Two job types are again designated as A and B. Initially, it is assumed that both jobs have a mean processing time of 10, a standard deviation of 1 and that $\alpha = 0.5$. The multipler b is set at 1, meaning the common due date d is 20, equal the sum of the two mean processing times.

If the problem were deterministic, then obviously either sequence (A-B or B-A) would be optimal because the total deviations are the same with either sequence under identical job processing times. Also, as observed earlier in section 4.2.1, when earliness weighting is less than tardiness $\alpha < 1$, idle time would be unnecessary if linear deviations are to be minimized although this is beneficial when quadratic penalty needs to be minimized. The deterministic problem results based on $\alpha = 0.5$ are summarized for optimal sequences, the idle times and the penalties in Table 4.9.

Table 4.9 Deterministic problem solution under identical processing times, α = 0.5

Penalty measures and resulting minimum values		Optimal Initial Idle Time, I_1	Optimal Sequence
Linear, *MWLD*	2.5000	0.0000	Any permutation
Quadratic, *MWQD*	4.0825	3.3333	Any permutation

If the earliness and tardiness were equally penalized ($\alpha = 1$), then as observed in the case of a single-job problem, the quadratic deviation would be minimum if the due date were positioned exactly in the middle of the straddling job. That would mean an idle time I_1 =

5 would be optimal for this problem. However, with $\alpha < 1$, tardiness is relatively undesirable so the idle time drops to 3.33, as is observed in the results of Table 4.9. If $\alpha > 1$, then the idle time would obviously be > 5.

The stochastic problem is now considered. Three independent replications are run considering both the linear and quadratic penalties as the objective functions. Table 4.10 illustrates the optimal idle time and the total penalties using a stochastic scheduling approach to this problem. Additionally, the deterministic solution results, which are based on using the deterministic optimal idle times (of Table 4.9) in a stochastically generated problem set is also provided in the last column of this Table.

Table 4.10 Stochastic scheduling under identical expected processing times, α = 0.5

Measure	Opt. Sequence	Stochastic Solution		Deterministic Solution	
		Idle Time	Penalty	Idle Time	Penalty
Linear Ê[MWLD]	Any permutation	0.57568	2.88724	0	2.92005
	Any permutation	0.59187	2.89362	0	2.92901
	Any permutation	0.63876	2.89028	0	2.93005
Quadratic Ê[MWQD]	A-B	3.38834	4.21281	3.33333	4.21307
	B-A	3.38644	4.21173	3.33333	4.21197
	A-B	3.39238	4.21222	3.33333	4.21252

The results in this section demonstrate that the penalties increase when deterministic optimal idle times are used in a stochastically generated problem set with either performance measure. The two sequences are indistinguishable resulting in the same penalty when linear measure is considered. In the quadratic measure the optimal sequence obtained is unique as observed in the Table 4.10. But this is a function of the sampling done in the experiment. Over many experiments, the expected values of the penalty for both sequences will be the same with either sequence for a quadratic measure when the process time means and standard deviations are identical.

Two jobs with identical expected values of the processing times but different variability (denoted by standard deviations) are now considered (i.e. $\mu_A = \mu_B$, $\sigma_A \neq \sigma_B$). It is important to realize that the variability in any preceding job in a sequence propagates through the succeeding jobs.

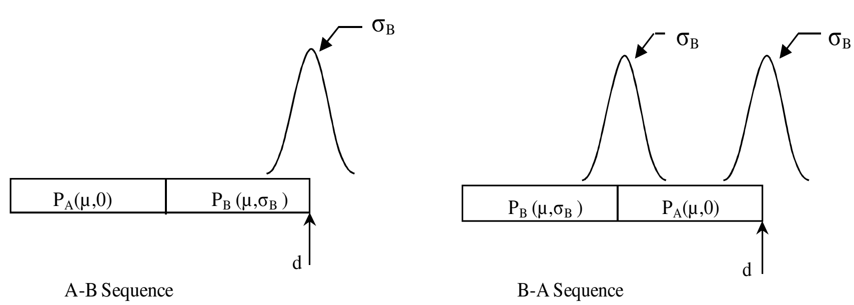


Figure 4.5 Illustration of the propagation of uncertainty

Figure 4.5 illustrates a situation where Job A is certain (no variability) and Job B is uncertain (variability) in its processing time. Both jobs have the same mean processing time, μ. If the completion time of the first job is the starting time of the second job, then the standard deviation of total completion time is $\sqrt{\sigma_B^2 + \sigma_A^2}$ since variances are additive. In the illustrated example, this implies the variance of completion times for both jobs is σ_B^2 if B-A is the sequence, while the variance of completion times is 0 and σ_B^2 if A-B is the sequence. Therefore, when the performance objective is based on due dates, sequence

A-B is more favourable. In particular, if jobs have identical mean processing times but different degrees of uncertainty it is advantageous to sequence the jobs on the basis of variability with the most uncertain job sequenced in the last position. This result has also been confirmed using example problems with more than two jobs. For example, Figure 4.6 shows the optimal schedule based on minimizing the quadratic penalty when there are 3 jobs (A, B, C) to be scheduled with identical mean processing times (μ = 10) but different variances. Using $\sigma*$ to represent standard deviations of completion times, this figure also shows how uncertainty gets transmitted from one job to the next.

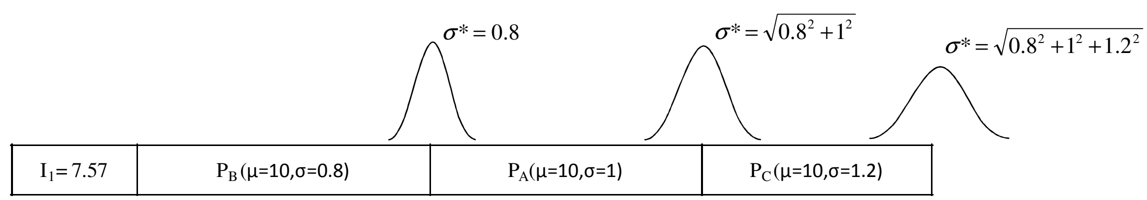


Figure 4.6 Scheduling jobs under identical means but different variability

The deterministic approach to this problem would suggest any sequence to be optimal. The results here demonstrate that the optimal sequence resulting from scheduling based on expected times may not be the true optimal sequence for a stochastic problem. Uncertainties influence the sequencing decisions even if the mean processing times of the jobs do not.

4.2.3.2 Scheduling when expected values of processing times are different

If the mean processing times of the jobs are also different, the optimal sequence is governed by whether the mean dominates the variability or vice versa. To demonstrate this, it was necessary to have the mean length of the jobs differ by a small amount. If the differences are large then the mean length would always be dominant and the influence of variability could not be well understood. Further, in the analysis in Section 4.2.3.1, it was demonstrated that quadratic penalties yield unique sequences and are therefore better measures in terms of studying with the effects of variability. The quadratic measure is therefore considered to illustrate the relationship of interest. Jobs A and B are assumed to have mean processing times of 10.5 and 10 respectively.

With equal variability for both jobs, common due dates equal to the cumulative mean processing time (d = 20.5) and $\alpha < 1$, the longer job (Job A) should always be positioned first in the sequence. This is the same optimal sequence we would get if the processing times were considered to be deterministic. These results have already been established in Sections 4.2.1 and 4.2.2.

To understand how the processing time variability affects this sequence, several values of variability on Job A processing time, P_A (=10.5), were assumed while holding Job B processing time, P_B, constant at 10. The earliness tardiness weighting ratio, α was set at 0.5.

Table 4.11 illustrates the average results with 95 % confidence interval half widths obtained from five independent replications of each experiment. A plot of the variability of the first job, expressed as standard deviations, against the average *Ê[MWQD]* is shown in Figure 4.7. The resulting (optimal) idle times and the minimum expected quadratic penalty with the optimal sequences are highlighted in Table 4.11.

Table 4.11 Results demonstrating job variability effects on two sequences, α = 0.5

	Seq. A-B		Seq. B-A	
σ_A	I_1	Ê[MWQD]	I_1	Ê[MWQD]
1.00	3.3333±0.0016	4.1715±0.0002	3.5612±0.0003	4.3256±0.0003
1.50	3.3302±0.0063	4.2764±0.0008	3.6299±0.0021	4.3743±0.0012
2.00	3.3246±0.0050	4.4128±0.0015	3.7084±0.0068	4.4418±0.0022
2.50	3.2896±0.0146	4.5727±0.0033	3.7833±0.0062	4.5282±0.0043
3.00	3.2928±0.0096	4.7541±0.0046	3.8773±0.0066	4.6219±0.0023
3.50	3.2597±0.0456	4.9444±0.0219	3.9521±0.0328	4.7266±0.0065

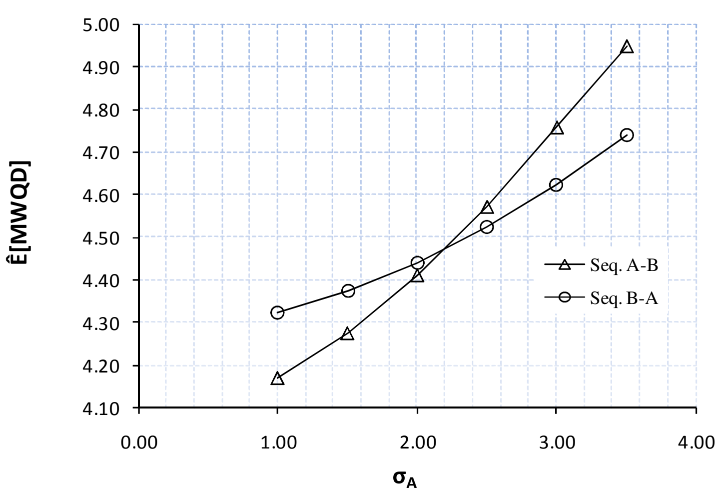


Figure 4.7 Illustration of job variability effect on the two sequences

As observed in Figure 4.7, at a standard deviation of approximately 2.2 on Job A mean processing time, either sequence is optimal. Above this value of the standard deviation, the longer mean processing time for Job A, which tends to position this job first, is offset by its higher variability, which tends to position it later. Therefore, for standard deviations above the critical value of about 2.2, sequence B-A is optimal and for values below the critical, sequence A-B is optimal.

Scheduling in a deterministic manner would yield A-B as the optimal sequence, which is observed to be correct only under low variability of Job A . The results here demonstrate that under uncertainty, the optimal position of a job is dependent on how variable each job is. Especially when job processing times are highly uncertain, scheduling on the basis of expected process times may lead to a non-optimal sequence and an inappropriate choice of idle times, with consequently higher penalties.

The critical value of the standard deviation where the sequence switchover occurs is a function of the difference of the mean processing times, δ of the two jobs for a given due date and the α. A predictive relationship can be established to estimate this critical standard deviation using the regression analysis. The regression analysis to predict σ at switchover is now presented.

<u>Predictive relationship for σ at sequence switchover</u>

Job B is fixed at a mean processing time of 10 while three levels of mean processing time for Job A are considered as shown in Table 4.12. Due date is fixed at d = 20.5 and α is maintained at 0.5.

Table 4.12 Inputs for predictive relationship analysis

μ_A	μ_B	δ	σ_A
10.1	10	0.1	0.5,1.0,1.5,2.0,2.5,3.0
10.3	10	0.3	0.5,1.0,1.5,2.0,2.5,3.0
10.5	10	0.5	0.5,1.0,1.5,2.0,2.5,3.0

Optimal solution with each sequence are obtained for each value of the standard deviations in Table 4.12. The results are plotted in Figure 4.8.

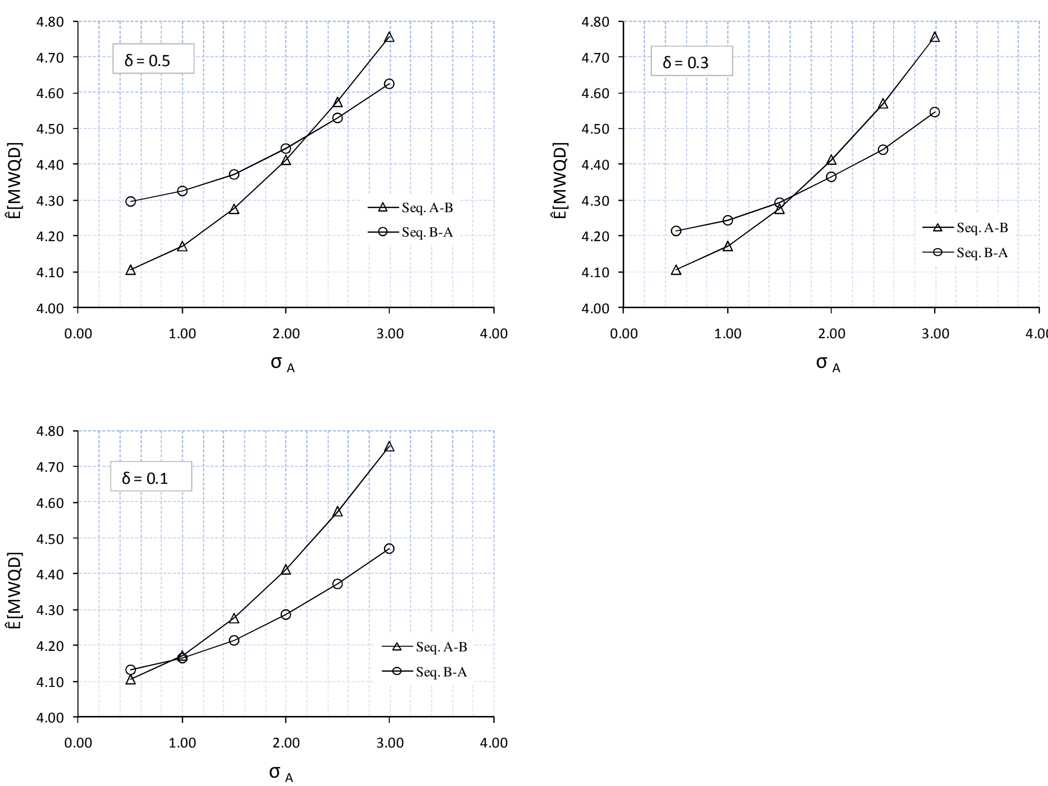


Figure 4.8 Variation of optimal sequence switchover with difference on mean processing times (δ) of the jobs

As observed from these results, the switchover occurs at different magnitudes of processing time variability on Job A, as the difference of the mean processing times of the two jobs is changed. However, it should be noted that the position of the curve for sequence AB remains the same at each δ. The regression analysis using AB sequence on the minimum penalty as the response is shown below. These results are obtained using Minitab15®:

Regression Analysis: $\hat{E}[MWQD]_{AB}$ versus σ_{AB}, σ^2_{AB}

$$\hat{E}[MWQD]_{AB} = 4.06 + 0.0601\, \sigma_{AB} + 0.0578\, \sigma^2_{AB} \tag{4.3}$$

Predictor	Coef	SE Coef	T	P
Constant	4.05776	0.00373	1086.78	0.000
σ_{AB}	0.060102	0.004885	12.30	0.000
σ^2_{AB}	0.057844	0.001366	42.33	0.000

S = 0.00361519   R-Sq = 100.0%   R-Sq(adj) = 100.0%

The regression analysis using BA as the sequence revealed δ also significant at 5% level of significance in addition to the linear and squared standard deviation factors.

