SCHEDULING WITH TOOL CHANGES TO MINIMIZE TOTAL COMPLETION TIME

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Evrim Didem Güneş

December, 1998

SCHEDULING WITH TOOL CHANGES TO MINIMIZE TOTAL COMPLETION TIME

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By
Evrim Didem Güneş

December, 1998

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

Asst. Prof. M. Selim Aktürk(Principal Advisor)

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

Assoc. Prof. Ömer Benli

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

Asst. Prof. Oya Ekin Karaşan

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

SCHEDULING WITH TOOL CHANGES TO MINIMIZE TOTAL COMPLETION TIME

Evrim Didem Güneş
M.S. in Industrial Engineering
Supervisor: Asst. Prof. M. Selim Aktürk
December, 1998

In the literature, scheduling models do not consider the unavailability of tools. The tool management literature separately considers tool loading problem when tool changes are due to part mix. However in manufacturing settings tools are changed more often due to tool wear. In this research, the problem of scheduling a set of jobs to minimize total completion time on a single CNC machine is considered where the cutting tool is subject to wear.

We show that this problem is NP-hard in the strong sense. We discuss the behavior of SPT heuristic and show that its worst case performance ratio is bounded above by a constant. A pseudo-polynomial dynamic programming formulation is provided to solve the problem optimally. Furthermore, heuristic algorithms are developed including dispatching heuristics and local search algorithms. It is observed that the performance of SPT rule gets worse as the tool change time increases and tool life decreases. The best improvement over the SPT rule's performance is achieved by the proposed genetic algorithm with problem space search.

Key words: Scheduling, Completion Time, Tool Management, Heuristics

ÖZET

KESİCİ UÇ DEĞİŞİMİ DURUMUNDA TOPLAM İŞ BİTİM ZAMANINI ENAZLAMAK İÇİN ÇİZELGELEME

Evrim Didem Güneş
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. M. Selim Aktürk
Aralık, 1998

Literatürdeki çizelgeleme modellerinde kesici uç kullanımında kısıt yoktur. Kesici uç işletim sistemi literatürü uç değişimi parça sırasına bağlı olduğunda kesici uç yükleme problemini ayrıca ele alır. Fakat üretim koşullarında kesici uçlar daha cok aşınmaya bağlı olarak değiştirilir. Bu çalışmada, kesici ucun aşınmaya maruz kaldığı tek bir CNC makinasinda bir grup işin toplam iş bitim zamanını enazlamak üzere çizelgelenmesi problemi ele alınmıştır.

Bu problemin kuvvetli anlamda NP-zor oldugu ve en kısa işlem süresi (EİS) kuralının en kötü durum performans oranının üstten bir sabitle sınırlı olduğu gösterilmiştir. Problemi eniyileyerek çözmek için bir sahte polinom dinamik programlama formülasyonu verilmiştir. Ayrıca, bazı hızlı sezgisel algoritmalar ve yerel tarama algoritmaları geliştirilmiştir. EİS kuralının performansının uç değiştirme zamani arttıkça ve uç kullanım ömrü azaldıkça kötüye gittiği gözlenmiştir. EİS kuralı üzerine en çok gelişmeyi problem uzayı taraması kullanan genetik algoritma sağlamiştır.

Anahtar sözcükler: Çizelgeleme, İş Bitim Zamanı, Kesici Uç İşletim Sistemi, Sezgisel Yöntemler.

To my family

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Selim Aktürk for his invaluable advises and encouragement during my graduate study. He has been guiding me with patience and everlasting interest not only for this research but also for my future career.

I am also indebted to Jay B. Ghosh for his great helps and guidence in all phases of this study. He has invaluable contribution to this theis.

I am grateful to Robert Storer for showing keen interest to the problem and helping for the solution methods.

I am also indebted to Ömer Benli and Oya Ekin Karaşan for showing keen interest to the subject matter and accepting to read and review this thesis.

I would like to thank to my friend Bora Uçar for his helps in C programming.

I would like to express my deepest gratitude to Hande Yaman, for being with me in every field of life and giving me morale and academic support during six years in the university. I owe much to her friendship and solidarity. I am also grateful to my officemate Ayten Türkcan for continuously providing morale support to me. She gave me courage whenever I was desperate, and also helped me a lot for technical issues. I cannot forget the six years' friendship of Deniz Özdemir and the morale support she provided even when she was far away. I am grateful to Senem Erdem for her sincere friendship and support, and the night calls, giving me relief in the hard times.

Finally, I want to thank to Gonca Yıdırım, Banu Yüksel, Özgür Ceyhan, Seçil Gergün, Barış Ata, Emre Alper Yıldırım, Murat Özgür Kara, Füsun Atik, Özlem Özhan and my sister Elif for their friendship and encouragement for my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scheduling has been an attractive field for researchers for a long time. It deserves this attention since scheduling is an important part of strategic planning in industry and has significant impact on all economic activities. The term scheduling can be defined as "the process of organizing, choosing and timing resource usage to carry out all activities/tasks necessary to produce the desired outputs at desired times, while satisfying a large number of time and relationship constraints among the activities and resources" [22]. In short, it is the allocation of scarce resources over time to a collection of tasks. Scheduling decisions in manufacturing organizations are associated with many cost terms. These are mainly related with customer satisfaction, such as tardiness costs, or investments into system resources as flowtime costs.

The manufacturing organizations have been using flexible manufacturing systems (FMSs) widely in recent years in order to be able to meet the diversified customer needs and compete in today's world market. An FMS is mainly defined as a system dealing with high level distribution data processing and automated material flow using computer controlled machines, assembly cells, industrial robots, inspection machines and so on, together with computer integrated material handling and storage systems. Basically, it is a group of CNC machine tools interconnected by a material handling system and controlled by a computer system.

Tool management is the most dynamic and critical facility in FMSs and requires keen attention. The cutting tools used in FMSs are subject to wear and they have relatively short tool lives in the planning horizon. Moreover, the tool holding capacity of FMSs is limited, so the machine may not be able to carry all the required tools to complete the jobs. For this reason, one main task to accomplish for tool management is to find a scheduling strategy to account for tool availability and tool changes.

Scheduling activities are done considering different goals. Sometimes the customer satisfaction in terms of meeting the deadline is taken into account, which is reflected as the tardiness cost in the objective function of the scheduling problem. Another cost measure is the time a job spends in the system, defined as the flowtime of the job. Flowtime cost is related with the investment into system resources and is reflected as the work in process inventories in the system. Minimizing the total flowtime is an important goal for scheduling activities considering the importance of maintaining low inventory levels for manufacturing firms.

In this study, a single CNC machine is considered, being a part of a flexible manufacturing system, and the scheduling problem with the objective of minimizing total flowtime is studied while also focusing on the tool management issue to cope with the tool changes due to tool wear. The existing studies in the literature ignore the interaction between the scheduling decisions and the tool change requirements due to tool wear. In the tool management literature, the tool changes are generally considered to be due to part mix, that is, due to different tooling requirements of the parts. The cost terms which are directly related with scheduling decisions such as flowtime of jobs are not included in the objective function. On the other side, the scheduling literature also does not consider the tool change requirements. There are few studies considering the resource unavailability, but they consider the resources as machines, and the unavailability is limited to occur for one time only.

As a result, the scheduling problem with tool changes due to tool wear is an untouched topic in the literature. In this study, we aim to show the validity of this problem and try to find solution methods to fill in this gap in the literature. We study the simplest case of the joint tool management and scheduling problem in order to provide some insights to find a solution for the more general cases. The problem we consider is characterized by the following conditions: There are n jobs with predetermined processing times. There is ample tool of single type, which has a constant tool life and constant tool changing time. When the tool life is over, the tool has to be changed. We assume that a manufacturing operation cannot be interrupted for a tool change due to surface finish requirements.

At the beginning, we analyze the complexity of this problem and show that it is strongly NP-hard. Afterwards, we investigate the behavior of the well known shortest processing time (SPT) heuristic for this problem and show that its worst case performance is bounded above by a constant. Then, we discuss some conditions which would guarantee optimality of SPT rule.

Since our problem is NP-hard, it is justified to solve it by heuristic approaches. In this study, we provide a dynamic programming algorithm which is shown to be pseudo-polynomial. Furthermore, we develop several heuristic algorithms including dispatching heuristics and local search algorithms. We test the performance of the proposed algorithms on a set of randomly generated problems and discuss the results. We show that an improvement over the performance of the SPT heuristic is provided with the proposed algorithms.

We will also elaborate on an extension of this problem in order to provide insights for a possible future research. We consider to incorporate the determination of the machining parameters which would affect the processing times and the tool life into the scheduling problem with tool changes. This time the objective function would also include the manufacturing costs in addition to the flowtime cost.

The remainder of the thesis can be outlined as follows. In the following chapter, we give a short review of the literature on the scheduling problems with an availability constraint and tool management studies along with some studies on bin packing problem, which is related with our problem in some aspects. We define the underlying assumptions, and give a list of notation in Chapter 3. In this chapter, we also discuss the characteristics of the problem and analyze the performance of the SPT heuristic for our problem providing some examples of interesting instances. We also investigate the worst case performance of SPT, and the conditions for optimality of this heuristic in this chapter. In Chapter 4, we explain the pseudo-polynomial dynamic programming formulation and the proposed heuristic algorithms in detail. Then, we illustrate these algorithms on a numerical example. In Chapter 5, we discuss the computational analysis done with these algorithms. The discussion on a possible extension of this problem is given in Chapter 6. Finally in Chapter 7, the concluding remarks of this study is provided with some suggestions for future research.

Chapter 2

Literature Review

Research on manufacturing has been traditionally done in separate veins for tool management issues and scheduling problems. In both fields, extensive research has been done for modeling the systems, and developing control methods. However, the interaction between these two levels of manufacturing decision processes has not been addressed by the researchers.

Scheduling can be defined as the allocation of scarce resources over time to a collection of tasks. It is a decision-making process that exists in most manufacturing systems, and also in most information-processing environments which plays a crucial role in strategic planning. For this reason it has attracted the attention of researchers since the beginning of this century, with the work of Henry Gantt and other pioneers. Since than, considerable amount of theoretical work has been done concerning various different models. An excellent overview of deterministic scheduling can be found in the textbook by Baker [7]. A recent book by Pinedo [24] deals with both deterministic and stochastic models with applications to real world problems.

There are many costs to the system associated with the scheduling decisions. Among them the most commons are cost of completing the tasks after the due date, which is called the tardiness cost, and cost of tasks waiting to be completed, called the flowtime cost. Flowtime of a job is the time it spends in the system. The cost of flowtime involves the investment into system resources, and is reflected as the inventory levels in the system. Especially with the emergence of new paradigms in production systems such as just in time philosophy, minimizing the inventory levels in the manufacturing environment gained importance. Consequently, the scheduling objective corresponding to this goal is minimizing the total flowtime. In this study, our objective will be minimizing the total flowtime.

Although the variety of different models studied in scheduling theory lies in a big range, there are still some deficiencies in the literature. The resources in scheduling theory are mostly considered as the machines, without referring to the tooling level. As discussed in Lee et al. [21] and Pinedo [24], most theoretical models do not take the unavailability of resources into account. It is usually assumed that the machine is available at all times. However in the real world, machines are usually not continuously available. Certainly, this observation is valid for the machine tools, and the unavailability of tools is a more common situation since the tools actually have short lives with respect to the planning horizon, as reported by Gray et al. [13].

In the literature, there are no studies considering the tool life and tool change time requirement due to tool wear, and incorporating them with scheduling objectives. However, there are some studies done in recent years considering the unavailability of machines. These problems have similar characteristics with the scheduling with tool changes problem.

The research on scheduling with availability constraint is mostly focused on machine breakdowns and maintenance intervals. The most common objective is minimizing the total flowtime. Adiri et al. [1] considered flowtime scheduling problem when machine faces breakdowns at stochastic time epochs, and repair time is also stochastic. The processing times are assumed constant. They have provided the NP completeness result of the problem, and showed that SPT minimizes expected total flowtime when times to breakdown are exponential. In the case of single breakdown and concave distribution function of the time to breakdown, they have again showed the stochastic optimality of SPT. They have also analyzed the single deterministic breakdown case, and found a worst case performance bound for SPT heuristic, which was 5/4.

Lee and Liman [17] have also studied the same problem, but considered only deterministic single scheduled maintenance case. In this study, they have given a simpler proof of NP completeness, and found a better bound for SPT, being 9/7. They have also shown that this bound is tight.

There are also some studies on flowshop and parallel machine scheduling with an availability constraint. Lee and Liman [18] considered two machines in parallel scheduling problem of minimizing the total completion time where one machine is available all the time and the other machine is available from time zero up to a fixed point in time. They have given NP-completeness proof for the problem, and provided a pseudo-polynomial dynamic programming algorithm. Moreover, in this study, a heuristic is proposed which is based on a slight modification of SPT rule considering the capacity of the machine with availability constraint. This heuristic is shown to have an error bound of 0.50.

Lee [19] studied minimizing the makespan in the two-machine flowshop scheduling problem. The availability constraint applies for one of the machines. The NP hardness proof is done and a pseudo-polynomial dynamic programming formulation is provided in this study. In addition, he provides two heuristics with an error bound analysis, for problems with availability constraint on the first machine, and on the second machine.

In a companion paper, Lee [20] discusses the machine scheduling with an availability constraint in more detail. He analyses the problem for different performance measures such as makespan, total weighted completion time, tardiness, and number of tardy jobs, and for different machine environments such as single machine, parallel machines, and two machine flowshop. In each case, the complexity issue is discussed, and either a polynomial algorithm is provided, or the NP hardness proof is done. In case of NP completeness of the problem, pseudo-polynomial dynamic programming algorithms are developed to solve it optimally, and/or a heuristic with an error bound analysis is provided. In this study, two different cases are considered, which are resumable, and nonresumable cases. A job is called *resumable* if it can be interrupted in case of an unavailability, and can be continued after machine is available again. In *nonresumable* case, the job has to be restarted rather than continue. The nonresumable case is similar to the tool change problem, since for surface finish quality considerations, we do not let the process on a job to be interrupted, and continued after a tool change.

However, all these studies assume a single breakdown or maintenance interval. But, in the scheduling problem with tool changes this is not a realistic assumption and we can have several tool changes in a given time period due to relatively short tool lives.

There is an increasing need for manufacturing industries to achieve diverse, small lot production to be able to compete in today's world market. Numerical control (NC) is a form of programmable automation, designed to accommodate variations in product configurations. Principal applications of NC are in low and medium volume stations, primarily in a batch production mode. The results of a U.S. Census Bureau survey of nearly 10,000 manufacturing firms in 1990 offered insights into use of 17 manufacturing technologies, such as CAD/CAE, robots. NC machine tools was the most widely used manufacturing technology, with 41.5% of the respondents indicating its use. Machinery production statistics released by the Japanese Ministry of International Trade and Industry showed that the number of NC machine tools produced in Japan was equal to 61,695 in 1990, which made more than 75% of total machine production shares (Asai and Takashima [4]). Furthermore, one of the major components of a flexible manufacturing system (FMS) is computer numerical control machine tools. An FMS is usually defined as a group of CNC machine tools interconnected by a material handling system and controlled by a computer system. In view of the high investment and operating costs of the CNC machines and hence of FMSs, attention should be paid to their effective utilization.

Tool management is another area of research which has been extensively studied for nearly a hundred years, since Taylor [29] first recognized that the machining conditions should be optimized to minimize the machining cost. There's a great deal of work in the area of optimizing machining processes, such as Ermer [10], Hitomi [14] and Gopalakrishnan and Al-Khayyal [12]. Extensive modeling efforts have been devoted to capturing the relationship between machining parameters (e.g., cutting speed, feed rates, etc.), quality requirements (e.g., surface finish), time to complete the job, and the tooling cost. These relationships have been well developed for a wide variety of machining activities. However, in most of the studies the tool change requirement due to tool wear and its contribution to cost has not been considered. Akturk and Avci [2] proposed a new solution methodology to solve the machining conditions optimization and tool allocation from among alternative tools simultaneously, taking the tool wear and tool replacing times into consideration. However, in this study the objective is to minimize the total production cost, and any traditional scheduling objective is not included in the cost calculation.

Gray et al. [13] and Veeramani et al. [31] give extensive surveys on the tool management issues in automated manufacturing systems, and emphasize that the lack of tool management considerations has resulted in the poor performance of these systems. Kouvelis [16] report that the tooling cost accounts for 25% to 30% of both fixed and variable cost of production.

According to Gray et al. [13] the tool management problem can be examined as tool-level, machine-level, and system-level issues. At the machine level, the tool management problem which is defined as the loading problem by Stecke [25] is, "the problem of allocating tools to the machine and simultaneously sequencing the parts to be processed so as to optimize some measure of production performance". Since machine flexibility is a direct consequence of the tool magazine capacity, planning models especially take into account the limitation of tool magazine, and the necessity of tool changes because of this limitation. Stecke [25] formulates this problem as a nonlinear mixed-integer programming problem and solves it through linearization techniques.

A general overview of problems studied and the solution methods proposed for tool management issues can be found in Crama [9]. In this research, the existing models on single machine, flow shop, parallel machine and robotic flow shop are discussed and some mathematical models are proposed for modeling tool loading problem. Various objectives are studied for one machine tool loading problem, such as minimizing the number of tool switches and number of switching instants, maximizing the number of parts without tool switches etc. as stated by Crama [9].

These models are mostly motivated from the industrial experience that time needed for tool interchanging is significant compared to processing times, as stated by Tang and Denardo [27]. And thus, from scheduling perspective, assuming that tool change time is too large that it dominates the processing times, they have tried to minimize the number of tool switches. They have studied a single machine case with given tool requirements, where tool changes are required due to part mix. They have provided heuristic algorithms for job scheduling in this environment, and an optimal procedure, namely the common sense rule Keep Tool Needed Soon(KTNS) for a fixed job sequence. They have also studied the case of parallel tool switchings in a companion paper [28], and this time the objective was chosen as minimizing the number of switching instants.

Crama et al. [8] have also studied the tool loading problem. They have proposed several heuristics including construction and improvement strategies, and done computational studies. They have also stated the NP hardness result for the problem. Tool loading problem is generally modeled as traveling salesman problem and TSP heuristics are widely utilized by the researchers, taking the estimate of maximum number of tool switches between two jobs as the length of the arc joining them.

In all these studies mentioned, all tool changes are considered due to part mix, that is, different parts require different tools, and since the tool magazine capacity is limited, it cannot hold all the necessary tools for completing all the jobs. The processing times and tool lives are assumed to be constant, ignoring the fact that tool wear, consequently the tool replacement frequency is directly related with the machining conditions selection. Moreover, in the multiple operations case, the tool replacements due to tool wear can have significant impact on total cost of production and throughput of parts as shown by Tetzlaff [30]. Gray et. al. [13] reported that tools are changed ten times more often due to tool wear than due to part mix because of relatively short tool lives of many turning tools.

In the tool management literature, as briefly summarized above, the scheduling problem with a traditional scheduling cost measure such as flowtime, tardiness etc. is not considered. The tool replacements are considered to be due to part mix, ignoring tool life restrictions, and tool change times are assumed to be so large that the number of tool replacements are tried to be minimized in most of the studies. However, with the new technology, in CNC machines tool change times are considerably reduced, so the processing times are not always dominated. For this reason, while scheduling a given set of jobs, considering only the tool change constraint would not result in good solutions with respect to the job attributes, such as flow times.

In the problem of scheduling with tool changes, there are two sources of input to the cost function. One is just the increase in flowtime of jobs by the total time spent for processing times up to that job (which is the classical total flowtime cost), and the other one is the increase in flowtime as a consequence of tool change times spent up to that job. When the tool change time is long, minimizing the number of tool changes done gains importance, although we are not saying that it minimizes the overall objective. From this aspect, our problem is similar to the famous bin packing problem, which is stated as: given a list of $L = (a_1, a_2, ..., a_n)$ of real numbers in $(0, 1]$, place the elements of L into a minimum number L^* of "bins" so that no bin contains numbers whose sum exceeds 1. This problem is especially used to model several practical problems in computer science and is well studied in the literature. Since bin packing problem is NP complete (Garey and Johnson [11]), the heuristic procedures and their worst case performances are widely investigated in the literature.

One of the pioneering studies in bin packing problem is done by Johnson et al. [15]. They have analyzed four heuristic algorithms, namely first-fit (FF), best fit (BF), first-fit-decreasing (FFD), and best-fit-decreasing (BFD). The first fit rule assigns each successive element into the first available bin of the sequence $B_1, B_2, ...$ into which it will fit. The best fit algorithm places each successive piece into the leftmost bin, for which the remaining unused capacity is the least. FFD and BFD are the applications of FF and BF respectively, after ordering the elements in nonincreasing order of their sizes. In this study, the worst case asymptotic performance bound for each of these algorithms, together with examples for worst instances are given. They have found that the first two heuristics, FF and BF have the same performance bound, $\frac{17}{10}$, where BFD and FFD have the performance ratio as $\frac{11}{9}$.

Another algorithm, called next-fit-decreasing is studied by Baker and Coffman [5]. The next-fit rule is applied by placing as many pieces into bin B_1 as can be done, then passing to next bin and placing as many possible into that bin, and continuing this way, without turning back to a bin even if it has enough capacity. Next-fit-decreasing (NFD) is the variation of this rule with a preordering of elements in nonincreasing order of sizes. In this study, the asymptotic bound for NFD rule is given as 1.691, and an example is provided showing the tightness of bound.

A recent study on bin-packing problem, done by Anily et al. [3], gives a brief overview on the heuristics for classical bin packing problem and their performances, and provides absolute performance bounds for next-fit-decreasing and next-fit-increasing heuristics, which are both equal to 1.75. Furthermore, they analyze the problem in case of more general cost structures, when the cost of a bin is a monotone and concave function of the number of items assigned to it. They show that NF, FF, BF, FFD, and BFD have neither finite absolute performance ratios, nor asymptotic performance ratios for the bin packing problem with general cost structures. Furthermore, they prove that the next-fit-increasing heuristic has an absolute worst case performance bound of no more than 1.75, and an asymptotic worst-case bound of 1.691 for any monotone and concave cost function.

In conclusion, in the existing literature, tool management issues and scheduling issues are considered separately, and the interaction between them is ignored. In this study, they will tried to be handled together. I tried to solve the simplest case of the joint tool management and scheduling problem, with a single tool and constant processing times, hoping to provide some insights to the characteristics of this problem to be a first step in search of solutions for more generalized cases.

Chapter 3

Problem Statement

Scheduling is an important part of strategic planning in industry, since it can have a significant impact on all economic activities. There are various costs associated with scheduling decisions. Scheduling activities are done considering different objectives according to the relative importance of these costs. Flowtime is the time a job spends in the system. Total flowtime is the sum of flowtimes of all the jobs. The costs associated with this objective are primarily the investments in system resources, reflected by the work-in-process inventories. The objective of scheduling to minimize total flow time is to maintain low inventory levels, which has been one of the key objectives in manufacturing organizations especially after the recognition of importance of "zero inventory" philosophy.

Flexible manufacturing systems (FMSs) have been widely used in manufacturing industries to cope with increasing competition. Tool management is the most dynamic and critical facility in FMSs and requires keen attention. Lack of attention to tooling issues in FMSs can affect all systems performance, since tool management is directly related with product design options, machine loading, job batching and capacity scheduling decisions. Automated machine tools have to be changed during production since they are subject to wear, and manufacturing processes are frequently interrupted for tool change due to tool wear compared to changes due to part mix. As explained in the previous chapter, in the literature there are no studies considering scheduling decisions in a manufacturing environment, where tool change due to tool wear occurs. This study aims to contribute to filling this gap in the literature.

The organization of this chapter is as follows. In §2.1 the definition of problem and underlying assumptions will be given. In §2.2 the structural properties of the problem will be explained, and some further analysis will be done concerning the complexity issues and performance of SPT heuristic for this problem. Finally, in §2.3 a brief summary will be done.

3.1 Problem Definition and Assumptions

In this study, our aim is to solve the scheduling problem with an availability constraint in an automated machining environment to minimize total flow time. The assumptions about the operating policy and the characteristics of the system considered in this study are as follows:

- There is a single machine which is continuously available.
- There are n jobs ready at time zero.
- The processing times of jobs are constant and known apriori.
- There is one type of tool used in this machine with a known, constant tool life.
- There is no limit on the amount of tool available.
- When the tool life ends (tool is worn out) tool has to be taken off the machine, and a new one has to be placed. The time spent for this process, *i.e.* tool change time, is constant.
- We do not allow a tool change during a manufacturing operation to achieve the desired surface finish quality.

Under these assumptions, we wish to find a schedule that minimizes the total flow time of jobs.

The notation used throughout the thesis is as follows:

T_L: Tool life

Tc: Tool change time

p_i: Processing time of job i

$p_{[i]}$: Processing time of job at position i

C_i: Completion time (flowtime) of job i

t_j: Sum of processing times of jobs using jth tool

m: Number of tools used in the optimal schedule

d: Number of tools used in the SPT schedule

θ: The fraction of number of tools used in SPT schedule to the number of tools used in the optimal schedule *i.e.* $d = \theta m$, where $\theta \geq 1$

S: The SPT schedule

S^*: The optimal schedule

η_j^σ: Number of jobs finished using jth tool in schedule σ

k: Number of jobs finished using the first tool in SPT schedule ($k = \eta_1^S$)

K_σ: Number of tools used by schedule σ

Z_σ: The total flow time of schedule σ

C_1^σ: The total flow time of schedule σ without considering the tool change times

C_2^σ: The contribution of tool change times to the flowtime of schedule σ

ρ: Performance ratio of SPT schedule over optimal schedule

3.2 Characteristics of the Problem

Minimizing the total flowtime is one of the basic objectives studied in the scheduling literature. There is a well known dispatching rule, namely the shortest processing time (SPT), which gives an optimal sequence for $1||\sum C_j$ problem. However, the structure of the problem changes dramatically when we consider tool changes.

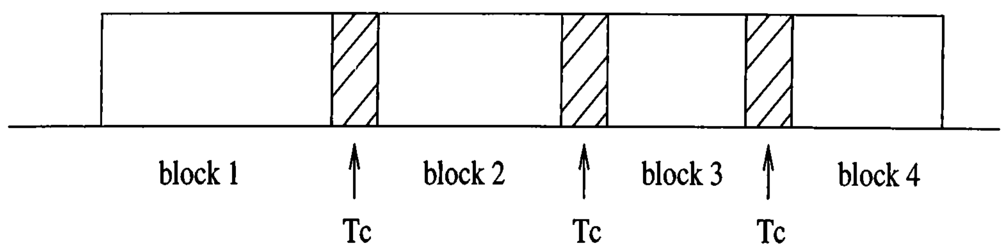

Figure 3.1: Representation of a schedule as blocks of jobs

If we consider the jobs sharing the same tool as a block, a schedule can be viewed as blocks of jobs separated by tool changes (see Figure 3.1). This representation would be helpful to gain more insight into the problem structure. Note that, the length of blocks do not have to be same. This is because, when we cannot assign more jobs to a tool although the tool life has not finished, there is no meaning in waiting till the end of tool life to make the tool change. So, in such cases, the tool is immediately replaced by a new one, and the block length shows the used portion of the tool. This property of the tool change problem is another point that makes it different from the existing models. In models of scheduled maintenance (see [19], [20], [17], [18] for examples) the unused capacity of the machine is counted as wasted time which is added to the flowtime of the latter jobs. However for the tool change problem this is not the case.

With this representation, we can consider the blocks as job strings which must be processed together, and has length $t_j + Tc$. Then we know the following structural properties of the problem from the scheduling literature [6];

- If jobs q and r are within the same block, then
  q precedes r if $p_q \leq p_r$.
- Furthermore, for blocks i and j:
  i precedes j if $\frac{t_i+Tc}{\eta_i} \leq \frac{t_j+Tc}{\eta_j}$

Thus we conjecture that, in an optimal schedule, blocks should be in non-increasing order of the number of jobs they have, and the jobs in a block must be in SPT order.

The total flowtime of such a sequence of jobs has two main parts, the first part shows the total flowtime without tool changes, and the second part is added as the increase in flowtime as a result of tool changes. When we ignore the tool changes, the total flowtime of a schedule σ is equal to:

$$C_1^\sigma = \sum_{q=1}^{n}(n-q+1)p_{[q]}$$

When we introduce the tool changes into the picture, we have to add the contribution of tool changes to the objective function, which can be written as:

$$C_2^\sigma = \sum_{j=1}^{K_\sigma}(j-1)\eta_j^\sigma Tc$$

This follows from the fact that before each job using the jth tool, $(j-1)$ tool changes would have been done, and this would increase each such job's completion time by Tc.

Then total flow time of a schedule σ is:

$$Z_\sigma = C_1^\sigma + C_2^\sigma = \sum_{q=1}^{n}(n-q+1)p_{[q]} + \sum_{j=1}^{K_\sigma}(j-1)\eta_j^\sigma Tc$$

These two parts of the objective function are conflicting in terms of the requirements to be minimized. In order to minimize C_1^σ, we should apply SPT, that is, the shorter jobs should be put in earlier blocks, and longer jobs be remained for the later blocks. In an SPT schedule, the number of jobs in the blocks, η_j's are in non-increasing order of j, since we can assign less number of jobs to a block if their processing times are long. And for some instances this may increase the number of tools used, under-utilizing the tool life in later blocks. On the other hand, for C_2^σ to be minimized, the number of blocks should be decreased, and this can be done only if some larger jobs are scheduled early so as to maintain balance in the later ones. From this aspect, problem is similar to bin-packing problem, but certainly not equivalent. Especially as Tc gets larger, this conflict between cost components makes the problem more difficult to solve.

Having defined some basic properties of the problem, we can analyze it in more detail. In the following sections, the complexity issues, and the performance of SPT heuristic for this problem will be discussed.

3.2.1 Complexity of the Problem

Although flowtime problem is very easily solved optimally by the SPT rule, when there are tool changes the problem becomes NP-hard in the strong sense. The proof will be done by transforming the 3-partition problem, which is a well-known NP-hard problem [11], into our problem, scheduling with tool changes. So, first we should state the 3-partition problem:

Given $3m$ integers $\{ a_1, a_2, \cdots, a_{3m} \}$ such that $\sum_{q=1}^{3m} a_q = mB$ and $\frac{B}{4} < a_q < \frac{B}{2}$, can this set be partitioned into m 3-tuples such that $\sum a_q = B$ for each 3-tuple?

The following theorem shows that tool change problem is NP-hard.

Theorem 3.1: $1/\text{tool change}/\sum C_i$ is strongly NP-hard.

