COMMON DUE DATE EARLY/TARDY SCHEDULING ON A SINGLE MACHINE WITH DETERIORATING JOBS AND DETERIORATING MAINTENANCE

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Fatma Şirvan

July, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

____

Prof. Ülkü Gürler (Advisor)

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

____

Assoc. Prof. Mehmet Rüştü Taner (Co-Advisor)

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

____

Assoc. Prof. Dr. Sinan Gürel

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

____

Asst. Prof. Dr. Emre Nadar

Approved for the Graduate School of Engineering and Science

____

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

COMMON DUE DATE EARLY/TARDY SCHEDULING ON A SINGLE MACHINE WITH DETERIORATING JOBS AND DETERIORATING MAINTENANCE

Fatma Şirvan
M.S. in Industrial Engineering
Supervisors: Prof. Ülkü Gürler, Assoc. Prof. Mehmet Rüştü Taner
July, 2013

This study considers a scheduling problem with position-dependent deteriorating jobs and a maintenance activity in a single machine. Even in the absence of maintenance activity and deterioration problem is NP-hard. A solution comprises the following: (i) positions of jobs, (ii) the position of the maintenance activity, (iii) starting time of the first job in the schedule. After the maintenance activity, machine will revert to its initial condition and deterioration will start anew. The objective is to minimize the total weighted earliness and tardiness costs. Jobs scheduled before (after) the due-date are penalized according to their earliness (tardiness) value. Polynomial ($O(n \log n)$) time solutions are provided for some special cases. No polynomial solution exists for instances with tight due-dates. We propose a mixed integer programming model and efficient algorithms for the cases where mathematical formulation is not efficient in terms of computational time requirements. Computational results show that the proposed algorithms perform well in terms of both solution quality and computation time.

Key words: Scheduling, deteriorating jobs, deteriorating maintenance activity, common due date, earliness, tardiness.

ÖZET

ORTAK TESLİM TARİHLİ POZİSYONA BAĞLI BOZULAN İŞLER İLE BAKIM FAALİYETİNİN, TEK MAKİNEDE ERKEN/GEÇ TAMAMLANMA MALİYETLERİNİN EN KÜÇÜKLENEREK ÇİZELGELENMESİ

Fatma Şirvan
Endüstri Mühendisliği Yüksek Lisans
Tez Yöneticileri: Prof. Ülkü Gürler, Doç Dr. Mehmet Rüştü Taner,
Temmuz, 2013

Bu çalışma pozisyonlarına bağlı olarak bozulan işlerin ve makine bakım faaliyetinin, tek makinede çizelgelenmesi problemini ele alır. Problem, bakım faaliyeti ve bozulma olmadığı durumda bile NP-zor'dur. Çözüm: (i) işlerin sıralarını, (ii) bakım faliyetinin pozisyonunu, (iii) çizelgedeki ilk işin başlama zamanını kapsamaktadır. Bakım faliyetinden sonra makine başlangıç haline geri döner ve bozulma yeniden başlar. Amaç fonksiyonu ağırlıklı erkenlik ve geçlik maliyetlerini en küçüklemektir. Teslim tarihinden önce (sonra) çizelgelenen işler erkenlik (geçlik) maliyetleri ile cezalandırılır. Özel durumlar için polinom zamanlı ($O(n \log n)$) çözümler üretilmiştir. Teslim tarihinin kısıtlayıcı olduğu durumlar için ise polinom zamanlı bir algoritma mevcut değildir. Tamsayılı programlama modeli ve bu modelin çözüm zamanı açısından etkin olmadığı durumlar için algoritmalar önerilmiştir. Sayısal veriler, önerilen algoritmaların hem sonuç kalitesi hem de çözüm zamanı açısından iyi performans sergilediğini göstermektedir.

Anahtar Kelimeler: Çizelgeleme, bozulan işler, bozulan bakım faaliyeti, ortak teslim tarihi, Erken–Geç Çizelgeleme; Bozulma; Bakım Faaliyeti

Acknowledgement

First, I would like to express my gratitude to Assoc. Prof. Dr. Mehmet Rüştü TANER for his supervision, guidance, support, and encouragement during this study. Without his knowledge and support throughout three years, I would not be able to complete this study. I am also thankful for his endless moral support and understanding.

Next, I would like to thank Prof. Dr. Ülkü GÜRLER for accepting to act as my supervisor after the transfer of Dr. Taner to TED University. I also thank her for insightful comments after reading my thesis. I am also grateful to my other thesis jury committee members, Assoc. Prof. Dr. Sinan Gürel and Asst. Prof. Dr. Emre Nadar, for their valuable comments and suggestions.

I wish to thank ASELSAN Inc. for giving me the opportunity of continuing my education after my employment. I also would like to thank to my director Dr. Hayrullah Yıldız and my manager Fahri Tamer Çukur for their support and patience during my thesis.

Since this thesis is supported by The Scientific and Technical Research Council of Turkey (TÜBİTAK), I also would like to thank TÜBİTAK. ..

I would like to extend my sincere appreciation to the members of the Department of Industrial Engineering at Bilkent University.

I wish to express my deepest gratitude to my mother and friends who never stop their encouragement and always continue to motivate me.

I extend my special thanks to my brothers, Muhammet Ali and Onur Can, for their love, trust, understanding and every kind of support not only throughout my thesis but also throughout my life.

And finally, I would like to express my deepest appreciation to Ceyhun for his endless support, and motivation. His support and patience made it possible for me to overcome the stressful days and complete this thesis.

This thesis is dedicated to my mother, Nurşen ŞİRVAN.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Baker (1974) defines scheduling as a decision making process that consists of allocation of resources over time to perform a collection of tasks.

Scheduling has a crucial role in the global competitive environment as it provides efficiency in capacity utilization. Job scheduling or sequencing has a wide variety of applications, from designing the product flow and order in a manufacturing facility to modeling queues in service industries. Due to its practical importance, there is an extensive amount of research in many kinds of machine scheduling problems.

The just-in-time (JIT) phenomenon involves producing goods on time; if jobs are produced earlier (tardier) inventory costs (tardy penalty) will occur. Since jobs (or orders) are scheduled to complete close to their due dates as much as possible, scheduling models with both earliness and tardiness costs are compatible with the JIT philosophy. Many firms choose to adapt the JIT philosophy to answer customer's expectations in today's competitive markets. In recent years, scheduling problems involving both earliness and tardiness costs have received significant attention. The problem of scheduling jobs with a common due date in a single machine has been studied by several authors. Majority of the studies consider due-dates as a decision variable that can be assigned in accordance with a selected optimality criterion for a given problem. However, there may be cases in which customers are to decide on the due dates. In this work, we take into account this realistic case and set our objective as minimizing total earliness and tardiness costs with a common due date. One of the instances in which a common due date occurs, as stated by Feldmann and Biskup (2003), is when a customer orders a bundle of perishable goods which have to be delivered together in a specified time. The problem of scheduling jobs on a single-machine against a restricted common due date with the objective of minimizing total earliness and tardiness penalties is known to be NP-hard as proven by Hall and Posner (1991).

Classically, most deterministic scheduling problems assume that job processing times are fixed, however actual processing time of a job may change because of learning, aging or deterioration effects. In scheduling with the aging (learning) effect the actual processing time of a job is longer (less) if it is scheduled later in a sequence. Example triggers of an aging effect are human fatigue or machine wear both of which tend to increase the production time. Deterioration may depend either on the starting time or on the order in processing sequence of the job.

The majority of the studies in machine scheduling literature assume that machines are continuously available; however there may be some unavailability periods on the machines due to various reasons such as material shortages or maintenance activities. Generally, maintenance time is assumed to be constant, however it may increase when delayed. This type of maintenance is called deteriorating maintenance. This happens when maintenance includes cleaning, recharging, refilling etc. (Kubzin and Strusevich (2006)). In this thesis there is deteriorating maintenance, so the actual duration of a maintenance in our problem depends on its position in a sequence.

Although scheduling with deteriorating jobs and scheduling with maintenance activities are two topics that have been independently studied in the literature quite extensively, there are few studies consideringdeteriorating jobs and deteriorating maintenance at the same time.

Our problem is single machine early tardy scheduling with position dependent deteriorating jobs and deteriorating maintenance. Steel cutting process provides a practical example for our problem. Since cutting tool loses its efficiency through number of processes, job processing times increase as a function of processing sequence. This brings a need for maintenance in the form of tool sharpening. Sharpening may require more time depending on the tool's degree of deterioration. When more parts are cut after the most recent sharpening, longer time is needed to sharpen the tool to make it as good as at the beginning.

We formulate this NP-hard problem as a mixed integer linear program and identify some properties of an optimal solution. Additionally we present a heuristic algorithm that is useful for big-sized problems for which the mathematical model may be inefficient.

In Chapter 2, we fully describe the problem and we present a classification scheme. In Chapter 3, we present a brief review of the related literature. In Chapter 4, we introduce the necessary notation and give a mathematical representation of the problem. In Chapter 5, we analyze polynomial time solvable cases and identify certain properties of an optimum solution. In Chapter 6, we propose heuristic methods for the general problem. In Chapter 7, we generalize our solution procedures for problems with multiple maintenance. Chapter 8 describes the test data and heuristic variables used for our computational experimentation, and it outlines the results with a discussion on them. Finally, Chapter 9 concludes our study with a summary of our findings and few ideas for possible future research directions.

Chapter 2

PROBLEM DEFINITION

In this chapter, we first provide a formal presentation of our problem with its underlying assumptions and then we introduce the classification scheme used throughout this thesis. Finally we present mathematical formulations.

2.1 Problem Statement

The problem under investigation can be described as follows. Problem is deterministic; all the parameter values, including processing times, costs, deterioration values and functions, are known. There are n jobs to be processed on a single machine. All jobs are ready at the beginning of the planning horizon, i.e., their release times are zero. No-preemption is allowed, jobs should be processed without any interruption, in other words, those jobs whose processing is interrupted by a maintenance activity (*ma*) must be restarted. Job processing does not progress during maintenance. All jobs have the same common due date, which can be restrictive or unrestrictive. In the literature, it is generally assumed that an unrestrictive common due date for the E/T scheduling problem is one that is greater than or equal to the sum of all job processing times. Furthermore, the due date is called unrestrictive also if it is a decision variable (Feldmann and Biskup, 2003).

There is a single maintenance that can be performed after any job. The machine reverts to its initial condition after maintenance and deterioration starts anew. It is assumed that machine is in perfect working condition at the starting time of the horizon, in other words there is no need for scheduling the maintenance at the beginning of the planning horizon. Obviously, doing so would not be useful as this would occupy the machine without improving its efficiency. Maintenance duration is an increasing function of its position "q", measured in terms of the number of jobs preceding it since the beginning of the schedule (if the job is scheduled before the maintenance) or since the maintenance (if the job is scheduled after the maintenance). The two parameters needed to mathematically express the maintenance duration are; basic maintenance time (μ) and deterioration calculated based on a deterioration factor σ. Maintenance duration can be calculated by feeding the maintenance position into a deterioration function expressed in terms of these two parameters. We will use the maintenance deterioration function given as $f(q) = \mu \times q^{\sigma}$ in our study. It is important however to note that, our techniques and findings are applicable to all functions pre-calculable in the sense that the duration can be calculated for any given position without having to produce the full solution. Some example deterioration functions used in the literature, which are compatible with our procedures are listed below.

$$f(q) = \mu \times q^{\sigma}$$
$$f(q) = \mu \pm q^{\sigma}$$
$$f(q) = \mu \times \sigma^{q}$$

Job j has a basic processing time p_j and job-independent deterioration factor $a > 0$. Again, our techniques and findings are applicable to other pre-calculable functions also. Some examples of $p(j,r)$ that depend on job and their position are listed below.

$p_{jr} = p_j \times r^a$ Biskup (1999) (for learning effect)

$p_{jr} = p_j \mp a_j \times r$ Bachman and Janiak (2004)

$p_{jr} = p_j + a_j \times r$ Yang Yang (2010) (a_j is identical in our study)

$p_{jr} = p_j \times a^r$

Our actual processing time, p_{jr}, of a job j scheduled in position r is defined as follows.

$$P_{jr} = \begin{cases} p_j\, r^{a} & if\ r \leq q \\ p_j (r-q)^{a} & if\ r > q \end{cases}$$

Starting time and completion time of a job at position r are denoted by $S_r \geq 0$ and C_r, respectively. Earliness and tardiness costs are job independent. Let α and β be the common unit earliness and tardiness cost weights. E_r and T_r represent the earliness and tardiness of a job that is assigned to position r, and these are calculated as follows.

$$E_r = \max(0, d - C_r)$$
$$T_r = \max(0, C_r - d)$$

Finally, the total cost (TC) of a schedule is calculated as follows.

$$TC = \sum_{r=1}^{n} (\alpha E_r + \beta T_r).$$

2.2 Problem Complexity

Different versions of the unrestricted problem are studied in the literature. The general case of the problem has job dependent earliness and tardiness costs (α_j and β_j, respectively). Let E_{jr} and T_{jr} represent, respectively, the earliness and tardiness of a job j that is assigned to position r. The total cost for this general problem is calculated as shown below.

$$TC = \sum_{r}^{n} (\alpha_j E_{jr} + \beta_j T_{jr})$$

Kanet (1981) shows that when the common due-date is unrestrictive and $\alpha_j = \beta_j = 1$, the single machine early-tardy scheduling problem can be solved by a polynomial algorithm of O($n$log$n$) complexity. Even if $\alpha_j = \alpha,\ \beta_j = \beta$, there is an exact polynomial time solution algorithm introduced by Panwalkar et al. (1982). For the general case in which there is no restriction on the penalties of the jobs (costs may be job dependent), Aker et al. (2002) propose an exact method that combines column generation with lagrangean relaxation. They solve problems with up to 125 jobs with job dependent earliness tardiness penalties. To the best of our knowledge, there is not any better solution procedure in the literature for this problem.

In the absence of deterioration, restrictive due-date problems appear to be more difficult. It is proven by Hall, Kubiak & Sethi (1991) that common restrictive due-date, early-tardy scheduling is NP-hard (even if $\alpha_j = \beta_j = 1$ for $j=1,\ldots,n$). Due to its complexity, most of the previous studies in the literature deal with this problem using heuristic and metaheuristic approaches.

Since our problem involves the additional complexities of job deterioration and a deteriorating maintenance , it is also NP-hard when $\alpha_j = \alpha,\ \beta_j = \beta$ is also NP-hard. A comprehensive survey on the common due date early tardy scheduling can be found in Gordon et al. (2002). For the restrictive common due date problem on single machine, Biskup and Feldmann (2001) generate a total of 280 benchmark problem instances with 10, 20, 50, 100, 200, 500 and 1000 jobs. In these problems, the common due date is restricted by 20%, 40%, 60%, and 80% of the sum of all processing times. Many researchers present heuristics for this problem and they demonstrate their algorithm's performance on these 280 benchmark problem instances. However these problems are not suitable for our purposes due to lack of deterioration. Therefore, we create a new experimental framework for our computational analysis. In the following section we present a classification scheme and introduce the necessary notation for the mathematical analysis of the problem.

2.3 Classification Scheme

Scheduling problems are usually described by using the classical three–field notation introduced by Graham et al. (1979). The parameters in the three field classification system $\alpha / \beta / \gamma$ are defined as follows.

1. α denotes the machine environment and the number of machines,
2. β denotes various constraints and job characteristics
3. γ denotes the optimality criterion.

Machine environment

Different configurations of machines are possible and they are discussed in detail in Pinedo (2002). This thesis deals with a single machine that is continuously available since time zero. Single machine environment is usually considered as a special case of many others and it provides a basis for studying more complex machine environments. Also in many real applications, it is common for one machine to causes a bottleneck, in which case it makes sense to model the entire system as a single machine.

Single Stage Systems

α = 1: There is only one machine.

Job characteristics

The second field β denotes the job characteristics, which are the following in our problem.

All jobs are ready at time zero, therefore the problem is known as static, Moreover, all parameter values are known a priori. Hence, it is deterministic. All the jobs have the same due-date which is known as a common due-date problem. Since, job processing times change as a function of their position in the schedule, there are deteriorating jobs.

Additionally, a maintenance can be scheduled to increase machine productivity. The maintenance activity takes longer time when its position in schedule is delayed. Thus, the problem also involves a deteriorating maintenance activity.

The second field β denotes the job characteristics such as presence or abscence of preemption, how jobs are resumed and existence of non-availability periods, etc.

Performance measures

The last field γ denotes the optimization criteria. Commonly used performance measure is:

$\sum_{j}^{n}(\alpha_j E_j + \beta_j T_j)$: Total weighted earliness and tardiness

The objective is a function of job completion times (C_j) and related due-dates. The tardiness of a job is defined as T_j =max $(C_j$-$d,0)$, which is positive if it is processed after its due-date. Likewise, earliness is given as E_j=max$(d$-C_j $,0)$. In this study earliness-tardiness costs are job independent ($\alpha_j = \alpha$ and $\beta_j = \beta$ for all jobs) and all jobs have the same due date ($d_j = d$).

According to this classification scheme, in three field notation, our problems can be represented as: $1/ma$: $f(q)=\mu \times q^{\sigma}$, $p_{jr} = p_j \times r^a / \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$. Single machine is described with the *1* at the beginning of the notation, *ma* shows that there is a maintenance activity, with deterioration function $f(q)=\mu \times q^{\sigma}$. Parameter, p_{jr} shows job processing duration, where $p_{jr} = p_j \times r^{\mathrm{a}}$ is the deterioration function. Rest of the notation ($\sum_{r=1}^{n} \alpha E_r + \beta T_r$) shows it is an early-tardy problem with job independent costs.

Chapter 3

REVIEW OF RELATED LITERATURE

In the literature, there are a large number of papers dealing separately with earliness tardiness criteria, deterioration and maintenance activities. Our problem has three main features:

- It is an E/T scheduling problem
- Jobs are deteriorating
- There is a deteriorating maintenance activity

Since we consider the interaction between all of these issues in this study, we present the literature about the theoretical background of these topics in an organized manner.

We start with E/T scheduling problems, and then discuss the literature on scheduling with deteriorating maintenance, followed by research on deteriorating jobs. Finally, we present those studies that consider deteriorating jobs or deteriorating maintenance, with different objectives.

3.1 Early/Tardy Scheduling

The JIT phenomenon is about producing or delivering the right amount of goods or services exactly at a specified point in time. Since both late and early delivery have negative consequences, it is desirable to finish jobs right on time. In some cases customer may order a bundle of products to be delivered at a given time resulting in a common due date for all jobs in the bundle. There are many real life examples with common due dates, one of which is when a firm requires weekly deliveries of a perishable item from its wholesaler. The problems, in which producing before or after a due-date creates inventory or penalty costs, are known as Early-Tardy (E/T) scheduling problems. Baker and Scudder’s review (1990) gives a general overview of research on E/T scheduling.

In some problems, due-dates are distinct for each job, or in others there may multiple due-dates assigned to different groups of jobs. In our particular problem we assume that there is a single common due date for all jobs. Thus in the rest of this part, we focus our attention on the common due date E/T problem.

In this part some different versions of the problem are analyzed. Firstly we introduce the classical E/T problem which minimizes total absolute deviation of completion times about a common due date. After that, we discuss problems with weighted earliness tardiness costs. Then we introduce problems with additional costs which is followed by nonlinear cost functions. Finally, we give information about problems with different due-dates.

3.1.1 Minimization of Total Absolute Deviation of Completion Times about a Common Due Date

This is a simpler special case of the E/T scheduling problem. All jobs have the same due date (*d*≥0). There are two types of common due dates. If due date is not large enough, then it will not be possible to fit sufficiently many jobs before the common due date to reduce the total cost. When due date is not early enough to act as a constraint on the scheduling decision, it is called unrestrictive, otherwise it is called restrictive. To summarize, a common due date is considered as unrestrictive, if the optimal sequence can be constructed without considering the (value of) the due date; otherwise it is restrictive. A problem that has an unrestrictive (restrictive) due date is called an unrestricted (a restricted) problem.

The origins of a E/T scheduling research direction can be traced to the work of Kanet (1981). This study considers the problem of minimizing the total (unweighted) earliness and tardiness with an unrestricted common due date that is greater than or equal to the total processing time (i.e., $d \geq \sum_{j=1}^{n} p_j$). Kanet provides an algorithm for finding an optimal solution in polynomial time. The objective function for a given schedule *S* is given as follows.

$$f(S) = \sum_{j=1}^{n} (E_j + T_j)$$

This study identifies the following properties of the optimal solution of an unrestricted problem.

Property 1. There is no idle time in the schedule.

Property 2. The schedule has a V-shape. (Non-tardy jobs are in non-increasing order of job processing times and tardy jobs are in non-decreasing order of job processing times, this property is called a V-shape property.)

Property 3. In an optimal schedule, there is a job that is completed exactly at the due date (i.e., exactly on time).

Property 1 implies that if the sequences of jobs and the starting time of the schedule are known, then optimal schedule is obtained. There are *n!* different possible sequences to search. Property 2 implies that if the set of jobs that start before (after) due date is known then the sequences of jobs are obtainable with V-shape property. Hence there are 2^n ways of forming sets, instead of all n!. Even if we know the optimal job sequence, however, we still have an infinite number of schedules to evaluate because the starting time is unresolved. Property 3 implies that if the early and tardy set of jobs are known, then the starting time is obtainable because the first tardy job starts at the due-date. These three propertiesgeneralize to certain more complicated problems.

Kanet (1981) presents an O(n log n) algorithm for solving the unrestricted version of this problem with the help of these properties. Panwalker et al. (1982) simultaneously determine an optimal common due date and an optimal schedule by minimizing a weighted sum of the due date, earliness, and tardiness and present an O(n log n) algorithm. Sundararaghavan and Ahmed (1984) extend Kanet's algorithm to solve the problem of minimizing the total (unweighted) earliness and tardiness with an unrestricted common due date on m>1 processors. Bagchi, Chang and Sullivan. (1987) give an O(n log n) algorithm (with a given due date) alternative to the matching procedure (with due date assignment) of Panwalker et al. (1982), in whichthe complexity is again O(n log n).