Regression Analysis: $\hat{E}[MWQD]_{BA}$ versus σ_{BA}, σ^2_{BA}, δ_{BA}

$$\hat{E}[MWQD]_{BA} = 4.08 + 0.0150\, \sigma_{BA} + 0.0338\, \sigma^2_{BA} + 0.398\, \delta_{BA} \tag{4.4}$$

Predictor	Coef	SE Coef	T	P
Constant	4.07825	0.00329	1238.10	0.000
σ_{BA}	0.014970	0.003975	3.77	0.002
σ^2_{BA}	0.033826	0.001112	30.43	0.000
δ_{BA}	0.397650	0.004245	93.67	0.000

S = 0.00294131   R-Sq = 100.0%   R-Sq(adj) = 100.0%

At the switchover point the minimum penalty with either sequence are identical. Based on the above regression equations, the predictive relationship to estimate the critical standard deviation where the switchover occurs can thus be obtained as:

$$\delta = \frac{1}{0.398}\left[0.024\sigma_A^{\,2} + .0451\sigma_A - 0.02\right] \tag{4.5}$$

As an illustration, the following values are obtained for the range of δ being considered in this problem. The plot for the predictive relationship in this range of δ is given in Figure 4.9.

δ	0.100	0.123	0.147	0.173	0.199	0.227	0.255	0.285	0.317	0.349	0.383	0.418	0.454	0.500
σ_A	0.898	1.000	1.100	1.200	1.300	1.400	1.500	1.600	1.700	1.800	1.900	2.000	2.100	2.224

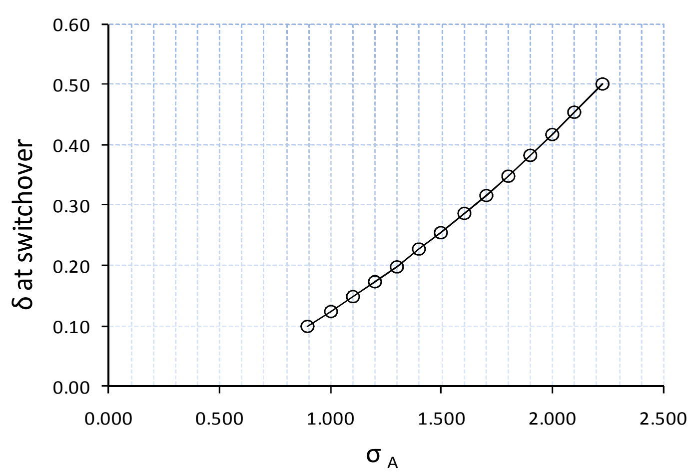


Figure 4.9 Predictive relationship plot of variability against difference of mean processing times at switchover

4.3 Analytical solution of a two-job stochastic problem

To this stage, analytical optimization models have been developed only for deterministic or single-job problems in this research. Two job stochastic problems have only been examined experimentally. In this section two job, single-machine problems are examined analytically.

Although the two-job stochastic problem can be formulated for normally distributed processing times in a manner similar to the single-job problem described in Section 4.1, finding the optimal solution is computationally difficult because of the resulting double integral and the fact the distribution is unbounded distribution. However, determining optimal decision variables is feasible if a simpler, bounded distribution is used, such as the continuous uniform distribution. An analytical formulation of a stochastic, two-job problem using a uniform processing time distribution is presented, along with some example solutions based on minimization using Mathematica®.

Jobs can be scheduled to complete early, tardy or on time. The scenario matrix in Table 4.13 is exhaustive for the two-job problem.

Table 4.13 Early and tardy jobs scenario for a two job problem

		Job in 1st position		
		Early	Tardy	On time
Job in 2nd position	Early	√	-	-
	Tardy	√	√	√
	On time	√	-	-

Depending on the relative position of the due dates, three feasible earliness-tardiness scenarios can be realized. These are illustrated in Figure 4.10, while the associated total weighted linear penalties are given in Table 4.14.

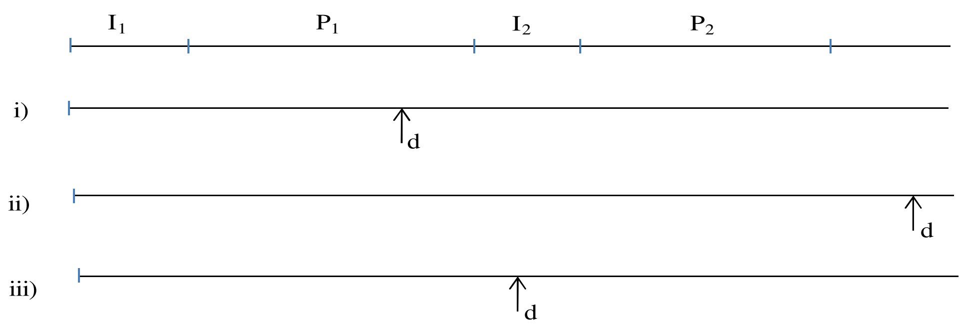


Figure 4.10 Two job problem realizations

Table 4.14 Total linear penalty and due date ranges for the two-job realizations

Scenario	Due date position	Total weighted linear penalty
(i) Both Jobs Tardy[TT]	$d < I_1+P_1 < I_1+P_1+ I_2+P_2$	$(I_1+P_1 - d) + (I_1+P_1 + I_2+P_2 - d)$
(ii)Both Jobs Early[EE]	$d > I_1+P_1+ I_2+P_2 > I_1+P_1$	$\alpha(d - I_1 - P_1) + \alpha(d - I_1 - P_1 - I_2 - P_2)$
(iii)1st Job Early, 2nd Job Tardy[ET]	$I_1+P_1 < d < I_1+P_1+I_2+P_2$	$\alpha(d - I_1 - P_1) + (I_1+P_1 + I_2+P_2 - d)$

Here, P_i (i = 1, 2) represents the processing time realized for job i and I_j (j = 1, 2) represents the idle time inserted before the j^{th} job in the sequence. Sequence 1-2 of jobs is considered to be Job 1 in position 1 and Job 2 in position 2.

Referring to the geometry of Figure 4.10, if the first job finishes exactly on time, then it satisfies $d = I_1+P_1$ resulting in:

$$P_1 = d - I_1 \tag{4.6}$$

If the second job finishes exactly on time, then this must satisfy $d = I_1+P_1 + I_2+P_2$, for which we obtain:

$$P_2= d- I_1 - P_1 - I_2 = d - I_1 - I_2 - P_1 \qquad (4.7)$$

Equation 4.6 gives the equation of a line with a constant distance from the ordinate (vertical axis), which will be referred to from now on as a 'vertical' line. Equation 4.7 represents a line which can be expressed in the slope-intercept form with a unit negative slope (-1) and a vertical axis intercept $d - I_1 - I_2$. The intercept depends on the values of I_1 and I_2 since d is fixed. This line is referred as a 'diagonal' line. These two lines are shown in Figure 4.11.

Considering the processing times as being uniformly distributed, Jobs 1 and 2 are arbitrarily labelled as $P_1 \sim U(a_1, b_1)$ and $P_2 \sim U(a_2, b_2)$. Given these bounds on the processing times (maximum and minimum) and on considering the sequence 1-2 of the jobs, a rectangular region with four corner points as the 'extremes' can be used to represent the sample (domain) space of the processing times as shown by the shaded portion of the Figure 4.11. This sample space will be fixed for a given problem. The optimization problem is then to appropriately lay the vertical and the diagonal lines within/outside this sample space such that the total penalty derived in Table 4.14 is minimized.

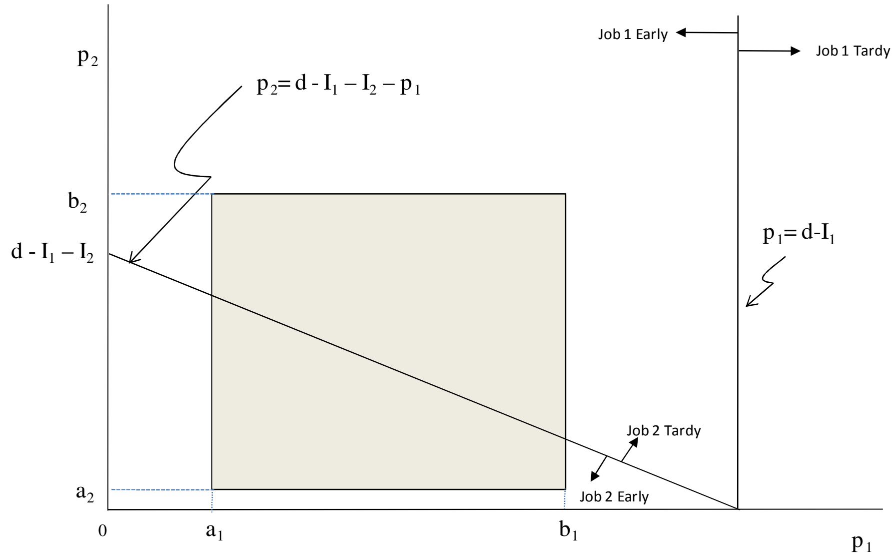


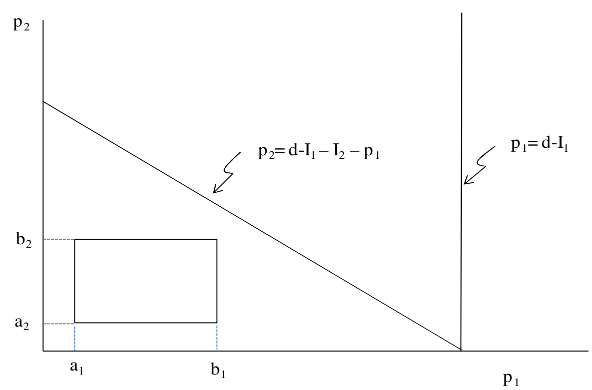
Figure 4.11 Representation of a two job problem with uniformly distributed processing times

Several possible ways of laying the two lines within/outside the sample space give rise to the cases whereby some or all of the corner points lie above/below and left /right of the diagonal and vertical lines respectively. A total of 10 such cases was found to exhaustively represent the problem. However, the necessary constraints on I_1 and I_2 limit the number of cases that can be realized for a given problem instance with known parameters d, a_1, a_2, b_1, b_2 and α. It should be noted that for a particular case to be feasible, all of the constraints on I_1 and I_2 for this case have to be simultaneously met.

The ten possible cases are illustrated in the following paragraphs with the total expected weighted linear penalty and the constraints on I_1 and I_2 for each case. The probability density functions of the processing times P_1 and P_2 (random variables) for the two jobs in each of these cases are given as $f_{P_1}(p_1) = \dfrac{1}{b_1 - a_1}$ and $f_{P_2}(p_2) = \dfrac{1}{b_2 - a_2}$.

Case (i):

All corner points of the rectangle are below the diagonal line. This is the situation where both jobs will be early.

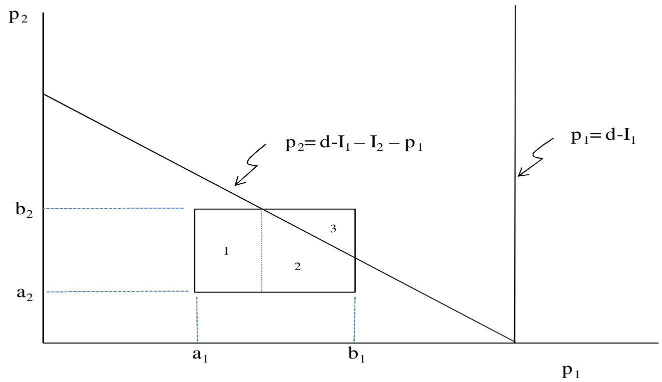


$$E[Penalty(i)] = \int_{p2=a2}^{b2} \int_{p1=a1}^{b1} \alpha[(d - I_1 - p_1) + (d - I_1 - p_1 - I_2 - p_2)] f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$Constraints: 0 \le I_1 + I_2 \le d - b_1 - b_2; 0 \le I_1 \le d - b_1;\ d \ge I_1 \ge 0;\ d \ge I_2 \ge 0;$$

Case (ii):

One of the corner points (b1, b2) is above the diagonal line. The total expected penalty can be calculated by summing the expected penalties over the three regions as illustrated below.



$$E[Penalty(ii)] =$$

$$\int_{p2=a2}^{b2} \int_{p1=a1}^{d-I_1-I_2-b2} \alpha[(d - I_1 - p_1) + (d - I_1 - p_1 - I_2 - p_2)] f_{P1}(p_1) f_{P2}(p_2)\, dp_1\, dp_2$$

$$+$$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{p1=d-I_1-I_2-b2}^{b1} \alpha[(d - I_1 - p_1) + (d - I_1 - p_1 - I_2 - p_2)] f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

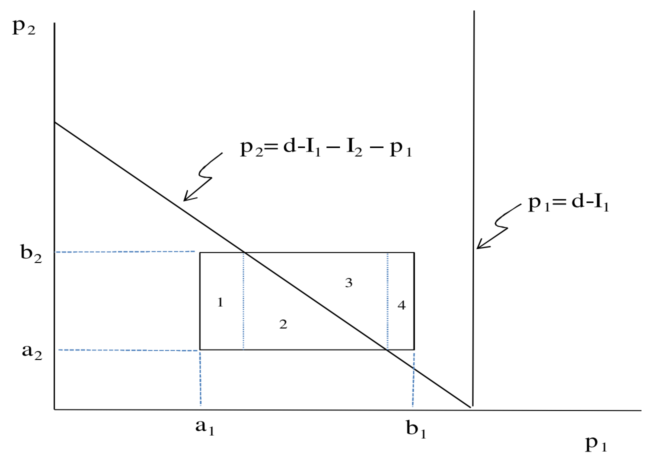
$$+$$

$$\int_{p2=d-I_1-I_2-p1}^{b2} \int_{p1=d-I_1-I_2-b2}^{b1} [\alpha(d - I_1 - p_1) + (I_1 + p_1 + I_2 + p_2 - d)] f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$\begin{aligned} Constraints: & \; d - b_1 - b_2 \le I_1 + I_2 \le d - a_1 - b_2; 0 \le I_1 + I_2 \le d - a_2 - b_1; \\ & \; 0 \le I_1 \le d - b_1; \\ & \; d \ge I_1 \ge 0;\; d \ge I_2 \ge 0; \end{aligned}$$