Proof: Let (P) denote the problem of scheduling with tool changes. Create an instance of (P) such that;

$$
\begin{aligned}
& n = 3m \\
& p_q = a_q, \qquad \text{let} \quad a_1 \leq a_2 \cdots \leq a_{3m} \\
& T_L = B \\
& A = \sum_{q=1}^{n} (n-q+1) a_{n-q+1} \\
& Tc = C >> A
\end{aligned}
$$

It was shown that for any schedule σ,

$$Z_\sigma = C_1^\sigma + C_2^\sigma = \sum_q (n-q+1) p_{[q]} + \sum_{j=1}^{K_\sigma} (j-1) \eta_j^\sigma Tc$$

Question: Does there exist a schedule with total completion time, $\sum_{q=1}^{n} C_q \leq A + \frac{3m(m-1)}{2} C$?

$\Rightarrow$ If 3-partition has a solution the answer is YES.

If 3-partition exists

$$\sum_{j=1}^{m} (j-1) \eta_j = \frac{3m(m-1)}{2}$$

Then,

$$
\begin{aligned}
\sum_{q=1}^{n} C_q &= \sum_{q=1}^{n} (n-q+1) p_{[q]} + [\sum_{j=1}^{m} (i-1) \eta_i] Tc \\
&= \sum_{q=1}^{n} (n-q+1) p_{[q]} + \frac{3m(m-1)}{2} C \\
&\leq A + \frac{3m(m-1)}{2} C
\end{aligned}
$$

Thus, the answer to question is yes.

$\Leftarrow$ If partition does not have a solution, the answer is NO.

If 3-partition does not exists, any schedule will have to use $m + l \ (l > 0)$ tools and for $1 \leq i \leq m + l, \ \ 0 < \eta_i \leq 3$ will be true.

Let $\Delta_i = 3 - \eta_i$ for $1 \leq i \leq m$. Then we know that $\Delta_i > 0$ for at least one $i, \ 1 \leq i \leq m$. Moreover, $\sum_{i=1}^{m+l} \eta_i = 3m$. Thus

$$\sum_{i=1}^{m} \eta_i + \sum_{i=m+1}^{m+l} \eta_i = 3m$$

$$\sum_{i=m+1}^{m+l} \eta_i = 3m - \sum_{i=1}^{m} \eta_i = \sum_{i=1}^{m} 3 - \sum_{i=1}^{m} \eta_i = \sum_{i=1}^{m} (3 - \eta_i)$$

$$\Rightarrow \sum_{i=1}^{l} \eta_{m+i} = \sum_{i=1}^{m} \Delta_i \qquad (*)$$

For any schedule consider the second part of the flowtime value:

$$[\sum_{i=1}^{m+l}(i-1)\eta_i]Tc = [\sum_{i=1}^{m}(i-1)\eta_i + \sum_{i=m+1}^{m+l}(i-1)\eta_i]C$$
$$= [\sum_{i=1}^{m}(i-1)(3-\Delta_i) + \sum_{i=m+1}^{m+l}(i-1)\eta_i]C$$
$$= [3\sum_{i=1}^{m}(i-1) - \sum_{i=1}^{m}(i-1)\Delta_i + \sum_{i=1}^{l}(m+i-1)\eta_{m+i}]C$$
$$= [\frac{3m(m-1)}{2} - \sum_{i=1}^{m}(i-1)\Delta_i + m\sum_{i=1}^{l}\eta_{m+i} + \sum_{i=1}^{l}(i-1)\eta_{m+i}]C$$
$$= \frac{3m(m-1)}{2}C + [m\sum_{i=1}^{l}\eta_{m+i} - \sum_{i=1}^{m}(i-1)\Delta_i + \sum_{i=1}^{l}(i-1)\eta_{m+i}]C$$
$$= \frac{3m(m-1)}{2}C + [m\sum_{i=1}^{m}\Delta_i - \sum_{i=1}^{m}(i-1)\Delta_i + \sum_{i=1}^{l}(i-1)\eta_{m+i}]C \qquad \text{(using (*))}$$
$$= \frac{3m(m-1)}{2}C + [\sum_{i=1}^{m}(m-i+1)\Delta_i + \sum_{i=1}^{l}(i-1)\eta_{m+i}]C$$

Clearly, the term within the braces is ≥ 1. Hence, for any schedule, $C_2 \geq \frac{3m(m-1)}{2}C + C$. Thus,

$$\sum_{q=1}^{n} C_q \geq \frac{3m(m-1)}{2}C + C$$

Since $C >> A$, we conclude

$$\sum_{q=1}^{n} C_q > \frac{3m(m-1)}{2}C + A$$

Thus, the answer to question is NO. Hence it is shown that, even for the special case, the problem is reduced to another problem that is known to be strongly NP-hard, namely 3-partition problem. This proves the strong NP-Hardness of our problem. □

Furthermore, even when we fix the number of tools that can be used as a constant, the problem still remains NP-hard. This is shown by the following theorem.

Theorem 3.2: 1/tool change-limited tools/$\sum C_i$ is NP-hard.

Proof: Proof will be done by using a reduction from the partition problem. Partition problem is stated as follows:

Given a set $A = a_1, \cdots, a_n$, can we partition A into subsets A_1, and A_2 such that $\sum_{i \in A_1} a_i = \sum_{i \in A_2} a_i$?

This problem is well known to be NP-Hard [11].

Create an instance of our problem such that:

$$m = 2 \qquad \text{(number of tools)}$$

$$n = \text{number of jobs}$$

$$p_q = a_q$$

$$T_L = \frac{1}{2} \sum_{q \in A} a_q$$

In this instance, it is obvious that there exists a feasible schedule if and only if partition problem has a solution. Therefore, answering the question if there is a feasible schedule with 2 tools is NP-Hard. Since even the feasibility check is NP-hard, the problem is NP-hard as well. □

Thus we have shown that tool change problem is NP-hard, even when the number of tools is fixed, and as low as 2.

3.2.2 Performance of SPT Heuristic

As mentioned before, for the classical flowtime problem, SPT is a very powerful rule. And, when $Tc \to 0$, it is obvious that SPT would be optimal, since then the problem would reduce to the classical $1||\sum C_i$ problem. However, for our problem, depending on the magnitude of Tc value, it may not perform as well. We can illustrate this with an example:

Let $n = 5$, $T_L = 6$, and $Tc = 2$. And let the p_q values for 5 jobs given as $1, 2, 2, 3, 4$. The SPT schedule would be as follows with respect to the processing times:

$$1 \;\; 2 \;\; 2 \;\; \mathbf{Tc} \;\; 3 \;\; \mathbf{Tc} \;\; 4$$

where **Tc** represents a tool change at that point. So, SPT schedule requires 2 tool changes, and the total flowtime for this schedule equals 35.

On the other hand, the optimal schedule would be:

$$1 \;\; 2 \;\; 3 \;\; \mathbf{Tc} \;\; 2 \;\; 4$$

with total flowtime equal to 34.

Now, let $Tc = 0.5$ for the same example. This time, $Z_S = 30.5$, where flowtime of the second schedule (which was optimal for $Tc = 2$) becomes 31.

As the above example suggests, the performance of the SPT heuristic changes with different Tc values. Intuitively, for small values of Tc, SPT should perform well, whereas it may lose power as Tc value increases. In order to understand if this intuition makes sense, we have made some experiments on small problem instances for which we can find the optimal objective function value. Different problems are solved for all possible cases with $n = 8, 12, 16$ and $\frac{Tc}{T_L} = 0.1, 1, 10$. For each of these 8 cases except $n = 16$, 4 problems with $K_S = 3$ and four having $K_S = 4$ are solved by SPT and compared with the optimal result. For 16 jobs we were not able to find the optimal solution when $K_S = 4$ in four hours of CPU time, so we have only considered $K_S = 3$. The results are summarized in Table 3.1. For each combination of n and $\frac{Tc}{T_L}$ values, the minimum percent deviation (MinPD), average percent deviation (APD) and maximum percent deviation (MaxPD) of the results obtained by SPT rule from the optimal objective function value are presented in this table.

As we can see from these results, performance of the SPT rule gets worse for our problem as tool change time increases. When the $\frac{Tc}{T_L}$ value is small (as 0.1), SPT is optimal most of the time, especially for the small problem sizes SPT rule dominates. Intuitively, one thinks that as $Tc \to \infty$, the problem can be seen as a bin packing problem, so as to minimize the number of tool changes needed, since this time Tc value would dominate. However, although

	$Tc/T_L = 0.1$			$Tc/T_L = 1$			$Tc/T_L = 10$		
n	MinPD	APD	MaxPD	MinPD	APD	MaxPD	MinPD	APD	MaxPD
8	0.00	0.00	0.00	0.00	1.2869	6.2500	0.00	3.8193	16.1591
12	0.00	0.00	0.00	0.00	1.2374	4.3564	0.00	5.5632	17.0616
16	0.00	0.0444	0.1776	0.00	2.0224	4.5399	0.00	5.8903	11.9041

Table 3.1: Comparison of SPT performance with optimal values

this observation is logical, the optimal bin packing solution for tool change problem does not always give the optimal flowtime objective.

Moreover, when $Tc \to \infty$, the performance of SPT heuristic for our problem does not coincide with its performance for the bin packing problem. The worst instance that SPT rule can behave for the bin packing problem is given by Johnson et al. [15]. This example is presented below and the ratio of C_2^σ values for optimal bin packing solution and SPT solution is calculated. Note that as $Tc \to \infty$, $Z_\sigma \to C_2^\sigma$.

Let T_L be 101 and $Tc \to \infty$. We have five job types with the following processing times p_i, and number of jobs s_k for job type k:

type $A \to p_i = 51$ $\quad s_A = 10$

type $B \to p_i = 34$ $\quad s_B = 10$

type $C \to p_i = 16$ $\quad s_C = 3$

type $D \to p_i = 10$ $\quad s_D = 7$

type $E \to p_i = 6$ $\quad s_E = 7$

The optimal bin packing solution will use 10 tools with the following allocation of jobs to tools:

Tools $1-7$: $(E \quad D \quad B \quad A)$

Tools $8-10$: $(C \quad B \quad A)$

This sequence has the C_2^σ value equal to $156Tc$.

The SPT sequence will use 17 tools with the following allocation:

Tool 1: $(E\ E\ E\ E\ E\ E\ E\ D\ D\ D\ D\ D)$

Tool 2: $(D\ D\ C\ C\ C)$

Tools $3-7$: $(B\ B\ )$

Tools $8-17$: (A)

This sequence has C_2^σ value equal to $160Tc$.

As we see in this example even if the bin packing solution of the SPT rule has a performance ratio of $\frac{17}{10} = 1.7$, its performance for our problem approaches to $\frac{160Tc}{156Tc} = 1.025$.

Furthermore, we can provide an instance where SPT schedule is better than the optimal bin packing solution, even when Tc approaches to infinity. Consider the following example:

Consider an instance where $Tc \to \infty$, $T_L = 1$, and there are four job types with the following processing times p_i, and number of jobs s_k for job type k:

type $A \to p_i = \frac{1}{3} + \epsilon$ $\qquad s_A = 6$

type $B \to p_i = \frac{1}{6} - \epsilon$ $\qquad s_B = 2$

type $C \to p_i = \frac{1}{9} - \epsilon$ $\qquad s_C = 3$

type $D \to p_i = \frac{1}{12} - \epsilon$ $\qquad s_D = 4$

The optimal bin packing solution uses 3 tools ordering the jobs as:

$$D\ D\ D\ D\ A\ A\ \mathbf{Tc}\ C\ C\ C\ A\ A\ \mathbf{Tc}\ B\ B\ A\ A$$

and will give $C_2^\sigma = 13Tc$.

However, SPT rule would result in the following order:

$$D \ D \ D \ D \ C \ C \ C \ B \ B \ \textbf{Tc} \ A \ A \ \textbf{Tc} \ A \ A \ \textbf{Tc} \ A \ A$$

with $C_2^S = 12Tc$.

This shows an instance where SPT schedule can still be better than bin packing solution even if $Tc \to \infty$. Since there is no tool change before the first block, and SPT fills all the small jobs to the first block, the effect of tool changes on flow time of the jobs is reduced although it requires one more tool change to complete the jobs.

In the worst example we could find for the performance of SPT rule for our problem when $Tc \to \infty$, the ratio $\frac{C_2^S}{C_2^{S*}}$ was 1.5. This instance is illustrated below:

Let $n = 6$, $T_L = 10$ and processing times of jobs be given as 1, 2, 3, 5, 6, 6. Then, if a job is represented by its processing time, the SPT sequence and the optimal sequence are as follows:

SPT: 1 2 3 **Tc** 5 **Tc** 6 **Tc** 6

Optimal: 2 3 5 **Tc** 1 6 **Tc** 6

The performance ratio of SPT would be $\frac{C_2^S}{C_2^{S*}} = \frac{6Tc}{4Tc} = 1.5$. This is the worst instance we have found for the performance of SPT rule when Tc approaches to infinity.

Having seen that SPT maintains its power for some instances, even when Tc is too large for SPT to be optimum, it can be expected that there is a bound on the worst case performance of SPT heuristic. So, the question is, how bad can SPT behave in the worst case, even when Tc value approaches to infinity? It turns out that, the performance ratio of SPT is bounded above by a constant, which will be proved after stating some properties needed in the proof:

1. We stated before that, in an optimal schedule, blocks should be in non-increasing order of the number of jobs they have, and within a block jobs are sorted in SPT order.

2. We know that the following relation holds between n and m:

$$n \geq m \geq \frac{np_{min}}{T_L}$$

where p_{min} is the minimum processing time value.
Then if $n \to \infty \Rightarrow m \to \infty$ also holds.

Having stated the necessary structural properties, we can investigate the worst case behavior of SPT heuristic.

In order to remind the notation and clarify the steps to be taken in the proof, let us rewrite the cost components and the expression for the performance ratio once more. We define the cost components as follows:

$$C_1^\sigma = \sum_{q=1}^{n}(n-q+1)p_{[q]}$$

$$C_2^\sigma = \sum_{j=1}^{K_\sigma}(j-1)\eta_j^\sigma Tc$$

Then the total flowtime is:

$$Z_\sigma = C_1^\sigma + C_2^\sigma$$

Thus the performance ratio is:

$$\rho = \frac{Z_S}{Z_{S^*}} = \frac{C_1^S + [\sum_{i=1}^{K_S}(i-1).\eta_i^S].T_c}{C_1^{S^*} + [\sum_{i=1}^{K_{S^*}}(i-1).\eta_i^{S^*}].T_c}$$

We know that as $T_c \to 0$,

$$\rho \to \frac{C_1^S}{C_1^{S^*}} = 1$$

When $T_c \to \infty$,

$$\rho \to \frac{C_2^S}{C_2^{S^*}}$$

We are trying to prove that this ratio is bounded by a constant. First of all we have made an observation:

Assuming that SPT schedule is not the optimal one, it must use more tools than the optimal schedule. Because, SPT minimizes C_1, and if there were equal tools in S and S^*, it would also minimize C_2, since the η_i's are non-increasing in i, and as i increases, the effect of η_i on C_2 also increases. Hence, if there were equal number of tools in S and S^*, S would be optimal.

So, based on the above observation, we can say that the part of C_2^S for $i \leq K_{S^*}$ is less than C_2^S, i.e. if we define,

$$C_2^S = C + \Delta$$

where

$$C = \sum_{i=1}^{K_{S^*}} (i-1)\eta_i^S T_c$$

$$\Delta = \sum_{i=K_{S^*}+1}^{K_S} (i-1)\eta_i^S T_c = \sum_{i=1}^{D} (K_{S^*} + i - 1)\eta_{K_{S^*}+i}^S T_c$$

and

$$D = K_S - K_{S^*}$$

Then $C < C_2^{S^*}$. This means, we can write $C_2^{S^*} = \theta C$, where $\theta > 1$
Consequently, the performance ratio would be as follows:

$$\rho \rightarrow \frac{C_2^S}{C_2^{S^*}}$$

$$\begin{aligned}\frac{C_2^S}{C_2^{S^*}} = \frac{C+\Delta}{C_2^{S^*}} &= \frac{\frac{C_2^{S^*}}{\theta} + \Delta}{C_2^{S^*}} \\ &= \frac{1}{\theta} + \frac{\Delta}{C_2^{S^*}} \\ &= \frac{1}{\theta} + \frac{\sum_{i=1}^{D}(K_{S^*} + i - 1)\eta_{K_{S^*}+i}^S T_c}{\sum_{i=1}^{K_{S^*}}(i-1)\eta_i^{S^*} T_c}\end{aligned}$$

Since $\eta_{K_{S^*}+1}^S \geq \eta_{K_{S^*}+i}^S$;

$$\frac{1}{\theta} + \frac{\sum_{i=1}^{D}(K_{S^*} + i - 1)\eta_{K_{S^*}+i}^S T_c}{\sum_{i=1}^{K_{S^*}}(i-1)\eta_i^{S^*} T_c} \leq \frac{1}{\theta} + \frac{\sum_{i=1}^{D}(K_{S^*} + i - 1)\eta_{K_{S^*}+1}^S}{\sum_{i=1}^{K_{S^*}}(i-1)\eta_i^{S^*}}$$

$$= \frac{1}{\theta} + \frac{\eta_{K_{S^*}+1}^S \sum_{i=1}^{D}(K_{S^*} + i - 1)}{\theta \sum_{i=1}^{K_{S^*}}(i-1)\eta_i^S}$$

$$\leq \frac{1}{\theta} + \frac{\eta^S_{K_{S^*}+1} \sum_{i=1}^{D}(K_{S^*} + i - 1)}{\theta \sum_{i=1}^{K_{S^*}}(i-1)\eta^S_{K_{S^*}}} \qquad (\text{since } \eta^S_i \geq \eta^S_{K_{S^*}})$$

$$\leq \frac{1}{\theta} + \frac{\eta^S_{K_{S^*}+1} \sum_{i=1}^{D}(K_{S^*} + i - 1)}{\theta\phi\eta^S_{K_{S^*}+1} \sum_{i=1}^{K_{S^*}}(i-1)} \qquad (\phi \geq 1)$$

$$\leq \frac{1}{\theta} + \frac{\sum_{i=1}^{D}(K_{S^*} + i - 1)}{\theta\phi \sum_{i=1}^{K_{S^*}}(i-1)}$$

$$\leq \frac{1}{\theta} + \frac{DK_{S^*} + D(D-1)/2}{\theta\phi(K_{S^*}(K_{S^*}-1)/2)}$$

$$\leq \frac{1}{\theta} + \frac{2DK_{S^*} + D^2 - D}{\theta\phi(K^2_{S^*} - K_{S^*})}$$

We can find a bound on D using the bounds for bin packing problem. When we make the SPT order, the bin-packing aspect of the problem, which determines the number of tools used, is solved by NFI (next fit increasing) method. This means, starting with item 1 place it in bin 1. When packing item j, put it in the highest indexed nonempty bin if possible, otherwise, place it in a new bin. The items are packed in the order of increasing sizes. SPT order exactly applies this method, when allocating tools to jobs. So, we can use the absolute bound for NFI (= 1.75) given by Anily et al. [3] in order to find a bound for K_S. Actually, the optimal schedule may not minimize number of tools, the optimal result of bin packing problem is a lower bound for K_{S^*}. So, we can say;

$$\frac{K_S}{K_{S^*}} \leq 1.75$$

then

$$D = K_S - K_{S^*} \leq 0.75 K_{S^*}$$

So, putting $(0.75.K_{S^*})$ instead of D in the last line of the previous equations, we get,

$$\frac{C_2^S}{C_2^{S^*}} \leq \frac{1}{\theta} + \frac{2.0625K^2_{S^*} - 0.75K_{S^*}}{\theta\phi(K^2_{S^*} - K_{S^*})}$$

where $\theta > 1$, and $\phi \geq 1$.

This is the final form of the bound we get, which is in fact a loose one. It seems that when $K_{S^*} = 2$, $\rho < 4.375$. As K_{S^*} increases this value decreases, and when $K_{S^*} \to \infty$, $\rho < 3.0625$. Hence, we conclude that the performance ratio for SPT is absolutely less than 4.375.

3.2.3 Conditions for optimality of SPT

As mentioned before, it is obvious that when $Tc = 0$, the SPT rule gives an optimal sequence. We searched for other conditions which guarantee optimality of SPT schedule even when $Tc \to \infty$. Among them some are trivial such as:

- $p_q \geq \frac{TL}{2} \quad \forall q$

  In this case, any non-SPT schedule would be worse than SPT schedule since the number of tools used cannot be decreased. If $p_q > \frac{TL}{2}$, only one job can be assigned to a tool. If there are more than one jobs such that $p_q = \frac{TL}{2}$, SPT schedule would assign them to one tool, whereas a non-SPT schedule may miss this opportunity. Hence, we conclude that in this case SPT rule gives the optimal schedule independent of Tc value.

- $K_S \leq 2$

  In this case, $K_{S^*} \leq 2$ must also hold. Because, SPT sequence already has the maximum number of jobs in the first block (hence minimum in the second block, which contribute to C_2^S), and if 2 tools are enough, there will be no decrease in C_2^σ by using one more tool in order to allocate at least as many jobs as η_2^S. Since we know that SPT also minimizes C_1^σ, we conclude that SPT schedule is optimum in this case.

In addition to these, we can find a bound on Tc below for which the SPT rule gives an optimal sequence.

To find a maximum Tc value below for which the SPT schedule will be optimal, we have considered an approach of comparing the cost components of the SPT schedule with any other non-SPT schedule. Here the absolute bound for worst case performance of SPT, found as 4.375, is used which is the only absolute bound we have. We did not use the asymptotic value of 3.0625, because the value obtained at the end of these computations is asymptotically going to zero.

To find a Tc value, x, below for which SPT is optimal may have two meanings:

1) if $Tc \leq x \Longrightarrow$ SPT is optimum
2) if SPT is optimum $\Longrightarrow Tc \leq x$

What we are looking for is the first one. So, we must find a minimum value for the right hand side of the equation (1) (whereas a maximum value would be needed for the second part). Below is the calculations for the first part.

Factoring out the Tc value from C_2, we redefine the cost components with a slight modification as follows:

$$C_1^\sigma = \sum_q (n-q+1)p_{[q]}$$

$$C_2^\sigma = \sum_{i=1}^{K_\sigma}(i-1).\eta_i^\sigma$$

Then total flow time is:

$$Z = \sum C_j = C_1 + C_2.Tc$$

Assume that there exists a schedule other than SPT schedule, which gives the optimal objective value. In this case, for SPT schedule to be optimal the following must hold:

$$S \text{ is optimal} \Longleftrightarrow C_1^S + C_2^S Tc \leq C_1^{S^*} + C_2^{S^*} Tc$$

$$\Longleftrightarrow (C_2^S - C_2^{S^*})Tc \leq C_1^{S^*} - C_1^S$$

$$\Longleftrightarrow Tc \leq \frac{C_1^{S^*} - C_1^S}{C_2^S - C_2^{S^*}} \qquad (1)$$

$$\frac{C_1^{S^*} - C_1^S}{C_2^S - C_2^{S^*}} > \frac{4.375(C_1^{S^*} - C_1^S)}{3.375 C_2^S} \qquad \text{(since } C_2^S/C_2^{S^*} < 4.375\text{)}$$

$$> \frac{4.375\Delta^p_{min}}{3.375\sum_{i=1}^{K_S}(i-1).\eta_i^S} \qquad (2)$$

$$> \frac{4.375\Delta^p_{min}}{3.375\sum_{i=1}^{K_S}(i-1).\eta_2^S} \qquad \text{(since } \eta_2^S \geq \eta_i^S, \forall i > 1\text{)}$$

$$= \frac{4.375\Delta^p_{min}}{3.375\eta_2^S K_S(K_S-1)/2}$$

$$= \frac{2.59\Delta^p_{min}}{\eta_2^S(K_S^2 - K_S)} \qquad (3)$$

where Δ^p_{min}=minimum difference in processing times of given jobs (greater than zero).

While writing Equation (2) we tried to find the minimum possible deviation of the total flow time of a non-SPT schedule from an SPT schedule. If only the two adjacent jobs with minimum difference in processing times (not equal to zero) are interchanged, the minimum deviation would occur. It can be calculated as:

$$(n-l)p^S_{[l+1]} + (n-l-1)p^S_{[l]} - [(n-l)p^S_{[l]} + (n-l-1)p^S_{[l+1]}] = p^S_{[l+1]} - p^S_{[l]} = \Delta^p_{min}$$

where l shows the rank of the jobs with minimum difference in processing times. Then, from (1) and (3) we can say that if

$$Tc \leq \frac{2.59\Delta^p_{min}}{\eta_2^S(K_S^2 - K_S)}$$

then SPT schedule is optimum.

3.3 Summary

In this chapter, after giving the definition of our problem together with the underlying assumptions, the characteristics of the problem has been analyzed.

It was shown that the problem is NP-hard even when the number of tools is limited, and in the general case, when there is no limit on the number of tools, the problem becomes strongly NP-hard. Furthermore, the behavior of SPT heuristic, which is very powerful for the scheduling problem with the same objective when there are no tool changes, is investigated. Some example instances are provided to illustrate the different possible results in performance of SPT for this problem. Then, it was shown that the worst case performance of the SPT heuristic is bounded above by a constant, which is asymptotically equal to 3.0625, and absolutely equal to 4.375. Using the information we have about the characteristics of SPT heuristic, finally, some conditions which assures optimality of SPT rule are given. These cases are as follows:

- $p_q \geq \frac{TL}{2} \quad \forall q$
- $K_S \leq 2$
- $Tc \leq \frac{2.59\Delta^P_{min}}{\eta^S_2(K^2_S - K_S)}$

In the next chapter, the algorithms proposed for solution of this problem will be explained.

Chapter 4

The Algorithms

In the previous chapter, the problem is defined with some further analysis in order to understand its characteristics. We have shown that problem of flowtime scheduling with tool changes is strongly NP-hard when there is no limit on the number of tools used. Moreover, the problem turns out to be NP-hard even when the number of tools is fixed. For this reason, no algorithm can be proposed for solving the problem optimally in polynomial time. Hence, it is justifiable to try heuristic methods to solve our problem. We have analyzed the performance of the SPT heuristic, which first comes to mind for the flowtime scheduling problem, in the previous chapter. It was shown to have a constant worst case performance bound.

The problem is NP-hard when tool number is fixed, so a pseudo-polynomial dynamic programming formulation will be given for this case. In addition, several heuristic procedures will be developed which are expected to perform better than SPT. Among them, there are static and dynamic dispatching rules, a single-pass procedure and two local search algorithms. In this chapter, the solution procedures including the above mentioned algorithms will be defined. In §4.1 the dynamic programming formulation will be given. In §4.2 the proposed heuristic methods will be explained in detail. Finally in section §4.3, the algorithms will be illustrated on an example problem.

4.1 Dynamic Programming Algorithm

Dynamic programming is basically a complete enumeration scheme which attempts to minimize the amount of computation to be done, with a divide-and-conquer approach. The approach solves a series of subproblems, depending on the choice of forward or backward programming, until it finds the optimal solution to the original problem. For each subproblem the optimal solution and its contribution to the objective function are determined. The solution to the subproblem is found by utilizing all the information obtained before solving the previous subproblems.

To solve the scheduling problem with tool changes we have developed a forward dynamic programming formulation. For the general case, when tool number is not fixed, this algorithm can be applied for $m = \lceil \frac{\sum_{q=1}^{n} p_q}{TL} \rceil$ to K_S, optimal being the minimum value obtained from those calculations. There is no need to carry on calculations beyond $m = K_S$, because no optimal schedule will use more tools than SPT does.

For a given number of tools denoted as m, the algorithm is as follows:

Reindex the jobs in SPT order. At stage k, when we are scheduling job k, define:

$t_j =$ total processing time assigned to the jth tool

$\eta_j =$ the number of jobs on tool j

$f_k(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m) = \text{Minimum} \sum_q C_q$ realizable at stage k

Initial conditions are given as:

$$f_0(0, \cdots, 0; 0, \cdots, 0) = 0$$

$$f_0(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m) = \infty \qquad \text{for } t_i \neq 0 \quad \text{and } \eta_i \neq 0$$

The recursive function of dynamic programming is given below:

$$f_k(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m) = \min_{1 \leq j \leq m}[f_{k-1}(t_1, \cdots, t_j - p_k, \cdots, t_m; \eta_1, \cdots, \eta_j - 1, \cdots, \eta_m)$$
$$+ \sum_{i=1}^{j} t_i + (j-1)Tc + \sum_{i=j+1}^{m} \eta_i p_k]$$

For any stage k, the feasibility requirements for a state are as follows:

(1) $p_k \leq t_j \leq TL$ and $1 \leq \eta_j \leq k$

(2) $\sum_{j=1}^{m} t_j = \sum_{i=1}^{k} p_i$

(3) $\sum_{j=1}^{m} \eta_j = k$

If a state $(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m)$ is infeasible, then set:

$$f_k(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m) = \infty$$

Optimal solution is $\min f_n(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m)$ over all $(t_1, \cdots, t_m; \eta_1, \cdots, \eta_m)$. If this is equal to infinity, we conclude that given problem is infeasible.

This algorithm considers assigning job k to feasible tools at stage k. For each possible assignment, state of former case is found as $(t_1, \cdots, t_j - p_k, \cdots, t_m; \eta_1, \cdots, \eta_j - 1, \cdots, \eta_m)$, and the increase in the objective value with this assignment is added to the objective value of the former case. Thus, after trying all possible tools for kth job, the minimum objective value realizable at stage k is found.

Since memory is very critical for the success of any dynamic programming algorithm, we employ a base representation for the states in order to save from memory space. Actually, each state is represented by two vectors of length m, the total processing time assigned to tools and the number of jobs on the tools. In order to decrease the need for memory space, we can represent each of these vectors by an integer, considering the vector as the equivalent of this number in base TL and η_1^S respectively. For example let m be 3. Then, a state $(t_1, t_2, t_3; \eta_1, \eta_2, \eta_3)$ is represented by (t, η) where
$t = TL^0 t_1 + TL^1 t_2 + TL^2 t_3$ and

$\eta = nmax^0\eta_1 + nmax^1\eta_2 + nmax^2\eta_3$
where nmax is equal to η_1^S, that is the maximum number of jobs that can be processed with a tool.

Thus every state can be represented by a unique pair of numbers (t, η) instead of two m vectors, which would be helpful in implementation of the algorithm.

Lemma 4.1: The proposed dynamic programming algorithm is pseudo-polynomial with complexity $O(nm(\frac{\sum_{q=1}^{n} p_q}{m})^m(\frac{n}{m})^m)$.