The general restrictive version is analyzed by several authors Sundararaghavan and Ahmed (1984) present a branch and bound algorithm and a heuristic procedure for a special case of the restrictive problem, in which the starting time of the first job is forced to be at time 0. Baker and Chadowitz (1989) relax this condition and generalize the algorithm to propose a modified version of the heuristic presented by Sundararaghavan and Ahmed (1984). Bagchi et al. (1986) propose a branching procedure. Their procedure assumes that the start time of the schedule is zero. Szwarc (1989) develops several dominance conditions that are used in a branch and bound algorithm and they point out that the optimal start time may be non-zero. Hall, Kubiak and Sethi (1991) prove that the unweighted earliness tardiness problem with a restricted common due date is NP hard. Moreover they develop several dominance conditions (first and second property) and a dynamic programming algorithm that gives an optimal solution in pseudo-polynomial time, problem becomes solvable from n=25 (Szwarc 1989) to n=1000 (for the case $\alpha = \beta$)). Ventura and Weng (1995) improve the efficiency of this algorithm by remarking that some of its subroutines are unnecessary and can therefore be eliminated. Hoogeveen, Oosterhout and Van de Velde (1994) present a branch and bound algorithm that uses lagrangean relaxation to calculate both lower and upper bounds and review Emmons' matching algorithm (1987) which is and $O(n \log n)$ algorithm for the case $\alpha \neq \beta$.

To summarize, the unrestricted problem is polynomial solvable through a procedure, which ranks the jobs and assigns them from the ranked list to positions in sequence. On the other hand, the restricted problem cannot be solved in general by anything other than an enumerative algorithm. Hall, Kubiak and Sethi (1991) demonstrate that the restricted version of the problem is NP hard. In the worst case, a soultion approach requires a comparison of 2^n schedules.

3.1.2 Weighted Earliness-Tardiness Costs

In this section we review the research on the E/T problem with weighted earliness and tardiness penalties, and then we analyze job dependent weighted earliness and tardiness penalties.

When earliness and tardiness costs are different from each other, the objective function becomes, $f(sS) = \sum_{\mathrm{j=1}}^{\mathrm{n}} \left(\alpha E_j + \beta T_j \right)$.

Likewise, there is a restricted as well as an unrestricted version of the problem. In the unrestricted version, an optimal solution has the same properties (property 1, 2 and 3) with the unweigted problem (Panwalker et al, 1982).

For the unrestrictive case of this problem, there is a matching algorithm introduced by Panwalker et al. 1982, which provides an optimum solution. They introduce another property.

Property 4. Due date coincides with a job completion time that has the position *r* calculated as $r = \left\lceil \frac{n\beta}{\alpha+\beta} \right\rceil$.

For the restricted version of the problem, which is NP-hard, Bagchi, Chang and Sullivan (1987) show that Properties 1 and 2 hold and Property 3 does not hold. They propose a branch and bound algorithm that is able to solve instances with up to 25 jobs. However, no optimizing procedure has been developed for the restricted version except when the zero start time assumption is imposed. Baker and Chadowitz (1989) present a heuristic solution approach for this version. Lee and Liman (1992) present an approximation algorithm for the case $\alpha = \beta$ with performance quarantee 3/2, that is, for any instance their approximation algorithm provides a solution with no more than 3/2 times of the optimal value.

3.1.3 Job Dependent Earliness Tardiness Costs

In some cases, costs may be job dependent. For example, if jobs have different characteristics (size, material, keeping conditions etc.) or if they are orders coming from different customers, then earliness (α_j) and tardiness costs (β_j) may be job dependent. When costs are job-dependent, the objective function becomes as follows.

$$f(S) = \sum_{j=1}^{n} (\alpha_j E_j + \beta_j T_j)$$

Baker and Scudder (1990) shows that Property 1 holds and Property 2 holds. Since earliness and tardiness costs are job independent, property 2 holds in this way: non-tardy jobs are in non-increasing order of p_j/a_j and tardy jobs are in non-decreasing order of p_j/a_j

This problem is proved NP-hard by Hall and Posner (1991). Biskup and Feldmann (2001) propose a data set of 280 instances for this problem. Many researchers propose various heuristic solution methods and they measure their performance on this problem set. Due to problem complexity, many authors addressed this problem using heuristic and metaheuristic approaches. James (1997), Lee and Kim (1995), Feldmann and Biskup (2003), Hino (2005), Liao (2007), for example,use this benchmark problem set in their studies, and they present heuristics with the aim of finding better results for these benchmark problems than the previous studies.

3.1.4 Additional Penalties

Panwalker et al. (1982) defines an additional penalty (δ) that occurs due to due date delays. With the integration of this new term, the objective function becomes:

$$f(S) = \sum_{j=1}^{n} (\alpha E_j + \beta T_j) + \delta d$$

Their study shows that Property 1-3 hold, and in this version of the problem due date coincides with a job completion time that has the position r defined as follows.

$$r = \left\lceil \frac{n(\beta - \delta)}{\alpha + \beta} \right\rceil$$

Costs are calculated as a function of the job position in a way that the nontardy jobs have $\{0\alpha, 1\alpha, 2\alpha, \ldots.\}$ penalties while the tardy jobs have $\{\ldots 3\beta, 2\beta, 1\beta\}$ penalties in accordance with their processing order with respect to the common due date. The optimum solution is constructed by matching the cost coefficients in nonincreasing order with the processing times in nondecreasing order. They consider the common due date as a decision variable and they present a polynomial algorithm.

Liman et al. (1996) propose due-window related costs such that if a job is completed out of an interval then earliness or tardiness costs occur, within a window there is not any earliness or tardiness penalties. In addition to the one for the due date delays, they integrate a further cost term (γ) which is related to due window width (D). They propose a O(n log n) algorithm that minimizes total cost with the following objective.

$$f(S) = \sum_{j=1}^{n} (\alpha E_j + \beta T_j) + \delta d + \gamma D$$

3.1.5 Nonlinear Costs

There is much research on E/T scheduling with quadratic cost functions. Using a quadratic penalty function, the problem penalizes larger deviations at a higher rate. Hence, use of quadratic functions is justified by the fact that in some cases, big variations from due date is not desirable. The most commonly used objective function in this category is the following.

$$f(S) = \sum_{j=1}^{n} (\, d - C_j)^2 = \sum_{j=1}^{n} (E_j^2 + T_j^2)$$

Bagchi, Sullivan and Chang (1987) show that the unrestricted problem with the previous quadratic cost function is equivalent to the total completion time variance problem which is studied by Eilon and Chowdhury (1977). Moreover they develop dominance properties and use them in a search procedure to solve the unrestricted problem. They show that properties 1 and 2 hold in quadratic cost functions. Their approach remains essentially an enumerative one. Eilon and Chowdhury (1977) prove that an optimal schedule is V shaped and propose a few heuristic.

Panwalkar, Smith, Seidmann (1982) consider the optimum common due date assignment to minimize total cost. They show that the optimum due date is the average of all completion times (C), then problem becomes total completion time variance minimization.

$$C = \frac{1}{n}\sum_{j=1}^{n} C_j$$

$$f(S) = \sum_{j=1}^{n}(d - C_j)^2 = \sum_{j=1}^{n}(C - C_j)^2$$

Kanet (1981b) proposes a heuristic for minimizing:

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

He shows that the problem is equivalent to minimizing the variance of completion times and improves results of Eilon and Chowdhury (1977).

Since the problem is NP-hard, some heuristic algorithms are developed in the literature. Kianfar and Moslehi (2012) give an excellent summary of the relevant literature with nonlinear earliness tardiness cost functions.

3.1.6 Models with Different Due Dates

The problems in which jobs have different due dates are more complicated than the ones with a common due date. Garey at al. (1988) give a proof of NP-hardness of the problem with this more general problem with the following objective.

$$f(S) = \sum_{j=1}^{n}(\alpha_j E_j + \beta_j T_j)$$

Seidmann et al. (1981) propose a polynomial (O (n log n)) solution to a version of the problem in which due dates are decision variables. Other related studies are Yano and Kim (1986), Abdul-Razaq and Potts (1988), Ow and Morton (1988, 1989).

Properties 1 and 2 do not hold and optimal schedule may not have V shape. Problem has two parts; finding the best sequences and inserting idle times.

For the problems allowing idle time insertion, idle times can be added with the help of linear programming, after finding a schedule with no idle time. Garey et al. (1988) provides an O(n log n) timetabling algorithm for the case when a sequence is given, they sort jobs in nondecreasing order of due date and then apply the timetabling procedure, all these steps are still O(n log n). Studies with inserted time are Fry et al (1987), Yano and Kim (1986, 1991), Szwarc ve Mukhopadhyay (1995).

Valente and Alves (2005) study the problem without any idle time and they propose filtered and recovering beam search algorithms which are able to solve big problems. Studies with no inserted time are Abdul-Razaq and Potts (1988), Ow and Morton (1989), Fry et al.(1987), Gupta ve Sen (1983).

3.2 Deteriorating Maintenance

In most of the early studies, machine is assumed available even during maintenance. However, in many practical applications a machine may occasionally become temporarily unavailable due to preventive maintenance or tool change. Hence machine scheduling with an availability constraint has much practical significance. Lee (1996) conducts an extensive study of scheduling problems with an availability constraint with respect to different performance measures. Sanlaville and Schmidt

(1998), Schmidt (2000), and Ma et al. (2009) provide extensive surveys related to machine scheduling problems with maintenance.

Although most of these studies assume that maintenance duration is constant, in reality, it may depend on the running time of the machine. That is, it may be the case that the later the maintenance is performed, the worse the machine condition becomes, and a longer time is needed to perform the maintenance. This kind of maintenance is known as deteriorating maintenance.

Job processing after maintenance may be resumable or non-resumable. If an interrupted job can continue its processing when machine is available again, it is called *resumable*. On the other hand, if the job has to restart after the maintenance, it is called *non-resumable*. Hence, if jobs are resumable pre-emption is allowed, otherwise it is not. For more details, the reader may refer to Lee, Lei, and Pinedo (1997).

Kubzin and Strusevich (2006) characterize the length of a maintenance period on a machine in the form of $\alpha + f(t)$, where t is the start time, α is a given positive constant (the duration of the standard tests), and $f(t)$ is a given monotone non-decreasing function. They study makespan minimization in a two-machine flowshop and a two-machine openshop. They show that the open shop problem is polynomially solvable. However, the flow shop problem is proved binary NP-hard and pseudo-polynomially solvable by dynamic programming.

Mosheiov and Sidney (2010) study single machine scheduling problems with an option to perform a deteriorating maintenance activity. They consider the following objective functions: makespan, flowtime, maximum lateness, total earliness, tardiness and due-date cost, and number of tardy jobs. They introduce polynomial time solutions to all these problems.

Lee and Lu (2012) consider identical parallel machine scheduling problems with a deteriorating maintenance activity. They study the objective functions of minimizing the total absolute differences in completion times (TADC) and minimizing the total absolute differences in waiting times (TADW). The solution requires deciding on when to schedule the maintenance activity and determining the sequence of jobs. They show that both problems are polynomial solvable.

Li et al. (2009) investigate a single machine scheduling problem with deteriorating jobs. They show that the optimal schedule to minimize the total absolute differences in completion times is V-shaped.

Mosheiov and Sarig (2009) consider a maintenance activity with a constant duration. They assume that after maintenance, machine will revert to its initial condition and deterioration will start anew. Due to the effects of deterioration, if a job is scheduled later in a sequence, the actual processing time of it will be longer. Thus scheduling a maintenance activity at time zero is undesirable as it would take up machine time without producing any benefit. Because of the effects of deterioration, the actual processing time of a job will be longer if it is scheduled later in a sequence. This study applies this phenomenon as well.

3.3 Deteriorating Jobs

The scheduling research generally assumes that job processing times are constant, however in many production environments, a job processed later on a machine tends to consume more time due to reasons such as decreasing machine efficiency. Actual processing time of a job may change because of the learning, aging or deterioration effects. Problems considering this effect are known as deteriorating job scheduling problems. A learning effect results in job processing times to have a decreasing function of starting time orprocessing sequence. On the other hand, when there is an aging or deterioration effect processing times are modeled as an increasing function of the job’s position or starting time.

There are many studies on deteriorating job problems in the literature. Gupta and Gupta (1988) were the first to propose a problem in this category. They give a practical example of deterioration from steel production where cold ingots require longer processing time as they need to be re-heated to be worked on. Similarly, Kunnathur and Gupta (1990) provide an example from firefighting efforts, where the time needed to end fire increases with delay. Mosheiov (1996) present yet another example of searching for an object under worsening weather conditions or growing darkness.

Recently, machine scheduling problems with deterioration have received increasing attention due to their practical applications in production systems. This trend and related works are discussed in surveys by Alidaee and Womer (1999), Cheng et al. (2004) and in a recent book by Gawiejnowicz (2008). Alidaee and Womer (1999) classify deteriorating jobs models into three: linear, piecewise linear and nonlinear. According to this classification, our study falls into the nonlinear deterioration group.

Later, Bachman and Janiak (2004) studied the deterioration effect with position dependencency. The position-dependent aging effect model was first introduced by Gawiejnowicz (1996). In this model, machine has a variable speed that is constant during production but it decreases after the completion of each job. Thus the processing speed of the machine depends on the number of jobs processed.

Furthermore, the studies of scheduling with simultaneous considerations of job deterioration and maintenance have been popular topics to researchers. Makespan, flow-time, total tardiness, total completion time, number of tardy jobs, etc. have been the most commonly used performance measures so far. In the next section, we review these studies with a classification based on objective functions, which obviously directly affect the optimality properties.

3.3.1 Makespan Minimization

Browne and Yechiali (1990) study a problem with jobs whose processing times are a non-decreasing function of their start-times. They show that the makespan objective has polynomial time solution.

Wu and Lee (2003) study a single-machine scheduling problem with deteriorating jobs to minimize makespan under an availability constraint. They propose a 0–1 integer programming technique to solve linear deteriorating model when the actual job processing time is proportional to the starting time. Ji et al. (2006) investigate a similar problem but they study the non-resumable case with the objectives of minimizing the makespan and minimizing the total completion time. They show that both problems are NP-hard and present pseudo-polynomial time optimal algorithms to solve them.

The objective of Low et al. (2008) is to minimize the makespan under the non-resumable assumption and a simple linear deterioration. After proving that the problem is NP-hard, they propose a binary integer programming model and three heuristic algorithms to solve it. Lodree and Geiger (2010) study a single-machine scheduling problem with time dependent linear deteriorating processing times and a rate modifying activity that improves machine efficiency. Their aim is to derive the optimal policy to assign a single rate modifying activity to the optimum position to minimize the makespan.

Wu and Lee (2008) show that the makespan problem is polynomial solvable under the assumption of a common linear deterioration rate. Wang and Wang (2012) consider the single-machine scheduling problems with nonlinear deterioration and they show that the makespan problem remains polynomial solvable.

Kuo and Yang (2008) study a single-machine scheduling problem with multi-maintenance activities with a group balance principle and a cyclic process of the two different aging effects. They assume that jobs require to be processed by some specific tool and the tool is maintained k times in a schedule. Then it forms a cyclic process of the aging effect. After every maintenance activity the tool will be restored to its initial consition and the aging effect resumes as well. They focus on position-dependent aging effect models and provide polynomial time solution algorithms (O(n log n)). Zhao and Tang (2010) extend the study of Kuo and Yang (2010) to the case with the job and position dependent aging effect described by a power function, they also provide an O(n^4) time weighted-bipartite matching algorithm for this problem.

Rudek and Rudek (2011) provide results on the computational complexity of makespan and maximum lateness problems involving both an aging effect and an additional resource allocation.

3.3.2 Total Completion Time

Under fixed processing times, total completion time problem is polynomial solvable by an indexing policy (SPT rule). However, when processing times are time dependent no optimal procedure is known (Alidaee and Womer (1999)).

Mosheiov (1994) considers a problem, in which that the processing time of each job is a simple linear increasing function of its starting time. He showes total completion time on a single machine is polynomial solvable. He considers linear deterioration where jobs have a fixed job-dependent growth rate but no basic processing time.

Wang et al. (2008) consider single-machine scheduling problems with deteriorating jobs and a group technology assumption. They show that the makespan minimization problem and the total weighted completion time minimization problem remain polynomial solvable.

Bachman et al. (2002) study the problem of scheduling jobs with starting time dependent processing times to minimize the total weighted completion time and they prove that it is NP-hard. Wu et al. (2007) give three heuristics and a branch-and-bound algorithm and study the effects of basic processing times and deterioration rates for the problem studied by Bachman et al. (2002).

Wang et al. (2006) consider the two-machine flowshop problem with a simple linear deterioration. They develop a branch-and-bound algorithm which provides an optimal solution to problems with up to 14 jobs. They also propose a heuristic algorithm that is effective in obtaining near-optimal solutions to larger problems.

He et al. (2009) study a single-machine total completion time problem with step-deteriorating jobs. They develop a branch and bound algorithm with several dominance properties and a secondary algorithm for the problem to search for near optimal (0.3% gap) solutions. Their branch-and-bound algorithm can solve most of the problems in their set with up to 24 jobs.

Yang and Yang (2010) consider a single-machine scheduling problem involving a position dependent aging effect in the presence of maintenance activities. The aging effect in job processing times is described by a power function, and maintenance activities also have variable duration. The objective is to minimize the total completion time when the upper bound of the maintenance frequency is given in advance. They show that this problem is polynomial solvable for some types of processing time functions. Our job deterioration function is also included in this study.

S-J Yang et al. (2011) study total weighted completion time minimization problem in a two-machine flow shop under simple linear deterioration. They propose a branch-and-bound procedure and a heuristic algorithm.

Wang and Wang (2012) show that the optimal schedule of the total completion time problem on a single-machine with nonlinear deterioration is V-shaped with respect to the basic processing times.

3.3.3 Other Objectives

Mosheiov (1994) provides polynomial solutions to problems with the objective functions of makespan, total flow time, sum of weighted completion times, total lateness, maximum lateness, maximum tardiness, and the number of tardy jobs. They consider linear deterioration where jobs have a fixed job-dependent growth rate but no basic processing time. They do not consider any maintenance.

Wang and Xia (2005) present optimal algorithms for single machine scheduling problems of minimizing the makespan, maximum lateness, maximum cost and number of late jobs under a special type of linear decreasing deterioration.

Yang (2009) introduces a new model of joint start-time dependent learning and position dependent aging effects into single-machine problems. The study considers different problems; the makespan, the total completion time, and the total absolute deviation of completion times objectives to find jointly the optimal maintenance position and the optimal sequence.

S-J Yang (2011) studies single-machine scheduling problems with a simultaneous consideration of learning and aging under deteriorating multi-maintenance. In his problem, the processing time of a job depends on both its starting time and position. He shows that the problem is polynomial solvable with the objectives of makespan, total completion time, total absolute deviation of completion times and due-window related costs.

3.4 Deteriorating Jobs and Deteriorating Maintenance

There are a few studies considering the deteriorating jobs and deteriorating maintenance at the same time. For example, Wu and Lee (2003) study scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine. Wang and Wei (2010) consider identical parallel machine problems with a deteriorating maintenance activity and they decide on the sequence of jobs and maintenance activities under various objective criteria. They show that the problems remain polynomial solvable under the proposed model.

Scheduling deteriorating jobs with positional dependency and deteriorating maintenance received little attention from the research community. To the best of our knowledge, Yang (2010), Yang et al. (2010), Yang and Yang (2010a), Yang and Yang (2010b) are the only studies in the literature studying both deteriorating jobs and deteriorating maintenance with positional dependency.

Yang (2010) introduces a new model of joint start-time dependent learning and position dependent aging effects into single-machine scheduling problems with deteriorating maintenance. The objectives are to find jointly the optimal maintenance position and the optimal sequence such that the makespan, the total completion time, and the total absolute deviation of completion times (TADC) are minimized. This study also aims to determine jointly the optimal maintenance position, the optimal due-window size and location, and the optimal sequence to minimize the sum of earliness, tardiness and due-window related costs. All the studied problems can be optimally solved by polynomial time algorithms.

Yang et al. (2010) study the problem of minimizing the total earliness, tardiness, and due-window related costs to find jointly the optimal location of the maintenance activity, the optimal location and size of the due-window, and the optimal job sequence. They show that the problem is optimally solved in O (n^4) time.

Yang-Yang (2010b) consider minimizing makespan in a single machine with position dependent deteriorating effect and time dependent deteriorating maintenance activities simultaneously on a single machine. They show that the problem is polynomially solvable.

Yang-Yang (2010a) consider minimizing the total completion time in a single machine with deteriorating effects and deteriorating maintenance activities simultaneously on a single machine. They show that the problem is polynomial solvable.

3.5 Early-Tardy Cost Minimization with Deterioration

Most of the papers with deteriorating jobs examine regular performance measures in the sense that earlier job completion is more desirable. Yet in certain situations one is more interested in performance measures that are non-regular. To the best of our knowledge, there exist only a few research results on scheduling models considering non-regular performance measures under deteriorating jobs and deteriorating maintenance. Cheng, Kang and Ng (2004, 2005) consider a single machine scheduling problem with linear job-independent deterioration. They give a polynomial time algorithm to find the optimal common due date and schedule to minimize the sum of due date, earliness and tardiness penalties.

Yang et al. (2010) consider the due-window assignment and scheduling problem with job-dependent aging effects and a deteriorating maintenance, they propose polynomial time solutions to the problem. Since they assign an unrestrictive due-window, the problem is unrestricted.

Cheng et al (2012) study a single-machine due-window assignment and scheduling problem with job-dependent aging effects and deteriorating maintenance. The objective is to find jointly the optimal time to perform maintenance, the optimal location and size of the due-window, and the optimal job sequence to minimize the total earliness, tardiness, and due-window related costs.

To the best of our knowledge there is no study in the literature about earliness-tardiness cost minimization with a simultaneous consideration of deteriorating jobs and deteriorating maintenance.