Case (iii):

Corner points (b1, b2) and (b1, a2) lie above the diagonal line. The total expected penalty can be found by dividing the rectangle into four areas and summing up the expected penalty for each area.



$$E[Penalty(iii)] =$$

$$\int_{p2=a2}^{b2} \int_{p1=a1}^{d-I_1-I_2-b2} \alpha\left[\left(d - I_1 - p_1\right) + \left(d - I_1 - p_1 - I_2 - p_2\right)\right] f_{P1}(p_1) f_{P2}(p_2)\, dp_1\, dp_2$$

$$+$$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{d-I_1-I_2-b2}^{d-I_1-I_2-a2} \alpha\left[\left(d - I_1 - p_1\right) + \left(d - I_1 - p_1 - I_2 - p_2\right)\right] f_{P1}(p_1) f_{P2}(p_2)\, dp_1\, dp_2$$

$$+$$

$$\int_{p2=d-I_1-I_2-p1}^{b2} \int_{p1=d-I_1-I_2-b2}^{d-I_1-I_2-a2} \left[\alpha\left(d - I_1 - p_1\right) + \left(I_1 + p_1 + I_2 + p_2 - d\right)\right] f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$+$$

$$\int_{p2=a2}^{b2} \int_{d-I_1-I_2-a2}^{b1} \left[\alpha\left(d - I_1 - p_1\right) + \left(I_1 + p_1 + I_2 + p_2 - d\right)\right] f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

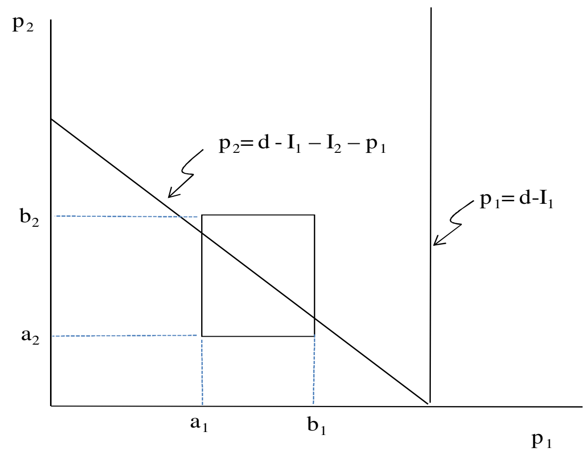
Constraints : $d - b_1 - b_2 < I_1 + I_2 \le d - a_1 - b_2 ; I_1 + I_2 > d - a_2 - b_1 ;$

$0 \le I_1 \le d - b_1 ;$

$d \ge I_1 \ge 0;\ d \ge I_2 \ge 0;$

Case(iv):

Corner points (b1, b2) and (a1, b2) lie above the diagonal line. The total expected penalty can be found by summing up the expected penalty over the two regions of the rectangle which are above and below the diagonal line.



$$E[Penalty(iv)] =$$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{p1=a1}^{b1} \alpha[(d - I_1 - p_1) + (d - I_1 - p_1 - I_2 - p_2)] f_{P1}(p_1) f_{P2}(p_2) dp_1 \, dp_2$$

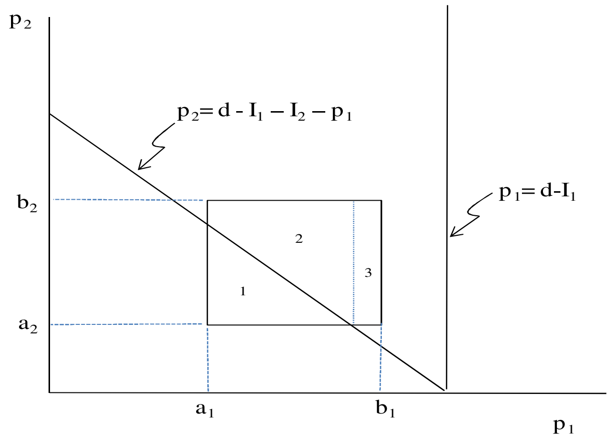
$$+$$

$$\int_{p2=d-I_1-I_2-p1}^{b2} \int_{p1=a1}^{b1} [\alpha(d - I_1 - p_1) + (I_1 + p_1 + I_2 + p_2 - d)] \; f_{P1}(p_1) f_{P2}(p_2) dp_1 \, dp_2$$

$$\begin{aligned} Constraints: \; & d - b_1 - b_2 < I_1 + I_2 \le d - b_1 - a_2; I_1 + I_2 > d - b_2 - a_1; \\ & 0 \le I_1 \le d - b_1; \\ & d \ge I_1 \ge 0; \; d \ge I_2 \ge 0; \end{aligned}$$

Case (v):

Three corner points (b1, b2), (b1, a2), (a1, b2) are above the diagonal line. Again the total expected penalty is the sum of expected penalties with each of the three areas considered.



$$E[Penalty(v)] =$$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{p1=a1}^{d-I_1-I_2-a2} \alpha[(d-I_1-p_1)+(d-I_1-p_1-I_2-p_2)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$+$$

$$\int_{p2=d-I_1-I_2-p1}^{b2} \int_{p1=a1}^{d-I_1-I_2-a2} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

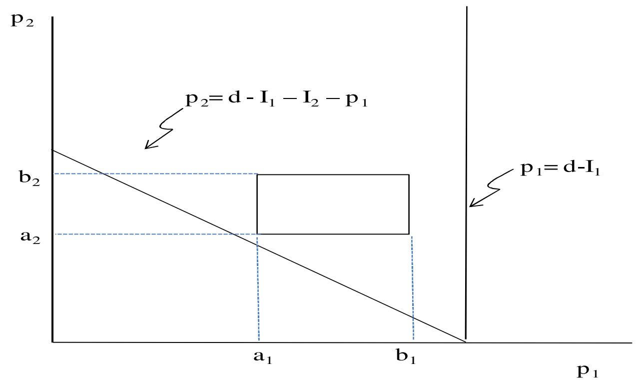
$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1-I_2-a2}^{b1} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$Constraints: d-b_2-a_1 < I_1+I_2 \le d-a_2-a_1; I_1+I_2 > d-a_2-b_1;$$
$$0 \le I_1 \le d-b_1;$$
$$d \ge I_1 \ge 0;\ d \ge I_2 \ge 0;$$

Case (vi):

All the corner points above the diagonal line. This is the case when second job will definitely be tardy while the first job is early.

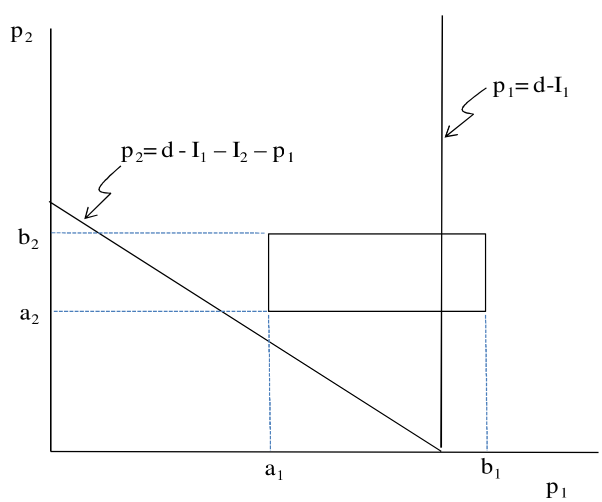


$$E\left[Penalty\,(vi)\right]=\int_{p2=a2}^{b2}\int_{p1=a1}^{b1}\left[\alpha\left(d-I_1-p_1\right)+\left(I_1+p_1+I_2+p_2-d\right)\right]f_{P1}(p_1)\,f_{P2}(p_2)\,dp_1\,dp_2$$

$$\begin{aligned}Constraints:\ & I_1+I_2>d-a_2-a_1;\\ & 0\le I_1\le d-b_1;\\ & d\ge I_1\ge 0;\ d\ge I_2\ge 0;\end{aligned}$$

Case (vii):

All corner points above the diagonal line with corner points (b1, b2) and (b1, a2) to the right of the vertical line. The total expected penalty is the sum of expected penalties over the two areas of the rectangle divided by the vertical line.



$$E[Penalty(vii)] =$$

$$\int_{p2=a2}^{b2} \int_{p1=a1}^{d-I_1} [\alpha(d - I_1 - p_1) + (I_1 + p_1 + I_2 + p_2 - d)] \; f_{P1}(p_1) f_{P2}(p_2) dp_1 \; dp_2$$

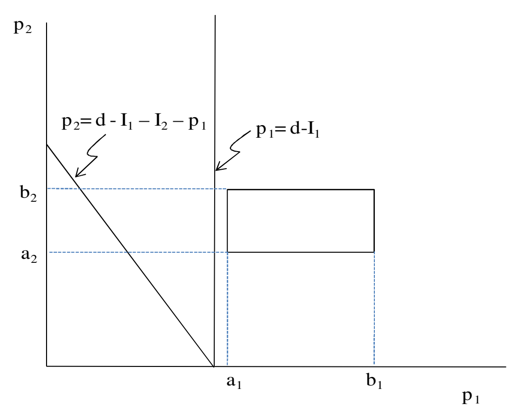
$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1}^{b1} [(I_1 + p_1 - d) + (I_1 + p_1 + I_2 + p_2 - d)] \; f_{P1}(p_1) f_{P2}(p_2) dp_1 \; dp_2$$

$$\begin{aligned} Constraints: & \; I_1 + I_2 > d - a_2 - a_1; \\ & \; d - b_1 < I_1 \le d - a_1; \\ & \; d \ge I_1 \ge 0; \; d \ge I_2 \ge 0; \end{aligned}$$

Case (viii):

All corner points lie to the right of the vertical line and above the diagonal line. Both jobs are tardy.

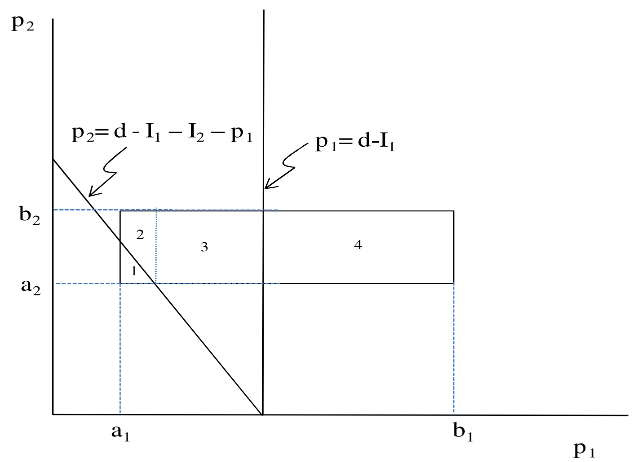


$$E[Penalty(viii)] = \int_{p2=a2}^{b2} \int_{p1=a1}^{b1} [(I_1 + p_1 - d) + (I_1 + p_1 + I_2 + p_2 - d)] \; f_{P1}(p_1) f_{P2}(p_2) dp_1 \; dp_2$$

$$\begin{aligned} Constraints: & \; I_1 + I_2 > d - a_2 - a_1; \\ & \; I_1 > d - a_1; \\ & \; d \ge I_1 \ge 0; \; d \ge I_2 \ge 0; \end{aligned}$$

Case (ix):

Corner point (a1, a2) is below the diagonal line, all other corner points are above this line with corner points (b1, b2) and (b1, a2) to the right of the vertical line. The total expected penalty is the sum of expected penalties over the four regions shown.



$E[Penalty(ix)] =$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{p1=a1}^{d-I_1-I_2-a2} \alpha[(d-I_1-p_1)+(d-I_1-p_1-I_2-p_2)]\ f_{P1}(p_1)f_{P2}(p_2)dp_1\,dp_2$$

$$+$$

$$\int_{p2=d-I_1-I_2-p1}^{b2} \int_{p1=a1}^{d-I_1-I_2-a2} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\ f_{P1}(p_1)f_{P2}(p_2)dp_1\,dp_2$$

$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1-I_2-a_2}^{d-I_1} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\ f_{P1}(p_1)f_{P2}(p_2)dp_1\,dp_2$$

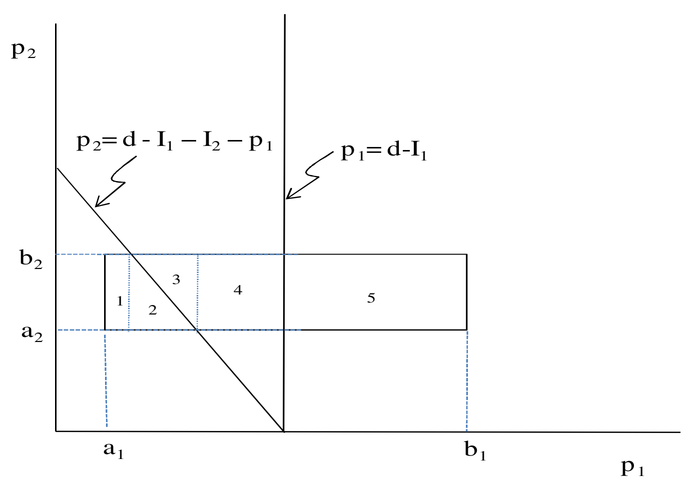
$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1}^{b1} [(I_1 + p_1 - d) + (I_1 + p_1 + I_2 + p_2 - d)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$\begin{aligned} Constraints: & \; d - b_2 - a_1 < I_1 + I_2 \le d - a_2 - a_1; I_1 + I_2 > d - a_2 - b_1; \\ & \; d - b_1 < I_1 \le d - a_1; \\ & \; d \ge I_1 \ge 0;\; d \ge I_2 \ge 0; \end{aligned}$$

Case (x):

Two corner points below the diagonal line are to the left of the vertical line and the two corner points above the diagonal line are to the right of the vertical line. The total expected penalty is the sum of penalties over the five regions shown.