Proof: Consider stage k of the algorithm. For each possible state in this stage, m alternatives are compared depending upon which tool the kth job is scheduled on. A state is described by total processing time on tool j for all $j = 1 \cdots m$ and total number of jobs on tool j for $j = 1 \cdots m$. The number of possible states for an alternative at this stage is found as follows:

number of possible states= $\prod_{j=1}^{m} t_j \cdot \prod_{i=1}^{k} \eta_i$

subject to:

$$\sum_{j=1}^{m} t_j = \sum_{i=1}^{k} p_i$$
$$\sum_{j=1}^{m} \eta_j = k$$

It is known from optimal subdivision problem's solution that the given number of possible states takes its maximum value when $t_j = \dfrac{\sum_{q=1}^{k} p_q}{m}$ and $\eta_j = \frac{k}{m}$. Hence, the maximum possible number of states for an alternative at stage k is found as:

$$(\frac{\sum_{q=1}^{k} p_q}{m})^m \cdot (\frac{k}{m})^m$$

Letting k be n, this value is bounded by

$$(\frac{\sum_{q=1}^{n} p_q}{m})^m \cdot (\frac{n}{m})^m$$

Then, the total computations at stage k for m alternatives is bounded by

$$m \cdot (\frac{\sum_{q=1}^{n} p_q}{m})^m \cdot (\frac{n}{m})^m$$

Since there are n stages, the overall computations are bounded by;

$$n \cdot m \cdot (\frac{\sum_{q=1}^{n} p_q}{m})^m \cdot (\frac{n}{m})^m$$

This is pseudo-polynomial when m is fixed.□

Using this dynamic programming algorithm we can find an optimal solution. However it requires so much computation time even for small problem sizes so it would not be useful for practical purposes. We have developed some heuristic methods to solve the problem, and decided to use the dynamic programming solution as a benchmark when possible.

4.2 Heuristic Algorithms

As mentioned before, NP-completeness of the problem justifies heuristic approaches, which would provide good solutions with reasonable computation times. We have basically worked on three different types of heuristic methods for the tool change problem. The first approach is using dispatching heuristics. These procedures are simple ones that are easy to understand and implement, second approach is a construction algorithm. Finally, we develop two local search algorithms, which would search over the space using some defined rules while trying to improve the initial solution. In this section these algorithms will be discussed. Throughout this section, the terms "block" and "tool" are used interchangeably, referring to the block representation of the tool change problem.

In all the heuristic algorithms we have developed, the first sequence found by applying the particular algorithm is a temporary one, which basically shows the assignments of jobs to blocks. On this sequence, some minor reprocessing must be done. This is a consequence of the information about the properties of an optimal schedule we stated in the previous chapter. Having known these properties, we first determine the sequence using the rule and do the assignments to tools. Afterwards, we resort the blocks, and jobs in a block, to fit to those properties explained in §2.2. We can briefly restate these properties as, "jobs sharing the same tool should be in SPT order", and "the tools should be ordered in nonincreasing order of the number of jobs using it". Throughout the section, this issue is referred as "checking the structural properties, then reordering the jobs and blocks if necessary", as the last step of all the algorithms proposed.

4.2.1 Dispatching Heuristics

Dispatching rules can be classified as *static* and *dynamic* rules. Static rules are the ones that are not time dependent. They are just a function of the problem data, and the order of jobs can be determined by applying the rule once, at the beginning. Dynamic rules are time dependent, hence they must be revised every time a job is scheduled. In this section, we will present four dispatching heuristics, namely shortest processing time, first fit decreasing, modified first fit decreasing and expected gain index. The first two are well known rules for other problems, whereas the last two are developed primarily for tool change problem.

Shortest Processing Time (SPT)

Shortest processing time (SPT) is one of the oldest, and best known dispatching rules in the scheduling theory. It gives an optimal sequence for the total flowtime problem, $1||\sum C_j$. But as discussed in the previous chapter, for tool change problem, it may not perform as well, while minimizing the first part of the objective function. On the other hand, SPT has a constant worst case performance ratio for our problem, and depending on the problem parameters it may perform quite well. The other heuristic algorithms are proposed in order to improve over the performance of the SPT rule.

SPT is the easiest rule to apply, only thing to be done is to order the jobs in nondecreasing order of their processing times. Then, without changing the sequence, jobs are assigned to tools to determine the times of tool changes. Note that, the resulting sequence would not need resorting.

First Fit Decreasing (FFD)

When the tool change times are very large, the second part of the objective function dominates, and the tool change problem gets closer to a bin packing problem. First-fit-decreasing (FFD) is one of the well known heuristics for bin packing problem. It has shown to have a worst case performance bound of 1.222 [15]. The algorithm adapted to tool change problem can be described as follows:

Step 1. Sort the jobs in nonincreasing order of processing times (in LPT order).

Step 2 Starting from the beginning, assign each successive job to the first available tool.

Step 3 Resort the blocks such that $\frac{t_i+Tc}{\eta_i} \leq \frac{t_j+Tc}{\eta_j}$ for $i < j$.

Step 4 Sort the jobs in each block in SPT.

This algorithm tries to use minimum number of tools, maintaining a balance in the used tool lifes, which are t_j values of the tools. For the tool change problem, it is expected that FFD would result in good solutions when the Tc value is very high, since then the C_2^σ would dominate our objective, which represents the bin-packing aspect of the problem.

Modified First Fit Decreasing (MFFD)

This algorithm combines the SPT and FFD rules to improve over the performance of these rules. The main motivation of this rule is to get benefit of the fact that no tool change time is added to the flowtimes of jobs using the first tool. SPT maximizes the number of jobs using the first tool. It can be beneficial to fill in the first block with the shortest jobs, so that the number of jobs being affected by the tool changes will be minimum.

In MFFD algorithm, we first order the jobs in SPT order until the first block is full. Then, we turn our attention to balancing the loads of the tools so as to minimize the tool usage. After all jobs are assigned to blocks, they are resorted to fit to the properties of an optimal schedule. This algorithm would also have less C_1^σ value than FFD algorithm has, because it assigns more shorter jobs to earlier blocks, that is it is closer to SPT. Formally, the algorithm is as follows:

Step 1. Sort the jobs in nondecreasing order of processing times (in SPT order).

Step 2 Starting from the beginning, assign each successive job to the first tool until no more can be assigned to it.

Step 3 Apply FFD algorithm to the remaining jobs.

Step 4 Resort the blocks such that $\frac{t_i+Tc}{\eta_i} \leq \frac{t_j+Tc}{\eta_j}$ for $i < j$.

Step 5 Sort the jobs in each block in SPT.

A Ranking Index Based Heuristic

In this section we will explain a dynamic dispatching rule proposed for the flowtime scheduling with tool changes problem. The rule is dynamic in the sense that an index has to be calculated for all the unscheduled jobs at each stage, although there is no time dependency. At each stage k, the jobs will be ranked according to this index, which represents the expected decrease in the total flowtime by scheduling that job to kth position, that is the expected gain. The jobs will be ordered in SPT at the beginning, and their indexes in this order will be kept, to be used in further steps in order to calculate the expected gain index (EGI). The index will be used to find an initial sequence. Afterwards, the final schedule will be found applying first fit algorithm to make tool assignments, and resorting if necessary after checking the structural properties defined before. First fit is a well known algorithm for bin packing problems as explained before. It is essentially the same as FFD except that the initial sorting in nonincreasing order of p_j's is skipped.

The index EGI_{qk} for job q at stage k is defined as:

$$EGI_{qk} = (p_q - p_{min})[\frac{Tc}{TL} - ([q] - k)]$$

where $[q]$ is the SPT index of job q, and p_{min} is the minimum processing time among the remaining jobs.

Now, we will briefly explain the logic behind this index. For a given SPT schedule, assume that there is a pair of jobs (q, r) $(q < r)$ such that their interchange is feasible with respect to tool life constraints. Hence we can swap these jobs without increasing the tool requirements. If we make this interchange, the increase in flowtime which is reflected in C_1^σ value can be calculated as follows:

$$\begin{aligned}\Delta C_1^S &= \textstyle\sum_{l \neq q,r}(n - l + 1)p_l + (n - q + 1)p_r + (n - r + 1)p_q \\ &- \textstyle\sum_{l \neq q,r}(n - l + 1)p_l + (n - q + 1)p_q + (n - r + 1)p_r \\ &= (p_r - p_q)(r - q)\end{aligned}$$

On the other hand, by moving a larger job to an earlier block, we have some saving in the tool usage of the later block, which is expected to decrease the C_2^σ portion of the objective function by allowing another job move to earlier blocks. This possible gain is represented as:

$$\Delta C_2^S = \frac{(p_r - p_q)}{TL} Tc$$

Finally, we can find the expected decrease in the overall objective as:

$$\begin{aligned}\Delta Z_S &= \Delta C_2^S - \Delta C_1^S \\ &= (p_r - p_q)[\tfrac{Tc}{TL} - (r - q)]\end{aligned}$$

While scheduling the kth job, if we were applying SPT, the job with $p_r = p_{min}$ would be assigned to kth position, whereas the position of job q would be $[q]$. Now, we get the expected gain from the cost by assigning job q to the kth position from the above equation for ΔZ_S using $q = k$, $p_q = p_{min}$, $r = [q]$ and $p_r = p_q$. This is certainly a rough approximation to the change in the objective value with this decision. Because while job q is assigned to kth position, the job with minimum processing time is not assigned to the $[q]th$ position, so there is not a real interchange. Furthermore, the expected gain from C_2^q is also only a rough representation of our expectation. Note that this calculation should only be done for the jobs that can be put into the current tool at that stage, without increasing the tool requirement.

Having explained the calculation of our ranking index, we can define the algorithm using this index as follows:

Step 1 Order the jobs according to shortest processing time rule. Determine the SPT index $[q]$ for each job q.

Step 2 Order the unscheduled jobs in SPT and for each job if it can fit to the current tool, calculate its index EGI_{qk}, else assign a very small value to its index.

Step 3 Select the job with highest index (in ties choose shorter job) to be scheduled as the next job. If all the jobs are not scheduled yet, Goto Step 2.

Step 4 Find the block assignments using first fit algorithm.

Step 5 Order the blocks such that $\frac{t_i + Tc}{\eta_i} \leq \frac{t_j + Tc}{\eta_j}$, for $i < j$.

Step 6. Order the jobs in the blocks such that $p_q \leq p_r$ for $q < r$.

The advantage of using this index is incorporating the information of $\frac{Tc}{TL}$ value to the solution. When this ratio is large, the algorithm favors larger jobs, so the sequence found at the end of Step 3 will be closer to LPT. Thus, the bin packing aspect of the problem will be solved better, since best bin packing algorithms like FFD uses LPT ordered sequence. On the other side, when $\frac{Tc}{TL}$ is small the algorithm favors smaller jobs, and the sequence found will be closer to SPT sequence.

The algorithm is applied for two steps for a small example problem below:

Let $n = 5$ $Tc = 36$, $T_L = 12$, and the processing times are given as:

l	1	2	3	4	5
p_l	5	3	8	6	10

The algorithm will proceed as follows:

Step 1 Sort the jobs in SPT order.

$[l]$	1	2	3	4	5
p_l	3	5	6	8	10

Step 2 The unscheduled jobs are already sorted in SPT order. $p_{min} = 3$

k = 1 $t_1 = 0$

$p_1 = 3 \Rightarrow$ feasible to fit the tool
$EGI_{11} = (3-3)[\frac{36}{12} - 1 + 1] = 0$

$p_1 = 5 \Rightarrow$ feasible to fit the tool
$EGI_{21} = (5-3)[\frac{36}{12} - 2 + 1] = 4$

$p_1 = 6 \Rightarrow$ feasible to fit the tool
$EGI_{31} = (6-3)[\frac{36}{12} - 3 + 1] = 3$

$p_1 = 8 \Rightarrow$ feasible to fit the tool
$EGI_{41} = (8-3)[\frac{36}{12} - 4 + 1] = 0$

$p_1 = 10 \Rightarrow$ feasible to fit the tool
$EGI_{51} = (10-3)[\frac{36}{12} - 5 + 1] = -7$

Step 3 Job 2 has the highest index, so assign it to the first position.

Step 2 k = 2, $t_1 = 5$

$p_1 = 3 \Rightarrow$ feasible to fit the tool
$EGI_{12} = (3-3)[\frac{36}{12} - 1 + 2] = 0$

$p_1 = 6 \Rightarrow$ feasible to fit the tool
$EGI_{32} = (6-3)[\frac{36}{12} - 3 + 2] = 6$

$p_1 = 8 \Rightarrow$ not feasible to fit the tool
$EGI_{41} = -999999$

$p_1 = 10 \Rightarrow$ not feasible to fit the tool
$EGI_{52} = -999999$

Step 3 Job 3 has the highest index, so assign it to the second position.

We proceed this way, until all the jobs are scheduled. After that, assignments of jobs to tools is done. Finally, the structural properties are checked and if it is necessary the order is changed.

4.2.2 Knapsack Heuristic

The next heuristic algorithm is a single-pass type procedure, which jointly uses SPT rule and solution of a knapsack problem. Our intuition is that, SPT rule must be applied to some extent for solving tool change problem. Because we know that it minimizes one part of the objective function. When SPT performs badly, it is mostly because of the underutilized blocks through the later portions of the schedule. Since larger jobs come together at the end of the sequence, usually few of them can fit to a block, leaving much space (tool life) unused. Thus the number of tools needed increases, which consequently leads to an increase in C_2^S. This can be avoided by moving some large jobs to earlier blocks, hence providing more flexibility for the last blocks. As explained before, this was also the underlying idea of calculations for ranking index.

To achieve our goal, we considered applying SPT until a predetermined amount of tool life is used, afterwards solving a knapsack problem to use the remaining tool life as efficiently as possible. In order to favor assignment of shorter jobs in case there are alternative solutions with same tool usage, the objective is written as the sum of the number of jobs assigned and the total processing times of the assigned jobs. Thus the objective function is: $\sum_{q=1}^{N} x_q + \sum_{q=1}^{N} p_q x_q$. Then the weight of item q in the objective function of the knapsack problem becomes $(1 + p_q)$. In fact, we have tried giving weights to the components of the objective function. However, it turned out to have no significant effect on the performance of the algorithm. Hence, we decided to ignore the weights in the objective function.

There is one parameter of this algorithm that has to be decided at the beginning. γ is a real number less than 1, which is used to determine when to stop SPT ordering. We schedule jobs in SPT until at most $\gamma \cdot T_L$ amount of the tool life is used. Suppose there are N unscheduled jobs at an instant of the algorithm, and the remaining tool life is RL. Note that RL can be greater than $(1-\gamma)T_L$. Then the mathematical programming formulation for the knapsack problem is given as:

$$\text{Minimize } \sum_{q=1}^{N} (1 + p_q) x_q$$

$$\text{Subject to:}$$

$$\sum_{q=1}^{N} p_q x_q \leq RL$$

where x_q is the $0-1$ binary decision variable which is equal to 1 if job q is assigned to the current block, for which we are solving the above mathematical programming problem.

The algorithm can be defined step by step as follows:

Step 1 Sort the jobs in SPT order.

Step 2 Schedule the jobs successively from the sorted list as long as the total used tool life is not more than $(\gamma \cdot TL)$.

Step 3 Solve the knapsack problem with the remaining jobs to fill up the block.

Step 4 If there are any unscheduled jobs goto step 1. Else, check the structural properties, and reorder the jobs and blocks if necessary.

This algorithm is expected to give better results than SPT algorithm does, when Tc value is high.

4.2.3 Local Search Algorithms

Local search approaches have two basic elements. One is the concept of a neighborhood of a solution and other is a mechanism to generate neighborhoods. The generating mechanism is a method of taking one sequence as a seed and systematically creating related sequences. The search algorithm takes the seed, generates neighborhood solutions using the generating mechanism and thus try to find a better result.

There are various search procedures used for scheduling problems. Simulated annealing, tabu search and genetic algorithms are some most popular search algorithms. We have studied two local search algorithms, one generates neighborhoods via knapsack problem solutions, the other one is a genetic algorithm using problem space search. These two algorithms are explained in detail in this section.

Two Bin Heuristic

The first local search algorithm we will discuss is called two bin heuristic. This is a local search procedure which takes SPT schedule as the initial seed and tries to improve over it. In this algorithm, the knapsack problem formulation will be used again. Our motivation is same with the knapsack heuristic. SPT solution is tried to be improved by getting more use of the tool lifes.

For generating new schedules, the main idea is reordering a two block length portion of the sequence so as to make it fill one of the blocks and leave more space in the other. At every iteration, two blocks are chosen randomly and a knapsack problem is solved for the jobs in these two blocks. As a result of the knapsack solution, these jobs are re-partitioned into two blocks. Hence we obtain a different schedule. This procedure continues for a fixed number of iterations. The best solution obtained at the end of this search is reported as the result of the algorithm. At every iteration, the objective value is calculated and compared with the best value obtained until that time, keeping the best schedule throughout the search.

The knapsack formulation used in this algorithm is similar to the one used in knapsack heuristic. But, this time the objective value is a weighted sum of the number of jobs and the total processing time assigned, so the weight for item q in the knapsack problem is $(w_1 + w_2 p_q)$. Moreover, in this case knapsack size is the whole tool life, not just a fraction of it.

We define w_1 as the weight of the number of jobs in the objective and w_2 as the weight of the sum of processing times assigned. Suppose two blocks are chosen randomly with totally N jobs in them. Then the problem to be solved would be formulated as follows:

$$\text{Minimize } \sum_{q=1}^{N} (w_1 + w_2 p_q) x_q$$

Subject to:

$$\sum_{q=1}^{N} p_q x_q \leq T_L$$

where x_q is the $0-1$ binary decision variable for assigning the jobs to a block. The jobs with $x_q = 1$ are assigned to one of the blocks while the others are assigned to the second block. The weights w_1 and w_2, which are both greater than zero, should be determined before the algorithm is applied.

The two bin algorithm is defined step by step as follows:

Step 1 Sort the jobs in SPT order. Calculate the objective function value and assign best value to SPT objective, best schedule to SPT schedule.

Step 2 Select two blocks randomly. Solve the knapsack problem defined above for these jobs.

Step 3 Rearrange these two blocks according to the solution of the knapsack problem.

Step 4 Reorder the block to fit into the structural properties. Calculate the objective value. If it is less than the best value assign best value to current objective value and best schedule to current schedule. If the number of iterations have not exceeded the predetermined number, increase number of iterations by one and goto Step 2.

Genetic Algorithm with Problem Space Search (GAPS)

As the last algorithm, we applied a recent search technique called problem space search combined with genetic algorithm.

Problem space search (PSS) algorithms are fundamentally local search heuristics, but are entirely different than most current applications of simulated annealing and tabu search which presume neighborhood structures based on interchanges (swaps) of combinatorial elements. The unifying feature of PSS algorithms is an implicit, underlying constructive algorithm upon which a search space is defined. A problem space search heuristic requires an initial feasible solution, a base heuristic, and a neighborhood definition. One of the most important features of the problem space search is that neighboring solutions are generated by first perturbing the problem data and then applying the base heuristic to the perturbed data. This, in turn, allows the base heuristic to generate alternative solutions. The cost of each alternative solution has to be determined using original problem data. Thus, a search space can be formed given a base heuristic and a specific problem perturbation method.([23], [26])

The problem space approach explicitly defines a neighborhood structure to which systematic search can be applied to seek the optimum. That is, rather than simply using perturbations to randomly generate alternative solutions, problem space provides neighborhood structure which can be exploited by search algorithms. To perform search in the problem space, different procedures can be used such as hill climbing, steepest descent, genetic algorithm or simulated annealing, etc. If only slight perturbations to the original problem data are made, it seems reasonable that "good" solutions will be generated by the base algorithm. That is, it is expected (and indeed have verified on numerous occasions) that perturbed problem data configurations in the vicinity of the original problem will, in general, map to good solutions. The ability to generate neighborhoods populated primarily by good solutions accounts for much of the success of the approach.

In this study, we have applied genetic algorithm with problem space search. As the base heuristic, we have chosen the dispatching heuristics SPT and FFD, which are expected to perform well in extreme values of Tc. The algorithm applies problem space search using both of the rules as a base heuristic independently, then gives the best result obtained.

Genetic algorithm is a local search technique stemming from the theory of evolution. The evolution is based on two random processes, mutation and crossover. The chromosomes in the gene pool randomly mates, and their offspring is added to the gene pool. Throughout the process, the stronger genotypes will survive as the theory of natural selection suggests. In some cases the chromosomes can mutate by some external effect. The genetic algorithms use all these ideas to generate different alternatives and search among the alternatives. In case of the problem space search, the chromosomes represent the perturbation vectors, and genes represent the perturbation amount for a single job.

Our GAPS heuristic starts with generating an initial population of perturbation vectors, called seeds. These are vectors of n random real numbers assigned within a specified range, which shows the perturbation amount to be applied on each job. The objective value corresponding to a perturbation vector is calculated in three steps. First, the processing times are perturbed by that amount. Second, with the new processing times, the base heuristic is applied. Then the information on resulting sequence is passed to the original problem data and objective value of that sequence is calculated using original processing times.

At each iteration of the genetic algorithm, the parents are selected randomly from the population. This is done by tournament selection method, in two steps. First, two seeds are chosen randomly and the one with better objective is chosen as mother seed. Then this procedure is repeated to select the father seed. After that, these two seeds are crossed to get a new offspring. Crossing is done by first determining a crossover point randomly, and then taking the genes of the mother vector until the crossover point and from the dad after the crossover point to combine them. This child may have mutation with some small probability, that is the perturbation amounts can be recalculated with a different random range of perturbation. Then, the objective value corresponding to this new child is calculated. Finally the child is added to the population, replacing the one having the worst objective. This process is repeated for the desired number of iterations.

In order to apply this algorithm some parameters have to be determined at the beginning. These are the *population size*, *perturbation range*, *mutation probability* and *mutation range*, in addition to the number of iterations. Perturbation range shows the interval from which random perturbations will be chosen, where mutation range shows the perturbation range when mutation occurs. Mutation probability is the probability of having mutation for a child. Finally, population size is the number of initial chromosomes generated.

The procedure can be summarized briefly as follows:

Step 1 Generate the initial population of perturbations and find the objective values for each member of the population.

Step 2 Select parents from the population and generate an offspring by crossing-over and mutation.

Step 3 Remove the seed with worst objective value and insert the newly generated offspring to the population. If the desired number of iterations is not reached go to step 2.

4.3 Example Problem

In this section, the heuristic algorithms will be illustrated on a numerical example with 20 jobs. The problem data is given as follows:

$$Tc = 182$$

$$T_L = 108$$

l	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_l	3	13	10	9	11	16	11	13	15	9	6	17	8	14	16	13	13	6	9	3

Now, let us explain briefly how each algorithm will proceed for this problem.

4.3.1 SPT

SPT rule finds the sequence as follows:

l	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_l	3	3	6	6	8	9	9	9	10	11	11	13	13	13	13	14	15	16	16	17

Then the jobs are assigned to tools in the same order. First 12 jobs' total processing time equals 98, and if $13th$ job is added it would be $98 + 13 = 111$ which is greater than the tool life. So, beginning from job 13 the second tool is used. The next 7 jobs processing times add to 100, so the last job with $p_{20} = 17$ cannot be processed with the second tool. Thus a third tool is used for the last job. The resulting schedule has total flowtime equal to 3439 as shown in Figure 4.1.

4.3.2 FFD

FFD algorithm starts with ordering the jobs in LPT, which is just the reverse order of the one done by SPT given above. The initial sequence would be:

l	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_l	17	16	16	15	14	13	13	13	13	11	11	10	9	9	9	8	6	6	3	3

The first 7 jobs can be assigned to first block, using 104 time units. For the $8th$ job a new block is opened, and all other jobs are assigned to second block until job 19. Since $p_{19} = 3$, it can fit to the first block, so it is assigned to the first block. And for the $20th$ job, second block is available. Thus using two tools all the jobs can be finished. At this step, the sequence found in terms of the processing times is as follows, where **Tc** representing tool change instance:

17 16 16 15 14 13 13 3 **Tc** 13 13 11 11 10 9 9 9 8 6 6 3

Then, the structural properties are checked. We see that the jobs in a block are not in SPT order. The second requirement is $\frac{t_1+Tc}{\eta_1} < \frac{t_2+Tc}{\eta_2}$, which is checked as follows:

$t_1 = 107 \quad t_2 = 108 \quad \eta_1 = 8 \quad \text{and} \quad \eta_2 = 12,$

$\Rightarrow \frac{107+182}{8} \not< \frac{108+182}{12}$

So the order of the blocks must also be changed. The final schedule can be

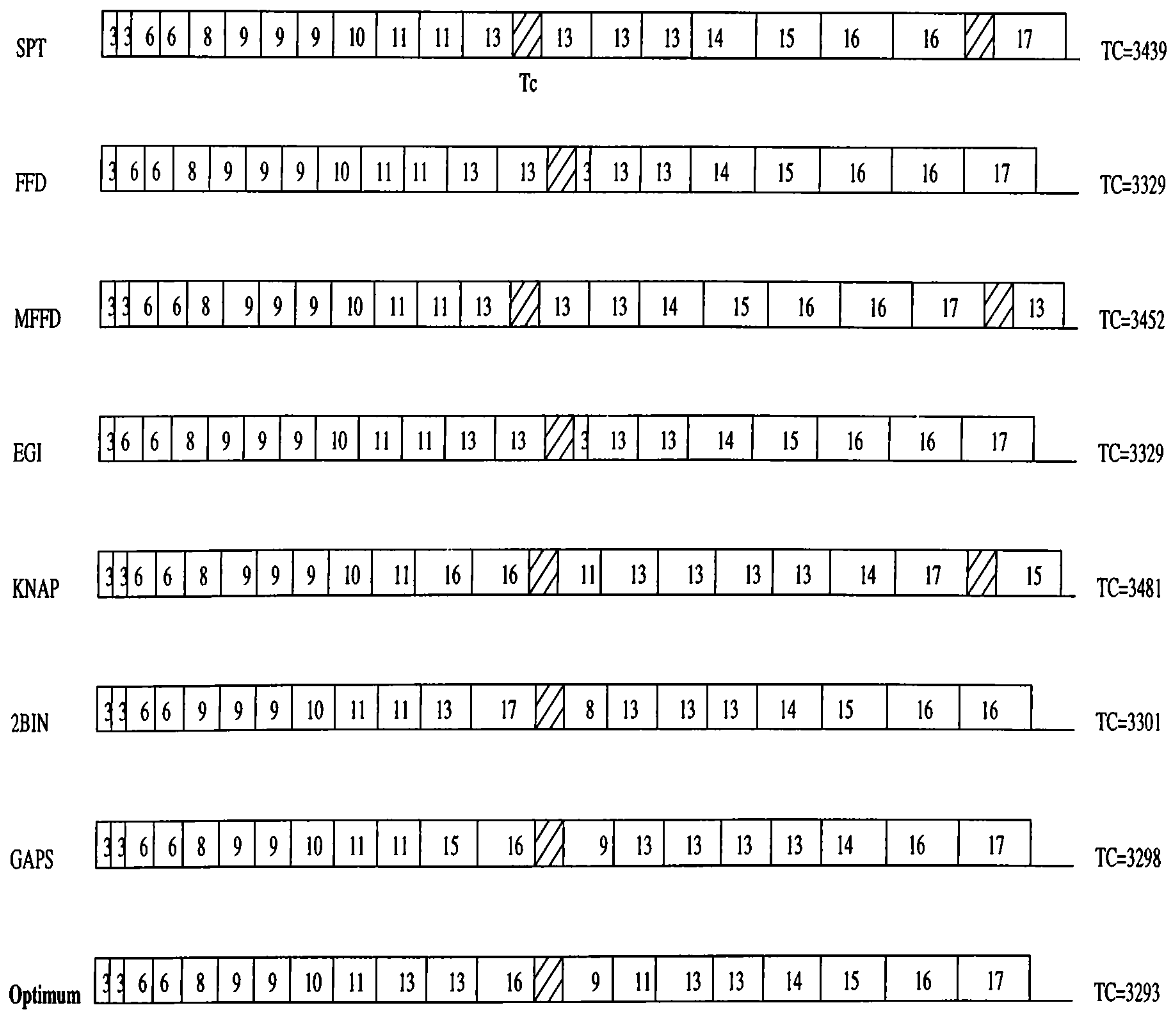

Figure 4.1: The schedules found by different algorithms

seen in figure 4.1 with total flowtime equal to 3329.

4.3.3 MFFD

Since MFFD is a combination of SPT and FFD, the initial sequence will be same as the SPT until the first tool is used up. Then, the remaining jobs are ordered in LPT, which will be same as the first 8 jobs in LPT order given in the FFD section. These jobs can be assigned to two tools. After tool assignments are done, the sequence will be as follows:

3 3 6 6 8 9 9 9 10 11 11 13 **Tc** 17 16 16 15 14 13 13 **Tc** 13

Finally, the structural properties are checked, and the schedule is revised to have the form as in Figure 4.1, having total flowtime as 3452.

4.3.4 Ranking Index EGI

The calculation of EGI_{qk} was explained step by step before. So here, it will not be done again. In this example, the $\frac{T_C}{T_L}$ value is not high, so EGI turns out to be larger for small jobs. As a result, the sequence is found as:

3 6 6 8 9 9 9 10 11 11 13 13 13 13 14 15 16 16 17 3

Then the tool assignments are done using first fit algorithm with this sequence. The schedule before revising according to structural properties is as follows:

3 6 6 8 9 9 9 10 11 11 13 13 **Tc** 13 13 14 15 16 16 17 3

The final schedule can be seen in Figure 4.1. The result of this algorithm turns out to be same with FFD schedule for this example, with total flowtime equal to 3329.

4.3.5 Knapsack Heuristic

The parameter of knapsack heuristic, γ, is chosen to be 0.7 in this example. Knapsack heuristic starts with ordering the jobs in SPT order. Since $(0.7 \cdot T_L) =$ 75.6, first 10 jobs with total processing time of 75 are assigned directly to the first block. The remaining jobs are as follows:

l	1	2	3	4	5	6	7	8	9	10
p_l	11	13	13	13	13	14	15	16	16	17

For this set of jobs the following knapsack problem is solved:

$$\text{Minimize } \sum_{q=1}^{10} x_q + \sum_{q=1}^{10} p_q x_q$$

$$\text{Subject to:}$$

$$\sum_{q=1}^{10} p_q x_q \leq 33$$

$$x_q = 0, 1$$

The solution of this problem is given as $x_8 = x_9 = 1$ all other variables being zero. So, the $8th$ and $9th$ jobs are assigned next to the first block, both of which have processing time of 16.

Then the remaining jobs are sorted in SPT and the procedure is repeated for the second block. Finally the schedule is found as shown in Figure 4.1, with total flowtime equal to 3481.