Chapter 4

PROBLEM ANALYSIS AND MATHEMATICAL FORMULATION

The problem under investigation has the following characteristics. Jobs have deteriorating processing times that depend on the order in processing sequence. There exists a possibility to perform a single maintenance. The maintenance can be performed after any job. Duration of maintenance depends on its position q such that it takes longer to perform maintenance after processing of each additional job. The maintenance activity aims to improve production efficiency by reverting the machine to its initial condition. The job deterioration process starts anew after maintenance.

There are n jobs to be processed on a single machine. All jobs are available at time zero.

In this chapter firstly we analyze the problem characteristics, and then propose two mathematical formulations

Lemma 1. In an optimal schedule, there is no idle time between any two consecutive jobs.

Proof: Suppose that, there exists a schedule in which there are Δ units of idle time between two consecutive jobs and it is in the part before (after) the common due date, then obviously if we shift the schedule to the right (left) so as to make $\Delta = 0$, it will reduce earliness (tardiness) costs.

Lemma 2. In an optimal solution, the maintenance activity (if performed) is scheduled between two consecutive jobs such that there is no idle time immediately before and immediately after it.

Proof: Suppose that there are Δ units of idle time after the finishing or before the starting time of maintenance. If this Δ units of idle time is after (before) due-date, then it will cause extra tardiness (earliness) costs for all of the following (previous) jobs. Hence, eliminating the idle time by making $\Delta = 0$ reduces costs.

Baker and Scudder (1990) shows that in an unrestricted early tardy scheduling problem (without deterioration and maintenance) there exists an optimal schedule which is V-shaped with respect to the jobs' basic processing times. In other words tardy (non-tardy) jobs are in non-decreasing (increasing) order of job processing times.

Lemma 3. In the absence of a maintenance activity, there exists an optimal schedule which is V-shaped with respect to the jobs' basic processing times. In other words tardy (non-tardy) jobs are in non-decreasing (increasing) order of job processing times. Moreover even if there is a deterioration it still holds.

Proof: Baker and Scudder (1990) devides total cost in two two terms as early and tardy costs. They show that earliness and tardiness costs are depending on a positional cost terms, and they show that solution can be determined by a matching algorithm and resulting processing times explicit V shape property. In our study deterioration function is an increasing function of its postion, therefore proof is applicable for our problem in the absence of maintenance activity.

Lemma 4. If there is a maintenance activity in an optimal schedule, then there may be an optimum solution in which basic processing times do not display a V shape.

Proof: Consider the optimum solution to the example given by the following parameter values n=8, $p_j = \{10,20,30,40,60,70,80\}$, $\alpha = 1, \beta = 5$,a $= 0.6, \mu = 5$, $\sigma = 0.2$, d=100. Figure 1. shows the processing times of the jobs assigned to positions 1 through 8 in the optimum solutions with and without a maintenance activity, respectively. The optimum solution in Figure 1 has only an approximate V shape with respect to the basic processing times (p_j) due to the deformation after the maintenance activity which occurs between positions 4 and 5. A longer job is assigned to position 5 due to increased machine efficiency in terms of processing speed. When we add an additional constraint to display V shape property, optimum result increases, therefore for this proplem there is not any alternative optimal solution that exploit V shape property. This distorted *V* shape can be observed in other examples in the presence of a maintenance activity.

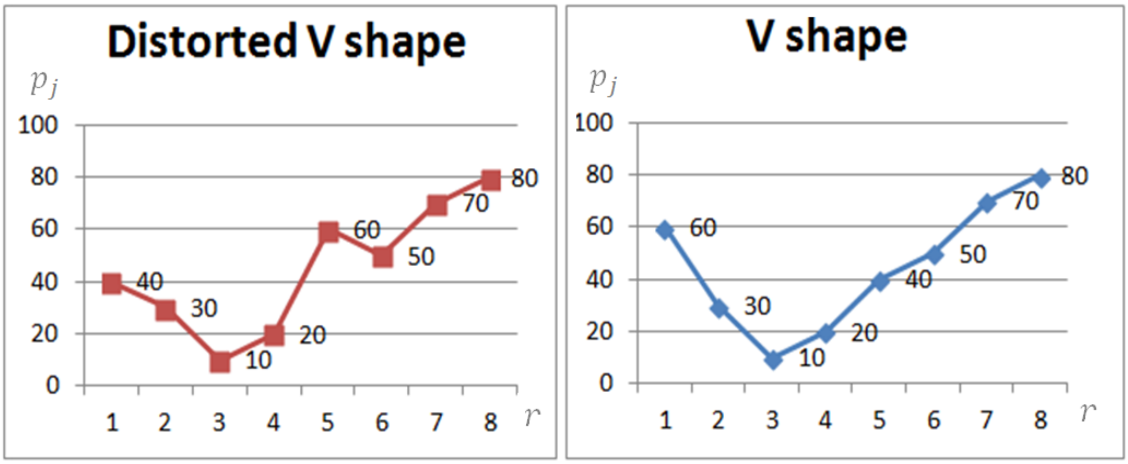

Figure 1. p_j values of Distorted V-Shape and V-Shape examples

Mathematical Formulation

In this chapter we propose two mixed integer programming formulations (MIPs) for different versions of our problem. The first model is for the general problem in which positions of both the jobs and the maintenance activity are to be determined in an optimum manner. In the second model, we consider a given position for the maintenance activity and only schedule the jobs. The second model is designed to investigate the implications on computational efficiency of an iterative procedure of solving the problem for every possible maintenance position.

4.1 Model 1

For $1/ma$: $f(q) = \mu \times q^{\sigma}$, $p_{jr} = p_{\ j} \times r^{a}\ / \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$ problem, we provide the following formulation.

Parameters:

α : earliness cost

β : tardiness cost

d: common due date

r: a position that holds a job in the schedule, *r=1,..,n.*

a: constant deterioration rate of jobs when delayed by one position.

μ : fixed time to perform the maintenance activity

σ : constant deterioration rate of maintenance when delayed by one position.

q: the job position immediately before the maintenance, q=1,..,n

$f(q)$*:* maintenance duration as a function of its position.

$$f(q) = \mu \times q^{\sigma}$$

p_j: basic processing time of job *j* without any deterioration effect, *j=1,..,n*.

p_{jrq} : processing time of deteriorated job *j*, scheduled in position *r* when maintenance is scheduled at position q.

$$p_{jrq} = \begin{Bmatrix} p_j \times r^a & \text{if } r \leq q \\ p_j \times (r-q)^a & \text{if } r > q \end{Bmatrix} \quad j=1,..,n \quad r=1,..,n \quad q=1,..,n$$

Decision Variables:

$$x_{jrq} = \begin{pmatrix} 1 \text{ if job } j \text{ is assigned to position } r \text{ while maintenance is at position } q \\ 0 \text{ otherwise} \end{pmatrix}$$

$y_q = \begin{pmatrix} 1 \text{ if maintenance is assigned to position } q \\ 0 \text{ otherwise} \end{pmatrix}$

S_r = Starting time of the job assigned to position r

C_r = Completion time of the job assigned to position r

T_r = Tardiness of the job assigned to position r

E_r = Earliness of the job assigned to position r

Formulation:

$Min\ z = \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$ (1)

st

$\sum_{r=1}^{n}\sum_{q=1}^{n} x_{jrq} = 1$ $j = 1,2,\ldots,n$ (2)

$\sum_{q=1}^{n}\sum_{j=1}^{n} x_{jrq} = 1$ $r = 1,2,\ldots,n$ (3)

$\sum_{r=1}^{n}\sum_{j=1}^{n} x_{jrq} = y_q \times n$ $q = 1,2,\ldots,n$ (4)

$\sum_{q=1}^{n} y_q = 1$ *(5)*

$C_r = S_r + \sum_{q=1}^{n}\sum_{j=1}^{n} x_{jrq}\, p_{jrq}$ $r = 1,2,\ldots,n$ *(6)*

$S_r = C_{r-1} + y_{r-1} \times \mu \times (r-1)^{\sigma}$ $r = 2,3,\ldots,n$ (7)

$T_r \geq C_r - d$ $r = 1,2,\ldots,n$ (8)

$E_r \geq d - C_r$ $r = 1,2,\ldots,n$ (9)

$S_1 \geq 0$ (10)

$T_r \geq 0$ $r = 1,2,\ldots,n$ (11)

$E_r \geq 0$ $r = 1,2,\ldots,n$ (12)

$x_{jrq} \in \{0,1\}$ $r = 1,2,\ldots,n$ $j = 1,2,\ldots,n$ (13)

$y_q \in \{0,1\}$ $q = 1,2,\ldots,n$ (14)

The objective function (1) minimizes the total earliness and tardiness cost. Constraint set 2, ensures that every job is assigned to exactly one position. Constraint set 3, restricts that every position holds exactly one job. Constraint set 4 restricts that all n jobs are scheduled with the selected maintenance position no jobs are scheduled with any other maintenance position. Constraint 5, restricts that only 1 maintenance activity can be scheduled in only one of all available positions. That is, no more than a single maintenance is allowed. Constraint set 6, calculates completion times of the jobs. Constraint set 7, gives starting times of the jobs considering also the maintenance duration. Constraint sets 8 and 9, respectively, calculate tardiness and earliness durations for every position. Constraint 10, ensures that starting time of the schedule is non-negative. Constraint sets 11 and 12, respectively, restrict that tardiness and earliness costs are non-negative variables. Lastly in constraint sets 13 and 14, x_{jrq} and y_q variables are restricted to take only "0" or "1" values.

4.2 Model 2

For 1/ma: $f(q)=\mu \times q^{\sigma}$, $p_{jr} = p_{\ j} \times r^{\mathrm{a}} \ / \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$ problem, with a given maintenance position, q, we provide the following formulation.

Additional Parameters:

q: maintenance position (given)

$p_{\ jr}$: processing time of deteriorated job j when scheduled in position r.

$$p_{\ jr} = \begin{Bmatrix} p_{\ j} \times r^{a} & if\ r \leq q \\ p_{\ j} \times (r-q)^{a} & if\ r > q \end{Bmatrix} \quad j=1,..,n \quad r=1,..,n$$

$f(q)=$: maintenance duration as a function of its position.

$$f(q)=\mu \times q^{\sigma}$$

Decision Variables:

$$x_{jr} = \begin{pmatrix} 1 \text{ if job } j \text{ is assigned to position } r \\ 0 \text{ otherwise.} \end{pmatrix}$$

$$Min\ z = \sum_{r=1}^{n}(\alpha E_r + \beta T_r) \quad (1)$$

st

$$\sum_{j=1}^{n} x_{jr} = 1 \qquad r = 1,2,\dots,n \quad (2)$$

$$\sum_{r=1}^{n} x_{jr} = 1 \qquad j = 1,2,\dots,n \quad (3)$$

$$C_r = S_r + \sum_{j=1}^{n} p_{jr} \times x_{jr} \qquad r = 1,2,\dots,n \quad (4)$$

$$S_r = C_{r-1} \qquad r = 2,3,\dots q-1,\ q+1, q+2,\dots,n \quad (5)$$

$$S_r = C_{r-1} + f(q) \qquad r = q \quad (6)$$

$$T_r \geq C_r - d \qquad r = 1,2,\dots,n \quad (7)$$

$$E_r \geq d - C_r \qquad r = 1,2,\dots,n \quad (8)$$

$$Ss_1 \geq 0 \quad (9)$$

$$T_r \geq 0 \qquad r = 1,2,\dots,n \quad (10)$$

$$E_r \geq 0 \qquad r = 1,2,\dots,n \quad (11)$$

$$x_{jr} \in \{0,1\} \qquad j = 1,2,\dots,n \qquad r = 1,2,\dots,n \quad (12)$$

The objective function (1) minimizes the total earliness and tardiness cost. Constraint set 2, ensures that every position holds exactly one job. Constraint set 3 restricts that every job is assigned to exactly one position. Constraint set 4, calculates completion times. Constraint set 5 sets jobs' processing times to start immediately after the completion time of the previous job if there is not any maintenance activity between them. Constraint 6 ensures that when there is maintenance immediately before a given job, starting time of that job is immediately after the previous job's completion time plus the maintenance duration. Constraint sets 7 and 8 calculate, respectively, the earliness and tardiness durations for jobs in every position. Constraint 9 ensures that starting time of the schedule is non-negative. Constraint sets 10 and 11, restrict that earliness/tardiness costs are non-negative variables. Lastly, x_{jr} variables can take only "0" or "1"values.

In this formulation maintenance position is assumed given. If it is a decision variable, formulation should be solved n times (i.e., for every possible maintenance position).

Since Model 2 is smaller, it is faster than Model 1. In Model 2 we divide the main problem into n sub-problems such that in every sub-problem, maintenance position is known a priori. Then we solve them independently, at the end we choose the one with the minimum cost as the optimum schedule. In Model 1, we solve the main problem directly, which requires additional variables and constraints due to the need for selecting the maintenance position. It gives the best solution in one step with the best maintenance position. Model 1 may be preferable for small problems.

Model 2 has fewer constraints and variables, which makes it faster. Since we use big sized problems in our computational experimentation, we use Model 2 in our experiments in the following chapters.

Note that it is necessary to solve Model 2 for all possible maintenance positions as the optimum objective function value is not necessarily convex with respect to the maintenance position. Consider the problem with the parameter values of $U(1,80)$, a = 0.05, α/β = 1, σ = 0.05, μ = 40, D = 1000. Figure 2 shows the optimum result as a function of the maintenance positions.

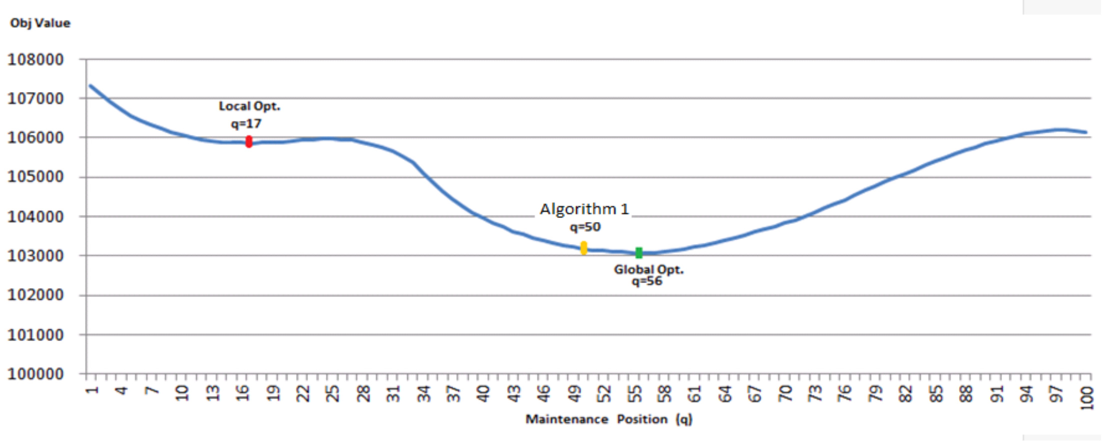

Figure 2. Average objective valuesfor the problem with $U(1,80)$, $a = 0.05$, $\alpha/\beta = 1$, $\sigma = 0.05$, $\mu = 40$, $D = 1000$

Chapter 5

POLYNOMIAL TIME SOLVABLE SPECIAL CASES

In this chapter we analyze the polynomial solvable cases of the problem. In the first part we investigate the problems that have a single common due date which is smaller than the earliest possible completion time of any job. In the second part, problems with unrestricted due-dates are analyzed.

5.1 Single Common Due-Date Smaller than the Earliest Possible Completion Time of Any Job

There may be some cases, in which the due date is already over due to some reasons or it may be smaller than the basic processing time of the smallest job. In both cases, there is not any possibility to perform any job before the due date. In other words all the jobs will be done late. Then we take due-date as zero because they are equivalent problems as shown in Theorem 1 below.

Theorem 1. When due-date is already over or it is smaller than the basic processing time of the smallest job, we can take due-date as zero and the problem becomes total completion time minimization.

Proof: Total cost of earliness tardiness: $TC = \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$

When the due-date is over or it is smaller than the basic processing time of any job, there is no possibility for finishing any jobs earlier than the due date. Thus, the total cost of the E/T problem becomes $TC = \beta \sum_{r=1}^{n} T_r$, that is, total tardiness multiplied by scalar β. Since the due-date is always smaller than the completion time of any job, we can modify the tardiness definition as follows.

$$T_r = \max(0, C_r - d) = C_r - d$$

$$\text{Total cost : } TC = \beta(\sum_{r=1}^{n} C_r - d) = \beta \sum_{r=1}^{n} C_r - \beta d$$

Since β and d are scalar, we can assume that d=0 and the problem becomes total completion time minimization.

Yang and Yang (2010) study the problem with position dependent deteriorating jobs and multiple time dependent deteriorating maintenance activities. They provide polynomial solutions for certain types of processing time functions including ours. Their findings are the most relevant ones in the literature to our eventhough they use a different maintenance duration function that depends on the starting time of the maintenance. Their relevant findings are summarized in two theorems which are applicable to both single and multiple maintenance problems where q_k is the position of k^{th} maintenance and k_0 is the maximum number of permissible maintenance activities.

Yang and Yang's Theorem 1

$1/ma$: $f(t_i) = \alpha + \beta t_i,\ k_0, p_{jr} = p_j \times r^{a_j} / \quad \sum_j C_j$, problem with job dependent positional deterioration effect and time dependent deteriorating maintenance can be solved in O (n^{k_0+3}) time. Where the duration of each maintenance activity is a linear function of the running time of the machine and is denoted by $f(t_i) = \alpha + \beta t_i$ and $\alpha > 0$ is the basic time of the maintenance activity. $\beta \geq 0$ is the deteriorating maintenance factor, t_i is the running time of the machine between the (i-1)$^{\text{th}}$ and i^{th} maintenance activities of the machine.

Lemma 5.The multiple maintenance version of our problem, $1/ma$: $f(q_k) = \mu \times q_k{}^{\sigma}, k_0, p_{jr} = p_j r^a \sum_j C_j$, problem can be solved in O ($n^{k_0+1} \log n$) time.

Proof. In Yang and Yang's study job deterioration function is the same as ours, however, their maintenance deterioration function is time dependent. However, in their proof they calculate earliness and tardiness costs based on positions and since our maintenance deterioration function is also position based, proof is applicable to our problem.

Yang and Yang's Theorem 2

If earliness and tardiness costs are job independent, the resulting problem, denoted as

1/ ma: $f(t_i) = \alpha + \beta t_i,\ k_0, p_{jr} = p_j \times r^a/\quad \sum_j C_j$, can be solved in O ($n^{k_0+1} \log n$) time.

Lemma 6. The multiple maintenance version of our problem with job dependent positional deterioration effect and position dependent deteriorating multiple maintenance, 1/*ma:* $f(q_k) = \mu \times {q_k}^{\sigma},\ k_0,\ p_{jr} = p_j r^{a_j}/\ \sum_j C_j$ can be solved in O (n^{k_0+3}) time.

Proof. Similarly, their proof is position based, changing maintenance function in the position based calculateion does not change anything.

5.2 Common Due-Date Unrestrictive

In this section we analyze problems with a common unrestrictive due date. Different ways are proposed for the determination of whether a common due date is restrictive or not. A common approach for the classical E/T scheduling problem, in the absence of deterioration or maintenance, is to consider a given common due date unrestrictive if it is greater than or equal to the sum of processing times of all jobs (Feldmann and Biskup, 2003). However, this definition does not seem very appropriate for our particular problem as we do not have the exact processing times until the job sequence is determined. Therefore, we opt to adopt a more recent definition of unrestrictive due dates, which is proposed by Ronconi and Kawamura in 2010. In this definition, a due date is called unrestrictive if its optimal value has to be determined as part of the solution or if its given value does not influence the structure of the optimal schedule. Note that, in an unrestricted problem, the optimal cost cannot decrease, when the common due date increases.

In Lemma 7 we prove that an optimal schedule exists in which the due-date coincides with a job completion time. Note that our proof mimics the approach used by Panwalker et al. (1982) for the classical problem in the abscense of deterioration or maintenance.

Lemma 7. In an optimal schedule, *d* coincides with a job completion time.

Proof. Suppose that the due date is bracketed by the completion times of two consecutive jobs in the schedule. There are two possible cases: there may or may not be a maintenance between these two consecutive jobs that bracket the due date. We address these two cases separately.

- Let C_q be the completion time of the job scheduled immediately before the maintenance activity and $p_{(r)}$ be the basic processing time of the job scheduled in position r. Suppose that there exists an optimal schedule such that; $C_q \leq d \leq C_{q+1}$ for some job j at position q. Let $\Delta = d - C_q$. Note that $0 < \Delta < C_{q+1} - C_q$ where

$$C_{q+1} = C_q + p_{(q)} + f(q) = C_q + p_{jq} + \mu \times q^{\sigma}$$

The earliness costs (E_r) associated with jobs at positions, 1, 2, …, q are given in reverse order as follows.

$E_q = \Delta$

$E_{q-1} = \Delta + p_{(q)} \times q^{a}$

$E_{q-2} = \Delta + p_{(q)} \times q^a + p_{(q-1)} \times (q-1)^a$

$\vdots$

$E_1 = \Delta + p_{(q)} \times q^a + p_{(q-1)} \times (q-1)^a + \cdots + p_{(2)} \times 2^a$

The tardiness durations (denoted by T_r) associated by position r,

$r=q+1,\ldots n$ are given by

$T_{q+1} = C_{q+1} - C_q - \Delta$

$T_{q+2} = C_{q+1} - C_q - \Delta + p_{(q+2)} \times (2)^a$

$\vdots$

$T_n = C_{q+1} - C_q - \Delta + p_{(q+2)} \times (2)^a + \ldots + p_{(n)} \times (n-q)^a$

Total Costs: $TC = \sum_{r=1}^{q} \alpha E_r + \beta \sum_{r=q+1}^{n} T_r$

Since TC is a linear function of Δ, it is minimized, either when $\Delta = 0$ or when $\Delta = C_{q+1} - C_q = f(q) + p_{(q+1)}$.

- Suppose that there exists an optimal schedule such that; $C_k < d < C_{k+1}$ for some job j at position $k>q$. Let $\Delta = d - C_k$. Note that $0 < \Delta < C_{k+1} - C_k$.

The earliness costs (E_r) associated with jobs at positions, 1, 2, …, k are given in reverse order as follows.