$E[Penalty(x)] =$

$$\int_{p2=a2}^{b2} \int_{p1=a1}^{d-I_1-I_2-b2} \alpha[(d - I_1 - p_1) + (d - I_1 - p_1 - I_2 - p_2)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$+$

$$\int_{p2=a2}^{d-I_1-I_2-p1} \int_{p1=d-I_1-I_2-b2}^{d-I_1-I_2-a2} \alpha[(d-I_1-p_1)+(d-I_1-p_1-I_2-p_2)]\, f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$+$$

$$\int_{d-I_1-I_2-p1}^{b2} \int_{p1=d-I_1-I_2-b2}^{d-I_1-I_2-a2} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\ f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2)$$

$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1-I_2-a_2}^{d-I_1} [\alpha(d-I_1-p_1)+(I_1+p_1+I_2+p_2-d)]\ f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$+$$

$$\int_{p2=a2}^{b2} \int_{p1=d-I_1}^{b1} [(I_1+p_1-d)+(I_1+p_1+I_2+p_2-d)]\ f_{P1}(p_1) f_{P2}(p_2) dp_1\, dp_2$$

$$\begin{aligned} Constraints: & \; d-a_2-b_1 < I_1+I_2 \le d-b_2-a_1; \\ & \; d-b_1 < I_1 \le d-a_1; \\ & \; d \ge I_1 \ge 0;\ d \ge I_2 \ge 0; \end{aligned}$$

Using Mathematica®, the total expected penalties for each of the cases (i) through (x) can be obtained in terms of I_1 and I_2 once all the other known parameters are defined. These can then be minimized with respect to I_1 and I_2 for each of the possible cases. This is repeated with the sequence Jobs 1-2 and then again with sequence Jobs 2-1. The optimal idle times I_1 and I_2 and sequence are therefore those which yield the minimum among all the possible cases.

Similar analysis can be performed for both the linear and a quadratic penalty functions. However, the square root in the quadratic formulation (i.e. using the root mean square) presents additional computational difficulties when using Mathematica®. Therefore the squared deviations, rather than the root mean squares, were used in minimizing the penalty function. This did not make any difference with respect to the optimal sequence and the idle times.

Table 4.15 Comparison of simulation and analytical results for a uniform distribution

Uniform distribution, low variability: $P_A \sim U(19, 21)$, $P_B \sim U(19.5, 20.5)$, $d = 40$, $\alpha = 0.5$

	Linear Measure			Quadratic Measure		
	Opt.Seq	I_1	Total Penalty	Opt.Seq	I_1	Total Penalty*
Simulation Results	(B-A)	0.334±0.016	5.184±0.005	B-A	6.6666	66.893±0.005
Analytical Results	(B-A)	0.3333	5.18229	B-A	6.6666	66.8958

Uniform distribution, high variability: $P_A \sim U(10,30)$, $P_B \sim U(12, 28)$, $d = 40$, $\alpha = 0.5$

	Linear Measure			Quadratic Measure		
	Opt.Seq	I_1	Total Penalty	Opt.Seq	I_1	Total Penalty*
Simulation Results	(B-A)	3.471±0.105	7.056±0.029	B-A	6.403±0.009	98.107±0.523
Analytical Results	(B-A)	3.3941	7.0657	B-A	6.3954	98.2078
**based on mean squared deviation only(without square root)*						

Table 4.15 compares the experimental results, using simulation and the Evolutionary Solver®, with the analytical results. High and low processing time variability with identical means is considered. The simulation results are based on the averages across ten independent replications of an experiment using different random numbers. The same random numbers were used for experimenting with the two penalty measures. The analytical and simulation results correspond quite closely.

4.4 Summary

The single machine scheduling problem with a common due date performance objective was analyzed in this chapter. In Section 4.1 stochastic single-job problem was used to demonstrate that optimal idle times are different if scheduling is based on expected times rather than stochastic times. It was demonstrated, using both the experimental and analytical approaches, that the optimal solution using the stochastic times was dependent on the relative weightings on earliness and tardiness. This section also introduced an analytical approach to finding the total earliness and tardiness penalties using Mathematica®. As well, use of statistical loss functions, or partial expectations, was demonstrated as a way to optimally solve single-job problems with normally distributed processing times. Later, in Section 4.3 analytical solutions to solve the two job problem were also presented. Due to the difficulties in analytically dealing with the normal distribution, this was demonstrated using a uniform distribution.

In Section 4.2 two job problems were considered. Using both the deterministic and stochastic problems, it was shown that scheduling with inserted idle time is beneficial when due dates are sufficiently loose, or unrestricted. As well, it may be beneficial with tighter due dates when earliness is penalized more heavily than tardiness. Considering several degrees of due date tightness and different relative weightings of earliness and tardiness, it was demonstrated that optimal sequences were dependent on both these two factors. In particular it was shown that the optimal sequence changed at some combination of the due date tightness and E/T weighting ratio. This switchover could occur at a less tight due date with lower values of the E/T weighting ratio.

It was also demonstrated in Section 4.2 that even though the optimal sequences may be the same, both the linear and quadratic penalty may increase if scheduling is done on the basis of expected times for a stochastic problem. An interesting observation made in this section is that while the quadratic penalty measure yields a unique optimal idle time for all relative weightings of earliness and tardiness, the linear does not. In particular, with equal weighting on both earliness and tardiness, multiple optimal idle times result.

The influence of processing time variability on sequencing decisions was studied in Section 4.2.3. It was demonstrated, using two and three job problems, how completion time uncertainty propagates along the jobs. When jobs have the same mean processing time but different degrees of variability, the most uncertain jobs should be sequenced to be last. When variability is identical but mean processing times are different, it was demonstrated that the longest job should be scheduled first. When neither of these are identical, it was demonstrated that the optimal sequence would change depending on which of these factors was dominant.

CHAPTER FIVE: TWO - MACHINE FLOW SHOP PROBLEM ANALYSIS

The minimization of the mean weighted quadratic deviation of job completion times from a common deterministic due date in a two-machine flow shop environment is examined in this chapter. Jobs in this problem now consist of two tasks. Consistent with notations used for a single-machine problem, the deterministic class of these problems are denoted as [*n/m(=2)/MWQD/d*] and the stochastic as [*n/m(=2)/Ê*[*MWQD*]*/d*]. The addition of another machine, while considering the same performance measure as in a similar single-machine scenario [*n/m(=1)/MWQD/d*], is important for two reasons. First, it serves as a stepping stone to the understanding of more complex multi-machine problems. Second, a direct comparison of performances can be made with the single-machine problem.

The focus of the study is on the stochastic class of scheduling problem with idle times allowed between or before tasks on both machines. Commensurate with the technological constraints in the problem, some of the key characteristics of stochastic scheduling in a multi-machine scenario are identified and the effects of uncertainty of processing times are analyzed. For example, since there are multiple tasks per job but only one job due date, the scheduling of upstream tasks based on the due date performance measure introduces additional challenges.

5.1 Description of the problem

The flow shop environment considered is characterized by $n=2$ jobs being processed on $m=2$ machines. It is assumed that each job consists of elementary operations or tasks that need to be performed on both the machines, so that jobs have to visit both the machines for completion. Also, the second machine cannot begin operation unless a task has been completed on the first machine.

Consider that Job A consists of elementary tasks A1 on machine 1 and A2 on machine 2 and Job B consists of elementary tasks B1 on machine 1 and B2 on machine 2. The machines are considered to be arranged in a serial configuration with machine 1 ahead of machine 2. Figure 5.1 is an illustrative Gantt chart of an un-optimized two-machine problem.

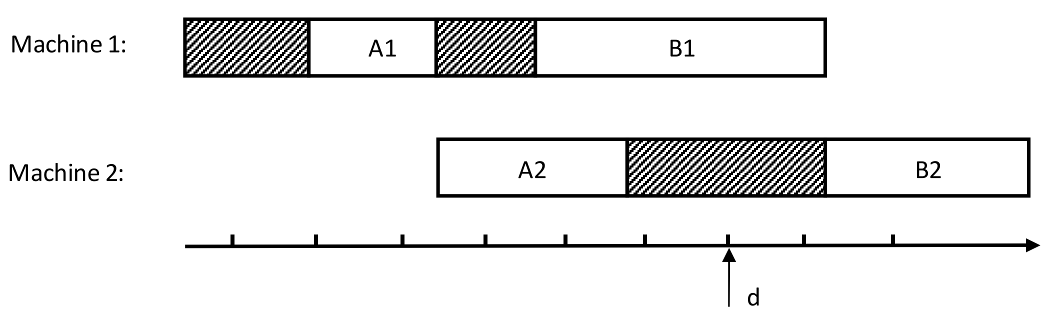


Figure 5.1 Gantt chart illustration of a two machine two jobs flow shop

Under the flowshop conditions described above, there are only $n!$ feasible sequences which need be considered to minimize the due date deviation, though there can be as many as $(n!)^2$ possible sequences if job shop is considered. This kind of sequencing for a multi-machine problem is called a permutation sequence.

This study therefore attempts to investigate the optimal scheduling for a permutation sequence of two jobs with a common due date and a two machine environment. Schedules are constructed using both deterministic and stochastic processing times and a static environment, with all the jobs available for processing at the start. No pre-emption is allowed. This means that once the task starts on a machine it cannot be interrupted until its completion on the machine. Idle times (I) between or before the tasks are allowed on both machines.

To keep the work-in-process to a minimum, it is advantageous to compress the schedule or to reduce the total flow time as much as possible. This necessitates that the individual tasks on each machine do not start too early. To achieve, an algorithm is built into the spreadsheet which pushes out the start times of each task during schedule optimization. This is achieved by considering the due date deviations of each of the tasks rather than considering the due date deviation for the jobs (i.e., last task) only. However, in the strictest sense the individual tasks do not have a due date (only the jobs have). The due date for the first task of each job i, (d_i*) is set up in the spreadsheet to be the completion time of the job minus the task time for the second task in each job. This is effectively the difference of the completion time and processing time of the job on the second machine, since each job completes on the second machine. With subscript 2 indicating the second machine, the due dates for the first task of each job can be expressed as Equation 5.1 for a deterministic problem.

$$d_i* = C_{i2} - P_{i2} \tag{5.1}$$

Equivalently, the j^{th} realization of this due date in a stochastic problem is given as Equation 5.2.

$$d_{ij}* = C_{ij2} - P_{ij2} \tag{5.2}$$

These task due dates, given by Equations 5.1 and 5.2, are included in the performance measure. Although the scheduling objective is to minimize the quadratic deviations of job completion times relative to a common due date *d* for the jobs, it is therefore necessary to modify the original objective functions (*MWQD* and $\hat{E}$ [*MWQD*]) in the spreadsheet in order to keep a "tight" schedule and minimize work-in-process (WIP) inventory. Therefore, the deviations of individual tasks on the first machine relative to the due dates d_i* are also taken into account. The modified objective functions are expressed as Equations 5.3 and 5.4 for the deterministic and stochastic problems respectively.

$$MWQD* \equiv \sqrt{\frac{\sum_{i=1}^{n}\left[\{\max(0, C_{i2} - d)\}^2 + \alpha\{\max(0, d - C_{i2})\}^2 + \{\max(0, C_{i1} - d_i*)\}^2 + \alpha\{\max(0, d_i* - C_{i1})\}^2\right]}{n}} \tag{5.3}$$

$$\hat{E}[MWQD]* \equiv \frac{\sum_{j=1}^{m}\left[\sqrt{\frac{\sum_{i=1}^{n}\left[\{\max(0, C_{ij2} - d)\}^2 + \alpha\{\max(0, d - C_{ij2})\}^2 + \{\max(0, C_{ij1} - d_{ij}*)\}^2 + \alpha\{\max(0, d_{ij}* - C_{ij1})\}^2\right]}{n}}\right]}{m} \tag{5.4}$$

In the above equations, C_{i1} and C_{i2} represent the deterministic completion time of the i^{th} job on machine 1 and 2 respectively. Equivalently, C_{ij1} and C_{ij2} represent the j^{th} realization of these completion times in a stochastic problem. The other notations are as defined in Chapter 3, Section 3.3.

Minimization of the functions represented by Equations 5.1 and 5.2 minimizes the total penalties of job due date deviations ($MWQD$ and $\hat{E}$ [$MWQD$]) along with maintaining a compressed schedule with minimum WIP. The $MWQD$ and $\hat{E}$ [$MWQD$] measures obtained in the optimal schedule should thus be understood as the best total weighted quadratic deviations of the jobs from a common due date that are possible by simultaneously having low WIP.

5.2 Analysis of the Problem

In this section the effect of processing time uncertainties in the two-machine flow shop problem is investigated. A common due date, d, is maintained across all experiments. The total expected work, equal to total mean processing time of the jobs, g, is also kept constant (i.e. $\mu_A + \mu_B = g$) across jobs in all experiments. This facilitates comparison among different experimental combinations. The study is performed under various settings of average machine loadings, ρ, is defined as the total of task times on each machine. The task times are the elementary operations particular to a job, performed on the particular machine. These are varied to consider different average machine loadings.

The job processing time uncertainties result from the individual task time uncertainties which make up each job.

5.2.1 Scheduling under identical task time variability

An initial problem is considered where the average load on machine 1 is lower than the average load on machine 2 (i.e. $\rho_1 < \rho_2$). The problem inputs are given in Table 5.1.

Table 5.1 Example inputs with unidentical average machine loadings ($\rho_1 < \rho_2$) and identical task time variability

Tasks	A1	B1	A2	B2
Mean Processing Time, μ	9.0	10.0	13.0	12.0
Std. Dev. of Processing Time, σ	0.5	0.5	0.5	0.5
d = 34, α =1				

The mean processing times μ and the standard deviations σ of the jobs A and B, are identical in this problem and are given as:

$$\mu_A = \mu_{A1} + \mu_{A2} = \mu_{B1} + \mu_{B2} = \mu_B = 22$$
$$\sigma_A = \sqrt{\sigma_{A1}^2 + \sigma_{A2}^2} = \sqrt{\sigma_{B1}^2 + \sigma_{B2}^2} = \sigma_B \approx 0.71$$

In the deterministic version of this problem, the quadratic penalty is at a minimum when the earliness and tardiness for the two jobs balance. Since both jobs complete on machine 2, LPT (longest processing time) sequencing on this machine will be optimal to minimize *MWQD*. This means the processing time of tasks on machine 2 will dictate the sequencing of tasks on both the machines. Figure 5.2 illustrates the result obtained using the Evolutionary Solver® for the deterministic problem.