4.3.6 Two Bin Heuristic

For the two bin algorithm, we have to determine the weights in the objective function of knapsack problem first. We have chosen $w_1 = 0.2$, and $w_2 = 0.8$. This algorithm starts with the SPT schedule, which is given in Figure 4.1. Then, two blocks are chosen randomly. For this particular example, the first and third blocks are chosen at the first iteration. So, the jobs to be rearranged by solving knapsack problem are as follows:

l	1	2	3	4	5	6	7	8	9	10	11	12	13
p_l	3	3	6	6	8	9	9	9	10	11	11	13	17

With this data as input, the following knapsack problem is solved:

$$\text{Minimize } 0.2\sum_{q=1}^{13} x_q + 0.8\sum_{q=1}^{13} p_q x_q$$

Subject to:

$$\sum_{q=1}^{13} p_q x_q \leq 108$$

$$x_q = 0, 1$$

The optimal solution to this problem gives $x_5 = 0$ and all other variables are 1. Thus, job 5 is assigned to one block, and the remaining jobs are assigned to the other block. Now, the three blocks would have the following jobs: (a job is represented by its processing time)

block 1 : 3 3 6 6 9 9 9 10 11 11 13 17
block 2 : 13 13 13 14 15 16 16
block 3 : 8

But note that, before the objective value is calculated the tool assignments for this sequence is revised. It was before the first iteration that there were 3 blocks, but now the job in last block can fit into the second block, so number of blocks needed is decreased with this iteration. Moreover, the jobs and blocks are resorted to obey the structural properties. Then the final schedule after this iteration would be:

3 3 6 6 9 9 9 10 11 11 13 17 **Tc** 8 13 13 13 14 15 16 16

This schedule has total flowtime as 3301, which cannot be improved in the following iterations. Since one of the blocks (the second one) uses all the tool life, it is hard to get a change with the knapsack problem solution after this step. The final schedule is shown in Figure 4.1.

4.3.7 GAPS

The GAPS heuristic involves too much computational work, which we cannot present all here. For this reason, we will just illustrate one iteration of the algorithm when base heuristic is SPT. The given jobs are also ordered in SPT before executing the algorithm. In this example the parameters are chosen as: population size=50, mutation probability=0.1, perturbation range=$(-3.5, 3.5)$, and mutation range=$(-1.75, 1.75)$. The algorithm is applied for 1000 iterations.

First of all a population is generated and the objective values corresponding to each individual in the population is found. Then, from this population two parents are chosen. In our example, the mom and dad vectors were chosen as shown in Table 4.1.

The crossover point is randomly chosen to be 11. Thus, the new offspring will take the first 11 genes from mom, and the remainings from dad. The offspring vector with its corresponding perturbed processing time (represented as pp_i) is shown in Table 4.2.

After this step, the base heuristic, SPT, is applied with this perturbed processing time data, and a new sequence is found as shown in Table 4.3.

So, the initial sequence generated at this step is:

3 6 6 3 9 10 11 8 9 13 9 **Tc** 13 11 13 13 16 14 15 16 **Tc** 17

After revising it according to structural properties, the schedule becomes:

3 3 6 6 8 9 9 9 10 11 13 **Tc** 11 13 13 13 14 15 16 **Tc** 17

This schedule has total flowtime 3443, which is worse than SPT. But this procedure is repeated 1000 times for both base heuristic SPT, and FFD. As a result, the schedule shown in Figure 4.1 is found with total flowtime equal to 3298.

q	mom	dad
1	2.325150	-2.552833
2	-2.196676	0.583532
3	-1.531537	-1.701707
4	-1.539781	1.417378
5	-0.094819	0.880233
6	2.383942	-1.285144
7	-1.089634	-2.582775
8	-1.478643	-1.701620
9	-1.553391	2.875961
10	2.159815	1.279671
11	-1.997042	3.435768
12	0.720116	1.950958
13	-0.007031	1.743355
14	0.830110	-1.436884
15	0.665036	0.285349
16	-1.745548	0.854848
17	-0.420383	3.431830
18	-2.384678	-1.239335
19	2.712744	2.494152
20	2.081396	3.207853

Table 4.1: GAPS-The parent vectors chosen from the initial population

q	offspring	pp_i
1	2.325150	5.325150
2	-2.196676	0.803324
3	-1.531537	4.468463
4	-1.539781	4.460219
5	-0.094819	9.415036
6	2.383942	11.383942
7	-1.089634	9.825670
8	-1.478643	7.521357
9	-1.553391	8.446609
10	2.159815	13.159815
11	-1.997042	9.002958
12	1.950958	11.365580
13	1.743355	14.402755
14	-1.436884	11.563116
15	0.285349	13.587250
16	0.854848	14.854848
17	3.431830	18.431829
18	-1.239335	14.760665
19	2.494152	18.494152
20	3.207853	20.207853

Table 4.2: GAPS-child vector and corresponding perturbed processing times

q	pp_q	corresponding p_q
1	0.803324	3
2	4.460219	6
3	4.468463	6
4	5.325150	3
5	7.521357	9
6	8.446609	10
7	9.002958	11
8	9.415036	8
9	9.825670	9
10	11.365580	13
11	11.383942	9
12	11.563116	13
13	13.159815	11
14	13.587250	13
15	14.402755	13
16	14.760665	16
17	14.854848	14
18	18.431829	15
19	18.494152	16
20	20.207853	17

Table 4.3: GAPS-perturbed sequence converted to original data

For this example problem, the optimal schedule turns out to be different from those all the heuristics have found. The optimal value of total flowtime equals 3293, and the schedule is given in Figure 4.1. Among the heuristics we have applied, the GAPS algorithm resulted in a schedule with total flowtime only 0.15% more than the optimum. The third best result was found by the two bin algorithm. This result is not surprising, because these two are search algorithms, and they are expected to find better schedules than the dispatching heuristics do.

In this chapter we have explained the algorithms we proposed for the solution of the scheduling with tool changes problem. In the next chapter, the computational results will be discussed.

Chapter 5

Experimental Design

In this chapter, the performance of the proposed algorithms is tested and compared with each other. Moreover, the dynamic-programming algorithm is coded and for some instances the optimal solutions are found. All of the algorithms are coded in C language and compiled with Gnu C compiler. The MIP formulations used in the two bin and knapsack heuristics are solved using callable library routines of CPLEX MIP solver. All the problems are solved on a sparc station 10 under SunOS 5.4.

The experimental setting is explained in §5.1. In §5.2, the computational results are presented and discussed. Finally, a brief summary is provided in §5.3.

5.1 Experimental Setting

There are four experimental factors that can affect the performance of the algorithms. In table 5.1 these factors are listed, where UN stands for a uniform distribution. Each factor can take values in two different levels. Thus the experimental design is a 2^4 full factorial design. Moreover, three different problem sizes are considered for each case, n being equal to 20, 50 and 100.

Factors	Definition	Low	High
v	Variance of processing time	$\frac{2\mu}{5}$	$\frac{3\mu}{4}$
p	Mean processing time	10	20
tl	Tool life	UN$[p_{max}, 3p_{max}]$	UN$[(\frac{n}{4} - 2)p_{max}, (\frac{n}{4} + 2)p_{max}]$
tc	Tool change time	UN$[p_{max}, 2p_{max}]$	UN$[10p_{max}, 15p_{max}]$

Table 5.1: Experimental design factors

For each combination of the factor levels, ten replications are taken. So as a total 480 randomly generated problems are solved.

The experimental factors can be briefly explained as follows:

- Variance of processing times affects the distribution of the processing time values, which directly affects the characteristics of the problem data. The range of processing times affects the flexibility of the jobs to use the tool life effectively, which has implications on the objective function value and performance of different algorithms.
- Mean processing time determines the processing time distribution together with the variance. Processing times are the main data that determines the total flowtime of a given sequence, so it is decided to be a factor.
- Tool life affects the number of tools that should be used. As tool life decreases a tool change is required more frequently and this increases the contribution of tool change time to the objective function value.
- Tool change time is related with the effect of number of tool changes on the objective function value. As tool change time increases, the second part of the objective function gains importance as discussed before.

The processing times are chosen from a discrete uniform random variable between (p_{min}, p_{max}). Hence there are two factors related with the processing

Processing time		mean	
		Low	High
variance	Low	UN[8,12]	UN[16,24]
	High	UN[3,17]	UN[5,35]

Table 5.2: Processing time distributions for different factor levels

times, the variance and mean. We designed the other factors so as to be dependent on the p_{max} value. Tool life has to be larger than the maximum processing time for the problem to be feasible. So we choose the tool life value from a discrete uniform distribution between some multiples of p_{max} value. When factor tc is at high level, the range of tool life depends on n, the number of jobs. This enables us to have the tool life value large enough to make the number of tools used as small as three, so that its effect becomes more visible. Tool change value is chosen from discrete uniform distribution ranging between the given values in table 5.1, which also depends on the p_{max} value.

The processing time distribution for different factor levels of v and p are shown in table 5.2.

After some trial runs are taken, the parameters used in the algorithms are determined as follows:

- The γ value used in knapsack heuristic, which determines the fraction of tool life that will be filled up using SPT rule, is decided to be 0.7.
- The weights in the objective function of the knapsack problem that is solved in two bin heuristic are determined as $w1 = 0.2$ and $w2 = 0.8$.
- Two bin heuristic is run for 50 iterations.
- For the GAPS heuristic the population size is 50, mutation probability is 0.1, perturbation range is $(-3.5, 3.5)$ and mutation range is $(-1.75, 1.75)$.
- GAPS heuristic is decided to be run for 1000 iterations.

The performance measures used in evaluating the results of the experiments are the total flowtime values and the run times in cpu seconds. The relative differences of the solutions of different algorithms are calculated in two ways. First one, denoted by d1, is the relative percent deviation of the heuristic from the minimum value. It is calculated as:

$$d1 = \frac{h - min}{min} 100$$

where h is the solution obtained by the given heuristic and min is the minimum objective value found by all the algorithms. The second measure of deviation, d2, is the relative deviation of the result from the minimum value, scaled by the range of the objective values found by all the algorithms. It is calculated as:

$$d2 = \frac{h - min}{max - min}$$

where max represents the maximum value of the objective function obtained by all the algorithms. In the next section, the results will be presented and discussed.

5.2 Experimental Results

The results will be analyzed first using the overall averages over 160 runs for each n value. Then, we will investigate the effects of the experimental factors. Performance of all the seven algorithms proposed are presented using the deviation terms d1 and d2 in Table 5.3. In this table ARPD stands for the average relative percent deviation, MRPD is the maximum relative percent deviation, NO is the number of problems that the given heuristic gives the best result, and ACT is the average computation time in seconds of cpu time. The summary results are presented for different problem sizes separately. The complete list of computational results can be seen in the tables in Appendix.

First observation we can make from these results is that the average percent deviation, d1, is higher when problem size is small. This can be related with the increase in magnitude of the objective function values with increasing problem

n=20						
	ARPD		MRPD		NO	ACT
	d1	d2	d1	d2		
SPT	3.59	0.43	29.67	1	56	0.001438
FFD	5.00	0.56	44.69	1	36	0.001063
MFFD	2.34	0.35	12.9	1	50	0.001509
EGI	3.27	0.45	29.15	1	34	0.043187
Knap	2.40	0.43	14.49	1	26	0.46125
2bin	0.56	0.07	9.44	0.87	82	5.0535
GAPS	0.38	0.04	9.44	0.79	138	0.953938
n=50						
	ARPD		MRPD		NO	ACT
	d1	d2	d1	d2		
SPT	2.23	0.39	19.53	1	47	0.001375
FFD	3.94	0.6	35.00	1	16	0.003
MFFD	1.93	0.43	10.36	1	19	0.003438
EGI	1.52	0.30	11.25	1	47	0.518438
Knap	1.84	0.48	11.69	1	9	0.8345
2bin	0.53	0.10	7.36	1	69	34.11788
GAPS	0.11	0.02	2.83	0.53	138	4.94875
n=100						
	ARPD		MRPD		NO	ACT
	d1	d2	d1	d2		
SPT	2.41	0.35	18.4	1	64	0.003438
FFD	2.40	0.51	19.36	1	31	0.00725
MFFD	1.80	0.39	15.1	1	50	0.006
EGI	1.52	0.29	12.3	1	54	1.449438
Knap	1.04	0.40	7.57	1	25	2.755375
2bin	1.25	0.17	16.3	1	77	51.92663
GAPS	0.21	0.02	7.9	0.64	130	9.795438

Table 5.3: Summary results for the overall computations

sizes. Since the improvement obtained does not increase by the same amount the flowtime increases, the percent deviations are less when there are more jobs.

Worst result on the average is obtained by FFD algorithm, after that comes SPT. This was expected, since all the other heuristics are developed in order to improve over these two simple rules' performance. FFD heuristic may deviate up to 44.69 percent from the best result. This problem instance was the one with all factors at high levels, that is when tool life, tool change time and processing time range are high. Thus we see that the problem cannot be seen as only a bin packing problem even when the tool change time is large. The d2 value of FFD shows that it is on the average worst among all the algorithms, although there are instances that it gives the best result. MFFD heuristic performs better than the first two. But as n increases its deviation also increases. The reason for this is that MFFD algorithm's main advantage comes from assigning more jobs to the first tool. When number of jobs increases, this advantage diminishes, since in a larger portion of the jobs FFD is applied again. Hence, as n increases, the performance of MFFD algorithm approaches to performance of FFD. But still MFFD is better than both SPT and FFD on the average.

The dispatching heuristics MFFD and EGI provides approximately 2% improvement over the worst result when number of jobs is 20 and 50. When n is 100, the range of improvement is small, the highest deviation being 2.41. In this case the improvement achieved by MFFD and EGI heuristics are normally smaller. For 20 jobs, MFFD gives better results than EGI, in terms of both measures of deviation, while for larger problem sizes EGI is better than MFFD. The computation times for these algorithms are very small. Among these four dispatching heuristics EGI requires the longest average computation time.

The fifth heuristic proposed was a construction type heuristic which is denoted as knap in the tables. Although the computation time spent is larger than the earlier algorithms, it is as small as 2.75 seconds even when n is 100. This algorithm performs better than SPT and FFD heuristics.

Although it requires more computation time, its d1 value is higher than EGI and MFFD, except when n is 100. For 100 jobs, the d2 value shows that it may perform worse than the others in many instances, although the average percent deviation is smaller in this case. The d2 value being large, shows that with small deviations, knapsack heuristic may give the largest flowtime value in some cases. The myopic nature of this algorithm may be the reason for this performance result. Knapsack heuristic assigns jobs to tools with the objective of maximizing the usage of that tool. It does not consider the later steps to be taken. Furthermore, the knowledge of Tc value is not used in this algorithm, so the average performance is not as good as expected.

Two bin and GAPS heuristics give better results than all the others, GAPS being the best. We can also see from the number of best (NO) column that GAPS dominates all the other algorithms, and its maximum d2 values show that GAPS never performed worst in these 480 problem instances. Moreover, the average computation time required for GAPS is 9.7 seconds for 100 jobs, which is a reasonable amount. On the other hand, two bin heuristic requires much more computation time. Actually, this value is inflated just because of a few instances where a difficult knapsack problem is encountered. In these cases, it took CPLEX to solve the problem more than half an hour, hence the average values increased. For most of the problems, two bin heuristic requires much less computation time than what is indicated in table 5.3.

The two bin algorithm gives very close results to the best one. However, when number of jobs is 100, its deviation d1 increases to more than one percent, contrary to the observation that the deviations decrease with increasing problem sizes. The reason for this may be the iteration number used. The instances when two bin deviates much are the ones when FFD algorithm performs well, generally when tool life is at the low level. We conclude that two bin may require more iterations for large problems, because at each iteration two blocks are considered. When there are many jobs and tool life is small, there would be many blocks to be processed, and 50 iterations may not be enough to achieve better solutions. The d2 values show that two bin may give the worst result in some cases. This may be due to the possibility of getting stuck at a local optima. If the tools are fully utilized at some iteration during the search, it may not change the partition after that point.

For the problem set when there are 20 jobs, we have also tried to find the optimal values of the objective function by using the dynamic programming formulation given in chapter 3. Although we have a pseudo-polynomial algorithm, we could only solve 17 problems among the 160. The complexity of the dynamic programming formulation depends on m in the exponent, so only for problems where tool life is very large and m is less than 3, the optimal solutions could be found. Among these 17 problems; for 13 of them, all the algorithms have found the optimal solution, for 3 problems, the result obtained by GAPS heuristic was the optimum and for one, the optimal solution was different from all the ones given by the proposed heuristics. But in this case, the deviation of GAPS was only 0.15 percent. Since the number of problems that could be solved optimally is only ten percent of all problems for one problem size, we did not include them in the summary calculations.

The results are summarized for the sixteen factors and three problem sizes in the tables in Appendix. It is seen that the two local search algorithms perform much better than all the others in all cases. In the remaining parts, in order to understand the effect of factor levels on the performance of the algorithms, averages taken over the same factor levels will be presented and discussed.

In Table 5.4, the effect of tool life and tool change time on the d1 value is investigated over all the problems we have solved. These values are averaged over 120 runs.

We see that when tool change time is at high level, the deviation of SPT from the best increases significantly. When at the same time the tool life is at its low level, SPT performs very bad compared to all the other algorithms. For high tc and tl level, on the average 6.84 percent improvement can be obtained over SPT using the GAPS heuristic. If we investigate the results for each factor combination given in Appendix in more detail, we see that the case with high variance and mean of processing times, high tc level and low tl level lead to the

tl-tc	SPT	FFD	MFFD	EGI	Knap	2Bin	GAPS
00	2.74	3.56	1.95	2.49	1.86	0.99	0.24
01	6.84	3.87	2.24	3.98	3.16	1.81	0.56
10	0.19	1.10	0.72	0.19	0.69	0.08	0.01
11	1.21	6.59	3.19	1.77	1.33	0.24	0.13

Table 5.4: Average percent deviations (d1) for changing tl-tc factors

worst performance results for SPT heuristic. We observe that FFD performs better than SPT when tc is at high level and tl is at low level. This justifies our intuition, when tl is low, the number of tool changes required increases and making economics of tool life gains more importance. But when tl is high, even if tc is also high, SPT performs better than FFD since this time not much tool is required even with the SPT algorithm.

Increase in the tool change time level affects all the algorithms' performance, although not as dramatically as it does for SPT. When tool life is low, we see that the percent deviation of FFD increases with a small amount by increasing the tc level. This shows that FFD algorithm finds relatively good sequence to minimize the C_2^σ portion of the objective function in this case. However when tool life is high, the deviation of FFD solution increases significantly. Thus we see that when total number of tools needed is not much, SPT rule outperforms FFD even for the second part of the objective function.

For all levels of tc and tl factors, the proposed algorithms provide some improvement over SPT heuristic. Among them, the best results are obtained by the two bin and GAPS algorithms as discussed earlier.

We analyzed the impact of v and p levels in table 5.5. It is observed that when variance of processing times is high, the deviations increase for all heuristics. The mean processing time seems to have a minor effect on the performances of the algorithms. The differences in deviation when p is high and low are generally very small, being approximately 0.50 percent.

v-p	SPT	FFD	MFFD	EGI	Knap	2Bin	GAPS
00	2.76	1.68	1.28	1.89	1.14	0.79	0.04
01	2.38	1.80	1.53	1.68	1.72	0.72	0.10
10	2.69	5.80	2.70	2.16	1.85	0.71	0.18
11	3.15	5.84	2.58	2.70	2.33	0.90	0.63

Table 5.5: Average percent deviations (d1) for changing v-p factors

On the other hand, when the variance of processing time distribution is increased there is much implication on the percent deviations of the heuristics. Especially for FFD algorithm, increasing the variance significantly increases the deviation from the best. In this case performance of it becomes worse than the SPT rules' performance, while it was better when the v level is low. The reason for this may be that FFD algorithm tries to find a balanced load for all tools, so when the variance is high, many large jobs will be assigned to the early positions which would increase the C_1^σ portion of the flowtime. In general, we can say that when variance of processing times is high, there is more room for improvement, since the processing times will lie in a larger range. When variance is low, because of the pattern of processing times, fewer alternative sequences will be found and this makes the average deviations of the heuristics decrease.

For all the values of n, the factor combination which results in largest deviation in most of the algorithms is (1101), as seen in the Appendix. In this case, even GAPS heuristic deviates more than one percent from the best. This suggests that in this factor combination, the problem becomes more difficult to obtain a good solution. In fact the reason is obvious. When tc is high and tl is low, the number of tool changes and their impact on the total flowtime value are important. At the same time, if the variance and mean of processing times are at high level, then the processing times are chosen randomly from a large range, allowing many alternative solutions. Hence this case turns out to be the most difficult one for our problem. The deviation of SPT algorithm is huge for this factor combination. The d2 value is 0.95 for n equals 100 which implies that SPT gives the worst solution most of the time for this case. GAPS and two bin heuristics on the other hand, still perform best among all the algorithms.

5.3 Summary

In this chapter, an experimental design is presented for the proposed algorithms. We have first explained the experimental factors and parameters. Then we discussed the computational results and the effects of the factors on the performance of the algorithms. Our findings can be summarized as follows:

- Among all the proposed heuristics, the worst ones in terms of the total flowtime is FFD and SPT algorithms.
- The computation time requirements are very low for all the algorithms, except for two bin heuristic in some exceptional cases. On the average two bin algorithm requires the longest run time but its performance in terms of total flowtime is very close to the best one on the average.
- MFFD algorithm performs better when the problem size is small.
- Among all proposed algorithms, GAPS gives the best result in terms of the objective value.
- Tool change time and tool life are important factors for performance of the algorithms. When tool change time is high and tool life is low, the proposed algorithms achieve the largest improvement over the performance of the SPT heuristic.
- Variance of processing time has an impact on the performance of the algorithms. In general, increase in the variance increases the percentage improvements that can be achieved.

- The problem is most difficult when tool change time and processing time range are high and tool life is low. Most of the algorithms show their worst performance in this case.

In the next chapter we will discuss some future research directions.

Chapter 6

Future Research Directions

In this thesis, the scheduling problem with tool changes with the objective of minimizing the total flowtime is studied. One possible extension of this study would be to add the total manufacturing cost to the objective function, considering the processing times as a consequence of the decision of machining conditions, rather than being constant. Thus, the integration of the tool management and scheduling problems would be improved. In this chapter this direction of possible future work will be discussed and some findings on this issue will be presented.

6.1 Problem Definition

In this chapter, we make a new problem definition. Most of the definitions and assumptions are the same with the ones used in this thesis study, but there are some further additions for this extended problem. For this reason, let us present the assumptions and the notation that will be used hereafter.

As a possible extension, the problem of determining the optimum machining conditions, and the schedule of a set of jobs can be considered. The objective is to minimize total cost of flow time, tooling, and depreciation. The current problem scope is defined with the following assumptions:

- There is a single machine, and a single tool type.
- There are N jobs with no precedence relation, all with ready times equal to zero.
- The depth of cut for a job is given, where cutting speed and feed rate are decision variables.
- There are ample tools at hand.
- When the tool is worn, it has to be replaced with a new one by spending time Tc.
- Tool change is not allowed during a manufacturing operation in order to achieve the desired surface finish quality.

The processing times of jobs, and tool life is determined by the machining parameters, v and f, by some well known formulas as discussed in Akturk and Avci [2].

The notation used is as follows:

α, β, γ : Speed, feed, depth of cut exponents for the cutting tool
C : Taylor's tool life constant for the cutting tool
C_m, b, c, e : Specific coefficient and exponents of the machine power constraint
C_D : Depreciation cost of the CNC machine, ($/min)
C_o : Flowtime cost ($/min)
C_s, g, h, l : Specific coefficient and exponents of the surface roughness constraint
C_t : Cost of the tool, ($/per tool)
D_i : Diameter of the generated surface for job i, (in.)
d_i : Depth of cut for job i, (in.)
f_i : Feed rate for job i, (ipr)
H : Maximum available machine power for all operations, (hp)

L_i : Length of the generated surface for the job i, (in.)
S_i : Maximum allowable surface roughness for operation i, (μin.)
t_{m_i} : Machining time of operation i, (min.)
Tc : Tool replacing time for the tool, (min.)
T_i : Tool life of the tool in job i, (min.)
u_i : Usage rate of the tool in job i
v_j : Cutting speed for job i, (fpm)

The usage rate of tool in operation i, denoted as u_i, is defined as the machining time to tool life ratio, and is expressed as follows:

$$u_i = \frac{t_{m_i}}{T_i} = \frac{(\pi.D_i.L_i)/(12.v_i.f_i)}{C/(v_i^{\alpha}.f_i^{\beta}.d_i^{\gamma})} = \frac{\pi \,.\, D_i \,.\, L_i \,.\, d_i^{\gamma}}{12 \,.\, C \,.\, v_i^{(1-\alpha)} \,.\, f_i^{(1-\beta)}}$$

6.2 Discussion

If a schedule is viewed as a sequence of blocks of jobs, which are separated by tool changes, the problem is deciding on the optimum machining conditions for each job, and partitioning the jobs into blocks. Then we can order the jobs within blocks in SPT, and the blocks can be ordered optimally using an SPT based rule for groups of jobs. However, it is obvious that these decisions are interacting, since the machining conditions determine the machining time, and the tool life. Partitioning the jobs into minimum number of blocks does not imply optimality. As the tool replacing time decreases, t_{m_i} dominates Tc for the scheduling decision, and number of tool changes done may become less significant. Therefore, a joint cost function for each job can be written as:

$$C_t.u_i + C_D.t_{m_i} + C_o.(N - n + 1).(t_{m_i} + (Tc))$$

where the cost components are; tooling cost, depreciation cost, and the flowtime cost, respectively. The value Tc is written in parenthesis on purpose, meaning it will be added only if a tool change is done before that job.

Relations between the variables can be summarized as:

$$u \propto v \propto \frac{1}{t_m}$$

Since increasing usage rate would cause a more frequent tool changes, we can also say:

$$u \propto \text{number of tool replacements}$$

Thus, we see that there is a trade off between the costs associated with machining time (depreciation, and flowtime costs) and the costs associated with non-machining time (tooling cost and flowtime cost), which should be taken into account in the solution procedure.

The solution to this problem can be investigated in two different cases as will be discussed in the following subsections.

6.2.1 Case I : $Tc = 0$

Automatic machine tools can replace the worn tool during set up of a workpiece; which means that $Tc = 0$. In this case, the problem can be formulated as an assignment problem.

In an optimal schedule, if job i is assigned to position j, its contribution to the cost function is,

$$C_t.u_i + C_D.t_{m_i} + C_o.(N - j + 1).t_{m_i}$$

where the cost components are; tooling cost, depreciation cost, and the flowtime cost respectively. The optimum machining parameters v, f, which minimizes this cost can be found by solving the single machining operation problem (SMOP), using the formulation of Akturk and Avci [2].

Hence the cost of assigning a job i to position j is the solution of the following program:

$$
\begin{aligned}
\text{Minimize} \quad & M_{ij} = C_1.v_i^{-1}.f_i^{-1} + C_2.v_i^{(\alpha-1)}.f_i^{(\beta-1)} & \\
\text{Subject to:} \quad & C'_m.v_i^b.f_i^c \leq 1 & \text{(Machine Power Constraint)} \\
& C'_s.v_i^g.f_i^h \leq 1 & \text{(Surface Roughness Constraint)} \\
& v_i, f_i > 0 &
\end{aligned}
$$

where,

$$C_1 = \frac{\pi.D_i.L_i.((N-j+1).C_o + C_D)}{12}, \quad C_2 = \frac{\pi.D_i.L_i.d_i^{\gamma}.C_t}{12.C}$$

$$C'_m = \frac{C_m.d_i^e}{H}, \text{ and } \quad C'_s = \frac{C_s.d_i^l}{S_i}$$

It follows that, the cost incurred by scheduling job i to position j, and then selecting the machining parameters optimally, does not depend on the other jobs, and influence the further decisions. Thus, we can formulate our problem as an N x N assignment problem.

The formulation would be as follows:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}.x_{ij} \\
\text{Subject to:} \quad & \sum_{j=1}^{N} x_{ij} = 1, i = 1, \ldots, N \\
& \sum_{i=1}^{N} x_{ij} = 1, j = 1, \ldots, N
\end{aligned}
$$

where,

$$x_{ij} = \begin{cases} 1, & \text{if job } i \text{ is scheduled at position } j \\ 0, & \text{otherwise} \end{cases}$$

and

M_{ij} = optimal result of SMOP formulated above

6.2.2 Case II : $Tc > 0$

In this case, the tool replacing time is a positive constant, and this makes the problem much more complicated. Since $Tc > 0$, the tool replacing time has a contribution to the cost function, and the existence of this additional time for a job depends not only on the usage rate of that job, but also on the usage rates of the previous jobs. The main difficulty arises for this reason. Consequently, the optimal machining conditions cannot be calculated for a position independent of the other jobs.

However, by making an approximation, we can also reduce this problem to an assignment problem. In this case, we write the contribution of a job to the total cost if it is assigned to position j as follows:

$$C_t.u_i + C_D.t_{m_i} + C_o.(N-j+1).t_{m_i} + C_o.(N-j+1).Tc.u_i$$

where the last term represents the flowtime cost that might be added as a result of tool replacement. The other cost components are the same as Case I.

In fact, we do not know whether there will be a tool change or not in advance. If there is one, its contribution would be $C_o.(N-j+1).Tc$. On the other hand, we know that, when the usage rate (u_i) increases, it is more likely that a tool change would be needed. Hence, multiplying this cost component by u_i, we approximately consider the tool replacing time cost in the total cost of assigning a job i to the jth position in the sequence. This cost can be calculated independent of the other jobs in the sequence, giving us the opportunity to formulate the problem as an assignment problem.

6.2.3 Further Issues for an Exact Algorithm

The total cost for case II can be written as follows:

$$TC = \sum_{i=1}^{N} C_t.u_i + C_D.t_{m_i} + \sum_{j=1}^{N}(N-j+1).Tc.y_{[j]} + \sum_{j=1}^{N}\sum_{i=1}^{N} C_o.(N-j+1).t_{m_i}.x_{ij}$$

where,

$$x_{ij} = \begin{cases} 1, & \text{if job } i \text{ is scheduled at position } j \\ 0, & \text{otherwise} \end{cases}$$

and,

$$y_{[j]} = \begin{cases} 1, & \text{if a tool replacement is required before the job at position } j \\ 0, & \text{otherwise} \end{cases}$$

Instead of trying to solve for the above cost function directly, we can consider an iterative solution method. A state represents a portion of the schedule completed up to that point. At each state, the jobs waiting to be scheduled, and the remaining tool life is known. So, the cost contribution (with optimal machining conditions) of each job for a given state can be easily calculated by the formulation given in section 2.1. And if the resulting usage rate is less then the remaining tool life, it is fine.

When the remaining life of the tool is less than the usage rate, the result may not be optimal since it ignores the tool change time, then there are two possibilities to be considered:

1. Either replacing the tool with a new one, thus spending time Tc,
2. Or, changing the machining conditions to fit the usage rate to the remaining tool life, thus increasing the machining time, t_{m_i}.

We see that there is a trade off between Tc and t_{m_i}, which affects the costs for alternative actions. We can summarize the relations between u_i and cost components as:

$$u_i \propto \text{tooling cost} \propto \text{tool replacing time cost} \propto \frac{1}{\text{machining time cost}}$$

After computing the cost contribution of all remaining jobs, we should consider which new states to generate. We are considering to work on finding some dominance relations between jobs for a position j, in order to decrease the number of alternatives at each state. Taking into account the above relations can be helpful for this purpose.