$E_k = \Delta$

$E_{k-1} = \Delta + p_{(k)} \times (k-q)^a$

$E_{k-2} = \Delta + p_{(k)} \times (k-q)^a + p_{(k-1)} \times (k-q-1)^a$

$\vdots$

$E_{q+1} = \Delta + p_{(k)} \times (k-q)^a + p_{(k-1)} \times (k-q-1)^a + \cdots + p_{(q+2)}$

$E_q = \Delta + p_{(k)} \times (k-q)^a + p_{(k-1)} \times (k-q-1)^a + \cdots + p_{(q+2)} + p_{(q+1)} + f(q)$

$E_{q-1} = \Delta + p_{(k)} \times (k-q)^a + p_{(k-1)} \times (k-q-1)^a + \cdots + p_{(q+2)} + p_{(q+1)} + f(q) + p_q \times q^a$

$\vdots$

$E_1 = \Delta + p_{(k)} \times (k-q)^a + p_{(k-1)} \times (k-q-1)^a + \cdots + p_{(q+2)} + p_{(q+1)} + f(q) + p_{(q)} \times q^a + \cdots + p_{(2)} \times 2^a$

The tardiness durations (denoted by T_r) associated by position r, r=q+1,…n is given by

$T_{k+1} = C_{q+1} - C_q - \Delta$

$T_{k+2} = C_{q+1} - C_q - \Delta + p_{(k+2)} \times (k+2-q)^a$

$\vdots$

$T_n = C_{q+1} - C_q - \Delta + p_{(k+2)} \times (k+2-q)^a + \ldots + p_n \times (n-q)^a$

Total Costs: $TC = \sum_{r=1}^{k} \alpha E_r + \beta \sum_{r=k+1}^{n} T_r$

Since TC is a linear function of Δ, it is minimized, either when $\Delta = 0$ or when $\Delta = C_{q+1} - C_q = p_{(q+1)}$.

- Note that the analysis of the case where there exists an optimal schedule such that; $C_k < d < C_{k+1}$ for some job j at position k<q is very similar to the case when k>q. Thus we skip it and conclude that there is an optimum solution in which the due date coincides with a job completion.

Next, we turn our attention to the calculation of the optimum due date position. Panwalker (1982) studies the classical early tardy scheduling problem with unrestricted due dates in the absence of deterioration and maintenance. His findings that are most relevant to our work are summarized in the following theorem.

Panwalker's Theorem

For an unrestricted early tardy problem without any deterioration and maintenance, optimal due-date is found with the given formula $\left(r = \lceil\frac{n(\beta-\gamma)}{\alpha+\beta}\rceil\right)$, in that study there is a due-date cost γ, in addition to early and tardy penalties. For any specified sequence π, there are r non tardy and (n-r) tardy jobs, where $r = \lceil\frac{n\beta}{\alpha+\beta}\rceil$.

Lemma 8. For the multiple maintenance version of our problem 1/ma: $f(q) = \mu \times q_k{}^{\sigma}$, k_0, $p_{jr} = p_j r^a / \sum_j C_j$, in the optimum sequence, there are r non tardy and (n-r) tardy jobs, where $r = \lceil\frac{n\beta}{\alpha+\beta}\rceil$.

Proof: Optimal due-date is found by Panwalker (1982) with the given formula $\left(r = \lceil\frac{n(\beta-\gamma)}{\alpha+\beta}\rceil\right)$. In that study there is a due-date cost γ, in addition to early and tardy penalties, however in our problem γ=0. In our problem there is also job deterioration and deteriorating maintenance, however as it is seen in the following proof, presence of deterioration or maintenance do not affect this property. From Lemma 7, we know that *k* coincides with some job completion times (i.e., *k* is an integer). It follows that *k* is the smallest integer greater than or equal to $r = \frac{n\beta}{\alpha+\beta}$.

Consider an optimal schedule and (an optimal) due-date such that d=C_k for some position *k* (from lemma 7). Let $0 \leq \Delta \leq p_{jk}$ for some job *j* at position *k*.

The effect of moving the due-date Δ units of time to the right: $\alpha r\Delta - \beta\ (n-r)\Delta$

Due to optimality we know that: $\alpha r\Delta - \beta\ (n-r)\Delta \geq 0$

Hence we have: $r \geq \beta n/(\alpha+\beta)$

Moving the due-date Δ units of time to the left: $-\alpha(r-1)\Delta+\beta(n-r+1)\Delta$

Due to optimality we know that: $\alpha r\Delta - \beta\ (n-r)\Delta \geq 0$

Again we have: $r \leq \beta n/(\alpha+\beta)+1$

Which gives us: $\beta n/(\alpha+\beta) + 1 \geq r \geq \beta n/(\alpha+\beta)$

Lemma 9. Positional weight of a job when scheduled in position "*r*" in the sequence is given by W_r:

$$W_r = \begin{pmatrix} \alpha(r-1)r^{\mathrm{a}} & r \leq q \; and \;\; C_r \leq d \\ \alpha(r-1)(r-q\,)^{\mathrm{a}} & r > q \; and \;\; C_r \leq d \\ \beta(n-r+1)r^{\mathrm{a}} & r \leq q \; and \; d < C_r \\ \beta(n-r+1)(r-q)^{\mathrm{a}} & r > q \; and \; d < C_r \end{pmatrix}$$

Proof: Total cost of the schedule is calculated as follows.

Total cost (TC) of a schedule: $TC = \sum_r^n \alpha E_r + \beta T_r$ =Earliness Costs +Tardiness Costs

If a job is completed before (after) due-date, it has earliness (tardiness) costs:

From Lemma 7 we know that an optimal schedule exists in which $d = C_k$ for some position k=1,…,n.

The earliness costs (E_r) associated with jobs at positions, 1, 2, …, k are given in reverse order as follows.

$E_k=0$

$E_{k-1}=\alpha\, p_{jk}$

$E_{k-2}=\alpha(\, p_{jk-1} + \, p_{jk})$

$E_{k-3}= \alpha(\, p_{jk-2}+p_{jk-1} + \, p_{jk})$

$\vdots$

$E_1=\alpha(\, p_{j2} + \, p_{j3} + \cdots + \, p_{jk-2} + \, p_{jk-1}+p_{jk})$

$\sum_{r=1}^{k} E_r = \alpha((k-1) \times p_{jk} + (k-2) \times p_{jk-1} + (k-3) \times p_{jk-2} + \ldots p_{j2})$

Total earliness costs = $\sum_{r=1}^{k} E_r = \alpha(r-1) \times p_{jr}$

Likewise, the tardiness costs (T_r) associated with jobs in positions k+1, k+2,…, n are as follows.

T_k=0

T_{k+1}= $\beta(p_{jk+1})$

T_{k+2}= $\beta(p_{jk+1} + p_{jk+2})$

⋮

T_n= $\beta(p_{jk+1} + p_{jk+2} + \ldots + p_{jn})$

Total tardiness costs =$\beta((n-k) \times p_{jk+1} + (n-k-1) \times p_{jk+2} + \ldots + p_{jn})$

$$\sum_{r=k+1}^{n} T_r = \beta(n-r+1) \times p_{jr}$$

Since the maintenance activity affects the processing times of succeeding jobs, it divides the cost calculation formula into two terms.

Earliness costs = $\sum_{r}^{k} E_r$

$$= \sum_{r}^{q} \alpha(r-1)r^{\mathrm{a}} \times p_j + \sum_{r=q+1}^{k} \alpha(r-1)(r-q)^{\mathrm{a}} \times p_j$$

Tardiness costs = $\sum_{k+1}^{n} T_r$

$$= \sum_{r}^{q} \beta(n-r+1)r^{\mathrm{a}} \times p_j + \sum_{r=q+1}^{n} \beta(n-r+1)(r-q)^{\mathrm{a}} \times p_j$$

Then we have the given positional weights in Lemma 9.

Lemma 10. Let there be two sequences of real numbers x_i and y_i. The sum of the elements, $\sum_{i=1}^{n} x_i\, y_i$, is least if sequences are monotonic in the opposite sense.

Proof: See page 261 Hardy et al. (1967).

Theorem 2. The scheduling problem $1/ma\colon f(q) = \mu \times q^{\sigma},\ \ p_{jr} = p_j r^{a} / \sum_j(\alpha E_j + \beta T_j)$ can be solved in O(n log n) time.

Proof: See algorithm 1 given below. We use Lemma 9 and Lemma 10 in constructing Algorithm 1. The optimal solution is constructed by matching the cost coefficients in nonincreasing order with the processing times in nondecreasing order. Algorithm 1 always gives the optimum schedule for an unrestricted problem with a single common due date.

Algorithm 1:

Step 0. Set q=1, TC^*=A big number

Step 1. Find the number of tardy and non-tardy jobs according to Property 4, which implies that there will be "*r*" non-tardy jobs and "*n*-*r*" tardy jobs.

Step 2. Calculate each value W_r, r = 1, 2, … , n as in Lemma 9.

Step 3. Assign the job with the longest original processing time to the position with the smallest value of W_r, the job with the second longer original processing time to the position with the second smaller value of W_r, and continue in this manner until all jobs are assigned. (Applying Lemma 10)

Step 4. For the given maintenance position (q), calculate deteriorated processing times (p_{jr}) of all jobs and the maintenance duration. Then start the schedule such that the completion time of the r^{th} job (found in step 1) coincides with the due date.

Step 5. Find the total cost of the schedule and record it as $TC(q)$. If $TC(q)<TC^*$ then set $TC^*= TC(q)$ and $q^*=q$.

Step 6. Set $q=q+1$ if $n>q$ go to step 1.

Step 7. Maintenance position q^* and its corresponding sequence give an optimal solution with the minimum total cost TC^*.

An example application of the algorithm is given in Appendix 2.

For a given problem. Smallest unrestricted due date can be found by Theorem 3.

Theorem 3. For any given problem the smallest unrestricted due date can be found by applying Algorithm 1 and starting the schedule at time zero,

Proof: First we assign a big number for due date ($d=M$) and find the optimal schedule. At the end we find starting time of the first job S_1 and we update d as: $d=M- S_1$. In this way we have the smallest unrestricted due date.

Chapter 6

HEURISTIC ALGORITHMS

In this chapter we analyze the problems with restricted due dates which are NP-hard. We present heuristic algorithms to obtain near optimum solutions for such problems efficiently.

When the common due-date is restrictive, the problem cannot be solved optimally in polynomial time. Therefore, a heuristic approach is justified especially for larger problems.

A constructive heuristic and an improvement step are proposed in order to obtain good solutions within a reasonable time frame.

As observed earlier in Chapter 4 a maintenance may create a deformation in the V shape. Nevertheless, preliminary experimentation indicates that even in problems with maintenance, the optimum schedule still displays a near V shape with respect to the basic job processing times. Thus, we exploit this observation and enforce Property 2 (V-shape) alongside Property 1 (no idle time in the schedule) in the heuristic solution. There is an additional property for restricted problems as it is described in Property 5. We need Property 5 to construct our heuristic.

Property 5. In a restricted problem schedule starts at time zero.

Due to definition of restrictiveness, in a restricted problem the optimal cost decreases when the common due date increases. It means the available time until due-date is not enough. In the case we do not have chance to delay the due date, it is best to start schedule as soon as possible to have more time before due date. This is why in our method restricted problems always start at time zero (as soon as possible).

In the E/T problems with no deterioration or maintenance, a V shape is normally observed at the due-date due to Property 2.

Note however that, the near V shape observed in the optimum solutions of our problems do not necessarily position the job with the shortest basic processing time at the due date. To clarify this issue we define the following additional notation.

vs: The position of the job with the shortest basic processing time.

***ds*:** The position (*r*) that holds the job whose starting and completion times bracket the due date (i.e., $S_r < d \leq C_r$).

The following example illustrates that there may not be an optimum solution satisfy $ds = vs$.

Example:

n=10, $p_j = \{10,20,30,40,50,60,70,80,90,100\}$, $\alpha = 1$, $\beta = 5$, $a = 0.6$, $\mu = 500$, $\sigma = 0.2$, *d=20*

Table 1. p_j and C_r values of the example that has V-shape after due date

R	Pj	Cr
1	80	80
2	50	156
3	30	214
4	10	237
5	20	289
6	40	406
7	60	599
8	70	843
9	90	1.179
10	100	1.578

Total cost= 26906

In this optimum solution, *ds*=1, but *vs*=4. Further experimentation suggests that if we increase "*n*", difference between *ds* and *vs* is non-decreasing. When we solve the same problem including *vs*=*ds* constraint, total cost increases (Total cost= 27323), thus there is not an alternative optimum solution satisfying that V shape is pivoted at the due-date.

In this section we propose a procedure, listed as Algorithm 2 below, to solve even large problems in reasonable computational time to near optimality. Then, we propose an improvement step as Algorithm 3.

We will need the following notation:

$J = \{ p_1, p_2, .., p_n\}$: Set of basic processing times of jobs.

$$\boldsymbol{p}_{jr} = \begin{Bmatrix} p_j \times r^a & if\ r \le q \\ p_j \times (r-q)^a & if\ r > q \end{Bmatrix} \quad \text{j=1,..,n r=1,..,n}$$

$\boldsymbol{EM(q)}$: A crude estimate of the makespan for a given q value calculated as follows: $EM(q) = f(q) + \sum_j^n \sum_r^n p_{jr} / n$

EP: Available time for production before the due date, it is equal to due date at the beginning of the algorithm.

***LP*:** Time between the due date and $EM(q)$.

***X*:** Remaining available time for processing before the due date in a partial sequence.

***Y*:** Remaining available time for processing between the due date and $EM(q)$ in a partial schedule. Everytime that a new job is assigned to a position; *X* and *Y* values should be updated.

$\boldsymbol{R = X/Y}$**:** Gives the proportion of the available space in both sides (*EP, LP*).

$\boldsymbol{A = \frac{EP}{LP}}$: gives a desirable R value. The *A* ratio is used to keep assignment in a balance by considering remaining available times in the current situation.

$\boldsymbol{S(q)}$**:** Schedule with given *q* maintenance position.

$\boldsymbol{Z(S(q))}$***:*** Total cost of given schedule *S(q),* calculated as: $Z(S(q)) = \sum_{r=1}^{n}(\alpha E_r + \beta T_r)$, where E_r and T_r values correspond to earliness and tardiness penalties of job *r* under sequence *S(q).*

Since the problem is restricted; starting time of a schedule is zero (property 5), completion times of jobs can be obtained in every position, then all earliness and tardiness costs can be calculated easily.

Algorithm 2:

Step 0: Re-index jobs in non-increasing order of basic processing times in set:

$$J = \{ p_1, p_2, .., p_n\}$$

Step 1:, *l=1.*

Step 2: *Set q=0, i=1,* $r_{min} = 1$, $r_{max} = n$, *if l=2 go to Step 4.*

Step 3: Set *q=q+1, EP=d-* $f(q)$, *LP=*$EM(q)$*-d*, $A = R = \frac{EP}{LP}$, *X=EP, Y=LP* and go to Step 5.

Step 4: Set *q=q+1, EP=d, LP=*$EM(q)$*-d-* $f(q)$, $A = R = \frac{EP}{LP}$, *X=EP, Y=LP.*

Step 5: If $X \leq 0$ or $R < A$ go to step 7.

Step 6: Assign the smallest indexed job in set J to $r_{\min}$ and delete it from set J.

Set $r_{min} = r_{min} + 1$, *i=i+1, X=X-*$p_{j,r_{min}}$, $R = \frac{X}{Y}$. Go to step 8.

Step 7: Assign the smallest indexed job in set J to $r_{\max}$ and delete it from set J.

Set $r_{max} = r_{max} - 1$, *i=i+1, Y=Y-*$p_{j,r_{max}}$, $R = \frac{X}{Y}$.

Step 8: If $i \leq n$ go to step 5.

Step 9: Obtain schedule *S(q)*. Calculate *Z(S(q)).*

If *Z(S(q))< Z(S(q***))* let q^**=q*. Set *q=q+1*. If *q<n*, go to step 2.

Step 10: If $l \leq 1$, set *l*=2 go to step 2, else stop.

At the end of algorithm $Z(S(q^*))$ gives the total cost and q^* gives the best maintenance position.

An example application of this algorithm is given in Appendix 3. The following algorithm is an improvement step that combines Algorithm 2 with Model 2.

Algorithm 3:

Step 1: Apply Algorithm 2 and find $q*$

Step 2: Apply Model 2 and find the optimum cost of the schedule with this $q*$ value.

Idea Behind the Algorithms 2 and 3

Since due date is restricted, there is not enough place before the due-date to distribute jobs liberally between the early and tardy sets to reduce total cost. That is, we may not be able to start as many jobs as we wish before the due date. Naturally, it is better to start the schedule at time *0*, that is, at the earliest possible time.

In any restricted problem we can take *EP* equal to the due date at the beginning (since it starts at time *0*), but we don't know the value of *LP*. Thus we estimate it using $EM(q)$. Estimated makespan ($EM(q)$) consists of the maintenance duration and approximate values of the deteriorated job processing times. The maintenance position q is assumed in the middle of the schedule ($q=n/2$). Since the exact positions of the jobs are not known, we obtain the approximate deteriorated job processing time for each given job by calculating its processing time in every possible position (1..*n*) and then taking the average of these.

Sum of expected processing times: $\sum_j^n \sum_r^n p_{jr} /n$

Adding maintenance time, gives $EM(q)$.

$$EM(q) = f(q) + \sum_j^n \sum_r^n p_{jr} /n$$

When we have the $EM(q)$ value, we can use it to estimate EP and LP values. However, since we do not know, if the maintenance is before or after the due-date, we do not know if EP is equal to the due date or due date minus maintenance duration. Likewise, LP may be equal to either $EM(q)$ minus the due date or $EM(q)$ minus the due date minus the maintance duration. Hence, we apply the heuristic for both cases this by the help of parameter *l*. When *l=1* it the maintenance starts before due-date. First, we reduce *EP* as much as maintenance duration (step 3) since *l*=1, and apply the rest of the algorithm, then we set *l*=2 at step 10 and we reduce *LP* as much as maintenance duration instead of *EP* (step 4) and apply the rest of the algorithm. When we estimate the values of *EP* and *LP*, we use their ratio *A*, and assign jobs so as to keep the current ratio (*R*) close the desired ratio (*A*) in every step until there is not sufficient space before the due date to assign any more jobs. If there is not space for another job before the due date, we do not need to check the ratios. As, we proceed by assigning all remaining jobs to the later part after the due date. After all jobs are assigned, we calculate the total costs. This procedure gives the heuristic result for the given maintenance position. To find a potentially better schedule, we run this algorithm for every possible maintenance position (*q*), and update the best result and its corresponding maintenance position. At the end, we find the best q for the algorithm. Complexity of the algorithm is $O(n^2)$.

Preliminary test runs indicate that Model 2 (section 4.2) is fast for a given maintenance position for problems with up to *n*=200 jobs (takes approximately 20 min). However, because we need to try all possible maintenance positions for optimum result it takes longer time. The aim of Algorithm 3 is that; if we have estimation about where the best maintenance position is from Algorithm 2, then we may find a potentially better result by solving the IP for that maintenance position.

Moreover this heuristic algorithm can be generalized for the restricted problems with multiple maintenances, which also provides optimum results with O(n^{k_0+2}) complexity.

Chapter 7

MULTIPLE MAINTENANCE PROBLEMS

In many practical applications, there can be a number of maintenances in a planning horizon, which are subject to deterioration. These may be some minor upkeep activities involving routine operations such as cleaning, lubrication, oil changes, and adjustments. For example, a delayed cleaning activity takes more time than it would when done in a more timely manner. In this chapter, we analyze the problems with multiple maintenance.

First, we extend one of our mathematical formulations (Model 2) to cover the case with multiple maintenance activities. This formulation is applicable for both restricted and unrestricted problems, however since we propose a polynomial time algorithm by extending Algorithm 2 for unrestricted problems with multiple maintenance at the end of this chapter, this formulation is particularly useful for the problem with restricted due-date.

7.1 Mathematical Formulation with Multiple Maintenance

For the problems with multiple maintenance, we propose the use of Model 2 with a small adaptation, for which the input parameters are calculated as follows.

Let k and q_k be the number of maintenance activities and the position of the k^{th} maintenance, respectively. Suppose that there is an upper bound k_0 on the number of maintenance activities (i.e., $k<=k_0$)

The upper bound on the number of maintenance activities should be known a priori. Then we consider all possible maintenance positions q_k (k=1, 2,…, k_0). B denotes the set of all maintenance positions: $B = \{q_1,q_2,\ldots,q_{k_0}\}$.

The duration of the maintenance activity scheduled in position r is given by

$$f(r) = \left\{ \begin{array}{cc} \mu \times r^{\sigma} & if\ r \leq q_1 \\ \mu \times (r-q_1)^{\sigma} & if\ \ q_1 < r \leq q_2 \\ \mu \times (r-q_2)^{\sigma} & if\ \ q_2 < r \leq q_3 \\ & \vdots \\ \mu \times \left(r-q_{k_0-1}\right)^{\sigma} & if\ \ q_{k_0-1} < r \leq q_{k_0} \end{array} \right\}$$

The processes of both job and maintenance deterioration start anew, after every maintenance activity. Since all maintenance activities affect processing times of jobs in the schedule we calculate the processing time of jobs in different positions as follows.

$$p_{jr} = \left\{ \begin{array}{cc} p_j \times r^{a} & if\ r \leq q_1 \\ p_j \times (r-q_2)^{a} & if\ \ q_1 < r \leq q_2 \\ p_j \times (r-q_3)^{a} & if\ \ q_2 < r \leq q_3 \\ & \vdots \\ p_j \times \left(r-q_{k_0}\right)^{a} & if\ \ q_{k_0-1} < r \leq q_{k_0} \end{array} \right\} \quad \text{j=1,..,n r=1,..,n}$$

We modify constraints 5 and 6 in Model 2 with new definition of maintenance durations as follows.

$$S_r = C_{r-1} \qquad\qquad r \notin B \quad (5)$$

$$S_r = C_{r-1} + f(q_{r-1}) \qquad\qquad r \in B \quad (6)$$

Once the maintenance durations ($f(q_r)$) are calculated in this way, Model 2 given in Section 4.2 can be used with this small adaptation in constraint sets 5 and 6 to solve the multiple maintenance problem with given maintenance positions. Obviously, since we solve the problem for all possible combinations of maintenance positions, the computational complexity of the solution procedure increases.