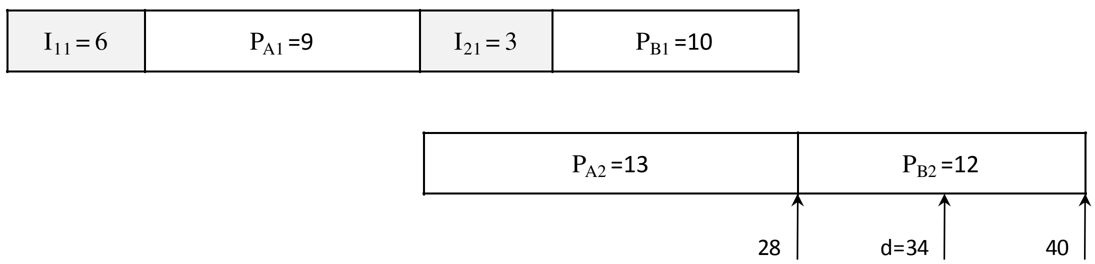


Figure 5.2 Deterministic scheduling example with unidentical average machine loading, $\rho_1 < \rho_2$

The tasks on machine 2 shift such that the Job A (task A2 completion) is 6 time units early and Job B (task B2 completion) is 6 time units tardy. The due date will be exactly at the centre of the mean processing time of the straddling job (task B2). This results in a minimum quadratic deviation, *MWQD*, of 6.

With deterministic processing times and sequence A-B, the completion time of B2, C_{B2}, can be expressed as Equation 5.5

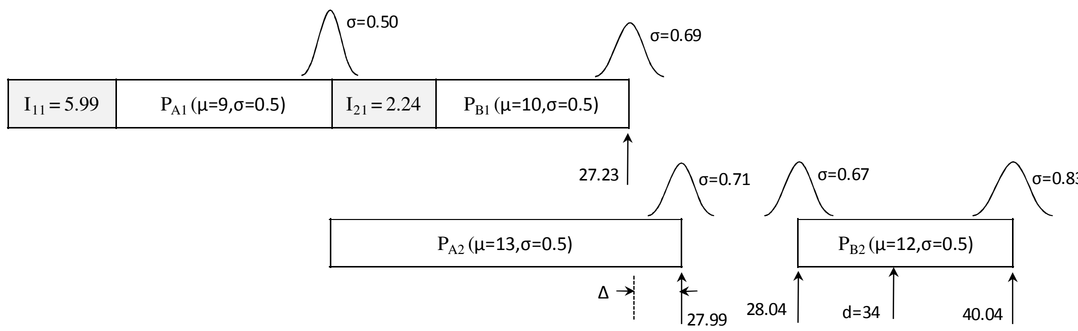
$$C_{B2} = I_{11} + P_{A1} + \max(I_{21} + P_{B1}, I_{12} + P_{A2}) + I_{22} + P_{B2} \tag{5.5}$$

A characteristic of the deterministic problem is that the last task on the first machine and the first task on the second machine should complete at the same time inorder to ensure mimimum penalty and simultaneous minimum work-in-process. This is true for all values of α. Therefore, considering Equation 5.3 and Figure 5.2, the following relation holds for a deterministic schedule.

$$I_{21} + P_{B1} = I_{12} + P_{A2} \tag{5.6}$$

When $I_{12} = 0$, the maximum optimal idle time of I_{21} is 3. Consequently $I_{11} = 6$ for this problem. These are the maximum idle times on the first machine, which ensure that the individual tasks start as late as possible without affecting the total earliness and tardiness on the jobs.

Now a stochastic version of the same problem is considered. If the variability of tasks completing on the second machine are equal ($\sigma_{A2} = \sigma_{B2} = 0.5$) the LPT sequence on this machine is still optimal, as in the deterministic case. However, the expected completion times for the last task on the first machine and the first task on the second machine are likely to be significantly different, unlike the deterministic case. A positive difference, Δ, between the average completion time of first task on machine 2 and second task on machine 1 is observed. This result, obtained using the Evolutionary Solver®, is illustrated in Figure 5.3.



(Average starting time for B2 = 28.0432 with a standard deviation of 0.67, $\hat{E}[MWQD]$= 6.06976, Δ = 0.76 and $\hat{E}\,[W]$=0.866)

Figure 5.3 Stochastic scheduling example with unidentical average machine loading, $\rho_1 < \rho_2$

When the problem is stochastic, the expected completion time for the last task on machine 2 (B2) is estimated as an average over m total realizations of the completion times given by Equation 5.7.

$$\hat{E}[C_{B2}] = \frac{1}{m}\sum_{j=1}^{m}(C_{B2})_j \tag{5.7}$$

where, $(C_{B2})_j$ is the j^{th} realization of these completion times, obtained using Equation 5.8.

$$(C_{B2})_j = I_{11} + (P_{A1})_j + \max\left(I_{21} + (P_{B1})_j, I_{12} + (P_{A2})_j\right) + I_{22} + (P_{B2})_j \tag{5.8}$$

Since the average completion time for the last task on the first machine $\hat{E}[C_{B1}]$ and the first task on second machine $\hat{E}[C_{A2}]$ are significantly different for a stochastic problem, the relationship in Equation 5.6 does not hold. The idle time determined on the basis of this equation would not be optimal for the stochastic problem.

It should be noted that any idle time I_{22} before the last task on the downstream machine causes deterioration of both the minimum flow-time, or WIP, and the minimum job deviations from the due date. This task should start immediately after either the completion of the second task on the upstream machine or the first task on the downstream machine, whichever is longer. In the stochastic problem, every realization of the starting time for the last task (B2) on machine 2, therefore can be obtained from Equation 5.9.

$$S_{B2} = \max(C_{B1}, C_{A2}) \tag{5.9}$$

Since this start time is the longer of the two completion times, this equation also suggests that the expected starting times (sample based estimates) $\hat{E}[S_{B2}]$ for second task on the downstream machine is always larger than either the expected completion times $\hat{E}[C_{B1}]$ for the last task on the upstream machine or the first task on the downstream machine. The last task on the downstream machine has to wait either for this machine to be free, or for the job to be available to work on. Further analysis of this waiting time follows.

Considering the A-B sequence of jobs, if job B is completed on machine 1 but machine 2 is not free, then this job (task B2) has a waiting time of $W_{B2\text{-}B1}$ relative to the completion time of B1. This is shown in case (a) of Figure 5.4. However, if machine 2 is free and job B is not completed on machine 1, then machine 2 is starved of a job. The machine waiting time relative to the completion time of A2, $W_{B2\text{-}A2}$, is shown as case (b) in Figure 5.4. The starting time of B2 is influenced by one of these two waiting times. The mathematical formulation of this expected wait time $\hat{E}[W]$ is now presented.

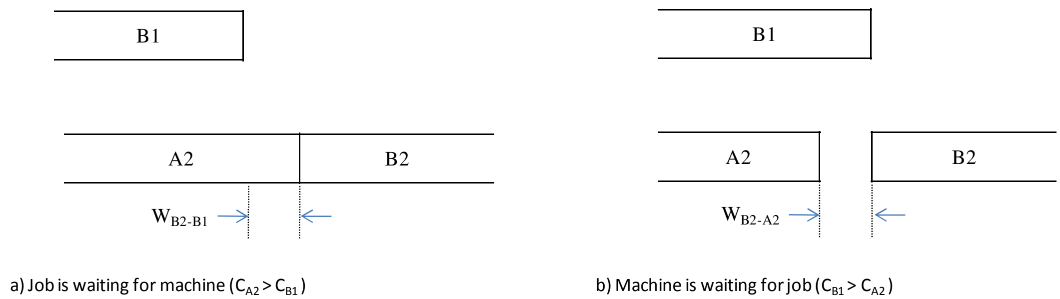


Figure 5.4 Illustration of job or machine waiting

In a stochastic problem, the waiting time, *W*, resulting from a job waiting for the machine or the machine waiting for a job is a random variable. Since processing times are random variables, the completion and start times are also random variables. The estimate of the expected value $\hat{E}$ [*W*] of this random variable can be obtained as a weighted average of the conditional expected value of *W*. This can be mathematically expressed by Equation 5.10.

$$\begin{aligned}\hat{E}[W] &= \hat{E}[W_{B2-B1}] + \hat{E}[W_{B2-A2}] \\ &= \hat{E}[W|C_{A2} > C_{B1}] \times P(C_{A2} > C_{B1}) + \hat{E}[W|C_{B1} > C_{A2}] \times P(C_{B1} > C_{A2})\end{aligned} \tag{5.10}$$

Each of the conditional expectations in Equation 5.10 are being weighted by the probability of the event on which it is conditioned (i.e. the proportion of time that condition (a) in Figure 5.4 occurs and the proportion of time that condition (b) occurs). All of the terms to the right of this equation can be easily calculated in the spreadsheet model for this problem. For example, referring to the example problem considered in

Figure 5.3, the following observations were made based on m = *1000* simulated problem instances.

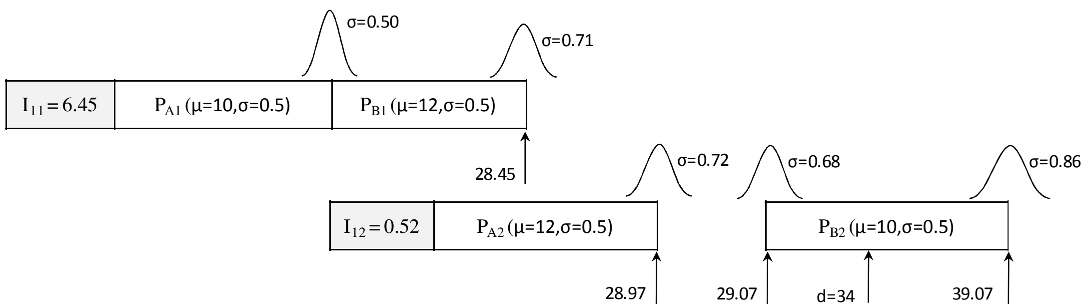
$$\begin{cases} P(C_{A2} > C_{B1}) = 0.853 \\ P(C_{B1} > C_{A2}) = 0.147 \\ \hat{E}[W | C_{A2} > C_{B1}] = 0.953 \\ \hat{E}[W | C_{B1} > C_{A2}] = 0.362 \end{cases}$$

Therefore, using Equation 5.10, the expected wait time is $\hat{E}\,[W] = \hat{E}\,[W_{B2\text{-}B1}] + \hat{E}\,[W_{B2\text{-}A2}]$ = 0.813+0.053 = 0.866.

Similar observations for Δ and $\hat{E}\,[W]$ were made when the mean task lengths on each machines were equal (i.e, when average machine loads were kept equal, $\rho_1 = \rho_2$). Table 5.2 gives the problem inputs. Results are shown in Figure 5.5 (not on scale).

Table 5.2 Example inputs with identical average machine loadings ($\rho_1 = \rho_2$) and identical task time variability

Tasks	A1	B1	A2	B2
Mean Processing Time, μ	10.0	12.0	12.0	10.0
Std. Dev. of Processing Times, σ	0.5	0.5	0.5	0.5
$d = 34$, $\alpha = 1$				



(Average starting time for B2=29.067 with a standard deviation of 0.68, $\hat{E}\,[MWQD] = 5.101$, $\Delta = 0.52$ and $\hat{E}\,[W] = 0.718$)

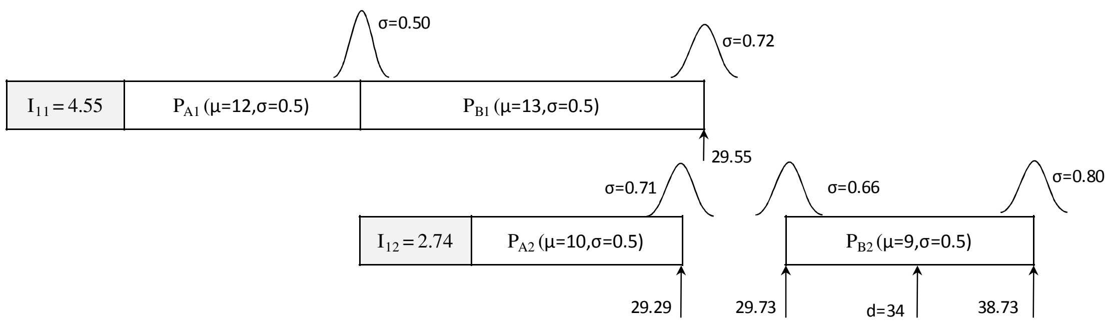
Figure 5.5 Stochastic scheduling example with identical average machine loadings, $\rho_1 = \rho_2$

The mean length of jobs on the second machine still governs the optimal sequence in this case but, when compared to Figure 5.3, a shift is noted in the way idle time is inserted between tasks.

However, when $\rho_1 > \rho_2$, Δ is observed to be negative (i.e. the average completion time of first task on the downstream machine is lower than the average completion time of the last task on the upstream machine). It is also noted that the idle time insertion is similar to the case with $\rho_1 = \rho_2$. The problem inputs with the results are given in Table 5.3 and Figure 5.6.

Table 5.3 Example inputs with unidentical average machine loadings ($\rho_1 > \rho_2$) and identical task time variability

	A1	B1	A2	B2
Mean Processing Time, μ	12.0	13.0	10.0	9.0
Std. Deviation of Processing Time, σ	0.5	0.5	0.5	0.5
$d = 34$, $\alpha = 1$				



(Average starting time for task B2=29.73 with a standard deviation of 0.66, $\hat{E}[MWQD]$ = 4.765, Δ = -0.26 and $\hat{E}\,[W]$ = 0.606)

Figure 5.6 Stochastic scheduling example with unidentical average machine loadings, $\rho_1 > \rho_2$

Another observation, given the identical variability in task times, is that the quadratic deviation measure improves if the upstream machine is loaded more heavily than the downstream machine. These observations are summarized in Table 5.4.

Table 5.4 Total quadratic penalty variation against machine loading

Relative machine loading	Ê[MWQD]
$\rho_1 < \rho_2$	6.070
$\rho_1 = \rho_2$	5.102
$\rho_1 > \rho_2$	4.765

The penalties are relative to job deviations from the due date and, since permutation flow shop is considered, each job completes on the downstream machine. With shorter tasks on second machine, the completion times are positioned closer to the due date than if longer tasks are loaded on this machine.