If we relax the assumption of not allowing tool change during a manufacturing operation, we can formulate the problem of minimizing the total flowtime as a mixed integer programming formulation, for given values of t_{m_i}, T_i, Tc, and TL (tool life) as follows:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j=1}^{N}(N-j+1).Tc.y_{[j]} + \sum_{j=1}^{N}\sum_{i=1}^{N}(N-j+1)t_{m_i}.x_{ij} \\
\text{Subject to:} \quad & \sum_{j=1}^{N} x_{ij} = 1, i = 1,\dots,N \\
& \sum_{i=1}^{N} x_{ij} = 1, j = 1,\dots,N \\
& \sum_{i=1}^{N} T_i.x_{ij} + S_{j-1} = S_j, \forall j \\
& \frac{S_j}{TL} \geq K_j, \forall j \\
& \frac{S_j}{TL} < K_j + 1, \forall j \\
& K_j - K_{j-1} \leq M.y_j, \forall j \\
& S_0 = 0, \\
& K_0 = 0, \\
& K_j \text{ is integer and } x_{ij},\ y_{[j]} \text{ are binary variables}
\end{aligned}
$$

where x_{ij} and $y_{[j]}$ are defined above before.

This formulation may be helpful to solve a subproblem within an algorithm that can be developed for this problem.

6.3 Summary

In this chapter we introduce a possible extension to our problem. There are many possible extensions to the scheduling with tool changes problem. The one considered here is considering the processing times as a consequence of the machining conditions decisions and including the tooling and depreciation cost to the objective value in addition to the flowtime cost.

We have discussed some difficulties to be faced with this problem. Then we have formulated the special case of the problem, when tool change time is insignificant, as an assignment problem. For the general case, we have discussed some approximations and possible directions for an exact algorithm.

In the next chapter some concluding remarks will be done.

Chapter 7

Conclusion

In this chapter a brief summary of the contributions of this study will be done and some possible extensions of this study will be presented for future research. In this thesis, we have considered the scheduling problem to minimize the total flowtime for a single CNC machine where tool changes are required due to tool wear. We analyzed the characteristics of the problem and proposed a pseudo-polynomial dynamic programming formulation and several heuristic algorithms. In §7.1 the results will be summarized and finally in §7.2 some future research directions will be suggested.

7.1 Results

The existing studies in the literature consider tool changes which are done due to different requirements of parts and they do not incorporate scheduling objectives in decision of the job sequence, rather try to minimize the number of tool changes. However, the tool replacements due to tool wear can have significant impact on total cost of production and throughput of parts. Moreover, the tools are changed ten times more often due to tool wear than due to part mix because of relatively short tool lives of many turning tools as stated by Gray et al. [13]. There are no studies in the literature which take these facts into account in the solution of scheduling problems. In scheduling literature, there are some studies concerning the unavailability of machines, but to the best of my knowledge, no published work exists which considers the unavailability of tools. This thesis is a first step to fill this gap in the literature.

In this study we have introduced an untouched problem, scheduling with tool changes with the objective of minimizing the total flowtime. We have shown that this problem is strongly NP-hard. Moreover, even when number of tools is fixed, it was shown to be NP-hard. Then we have discussed the performance of the SPT heuristic for our problem and shown that its worst case performance is bounded above by a constant. Some conditions concerning the problem data were also presented which would guarantee the optimality of SPT rule for scheduling with tool changes problem to minimize total flowtime.

We have developed a pseudo-polynomial dynamic programming formulation for the problem, but for computational purposes this algorithm was not practical. Since the problem is NP-hard, heuristic approaches are justified. In this study, several heuristic algorithms were proposed for the solution of this problem in order to improve the simplest heuristic, SPT, which first comes to mind for flowtime problems, and FFD, which is a bin packing heuristic and was expected to work well for tool change problem when tool change time is very large.

We have proposed two dispatching heuristics, one of which is a modification to FFD, and the other uses a simple dynamic ranking index. In the proposed computational studies, these algorithms perform better than SPT and FFD even though they require very short computation time. Another algorithm presented was a construction algorithm based on a knapsack problem. It provided some improvement on the average with a small additional computational effort. However, this algorithm showed erratic behavior during the computational analysis and could not always outperform the dispatching heuristics.

The best improvement was achieved by the two local search algorithms. We have used the problem space search technique with genetic algorithm which proved to be the most effective of all to find a relatively good solution to the scheduling with tool changes problem with respect to both the objective function value and computation time.

As a result, we have shown that for this problem, tool change time and tool life values are important data which together determine the characteristics of the problem. Thus, we can conclude that the solution procedure for this problem should take into account these values. We have observed that when tool change time is small and tool life is large, SPT heuristic can perform quite well and we cannot improve much over its performance. However, when the tool change time is large and tool life is low the problem gets complicated and SPT cannot perform as well. In this case, we can provide on the average 6.84 percent improvement over SPT rule's solution. Moreover, over the small set of problems we could solve optimally by using the dynamic programming algorithm, we have seen that the genetic algorithm with problem space search finds very close results to the optimum.

7.2 Future Research Directions

Finally, some future research directions can be suggested as follows:

- In this study, only a single tool type was considered. The study can be extended considering multiple tool types and different tooling requirements of jobs.
- We have assumed the processing times to be constant. However, the processing times and tool life are direct consequences of the machining parameters. The decision of machining parameters can be incorporated into this study, for which we have started the discussion, in Chapter 6.
- The scheduling with tool changes problem can be considered with some other performance measures, such as weighted flowtime, tardiness etc.

Bibliography

[1] Adiri I., Bruno, J., Frostig, E. and Rinnoy Kan, A.H.G., "Single machine flow-time scheduling with a single breakdown", *Acta Informatica*, **26**, 679-696, 1989.

[2] Akturk, M.S. and Avci S., "Tool allocation and machining conditions optimization for CNC machines", *European Journal of Operational Research*, **94** (2), 335-348, 1996.

[3] Anily S., Bramel J. and Simchi-Levi, D., "Worst-case analysis of heuristics for the bin-packing problem with general cost structures", *Operations Research*, **42** (2), 287-298, 1994.

[4] Asai, K. and Takashima, S., *Manufacturing, Automation Systems and CIM Factories*, Chapman & Hall, London, 1994.

[5] Baker, B.S., and Coffman E.G., "A tight asymptotic bound for next-fit-decreasing bin-packing", *SIAM J. Alg. Discr. Meth.*, **2** (2), 147-152, 1981.

[6] Baker, K.R., *Introduction to Sequencing and Scheduling*, Wiley, New York, 1974.

[7] Baker, K.R., *Elements of Sequencing and Scheduling*, Darmouth College, Hannover, NH, 1994.

[8] Crama, Y., Kolen, A.W.J, Oerlemans, A.G. and Spieksma, F.C.R., "Minimizing the number of tool switches on a flexible machine", *International Journal of Flexible Manufacturing Systems*, **6** (1), 33-54, 1994.

[9] Crama, Y., "Combinatorial optimization models for production scheduling in automated manufacturing systems", *European Journal of Operational Research*, **99** (1), 136-153, 1997.

[10] Ermer, D.S., "Optimization of the constrained machining economics problem by geometric programming", *Journal of Engineering Industry*, **93** (4), 1067-1072, 1971.

[11] Garey, M.R. and Johnson, D.S., *Computers and Intractability:A Guide to the Theory of NP-Completeness*, W.H.Freeman and Company, New York, 1979.

[12] Gopalakrishnan, B., and Al-Khayyal, "Machine parameter selection for turning with constraints: An analytical approach based on geometric programming", *International Journal of Production Research*, **29** (9), 1897-1908, 1991.

[13] Gray, E., Seidmann, A. and Stecke, K.E., "A synthesis of decision models for tool management in automated manufacturing", *Management Science*, **27** (10), 549-567, 1993.

[14] Hitomi, K., "Analsis of optimal machining speeds for automatic manufacturing", *International Journal of Production Research*, **27** (10), 1685-1691, 1989.

[15] Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R. and Graham, R.L, "Worst-case performance bounds for simple one-dimensional packing algorithms", *SIAM J. Comput.*, **3** (4), 299-325, 1974.

[16] Kouvelis, P., "An optimal tool selection procedure for the initial design phase of a flexible manufacturing system", *European Journal of Operational Research*, **55** (2), 201-210, 1991.

[17] Lee, C.Y., and Liman, S.D., "Single machine flow-time scheduling with scheduled maintenance", *Acta Informatica*, **29**, 375-382, 1992.

[18] Lee, C.Y., and Liman, S.D., "Capacitated two-parallel machines scheduling to minimize sum of job completion times", *Discrete Applied Mathematics*, **41**, 211-222, 1993.

[19] Lee, C.Y., "Minimizing the makespan in two-machine flowshop scheduling problem with an availability constraint", *Operations Research Letters*, **20** (3), 129-139, 1997.

[20] Lee, C.Y., "Machine scheduling with an availability constraint",*Journal of Global Optimization*, **9** (3/4), 395-416, 1996.

[21] Lee, C.Y., Lei, L., and Pinedo, M., "Current trends in deterministic scheduling", *Annals of Operations Research*, **70**, 1-41, 1997.

[22] Morton, T.E., and Pentico, D.W., *Heuristic Scheduling Systems With Applications to Production Systems and Project Management*, John Wiley&Sons, Inc., New York, 1993.

[23] Naphade, K.S., Wu, D.S, Storer R.H., "Problem space search algorithms for resource-constrained project scheduling", *Annals of Operations Research*, **70**, 327-357, 1997.

[24] Pinedo, M., *Scheduling:Theory,Algorithms and Systems*, Prentice Hall, New Jersey, 1995.

[25] Stecke, K.E., "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems", *Management Science*, **29**, 273-288, 1983.

[26] Storer, R.H., Wu, D.S., Vaccari, R., "New search spaces for sequencing problems with application to job shop scheduling", *Management Science*, **38**, 1495-1509, 1992.

[27] Tang, C.S. and Denardo, E.V., "Models arising from a flexible manufacturing machine, Part I:Minimization of the number of tool switches", *Operations Research*, **36**, 767-777, 1988.

[28] Tang, C.S. and Denardo, E.V., "Models arising from a flexible manufacturing machine, Part II:Minimization of the number of switching instants",*Operations Research*, **36**, 778-784, 1988.

[29] Taylor, F.W., "On the art of cutting metals", *Transaction ASME*, **28**, 31-350, 1906.

[30] Tetzlaff, U.A.W., "A queuing network model for flexible manufacturing systems with tool management", *IIE Transactions*, **28** (4), 309-317, 1996.

[31] Veeramani, D., Upton D.M. and Barash, M.M., "Cutting tool management in computer integrated manufacturing", *International Journal of Flexible Manufacturing Sytems*, **4** (3), 237-265, 1992.

Appendix A

Computational Results for 20 Jobs

v-p-tl-tc	SPT Flowtime	SPT d1	SPT d2	FFD Flowtime	FFD d1	FFD d2	MFFD Flowtime	MFFD d1	MFFD d2	EGI Flowtime	EGI d1	EGI d2
0000	3974.40	3.12	0.53	3874.20	0.69	0.34	3860.00	0.27	0.06	3939.40	2.42	0.57
0001	19550.10	6.98	0.60	18463.60	0.43	0.21	18538.40	0.97	0.09	18823.40	2.74	0.37
0010	2422.70	0.65	0.26	2435.60	1.25	0.51	2434.60	1.16	0.43	2422.70	0.65	0.26
0011	6253.30	2.92	0.49	6360.90	4.97	0.70	6236.60	2.77	0.45	6292.60	3.75	0.70
0100	7924.80	2.16	0.51	7865.20	1.33	0.52	7823.50	0.71	0.19	7930.70	2.18	0.67
0101	38670.50	5.37	0.60	37552.00	1.40	0.40	37586.60	1.48	0.21	37712.70	1.98	0.52
0110	4885.40	0.74	0.25	4900.80	1.11	0.51	4915.10	1.35	0.52	4885.40	0.74	0.25
0111	12651.70	3.49	0.52	12773.90	4.71	0.74	12762.50	4.48	0.66	12698.90	4.10	0.77
1000	2788.40	3.62	0.46	2937.00	9.00	0.97	2770.90	3.10	0.45	2774.10	3.08	0.40
1001	12438.50	8.23	0.65	12884.50	12.09	0.90	11921.80	4.06	0.35	12296.70	6.75	0.51
1010	1812.70	0.32	0.17	1830.70	1.37	0.38	1836.30	1.49	0.51	1812.70	0.32	0.17
1011	4089.10	2.25	0.22	4539.30	11.64	0.52	4167.90	3.82	0.38	4188.20	4.07	0.35
1100	5556.20	4.19	0.62	5722.40	7.44	0.78	5457.90	2.75	0.36	5488.80	3.01	0.48
1101	24887.20	10.32	0.79	24520.00	9.07	0.63	23214.70	3.88	0.33	24901.50	10.16	0.69
1110	3591.10	0.41	0.05	3694.40	3.28	0.42	3620.30	1.21	0.21	3591.10	0.41	0.05
1111	8140.70	2.73	0.23	8815.60	10.14	0.48	8240.50	3.90	0.31	8422.80	5.95	0.43
AVG	9977.30	3.59	0.43	9948.13	5.00	0.56	9711.73	2.34	0.35	9886.36	3.27	0.45

Table A.1: Averages over ten replications for n=20

v-p-tl-tc	Knap Flowtime	Knap d1	Knap d2	2Bin Flowtime	2Bin d1	2Bin d2	GAPS Flowtime	GAPS d1	GAPS d2
0000	3886.40	1.01	0.29	3857.90	0.19	0.03	3851.30	0.00	0.00
0001	18608.40	1.23	0.15	18488.80	0.47	0.02	18414.30	0.00	0.00
0010	2436.60	1.25	0.71	2412.80	0.27	0.11	2406.10	0.00	0.00
0011	6162.30	1.79	0.32	6077.70	0.41	0.06	6055.40	0.00	0.00
0100	7855.40	1.21	0.40	7781.20	0.08	0.03	7777.90	0.03	0.00
0101	37526.50	1.20	0.17	37194.40	0.02	0.00	37237.60	0.23	0.04
0110	4961.20	2.28	0.87	4865.30	0.35	0.11	4847.50	0.00	0.00
0111	12744.40	4.06	0.49	12270.90	0.67	0.08	12241.40	0.34	0.08
1000	2740.60	2.18	0.33	2725.20	1.24	0.16	2701.50	0.51	0.04
1001	11941.10	4.58	0.34	11734.00	1.61	0.15	11488.70	0.00	0.00
1010	1820.40	0.72	0.41	1808.60	0.06	0.15	1807.60	0.00	0.00
1011	4115.50	2.89	0.28	4019.80	0.38	0.05	4001.10	0.00	0.00
1100	5472.40	2.75	0.48	5387.00	1.22	0.12	5361.70	0.92	0.13
1101	23928.80	6.15	0.46	22955.60	1.87	0.09	23156.20	3.30	0.23
1110	3634.10	1.53	0.65	3580.90	0.11	0.01	3581.00	0.10	0.03
1111	8227.80	3.57	0.54	7913.50	0.04	0.00	7962.60	0.64	0.09
AVG	9753.87	2.40	0.43	9567.10	0.56	0.07	9555.74	0.38	0.04

Table A.2: Averages over ten replications for n=20 (continued)

v-p-tl-tc	SPT Flowtime	SPT d1	SPT d2	FFD Flowtime	FFD d1	FFD d2	MFFD Flowtime	MFFD d1	MFFD d2	EGI Flowtime	EGI d1	EGI d2
0000	3187.00	3.07	1.00	3095.00	0.10	0.03	3095.00	0.10	0.03	3187.00	3.07	1.00
0000	4208.00	14.04	1.00	3690.00	0.00	0.00	3737.00	1.27	0.09	4208.00	14.04	1.00
0000	3776.00	0.00	0.00	3776.00	0.00	0.00	3776.00	0.00	0.00	3776.00	0.00	0.00
0000	4191.00	4.38	1.00	4015.00	0.00	0.00	4015.00	0.00	0.00	4090.00	1.87	0.43
0000	5658.00	6.77	1.00	5304.00	0.09	0.01	5299.00	0.00	0.00	5299.00	0.00	0.00
0000	4881.00	0.00	0.00	4917.00	0.74	0.68	4881.00	0.00	0.00	4934.00	1.09	1.00
0000	2832.00	1.32	0.39	2891.00	3.43	1.00	2825.00	1.07	0.31	2832.00	1.32	0.39
0000	4718.00	0.00	0.00	4756.00	0.81	0.67	4718.00	0.00	0.00	4775.00	1.21	1.00
0000	3297.00	0.00	0.00	3297.00	0.00	0.00	3297.00	0.00	0.00	3297.00	0.00	0.00
0000	2996.00	1.56	0.90	3001.00	1.73	1.00	2957.00	0.24	0.14	2996.00	1.56	0.90
0001	14374.00	8.36	1.00	13265.00	0.00	0.00	13265.00	0.00	0.00	13340.00	0.57	0.07
0001	20840.00	29.67	1.00	16098.00	0.17	0.01	16805.00	4.57	0.15	19204.00	19.49	0.66
0001	20870.00	0.00	0.00	20870.00	0.00	0.00	20870.00	0.00	0.00	20870.00	0.00	0.00
0001	19041.00	8.71	1.00	17515.00	0.00	0.00	17515.00	0.00	0.00	17726.00	1.20	0.14
0001	29374.00	9.99	1.00	26710.00	0.02	0.00	26705.00	0.00	0.00	26710.00	0.02	0.00
0001	28821.00	0.00	0.00	28857.00	0.12	0.86	28821.00	0.00	0.00	28863.00	0.15	1.00
0001	9962.00	6.55	1.00	9446.00	1.03	0.16	9610.00	2.78	0.42	9618.00	2.87	0.44
0001	27708.00	0.00	0.00	27746.00	0.14	0.64	27708.00	0.00	0.00	27767.00	0.21	1.00
0001	13517.00	0.00	0.00	13517.00	0.00	0.00	13517.00	0.00	0.00	13517.00	0.00	0.00
0001	10994.00	6.52	1.00	10612.00	2.82	0.43	10568.00	2.39	0.37	10619.00	2.89	0.44
0010	2212.00	0.00	0.00	2253.00	1.85	0.71	2270.00	2.62	1.00	2212.00	0.00	0.00
0010	2498.00	0.00	0.00	2533.00	1.40	1.00	2515.00	0.68	0.49	2498.00	0.00	0.00
0010	2360.00	1.03	0.43	2392.00	2.40	1.00	2360.00	1.03	0.43	2360.00	1.03	0.43
0010	2489.00	0.00	0.00	2523.00	1.37	0.43	2489.00	0.00	0.00	2489.00	0.00	0.00
0010	2826.00	2.54	0.67	2769.00	0.47	0.12	2861.00	3.81	1.00	2826.00	2.54	0.67
0010	2529.00	0.32	0.19	2545.00	0.95	0.57	2545.00	0.95	0.57	2529.00	0.32	0.19
0010	2247.00	0.00	0.00	2247.00	0.00	0.00	2247.00	0.00	0.00	2247.00	0.00	0.00
0010	2558.00	1.67	1.00	2528.00	0.48	0.29	2521.00	0.20	0.12	2558.00	1.67	1.00
0010	2277.00	0.98	0.28	2335.00	3.55	1.00	2307.00	2.31	0.65	2277.00	0.98	0.28
0010	2231.00	0.00	0.00	2231.00	0.00	0.00	2231.00	0.00	0.00	2231.00	0.00	0.00
0011	4924.00	0.00	0.00	5466.00	11.01	1.00	4982.00	1.18	0.11	5063.00	2.82	0.26
0011	6590.00	1.90	0.78	6625.00	2.44	1.00	6607.00	2.16	0.89	6485.00	0.28	0.11
0011	6356.00	5.95	0.71	6499.00	8.33	1.00	6356.00	5.95	0.71	6406.00	6.78	0.81
0011	6239.00	0.00	0.00	6739.00	8.01	1.00	6239.00	0.00	0.00	6660.00	6.75	0.84
0011	8370.00	6.79	0.81	8497.00	8.41	1.00	8434.00	7.60	0.90	8472.00	8.09	0.96
0011	7947.00	3.38	0.79	8011.00	4.21	0.99	8011.00	4.21	0.99	8015.00	4.27	1.00
0011	4202.00	0.00	0.00	4202.00	0.00	0.00	4202.00	0.00	0.00	4232.00	0.71	1.00
0011	8124.00	5.25	1.00	7731.00	0.16	0.03	7724.00	0.06	0.01	7731.00	0.16	0.03
0011	5357.00	5.97	0.84	5415.00	7.12	1.00	5387.00	6.57	0.92	5415.00	7.12	1.00
0011	4424.00	0.00	0.00	4424.00	0.00	0.00	4424.00	0.00	0.00	4447.00	0.52	1.00
0100	6403.00	3.07	1.00	6215.00	0.05	0.02	6215.00	0.05	0.02	6403.00	3.07	1.00
0100	7684.00	6.03	1.00	7306.00	0.81	0.14	7314.00	0.92	0.15	7684.00	6.03	1.00
0100	8137.00	0.00	0.00	8137.00	0.00	0.00	8137.00	0.00	0.00	8137.00	0.00	0.00
0100	8374.00	4.47	1.00	8016.00	0.00	0.00	8016.00	0.00	0.00	8165.00	1.86	0.42
0100	10469.00	0.00	0.00	10480.00	0.11	1.00	10475.00	0.06	0.55	10471.00	0.02	0.18
0100	9984.00	0.00	0.00	10058.00	0.74	0.61	9984.00	0.00	0.00	10105.00	1.21	1.00
0100	5706.00	1.89	0.57	5786.00	3.32	1.00	5607.00	0.13	0.04	5706.00	1.89	0.57
0100	9512.00	0.00	0.00	9591.00	0.83	0.54	9512.00	0.00	0.00	9657.00	1.52	1.00
0100	6948.00	4.80	0.84	7007.00	5.69	1.00	7007.00	5.69	1.00	6948.00	4.80	0.84
0100	6031.00	1.34	0.66	6056.00	1.76	0.87	5968.00	0.29	0.14	6031.00	1.34	0.66
0101	28777.00	8.37	1.00	26555.00	0.00	0.00	26555.00	0.00	0.00	26896.00	1.28	0.15
0101	35668.00	14.06	1.00	31330.00	0.19	0.01	32130.00	2.75	0.20	32130.00	2.75	0.20
0101	45928.00	0.00	0.00	45928.00	0.00	0.00	45928.00	0.00	0.00	45928.00	0.00	0.00
0101	38074.00	8.73	1.00	35016.00	0.00	0.00	35016.00	0.00	0.00	35141.00	0.36	0.04
0101	53281.00	0.00	0.00	53292.00	0.02	1.00	53287.00	0.01	0.55	53292.00	0.02	1.00
0101	57674.00	0.00	0.00	57748.00	0.13	0.70	57674.00	0.00	0.00	57780.00	0.18	1.00
0101	19966.00	6.71	1.00	18925.00	1.15	0.17	18717.00	0.04	0.01	19266.00	2.97	0.44
0101	55682.00	0.00	0.00	55761.00	0.14	0.66	55682.00	0.00	0.00	55801.00	0.21	1.00
0101	29628.00	9.45	0.98	29687.00	9.67	1.00	29687.00	9.67	1.00	29687.00	9.67	1.00
0101	22027.00	6.35	1.00	21278.00	2.73	0.43	21190.00	2.31	0.36	21206.00	2.39	0.38

Table A.3: Computational results for SPT, FFD, MFFD, EGI algorithms for n=20

v-p-tl-tc	SPT Flowtime	SPT d1	SPT d2	FFD Flowtime	FFD d1	FFD d2	MFFD Flowtime	MFFD d1	MFFD d2	EGI Flowtime	EGI d1	EGI d2
0110	4479.00	0.00	0.00	4535.00	1.25	0.41	4533.00	1.21	0.40	4479.00	0.00	0.00
0110	4984.00	0.00	0.00	5052.00	1.36	1.00	5002.00	0.36	0.26	4984.00	0.00	0.00
0110	4762.00	0.00	0.00	4815.00	1.11	0.69	4839.00	1.62	1.00	4762.00	0.00	0.00
0110	5062.00	1.81	0.56	5042.00	1.41	0.43	5062.00	1.81	0.56	5062.00	1.81	0.56
0110	5628.00	2.46	0.64	5520.00	0.49	0.13	5686.00	3.51	0.91	5628.00	2.46	0.64
0110	5133.00	0.39	0.19	5173.00	1.17	0.58	5163.00	0.98	0.48	5133.00	0.39	0.19
0110	4491.00	0.00	0.00	4491.00	0.00	0.00	4491.00	0.00	0.00	4491.00	0.00	0.00
0110	5192.00	1.62	0.27	5139.00	0.59	0.10	5192.00	1.62	0.27	5192.00	1.62	0.27
0110	4588.00	0.39	0.13	4706.00	2.98	1.00	4648.00	1.71	0.57	4588.00	0.39	0.13
0110	4535.00	0.73	0.73	4535.00	0.73	0.73	4535.00	0.73	0.73	4535.00	0.73	0.73
0111	10129.00	0.00	0.00	10956.00	8.16	1.00	10794.00	6.57	0.80	10160.00	0.31	0.04
0111	13168.00	1.89	0.72	13236.00	2.41	0.92	13186.00	2.03	0.78	13262.00	2.62	1.00
0111	12718.00	5.97	0.72	12992.00	8.25	1.00	12855.00	7.11	0.86	12818.00	6.80	0.82
0111	13162.00	5.53	0.68	13494.00	8.19	1.00	13162.00	5.53	0.68	13311.00	6.73	0.82
0111	16716.00	6.76	0.81	16969.00	8.38	1.00	16828.00	7.48	0.89	16917.00	8.05	0.96
0111	15926.00	3.38	0.79	16058.00	4.24	0.98	16042.00	4.14	0.96	16068.00	4.30	1.00
0111	8401.00	0.00	0.00	8401.00	0.00	0.00	8401.00	0.00	0.00	8460.00	0.70	1.00
0111	16370.00	5.22	0.36	15588.00	0.19	0.01	16370.00	5.22	0.36	15596.00	0.24	0.02
0111	10748.00	2.85	0.72	10866.00	3.98	1.00	10808.00	3.43	0.86	10866.00	3.98	1.00
0111	9179.00	3.27	0.45	9179.00	3.27	0.45	9179.00	3.27	0.45	9531.00	7.23	1.00
1000	2168.00	0.65	0.15	2233.00	3.67	0.82	2250.00	4.46	1.00	2168.00	0.65	0.15
1000	2924.00	3.21	0.61	2983.00	5.29	1.00	2963.00	4.59	0.87	2924.00	3.21	0.61
1000	2793.00	4.37	0.18	3330.00	24.44	1.00	2780.00	3.89	0.16	2793.00	4.37	0.18
1000	2927.00	5.63	0.73	2984.00	7.69	1.00	2943.00	6.21	0.81	2802.00	1.12	0.15
1000	4141.00	5.96	0.59	4303.00	10.11	1.00	4032.00	3.17	0.31	4152.00	6.24	0.62
1000	2781.00	1.46	0.13	3042.00	10.98	1.00	2741.00	0.00	0.00	2747.00	0.22	0.02
1000	2409.00	1.73	0.45	2459.00	3.84	1.00	2396.00	1.18	0.31	2409.00	1.73	0.45
1000	3244.00	5.12	0.42	3465.00	12.28	1.00	3192.00	3.43	0.28	3249.00	5.28	0.43
1000	2084.00	2.46	0.36	2174.00	6.88	1.00	2036.00	0.10	0.01	2084.00	2.46	0.36
1000	2413.00	5.56	1.00	2397.00	4.86	0.87	2376.00	3.94	0.71	2413.00	5.56	1.00
1001	8408.00	2.11	0.17	9240.00	12.22	1.00	8490.00	3.11	0.25	8630.00	4.81	0.39
1001	12648.00	7.10	0.93	12707.00	7.60	1.00	12687.00	7.44	0.98	12054.00	2.07	0.27
1001	14254.00	9.33	0.40	16047.00	23.08	1.00	13142.00	0.80	0.03	13244.00	1.58	0.07
1001	12512.00	10.81	0.90	12642.00	11.97	1.00	12315.00	9.07	0.76	11610.00	2.83	0.24
1001	20055.00	9.91	1.00	19781.00	8.41	0.85	18420.00	0.95	0.10	19399.00	6.31	0.64
1001	14529.00	3.59	0.24	15680.00	11.79	0.80	14133.00	0.76	0.05	16098.00	14.77	1.00
1001	7625.00	7.47	0.71	7838.00	10.47	1.00	7449.00	4.99	0.48	7473.00	5.33	0.51
1001	17864.00	9.39	0.73	18429.00	12.85	1.00	17124.00	4.86	0.38	18349.00	12.36	0.96
1001	7855.00	6.05	0.35	8411.00	13.55	0.78	7409.00	0.03	0.00	8690.00	17.32	1.00
1001	8635.00	16.53	1.00	8070.00	8.91	0.54	8049.00	8.62	0.52	7420.00	0.13	0.01
1010	1664.00	0.36	0.67	1664.00	0.36	0.67	1664.00	0.36	0.67	1664.00	0.36	0.67
1010	1910.00	0.00	0.00	1919.00	0.47	0.31	1910.00	0.00	0.00	1910.00	0.00	0.00
1010	1824.00	0.00	0.00	1824.00	0.00	0.00	1857.00	1.81	1.00	1824.00	0.00	0.00
1010	1839.00	0.00	0.00	1839.00	0.00	0.00	1886.00	2.56	1.00	1839.00	0.00	0.00
1010	2203.00	0.00	0.00	2207.00	0.18	0.03	2331.00	5.81	1.00	2203.00	0.00	0.00
1010	1631.00	2.58	0.26	1745.00	9.75	1.00	1646.00	3.52	0.36	1631.00	2.58	0.26
1010	1972.00	0.00	0.00	2025.00	2.69	1.00	1985.00	0.66	0.25	1972.00	0.00	0.00
1010	1879.00	0.21	0.80	1879.00	0.21	0.80	1879.00	0.21	0.80	1879.00	0.21	0.80
1010	1440.00	0.00	0.00	1440.00	0.00	0.00	1440.00	0.00	0.00	1440.00	0.00	0.00
1010	1765.00	0.00	0.00	1765.00	0.00	0.00	1765.00	0.00	0.00	1765.00	0.00	0.00

Table A.4: Computational results for SPT, FFD, MFFD, EGI algorithms for n=20 (continued)