In particular, we solve n^{k_0} problems to find the optimum solution, where k_0 is the upper bound on k as defined before. Note however that, in some cases it is possible to find a tighter bound on the optimum number of maintenance activities and stop the procedure without having to consider all combinations planned at the beginning. In other words if we notice that a problem's total cost with k^*+1 maintenance is not less than the one with k^*, it means that there is no need for considering any additional maintenance. In that case k^* is a tighter upper bound than k_0.

7.2 Multiple Maintenance with Unrestricted Common Due-Date

In this section, we consider the case of multiple maintenance activities with an unrestricted common due date. These multiple maintenance problems can be solved pseudo-polynomially.

Theorem 4. $1/ma\colon f(q_k) = \mu \times q_k{}^{\sigma},\;\; p_{jr} = p_j r^{a} / \sum_j(\alpha E_j + \beta T_j)$ problem can be solved in O ($n^{k_0+1}\log n$) time.

Proof: In an optimal schedule, the positional weight of a job when scheduled in position "r" in the sequence is given by W_r calculated as follows.

If the vector of maintenance positions, *Q(q,k)*=($q_1, q_2, \dots, q_{k+1}$) is known in advance, then we can calculate the weights of positions as follows.

$$W_r = \begin{pmatrix} \alpha(r-1)r^{a} & r \le q_1 \; and \;\; C_r < d \\ \alpha(r-1)(r-q_1)^{a} & r > q_1, r \le q_2 \; and \;\; C_r < d \\ \alpha(r-1)(r-q_2)^{a} & r > q_2, r \le q_3 \; and \;\; C_r < d \\ & \vdots \\ \beta(n-r+1)(r-q_k)^{a} & r > q_k, r \le q_{k+1} \;\; and \;\; d < C_r \\ \beta(n-r+1)(r-q_{k+1})^{a} & r > q_{k+1}, r \le q_{k+2} \;\; and \; d < C_r \\ & \vdots \end{pmatrix}$$

For a given set of maintenance positions (*Q(q,k))* and corresponding positional weights (W_r), the problem can be solved in O ($n \log n$) time, after arranging the jobs in non-decreasing order of their basic processing times, by Algorithm 1.

However, it is necessary to try all possible combinations of maintenance positions with related W_r values and repeat the algorithm to find the optimum solution. Since *Q(q,k)* vector is bounded by n^{k_0}. Therefore the time complexity is O ($n^{k_0+1}\log n$).

Chapter 8

COMPUTATIONAL RESULTS

In this chapter, we experimentally test the performance of the heuristic algorithm and the mathematical formulation. In the following, we present the experimental framework, computational results and our inferences. In the first part we analyze problems with single maintenance, in the second part we analyze unrestricted problems with multiple maintenance.

8.1 Single Maintenance Tests

We generate several problem combinations with single maintenance using the parameters listed below.

Basic Processing Times: Basic processing times are randomly drawn from a discrete uniform distribution over [1, A]. We consider two levels for A, i.e., we use two levels of basic processing times in our experiments. In Level 1, A=40, i.e., $p_j \sim U[1,40]$, and in Level 2, $A = 80$, i.e., $p_j \sim U[1,80]$.

Maintenance Duration: Similar to the processing times, the maintenance duration also depends on the position of the maintenance activity in the schedule. As in the case of processing times, the maintenance has a basic time component that is independent of its position (e.g., time required to turn on the machine), which is called the basic maintenance time μ. We choose the value of μ as half of A. That is the basic maintenance time can take on values of 20 and 40 depending on the given level of A in a particular treatment.

Job and Maintenance Deterioration Factor: Deterioration of both the processing times and the maintenance duration are determined based on their position and the value of a deterioration factor. The deterioration factor of the processing times a, and the deterioration factor of the maintenance σ are varied between two levels, 0.05 and 0.2. Hence, for a 100-job problem, in the last position of the schedule, deterioration becomes approximately 1.25 ($100^{0.05}$) and 2.5 ($100^{0.2}$) times the basic maintenance time when the deterioration factor is set as 0.05 and 0.2, respectively.

A similar factor applies to maintenance also. We use the same deterioration factors of job processing and maintenance duration in our experiment but they may be different in general and the general case can also be solved by our procedures in the same manner.

Problem Size: The problem size is determined by the number of jobs, n. We use n=100 in all experiments except in the ones in which we assess the computational efficiency as a function of the problem size.

Early/Tardy Penalties: We also manipulate the ratio of the earliness to tardiness penalties based on three levels. In particular, we consider early penalties (α) that are equal, twice and half of the corresponding tardy penalty (β).

Due Date: We consider both tight and large common due-date (*d*) problems.

In the previous literature, h is used as a restriction factor of due date as shown in the following formula:

$$d = h \times makespan$$

If there is not any deterioration, makespan is defined as the sum of all processing times. However due to deterioration in our problem, we cannot calculate makespan without having the optimal schedule. Thus we define $EM(q)$ as an estimation of the makespan.

Once the makespan is estimated, we find the h value (restrictiveness of due-date) with the following formula.

$$h = \frac{d}{EM(q)}$$

Several authors (e.g., Ow and Morton (1989), Yano and Kim (1991)) describe test data generation procedures for various E/T problems. However, to the best of our knowledge, no such procedure has been proposed for the E/T problems with deterioration. Since deterioration has an important effect affecting the nature of the problem, we need to generate new test data. Consequently, test problems are generated using all combinations of the factors given in Table 2 below.

Table 2. All factor levels in experimental design

U	a	a/β	σ	μ	D
1-40	0.05	1	0.05	20	200
1-80	0.2	0.5	0.2	40	1000
		2			

Using the parameters that are given in Table 2, h values of the experimental design are calculated as given in the following table.

Table 3. h values of experimental design

U	a	σ	μ	D	h
1-40	0.05	0.05	20	200	0.1 (~0.095)
1-40	0.05	0.05	20	1000	0.5 (~0.474)
1-80	0.2	0.2	40	200	0.03 (~0.029)
1-80	0.2	0.2	40	1000	0.15 (~0.147)

The solution quality in terms of percentage deviation from the optimum result, denoted by ε, is measured as follows.

$$\varepsilon = \left(\frac{TC_h - TC_{opt}}{TC_{opt}}\right) * 100$$

TC_h is the total cost of the heuristic model and TC_{opt} is the optimum solution obtained by the mathematical model.

Full combination of the factors given in Table 2 creates 96 problems (treatments). And we create 10 different basic processing time values for each of U(1,40) and U(1,80). These processing times are listed in Appendix 4.

Hence we solve 960 problem instances. We apply both the heuristic and the exact approaches to these instances. Since the exact procedure solves the MIP for each possible maintenance position in a given problem, the MIP is solved 100 times for each problem instance in GAMS resulting in a total of 96,000 MIP solutions for all of the 960 problems in the problem set. The average percentage errors, average CPU times and objective values are also given in Appendix 4.

The proposed heuristic is coded using MATLAB and the MIP is solved in GAMS. The experimentation is run on a computer with an i7-2670QM 2.2 Ghz processor and 8GB of RAM.

Apparently, problem size is a major factor that affects the CPU times. To test the effect of this factor we first solve problems for different number of jobs selected as $n \in \{25, 50, 100, 200\}$ and other parameters as given in Table 4.

Table 4. Factor levels for assessing computational efficiency

Problem sets	$\boldsymbol{U}$	$\boldsymbol{A}$	$\boldsymbol{\alpha/\beta}$	$\boldsymbol{\sigma}$	$\boldsymbol{\mu}$	$\boldsymbol{D}$
1	1-40	0.05	0,5	0.05	20	200
2	1-40	0.05	1	0.05	20	200
3	1-40	0.05	2	0.05	20	200
4	1-40	0.05	0,5	0.05	20	1000
5	1-40	0.05	1	0.05	20	1000
6	1-40	0.05	2	0.05	20	1000
7	1-80	0.2	0.5	0.2	40	200
8	1-80	0.2	1	0.2	40	200
9	1-80	0.2	2	0.2	40	200
10	1-80	0.2	0.5	0.2	40	1000
11	1-80	0.2	1	0.2	40	1000
12	1-80	0.2	2	0.2	40	1000

Average solution times of exact and heuristic approaches are given in Table 5.

Table 5. Solution times of mathematical model and heuristic

N	Math. Model		Algorithm 2 (Sec.)	Algorithm 3	
	Mean Time	**Standart Deviation (sec)**		**(Algorithm 2+ Math. Model 1 instance) (Sec.)**	**Average Gap (%) in Objective Function(%)**
25	20 sec.	0,64	0	1	0,244
50	45 sec.	1,2	0	1	0,122
100	125 sec.	1,3	1	2	0,106
200	24 min.	1,1	3	10	0,163
300	122 min. (25 sec per instance)	-	7	32	-
500	1 day + (3min per instance)	-	19	3 min	-
750	750x8 (8 min per instance)	-	49	9 min	-
1000	1000x28 (28 min per instance)	-	98	30 min	-

For small-sized problems, we are able to obtain optimum solutions using the mathematical model repeatedly for every possible maintenance position. As seen on Table 5, if a problem has more than 200 jobs, mathematical formulation takes a long time to give the optimum solution. Since we have to try *n* (instances) possible maintenance positions, for every problem instance we have to run the model *n* times and one of them takes almost 7 sec., which makes 24 min in total (7*200 sec). Consequently, it was not practical to solve larger problems with n>200 repeatedly for every possible maintenance position. Since standard deviations of the solution times are small (1.06 sec) in smaller sized problems, we assumed that it takes virtually the same amount of CPU time to solve for any maintenance position of a given problem size. Hence, we solved larger instances with more than 200 jobs (i.e., n∈{300,500,750,1000}) for a single maintenance position, $q = \left(\frac{n}{2}\right)$, which is in the which is in the middle of the schedule, and multiplied it by the numbe of positions to estimate CPU times of the mathematical model for 300, 500, 750 and 1000-job problems listed in Table 5. Since the exact procedure fails to provide results for these larger problems with a reasonable computational effort, use of a quicker heuristic approach is justified. Morover, we found the average objective error for 25, 50, 100, 200-job problems with the problem set given in Table 4. As it is seen, when problem size is small (n=25) average gap is higher, however if we increase *n*, average error is almost stable.

Having tested the effects of problem size on the computational performance of our algorithms, we return to our 100-job problem sets to analyze the effects of the other factors listed in Table 2 on various response characteristics. We analyze the effects of the input factors on objective value by Anova given in Table 6.

Table 6. Analysis of Variance for Objective Value

Source	DF	SS	Q	F	P	
U	1	3,15E+11	3,15E+11	394,86	0	<0.05
a	1	1,32E+11	1,32E+11	165,54	0	<0.05
α/β	2	2,24E+11	1,12E+11	140,25	0	<0.05
σ	1	65662642	65662642	0,08	0,775	
μ	1	92421387	92421387	0,12	0,735	
D	1	6,81E+10	6,81E+10	85,28	0	<0.05
(α/β)*D	2	1,79E+10	8,95E+09	11,21	0	<0.05
σ* μ	1	4023788	4023788	0,01	0,944	
U*a	1	1,82E+10	1,82E+10	22,82	0	<0.05
Error	84	6,71E+10	7,99E+08			
Total	95	8,43E+11				

```
S = 28262,0   R-Sq = 92,04%   R-Sq(adj) = 91,00%
```

ANOVA results indicate that the optimum objective value depends mostly on basic processing times, the deterioration factor of the processing times, early-tardy costs and due date. Since maintenance duration (basic time plus its deterioration) is not so long in comparison to the schedule duration, it does not seem to have a significant effect on the objective function value for these 100-job problems.

Using results that are listed in Appendix 4 and summarized in Figure 3, we analyze effects of all factors' on objective function value. Figure 3 shows the average objective values of the instances for different levels of the 6 factors studied.

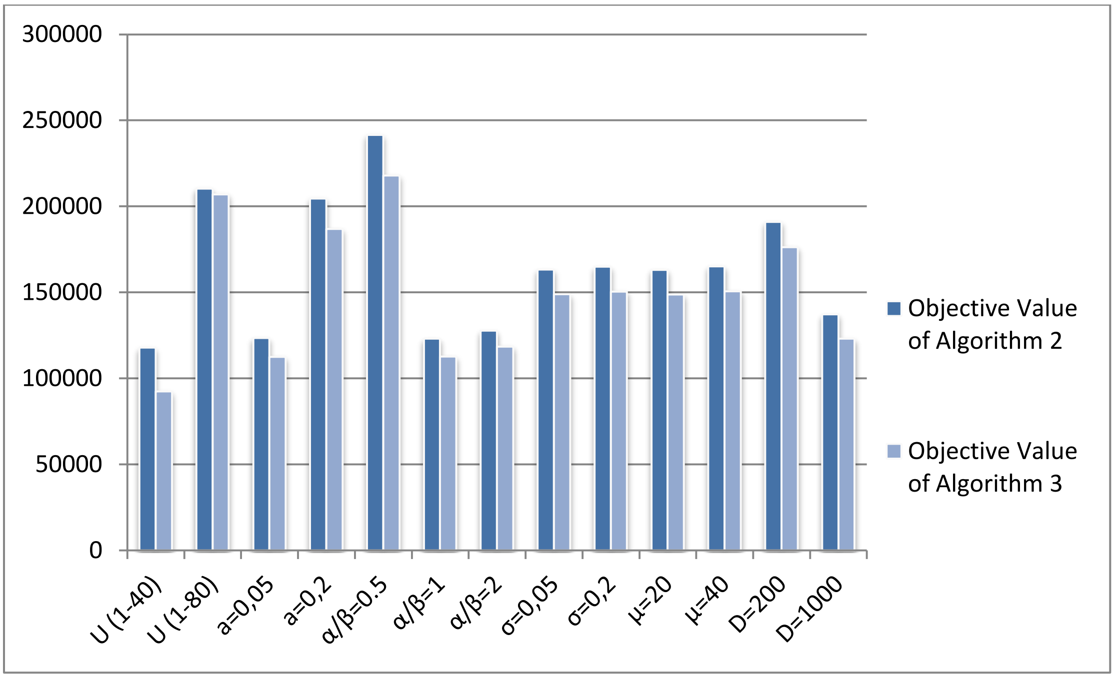

Figure 3. Average objective values for different factor levels

As seen in Figure 3, when parameters' values increase objective value increases, except α/β and D factors.

- U values show the sampling distribution of the basic processing times of jobs. Factor a is the deterioration factor for these processing times. When either of these factors increases the actual processing times get larger and hence more jobs get completed farther from the due date. As a natural result of this, the objective function value also increases.

- When α/β ratio takes a value of 0.5, the tardy penalty is twice the early penalty (α=1 *and* β=2), and hence scheduling jobs early as opposed to tardy becomes more desirable. However for restricted problems, the number of jobs that can be scheduled before the due date is limited. Therefore, objective value is larger when the earliness and tardiness cost ratio is greater (α=2 *and* β=1). In the case α/β=1 (α=1 *and* β=1), objective function is smallest as expected, since cost multipliers are smaller.

- Smaller due dates create restriction on the schedule, and in turn an increase in penalty costs. When due date increases, problem becomes less restricted and objective value decreases.

- Neither the basic time nor the deterioration factor of the maintenance activity seems to have significant effect on the objective function value in our experiments. This graphical observation is in parallel with the previously reported ANOVA results.

The ANOVA results given in Table 7 reveal that the statistically significant factors affecting the performance of Algorithm 3 are basic processing times, early-tardy costs and due date.

Table 7. Analysis of Variance for Heuristic % Gap

Source	**DF**	**SS**	**Q**	**F**	**P**	
U	1	0,74	0,74	13,10	0,00	<0.05
a	1	0,10	0,10	1,72	0,19	
α/β	2	1,04	0,52	9,16	0,00	<0.05
σ	1	0,00	0,00	0,01	0,94	
μ	1	0,03	0,03	0,54	0,47	
D	1	1,06	1,06	18,72	0,00	<0.05
(α/β)*D	2	0,89	0,45	7,86	0,00	<0.05
σ* μ	1	0,01	0,01	0,14	0,71	

U*a	1	0,32	0,32	5,66	0,02	<0.05
Error	84	4,77	0,06			
Total	95	8,97				

```
S = 0,238309   R-Sq = 46,81%   R-Sq(adj) = 39,84%
```

As in the case with the objective function, maintenance duration again is not a factor that affects heuristic performance significantly. Note however that although the magnitude of maintenance duration does not affect heuristic performance and objective function; maintenance position is critical due to its effect on the job processing times. Thus it is still necessary for the algorithm to test for every possible maintenance position and pick the best one.

In Figure 4, we take average percentage deviation from optimum of the instances (Appendix 4) for different levels of the factors studied.

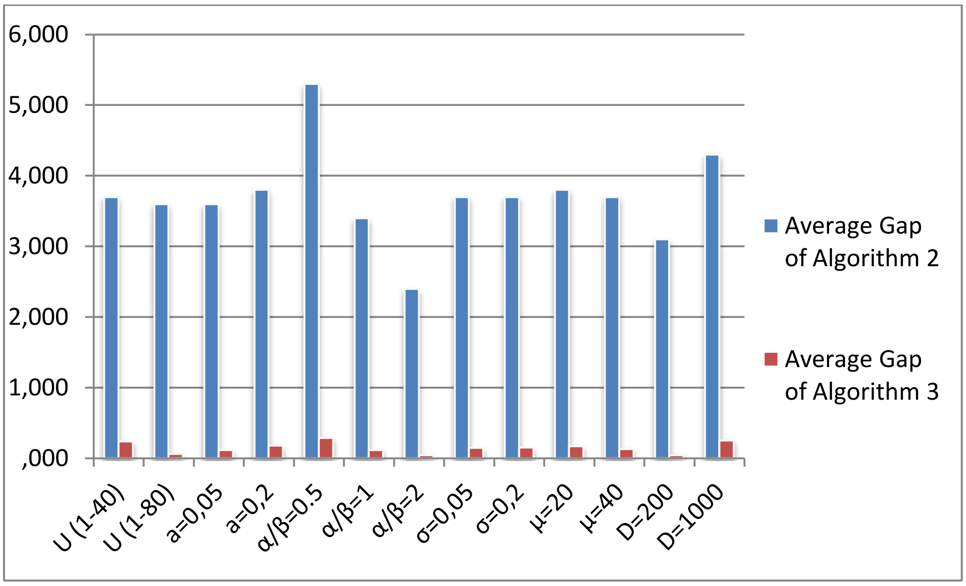

Figure 4. Average percentage gap (error) of given factors

As seen in the figure, Algorithm 3 performs markedly better than Algorithm 2 for all factor levels. The figure also reveals the following observations.

- Due date has a critical role on algorithms performance. If due date is large, the problem is less restricted. As we mention earlier, unrestricted problems are easier to solve. When due date increases, problem becomes less restricted and average error decreases.

- If α/β ratio increases, it means early penalty is increasing in comparison to tardy penalty, scheduling jobs late becomes more desirable. Since there is not any limit after due date, we can assign as many jobs as we desire after the due date and hence, the problem becomes easier in the sense that the difficult decision to schedule a given job early or tardy is usually answered as tardy. For such easier problems, the heuristic performs better and hence we observe smaller gaps.

- Note that the effect on the heuristic performance of a level change in factor U is smaller than that of due date and early tardy costs. As expected when processing times are smaller the total cost is also smaller (recall from Figure 3), and a small difference from optimal schedulemay creates big percentage gaps.

- Finally the effect of maintenance is not significant, which may be due to the fact that try all possible positions.

Overall, computational results show that the proposed algorithm performs well in terms of solution quality. Average gap of Algorithm 2 is 3.7% and worst case error is 11.66%. On the average, Algorithm 3 provides very close results to optimum. In our problem set (given in Appendix 4) average gap is 0.15% and worst case error is 3%.

8.2 Multiple Maintenance Tests

Recall from Section 7.2 that we propose a polynomial time solution approach to unrestricted problems with multiple maintenance activities. This section presents experimental results to observe the effects of having the liberty of performing multiple maintenance activities on the characteristics of the optimum solution. The question of particular interest is how many maintenance activities are desirable under different factor combinations. We also look into the factor effects on optimum objective value and CPU time performance of the solution procedure in this new problem setup.

The relevant factors that determine the problem characteristics in the experimental framework are the number of jobs, variability in job processing times, early-tardy penalties, fixed time of maintenance, and finally the deterioration rates of maintenance activities and jobs given in Table 8. In Table 8, there are additional factor values relative to those Table 2 except for due date (*D*). Since we choose an unrestricted due date, we use only one value, because any unrestrictive due date does not change objective value. Because the problem is unrestricted, we use the polynomial time algorithm, therefore it is easier to solve larger problem sets with more threatments.

Table 8. Factor levels in experimental design for multiple maintenance problems

U	a	α/β	σ	μ	D
1-80	0	1	0	0	2500
1-40	0.05	0.2	0.05	20	
	0.2	0.5	0.2	40	
	0.5	2	0.5	80	
		5			

Processing Times: Processing times are sampled from a discrete Uniform distribution between (1-80) and (1-40).

Basic Duration of Maintenance: We choose its value as different multiples of the upper limit of the range of processing times, A. We consider four levels of this factor as, 0, 0.5A=(20,40) and A = (40,80).

Job and Maintenance Deterioration Factor: We use the same deterioration factor for jobs and maintenance activities in any given problem. The four levels of this factor considered in the experimental design are 0, 0.05, 0.2 and 0.5. Levels of 0 and 0.5 are instrumental in analyzing the cases in which there is no deterioration effect or where there is a considerable deterioration effect.

Problem Size: We consider n=30 jobs in all problems. Since three maintenance activities are allowed in the problem it increases problem complexity n^3 times in comparison to the single maintenance case. Thus n=30 is chosen instead of the 100 in earlier experiments to keep the computational effort at a reasonable level.