5.2.2 Scheduling under varying task time variability

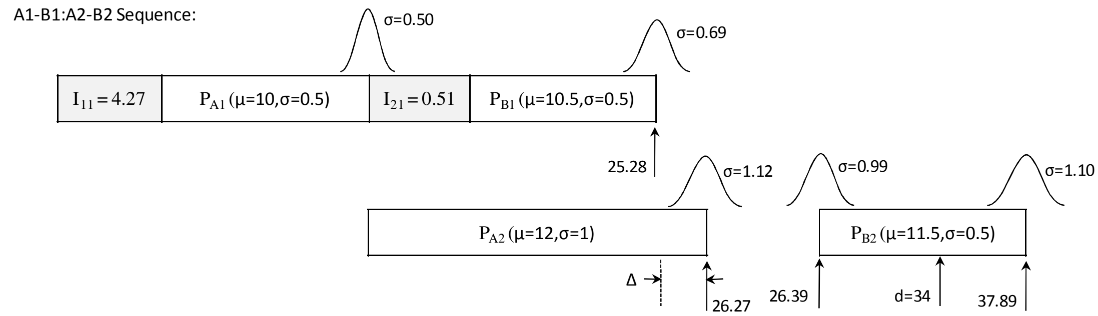
This section examines how processing time variability influences the schedule in a two-machine environment. The experimental analysis is similar to the single-machine problem. Different relative weightings on earliness and tardiness is used ($\alpha \neq 1$) to ensure that the idle times obtained are unique and the optimal sequence flip over with the increase of uncertainty, if any, could be well perceived.

Initially the job mean processing times are kept identical but the variability is not ($\mu_A = \mu_B, \sigma_A \neq \sigma_B$). This is done by progressively increasing the standard deviations for P_{A2}. An earliness-tardiness weighting ratio of $\alpha = 0.5$ and a common due date of $d = 34$ are used. Table 5.5 illustrates the problem inputs for an initial experiment.

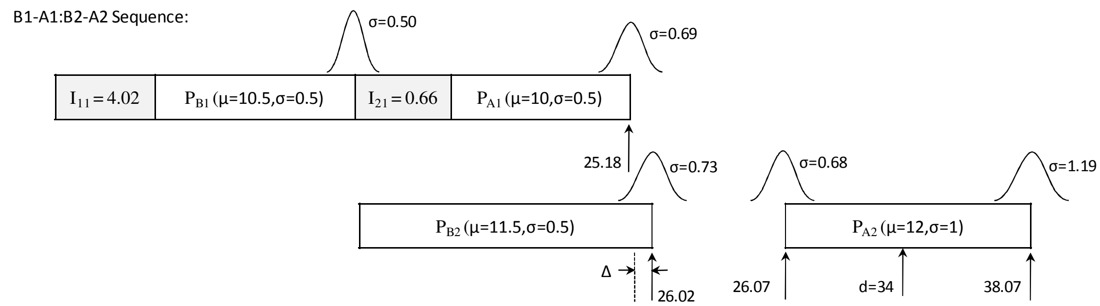
Table 5.5 Example input with different task time variability on second machine

	A1	B1	A2	B2
Mean Processing Time, μ	10.0	10.5	12.0	11.5
Std. Dev. of Processing Time, σ	0.5	0.5	1	0.5
$d = 34, \alpha = 0.5$				

The results using the two possible sequences for the problem in Table 5.5 are illustrated in Figure 5.7. These results are obtained considering only the idle times as decision variables and locking the sequence and running the Evolutionary Solver to optimize the idle times for a given sequence



(Average starting time for task B2=26.39 with a standard deviation of 0.99, $\hat{E}[MWQD] = 4.832$, $\Delta = 0.99$ and $\hat{E}\,[W] = 1.228$)



(Average starting time for task A2=26.07 with a standard deviation of 0.68, $\hat{E}[\mathrm{MWQD}] = 4.986$, $\Delta = 0.84$ and $\hat{E}\,[W] = 0.933$)

Figure 5.7 Stochastic scheduling example under different task time variability

Table 5.6 shows the optimal results when increasing the variability on P_{A2} using the two sequences. The results in Table 5.6 indicate that the sequence A1-B1:A2-B2 is optimal for standard deviations σ_{A2} lower than 2.5. At high values of σ_{A2}, sequence B1-A1:B2-A2 is optimal.

Table 5.6 Results using the two sequences with different task time variability on second machine

Seq. A1-B1:A2-B2				
σ_{A2}	Ê[MWQD]	I_{11}	I_{21}	Ê[W]
1	4.832	4.270	0.510	1.228
1.5	4.943	4.200	0.410	1.577
2	5.065	4.120	0.400	1.862
2.5	5.221	4.000	0.430	2.225
3	5.384	3.890	0.480	2.527
3.5	5.594	3.730	0.620	2.924

Seq. B1-A1:B2-A2				
σ_{A2}	Ê[MWQD]	I_{11}	I_{21}	Ê[W]
1	4.986	4.020	0.660	0.933
1.5	5.025	4.080	`0.660	0.933
2	5.095	4.150	0.700	0.904
2.5	5.162	4.220	0.670	0.926
3	5.254	4.300	0.660	0.933
3.5	5.357	4.400	0.680	0.919

These results are plotted for $\hat{E}[MWQD]$ against σ_{A2} in Figure 5.8. A switchover in the optimal sequence is obtained for a value of σ_{A2} between 2.1 and 2.2. This demonstrates that the variability of tasks on machine 2 can affect the optimal sequence for a two-machine flowshop.

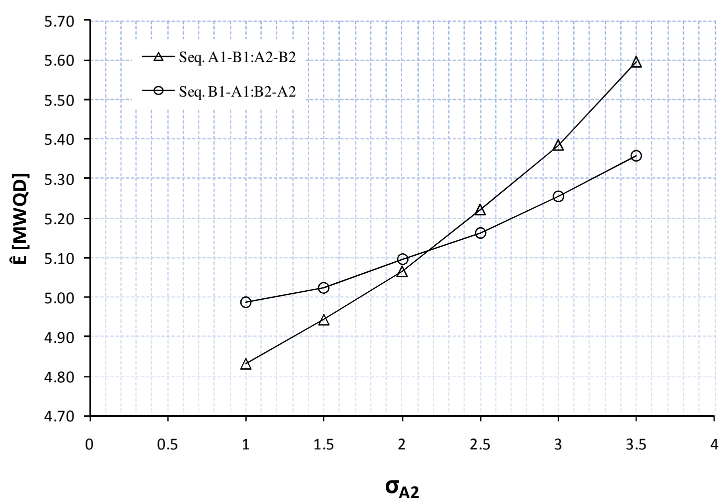


Figure 5.8 Illustration of optimal sequence switchover in a two machine problem against the task time variability on the second machine

The variability of A2 with sequence B-A on machine 2 does not affect the variability of any other tasks since this is the last task to complete. However if sequence A-B is used, variability on A2 propagates through B2 so that if the completion time of A2 is highly variable, the completion of B2 also becomes highly variable. This will cause deterioration of the due date performance. Therefore, with high variability on A2, the optimal sequence switches over to B-A.

However the variability of tasks on machine 1 can also cause the sequence to flip over. To demonstrate this, the problem inputs in Table 5.5 are again considered but now the variability is incremented progressively on task A1. The variability of A2 and all other tasks standard deviations are maintained at a standard deviation of 0.5. The results of increasing the variability on P_{A1} are shown in Table 5.7. Figure 5.9 shows $\hat{E}[MWQD]$ plotted against σ_{A1}.

Table 5.7 Results using the two sequences with different task time variability on first machine

Seq. A1-B1:A2-B2					
σ_{A1}	Ê[MWQD]	I_{11}	I_{21}	I_{12}	Ê[W]
1.0	4.820	4.320	0.700	-	0.887
1.5	4.902	4.320	0.700	-	0.887
2.0	5.021	4.310	0.700	-	0.887
2.5	5.169	4.300	0.700	-	0.887
3.0	5.334	4.240	0.710	-	0.879

Seq. B1-A1:B2-A2					
σ_{A1}	Ê[MWQD]	I_{11}	I_{21}	I_{12}	Ê[W]
1	4.984	3.960	0.370	-	1.318
1.5	5.015	3.920	0.060	-	1.773
2	5.060	3.750	-	0.110	2.129
2.5	5.119	3.560	-	0.230	2.494
3	5.176	3.360	-	0.370	2.858

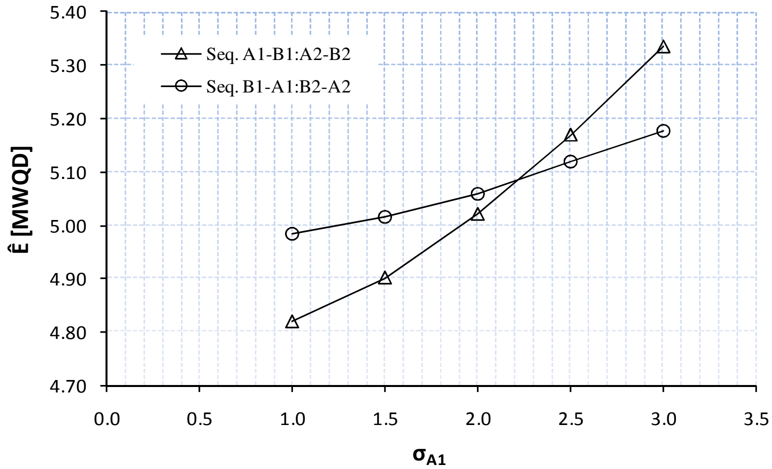


Figure 5.9 Illustration of optimal sequence switchover in a two machine problem against the task time variability on the first machine

Here again, if sequence B-A is used, variability on A1 propagates only through A2 because A2 starts after A1. However, if A-B is the sequence, variability on A1 propagates through all of the tasks B1, A2 and B2. This variability is undesirable when conformance to due dates is the objective. Hence, for high variability on A1, sequence B-A is preferred. With increasing variability on A1 and sequence B-A, it is also interesting to observe that a shift occurs in the inserted idle time pattern.

The results of this section clearly show that stochastic scheduling is required to guarantee optimal flow shop sequences. For a two-job, two-machine flow shop the optimal sequence is dependent on the variability of both the first and second tasks in the job. Deterministic scheduling approaches do not take variability into account and would therefore always result in the same sequence, regardless of the variability.

5.3 Summary

Analysis in this chapter revealed several features of a stochastic flow shop problem involving due date performance. A tight schedule, which reduces the total flow time of a job, is one where a job can begin on the downstream machine immediately after it is completed on the upstream machine. The role of idle times on the first machine is to push the task completion times as late possible without unduly affecting minimization of the total quadratic penalty, based on job due dates.

It was observed in the deterministic problem that it is always possible to complete the second task on the upstream machine at exactly at the same time as the first task on the second machine, ensuring minimum job flow times without due date penalty deterioration. In the stochastic problem these expected completion times are significantly different. The expected waiting time, whereby the job either waits for a downstream machine to be free or the downstream machine waits for the job to complete on the upstream machine, is an important characteristic in a stochastic problem. The expected waiting time pattern is a function of both the relative task lengths and variability. It is also interrelated with the optimal inserted idle time pattern. A shift in the idle time insertion pattern is also noticed when the relative loadings on the two machines are varied. This affects the job deviations from the due date and the resulting total quadratic penalty.

In Section 5.2.2 it has been demonstrated that the variability of tasks on either machine can affect the optimal sequence and the optimal idle times. It has been observed that highly variable jobs on either machine are preferably scheduled later. This is consistent with the analysis using single-machine problems.

CHAPTER SIX: CONCLUSION

This research has addressed issues involved in scheduling stochastic problems with a common due date and a non-regular performance measure, involving penalties for both early and late completion of jobs. Deterministic counterparts have been used for comparison in single-machine and two machine examples. The influence of variability on the optimal schedule, using normally distributed processing times, has been thoroughly investigated. Methodological issues in the prior research have been identified and a new approach of significant importance has been introduced and applied. This chapter discusses the practical and theoretical contributions of this research and suggests possible future extensions.

6.1 Contributions

This thesis contributes to a better understanding of stochastic scheduling. Using small, static problems, it has been demonstrated that scheduling on the basis of expected processing times may lead to an inappropriate choice of sequence and idle times if processing times are actually stochastic.

Since scheduling is a decision making problem, the models and insights developed can help a decision maker to predict performance more accurately and to select better schedules in practice. Efficient schedule construction is consistent with the Just-In-Time (JIT) philosophy. In facilities where processing time randomness is significant, scheduling in a deterministic manner may lead to a schedule which may be too 'optimistic' with respect to the total losses resulting from due date non-conformance. For example, in some process industries changes in raw materials can create variations of over 100% in the processing times (Taylor and Bolander, 1991).

This research has made several theoretical contributions to the scheduling literature. First, it has extended previous work in deterministic versus stochastic scheduling. Prior research shows that the deterministic and stochastic scheduling approaches yield identical sequences and total penalties if the objective is to minimize the total weighted flow time (Pinedo, 1995). However, Baker and Trietsch (2007) illustrated, using minimization of proportion tardy jobs (regular measure), that both the optimal sequence and penalty may be different for a deterministic versus stochastic problem. This research, by considering earliness as well as tardiness (non-regular measure) penalties, establishes that the two scheduling approaches may lead to different optimal sequences and forced idle times. As well, the objective function values will also differ.

Second, the effect of using linear versus quadratic penalty measures has been demonstrated. In particular, it has been demonstrated that using a quadratic measure will always yield a unique optimal schedule. This may not be true for a linear measure since multiple optimal schedules frequently result.

Third, this research has examined the problem of finding analytical solutions for stochastic problems. Scheduling problems are generally difficult due to the combinatorial nature of the problem. This difficulty is compounded if processing times are random. This research has proposed an analytical formulation for bounded processing time distributions, like the uniform distribution, and simple one or two job problems. Solution using Mathematica® has been demonstrated. The results obtained are important to establish the correctness of the experimental model as well as to ascertain that the spreadsheet optimization methodology developed is yielding optimal or near optimal results. Though the problem is less tractable using the normal distribution, for which no analytical solution could be obtained, use of the simpler uniform distribution facilitated verification.

Finally, a significant contribution of this research is in the extended application of a unique methodology to solve stochastic scheduling problems. The sample based optimization approach adopted by Baker and Trietsch (2007), which was based on the discrete probability distribution, has been extended for use with continuous distributions. Sample based simulation-optimization techniques like this have not been used elsewhere in static scheduling research.

In practice, the availability of generic computer codes to solve stochastic problems is limited. Spreadsheet based simulation-optimization tools, such as Excel® integrated with Evolutionary Solver®, are readily available. The relationships of interest are transparent, computations are quick, and outputs are easy to evaluate. All of these features help the scheduler explore different scenarios or perform sensitivity analysis efficiently. As well, different performance measures can readily be incorporated in the same spreadsheet model.