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
1011	3424.00	5.10	0.64	3424.00	5.10	0.64	3469.00	6.48	0.82	3516.00	7.92	1.00
1011	4341.00	0.00	0.00	4915.00	13.22	1.00	4341.00	0.00	0.00	4580.00	5.51	0.42
1011	4336.00	0.00	0.00	5742.00	32.43	1.00	4369.00	0.76	0.02	4369.00	0.76	0.02
1011	4182.00	0.00	0.00	4182.00	0.00	0.00	4229.00	1.12	1.00	4200.00	0.43	0.38
1011	5691.00	0.00	0.00	7558.00	32.81	1.00	6255.00	9.91	0.30	6331.00	11.25	0.34
1011	4479.00	9.73	0.34	5244.00	28.47	1.00	4565.00	11.8	0.42	4329.00	6.05	0.21
1011	3439.00	4.28	0.77	3329.00	0.94	0.17	3452.00	4.67	0.84	3329.00	0.94	0.17
1011	5319.00	3.42	0.43	5319.00	3.42	0.43	5319.00	3.42	0.43	5548.00	7.87	1.00
1011	2634.00	0.00	0.00	2634.00	0.00	0.00	2634.00	0.00	0.00	2634.00	0.00	0.00
1011	3046.00	0.00	0.00	3046.00	0.00	0.00	3046.00	0.00	0.00	3046.00	0.00	0.00
1100	4336.00	3.14	0.39	4217.00	0.31	0.04	4543.00	8.06	1.00	4336.00	3.14	0.39
1100	5853.00	3.61	0.87	5855.00	3.65	0.88	5652.00	0.05	0.01	5853.00	3.61	0.87
1100	5640.00	5.90	0.41	6086.00	14.27	1.00	5637.00	5.84	0.41	5640.00	5.90	0.41
1100	5573.00	2.39	0.28	5901.00	8.41	1.00	5682.00	4.39	0.52	5456.00	0.24	0.03
1100	8214.00	9.11	0.98	8203.00	8.97	0.96	7542.00	0.19	0.02	8228.00	9.30	1.00
1100	5467.00	4.49	0.18	6511.00	24.45	1.00	5233.00	0.02	0.00	5232.00	0.00	0.00
1100	4861.00	1.72	1.00	4835.00	1.17	0.68	4779.00	0.00	0.00	4861.00	1.72	1.00
1100	6604.00	5.44	0.94	6400.00	2.19	0.38	6600.00	5.38	0.93	6268.00	0.08	0.01
1100	4177.00	2.40	0.34	4367.00	7.06	1.00	4079.00	0.00	0.00	4177.00	2.40	0.34
1100	4837.00	3.71	0.77	4849.00	3.97	0.83	4832.00	3.60	0.75	4837.00	3.71	0.77
1101	16876.00	9.41	0.73	15437.00	0.08	0.01	17413.00	12.9	1.00	17009.00	10.28	0.80
1101	25873.00	10.70	0.99	25105.00	7.41	0.68	24132.00	3.25	0.30	24661.00	5.52	0.51
1101	29219.00	16.74	1.00	29019.00	15.94	0.95	26309.00	5.11	0.31	26779.00	6.99	0.42
1101	23093.00	4.55	0.75	23421.00	6.04	1.00	22764.00	3.07	0.51	23375.00	5.83	0.97
1101	40470.00	17.57	1.00	37323.00	8.43	0.48	34422.00	0.00	0.00	38589.00	12.11	0.69
1101	28159.00	9.44	0.32	33229.00	29.15	1.00	25729.00	0.00	0.00	33229.00	29.15	1.00
1101	15581.00	7.50	1.00	15220.00	5.01	0.67	14494.00	0.00	0.00	14990.00	3.42	0.46
1101	36694.00	10.18	0.74	33304.00	0.00	0.00	35274.00	5.92	0.43	37858.00	13.67	1.00
1101	16038.00	6.06	0.43	16637.00	10.02	0.71	15122.00	0.00	0.00	17260.00	14.14	1.00
1101	16869.00	11.04	0.97	16505.00	8.64	0.76	16488.00	8.53	0.75	15265.00	0.48	0.04
1110	3234.00	0.00	0.00	3234.00	0.00	0.00	3274.00	1.24	0.32	3234.00	0.00	0.00
1110	3786.00	0.00	0.00	3786.00	0.00	0.00	3786.00	0.00	0.00	3786.00	0.00	0.00
1110	3588.00	0.81	0.07	3991.00	12.14	1.00	3649.00	2.53	0.21	3588.00	0.81	0.07
1110	3659.00	0.63	0.19	3659.00	0.63	0.19	3757.00	3.33	1.00	3659.00	0.63	0.19
1110	4350.00	0.00	0.00	4601.00	5.77	1.00	4413.00	1.45	0.25	4350.00	0.00	0.00
1110	3259.00	2.52	0.20	3586.00	12.80	1.00	3289.00	3.46	0.27	3259.00	2.52	0.20
1110	3861.00	0.00	0.00	3861.00	0.00	0.00	3861.00	0.00	0.00	3861.00	0.00	0.00
1110	3744.00	0.11	0.07	3796.00	1.50	1.00	3744.00	0.11	0.07	3744.00	0.11	0.07
1110	2843.00	0.00	0.00	2843.00	0.00	0.00	2843.00	0.00	0.00	2843.00	0.00	0.00
1110	3587.00	0.00	0.00	3587.00	0.00	0.00	3587.00	0.00	0.00	3587.00	0.00	0.00
1111	6204.00	0.00	0.00	6204.00	0.00	0.00	6244.00	0.64	0.09	6230.00	0.42	0.06
1111	8791.00	0.00	0.00	10034.00	14.14	1.00	9352.00	6.38	0.45	8876.00	0.97	0.07
1111	8756.00	8.35	0.48	9482.00	17.34	1.00	8817.00	9.11	0.53	9482.00	17.34	1.00
1111	8477.00	5.75	0.70	8477.00	5.75	0.70	8575.00	6.97	0.85	8674.00	8.21	1.00
1111	11518.00	0.00	0.00	13701.00	18.95	1.00	11581.00	0.55	0.03	13701.00	18.95	1.00
1111	9115.00	9.78	0.22	12014.00	44.69	1.00	9290.00	11.8	0.27	8349.00	0.55	0.01
1111	6206.00	0.00	0.00	6206.00	0.00	0.00	6206.00	0.00	0.00	6206.00	0.00	0.00
1111	10824.00	3.42	0.93	10522.00	0.54	0.15	10824.00	3.42	0.93	10535.00	0.66	0.18
1111	5297.00	0.00	0.00	5297.00	0.00	0.00	5297.00	0.00	0.00	5956.00	12.44	1.00
1111	6219.00	0.00	0.00	6219.00	0.00	0.00	6219.00	0.00	0.00	6219.00	0.00	0.00
AVG	9977.30	3.59	0.43	9948.13	5.00	0.56	9711.73	2.34	0.35	9886.36	3.27	0.45

Table A.5: Computational results for SPT, FFD, MFFD, EGI algorithms for n=20 (continued)

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	3104.00	0.39	0.13	3094.00	0.06	0.02	3092.00	0.00	0.00
0000	3854.00	4.44	0.32	3742.00	1.41	0.10	3690.00	0.00	0.00
0000	3776.00	0.00	0.00	3776.00	0.00	0.00	3776.00	0.00	0.00
0000	4080.00	1.62	0.37	4015.00	0.00	0.00	4015.00	0.00	0.00
0000	5304.00	0.09	0.01	5299.00	0.00	0.00	5299.00	0.00	0.00
0000	4881.00	0.00	0.00	4881.00	0.00	0.00	4881.00	0.00	0.00
0000	2832.00	1.32	0.39	2798.00	0.11	0.03	2795.00	0.00	0.00
0000	4718.00	0.00	0.00	4718.00	0.00	0.00	4718.00	0.00	0.00
0000	3329.00	0.97	1.00	3297.00	0.00	0.00	3297.00	0.00	0.00
0000	2986.00	1.22	0.71	2959.00	0.31	0.18	2950.00	0.00	0.00
0001	13274.00	0.07	0.01	13265.0	0.00	0.00	13265.00	0.00	0.00
0001	17846.00	11.04	0.37	16810.0	4.60	0.15	16071.00	0.00	0.00
0001	20870.00	0.00	0.00	20870.0	0.00	0.00	20870.00	0.00	0.00
0001	17580.00	0.37	0.04	17515.0	0.00	0.00	17515.00	0.00	0.00
0001	26710.00	0.02	0.00	26705.0	0.00	0.00	26705.00	0.00	0.00
0001	28821.00	0.00	0.00	28821.0	0.00	0.00	28821.00	0.00	0.00
0001	9387.00	0.40	0.06	9352.00	0.02	0.00	9350.00	0.00	0.00
0001	27708.00	0.00	0.00	27708.0	0.00	0.00	27708.00	0.00	0.00
0001	13549.00	0.24	1.00	13517.0	0.00	0.00	13517.00	0.00	0.00
0001	10339.00	0.17	0.03	10325.0	0.04	0.01	10321.00	0.00	0.00
0010	2212.00	0.00	0.00	2212.00	0.00	0.00	2212.00	0.00	0.00
0010	2531.00	1.32	0.94	2498.00	0.00	0.00	2498.00	0.00	0.00
0010	2385.00	2.10	0.88	2340.00	0.17	0.07	2336.00	0.00	0.00
0010	2569.00	3.21	1.00	2489.00	0.00	0.00	2489.00	0.00	0.00
0010	2776.00	0.73	0.19	2772.00	0.58	0.15	2756.00	0.00	0.00
0010	2563.00	1.67	1.00	2523.00	0.08	0.05	2521.00	0.00	0.00
0010	2268.00	0.93	1.00	2247.00	0.00	0.00	2247.00	0.00	0.00
0010	2547.00	1.23	0.74	2539.00	0.91	0.55	2516.00	0.00	0.00
0010	2282.00	1.20	0.34	2277.00	0.98	0.28	2255.00	0.00	0.00
0010	2233.00	0.09	1.00	2231.00	0.00	0.00	2231.00	0.00	0.00
0011	4924.00	0.00	0.00	4924.00	0.00	0.00	4924.00	0.00	0.00
0011	6491.00	0.37	0.15	6468.00	0.02	0.01	6467.00	0.00	0.00
0011	6381.00	6.37	0.76	6011.00	0.20	0.02	5999.00	0.00	0.00
0011	6619.00	6.09	0.76	6239.00	0.00	0.00	6239.00	0.00	0.00
0011	7858.00	0.26	0.03	7854.00	0.20	0.02	7838.00	0.00	0.00
0011	7729.00	0.55	0.13	7689.00	0.03	0.01	7687.00	0.00	0.00
0011	4223.00	0.50	0.70	4202.00	0.00	0.00	4202.00	0.00	0.00
0011	7750.00	0.40	0.08	7742.00	0.30	0.06	7719.00	0.00	0.00
0011	5222.00	3.30	0.46	5224.00	3.34	0.47	5055.00	0.00	0.00
0011	4426.00	0.05	0.09	4424.00	0.00	0.00	4424.00	0.00	0.00
0100	6367.00	2.50	0.81	6215.00	0.05	0.02	6212.00	0.00	0.00
0100	7433.00	2.57	0.43	7247.00	0.00	0.00	7268.00	0.29	0.05
0100	8137.00	0.00	0.00	8137.00	0.00	0.00	8137.00	0.00	0.00
0100	8153.00	1.71	0.38	8035.00	0.24	0.05	8016.00	0.00	0.00
0100	10475.00	0.06	0.55	10469.0	0.00	0.00	10469.00	0.00	0.00
0100	9984.00	0.00	0.00	9984.00	0.00	0.00	9984.00	0.00	0.00
0100	5716.00	2.07	0.62	5622.00	0.39	0.12	5600.00	0.00	0.00
0100	9512.00	0.00	0.00	9512.00	0.00	0.00	9512.00	0.00	0.00
0100	6705.00	1.13	0.20	6630.00	0.00	0.00	6630.00	0.00	0.00
0100	6072.00	2.03	1.00	5961.00	0.17	0.08	5951.00	0.00	0.00
0101	26707.00	0.57	0.07	26555.0	0.00	0.00	26555.00	0.00	0.00
0101	33569.00	7.35	0.52	31271.0	0.00	0.00	31284.00	0.04	0.00
0101	45928.00	0.00	0.00	45928.0	0.00	0.00	45928.00	0.00	0.00
0101	35153.00	0.39	0.04	35035.0	0.05	0.01	35016.00	0.00	0.00
0101	53287.00	0.01	0.55	53281.0	0.00	0.00	53281.00	0.00	0.00
0101	57674.00	0.00	0.00	57674.0	0.00	0.00	57674.00	0.00	0.00
0101	18826.00	0.62	0.09	18736.0	0.14	0.02	18710.00	0.00	0.00
0101	55682.00	0.00	0.00	55682.0	0.00	0.00	55682.00	0.00	0.00
0101	27145.00	0.28	0.03	27070.0	0.00	0.00	27070.00	0.00	0.00
0101	21294.00	2.81	0.44	20712.0	0.00	0.00	21176.00	2.24	0.35

Table A.6: Computational results for Knap, 2Bin, GAPS algorithms for n=20

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0110	4614.00	3.01	1.00	4479.00	0.00	0.00	4479.00	0.00	0.00
0110	5030.00	0.92	0.68	4984.00	0.00	0.00	4984.00	0.00	0.00
0110	4828.00	1.39	0.86	4762.00	0.00	0.00	4762.00	0.00	0.00
0110	5134.00	3.26	1.00	5042.00	1.41	0.43	4972.00	0.00	0.00
0110	5705.00	3.86	1.00	5532.00	0.71	0.18	5493.00	0.00	0.00
0110	5217.00	2.03	1.00	5122.00	0.18	0.09	5113.00	0.00	0.00
0110	4524.00	0.73	1.00	4491.00	0.00	0.00	4491.00	0.00	0.00
0110	5420.00	6.09	1.00	5144.00	0.69	0.11	5109.00	0.00	0.00
0110	4593.00	0.50	0.17	4588.00	0.39	0.13	4570.00	0.00	0.00
0110	4547.00	1.00	1.00	4509.00	0.16	0.16	4502.00	0.00	0.00
0111	10264.00	1.33	0.16	10129.0	0.00	0.00	10129.00	0.00	0.00
0111	12950.00	0.20	0.08	12924.0	0.00	0.00	12931.00	0.05	0.02
0111	12784.00	6.52	0.79	12718.0	5.97	0.72	12002.00	0.00	0.00
0111	13234.00	6.11	0.75	12487.0	0.12	0.01	12472.00	0.00	0.00
0111	16793.00	7.26	0.87	15696.0	0.25	0.03	15657.00	0.00	0.00
0111	15508.00	0.67	0.16	15405.0	0.00	0.00	15926.00	3.38	0.79
0111	8434.00	0.39	0.56	8401.00	0.00	0.00	8401.00	0.00	0.00
0111	17813.00	14.49	1.00	15593.0	0.22	0.02	15558.00	0.00	0.00
0111	10473.00	0.22	0.06	10461.0	0.11	0.03	10450.00	0.00	0.00
0111	9191.00	3.41	0.47	8895.00	0.08	0.01	8888.00	0.00	0.00
1000	2177.00	1.07	0.24	2156.00	0.09	0.02	2154.00	0.00	0.00
1000	2945.00	3.95	0.75	2892.00	2.08	0.39	2833.00	0.00	0.00
1000	2793.00	4.37	0.18	2679.00	0.11	0.00	2676.00	0.00	0.00
1000	2787.00	0.58	0.08	2804.00	1.19	0.15	2771.00	0.00	0.00
1000	3926.00	0.46	0.05	4130.00	5.68	0.56	3908.00	0.00	0.00
1000	2788.00	1.71	0.16	2772.00	1.13	0.10	2741.00	0.00	0.00
1000	2431.00	2.66	0.69	2373.00	0.21	0.05	2368.00	0.00	0.00
1000	3086.00	0.00	0.00	3086.00	0.00	0.00	3244.00	5.12	0.42
1000	2084.00	2.46	0.36	2054.00	0.98	0.14	2034.00	0.00	0.00
1000	2389.00	4.51	0.81	2306.00	0.87	0.16	2286.00	0.00	0.00
1001	8417.00	2.22	0.18	8236.00	0.02	0.00	8234.00	0.00	0.00
1001	12669.00	7.28	0.96	11828.0	0.16	0.02	11809.00	0.00	0.00
1001	14254.00	9.33	0.40	13041.0	0.02	0.00	13038.00	0.00	0.00
1001	11307.00	0.14	0.01	11780.0	4.33	0.36	11291.00	0.00	0.00
1001	18314.00	0.37	0.04	19826.0	8.65	0.87	18247.00	0.00	0.00
1001	14536.00	3.64	0.25	14342.0	2.25	0.15	14026.00	0.00	0.00
1001	7484.00	5.48	0.52	7100.00	0.07	0.01	7095.00	0.00	0.00
1001	16330.00	0.00	0.00	16330.0	0.00	0.00	16330.00	0.00	0.00
1001	7855.00	6.05	0.35	7427.00	0.27	0.02	7407.00	0.00	0.00
1001	8245.00	11.27	0.68	7430.00	0.27	0.02	7410.00	0.00	0.00
1010	1667.00	0.54	1.00	1664.00	0.36	0.67	1658.00	0.00	0.00
1010	1939.00	1.52	1.00	1910.00	0.00	0.00	1910.00	0.00	0.00
1010	1824.00	0.00	0.00	1824.00	0.00	0.00	1824.00	0.00	0.00
1010	1839.00	0.00	0.00	1839.00	0.00	0.00	1839.00	0.00	0.00
1010	2203.00	0.00	0.00	2203.00	0.00	0.00	2203.00	0.00	0.00
1010	1633.00	2.70	0.28	1590.00	0.00	0.00	1590.00	0.00	0.00
1010	2014.00	2.13	0.79	1972.00	0.00	0.00	1972.00	0.00	0.00
1010	1880.00	0.27	1.00	1879.00	0.21	0.80	1875.00	0.00	0.00
1010	1440.00	0.00	0.00	1440.00	0.00	0.00	1440.00	0.00	0.00
1010	1765.00	0.00	0.00	1765.00	0.00	0.00	1765.00	0.00	0.00

Table A.7: Computational results for Knap, 2Bin, GAPS algorithms for n=20 (continued)

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
1011	3427.00	5.19	0.66	3266.00	0.25	0.03	3258.00	0.00	0.00
1011	4557.00	4.98	0.38	4341.00	0.00	0.00	4341.00	0.00	0.00
1011	4336.00	0.00	0.00	4336.00	0.00	0.00	4336.00	0.00	0.00
1011	4182.00	0.00	0.00	4182.00	0.00	0.00	4182.00	0.00	0.00
1011	5691.00	0.00	0.00	5691.00	0.00	0.00	5691.00	0.00	0.00
1011	4481.00	9.77	0.34	4082.00	0.00	0.00	4082.00	0.00	0.00
1011	3481.00	5.55	1.00	3301.00	0.09	0.02	3298.00	0.00	0.00
1011	5320.00	3.44	0.44	5319.00	3.42	0.43	5143.00	0.00	0.00
1011	2634.00	0.00	0.00	2634.00	0.00	0.00	2634.00	0.00	0.00
1011	3046.00	0.00	0.00	3046.00	0.00	0.00	3046.00	0.00	0.00
1100	4222.00	0.43	0.05	4244.00	0.95	0.12	4204.00	0.00	0.00
1100	5883.00	4.14	1.00	5655.00	0.11	0.03	5649.00	0.00	0.00
1100	5414.00	1.65	0.12	5326.00	0.00	0.00	5390.00	1.20	0.08
1100	5498.00	1.01	0.12	5467.00	0.44	0.05	5443.00	0.00	0.00
1100	7787.00	3.44	0.37	7805.00	3.68	0.40	7528.00	0.00	0.00
1100	5471.00	4.57	0.19	5467.00	4.49	0.18	5467.00	4.49	0.18
1100	4856.00	1.61	0.94	4783.00	0.08	0.05	4832.00	1.11	0.65
1100	6625.00	5.78	1.00	6302.00	0.62	0.11	6263.00	0.00	0.00
1100	4080.00	0.02	0.00	4124.00	1.10	0.16	4177.00	2.40	0.34
1100	4888.00	4.80	1.00	4697.00	0.71	0.15	4664.00	0.00	0.00
1101	15442.00	0.12	0.01	15464.0	0.26	0.02	15424.00	0.00	0.00
1101	25903.00	10.83	1.00	23372.0	0.00	0.00	24129.00	3.24	0.30
1101	26732.00	6.80	0.41	25029.0	0.00	0.00	26708.00	6.71	0.40
1101	22142.00	0.25	0.04	22111.0	0.11	0.02	22087.00	0.00	0.00
1101	36907.00	7.22	0.41	36925.0	7.27	0.41	35304.00	2.56	0.15
1101	28163.00	9.46	0.32	28159.0	9.44	0.32	28159.00	9.44	0.32
1101	15241.00	5.15	0.69	14544.0	0.34	0.05	15217.00	4.99	0.67
1101	36715.00	10.24	0.75	33560.0	0.77	0.06	33304.00	0.00	0.00
1101	15123.00	0.01	0.00	15167.0	0.30	0.02	16038.00	6.06	0.43
1101	16920.00	11.37	1.00	15225.0	0.22	0.02	15192.00	0.00	0.00
1110	3360.00	3.90	1.00	3234.00	0.00	0.00	3234.00	0.00	0.00
1110	3899.00	2.98	1.00	3786.00	0.00	0.00	3786.00	0.00	0.00
1110	3615.00	1.57	0.13	3559.00	0.00	0.00	3567.00	0.22	0.02
1110	3649.00	0.36	0.11	3636.00	0.00	0.00	3659.00	0.63	0.19
1110	4520.00	3.91	0.68	4350.00	0.00	0.00	4350.00	0.00	0.00
1110	3193.00	0.44	0.03	3213.00	1.07	0.08	3179.00	0.00	0.00
1110	3871.00	0.26	1.00	3861.00	0.00	0.00	3861.00	0.00	0.00
1110	3772.00	0.86	0.57	3740.00	0.00	0.00	3744.00	0.11	0.07
1110	2864.00	0.74	1.00	2843.00	0.00	0.00	2843.00	0.00	0.00
1110	3598.00	0.31	1.00	3587.00	0.00	0.00	3587.00	0.00	0.00
1111	6660.00	7.35	1.00	6204.00	0.00	0.00	6204.00	0.00	0.00
1111	9289.00	5.66	0.40	8791.00	0.00	0.00	8791.00	0.00	0.00
1111	8783.00	8.69	0.50	8081.00	0.00	0.00	8089.00	0.10	0.01
1111	8029.00	0.16	0.02	8016.00	0.00	0.00	8477.00	5.75	0.70
1111	12584.00	9.26	0.49	11518.0	0.00	0.00	11518.00	0.00	0.00
1111	8317.00	0.17	0.00	8337.00	0.41	0.01	8303.00	0.00	0.00
1111	6216.00	0.16	1.00	6206.00	0.00	0.00	6206.00	0.00	0.00
1111	10852.00	3.69	1.00	10466.0	0.00	0.00	10522.00	0.54	0.15
1111	5318.00	0.40	0.03	5297.00	0.00	0.00	5297.00	0.00	0.00
1111	6230.00	0.18	1.00	6219.00	0.00	0.00	6219.00	0.00	0.00
AVG	9753.87	2.40	0.43	9567.10	0.56	0.07	9555.74	0.38	0.04

Table A.8: Computational results for Knap, 2Bin, GAPS algorithms for n=20 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0000	0	0	0	0.07	0.75	4.77	0.97
0000	0	0	0	0.09	1.03	4.69	1.03
0000	0	0	0	0.06	1.29	4.48	1.23
0000	0	0	0	0.07	0.73	4.55	0.91
0000	0	0	0	0.07	1.17	4.7	1.14
0000	0	0	0	0.06	1.18	4.38	1.17
0000	0	0	0	0.04	0.49	4.76	0.96
0000	0	0	0.01	0.08	0.53	4.49	1.26
0000	0	0	0	0.02	0.76	4.54	0.97
0000	0	0.01	0	0.02	0.49	4.54	0.96
0001	0	0	0	0.05	0.75	4.7	0.98
0001	0	0	0	0.06	1.03	4.7	1.02
0001	0.01	0.01	0	0.05	1.29	4.55	1.13
0001	0	0	0	0.05	0.73	4.73	0.91
0001	0	0	0	0.04	1.17	4.67	1.01
0001	0.01	0	0	0.08	1.18	4.39	1.22
0001	0	0	0.01	0.05	0.49	4.55	0.88
0001	0	0	0	0.05	0.53	4.55	1.15
0001	0	0	0	0.06	0.76	4.73	0.88
0001	0	0.01	0	0.02	0.49	4.69	0.9
0010	0	0	0.01	0.04	0.3	4.73	0.89
0010	0.1	0	0	0.04	0.35	4.78	0.83
0010	0	0	0	0.05	0.39	4.58	0.93
0010	0	0	0	0.03	0.29	5.1	0.92
0010	0	0	0	0.03	0.37	4.81	0.86
0010	0	0	0	0.07	0.42	4.87	0.9
0010	0	0	0	0.04	0.18	5.4	0.84
0010	0	0	0	0.06	0.46	4.77	0.82
0010	0	0	0	0.04	0.32	4.79	0.87
0010	0	0	0	0.04	0.21	5.12	0.85
0011	0	0.01	0.1	0	0.26	4.78	0.94
0011	0	0	0	0.01	0.36	4.72	0.94
0011	0	0	0	0.04	0.4	4.83	0.99
0011	0	0	0	0.05	0.26	5.25	0.96
0011	0	0	0	0.07	0.38	4.99	0.97
0011	0	0	0.01	0.07	0.43	4.89	0.97
0011	0	0	0	0.05	0.17	5.08	0.96
0011	0	0	0	0.05	0.5	4.77	0.91
0011	0	0	0	0.07	0.29	4.94	0.94
0011	0	0	0	0.03	0.17	5.17	0.98
0100	0	0	0.01	0.05	0.71	4.93	0.88
0100	0	0	0	0.01	0.94	4.64	0.97
0100	0	0.01	0	0.02	1.35	4.59	1.2
0100	0	0	0	0.06	0.74	4.78	0.82
0100	0	0	0	0.07	1.39	4.47	1.17
0100	0	0	0	0.09	0.91	4.32	1.21
0100	0	0	0	0.06	0.45	4.85	0.92
0100	0	0.01	0	0.05	0.19	4.27	1.27
0100	0	0	0	0.02	0.7	4.31	0.89
0100	0	0	0	0.05	0.46	4.83	0.96
0101	0	0	0	0.04	0.68	4.94	0.95
0101	0	0	0	0.03	0.83	4.63	0.89
0101	0	0	0	0.04	1.29	4.48	1.13
0101	0	0	0.01	0.04	0.69	4.66	0.95
0101	0	0	0	0.05	1.3	4.55	1.1
0101	0	0	0	0.02	0.89	4.46	1.25
0101	0.01	0	0	0.03	0.47	4.86	0.97
0101	0	0	0	0.02	0.19	4.35	1.2
0101	0	0	0	0.06	0.78	4.71	0.92
0101	0	0	0	0.04	0.51	4.98	0.93

Table A.9: Computation times for n=20

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0110	0	0	0	0.04	0.32	5.48	0.9
0110	0	0	0	0.04	0.27	4.8	0.94
0110	0	0	0	0.04	0.4	4.87	0.89
0110	0	0	0	0.06	0.27	5.15	0.95
0110	0	0	0	0.05	0.34	4.88	0.92
0110	0	0	0	0.04	0.42	5	0.94
0110	0	0.01	0	0.07	0.17	5.76	0.93
0110	0	0	0	0.04	0.37	4.98	0.94
0110	0	0	0	0	0.25	4.77	0.97
0110	0	0	0	0.03	0.33	6.05	0.92
0111	0	0.01	0.01	0.06	0.26	5.2	0.86
0111	0.01	0.01	0.01	0.03	0.26	4.81	0.89
0111	0	0	0	0.03	0.4	4.85	0.8
0111	0	0	0	0.02	0.3	5.18	0.89
0111	0	0	0	0.08	0.34	4.62	0.94
0111	0	0	0	0.05	0.35	5.01	0.84
0111	0	0	0	0.07	0.21	9.18	0.79
0111	0.01	0	0	0.06	0.37	4.87	0.78
0111	0	0	0	0.06	0.29	4.93	0.96
0111	0	0	0	0.03	0.33	5.96	0.87
1000	0.01	0	0	0.05	0.46	4.69	0.94
1000	0	0	0	0.05	0.46	4.62	0.84
1000	0	0.01	0	0.04	0.75	4.26	0.98
1000	0.01	0	0.01	0.05	0.48	4.63	0.79
1000	0	0	0	0.07	0.66	4.55	0.81
1000	0	0	0	0.07	0.47	4.45	0.93
1000	0	0	0	0.06	0.39	4.93	0.91
1000	0	0	0	0.04	0.47	4.59	1.06
1000	0	0.01	0	0.03	0.36	4.48	0.95
1000	0.01	0	0.01	0.03	0.49	4.42	0.87
1001	0	0	0	0.03	0.48	4.44	0.92
1001	0	0	0	0.06	0.42	4.67	0.95
1001	0	0.01	0	0.03	0.75	4.45	0.84
1001	0	0	0	0.04	0.53	4.64	1.01
1001	0.01	0	0	0.05	0.64	4.5	0.9
1001	0	0	0	0.06	0.47	4.84	0.95
1001	0	0	0	0.04	0.42	5.31	0.85
1001	0	0	0	0.04	0.53	4.94	0.97
1001	0	0.01	0	0.06	0.34	4.82	0.92
1001	0	0	0	0.03	0.44	4.66	0.82
1010	0	0	0	0.04	0.19	5.56	0.92
1010	0	0	0.01	0	0.25	5	0.93
1010	0	0	0	0.04	0.28	5.51	0.88
1010	0	0	0	0.04	0.18	5.55	0.84
1010	0	0	0	0.02	0.29	5.34	0.85
1010	0	0	0	0.07	0.3	4.42	0.92
1010	0	0	0	0.08	0.17	7.02	0.94
1010	0	0	0	0.02	0.29	5.25	0.89
1010	0	0	0	0.05	0.18	5.05	0.87
1010	0	0	0	0.05	0.22	4.8	0.88