Due Date: Note that the specific value of an unrestricted due date does not have an effect on the optimum solution. That is, the optimum solution remains the same for any larger due date value. Thus, we use a large due date value (d=2500) which is unrestrictive for all the problems.

We consider all combinations of the control factor levels, resulting in a total of 640 (2x4x5x4x4) test problems. Moreover we use ten replications of each test problem to account for the stochastic processing time factor. Therefore, there are 6400 problems instances in this test bed.

When the solution gives maintenance positions q_1, q_2, q_3 to be all equal to 30 (i.e., the problem size in terms of the number of jobs) then even the first maintenance is scheduled in the very last position. This means that there is no maintenance in the optimum schedule. Likewise, if q_2 and q_3 are both 30 but q_1 is strictly smaller than 30, then there is only one maintenance in the optimum solution. Finally if q_3=30 and $q_1<q_2<30$, then there are exactly two scheduled maintenance activities in the optimum solution with no need for a third maintenance.

We analyze the effects of the input factors for the multiple maintenance case by Anova given in Table 9. CPU time depends mostly on the α/β ratio. Other factors do not significantly affect solution times.

Table 9. Analysis of Variance for CPU Time

Source	DF	SS	MS	F	P	
U	2	56,2	18,2	1,4	0,210	
a	3	57,8	19,3	1,54	0,202	
α/β	4	22801,8	5700,4	455,56	0,00	<0.05
σ	3	4,0	1,3	0,11	0,956	
μ	3	73,4	24,5	1,95	0,119	
Error	3186	39866,4	12,5			
Total	3199	62803,3				

S = 3,53737   R-Sq = 36,52%   R-Sq(adj) = 36,26%

As seen in Figure 5, when the ratio of earliness and tardiness increases, the algorithm takes longer time. This may be explained by the fact that when the early penalty is larger than the tardy penalty, it becomes more critical to schedule in the limited number of positions available before due date.

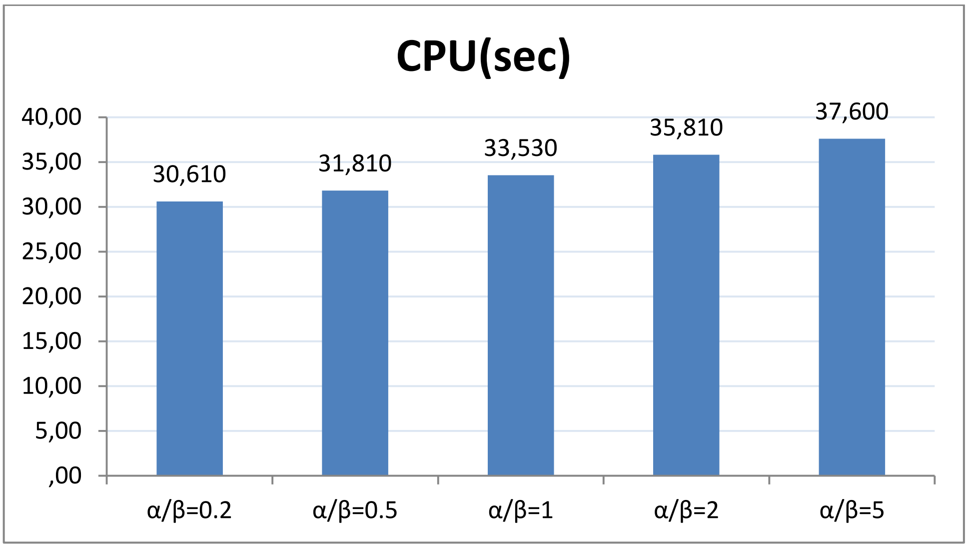

Figure 5. Average *CPU* times for the multiple maintenance problem

As expected, the objective value depends on all of the factors, as seen in ANOVA shown in Table 10.

Table 10. Analysis of Variance for Objective Value

Source	DF	SS	MS	F	P	
U	3	36909255808	12303085269	2792,47	0,000	<0.05
μ	3	1578794633	526264878	119,45	0,000	<0.05
σ	3	367918965	122639655	27,84	0,000	<0.05
α/β	4	11977367908	2994341977	679,64	0,000	<0.05
Error	3186	14036887672	4405803			
Total	3199	64870224986				

S = 2099,00   R-Sq = 78,36%   R-Sq(adj) = 78,27%

Figure 6. reveals the following observations.

- When *U* value is smaller the total cost is smaller, because longer jobs create bigger earliness and tardiness costs.

- The pattern of change in the gap as a function of α/β ratios is rather interesting. When the early penalty increase in comparison to the tardy penalty, scheduling jobs in the tardy set becomes more desirable. Since there is not any limit in the number of jobs that can be scheduled after the due date, we can assign as many jobs as we desire in the tardy set and hence, the problem becomes easier in the sense that the difficult decision to schedule a given job early or tardy is usually answered as tardy. For such easier problems, the heuristic performs better and hence we observe smaller gaps.

- When we decrease tardiness cost while tardiness earliness cost is the same, objective value decreases since the cost multiplier decreases. When we increase earliness cost while tardiness earliness cost is the same, objective value increases since the cost multiplier increases.

- Finally the effect of maintenance is not significant, which may be due to the fact that we try all possible positions.

As for the factor effects on the objective function value, if the factor values increase (except earliness tardiness ratio), objective value increases in parallel.

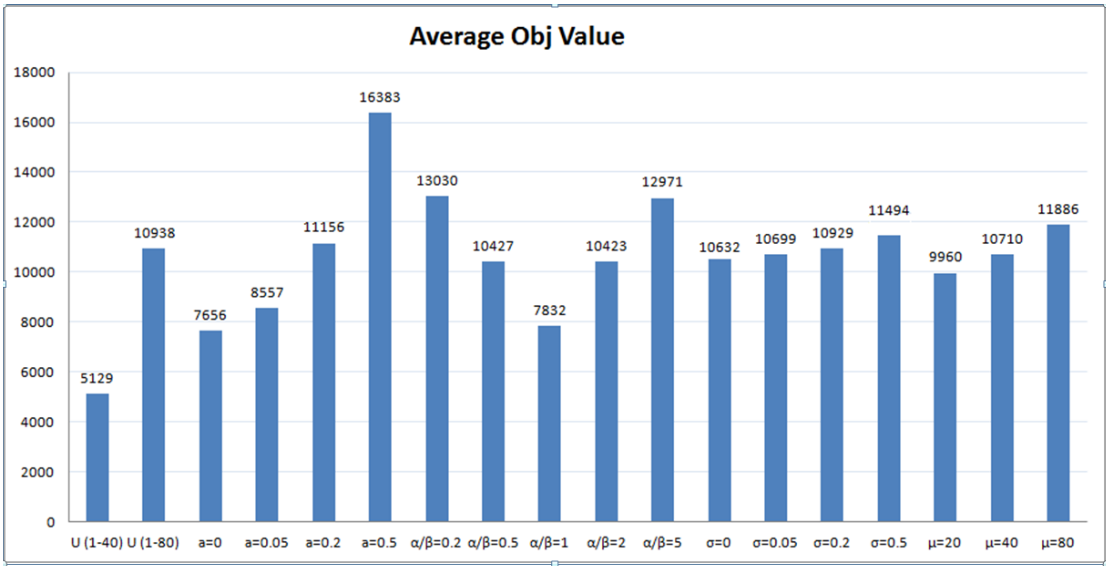

Figure 6. Average objective value for the multiple maintenance problems

We also analyze maintenance positions using Anova given in Table 11.

Table 11. Analysis of Variance for Maintenance Positions

	First Maintenance (q1)					
Source	**DF**	**SS**	**MS**	**F**	**P**	
U	2	210011	93456	1234,1	0	<0.05
a	3	313388	104463	2836,1	0	<0.05
μ	3	14611	4870	132,23	0	<0.05
σ	3	715	238	6,47	0	<0.05
α/β	4	10978	2745	74,51	0	<0.05
Error	3186	117350	37			
Total	3199	457043				

S = 6,06903   R-Sq = 74,32%   R-Sq(adj) = 74,22%

	Second Maintenance (q_2)					
Source	**DF**	**SS**	**MS**	**F**	**P**	
U	2	210134	93144	1134,1	0	<0.05
a	3	143231	47744	2034,02	0	<0.05
μ	3	10759	3586	152,78	0	<0.05

σ	3	38	13	0,54	0,656	
α/β	4	5390	1347	57,41	0	<0.05
Error	3186	74784	23			
Total	3199	234201				

S = 4,84485   R-Sq = 68,07%   R-Sq(adj) = 67,94%

	Third Maintenance q3					
Source	**DF**	**SS**	**MS**	**F**	**P**	
U	2	277691	91346	1554,1	0	<0.05
a	3	31398,5	10466,2	1106,31	0	<0.05
μ	3	4489,6	1496,5	158,19	0	<0.05
σ	3	17	5,7	0,6	0,616	
α/β	4	271,4	67,9	7,17	0	<0.05
Error	3186	30141	9,5			
Total	3199	66317,6				

S = 3,07579   R-Sq = 54,55%   R-Sq(adj) = 54,37%

ANOVA results indicate that the position of the first maintenance (q_1) is significantly affected by all four parameters. On the other hand, positions of the second and third maintenances (q_2 and q_3, respectively) are affected by only a, μ and α/β, and not by σ. That is maintenance deterioration has no significant effect on the positions of the second and third maintenances in our problem set. In fact, this is quite intuitive for the small sized problem (n=30) at hand, in which there are not many jobs left after the first maintenance. In most instances, no considerable deterioration effect occurs in the remaining few positions after the first maintenance. In larger problems, possibly there would be more jobs scheduled after the second and third maintenance, and deterioration then might have a significant effect on maintenance positions.

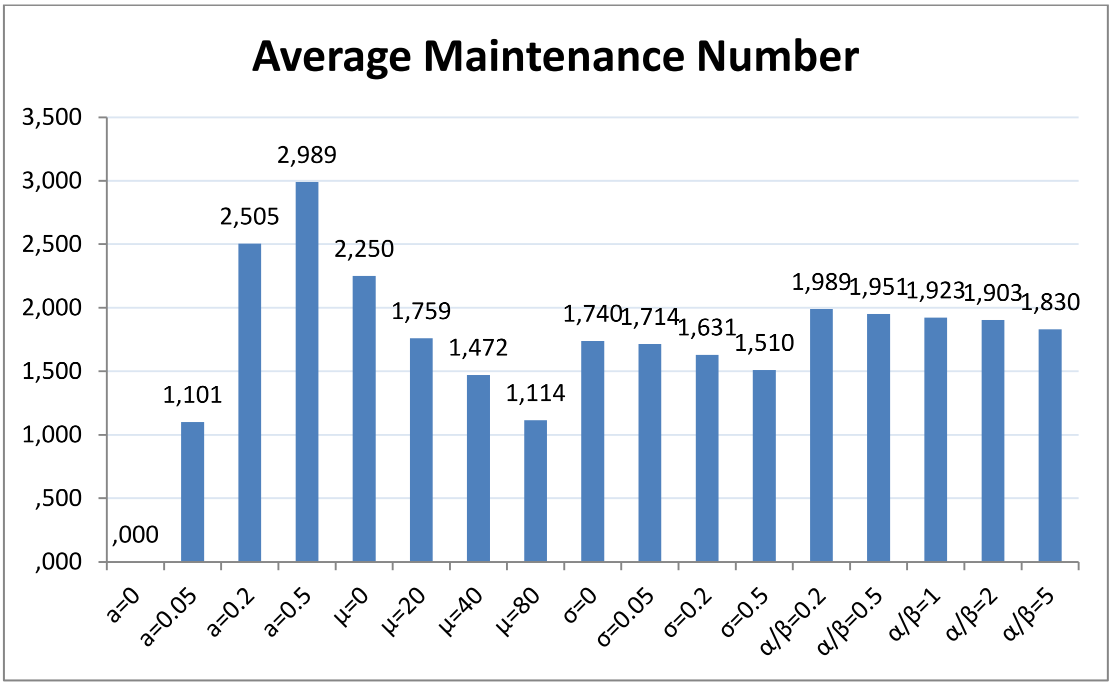

Figure 7. Average number of maintenance activities in the optimum schedule for different factor levels

As seen on Figure 7, all factors affect the average number of maintenance activities in the optimum solution. When job deterioration is higher (less), maintenance is more (less) desirable. On the other hand, when basic time of maintenance or its deterioration factor get larger (smaller), maintenance is less (more) desirable, because it becomes more (less) costly due to its duration. The α/β ratio has an effect on maintenance positions as well, if α/β ratio decreases, assigning jobs earlier becomes more desirable. Since there is a restricted time frame until due date, maintenance activity may be preferable to reduce job processing times. This is probably the reason why it is observed that when α/β ratio decreases, maintenance activity is more desirable.

Chapter 8

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this chapter, we provide a brief summary and suggest possible extensions for future research. We addressed the single machine early-tardy scheduling problem where both jobs and maintenance are deteriorating based on their position in the schedule. To the best of our knowledge, there is no previously reported study on our problem.

We proposed some properties of an optimum solution, then we defined and formulated the problem with a single maintenance activity as a mixed integer linear program. We developed a formulation for the general problem in which positions of both the jobs and the maintenance activity are to be determined in an optimum manner. Since this formulation is not fast enough, we developed another formulation.

In this second model, we considered a given position for the single maintenance activity and only scheduled the jobs. The second model is an iterative procedure of solving the problem for every possible maintenance position. The second formulation is faster than the first one although we solve the problem for every maintenance position. Still, mathematical formulation is not fast enough for big problems.

We identified two special cases of the problem, which are solvable in polynomial time. If the common due date is zero or smaller than the basic processing times of all jobs, the problem becomes the total completion time minimization which is polynomial time solvable (Yang-Yang 2010). Also, for unrestricted problems, we developed an algorithm, Algorithm 1 with $O(n \log n)$ complexity, that provides optimal results.

The restricted problems are NP-hard. Thus we proposed a heuristic algorithm (Algorithm 2) of $O(n^2)$ complexity as a heuristic to solve large instances for which the exact procedure fails to provide results in reasonable computational time. This algorithm exploits several properties of an optimum solution. Further, we proposed an improvement step (Algorithm 3), which takes in the maintenance position reported by Algorithm 2 and solves Model 2 to obtain a potentially better solution. Computational time of Algorithm 3 is relatively short for problems with up to 1000 jobs.

We also analyzed the multiple maintenance case by allowing upto $k_0 > 1$ maintenance activities within a schedule We generalized Model 2 to accommodate the multiple maintenance problems. Moreover, we adapted Algorithm 1 for the unrestricted problems with multiple maintenances. This modified algorithm provides optimum results with O $(n^{k_0+1} \log n)$ complexity.

Computational results show that the proposed algorithms perform well in terms of solution quality. For Algorithm 2, in the problem set, given in Appendix 4, average gap from optimum is 3.7% and worst case error is 11.66%. Algorithm 3 is almost optimal. With the same problem set, the average gap is 0.15% and the worst case error is 3%.

Without the improvement algorithm (i.e., Algorithm 3), results of Algorithm 2 are not very satisfactory, however even the worst case in Algorithm 2 with 11.66 % gap, becomes 1.5% with the help of improvement Algorithm 3.

Based on our experiments and Anova analysis for restricted problems, we observe that due date has a significant effect on the performance of our algorithm. The heuristics become more effective for problems with larger due dates even if they may still fall into the restrictive category.

We observed that the effects of the maintenance duration and maintenance deterioration are not very significant on our algorithm's performance and objective value. Other than maintenance related parameters, job processing times and its deterioration, early tardy costs and due date size also affect the algorithm performance and objective function value significantly.

After single maintenance problem computations, we also analyzed the multiple maintenance case and conducted an experiment to see in which cases performing multiple maintenance is desirable. We found average optimum maintenance numbers as a function of various factors. Due to the increased computational requirements in this case, we solved smaller sized problems (n=30). Our test results show that when job deterioration is larger (smaller), maintenance is more (less) preferable. When job deterioration is 0.5, it is observed that almost in every problem, maintenance activity is performed 3 times (average maintenance number is 2.99). On the other hand, when maintenance duration (basic time or deterioration) gets higher (smaller), maintenance is less (more) desirable, because it becomes more (less) costly causing delays in the schedule.

There are a few possible directions for future research. In this study we considered earliness and tardiness penalties that are job independent. A possible extension may be to study the case with job dependent early/tardy penalties. Other possible extensions may be to consider multiple due dates or due windows. Finally, consideration of other due date related objectives also offer a potential avenue for further studies.

BIBLIOGRAPHY

1. K.R. Baker. “Introduction to Sequencing and Scheduling“. *JohnWiley*, New York, (1974).
2. M. Feldmann, D. Biskup, “Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches“, *Computers and Industrial Engineering*, 44 (2003), 307-323.
3. N.G. Hall, M.E. Posner, “Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date“, *Operations Research*, 39 (1991), 836-846.
4. M.A. Kubzin, V.A. Strusevich, “Planning machine maintenance in two-machine shop scheduling“, *Operation Research*, 54 (2006), 789–800.
5. D. Biskup, “Single-machine scheduling with learning considerations“, *European Journal of Operational Research*, 115 (1999),173–8.
6. A. Bachman, A. Janiak. “ Scheduling jobs with position-dependent processing times“, *Journal of the Operational Research Society*, 55 (2004), 257–264
7. S.-J. Yang , D.-L. Yang, “Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities“, *Computers and Mathematics with Applications*, 60 (2010), 2161-2169.
8. J. J. Kanet, “Minimizing the average deviation of job completion times about a common date”, *Naval Research Logistics Quarterly*, 28 (1981a), 643-651.
9. S. S. Panwalkar, M. L. Smith, A Seidmann, “Common due date assignment to minimize total penalty for the one machine scheduling problem”, *Operations Research*, 30 (1982), 391-399.
10. M. van den Akker, H. Hoogeveen and S. van de Velde, “Combining column generation and Lagrangean relaxation to solve a single-machine common due date“, *INFORMS Journal on Computing*, 14 (2002), 37-51.
11. V. Gordon, J.-M. Proth, C. Chu. “A survey of the state-of-the-art of common due date assignment and scheduling research“, *European Journal of Operational Research*, 139 (2002), 1–25.
12. D. Biskup, M. Feldmann, “Benchmarks for scheduling on a single-machine against restrictive and unrestrictive common due dates“, *Computers and Operations Research*, 28 (2001), 787-801.
13. R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey“, *Ann. Discrete Math*., 4 (1979), 287-326.
14. M. Pinedo, “Scheduling: Theory, Algorithms, and Systems”, 2nd ed., *Prentice Hall*, New Jersey, (2002).

15. K.R.Baker, G.D.Scudder,"Sequencing with Earliness and Tardiness Penalties: A Review", *Operations Research*, 38 (1990), 22-36
16. P. S. Sundararaghavan, M. U. Ahmed, "Minimizing the sum of absolute lateness in single machine and multimachine scheduling", *Naval Research Logistics Quarterly*, 31 (1984), 325-333 (1984).
17. U. Bagchi, Y. L. Chang, R. S. Sullivan, "Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties and a common due date", *Naval Research Logistics*, 34 (1987), 739-751.
18. K. R. Baker, A. Chadowitz, "Algorithms for minimizing earliness and tardiness penalties with a common due date", *Working Paper, Amos Tuck School of Business Administration*, Dartmouth College, Hanover, N. H., (1989), 240.
19. U. Bagchi, R. S. Sullivan, L. Chang, "Minimizing mean absolute deviation of completion times about a common due date", *Naval Research Logistics*, 33(1986), 227-240.
20. W. Szwarc, "Single machine scheduling to minimize absolute deviation of completion times from a common due date", *Naval Research Logistics*, 36(1989), 663- 673.
21. NG. Hall, W. Kubiak, SP. Sethi, "Earliness-tardiness scheduling problems, II:Deviation of completion times about a restrictive common due date". *Operations Research*,39 (1991), 847-56.
22. J. A.Ventura, M.X. Weng, "An improved dynamic programming algorithm for the single machine mean absolute deviation problem with a restrictive common due date". *Operations Research Letters*, 17 (1995), 149-152.
23. J. Hoogeveen, H. Oosterhout, S.L. Van de Velde, "New lower and upper bounds for scheduling around a small common due date". *Operations Research*, 42 (1994), 102-110.
24. H. Emmons, "Scheduling to a common due date on parallel uniform processors", *Naval Research Logistics*, 34 (1987): 803-810.
25. C.Y Lee, S.D. Liman, "Single machine flow-time scheduling with scheduled maintenance", *Acta Informatica*, 29 (1992), 375–382.
26. R.J.W. James, "Using tabu search to solve the common due date early/tardy machine scheduling problem", *Computers and Operations Research*, 24 (1997), 199–208.
27. C.Y. Lee, S.J. Kim, "Parallel genetic algorithm for the earliness-tardiness job scheduling problem with general penalty weights", *Computers and Industrial Engineering*, 28 (1995), 231–243.
28. C.M. Hino, D.P. Ronconi, A.B. Mendes, "Minimizing earliness and tardiness penalties in a single-machine problem with a common due date", *European Journal of Operational Research*, 160 (2005), 190–201.
29. C.J. Liao, C.C. Cheng, "A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date", *Computers and Industrial Engineering*, 52 (2007), 404–413.