6.2 Further extensions

Several research extensions can be considered using the existing spreadsheet model. First, more work could be done using the same performance measures with larger problems. Some challenges with the reliability of finding an optimal solution exist but such work could be useful in identifying more general scheduling recommendations or policies. This requires only minor changes to the existing spreadsheet model and the practicality has already been demonstrated.

Second, more analysis could be done using a distinct instead of common due dates. This requires no further modification to the spreadsheet model. The main challenges relate to finding meaningful insights.

Third, the assumption of a static problem with all jobs initially available could be relaxed. For example, dynamic job arrivals could be considered. Dynamic job arrivals can be easily simulated using discrete-event simulation tools like Arena®. These simulation tools could then be linked into the existing spreadsheet model to re-optimize the schedule, perhaps with the arrival of each new job.

Fourth, other processing time distributions could be considered and results compared with those obtained using the normal distribution. This type of study would help determine how important the shape of the processing time distribution is. If it could be determined that considering only the mean and variance provides a near optimal solution, application of this stochastic scheduling approach would be more practical.

Finally, further research opportunities also exist in the analysis of multi-machine problems, such as flow shop problems. In this research a penalty was attached to the completion time for the first task in a job to prevent it from starting too early. In other words, compression of the schedule to prevent excessive work-in-process inventory was accomplished by creating task due dates as part of the scheduling algorithm. An alternative approach would be to base the schedule on an economic measure, comprised of holding and tardiness costs. Consideration of inventory holding costs would discourage any task from starting earlier than necessary.

BIBLIOGRAPHY

- Baker, K.R. and Scudder, G.D. (1990). Sequencing with earliness and tardiness penalties – a review. *Operations Research*, Vol. 38, No. 1, 22-36.
- Baker, K.R. and Trietsch, D. (2007). Safe scheduling, chapter 5 in tutorials in operations research (T.Klastorin, ed.). *Institute for Operations Research and Management Science* (*INFORMS*), 79-101.
- Birge, J., Frenk, J.B.G., Mittenthal, J. and Kan, A.H.G.R. (1990). Single machine scheduling subject to stochastic breakdowns. *Naval Research Logistics*, Vol. 37, No. 5, 661-677.
- Cai, X. and Tu, F.S. (1996). Scheduling jobs with random processing times on a single machine subject to stochastic breakdowns to minimize early tardy penalties. *Naval Research Logistics*, Vol. 43, No. 8, 1127-1146.
- Cai, X. and Zhou, S. (1997). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics*, Vol. 44, No. 6, 531-557.
- Cai, X., Lum, V.Y.S., and Chan, J.M.T. (1997). Scheduling about a common due date with job dependent asymmetric earliness and tardiness penalties. *European Journal of Operational Research*, Vol. 98, No.1, 154-168.
- De. P., Ghosh, J.B. and Wells, C.E. (1991). Scheduling to minimize weighted earliness and tardiness about a common due date. *Computers and Operations Research*, Vol. 18, No. 5, 465-475.
- Gupta, S.K. and Kyparisis, J. (1987). Single machine scheduling research. *OMEGA-International Journal of Management Science*, Vol. 15, No. 3, 207-227.

- Hino, C.M., Ronconi, D.P. and Mendes, A.B. (2005). Minimizing earliness and tardiness penalties in a single machine problem with a common due date. *European Journal of Operational Research*, Vol. 160, No.1, 190-201.
- Hussain, S.A. and Sastry, V.U.K. (1999). Application of genetic algorithm to stochastic single machine scheduling problem with earliness and tardiness costs. *International Journal of Computer Mathematics*, Vol. 70, No. 3, 383-391.
- Jackson, J.R. (1956). An extension of Johnson's results on job lot scheduling. *Naval Research Logistics Quarterly*, Vol. 3, No. 1-2, 201-203.
- Johnson, S.M. (1954). Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, Vol. 1, No.1-2, 61-67.
- Lauff, V. and Werner, F. (2004). Scheduling with common due date, earliness and tardiness penalties for multimachine problems: a survey. *Mathematical and Computer Modelling*, Vol. 40, No. 5-6, 637-655.
- Mondal, S.A. and Sen, A.K. (2001). Single machine weighted earliness tardiness penalty problem with a common due date. *Computers and Operations Research*, Vol. 28, No. 7, 649-669.
- Mondal, S.A. (2002). Minimization of squared deviation of completion times about a common due date. *Computers and Operations Research*, Vol. 29, No. 14, 2073-2085.
- Moslehi, G., Mirzaee, M., Vasei, M., Modarres, M. and Azaron, A. (2009). Two machine flow shop scheduling to minimize the sum of maximum earliness and tardiness. *International Journal of Production Economics*, Vol. 122, No. 2, 763-773.
- Pinedo, M. (1995). *Scheduling theory, algorithms and systems*. New Jersey: Prentice Hall
- Ramasesh, R. (1990). Dynamic job shop scheduling: a survey of simulation research. *OMEGA- International Journal of Management Science,* Vol. 18, No.1, 43-57.

- Sakurba, C.S., Ronconi, D.P. and Sourd, F. (2009). Scheduling in a two machine flowshop for the minimization of the mean absolute deviation from a common due date. *Computers and Operations Research*, Vol. 36, No. 1, 60-72.
- Sarin, S.C., Erel, E. and Steiner, G. (1991). Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research*, Vol. 51, No.2, 188-198.
- Sarper, H. (1995). Minimizing the sum of absolute deviations about a common due date for the two machine flow shop problem. *Applied Mathematical Modelling*, Vol. 19, No. 3, 153-161.
- Seo, D.K., Klein, C.A. and Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers and Industrial Engineering*, Vol. 48, No. 2, 153-161.
- Silver, E.A., Pyke, D.F. and Peterson, R. (1998). *Inventory Management and Production Planning and Scheduling* (3rd edition). New York: John Wiley.
- Smith, W.E., (1956). Various optimizers for single stage production. *Naval Research Logistics Quarterly*, Vol. 3, No.1-2, 59-66.
- Soroush, H.M. and Fredendall, L.D. (1994). The stochastic single machine scheduling problem with earliness and tardiness costs. *European Journal of Operational Research*, Vol. 88, No.2, 328-335.
- Soroush, H.M. (1999). Sequencing and due date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operational Research*, Vol. 113, No.2, 450-468.
- Soroush, H.M. and Allahverdi, A. (2005). Stochastic two machine flowshop scheduling problem with total completion time criterion. *International Journal of Industrial Engineering – Theory, Applications and Practice*, Vol. 12, No. 2, 159-171.

- Sung, C.S. and Min, J.I. (2001). Scheduling in a two machine flowshop with batch processing machines for earliness / tardiness measure under a common due date. *European Journal of Operational Research*, Vol. 131, No.1, 95-106.
- Tang, H.Y., Zhao, C.L. and Cheng, C.D. (2008). Single machine stochastic JIT scheduling problem subject to machine breakdowns. *Science in China Series A – Mathematics*, Vol. 51, No. 2, 273-292.
- Taylor, S.G. and Bolander, S.F. (1991). Process flow scheduling workshop report. In *APICS 34th International Conference Proceedings – The Integrated Management Revolution: A Global Competitive Challenge.* Book series: Library of American Production, 560-562.
- Vollmann, T.E., Berry, W.L. and Jacobs, F.R. (2005). *Manufacturing Planning and Control for Supply Chain Management* (International edition). New York: McGraw-Hill.
- Weng, X.H. and Ventura, J.A. (1996). Scheduling about a given common due date to minimize mean squared deviation of completion times. *European Journal of Operational Research*, Vol. 88, No.2, 328-335.
- Xia, Y., Chen, B., and Yue, J. (2005). Sequence jobs and assign due dates with uncertain processing times and quadratic penalty functions. In proceedings of the *1st International Conference on Algorithmic Applications in Management*, Xian, PR China, eds. Megiddo, N., Xu, Y. and Zhu, B.
- Yang, J. (2009). Two machine flow shop scheduling problem with weighted WIP costs. *Computers and Operations Research*, Vol. 36, No. 2, 472 - 486.

APPENDIX A: MATHEMATICA CODE

```
(*Mathematica code using uniform distribution for 2 jobs seq. problem--*)
(*----Linear penalty case------*)

(*---------------Initialization------------------------------------*)

Remove["Global`*"];
d=.;a1=.;a2=.;b1=.;b2=.;
currentmin=99999999;
currentargmin={-999999,-999999};
sequence=0;

(*---------------Example problem inputs ---------------------------*)


d=40;a1=19;a2=19.5;b1=21;b2=20.5;α=0.5;

(*---------------Programming---------------------------------------*)

Do[        (* do twice, once in each possible sequence *)
  If[i==2,a1hold=a1;
   a1=a2;a2=a1hold;
   b1hold=b1;
   b1=b2;b2=b1hold];

(*----Defining the earliness and tardiness scenario----*)

  (* job 1 is early, job 2 is early *)
  EE[p1_,p2_]:=(α*(d-p1-I1)+α*(d-p1-p2-I1-I2))/2;

  (* job 1 is early, job 2 is tardy *)
  ET[p1_,p2_]:=(α*(d-p1-I1)+(p1+p2+I1+I2-d))/2;

   (* job 1 is tardy, job 2 is tardy *)
  TT[p1_,p2_]:=((p1+I1-d)+(p1+p2+I1+I2-d))/2;

(*---------------------------------------------------------------*)

(* Defining where the four endpoints are relative to the diagonal and
vertical lines for given I1 and I2 *)

  below[x_,y_]:=0≤I1+I2&&I1+I2<d-x-y;
  above[x_,y_]:=Not[below[x,y]];
  left[x_,y_]:=0≤I1&&I1<d-x;
  right[x_,y_]:=Not[left[x,y]];

(*---------------------------------------------------------------*)




(* Finding  all ranges for I1 and I2 *)


rangei=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],below[b1,a2],below[a1,b2],below[b1,b2]},{I1,I2}];
```

```
rangeii=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],below[b1,a2],below[a1,b2],above[b1,b2],left[b1,b2]},{
I1,I2}];

rangeiii=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],above[b1,a2],below[a1,b2],above[b1,b2],left[b1,a2],le
ft[b1,b2]},{I1,I2}];

rangeiv=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],below[b1,a2],above[a1,b2],above[b1,b2],left[a1,b2],le
ft[b1,b2]},{I1,I2}];

rangev=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],above[b1,a2],above[a1,b2],above[b1,b2],left[b1,a2],le
ft[a1,b2],left[b1,b2]},{I1,I2}];

  rangevi=Reduce[{d≥ I1≥0,d≥
I2≥0,above[a1,a2],above[b1,a2],above[a1,b2],above[b1,b2],left[a1,a2],le
ft[b1,a2],left[a1,b2],left[b1,b2]},{I1,I2}];

  rangevii=Reduce[{d≥ I1≥0,d≥
I2≥0,above[a1,a2],above[b1,a2],above[a1,b2],above[b1,b2],left[a1,a2],ri
ght[b1,a2],left[a1,b2],right[b1,b2]},{I1,I2}];

  rangeviii=Reduce[{d≥ I1≥0,d≥
I2≥0,above[a1,a2],above[b1,a2],above[a1,b2],above[b1,b2],right[a1,a2],r
ight[b1,a2],right[a1,b2],right[b1,b2]},{I1,I2}];

  rangeix=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],above[b1,a2],above[a1,b2],above[b1,b2],left[a1,a2],ri
ght[b1,a2],left[a1,b2],right[b1,b2]},{I1,I2}];

  rangex=Reduce[{d≥ I1≥0,d≥
I2≥0,below[a1,a2],above[b1,a2],below[a1,b2],above[b1,b2],left[a1,a2],ri
ght[b1,a2],left[a1,b2],right[b1,b2]},{I1,I2}];

(*----------------------------------------------------------------*)
(*Define all cases i through x*)
(*Minimize objective function*)
(*Output the current minimum results and the sequence*)

(*----------------------------------------------------------------*)

(* Case i *)

  I1=.;I2=.;
  If[TrueQ[Boole[rangei]==0],Print["no case i"],
   Print["Case i expected penalty:  "];
   EPCasei=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,b1},{p2,a2,b2}];
   Print[EPCasei];
   newmin=MinValue[{EPCasei,{rangei}},{I1,I2}];
   newargmin=ArgMin[{EPCasei,{rangei}},{I1,I2}];
   Print["Minimum value in case i is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];
```

```
(*----------------------------------------------------------------*)

 (* Case ii *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangeii]==0],Print["no case ii"],
   Print["Case ii expected penalty:  "];
   EPCaseii=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,(d-I1-
I2)-b2},{p2,a2,b2}]+Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-
I1-I2)-b2,b1},{p2,a2,(d-I1-I2)-p1}]+Integrate[ET[p1,p2]*(1/(b1-
a1))*(1/(b2-a2)),{p1,(d-I1-I2)-b2,b1},{p2,(d-I1-I2)-p1,b2}];
   Print[EPCaseii];
   newmin=MinValue[{EPCaseii,{rangeii}},{I1,I2}];
   newargmin=ArgMin[{EPCaseii,{rangeii}},{I1,I2}];
   Print["Minimum value in case ii is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*----------------------------------------------------------------*)

(* Case iii *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangeiii]==0],Print["no case iii"],
   Print["Case iii expected penalty:  "];
   EPCaseiii=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,(d-I1-
I2)-b2},{p2,a2,b2}]+Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-
I1-I2)-b2,(d-I1-I2)-a2},{p2,a2,(d-I1-I2)-
p1}]+Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-I1-I2)-b2,(d-I1-
I2)-a2},{p2,(d-I1-I2)-p1,b2}]+Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,(d-I1-I2)-a2,b1},{p2,a2,b2}];
   Print[EPCaseiii];
   newmin=MinValue[{EPCaseiii,{rangeiii}},{I1,I2}];
   newargmin=ArgMin[{EPCaseiii,{rangeiii}},{I1,I2}];
   Print["Minimum value in case iii is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*---------------------------------------------------------------*)

 (* Case iv *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangeiv]==0],Print["no case iv"],
   Print["Case iv expected penalty:  "];
   EPCaseiv=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,b1},{p2,a2,(d-I1-I2)-p1}]+Integrate[ET[p1,p2]*(1/(b1-
a1))*(1/(b2-a2)),{p1,a1,b1},{p2,(d-I1-I2)-p1,b2}];
   Print[EPCaseiv];
   newmin=MinValue[{EPCaseiv,{rangeiv}},{I1,I2}];
   newargmin=ArgMin[{EPCaseiv,{rangeiv}},{I1,I2}];
   Print["Minimum value in case iv is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
```

```
",currentargmin," from sequence ",sequence]];

(*------------------------------------------------------------------*)

(* Case v *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangev]==0],Print["no case v"],
   Print["Case v expected penalty:  "];
   EPCasev=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,(d-I1-I2)-
a2},{p2,a2,(d-I1-I2)-p1}]+Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,(d-I1-I2)-a2},{p2,(d-I1-I2)-
p1,b2}]+Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-I1-I2)-
a2,b1},{p2,a2,b2}];
   Print[EPCasev];
   newmin=MinValue[{EPCasev,{rangev}},{I1,I2}];
   newargmin=ArgMin[{EPCasev,{rangev}},{I1,I2}];
   Print["Minimum value in case v is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*-----------------------------------------------------------------*)

(* Case vi *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangevi]==0],Print["no case vi"],
   Print["Case vi expected penalty:  "];
   EPCasevi=Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,b1},{p2,a2,b2}];
   Print[EPCasevi];
   newmin=MinValue[{EPCasevi,{rangevi}},{I1,I2}];
   newargmin=ArgMin[{EPCasevi,{rangevi}},{I1,I2}];
   Print["Minimum value in case vi is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*-----------------------------------------------------------------*)

(* Case vii *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangevii]==0],Print["no case vii"],
   Print["Case vii expected penalty:  "];
   EPCasevii=Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,d-
I1},{p2,a2,b2}]+Integrate[TT[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,d-
I1,b1},{p2,a2,b2}];
   Print[EPCasevii];
   newmin=MinValue[{EPCasevii,{rangevii}},{I1,I2}];
   newargmin=ArgMin[{EPCasevii,{rangevii}},{I1,I2}];
   Print["Minimum value in case vii is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];
```

```
 (*-------------------------------------------------------------*)

 (* Case viii *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangevii]==0],Print["no case viii"],
   Print["Case viii expected penalty:  "];
   EPCaseviii=Integrate[TT[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,b1},{p2,a2,b2}];
   Print[EPCaseviii];
   newmin=MinValue[{EPCaseviii,{rangeviii}},{I1,I2}];
   newargmin=ArgMin[{EPCaseviii,{rangeviii}},{I1,I2}];
   Print["Minimum value in case viii is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*-------------------------------------------------------------*)


(* Case ix *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangeix]==0],Print["no case ix"],
   Print["Case ix expected penalty:  "];
   EPCaseix=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,(d-I1-
I2)-a2},{p2,a2,(d-I1-I2)-p1}]+Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-
a2)),{p1,a1,(d-I1-I2)-a2},{p2,(d-I1-I2)-p1,b2}]+
     Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-I1-I2)-a2,d-
I1},{p2,a2,b2}]+
     Integrate[TT[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,d-
I1,b1},{p2,a2,b2}];
   Print[EPCaseix];
   newmin=MinValue[{EPCaseix,{rangeix}},{I1,I2}];
   newargmin=ArgMin[{EPCaseix,{rangeix}},{I1,I2}];
   Print["Minimum value in case ix is ",N[newmin],"  occurs
at=",newargmin];

If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*-------------------------------------------------------------*)
 (* Case x *)
  I1=.;I2=.;
  If[TrueQ[Boole[rangex]==0],Print["no case x"],
   Print["Case x expected penalty:  "];
   EPCasex=Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,a1,(d-I1-I2)-
b2},{p2,a2,b2}]+Integrate[EE[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-I1-
I2)-b2,(d-I1-I2)-a2},{p2,a2,(d-I1-I2)-p1}]+Integrate[ET[p1,p2]*(1/(b1-
a1))*(1/(b2-a2)),{p1,(d-I1-I2)-b2,(d-I1-I2)-a2},{p2,(d-I1-I2)-p1,b2}]+
Integrate[ET[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,(d-I1-I2)-a2,d-
I1},{p2,a2,b2}]+
Integrate[TT[p1,p2]*(1/(b1-a1))*(1/(b2-a2)),{p1,d-I1,b1},{p2,a2,b2}];
Print[EPCasex];
newmin=MinValue[{EPCasex,{rangex}},{I1,I2}];
newargmin=ArgMin[{EPCasex,{rangex}},{I1,I2}];
Print["Minimum value in case x is ",N[newmin],"  occurs at=",newargmin];
```

```
If[newmin<currentmin,currentmin=newmin;currentargmin=newargmin;sequence=
i];
   Print["Current min and argmin=",currentmin,"  ",N[currentmin],"
",currentargmin," from sequence ",sequence]];

(*-----------------------------------------------------------------*)
(*--Print final optimal idle times and total penalty and the resulting
sequence--*)


Print[" "],{i,1,2}];
Print["Final optimal {I1,I2} and penalty in rational and decimal form=
",currentargmin,"  ",N[currentargmin],"       ",currentmin,"
",N[currentmin]];

(*-----------------------------------------------------------------*)
```