Table A.10: Computation times for n=20 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
1011	0	0.01	0	0.04	0.21	5.7	0.85
1011	0	0.01	0	0.04	0.2	4.95	0.88
1011	0	0	0	0.03	0.29	5.64	0.75
1011	0.01	0	0	0.02	0.19	4.93	0.89
1011	0	0	0	0.03	0.3	5.79	0.86
1011	0	0	0	0.03	0.27	4.79	0.83
1011	0	0	0	0.04	0.19	6.89	0.97
1011	0	0.01	0	0.04	0.26	5.47	0.84
1011	0	0	0	0.03	0.19	4.9	0.88
1011	0.01	0	0	0	0.16	4.81	0.92
1100	0	0	0.01	0.04	0.54	4.73	1.15
1100	0	0	0	0.06	0.48	4.77	1.18
1100	0	0	0	0.06	0.77	4.36	1.2
1100	0	0	0	0.05	0.49	4.73	1.34
1100	0	0	0	0.03	0.64	4.61	1.24
1100	0	0	0	0.02	0.4	4.59	1.28
1100	0	0	0.01	0.02	0.35	5.06	1.09
1100	0	0	0	0.05	0.59	4.9	1.28
1100	0	0	0	0.01	0.35	5.34	1.24
1100	0	0	0	0.03	0.42	5.31	1.22
1101	0	0	0	0.05	0.53	4.84	0.93
1101	0	0	0	0.05	0.56	4.73	0.96
1101	0	0	0	0.04	0.7	4.38	0.91
1101	0	0	0	0.08	0.45	4.87	0.94
1101	0	0	0	0.02	0.69	4.63	0.92
1101	0	0	0	0.04	0.38	4.48	0.95
1101	0	0	0.01	0.03	0.4	5.06	1.04
1101	0	0	0	0.04	0.51	4.67	0.98
1101	0	0	0	0.04	0.4	5.05	0.89
1101	0	0	0	0.02	0.39	4.91	0.98
1110	0	0	0	0.02	0.21	5.88	0.98
1110	0	0	0	0.04	0.19	5.44	0.95
1110	0	0	0	0.04	0.18	5.11	0.89
1110	0	0	0	0.03	0.18	6.12	0.86
1110	0	0	0	0.05	0.18	5.06	0.93
1110	0	0	0	0.03	0.19	5.01	0.9
1110	0	0.01	0	0.02	0.18	6.78	0.88
1110	0	0	0	0.06	0.3	5.37	0.9
1110	0	0	0	0.05	0.17	5.16	0.92
1110	0	0	0	0.06	0.2	10.47	0.84
1111	0	0	0	0.06	0.21	5.56	0.83
1111	0	0	0	0.05	0.19	6.06	0.91
1111	0.01	0	0	0.02	0.18	5.16	0.85
1111	0	0	0	0.02	0.22	6.37	0.79
1111	0	0	0	0.03	0.2	5.02	0.85
1111	0	0	0	0.04	0.2	5.01	0.83
1111	0	0	0	0.02	0.19	7.43	0.94
1111	0	0		0.04	0.3	5.58	0.81
1111	0.01	0	0	0.04	0.17	4.9	0.89
1111	0	0	0	0.05	0.23	11.3	0.91
AVG	0.001438	0.001063	0.001509	0.043187	0.46125	5.0535	0.953938

Table A.11: Computation times for n=20 (continued)

Appendix B

Computational Results for 50 Jobs

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	18901.50	2.25	0.68	18624.80	0.82	0.29	18734.30	1.37	0.50	18901.50	2.25	0.68
0001	76258.80	6.81	0.99	71911.60	1.06	0.23	73026.60	2.51	0.47	73358.10	3.02	0.47
0010	12491.60	0.05	0.08	12588.40	0.84	0.62	12568.30	0.67	0.62	12491.60	0.05	0.08
0011	19085.20	0.61	0.16	19793.80	4.34	0.87	19351.80	2.00	0.41	19100.50	0.69	0.26
0100	38453.70	2.10	0.63	37953.10	0.85	0.32	38017.60	0.97	0.34	38453.70	2.10	0.63
0101	156268.80	6.78	0.89	148177.80	1.50	0.19	148888.20	1.84	0.24	151372.50	3.60	0.39
0110	25164.60	0.04	0.03	25366.70	0.85	0.52	25365.00	0.85	0.59	25164.60	0.04	0.03
0111	38668.80	0.79	0.14	39983.00	4.20	0.79	39606.40	3.26	0.56	38683.80	0.83	0.15
1000	14585.70	1.52	0.39	15017.90	4.29	0.92	14633.00	1.92	0.52	14585.70	1.52	0.39
1001	55564.10	5.07	0.78	55145.40	4.29	0.68	53756.40	1.90	0.30	54632.20	3.54	0.57
1010	9836.90	0.07	0.13	9944.90	1.11	0.58	9885.30	0.57	0.39	9836.90	0.07	0.13
1011	13913.10	0.84	0.16	15456.40	12.42	0.62	14365.80	3.98	0.59	13927.00	0.94	0.16
1100	28844.10	2.11	0.42	29958.50	5.79	0.90	28913.70	2.37	0.41	28844.10	2.11	0.42
1101	110351.00	6.42	0.78	111355.00	7.13	0.79	107130.00	3.32	0.37	107212.60	3.45	0.40
1110	19342.80	0.00	0.00	19767.00	2.23	0.53	19431.30	0.47	0.15	19342.80	0.00	0.00
1111	27045.30	0.17	0.02	29994.10	11.35	0.69	27837.80	2.95	0.46	27045.30	0.17	0.02
AVG	41548.50	2.23	0.39	41314.90	3.94	0.60	40719.47	1.93	0.43	40809.56	1.52	0.30

Table B.1: Averages over ten replications for n=50

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	18842.20	1.98	0.64	18558.20	0.46	0.12	18491.00	0.09	0.03
0001	72933.00	2.39	0.32	71836.70	0.81	0.07	71251.30	0.05	0.01
0010	12528.10	0.34	0.45	12491.60	0.05	0.08	12485.40	0.00	0.00
0011	19083.10	0.61	0.30	18988.60	0.10	0.02	19015.80	0.23	0.07
0100	38773.00	2.88	0.91	37865.90	0.56	0.19	37719.90	0.19	0.03
0101	151623.20	3.48	0.48	148229.00	1.25	0.14	146626.70	0.32	0.03
0110	25355.00	0.80	0.70	25163.60	0.04	0.03	25154.40	0.00	0.00
0111	38763.90	1.03	0.37	38709.70	0.93	0.14	38365.60	0.00	0.00
1000	14663.40	2.04	0.59	14431.20	0.52	0.15	14355.40	0.00	0.00
1001	55370.00	4.79	0.78	53253.30	1.07	0.17	52657.60	0.00	0.00
1010	9856.50	0.26	0.50	9835.90	0.06	0.12	9830.60	0.00	0.00
1011	13883.90	0.61	0.14	13828.50	0.17	0.02	13806.00	0.00	0.00
1100	28969.40	2.48	0.45	28556.40	1.14	0.21	28258.30	0.13	0.02
1101	109269.70	5.18	0.61	105060.00	1.33	0.14	104084.90	0.65	0.10
1110	19376.40	0.18	0.33	19342.80	0.00	0.00	19342.80	0.00	0.00
1111	27114.40	0.44	0.05	26999.10	0.00	0.00	27045.30	0.17	0.02
AVG	41025.33	1.84	0.48	40196.91	0.53	0.10	39905.69	0.11	0.02

Table B.2: Averages over ten replications for n=50 (continued)

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	16858.00	0.18	0.07	16936.00	0.65	0.26	16936.00	0.65	0.26	16858.00	0.18	0.07
0000	20138.00	2.88	0.66	19914.00	1.74	0.40	19914.00	1.74	0.40	20138.00	2.88	0.66
0000	16207.00	0.39	0.21	16288.00	0.89	0.48	16441.00	1.84	1.00	16207.00	0.39	0.21
0000	17694.00	0.96	0.42	17644.00	0.67	0.29	17644.00	0.67	0.29	17694.00	0.96	0.42
0000	18305.00	1.98	1.00	18291.00	1.90	0.96	18291.00	1.90	0.96	18305.00	1.98	1.00
0000	17864.00	3.24	1.00	17304.00	0.00	0.00	17484.00	1.04	0.32	17864.00	3.24	1.00
0000	24149.00	1.94	1.00	23689.00	0.00	0.00	24151.00	1.95	1.00	24149.00	1.94	1.00
0000	17692.00	1.17	0.47	17502.00	0.09	0.03	17502.00	0.09	0.03	17692.00	1.17	0.47
0000	20404.00	4.37	1.00	19932.00	1.95	0.45	19782.00	1.19	0.27	20404.00	4.37	1.00
0000	19704.00	5.39	1.00	18748.00	0.28	0.05	19198.00	2.69	0.50	19704.00	5.39	1.00
0001	62962.00	2.64	1.00	62362.00	1.67	0.63	62362.00	1.67	0.63	62408.00	1.74	0.66
0001	74038.00	6.33	1.00	71658.00	2.92	0.46	71658.00	2.92	0.46	71679.00	2.95	0.47
0001	56247.00	3.18	0.91	54898.00	0.70	0.20	56151.00	3.00	0.86	56415.00	3.49	1.00
0001	68862.00	4.24	1.00	67090.00	1.56	0.37	67090.00	1.56	0.37	67052.00	1.50	0.35
0001	66605.00	6.01	0.96	64659.00	2.91	0.46	64659.00	2.91	0.46	66770.00	6.27	1.00
0001	74184.00	8.90	1.00	68124.00	0.00	0.00	69954.00	2.69	0.30	69926.00	2.65	0.30
0001	106546.0	3.49	1.00	102957.0	0.00	0.00	106548.0	3.49	1.00	102957.00	0.00	0.00
0001	68444.00	4.49	1.00	65936.00	0.66	0.15	65936.00	0.66	0.15	65996.00	0.75	0.17
0001	101156.0	16.13	1.00	87244.00	0.16	0.01	87430.00	0.37	0.02	94234.00	8.18	0.51
0001	83544.00	12.69	1.00	74188.00	0.07	0.01	78478.00	5.86	0.46	76144.00	2.71	0.21
0010	12243.00	0.19	0.13	12402.00	1.49	1.00	12268.00	0.39	0.26	12243.00	0.19	0.13
0010	12866.00	0.00	0.00	12962.00	0.75	1.00	12890.00	0.19	0.25	12866.00	0.00	0.00
0010	12439.00	0.00	0.00	12511.00	0.58	0.71	12541.00	0.82	1.00	12439.00	0.00	0.00
0010	12174.00	0.00	0.00	12489.00	2.59	1.00	12278.00	0.85	0.33	12174.00	0.00	0.00
0010	12265.00	0.10	0.10	12367.00	0.93	0.97	12371.00	0.96	1.00	12265.00	0.10	0.10
0010	12416.00	0.16	0.50	12416.00	0.16	0.50	12416.00	0.16	0.50	12416.00	0.16	0.50
0010	13281.00	0.05	0.06	13281.00	0.05	0.06	13388.00	0.86	1.00	13281.00	0.05	0.06
0010	12172.00	0.00	0.00	12396.00	1.84	1.00	12358.00	1.53	0.83	12172.00	0.00	0.00
0010	12496.00	0.00	0.00	12496.00	0.00	0.00	12496.00	0.00	0.00	12496.00	0.00	0.00
0010	12564.00	0.00	0.00	12564.00	0.00	0.00	12677.00	0.90	1.00	12564.00	0.00	0.00
0011	18232.00	1.75	0.46	18604.00	3.83	1.00	18392.00	2.65	0.69	18232.00	1.75	0.46
0011	20104.00	0.00	0.00	20200.00	0.48	1.00	20128.00	0.12	0.25	20200.00	0.48	1.00
0011	17929.00	0.41	0.08	18725.00	4.87	1.00	18041.00	1.04	0.21	17939.00	0.46	0.10
0011	18078.00	0.00	0.00	18762.00	3.78	1.00	18182.00	0.58	0.15	18078.00	0.00	0.00
0011	18889.00	2.03	0.78	18853.00	1.84	0.71	18995.00	2.60	1.00	18889.00	2.03	0.78
0011	18796.00	1.29	0.16	20068.00	8.15	1.00	19192.00	3.43	0.42	18796.00	1.29	0.16
0011	22072.00	0.66	0.10	23413.00	6.78	1.00	22703.00	3.54	0.52	22119.00	0.88	0.13
0011	18028.00	0.00	0.00	19069.00	5.77	1.00	18458.00	2.39	0.41	18028.00	0.00	0.00
0011	19440.00	0.00	0.00	19440.00	0.00	0.00	19440.00	0.00	0.00	19440.00	0.00	0.00
0011	19284.00	0.00	0.00	20804.00	7.88	1.00	19987.00	3.65	0.46	19284.00	0.00	0.00
0100	34349.00	1.20	0.57	34121.00	0.53	0.25	34121.00	0.53	0.25	34349.00	1.20	0.57
0100	40943.00	2.62	0.45	40404.00	1.27	0.22	40397.00	1.26	0.22	40943.00	2.62	0.45
0100	33508.00	1.33	0.73	33579.00	1.54	0.85	33579.00	1.54	0.85	33508.00	1.33	0.73
0100	35810.00	1.05	0.42	35606.00	0.47	0.19	35606.00	0.47	0.19	35810.00	1.05	0.42
0100	36679.00	1.73	0.74	36902.00	2.35	1.00	36902.00	2.35	1.00	36679.00	1.73	0.74
0100	36381.00	2.50	0.98	35904.00	1.16	0.45	35904.00	1.16	0.45	36381.00	2.50	0.98
0100	48319.00	1.93	0.34	47405.00	0.00	0.00	48319.00	1.93	0.34	48319.00	1.93	0.34
0100	36091.00	1.31	0.53	35647.00	0.06	0.02	35647.00	0.06	0.02	36091.00	1.31	0.53
0100	42168.00	5.75	1.00	40296.00	1.06	0.18	40039.00	0.41	0.07	42168.00	5.75	1.00
0100	40289.00	1.58	0.56	39667.00	0.01	0.00	39662.00	0.00	0.00	40289.00	1.58	0.56
0101	130399.0	5.79	1.00	124973.0	1.39	0.24	124973.0	1.39	0.24	124889.00	1.32	0.23
0101	152439.0	6.29	0.62	148204.0	3.34	0.33	148197.0	3.34	0.33	152480.00	6.32	0.62
0101	122828.0	5.85	1.00	116299.0	0.22	0.04	116299.0	0.22	0.04	119060.00	2.60	0.44
0101	137314.0	4.63	1.00	131742.0	0.38	0.08	131742.0	0.38	0.08	133711.00	1.88	0.41
0101	136203.0	8.06	0.86	133646.0	6.03	0.64	133646.0	6.03	0.64	137842.00	9.36	1.00
0101	153421.0	6.41	1.00	148544.0	3.02	0.47	148544.0	3.02	0.47	148544.00	3.02	0.47
0101	212560.0	3.48	0.45	205409.0	0.00	0.00	212560.0	3.48	0.45	205409.00	0.00	0.00
0101	137179.0	4.40	1.00	132118.0	0.55	0.13	132118.0	0.55	0.13	132292.00	0.68	0.16
0101	207336.0	19.53	1.00	173496.0	0.02	0.00	173461.0	0.00	0.00	192048.00	10.72	0.55
0101	173009.0	3.39	0.92	167347.0	0.00	0.00	167342.0	0.00	0.00	167450.00	0.06	0.02

Table B.3: Computational results for SPT, FFD, MFFD, EGI algorithms for n=50

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0110	24677.00	0.15	0.11	24843.00	0.83	0.59	24985.00	1.40	1.00	24677.00	0.15	0.11
0110	25823.00	0.00	0.00	26347.00	2.03	1.00	25894.00	0.27	0.14	25823.00	0.00	0.00
0110	25012.00	0.00	0.00	25393.00	1.52	1.00	25312.00	1.20	0.79	25012.00	0.00	0.00
0110	24495.00	0.05	0.03	24854.00	1.52	0.78	24961.00	1.96	1.00	24495.00	0.05	0.03
0110	24550.00	0.00	0.00	24729.00	0.73	0.66	24659.00	0.44	0.40	24550.00	0.00	0.00
0110	25005.00	0.10	0.15	25005.00	0.10	0.15	25145.00	0.66	0.97	25005.00	0.10	0.15
0110	26671.00	0.00	0.00	26671.00	0.00	0.00	26671.00	0.00	0.00	26671.00	0.00	0.00
0110	24776.00	0.10	0.06	25163.00	1.66	0.95	25186.00	1.76	1.00	24776.00	0.10	0.06
0110	25168.00	0.00	0.00	25168.00	0.00	0.00	25168.00	0.00	0.00	25168.00	0.00	0.00
0110	25469.00	0.00	0.00	25494.00	0.10	0.07	25669.00	0.79	0.58	25469.00	0.00	0.00
0111	36655.00	1.92	0.18	38356.00	6.65	0.64	39693.00	10.36	1.00	36655.00	1.92	0.18
0111	40299.00	0.73	0.17	41747.00	4.35	1.00	40370.00	0.91	0.21	40370.00	0.91	0.21
0111	36452.00	1.22	0.39	37151.00	3.17	1.00	36972.00	2.67	0.84	36452.00	1.22	0.39
0111	36939.00	0.70	0.17	38197.00	4.13	1.00	37893.00	3.30	0.80	36939.00	0.70	0.17
0111	37616.00	0.00	0.00	37795.00	0.48	0.66	37725.00	0.29	0.40	37616.00	0.00	0.00
0111	37765.00	1.25	0.20	39655.00	6.32	1.00	38470.00	3.15	0.50	37765.00	1.25	0.20
0111	44788.00	1.39	0.24	46727.00	5.78	1.00	46519.00	5.31	0.92	44867.00	1.57	0.27
0111	37169.00	0.73	0.10	38449.00	4.20	0.57	38065.00	3.15	0.43	37169.00	0.73	0.10
0111	39376.00	0.00	0.00	39376.00	0.00	0.00	39376.00	0.00	0.00	39376.00	0.00	0.00
0111	39629.00	0.00	0.00	42377.00	6.93	1.00	40981.00	3.41	0.49	39629.00	0.00	0.00
1000	12461.00	0.45	0.28	12604.00	1.60	1.00	12517.00	0.90	0.56	12461.00	0.45	0.28
1000	15997.00	1.03	0.20	16635.00	5.06	1.00	16103.00	1.70	0.34	15997.00	1.03	0.20
1000	13442.00	0.31	0.10	13829.00	3.20	1.00	13659.00	1.93	0.60	13442.00	0.31	0.10
1000	12529.00	0.98	0.39	12721.00	2.53	1.00	12621.00	1.72	0.68	12529.00	0.98	0.39
1000	12537.00	0.55	0.38	12651.00	1.47	1.00	12537.00	0.55	0.38	12537.00	0.55	0.38
1000	14117.00	1.42	0.18	15013.00	7.86	1.00	14394.00	3.41	0.43	14117.00	1.42	0.18
1000	19849.00	2.13	0.21	21446.00	10.35	1.00	19628.00	0.99	0.10	19849.00	2.13	0.21
1000	13212.00	1.80	0.84	13209.00	1.78	0.83	13209.00	1.78	0.83	13212.00	1.80	0.84
1000	16272.00	3.06	0.92	15986.00	1.25	0.38	16248.00	2.91	0.88	16272.00	3.06	0.92
1000	15441.00	3.50	0.45	16085.00	7.82	1.00	15414.00	3.32	0.42	15441.00	3.50	0.45
1001	43212.00	3.51	0.72	42711.00	2.31	0.47	42141.00	0.95	0.19	43404.00	3.97	0.81
1001	56763.00	3.64	0.69	57183.00	4.41	0.84	55343.00	1.05	0.20	55894.00	2.05	0.39
1001	45217.00	1.82	0.46	46154.00	3.93	1.00	45208.00	1.80	0.46	44942.00	1.20	0.31
1001	44015.00	3.50	0.99	44034.00	3.54	1.00	43415.00	2.09	0.59	43239.00	1.67	0.47
1001	43309.00	2.45	0.47	43031.00	1.79	0.35	42721.00	1.06	0.20	44462.00	5.18	1.00
1001	56277.00	6.06	0.88	56708.00	6.87	1.00	54292.00	2.32	0.34	53402.00	0.64	0.09
1001	82549.00	7.09	0.63	85818.00	11.33	1.00	80029.00	3.82	0.34	81330.00	5.51	0.49
1001	46752.00	4.32	0.98	45029.00	0.48	0.11	45029.00	0.48	0.11	46762.00	4.35	0.98
1001	74676.00	10.33	1.00	68267.00	0.86	0.08	69942.00	3.33	0.32	72548.00	7.18	0.70
1001	62871.00	8.01	1.00	62519.00	7.41	0.92	59444.00	2.13	0.27	60339.00	3.66	0.46
1010	9401.00	0.11	0.56	9405.00	0.15	0.78	9401.00	0.11	0.56	9401.00	0.11	0.56
1010	10353.00	0.00	0.00	11193.00	8.11	1.00	10536.00	1.77	0.22	10353.00	0.00	0.00
1010	10382.00	0.08	0.08	10387.00	0.13	0.13	10382.00	0.08	0.08	10382.00	0.08	0.08
1010	8921.00	0.17	0.21	8972.00	0.74	0.90	8979.00	0.82	1.00	8921.00	0.17	0.21
1010	8589.00	0.22	0.45	8579.00	0.11	0.21	8612.00	0.49	1.00	8589.00	0.22	0.45
1010	9918.00	0.00	0.00	9963.00	0.45	1.00	9918.00	0.00	0.00	9918.00	0.00	0.00
1010	11177.00	0.00	0.00	11302.00	1.12	1.00	11177.00	0.00	0.00	11177.00	0.00	0.00
1010	9384.00	0.12	0.05	9389.00	0.17	0.07	9604.00	2.46	1.00	9384.00	0.12	0.05
1010	10116.00	0.00	0.00	10131.00	0.15	0.68	10116.00	0.00	0.00	10116.00	0.00	0.00
1010	10128.00	0.00	0.00	10128.00	0.00	0.00	10128.00	0.00	0.00	10128.00	0.00	0.00

Table B.4: Computational results for SPT, FFD, MFFD, EGI algorithms for n=50 (continued)

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
1011	12782.00	1.36	0.08	14886.00	18.04	1.00	13183.00	4.54	0.25	12782.00	1.36	0.08
1011	14931.00	0.00	0.00	18000.00	20.55	1.00	15332.00	2.69	0.13	15070.00	0.93	0.05
1011	14257.00	1.16	0.31	14107.00	0.09	0.02	14623.00	3.75	1.00	14257.00	1.16	0.31
1011	12035.00	1.59	0.06	14973.00	26.39	1.00	12093.00	2.08	0.08	12035.00	1.59	0.06
1011	11725.00	1.87	0.90	11519.00	0.08	0.04	11748.00	2.07	1.00	11725.00	1.87	0.90
1011	13793.00	0.00	0.00	14148.00	2.57	0.58	14406.00	4.44	1.00	13793.00	0.00	0.00
1011	17238.00	0.00	0.00	17572.00	1.94	0.40	18065.00	4.80	1.00	17238.00	0.00	0.00
1011	12996.00	1.43	0.04	17297.00	35.00	1.00	13388.00	4.49	0.13	12996.00	1.43	0.04
1011	14826.00	1.03	0.18	14841.00	1.13	0.20	15526.00	5.80	1.00	14826.00	1.03	0.18
1011	14548.00	0.00	0.00	17221.00	18.37	1.00	15294.00	5.13	0.28	14548.00	0.00	0.00
1100	25091.00	0.50	0.21	25557.00	2.37	1.00	25097.00	0.53	0.22	25091.00	0.50	0.21
1100	31331.00	2.14	0.41	32293.00	5.28	1.00	31064.00	1.27	0.24	31331.00	2.14	0.41
1100	27019.00	1.71	0.46	26945.00	1.43	0.39	27544.00	3.69	1.00	27019.00	1.71	0.46
1100	24771.00	1.02	0.32	25296.00	3.16	1.00	25070.00	2.24	0.71	24771.00	1.02	0.32
1100	25024.00	2.43	0.90	24846.00	1.70	0.63	24527.00	0.39	0.15	25024.00	2.43	0.90
1100	28175.00	1.77	0.15	30987.00	11.93	1.00	28773.00	3.93	0.33	28175.00	1.77	0.15
1100	38715.00	3.90	0.30	42046.00	12.84	1.00	38937.00	4.50	0.35	38715.00	3.90	0.30
1100	25793.00	2.17	0.39	26669.00	5.64	1.00	25887.00	2.55	0.45	25793.00	2.17	0.39
1100	31879.00	3.07	0.79	32134.00	3.89	1.00	31257.00	1.06	0.27	31879.00	3.07	0.79
1100	30643.00	2.43	0.25	32812.00	9.68	1.00	30981.00	3.56	0.37	30643.00	2.43	0.25
1101	88121.00	3.52	0.95	87267.00	2.52	0.68	85157.00	0.04	0.01	85482.00	0.42	0.11
1101	111431.0	5.75	0.76	113293.0	7.52	1.00	108014.0	2.51	0.33	112504.00	6.77	0.90
1101	92414.00	7.93	1.00	86598.00	1.14	0.14	89749.00	4.82	0.61	91452.00	6.81	0.86
1101	87783.00	4.63	0.88	88292.00	5.23	1.00	87720.00	4.55	0.87	85253.00	1.61	0.31
1101	90229.00	6.62	1.00	88058.00	4.05	0.61	85277.00	0.76	0.12	86452.00	2.15	0.33
1101	111115.0	4.67	0.36	119883.0	12.93	1.00	109161.0	2.83	0.22	106496.00	0.32	0.02
1101	162267.0	7.55	0.61	169459.0	12.31	1.00	158601.0	5.12	0.42	153069.00	1.45	0.12
1101	92178.00	5.95	0.66	94882.00	9.06	1.00	89073.00	2.38	0.26	88512.00	1.74	0.19
1101	145279.0	11.25	1.00	137077.0	4.97	0.44	137384.0	5.20	0.46	145282.00	11.25	1.00
1101	122693.0	6.34	0.55	128741.0	11.58	1.00	121164.0	5.01	0.43	117624.00	1.95	0.17
1110	18555.00	0.00	0.00	18558.00	0.02	0.01	18924.00	1.99	1.00	18555.00	0.00	0.00
1110	20360.00	0.00	0.00	21855.00	7.34	1.00	20615.00	1.25	0.17	20360.00	0.00	0.00
1110	20467.00	0.00	0.00	20475.00	0.04	0.19	20467.00	0.00	0.00	20467.00	0.00	0.00
1110	17571.00	0.00	0.00	18258.00	3.91	1.00	17736.00	0.94	0.24	17571.00	0.00	0.00
1110	16645.00	0.00	0.00	16645.00	0.00	0.00	16645.00	0.00	0.00	16645.00	0.00	0.00
1110	19715.00	0.00	0.00	19715.00	0.00	0.00	19715.00	0.00	0.00	19715.00	0.00	0.00
1110	21880.00	0.00	0.00	22132.00	1.15	1.00	21880.00	0.00	0.00	21880.00	0.00	0.00
1110	18143.00	0.00	0.00	19768.00	8.96	1.00	18239.00	0.53	0.06	18143.00	0.00	0.00
1110	19965.00	0.00	0.00	19968.00	0.02	0.07	19965.00	0.00	0.00	19965.00	0.00	0.00
1110	20127.00	0.00	0.00	20296.00	0.84	1.00	20127.00	0.00	0.00	20127.00	0.00	0.00
1111	24825.00	0.00	0.00	33070.00	33.21	1.00	25524.00	2.82	0.08	24825.00	0.00	0.00
1111	28910.00	0.00	0.00	31305.00	8.28	1.00	29615.00	2.44	0.29	28910.00	0.00	0.00
1111	27804.00	0.90	0.04	33747.00	22.46	1.00	28490.00	3.39	0.15	27804.00	0.90	0.04
1111	23623.00	0.00	0.00	24310.00	2.91	1.00	24144.00	2.21	0.76	23623.00	0.00	0.00
1111	22315.00	0.00	0.00	22315.00	0.00	0.00	22315.00	0.00	0.00	22315.00	0.00	0.00
1111	27690.00	0.78	0.20	27690.00	0.78	0.20	28524.00	3.82	1.00	27690.00	0.78	0.20
1111	33544.00	0.00	0.00	34660.00	3.33	0.67	35211.00	4.97	1.00	33544.00	0.00	0.00
1111	24178.00	0.00	0.00	29902.00	23.67	1.00	24274.00	0.40	0.02	24178.00	0.00	0.00
1111	29037.00	0.00	0.00	29040.00	0.01	0.00	30350.00	4.52	1.00	29037.00	0.00	0.00
1111	28527.00	0.00	0.00	33902.00	18.84	1.00	29931.00	4.92	0.26	28527.00	0.00	0.00
AVG	41548.50	2.23	0.39	41314.90	3.94	0.60	40719.47	1.93	0.43	40809.56	1.52	0.30

Table B.5: Computational results for SPT, FFD, MFFD, EGI algorithms for n=50 (continued)