30. S. D. Liman, S. S. Panwalkar, S. Thongmee, “Determination of common due window location in a single machine scheduling problem”, *European Journal of Operational Research,* 93 (1996) 68-74.
31. U. Bagchi, R. S. Sullivan, L. Chang, “Minimizing mean squared deviation of completion times about a common due date”, *Management Science*, 33 (1987), 894-906.
32. S. Eilon, I. Chowdhury, “Minimizing waiting time variance in the single machine problem“, *Management Science*, 23 (1977), 567-575.
33. J. J. Kanet, “Minimizing variation of flow time in single machine systems“, *Management Science*, 27(1981b), 1453-1459.
34. K. Kianfar, G. Moslehi, “A branch-and-bound algorithm for single machine scheduling with quadratic earliness and tardiness penalties“, *Computers & OR,* 39 (2012), 2978-2990.
35. M. R. Garey, R. E. Tarjan, G. T.Wilfong, “One processor scheduling with symmetric earliness and tardiness penalties”, *Mathematics of Operations Research*, 13 (1988), 330-348.
36. A. Seidmann, S. S. Panwalkar, M. L. Smith, “Optimal assignment of due dates for a single processor scheduling problem”, *International Journal of Production Research*, 19 (1981), 393-399.
37. C. A. Yano, Y. D. Kim, “Algorithms for single machine scheduling problems minimizing tardiness and earliness”, *Technical Report,* Department *of Industrial Engineering, University of Michigan*, Ann Arbor, 86 (1986).
38. T. Abdul-Razaq, C. Potts, “Dynamic programming state-space relaxation for single-machine scheduling”, *Journal of Operational Research Society*, 39 (1988), 141-152.
39. P. S. Ow, T. E. Morton, “Filtered beam search in scheduling”, *International Journal of Production Research*, 26(1988): 35-62.
40. P. S. Ow, T. E. Morton, “The single machine early/tardy problem“, *Management Science*, 35 (1989), 177-191.
41. T.D. Fry, R.D. Armstrong, J.H. Blackstone, “Minimizing weighted absolute deviation in single machine scheduling”, *IIE Transactions*, 19 (1987), 445-450.
42. C.A. Yano, Y.D. Kim, “Algorithms for a class of single machine tardiness and earliness problems“, *European Journal of Operational Research*, 52 (1991), 167-178.
43. W. Szwarc, S. K. Mukhopadhyay, “Optimal timing schedules in earliness tardiness single machine sequencing”, *Naval Research Logistics*, 42 (1995), 1109-1114.
44. J.M.S. Valente, R.A.F.S. Alves, “Filtered and recovering beam search algorithms for early/tardy scheduling problem with no idle time”, *Computers and Industrial Engineering*, 48 (2005), 363-375.
45. S. Gupta, T. Sen, “Minimizing a quadratic function of job lateness on a single machine”, *Engineering Costs Production Economics*, 7 (1983), 181-194.

46. C.Y. Lee, “Machine scheduling with an availability constraint“, *Journal of Global Optimization*, 8 (1996), 395–417

47. E. Sanlaville, G. Schmidt , “Machine scheduling with availability constraints“, *Acta Informatica* 35 (1988), 795–811.

48. G. Schmidt, “Scheduling with limited machine availability“, *European Journal of Operational Research*, 121 (2000), 1–15.

49. Y. Ma, C .Chu, C. Zuo, “A survey of scheduling with deterministic machine availability constraints“. *Computers & Industrial Engineering*, 58 (2010), 199-211.

50. C.Y. Lee, L. Lei, M. Pinedo, “Current trend in deterministic scheduling“, *Annals of Operations Research*, 70 (1997), 1-42.

51. G. Mosheiov, J.B. Sidney, “Scheduling a deteriorating maintenance activity on a single machine“, *Journal of the Operational Research Society,* 61 (2010), 882–887.

52. W.C. Lee and Z.-S. Lu, “Group scheduling with deteriorating jobs to minimize the total weighted number of late jobs,” *Applied Mathematics and Computation*, vol. 218, 17 (2012), 8750–8757, 2012.

53. Li, Y. Q., Li, G., Sun, L. Y., Xu, Z. Y., “Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times“, *International Journal of Production Economics*, 118 (2009), 424–429.

54. G. Mosheiov, A. Sarig, Scheduling a maintenance activity and due-window assignment on a single machine. *Computers & Operational Research*, 36 (2009), 2541–2545.

55. J.N.D. Gupta, S.K. Gupta,“Single facility scheduling with nonlinear processing times“, *Computers and Industrial Engineering*, 14 (1988), 387–393.

56. A.S. Kunnathur, S.K. Gupta, “Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem“, *European Journal of Operational Research*, 47 (1990), 56–64.

57. G. Mosheiov, “ ʌ-shaped policies to schedule deteriorating jobs“, *Journal of Operational Reearch Society*, 47 (1996),1184–1191.

58. B. Alidaee, N.K. Womer, “Scheduling with time dependent processing times: review and extensions“, *Journal of Operational Reearch Society*, 50 (1999), 711–720.

59. T.C.E. Cheng, O. Ding, B.M.T. Lin, “A concise survey of scheduling with time-dependent processing times“, *European Journal of Operational Research*, 152 (2004), 1–13.

60. S. Gawiejnowicz, “Time-dependent Scheduling“, *Springer-Verlag Inc.*, New York, 2008.

61. S. Gawiejnowicz, “A note on scheduling on a single processor with speed dependent on a number of executed jobs“, *Information Processing Letters*, 57 (1996), 297-300.

62. S. Browne, U. Yechiali, “Scheduling deteriorating jobs on a single processor, *Operations Research*, 38 (3), (1990), 495–498.

63. C.-C.Wu, W.-C. Lee, “Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine“, *Information Processing Letters,* 87 (2003), 89-93.
64. M. Ji, Y. He, T.C.E. Cheng, “Scheduling linear deteriorating jobs with an availability constraint on a single machine“, *Theoretical Computer Science*, 362 (2006), 115–126
65. C.Y. Low, C.-J. Hsu, C.-T. Su, “Minimizing the makespan with an availability constraint on a single machine under simple linear deterioration“, *Computers and Mathematics with Applications*, 56 (2008), 257–265
66. E.J. Lodree Jr., C.D. Geiger, “A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration“, *European Journal of Operational Research*, 201 (2010), 644–648.
67. C.C. Wu, W.C. Lee, “Single-machine group-scheduling problems with deteriorating setup times and job-processing times“, *International Journal of Production Economics*, 115 (2008), 128–13.
68. J.B. Wang, M.Z. Wang, “Single-machine scheduling with nonlinear deterioration“, *Optimization Letters*, 6 (2012), 87–98.
69. W.-H. Kuo, D.-L. Yang, “Minimizing the makespan in a single machine scheduling problem with the cyclic process of an aging effect“, *Journal of the Operational Research Society*, 59 (2008), 416–420.
70. C.-L. Zhao, H.-Y. Tang, “Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan“, *Applied Mathematical Modelling*, 34 (2010), 837–841
71. A. Rudek, R. Rudek, “ A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models“, *Computers and Mathematics with Applications*, 62 (2011), 1870–1878
72. G. Mosheiov, “Scheduling jobs under simple linear deterioration“, *Computers and Operations Research*, 21 (6) (1994), pp. 653–659
73. J.-B. Wang, L. Lin, and F. Shan, “Single-machine group scheduling problems with deteriorating jobs,” *International Journal of Advanced Manufacturing Technology,* vol. 39, no 7-8 (2008), 808–812.
74. A. Bachman, A. Janiak, M.Y. Kovalyov, “Minimizing the total weighted completion time of deteriorating job“s, *Information Processing Letters*, 81 (2) (2002), 81–84.
75. C.C. Wu, W.C. Lee, Y.R. Shiau, “Minimizing the total weighted completion time on a single machine under linear deterioration“, *International Journal of Advanced Manufacturing Technology*, 33 (11–12) (2007), 1237–1243
76. J.B. Wang, C.T. Ng, T.C.E. Cheng, L.L. Liu, “Minimizing total completion time in a two-machine flow shop with deteriorating jobs“, *Applied Mathematics and Computation*, 180 (2006), 185–193

77. He, C. C., Wu, C. C., & Lee, W. C., “Branch-and-bound and weight-combination search algorithms for the total completion time problem with stepdeteriorating jobs“. *Journal of the Operational Research Society*, 60 (2009), 1759–1766.
78. S.-J. Yang, “Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deteriorating maintenance consideration“, *Applied Mathematics and Computation*, 217 (2010) 3321–3329.
79. J.B. Wang, Z.Q. Xia, “Scheduling jobs under decreasing linear deterioration“, *Information Processing Letters*, 94 (2) (2005), 63–69.
80. C,C.Wu,W.C.Lee. “Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine“, *Information Processing Letters,* 87 (2003)89-93.
81. Ji-Bo Wang, Cai Min Wei, “Parallel machine scheduling with a deteriorating maintenance activity and total absolute differences penalties“, *Applied Mathematics and Computation*, 217 (2011), 8093-8099.
82. S.J. Yang, D. L. Yang, T.C.E. Cheng, “Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance“, *Computers and Operations Research*, 37 (2010) 1510–1514.
83. S.J. Yang, D.L. Yang, “Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities“, *Omega*, 38 (2010b) 528–533.
84. T.C.E. Cheng, L.Y. Kang, C.T. Ng, “Due-date assignment and single machine scheduling with deteriorating jobs“, *Journal of the Operational Research Soci*ety, 55 (2) (2004), 198–203.
85. T.C.E. Cheng, L.Y. Kang, C.T. Ng, “Single machine due-date scheduling of jobs with decreasing start-time dependent processing times“, *International Transactions in Operational Research*, 12 (3) (2005), 355–366
86. T.C.E. Cheng, S.-J. Yang, D.-L. Yang, “Common due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity“, *International Journal of Production Economics*, 135 (2012),154-161.
87. G. Mosheiov, “A note on scheduling deteriorating jobs“, *Mathematical and Computer Modelling*, 41 (8-9) (2005), 883-886.
88. Hardy, G.H., Littlewood, J.E., Polya, G. , “Inequalities“, *Cambridge University Press*, London, (1967).

Appendices

Appendix 1. Algorithm 1

Example: α=1, β=2 and n=6 $p_j = \{10,20,30,40,50,60\}$ a=σ=0.5, μ=5, d=250

Solution:

$$W_r = \begin{pmatrix} \alpha(r-1)r^{\mathrm{a}} & r \le q \; and \; C_r \le d \\ \alpha(r-1)(r-q)^{\mathrm{a}} & r > q \; and \; C_r \le d \\ \beta(n-r+1)r^{\mathrm{a}} & r \le q \; and \; d < C_r \\ \beta(n-r+1)(r-q)^{\mathrm{a}} & r > q \; and \; d < C_r \end{pmatrix}$$

Step 0: Set q=1, TC^*=1.000.000

Step 1: Due date coincides with the completion time of a job which is scheduled at position: r=6*2/3=4. So when we get the sequences of jobs we will use this information to find starting time of the schedule.

Step 2: Calculate positional weights:

$W_1 = \alpha(r-1)r^{\mathrm{a}} = 1 * 0 * 1 = 0$

$W_2 = \alpha(r-1)(r-q)^{\mathrm{a}} = 1 * 1 * 1 = 1$

$W_3 = \alpha(r-1)(r-q)^{\mathrm{a}} = 1 * 2 * 2^{0,5} = 2,83$

$W_4 = \alpha(r-1)(r-q)^{\mathrm{a}} = 1 * 3 * 3^{0,5} = 5,20$

$W_5 = \beta(n-r+1)(r-q)^{\mathrm{a}} = 2 * 2 * 4^{0,5} = 8$

$W_6 = \beta(n-r+1)(r-q)^{\mathrm{a}} = 2 * 1 * 5^{0,5} = 4,47$

Step 3: Assign longest weight to job with smallest processing time value.

α=1, β=2 and n=6 $p_j = \{10,20,30,40,50,60\}$ a=σ=0.5, μ=5, d=250

Weights are in non-increasing order → $w_j = \{ W_5, W_4, W_6, W_3, W_2, W_1\}$

So the schedule with q=1 is {60, 50, 40, 20, 10, 30}

Step 4: p_{jr} ={ 60, 70.7, 56.6, 28.3, 14.1, 42.4 } $f(q) = \mu \times q^{\sigma}$=5× $1^{0,5}$=5

Step 5:

	q=1					
R	P_j	P_{jr}	S_r	C_r	E_r	T_r
1	60,0	60,0	43,8	103,8	146,2	
2	50,0	50,0	108,8	158,8	91,2	
3	40,0	56,6	158,8	215,4	34,6	
4	20,0	34,6	215,4	250,0	0,0	
5	10,0	20,0	250,0	270,0		40,0
6	30,0	67,1	270,0	337,1		174,2
					TC(1)	486,2

TC*= 441,2 and q*=1

Step 6: q=2 and go to step 1. Repeating same steps we get following results:

$TC(2)$= 438.988 TC^*= 438.988 and q^*=2

$TC(3)$= 446.465

$TC(4)$= 467.771

$TC(5)$= 486.437

$TC(6)$= 565.687

And q^*=2 TC^*=438.988 and related information:

$W_1 = a(r-1)r^{\mathrm{a}} = 1 * 0 * 1 = 0$

$W_2 = a(r-1)(r)^{\mathrm{a}} = 1 * 1 * 2^{0,5} = 1{,}41$

$W_3 = a(r-1)(r-q)^{\mathrm{a}} = 1 * 2 * 1^{0,5} = 2$

$W_4 = a(r-1)(r-q)^{\mathrm{a}} = 1*3*2^{0,5} = 4{,}24$

$W_5 = \beta(n-r+1)(r-q)^{\mathrm{a}} = 2*2*3^{0,5} = 6{,}93$

$W_6 = \beta(n-r+1)(r-q)^{\mathrm{a}} = 2*1*4^{0,5} = 4$

	q=2					
R	P_j	P_{jr}	S_r	C_r	E_r	T_r
1	60,0	60,0	43,9	103,9	146,1	0
2	50,0	70,7	103,9	174,6	75,4	0
3	40,0	40,0	181,7	221,7	28,3	0
4	20,0	28,3	221,7	250,0	0	0
5	10,0	17,3	250,0	267,3	0	34,6
6	30,0	60,0	267,3	327,3	0	154,6
					TC(1)	439

Step 7: $TC(2)$= 438.988 TC^*= 438.988 and q^*=2

Appendix 2. Algorithm 2

$J = \{ 10,20,30,40\}$ d=40, $a=\sigma$=0.5, μ=5.

Step 0: $J = \{ 40,30,20,10 \}$

Step 1: l=1

Step 2: q=0, i=1, l=1, $r_{min} = 1$, $r_{max} = 4$,

Step 3: q=1, $\mathrm{f}(q) = \mu \times q^{\sigma}$ → f(1)=5=5

	Pjr (q=1)			
	r=1	r=2	r=3	r=4
j=1	10	10	14,14	17,32
j=2	20	20	28,28	34,64
j=3	30	30	42,43	51,96
j=4	40	40	56,57	69,28

$EM(1) = f(1) + \sum_j^n \sum_r^n p_{jr} /n$

EM(1)= 514,62/4 + 5=128,656+5=133,656

EP=d- $f(1)$=40-5=35

LP=$EM(1)$-d=133,656-40=93,656

$A = R = \frac{EP}{LP} = 0{,}3737$, X=35, Y=93,656 go to Step 5.

Step 5: $x > 0$ and $R = A$ then apply step 6.

Step 6: $r_{\min}$ =1 first position is assigned to smallest indexed job (p_j=40)

And delete this job from the set $J = \{ 30,20,10\}$.

$r_{min} = r_{min} + 1 = 2$, i=i+1=2, X=X-$p_{j,r_{\min}} = 35 - 40 = -5$

$R = \frac{X}{Y} = -\frac{5}{93{,}656} = -0{,}05$. Go to Step 8.

Step 8: Since $i = 2 \leq n = 4$ go to Step 5.

Step 5: X<0 then all the rest of the jobs should be assigned after due-date. Means step 6 is no longer necessary, we can continue on assigning jobs with step 7.

Step 7: Assign the smallest indexed job in set $J = \{30,20,10\}$ to r_{max} and delete it from set J. r_{max} =4 last position is assigned to smallest indexed job (p_j=30)

And delete this job from the set $J = \{20,10\}$.

$r_{max} = r_{max} - 1 = 3$, i=i+1=3, $p_{j,r_{max}} = 30 * 3^{0,5} = 51,96$ (q=1 then this is the third job (r-q=4-1) after maintenance this is why we take power as 3)

Y=Y-$p_{j,r_{max}} = 93,656 - 51,96 = 41,7$, $R = \frac{X}{Y} = -\frac{5}{41,7} = -0,11$.

Step 8: Since $i = 3 \leq n = 4$ go to Step 5.

Step 5: X<0 go to step 7.

Step 7: Assign the smallest indexed job in set $J = \{20,10\}$ to r_{max} and delete it from set J. r_{max} =3 third position is assigned to smallest indexed job (p_j=20)

And delete this job from the set $J = \{20,10\}$.

$r_{max} = r_{max} - 1 = 2$, i=i+1=4, $p_{j,r_{max}} = 20 * 2^{0,5} = 28,28$, $J = \{10\}$.

Y=Y-$p_{j,r_{max}} = 41,7 - 28,28 = 13,41$, $R = \frac{X}{Y} = -\frac{5}{13,41} = -0.37$.

Step 8: Since $i = 4 \leq n = 4$ go to step 5.

Step 5: x<0 go to step 7.

Step 7: Assign the smallest indexed job in set $J = \{10\}$ to r_{max} and delete it from set J. r_{max} =2, second position is assigned to smallest indexed job (pj=10)

And delete this job from the set $J = \{10\}$. Then $J = \emptyset$, all jobs are assigned.

$r_{max} = r_{max} - 1 = 2$, i=i+1=5, $p_{j,r_{max}} = 10 * 1^{0,5} = 10$

Y=Y-$p_{j,\mathrm{r}_{\max}} = 13{,}41 - 10 = 3{,}41,\ R = \frac{X}{Y} = -\frac{5}{3{,}41} = -1{,}47$. Go to step 7.

Step 8: i $= 5 > n = 4$ go to step 9.

Step 9: Calculate Z(S(q)).

S(1)={40,10,20,30) and their finishing times={40, 55, 83.28, 135.24}

So there is not any early job and total tardiness cost is=307 now Z(S(1))=307

Z(S^*)=307 q*=1

Let us try *q*=*q*+1=2 go to step 2.

All previous steps are applied and Z(S(2))=278 and we update Z(S^*)=278 q*=2

Then *q*=3 Z(S(3))=307, Z(S^*)=278 *q**=2

q=4 Z(S(4))=343,4, Z(S^*)=278 q*=2

Step 10: $\mathrm{l} = 1 \leq 1$, set *l*=2 go to step 2.

Step 2: *q=0, i=1, l=1,* $\mathrm{r}_{\min} = 1$, $\mathrm{r}_{\max} = \mathrm{n}$, go to step 4.

Step 4: q=q+1=1, EP=d=40, LP=$EM(q)$-*d*- $f(q)$= 133,6566-40-5=88.6566,

$A = R = \frac{EP}{LP}$=0,45, *X*=*EP*, *Y*=*LP*

Applying rest of the algorithm like we did for *l*=1 case we get following results:

Z(*S*(1))=307 *Z*(S^*)=278 *q**=2

Z(*S*(2))=278 and we update *Z*(S^*)=278 *q**=2

Z(*S*(3))=307, *Z*(S^*)=278 *q**=2

q=4 *Z*(*S*(4))=343,4, *Z*(S^*)=278 *q**=2

Step 9: $l = 2 > 1$ stop. Best q value *q**=2. So the best result is reached when maintenance is done after second job (before third job). Then for better result apply mathematical formulation with q=2. Result of mathematical formulation is 256 which is less than heuristic result (278).

Then we apply mathematical formulation's result which has that sequence in basic processing times: {30, 10, maintenance, 20, 40}. If we try all the available maintenance positions in formulation we can find the optimal result is 256 as it is seen on following table.

	Heuristic	**Mathematical Formulation**
***q*=1**	307	291,7
***q*=2**	278	256,3
***q*=3**	307	270,7
***q*=4**	343	333,4

This is a small problem this is why results didn't change over different "l" values. And we found optimal solution with the help of heuristic, which gave us best maintenance position. For small problems it is easy to try all of them in mathematical formulation, but in big-sized problems it has big importance, since it is almost impossible to try all cases.