APPENDIX B: SNAPSHOT OF A TWO-MACHINE SPREADSHEET MODEL

	Model Inputs	M1 TaskA1	M1 TaskB1	M2 TaskA2	M2 TaskB2
9	Job Type	1	2	3	4
10	Mean Proc Time	9.00	10.00	13.0	12.0
11	Std. Dev. Proc. Time	0.50	0.50	0.5	0.5
12	Arrival Time	0.00	0.00	0.00	0.00
13	Due Date Time			34	34

Job A: M1TaskA1 to M2TaskA2
Job B: M1TaskB1 to M2TaskB2

	E/T Weighting ratio (α)	Current Time	No. of Jobs
20	1	0.00	2
21			
22			

Decision Variables

21	Opt. Idle Time, *Ii*	5.99	2.240	0.000	0.000
22	Opt Seq	1	2	3	4

24	Job Types	1	2	3	4
25	Due date time, *d*			34.00	34.00
26	Processing time, *Pj*	9.00	10.00	13.0	12.0

Model Outputs

Deterministic Case:

						Completion Time				MVLD	MVQD
25	Time to due date	0.00	0.00	34.00	34.00						
26	Processing Time	9.00	10.00	13.00	12.00	**14.99**	**27.23**	**27.99**	**39.99**	**6.38**	**6.03**

Considering job deviations only

Generate Proc Times

Random Processing Time Generation:

	Sequence in Queue	1	2	3	4
35	2	8.69	10.72	13.8502	12.1997
36	3	8.07	10.37	13.193	12.2649
1033	1000	9.01	10.68	13.227	11.4479
1034	1001	9.35	9.66	12.8576	11.6505

Stochastic Case:

Expected values of the measures based on realized samples

						Completion Time				M2T2 Start Time	E[W/CA2>CB1]	E[W/CB1>CA2]	Ê[MVLD]	Ê[MVQD]	Ê[MVQD]
34	Opt. Seq.	1	2	3	4	**14.99**	**27.23**	**27.99**	**40.04**	**28.044**	**0.95**	**0.38**	**6.43**	**6.11**	**6.07**
35		8.69	10.72	13.85	12.20	14.68	27.64	28.53	40.73	28.53	0.89	FALSE	6.54	6.16	6.13
36	Realized Processing	8.07	10.37	13.19	12.26	14.06	26.67	27.25	39.52	27.25	0.58	FALSE	6.42	6.18	6.16
1033	Times, *Pij* reshuffled	9.01	10.68	13.23	11.45	15.00	27.92	28.22	39.67	28.22	0.31	FALSE	5.88	5.73	5.72
1034		9.35	9.66	12.86	11.65	15.34	27.23	28.19	39.84	28.19	0.96	FALSE	6.30	5.86	5.83

(Note: Additional columns of calculations relevant to two machine problem are included but some of the columns using calculations that were similar to the single machine problem have been hidden here, Jobs are designated as A and B. M1T1 represents machine 1 task 1 and accordingly for others. CA1 represents completion time of job A on machine 1 and accordingly for others)

APPENDIX C: FORMULA PRINT OF A SINGLE MACHINE MODEL

	A	B	C	D	E	F	G	H	I
8		**Model Inputs**							
9		Job Type	1	2					
10		Mean Proc Time	9	10					
11		Std. Dev. Proc. Tir	1.5	0.5					
12		Arrival Time	0	0					
13		Due Date Time	=(C10+D10)*C22	=C13					
17									
18			Current	No. of					
19		E/T Weighting rati	Time	Jobs					
20		0.5	0	2				**Decision Variables**	
21		Due date tightness					Opt. Idle Time,*Ii*	9	0
22		*b* =	=25/19				Opt Seq	2	1
23									
24		Job Types	=C9	=D9		**Determinis**			
25		Due date time, *d*	=C13	=D13			Time to due date	=+HLOOKUP(H$22,$C$24:$D$26,2)-$C$20	=+HLOOKUP(I$22,$C$24:$D$26,2)-$C$20
26		Processing time, *Pi*	=C10	=D10			Processing Time	=HLOOKUP(H$22,$C$24:$D$26,3)	=+HLOOKUP(I$22,$C$24:$D$26,3)
27									
28									
29									
30			Generate Proc Times			**Stochastic**			
31									
32									
33		**Random Processi**							
34		Sequence in Queue	=C9	=D9			Opt. Seq.	=H22	=I22
35		2	9.72101139778992	9.77152777647624				=+HLOOKUP(H$22,$C$34:$D$1034,$B35)	=+HLOOKUP(I$22,$C$34:$D$1034,$B35)
36		=+B35+1	7.18705237632327	9.23888700395626			Realized Processing	=+HLOOKUP(H$22,$C$34:$D$1034,$B36)	=+HLOOKUP(I$22,$C$34:$D$1034,$B36)
1033		=+B1032+1	7.18966713628247	10.4572548869891			Times,*Pij* reshuffled	=+HLOOKUP(H$22,$C$34:$D$1034,$B1033	=+HLOOKUP(I$22,$C$34:$D$1034,$B1033)
1034		=+B1033+1	8.54904754769593	10.160792837902				=+HLOOKUP(H$22,$C$34:$D$1034,$B1034	=+HLOOKUP(I$22,$C$34:$D$1034,$B1034)
1035									
1036									

	J	K	L	M	N	O	P
21							
22							
23							
24							Prop.
25		Completion Time			No. Tardy		Tardy
26		**=H26+H21**	**=K26+I21+I26**		**=IF(+H26+H$21-H$25>0,1,0)**	**=IF(H21+H26+I26+I$21-I$25>0,1,0)**	**=SUM(N26:O26)/$D$20**
27							
28							
29							
30							
31							
32							Prop.
33		Completion Time			No. Tardy		Tardy
34		**=AVERAGE(K35:K1034)**	**=AVERAGE(L35:L10**		**=+AVERAGE(N35:N1034)**	**=+AVERAGE(O35:O1034)**	**=+AVERAGE(P35:P1034)**
35		=H35+H$21	=K35+$I$21+I35		=IF(+H35+$H$21-$H$25>0,1,0)	=IF(H35+$H$21+I35+$I$21-$I$25>0,1,0)	=SUM(N35:O35)/$D$20
36		=H36+H$21	=K36+$I$21+I36		=IF(+H36+$H$21-$H$25>0,1,0)	=IF(H36+$H$21+I36+$I$21-$I$25>0,1,0)	=SUM(N36:O36)/$D$20
1033		=H1033+H$21	=K1033+$I$21+I1033		=IF(+H1033+$H$21-$H$25>0,1,0)	=IF(H1033+$H$21+I1033+$I$21-$I$25>0,1,0)	=SUM(N1033:O1033)/$D$20
1034		=H1034+H$21	=K1034+$I$21+I1034		=IF(+H1034+$H$21-$H$25>0,1,0)	=IF(H1034+$H$21+I1034+$I$21-$I$25>0,1,0)	=SUM(N1034:O1034)/$D$20
1035							
1036							
1037							

	R	S	T	V	W	X
24			Mean			Mean
25	Tardiness		Tardiness	Earliness		Earliness
26	**=+MAX(0,K26-H$25)**	**=+MAX(0,L26-I$25)**	**=SUM(R26:S26)/$D$20**	**=+MAX(0,-(K26-H$25))**	**=+MAX(0,-(L26-I$25))**	**=SUM(V26:W26)/$D$20**
27						
28						
29						
30						
31						
32			Mean			Mean
33	Tardiness		Tardiness			Earliness
34	**=+AVERAGE(R35:R1034**	**=+AVERAGE(S35:S1034**	**=+AVERAGE(T35:T1034)**	**=+AVERAGE(V35:V1034)**	**=+AVERAGE(W35:W1034)**	**=+AVERAGE(X35:X1034)**
35	=+MAX(0,K35-H$25)	=+MAX(0,L35-I$25)	=SUM(R35:S35)/$D$20	=+MAX(0,-(K35-H$25))	=+MAX(0,-(L35-I$25))	=SUM(V35:W35)/$D$20
36	=+MAX(0,K36-H$25)	=+MAX(0,L36-I$25)	=SUM(R36:S36)/$D$20	=+MAX(0,-(K36-H$25))	=+MAX(0,-(L36-I$25))	=SUM(V36:W36)/$D$20
1033	=+MAX(0,K1033-H$25)	=+MAX(0,L1033-I$25)	=SUM(R1033:S1033)/$D$20	=+MAX(0,-(K1033-H$25))	=+MAX(0,-(L1033-I$25))	=SUM(V1033:W1033)/$D$20
1034	=+MAX(0,K1034-H$25)	=+MAX(0,L1034-I$25)	=SUM(R1034:S1034)/$D$20	=+MAX(0,-(K1034-H$25))	=+MAX(0,-(L1034-I$25))	=SUM(V1034:W1034)/$D$20
1035						

	Z	AA
24		
25	**MWLD**	**MWQD**
26	**=T26+$B$20*X26**	**=SQRT((R26^2+$B$20*V26^2+S26^2+$B$20*W26^2)/$D$20)**
27		
28		
29		
30		
31		
32		
33	**$\hat{E}$[MWLD]**	**$\hat{E}$[MWQD]**
34	**=+AVERAGE(Z35:Z1034)**	**=+AVERAGE(AA35:AA1034)**
35	=T35+$B$20*X35	=SQRT((R35^2+$B$20*V35^2+S35^2+$B$20*W35^2)/$D$20)
36	=T36+$B$20*X36	=SQRT((R36^2+$B$20*V36^2+S36^2+$B$20*W36^2)/$D$20)
1033	=T1033+$B$20*X1033	=SQRT((R1033^2+$B$20*V1033^2+S1033^2+$B$20*W1033^2)/$D$20)
1034	=T1034+$B$20*X1034	=SQRT((R1034^2+$B$20*V1034^2+S1034^2+$B$20*W1034^2)/$D$20)
1035		