Appendix 3. p_j Values of the Single Maintenance Test Set

	U **(1-80)**										***U*** **(1-40)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
2	3	1	1	2	1	1	3	3	2	2	2	1	1	1	1	1	2	2	1	1
3	3	2	2	2	1	1	3	3	4	2	2	1	1	1	1	1	2	2	2	1
4	4	2	2	2	2	1	4	6	4	2	2	1	1	1	1	1	2	3	2	1
5	5	3	2	3	3	2	4	7	5	2	3	2	1	2	2	1	2	4	3	1
6	6	3	3	4	4	3	4	7	6	4	3	2	2	2	2	2	2	4	3	2
7	6	4	4	4	5	3	5	7	6	5	3	2	2	2	3	2	3	4	3	3
8	6	5	4	6	6	4	5	8	7	6	3	3	2	3	3	2	3	4	4	3
9	7	5	5	6	6	6	6	8	7	7	4	3	3	3	3	3	3	4	4	4
10	9	6	5	7	6	7	8	9	7	7	5	3	3	4	3	4	4	5	4	4
11	9	7	6	7	12	7	9	9	7	8	5	4	3	4	6	4	5	5	4	4
12	10	7	8	7	13	7	9	10	9	9	5	4	4	4	7	4	5	5	5	5
13	10	9	9	10	14	8	9	10	9	10	5	5	5	5	7	4	5	5	5	5
14	11	9	9	11	14	10	9	11	10	10	6	5	5	6	7	5	5	6	5	5
15	11	10	9	12	17	12	10	11	10	10	6	5	5	6	9	6	5	6	5	5
16	11	10	10	12	17	13	10	11	11	12	6	5	5	6	9	7	5	6	6	6
17	11	11	11	13	18	14	12	11	12	14	6	6	6	7	9	7	6	6	6	7
18	13	11	12	13	18	14	12	12	12	16	7	6	6	7	9	7	6	6	6	8
19	13	13	14	13	20	14	12	13	12	17	7	7	7	7	10	7	6	7	6	9
20	13	13	16	13	21	14	13	14	13	18	7	7	8	7	11	7	7	7	7	9
21	13	13	18	14	21	16	13	15	15	18	7	7	9	7	11	8	7	8	8	9
22	14	13	18	15	22	16	15	17	18	19	7	7	9	8	11	8	8	9	9	10
23	14	13	19	15	22	16	17	18	20	20	7	7	10	8	11	8	9	9	10	10
24	15	14	22	16	23	20	18	18	20	21	8	7	11	8	12	10	9	9	10	11
25	15	15	23	18	24	21	19	18	20	21	8	8	12	9	12	11	10	9	10	11
26	15	15	23	18	24	22	20	21	22	22	8	8	12	9	12	11	10	11	11	11
27	17	15	24	19	26	25	20	21	22	23	9	8	12	10	13	13	10	11	11	12
28	18	15	26	21	26	27	21	21	26	25	9	8	13	11	13	14	11	11	13	13
29	19	16	26	22	26	28	21	22	26	26	10	8	13	11	13	14	11	11	13	13
30	20	16	27	23	27	29	21	23	27	27	10	8	14	12	14	15	11	12	14	14
31	20	16	28	25	27	30	22	23	27	27	10	8	14	13	14	15	11	12	14	14
32	21	16	28	25	28	30	23	25	27	27	11	8	14	13	14	15	12	13	14	14
33	22	17	28	25	28	32	24	25	30	28	11	9	14	13	14	16	12	13	15	14
34	22	19	28	26	29	33	24	26	30	28	11	10	14	13	15	17	12	13	15	14
35	23	19	30	26	30	34	26	27	31	29	12	10	15	13	15	17	13	14	16	15
36	23	20	30	27	30	34	27	27	31	29	12	10	15	14	15	17	14	14	16	15
37	24	21	32	27	31	34	27	27	31	29	12	11	16	14	16	17	14	14	16	15
38	24	22	33	27	31	34	28	28	32	30	12	11	17	14	16	17	14	14	16	15
39	25	23	34	27	32	35	28	28	32	30	13	12	17	14	16	18	14	14	16	15
40	25	24	34	28	32	36	29	28	33	32	13	12	17	14	16	18	15	14	17	16
41	25	24	34	30	32	36	29	29	34	34	13	12	17	15	16	18	15	15	17	17
42	26	26	36	30	34	37	29	29	35	36	13	13	18	15	17	19	15	15	18	18
43	26	27	37	33	34	37	29	29	35	36	13	14	19	17	17	19	15	15	18	18
44	26	30	37	33	35	38	31	32	36	37	13	15	19	17	18	19	16	16	18	19
45	27	30	38	34	35	39	31	32	36	37	14	15	19	17	18	20	16	16	18	19
46	28	31	38	34	35	40	32	32	38	37	14	16	19	17	18	20	16	16	19	19
47	28	31	39	35	36	41	32	33	38	37	14	16	20	18	18	21	16	17	19	19
48	29	32	39	36	37	41	33	33	40	38	15	16	20	18	19	21	17	17	20	19
49	29	32	40	36	37	43	34	35	41	39	15	16	20	18	19	22	17	18	21	20
50	29	32	41	38	38	45	36	35	43	39	15	16	21	19	19	23	18	18	22	20
51	30	32	43	41	38	45	37	36	44	41	15	16	22	21	19	23	19	18	22	21
52	30	33	46	41	41	45	37	36	44	41	15	17	23	21	21	23	19	18	22	21

53	31	33	47	42	41	46	37	38	45	41	16	17	24	21	21	23	19	19	23	21
54	32	34	48	42	41	46	37	38	46	41	16	17	24	21	21	23	19	19	23	21
55	32	35	49	42	42	47	38	38	47	42	16	18	25	21	21	24	19	19	24	21
56	34	37	50	44	43	47	38	41	47	42	17	19	25	22	22	24	19	21	24	21
57	35	37	51	45	44	47	39	44	47	42	18	19	26	23	22	24	20	22	24	21
58	38	40	52	45	44	47	39	45	50	43	19	20	26	23	22	24	20	23	25	22
59	40	40	52	45	44	48	39	45	50	43	20	20	26	23	22	24	20	23	25	22
60	41	41	52	46	44	49	41	46	54	44	21	21	26	23	22	25	21	23	27	22
61	41	41	54	46	45	49	41	47	54	46	21	21	27	23	23	25	21	24	27	23
62	43	42	55	47	46	49	42	47	55	50	22	21	28	24	23	25	21	24	28	25
63	44	42	56	47	46	50	44	47	55	52	22	21	28	24	23	25	22	24	28	26
64	45	43	57	47	46	50	46	48	55	52	23	22	29	24	23	25	23	24	28	26
65	45	44	57	48	47	50	47	49	55	53	23	22	29	24	24	25	24	25	28	27
66	47	44	57	48	48	50	48	49	56	54	24	22	29	24	24	25	24	25	28	27
67	47	44	58	49	48	51	48	49	57	54	24	22	29	25	24	26	24	25	29	27
68	47	45	58	49	48	52	51	51	58	55	24	23	29	25	24	26	26	26	29	28
69	48	48	60	50	50	53	51	51	58	57	24	24	30	25	25	27	26	26	29	29
70	48	48	61	51	51	55	53	51	58	57	24	24	31	26	26	28	27	26	29	29
71	50	48	61	53	51	55	53	52	60	58	25	24	31	27	26	28	27	26	30	29
72	51	48	62	54	53	56	57	52	60	59	26	24	31	27	27	28	29	26	30	30
73	51	49	62	54	55	59	58	53	61	61	26	25	31	27	28	30	29	27	31	31
74	52	49	64	55	56	59	59	55	61	61	26	25	32	28	28	30	30	28	31	31
75	52	50	65	55	57	59	61	57	62	64	26	25	33	28	29	30	31	29	31	32
76	52	51	65	55	59	61	62	58	63	64	26	26	33	28	30	31	31	29	32	32
77	53	52	66	55	60	62	65	59	64	64	27	26	33	28	30	31	33	30	32	32
78	54	52	66	55	61	64	66	60	64	67	27	26	33	28	31	32	33	30	32	34
79	54	53	66	57	61	65	67	61	64	68	27	27	33	29	31	33	34	31	32	34
80	58	55	68	58	62	66	67	63	65	69	29	28	34	29	31	33	34	32	33	35
81	59	55	69	58	62	66	68	65	68	70	30	28	35	29	31	33	34	33	34	35
82	60	56	69	58	62	66	68	65	69	70	30	28	35	29	31	33	34	33	35	35
83	61	56	70	59	64	67	69	65	69	70	31	28	35	30	32	34	35	33	35	35
84	61	59	72	60	65	67	70	65	70	70	31	30	36	30	33	34	35	33	35	35
85	61	59	72	61	66	68	70	67	70	72	31	30	36	31	33	34	35	34	35	36
86	62	62	73	65	67	69	70	68	71	73	31	31	37	33	34	35	35	34	36	37
87	62	63	73	65	70	70	71	70	71	73	31	32	37	33	35	35	36	35	36	37
88	63	63	73	66	70	71	71	70	71	73	32	32	37	33	35	36	36	35	36	37
89	63	64	73	67	71	71	71	71	72	73	32	32	37	34	36	36	36	36	36	37
90	64	65	73	67	72	72	73	72	74	74	32	33	37	34	36	36	37	36	37	37
91	67	67	74	68	73	72	74	72	75	74	34	34	37	34	37	36	37	36	38	37
92	68	68	75	69	75	73	74	73	75	74	34	34	38	35	38	37	37	37	38	37
93	69	75	75	71	75	74	74	73	77	75	35	38	38	36	38	37	37	37	39	38
94	70	75	76	73	76	75	75	76	78	76	35	38	38	37	38	38	38	38	39	38
95	74	76	77	74	77	75	75	77	78	76	37	38	39	37	39	38	38	39	39	38
96	75	76	78	75	78	78	76	77	78	79	38	38	39	38	39	39	38	39	39	40
97	78	77	79	77	78	79	77	77	78	80	39	39	40	39	39	40	39	39	39	40
98	80	79	79	78	78	79	78	77	79	80	40	40	40	39	39	40	39	39	40	40
99	80	79	80	79	79	80	78	78	79	80	40	40	40	40	40	40	39	39	40	40
100	80	79	80	80	79	80	79	78	80	80	40	40	40	40	40	40	40	39	40	40

Appendix 4. The Single Maintenance Test Set Results

							Algorithm 2			Algorithm 3					
	U	**a**	**α/β**	**σ**	**μ**	**d**	**AVG GAP**	**AVG CPU**	**AVG OBJ Value**	**AVG GAP**	**Heuristic AVG CPU (sec)**	**Math. Model + Algorithm 2 (sec)**	**Math. Model AVG CPU (sec)**	**AVG OBJ Value**	**AVG. Opt. Obj. Value**
1	1-40	0.05	1	0.05	20	200	4,14	02:43	79858	0,16	02:43	106	103	63645	63575
2	1-40	0.05	1	0.05	20	1000	0,75	02:44	51104	0,05	02:44	108	105	39690	39668
3	1-40	0.05	1	0.05	40	200	3,97	02:42	80969	0,02	02:42	127	124	64750	64721
4	1-40	0.05	1	0.05	40	1000	0,76	02:46	51882	0,11	02:46	111	108	40433	40361
5	1-40	0.05	1	0.2	20	200	4,03	02:41	80789	0,06	02:41	127	124	64561	64525
6	1-40	0.05	1	0.2	20	1000	1,11	02:45	51822	0,55	02:45	118	115	40364	40201
7	1-40	0.05	1	0.2	40	200	3,77	02:46	81804	0	02:46	120	117	65410	65410
8	1-40	0.05	1	0.2	40	1000	0,46	02:57	52298	0	02:57	124	121	40496	40496
9	1-40	0.05	2	0.05	20	200	3,04	02:39	80362	0,14	02:39	120	117	64789	64730
10	1-40	0.05	2	0.05	20	1000	4,74	02:47	63962	0,32	02:47	109	106	51979	51882
11	1-40	0.05	2	0.05	40	200	2,92	02:34	81470	0,02	02:34	114	111	65886	65858
12	1-40	0.05	2	0.05	40	1000	3,63	02:50	64465	0,09	02:50	116	113	52808	52748
13	1-40	0.05	2	0.2	20	200	2,96	02:35	81290	0,05	02:35	110	107	65698	65668
14	1-40	0.05	2	0.2	20	1000	4,03	02:46	64483	0,02	02:46	106	103	52629	52616
15	1-40	0.05	2	0.2	40	200	2,73	02:31	82303	0	02:31	117	114	66545	66545
16	1-40	0.05	2	0.2	40	1000	3,42	02:43	64855	0,02	02:43	110	107	52908	52903
17	1-40	0.05	0.5	0.05	20	200	4,85	02:29	159196	0,16	02:29	119	116	125964	125823
18	1-40	0.05	0.5	0.05	20	1000	5,24	02:36	88960	0,39	02:36	109	106	62646	62343
19	1-40	0.05	0.5	0.05	40	200	4,67	02:30	161421	0,08	02:30	116	113	128201	128116
20	1-40	0.05	0.5	0.05	40	1000	4,77	02:36	90371	0,47	02:36	107	104	64272	63765
21	1-40	0.05	0.5	0.2	20	200	4,73	02:30	161060	0,06	02:30	119	116	127796	127724
22	1-40	0.05	0.5	0.2	20	1000	5,03	02:36	90276	0,79	02:36	120	117	64272	63607
23	1-40	0.05	0.5	0.2	40	200	4,47	02:31	163093	0	02:31	119	116	129494	129494
24	1-40	0.05	0.5	0.2	40	1000	4,00	02:38	91261	0	02:38	108	105	64272	64272
25	1-40	0.2	1	0.05	20	200	3,22	03:10	129033	0,05	03:10	107	104	104180	104144
26	1-40	0.2	1	0.05	20	1000	4,34	02:39	89344	0,47	02:39	106	103	67747	67426
27	1-40	0.2	1	0.05	40	200	3,16	03:07	130359	0,03	03:07	119	116	105513	105477
28	1-40	0.2	1	0.05	40	1000	4,49	02:39	90580	0,35	02:39	127	124	68792	68543
29	1-40	0.2	1	0.2	20	200	2,92	03:09	129823	0,05	03:09	126	123	105209	105171
30	1-40	0.2	1	0.2	20	1000	4,41	02:38	90331	0,36	02:38	115	112	68615	68344
31	1-40	0.2	1	0.2	40	200	3,09	03:10	132378	0,01	03:10	125	122	107548	107510
32	1-40	0.2	1	0.2	40	1000	4,62	02:39	92550	0,32	02:39	110	107	70653	70348
33	1-40	0.2	2	0.05	20	200	2,71	03:13	129452	0,02	03:13	119	116	105078	105060
34	1-40	0.2	2	0.05	20	1000	0,97	02:45	99232	0,02	02:45	110	107	80204	80177

35	1-40	0.2	2	0.05	40	200	2,67	03:11	130776	0,01	03:11	126	123	106403	106382
36	1-40	0.2	2	0.05	40	1000	0,98	02:44	100430	0,03	02:44	126	123	81370	81354
37	1-40	0.2	2	0.2	20	200	2,69	03:11	130477	0,02	03:11	114	111	106106	106083
38	1-40	0.2	2	0.2	20	1000	0,98	02:41	100196	0,03	02:41	120	117	81154	81129
39	1-40	0.2	2	0.2	40	200	2,62	03:15	132795	0	03:15	108	105	108423	108406
40	1-40	0.2	2	0.2	40	1000	0,99	03:12	102352	0,03	03:12	118	115	83240	83228
41	1-40	0.2	0.5	0.05	20	200	3,49	03:12	257645	0,08	03:12	118	115	207437	207335
42	1-40	0.2	0.5	0.05	20	1000	8,98	02:39	168534	1,36	02:39	118	115	121094	119432
43	1-40	0.2	0.5	0.05	40	200	3,42	03:08	260300	0,04	03:08	116	113	210113	210016
44	1-40	0.2	0.5	0.05	40	1000	9,16	02:46	171049	1,44	02:46	110	107	123210	121607
45	1-40	0.2	0.5	0.2	20	200	3,45	03:03	259700	0,06	03:03	123	120	209486	209397
46	1-40	0.2	0.5	0.2	20	1000	9,05	02:45	170530	1,51	02:45	122	119	122921	121232
47	1-40	0.2	0.5	0.2	40	200	3,34	02:52	264338	0,02	02:52	115	112	214199	214097
48	1-40	0.2	0.5	0.2	40	1000	9,31	02:35	175047	1,54	02:35	108	105	126881	125149

							Algorithm 2			Algorithm 3					
	U	***a***	***α/β***	***σ***	***μ***	***d***	**AVG GAP**	**AVG CPU**	**AVG OBJ Value**	**AVG GAP**	**Heuristic AVG CPU (sec)**	**Math. Model + Algorithm 2 (sec)**	**Math. Model AVG CPU (sec)**	**AVG OBJ Value**	**AVG. Opt. Obj. Value**
1	1-80	0.05	1	0.05	20	200	3,09	02:31	138199	0,02	02:31	111	108	136780	136441
2	1-80	0.05	1	0.05	20	1000	4,46	02:52	97279	0,08	02:52	121	118	95501	95111
3	1-80	0.05	1	0.05	40	200	3,03	02:29	139400	0,06	02:29	113	110	138075	137693
4	1-80	0.05	1	0.05	40	1000	4,59	02:49	98493	0,12	02:49	119	116	96591	96170
5	1-80	0.05	1	0.2	20	200	3,05	02:27	139173	0,05	02:27	109	106	137816	137441
6	1-80	0.05	1	0.2	20	1000	4,54	02:40	98273	0,14	02:40	106	103	96453	95998
7	1-80	0.05	1	0.2	40	200	2,97	02:28	141263	0,03	02:28	126	123	139954	139579
8	1-80	0.05	1	0.2	40	1000	4,65	02:55	100315	0,31	02:55	126	123	98373	97832
9	1-80	0.05	2	0.05	20	200	2,68	02:37	138394	0,02	02:37	109	106	137494	137160
10	1-80	0.05	2	0.05	20	1000	1,28	02:43	104567	0,01	02:43	110	107	105493	105236
11	1-80	0.05	2	0.05	40	200	2,62	02:46	139596	0,06	02:46	108	105	138789	138412
12	1-80	0.05	2	0.05	40	1000	1,37	02:44	105745	0,04	02:44	117	114	106613	106321
13	1-80	0.05	2	0.2	20	200	2,65	02:46	139368	0,05	02:46	117	114	138530	138160
14	1-80	0.05	2	0.2	20	1000	1,33	02:47	105535	0,04	02:47	127	124	106427	106145
15	1-80	0.05	2	0.2	40	200	2,57	02:50	141458	0,03	02:50	114	111	140654	140298
16	1-80	0.05	2	0.2	40	1000	1,47	02:49	107528	0,07	02:49	119	116	108284	107970
17	1-80	0.05	0.5	0.05	20	200	3,34	02:46	276202	0,02	02:46	110	107	272736	272048
18	1-80	0.05	0.5	0.05	20	1000	7,62	03:04	187269	0,12	03:04	118	115	178890	177965
19	1-80	0.05	0.5	0.05	40	200	3,27	02:44	278605	0,06	02:44	120	117	275317	274552
20	1-80	0.05	0.5	0.05	40	1000	7,70	03:05	189728	0,15	03:05	107	104	181103	180153
21	1-80	0.05	0.5	0.2	20	200	3,30	02:54	278150	0,05	02:54	117	114	274799	274048
22	1-80	0.05	0.5	0.2	20	1000	7,66	03:16	189279	0,17	03:16	127	124	180800	179776
23	1-80	0.05	0.5	0.2	40	200	3,21	02:49	282330	0,03	02:49	120	117	279076	278323
24	1-80	0.05	0.5	0.2	40	1000	7,61	03:15	193407	0,4	03:15	113	110	184913	183651
25	1-80	0.2	1	0.05	20	200	2,50	03:13	220587	0,06	03:13	125	122	219497	218945
26	1-80	0.2	1	0.05	20	1000	5,00	03:07	169351	0,04	03:07	121	118	165030	164613
27	1-80	0.2	1	0.05	40	200	2,48	03:11	221931	0,04	03:11	119	116	220802	220313
28	1-80	0.2	1	0.05	40	1000	4,97	03:01	170627	0,04	03:01	110	107	166308	165872
29	1-80	0.2	1	0.2	20	200	2,49	03:01	221620	0,05	03:01	110	107	220494	219986
30	1-80	0.2	1	0.2	20	1000	4,98	02:57	170353	0,04	02:57	108	105	166028	165606
31	1-80	0.2	1	0.2	40	200	2,45	03:08	223976	0,05	03:08	121	118	222925	222382
32	1-80	0.2	1	0.2	40	1000	4,97	04:26	172707	0,01	04:26	125	122	168238	167844
33	1-80	0.2	2	0.05	20	200	2,36	03:01	220758	0,05	03:01	120	117	219940	219414
34	1-80	0.2	2	0.05	20	1000	2,75	02:48	175096	0,03	02:48	126	123	174081	173745
35	1-80	0.2	2	0.05	40	200	2,34	02:54	222102	0,01	02:54	120	117	221248	220775

36	1-80	0.2	2	0.05	40	1000	2,74	02:57	176374	0,03	02:57	113	110	175360	175011
37	1-80	0.2	2	0.2	20	200	2,34	03:09	221790	0,04	03:09	116	113	220939	220452
38	1-80	0.2	2	0.2	20	1000	2,74	03:07	176099	0,03	03:07	112	109	175070	174741
39	1-80	0.2	2	0.2	40	200	2,32	03:02	224148	0,04	03:02	113	110	223364	222840
40	1-80	0.2	2	0.2	40	1000	2,75	03:00	178438	0,01	03:00	127	124	181845	176985
41	1-80	0.2	0.5	0.05	20	200	2,57	02:56	441002	0,06	02:56	115	112	438547	437416
42	1-80	0.2	0.5	0.05	20	1000	6,67	02:35	332956	0,03	02:35	108	105	319714	318770
43	1-80	0.2	0.5	0.05	40	200	2,55	03:02	443692	0,04	03:02	126	123	441156	440157
44	1-80	0.2	0.5	0.05	40	1000	6,63	03:06	335507	0,02	03:06	119	116	322288	321316
45	1-80	0.2	0.5	0.2	20	200	2,56	02:59	443069	0,05	02:59	117	114	440539	439500
46	1-80	0.2	0.5	0.2	20	1000	6,64	02:59	334959	0,02	02:59	113	110	321758	320770
47	1-80	0.2	0.5	0.2	40	200	2,52	03:04	447781	0,05	03:04	123	120	445407	444302
48	1-80	0.2	0.5	0.2	40	1000	6,60	02:53	339684	0,02	02:53	126	123	326216	325292

Appendix 5. p_j Values of Treatments for Multi Maintenance Tests

Jobs	Sets									
	1	**2**	**3**	**4**	**5**	6	7	8	9	10
1	6	3	1	1	4	1	1	2	1	2
2	11	4	3	5	9	3	2	3	4	2
3	11	7	6	5	9	7	2	4	5	6
4	13	9	6	6	10	7	3	10	5	8
5	13	9	11	10	13	10	5	14	11	9
6	14	10	11	11	16	11	9	24	12	9
7	18	13	13	13	16	13	13	28	16	18
8	19	15	14	16	17	13	15	30	18	23
9	21	22	15	19	19	14	16	40	19	28
10	24	23	15	22	20	15	21	41	22	28
11	25	23	17	23	24	15	27	43	23	30
12	25	24	20	24	26	15	27	47	26	32
13	28	26	20	30	30	32	28	48	26	33
14	29	28	22	30	31	35	32	50	34	34
15	29	34	25	30	41	37	32	51	36	34
16	35	38	26	31	41	40	37	52	37	37
17	41	41	26	32	42	40	44	54	38	38
18	45	44	27	32	48	43	45	57	39	39
19	47	45	29	33	48	44	46	57	49	52
20	48	48	31	33	48	44	48	57	56	52
21	52	51	32	34	49	50	49	58	58	55
22	54	51	40	42	52	56	52	60	61	61
23	54	52	43	47	53	56	62	62	64	65
24	61	52	47	51	55	59	65	68	65	70
25	63	53	50	59	55	59	67	69	66	72
26	67	58	60	62	63	62	75	73	66	73
27	68	61	61	74	63	64	76	73	66	73
28	69	62	63	75	75	68	77	75	69	73
29	80	64	75	76	76	79	79	78	72	74
30	80	70	78	80	79	79	80	79	80	77
Average	38	35	30	34	38	36	38	47	38	40