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	17242.00	2.47	1.00	16858.00	0.18	0.07	16827.00	0.00	0.00
0000	20429.00	4.37	1.00	19652.00	0.40	0.09	19574.00	0.00	0.00
0000	16213.00	0.43	0.23	16144.00	0.00	0.00	16168.00	0.15	0.08
0000	17930.00	2.31	1.00	17594.00	0.39	0.17	17526.00	0.00	0.00
0000	18004.00	0.30	0.15	17950.00	0.00	0.00	17971.00	0.12	0.06
0000	17542.00	1.38	0.43	17322.00	0.10	0.03	17304.00	0.00	0.00
0000	23986.00	1.25	0.64	23690.00	0.00	0.00	23689.00	0.00	0.00
0000	17921.00	2.48	1.00	17648.00	0.92	0.37	17487.00	0.00	0.00
0000	19692.00	0.73	0.17	19550.00	0.00	0.00	19668.00	0.60	0.14
0000	19463.00	4.10	0.76	19174.00	2.56	0.47	18696.00	0.00	0.00
0001	61764.00	0.69	0.26	61500.00	0.26	0.10	61340.00	0.00	0.00
0001	74021.00	6.31	1.00	69701.00	0.10	0.02	69628.00	0.00	0.00
0001	55043.00	0.97	0.28	54514.00	0.00	0.00	54729.00	0.39	0.11
0001	66392.00	0.50	0.12	66059.00	0.00	0.00	66157.00	0.15	0.03
0001	63268.00	0.70	0.11	62942.00	0.18	0.03	62831.00	0.00	0.00
0001	70012.00	2.77	0.31	68142.00	0.03	0.00	68124.00	0.00	0.00
0001	103254.0	0.29	0.08	102957.00	0.00	0.00	102957.00	0.00	0.00
0001	65989.00	0.74	0.16	65556.00	0.08	0.02	65506.00	0.00	0.00
0001	88684.00	1.81	0.11	88542.00	1.65	0.10	87105.00	0.00	0.00
0001	80903.00	9.13	0.72	78454.00	5.82	0.46	74136.00	0.00	0.00
0010	12359	1.14	0.76	12243.00	0.19	0.13	12220.00	0.00	0.00
0010	12941	0.58	0.78	12866.00	0.00	0.00	12866.00	0.00	0.00
0010	12451	0.10	0.12	12439.00	0.00	0.00	12439.00	0.00	0.00
0010	12178	0.03	0.01	12174.00	0.00	0.00	12174.00	0.00	0.00
0010	12339	0.70	0.73	12265.00	0.10	0.10	12253.00	0.00	0.00
0010	12436	0.32	1.00	12416.00	0.16	0.50	12396.00	0.00	0.00
0010	13285	0.08	0.10	13281.00	0.05	0.06	13274.00	0.00	0.00
0010	12172	0.00	0.00	12172.00	0.00	0.00	12172.00	0.00	0.00
0010	12553	0.46	1.00	12496.00	0.00	0.00	12496.00	0.00	0.00
0010	12567	0.02	0.03	12564.00	0.00	0.00	12564.00	0.00	0.00
0011	18235	1.77	0.46	17998.00	0.45	0.12	17918.00	0.00	0.00
0011	20179	0.37	0.78	20104.00	0.00	0.00	20104.00	0.00	0.00
0011	17951	0.53	0.11	17856.00	0.00	0.00	17939.00	0.46	0.10
0011	18082	0.02	0.01	18078.00	0.00	0.00	18078.00	0.00	0.00
0011	18825	1.69	0.65	18513.00	0.00	0.00	18739.00	1.22	0.47
0011	18596	0.22	0.03	18647.00	0.49	0.06	18556.00	0.00	0.00
0011	21927	0.00	0.00	21938.00	0.05	0.01	22072.00	0.66	0.10
0011	18028	0.00	0.00	18028.00	0.00	0.00	18028.00	0.00	0.00
0011	19721	1.45	1.00	19440.00	0.00	0.00	19440.00	0.00	0.00
0011	19287	0.02	0.00	19284.00	0.00	0.00	19284.00	0.00	0.00
0100	34655	2.10	1.00	34089.00	0.44	0.21	33941.00	0.00	0.00
0100	42204	5.79	1.00	39896.00	0.00	0.00	40283.00	0.97	0.17
0100	33670	1.82	1.00	33232.00	0.49	0.27	33069.00	0.00	0.00
0100	36315	2.47	1.00	35617.00	0.51	0.20	35438.00	0.00	0.00
0100	36824	2.14	0.91	36125.00	0.20	0.08	36054.00	0.00	0.00
0100	36403	2.56	1.00	35911.00	1.18	0.46	35493.00	0.00	0.00
0100	50117	5.72	1.00	48319.00	1.93	0.34	47405.00	0.00	0.00
0100	36500	2.45	1.00	35934.00	0.86	0.35	35626.00	0.00	0.00
0100	40270	0.99	0.17	39874.00	0.00	0.00	40228.00	0.89	0.15
0100	40772	2.80	1.00	39662.00	0.00	0.00	39662.00	0.00	0.00
0101	124377	0.90	0.16	123448.00	0.15	0.03	123265.00	0.00	0.00
0101	158012	10.1	1.00	143412.00	0.00	0.00	147470.00	2.83	0.28
0101	116390	0.30	0.05	116039.00	0.00	0.00	116206.00	0.14	0.02
0101	132451	0.92	0.20	131392.00	0.12	0.03	131240.00	0.00	0.00
0101	133568	5.97	0.64	129529.00	2.76	0.30	126046.00	0.00	0.00
0101	149043	3.37	0.53	148551.00	3.03	0.47	144184.00	0.00	0.00
0101	221189	7.68	1.00	212560.00	3.48	0.45	205409.00	0.00	0.00
0101	132242	0.65	0.15	131393.00	0.00	0.00	131608.00	0.16	0.04
0101	175468	1.16	0.06	178624.00	2.98	0.15	173496.00	0.02	0.00
0101	173492	3.68	1.00	167342.00	0.00	0.00	167343.00	0.00	0.00

Table B.6: Computational results for Knap, 2bin, GAPS algorithms for n=50

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0110	24897	1.05	0.75	24677.00	0.15	0.11	24639.00	0.00	0.00
0110	25829	0.02	0.01	25823.00	0.00	0.00	25823.00	0.00	0.00
0110	25134	0.49	0.32	25012.00	0.00	0.00	25012.00	0.00	0.00
0110	24779	1.21	0.62	24495.00	0.05	0.03	24482.00	0.00	0.00
0110	24820	1.10	1.00	24550.00	0.00	0.00	24550.00	0.00	0.00
0110	25151	0.69	1.00	25005.00	0.10	0.15	24979.00	0.00	0.00
0110	26951	1.05	1.00	26671.00	0.00	0.00	26671.00	0.00	0.00
0110	24866	0.46	0.26	24766.00	0.06	0.03	24751.00	0.00	0.00
0110	25308	0.56	1.00	25168.00	0.00	0.00	25168.00	0.00	0.00
0110	25815	1.36	1.00	25469.00	0.00	0.00	25469.00	0.00	0.00
0111	36649	1.90	0.18	36035.00	0.19	0.02	35966.00	0.00	0.00
0111	40305	0.74	0.17	40024.00	0.04	0.01	40007.00	0.00	0.00
0111	36574	1.56	0.49	36206.00	0.54	0.17	36011.00	0.00	0.00
0111	36979	0.81	0.20	36939.00	0.70	0.17	36681.00	0.00	0.00
0111	37886	0.72	1.00	37616.00	0.00	0.00	37616.00	0.00	0.00
0111	37911	1.65	0.26	37331.00	0.09	0.01	37297.00	0.00	0.00
0111	45068	2.03	0.35	44325.00	0.35	0.06	44172.00	0.00	0.00
0111	37016	0.31	0.04	39616.00	7.36	1.00	36901.00	0.00	0.00
0111	39516	0.36	1.00	39376.00	0.00	0.00	39376.00	0.00	0.00
0111	39735	0.27	0.04	39629.00	0.00	0.00	39629.00	0.00	0.00
1000	12550	1.17	0.73	12419.00	0.11	0.07	12405.00	0.00	0.00
1000	16219	2.43	0.48	15872.00	0.24	0.05	15834.00	0.00	0.00
1000	13670	2.01	0.63	13409.00	0.07	0.02	13400.00	0.00	0.00
1000	12533	1.02	0.40	12499.00	0.74	0.29	12407.00	0.00	0.00
1000	12629	1.29	0.88	12509.00	0.33	0.22	12468.00	0.00	0.00
1000	13991	0.52	0.07	13940.00	0.15	0.02	13919.00	0.00	0.00
1000	20058	3.21	0.31	19500.00	0.33	0.03	19435.00	0.00	0.00
1000	13257	2.15	1.00	13059.00	0.62	0.29	12978.00	0.00	0.00
1000	16312	3.31	1.00	15965.00	1.11	0.34	15789.00	0.00	0.00
1000	15415	3.32	0.43	15140.00	1.48	0.19	14919.00	0.00	0.00
1001	43784	4.88	1.00	42150.00	0.97	0.20	41745.00	0.00	0.00
1001	57639	5.24	1.00	55220.00	0.82	0.16	54770.00	0.00	0.00
1001	45755	3.03	0.77	44506.00	0.22	0.06	44410.00	0.00	0.00
1001	43846	3.10	0.88	42743.00	0.51	0.14	42527.00	0.00	0.00
1001	43401	2.67	0.52	42715.00	1.04	0.20	42274.00	0.00	0.00
1001	54291	2.32	0.34	53155.00	0.18	0.03	53062.00	0.00	0.00
1001	82758	7.36	0.65	78647.00	2.03	0.18	77081.00	0.00	0.00
1001	46797	4.42	1.00	45576.00	1.70	0.38	44814.00	0.00	0.00
1001	73774	8.99	0.87	68207.00	0.77	0.07	67686.00	0.00	0.00
1001	61655	5.92	0.74	59614.00	2.42	0.30	58207.00	0.00	0.00
1010	9409	0.19	1.00	9401.00	0.11	0.56	9391.00	0.00	0.00
1010	10394	0.40	0.05	10353.00	0.00	0.00	10353.00	0.00	0.00
1010	10476	0.98	1.00	10382.00	0.08	0.08	10374.00	0.00	0.00
1010	8923	0.19	0.23	8911.00	0.06	0.07	8906.00	0.00	0.00
1010	8595	0.29	0.60	8589.00	0.22	0.45	8570.00	0.00	0.00
1010	9918	0.00	0.00	9918.00	0.00	0.00	9918.00	0.00	0.00
1010	11187	0.09	0.08	11177.00	0.00	0.00	11177.00	0.00	0.00
1010	9391	0.19	0.08	9384.00	0.12	0.05	9373.00	0.00	0.00
1010	10138	0.22	1.00	10116.00	0.00	0.00	10116.00	0.00	0.00
1010	10134	0.06	1.00	10128.00	0.00	0.00	10128.00	0.00	0.00

Table B.7: Computational results for Knap, 2Bin, GAPS algorithms for n=50 (continued)

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
1011	12629	0.14	0.01	12625.00	0.11	0.01	12611.00	0.00	0.00
1011	14972	0.27	0.01	14931.00	0.00	0.00	14931.00	0.00	0.00
1011	14196	0.72	0.19	14148.00	0.38	0.10	14094.00	0.00	0.00
1011	12037	1.60	0.06	11852.00	0.04	0.00	11847.00	0.00	0.00
1011	11731	1.92	0.93	11530.00	0.17	0.08	11510.00	0.00	0.00
1011	13793	0.00	0.00	13793.00	0.00	0.00	13793.00	0.00	0.00
1011	17248	0.06	0.01	17238.00	0.00	0.00	17238.00	0.00	0.00
1011	12831	0.14	0.00	12924.00	0.87	0.02	12813.00	0.00	0.00
1011	14848	1.18	0.20	14696.00	0.14	0.02	14675.00	0.00	0.00
1011	14554	0.04	0.00	14548.00	0.00	0.00	14548.00	0.00	0.00
1100	25246	1.13	0.47	25091.00	0.50	0.21	24965.00	0.00	0.00
1100	31327	2.13	0.40	30674.00	0.00	0.00	31065.00	1.27	0.24
1100	27147	2.19	0.59	26848.00	1.07	0.29	26565.00	0.00	0.00
1100	24769	1.02	0.32	24771.00	1.02	0.32	24520.00	0.00	0.00
1100	25090	2.70	1.00	24690.00	1.06	0.39	24431.00	0.00	0.00
1100	28388	2.54	0.21	28165.00	1.73	0.15	27685.00	0.00	0.00
1100	39778	6.76	0.53	38178.00	2.46	0.19	37261.00	0.00	0.00
1100	25562	1.26	0.22	25284.00	0.16	0.03	25244.00	0.00	0.00
1100	31366	1.41	0.36	31220.00	0.94	0.24	30930.00	0.00	0.00
1100	31021	3.69	0.38	30643.00	2.43	0.25	29917.00	0.00	0.00
1101	88276	3.70	1.00	85341.00	0.25	0.07	85124.00	0.00	0.00
1101	110077	4.46	0.59	105374.00	0.00	0.00	108058.00	2.55	0.34
1101	90309	5.47	0.69	89299.00	4.29	0.54	85624.00	0.00	0.00
1101	85645	2.08	0.40	83900.00	0.00	0.00	86220.00	2.77	0.53
1101	88675	4.78	0.72	84630.00	0.00	0.00	85180.00	0.65	0.10
1101	111328	4.87	0.38	106373.00	0.21	0.02	106154.00	0.00	0.00
1101	168514	11.6	0.95	156655.00	3.83	0.31	150882.00	0.00	0.00
1101	89462	2.83	0.31	87245.00	0.28	0.03	87000.00	0.00	0.00
1101	136990	4.90	0.44	136404.00	4.45	0.40	130590.00	0.00	0.00
1101	123421	6.97	0.60	115379.00	0.00	0.00	116017.00	0.55	0.05
1110	18555	0.00	0.00	18555.00	0.00	0.00	18555.00	0.00	0.00
1110	20378	0.09	0.01	20360.00	0.00	0.00	20360.00	0.00	0.00
1110	20510	0.21	1.00	20467.00	0.00	0.00	20467.00	0.00	0.00
1110	17658	0.50	0.13	17571.00	0.00	0.00	17571.00	0.00	0.00
1110	16645	0.00	0.00	16645.00	0.00	0.00	16645.00	0.00	0.00
1110	19756	0.21	1.00	19715.00	0.00	0.00	19715.00	0.00	0.00
1110	21888	0.04	0.03	21880.00	0.00	0.00	21880.00	0.00	0.00
1110	18232	0.49	0.05	18143.00	0.00	0.00	18143.00	0.00	0.00
1110	20006	0.21	1.00	19965.00	0.00	0.00	19965.00	0.00	0.00
1110	20136	0.04	0.05	20127.00	0.00	0.00	20127.00	0.00	0.00
1111	24825	0.00	0.00	24825.00	0.00	0.00	24825.00	0.00	0.00
1111	28928	0.06	0.01	28910.00	0.00	0.00	28910.00	0.00	0.00
1111	27847	1.05	0.05	27557.00	0.00	0.00	27804.00	0.90	0.04
1111	23710	0.37	0.13	23623.00	0.00	0.00	23623.00	0.00	0.00
1111	22315	0.00	0.00	22315.00	0.00	0.00	22315.00	0.00	0.00
1111	27731	0.93	0.24	27475.00	0.00	0.00	27690.00	0.78	0.20
1111	33552	0.02	0.00	33544.00	0.00	0.00	33544.00	0.00	0.00
1111	24622	1.84	0.08	24178.00	0.00	0.00	24178.00	0.00	0.00
1111	29078	0.14	0.03	29037.00	0.00	0.00	29037.00	0.00	0.00
1111	28536	0.03	0.00	28527.00	0.00	0.00	28527.00	0.00	0.00
AVG	41025.33	1.84	0.48	40196.91	0.53	0.10	39905.69	0.11	0.02

Table B.8: Computational results for Knap, 2Bin, GAPS algorithms for n=50 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0000	0	0.01	0.02	1.58	1.4	4.92	11.38
0000	0	0	0.01	1.71	1.33	5.16	11.39
0000	0	0.01	0.01	1.9	1.4	4.92	17.46
0000	0	0	0.01	1.35	1.3	5.09	10.36
0000	0.01	0	0.01	1.52	1.73	4.91	11.48
0000	0	0.01	0.01	1.61	1.55	4.98	14.95
0000	0	0.01	0.01	1.49	1.89	4.95	11.39
0000	0	0.01	0.01	1.55	1.23	4.89	11.88
0000	0	0.01	0.01	1.87	2.32	5.1	15.11
0000	0.01	0	0.02	2.01	1.71	4.91	17.67
0001	0	0	0	1.31	1.44	4.99	11.41
0001	0	0.01	0	1.39	1.31	4.97	11.7
0001	0	0.01	0.02	1.7	1.33	5.05	17.75
0001	0	0	0	1.31	1.26	4.86	10.51
0001	0	0	0.01	1.36	1.6	4.71	11.24
0001	0.01	0.01	0.02	1.61	1.56	4.99	15.14
0001	0	0	0	1.4	1.92	4.9	11.23
0001	0	0	0.01	1.56	1.29	4.97	11.92
0001	0	0.01	0.02	1.83	2.3	4.81	14.95
0001	0	0.01	0.02	2.02	1.69	4.69	17.55
0010	0.01	0.01	0.01	0.21	0.72	5.76	3.04
0010	0	0	0	0.21	0.33	5.3	3.23
0010	0	0	0	0.26	0.27	5.17	3.12
0010	0	0.01	0	0.29	0.3	5	3.17
0010	0	0	0	0.18	0.26	5.44	3.16
0010	0	0	0	0.21	0.31	7.8	3.13
0010	0	0	0	0.23	0.34	5.38	3.15
0010	0	0	0	0.23	0.28	5.44	3.16
0010	0	0	0	0.26	0.51	5.38	2.9
0010	0.01	0.01	0	0.24	0.43	18.36	3.06
0011	0	0	0	0.26	0.75	5.64	3.19
0011	0	0.01	0.01	0.24	0.32	5.21	3.25
0011	0	0	0	0.22	0.29	5.03	3.21
0011	0	0	0	0.22	0.31	5.05	3.24
0011	0	0	0	0.24	0.33	5.37	3.2
0011	0.01	0	0.01	0.2	0.33	5.25	3.21
0011	0	0.01	0.01	0.21	0.33	6.07	3.34
0011	0	0	0	0.23	0.29	5.05	3.3
0011	0	0	0	0.24	0.57	5.82	3.03
0011	0	0	0	0.27	0.39	17.88	3.42
0100	0	0.01	0	0.36	1.28	4.94	3
0100	0	0	0.01	0.25	1.36	4.9	2.97
0100	0.01	0	0	0.35	1.42	4.93	3.1
0100	0	0	0	0.21	1.44	4.95	3.17
0100	0	0	0	0.28	1.72	4.99	3.01
0100	0	0.01	0.01	0.34	2.08	4.91	3.32
0100	0	0	0	0.29	2.12	4.75	3.31
0100	0	0	0	0.35	1.43	5.17	3.17
0100	0.01	0	0	0.4	2.35	4.64	3.23
0100	0	0	0	0.31	1.9	4.7	3.2
0101	0	0	0.01	0.33	1.33	4.91	3.61
0101	0	0.01	0	0.4	1.48	5	3.33
0101	0	0	0	0.48	1.33	4.89	3.42
0101	0	0	0	0.42	1.45	5.06	3.51
0101	0	0	0	0.35	1.73	5.21	3.48
0101	0	0	0.01	0.41	1.98	4.74	3.76
0101	0	0	0	0.44	2.12	4.89	3.83
0101	0	0.01	0	0.39	1.39	4.87	3.47
0101	0	0	0	0.43	2.42	4.59	3.75
0101	0	0	0	0.38	1.95	4.69	3.72

Table B.9: Computation times for n=50

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0110	0	0.01	0	0.25	0.29	5.8	3.11
0110	0	0.01	0	0.21	0.33	4.82	3.07
0110	0.01	0	0	0.25	0.58	5.62	3.08
0110	0	0	0	0.23	0.48	6.28	3.09
0110	0	0	0	0.24	0.62	5.43	3.12
0110	0	0	0.01	0.28	0.42	5.97	3.07
0110	0	0	0	0.29	0.86	8.25	2.96
0110	0	0.01	0	0.21	0.28	6.2	3.05
0110	0.01	0	0	0.27	0.3	2320.12	2.94
0110	0	0	0	1.31	0.45	6.23	3.09
0111	0	0.01	0	0.27	0.33	6.58	3.08
0111	0	0	0	0.26	0.38	5.64	3.23
0111	0.01	0	0	0.27	0.58	6.69	3.12
0111	0	0	0	0.29	0.49	6.75	3.19
0111	0	0	0	0.26	0.66	5.72	3.17
0111	0	0	0	0.24	0.36	6.53	3.09
0111	0	0.01	0.01	0.32	0.8	8.41	2.99
0111	0	0	0	0.28	0.32	6.17	3.06
0111	0.01	0	0	0.23	0.36	2126.06	2.94
0111	0	0	0	0.23	0.43	6.28	3.11
1000	0	0	0.01	0.27	0.98	4.78	2.83
1000	0	0	0.01	0.32	0.99	5.42	2.79
1000	0	0.01	0	0.32	1.06	5.18	2.82
1000	0	0.01	0	0.29	0.96	5.06	2.85
1000	0	0	0	0.29	0.96	6.38	2.82
1000	0	0	0	0.27	1.21	5.25	2.99
1000	0	0	0.01	0.2	1.39	4.99	2.92
1000	0	0	0.01	0.2	0.95	5.16	2.87
1000	0	0.01	0	0.21	1.69	5.03	3.16
1000	0	0	0	0.34	0.06	5.2	2.9
1001	0	0	0	0.27	0.98	5.42	2.89
1001	0	0	0	0.27	0.99	4.84	2.91
1001	0	0	0	0.25	1.06	5.05	2.92
1001	0	0	0	0.21	0.96	4.88	2.85
1001	0	0	0	0.23	0.96	5.11	2.89
1001	0	0.01	0	0.25	1.21	4.57	2.76
1001	0.01	0	0.01	0.28	1.39	4.79	3.02
1001	0	0	0	0.27	0.95	4.75	2.9
1001	0	0	0	0.28	1.69	4.62	3.11
1001	0	0	0	0.34	0.06	4.74	2.82
1010	0	0.01	0	0.22	0.22	5.97	3.84
1010	0	0	0.01	0.23	0.28	6.47	3.92
1010	0	0	0	0.34	0.22	5.25	3.8
1010	0	0	0	0.29	0.21	8.32	4.03
1010	0	0	0	0.28	0.24	6.06	4.29
1010	0.01	0.01	0	0.28	0.2	6.61	3.83
1010	0	0.01	0.01	0.31	0.25	7.15	3.79
1010	0	0	0.01	0.31	0.36	6.07	4.05
1010	0	0	0	0.32	0.24	6.92	3.77
1010	0	0	0	0.2	0.2	8.3	3.76

Table B.10: Computation times for n=50 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
1011	0.01	0	0	0.29	0.3	5.7	3.47
1011	0	0	0.01	0.17	0.25	6.08	3.48
1011	0	0.01	0	0.29	0.19	5.44	3.2
1011	0.01	0	0	0.28	0.19	8.28	3.16
1011	0.01	0	0	0.3	0.2	5.65	3.75
1011	0	0	0	0.35	0.18	6.62	3.22
1011	0	0	0	0.21	0.22	6.62	3.24
1011	0	0	0.01	0.29	0.37	5.77	3.5
1011	0	0	0	0.24	0.24	6.64	3.22
1011	0	0	0	0.26	0.23	8.46	3.18
1100	0	0.01	0.01	0.3	1.04	5.72	3.28
1100	0	0	0	0.29	0.74	9.09	3.47
1100	0	0	0	0.34	0.88	10.4	3.24
1100	0	0	0	0.21	0.8	6.01	3.5
1100	0	0.01	0	0.29	0.88	24.88	3.58
1100	0	0.01	0	0.33	1.18	5.55	3.14
1100	0	0	0	0.27	1.19	11.97	3.11
1100	0	0	0	0.26	0.95	5.06	3.44
1100	0	0	0	0.32	1.26	6.65	3.19
1100	0	0	0	0.29	0.09	7.57	3.18
1101	0	0.01	0	0.25	1.03	5.41	2.73
1101	0	0	0	0.23	0.83	5.82	2.7
1101	0	0.01	0	0.28	0.89	5.47	2.79
1101	0	0	0	0.27	0.81	6.24	2.74
1101	0	0	0.01	0.29	0.93	5.7	2.84
1101	0	0	0	0.3	1.15	4.98	2.76
1101	0	0	0	0.24	1.32	5.21	2.85
1101	0	0	0	0.28	1.02	5.53	2.79
1101	0	0.01	0	0.39	1.35	5.01	3.05
1101	0	0.01	0.01	0.22	1.1	5.16	2.79
1110	0	0	0.01	1.43	0.39	5.72	10.73
1110	0	0.01	0	1.22	0.35	9.09	10.96
1110	0	0	0.01	1.31	0.24	10.4	10.29
1110	0	0.01	0.01	1.32	0.23	6.01	10.94
1110	0	0.01	0	1.31	0.21	24.88	10.65
1110	0	0	0.01	1.35	0.47	5.55	10.86
1110	0	0.01	0.01	1.54	0.29	11.97	11.05
1110	0	0.01	0	1.41	0.33	5.06	10.56
1110	0.01	0	0.01	1.43	0.23	6.65	10.7
1110	0.01	0	0.01	1.37	0.26	7.57	10.95
1111	0.01	0	0	0.3	0.39	5.5	3.39
1111	0	0	0.01	0.25	0.35	9.48	3.46
1111	0.01	0	0	0.27	0.24	10.86	3.21
1111	0	0.01	0	0.26	0.27	5.67	3.57
1111	0	0.01	0	0.27	0.21	25.18	3.74
1111	0	0	0	0.33	0.45	5.51	3.14
1111	0	0	0.01	0.25	0.3	11.83	3.2
1111	0	0	0.01	0.28	0.3	4.98	3.48
1111	0.01	0	0	0.25	0.22	6.86	3.11
1111	0	0.01	0	0.24	0.24	7.61	3.22
AVG	0.001375	0.003	0.003438	0.518438	0.8345	34.11788	4.94875

Table B.11: Computation times for n=50 (continued)

Appendix C

Computational Results for 100 Jobs

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	101788.10	3.02	0.51	99340.30	0.24	0.27	99299.70	0.22	0.14	101868.70	3.08	0.72
0001	503556.30	6.63	0.59	478657.70	0.55	0.23	478617.10	0.55	0.10	492204.20	3.90	0.60
0010	48404.40	0.00	0.02	48586.50	0.38	0.51	48528.20	0.26	0.34	48404.40	0.00	0.02
0011	63487.70	0.11	0.02	66323.10	4.57	0.95	65069.40	2.62	0.63	63487.70	0.11	0.02
0100	205259.80	2.14	0.56	201814.70	0.20	0.18	201732.70	0.17	0.06	205416.90	2.20	0.77
0101	1014380.1	4.58	0.60	980220.90	0.33	0.17	980138.90	0.33	0.05	992411.10	2.09	0.50
0110	97393.40	0.02	0.04	97769.70	0.41	0.44	97743.00	0.38	0.48	97393.40	0.02	0.04
0111	127923.30	0.28	0.07	133620.20	4.75	0.96	130808.70	2.54	0.58	127923.30	0.28	0.07
1000	76010.00	3.09	0.60	78706.40	6.26	0.88	77674.10	5.02	0.64	75711.50	2.72	0.53
1001	353792.80	7.11	0.95	347844.40	4.83	0.63	342504.20	3.41	0.45	339667.40	2.75	0.34
1010	38654.40	0.00	0.01	38711.90	0.15	0.44	38654.40	0.00	0.01	38654.40	0.00	0.01
1011	48134.80	0.12	0.04	49063.50	2.10	0.19	49553.90	3.12	0.81	48134.80	0.12	0.04
1100	150899.20	3.50	0.63	154771.20	5.74	0.82	152702.50	4.53	0.68	150270.00	3.17	0.60
1101	709895.00	7.78	0.95	684963.60	3.81	0.48	674933.90	2.63	0.38	683428.40	3.75	0.39
1110	76190.40	0.01	0.00	76350.30	0.21	0.61	76314.80	0.17	0.10	76190.40	0.01	0.00
1111	94564.30	0.18	0.04	98025.60	3.92	0.38	97036.80	2.84	0.73	94564.30	0.18	0.04
AVG	231895.88	2.41	0.35	227173.13	2.40	0.51	225707.02	1.80	0.39	227233.18	1.52	0.29

Table C.1: Averages taken over 10 replication for n=100

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	100216.50	1.27	0.45	101088.20	2.18	0.32	99129.30	0.01	0.01
0001	481490.50	1.41	0.33	495975.80	4.48	0.35	476992.50	0.00	0.00
0010	48511.90	0.23	0.61	48404.40	0.00	0.02	48402.40	0.00	0.00
0011	63497.20	0.13	0.03	63444.40	0.04	0.01	63459.90	0.07	0.01
0100	204057.30	1.51	0.59	204168.70	1.50	0.36	201434.00	0.00	0.00
0101	988671.50	1.51	0.38	1004074.4	3.08	0.37	978071.20	0.00	0.00
0110	97710.60	0.35	0.67	97393.40	0.02	0.04	97369.70	0.00	0.00
0111	128048.40	0.38	0.09	127678.70	0.09	0.02	127655.40	0.07	0.02
1000	74478.80	1.20	0.33	74505.60	1.17	0.24	74146.30	0.62	0.07
1001	337255.80	2.45	0.34	337436.40	2.29	0.30	333699.60	0.98	0.13
1010	38714.70	0.16	0.79	38653.30	0.00	0.00	38653.50	0.00	0.00
1011	48218.50	0.29	0.18	48081.80	0.00	0.00	48079.90	0.00	0.00
1100	147894.00	1.84	0.45	147856.30	1.57	0.29	146200.40	0.44	0.05
1101	675088.30	3.51	0.53	679624.00	3.48	0.41	665907.90	1.14	0.09
1110	76296.70	0.15	0.56	76190.30	0.01	0.00	76186.00	0.00	0.00
1111	94595.20	0.21	0.06	94452.90	0.06	0.01	94420.60	0.02	0.01
AVG	225296.62	1.04	0.40	227439.29	1.25	0.17	223113.04	0.21	0.02

Table C.2: Averages taken over 10 replication for n=100 (continued)

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0000	94982.00	5.19	1	90560.00	0.291	0.06	90560.00	0.29	0.06	94982.00	5.19	1
0000	97089.00	0	0	97277.00	0.194	0.11	97277.00	0.19	0.11	97277.00	0.19	0.1
0000	141702.00	0	0	141927.00	0.159	0.67	141702.00	0	0	142039.00	0.24	1
0000	77810.00	0.58	0.56	78156.00	1.024	1	78156.00	1.02	1	77810.00	0.58	0.6
0000	89302.00	0	0	89503.00	0.225	0.15	89503.00	0.23	0.15	89302.00	0	0
0000	95182.00	8.49	1	87767.00	0.042	0	87767.00	0.04	0	95182.00	8.49	1
0000	75894.00	4.53	1	72882.00	0.384	0.08	72882.00	0.38	0.08	75894.00	4.53	1
0000	92364.00	1.18	0.51	91284.00	0	0	91284.00	0	0	92364.00	1.18	0.5
0000	104502.00	10.2	1	94812.00	0	0	94812.00	0	0	104502.00	10.2	1
0000	149054.00	0	0	149235.00	0.121	0.64	149054.00	0	0	149335.00	0.19	1
0001	405578.00	10.6	1	370268.00	0.974	0.09	370268.00	0.97	0.09	383847.00	4.68	0.4
0001	428324.00	0	0	428512.00	0.044	0.11	428512.00	0.04	0.11	428512.00	0.04	0.1
0001	809952.00	0	0	810177.00	0.028	0.67	809952.00	0	0	810289.00	0.04	1
0001	296905.00	3.69	0.93	290871.00	1.579	0.4	290871.00	1.58	0.4	297705.00	3.97	1
0001	384208.00	0	0	384409.00	0.052	0.15	384409.00	0.05	0.15	384488.00	0.07	0.2
0001	522910.00	16.2	1	449975.00	0.008	0	449975.00	0.01	0	487938.00	8.45	0.5
0001	310854.00	13.9	1	280602.00	2.831	0.2	280602.00	2.83	0.2	298995.00	9.57	0.7
0001	431514.00	3.45	1	417134.00	0	0	417134.00	0	0	424377.00	1.74	0.5
0001	583464.00	18.4	1	492594.00	0	0	492594.00	0	0	543756.00	10.4	0.6
0001	861854.00	0	0	862035.00	0.021	0.64	861854.00	0	0	862135.00	0.03	1
0010	49790.00	0	0	50330.00	1.085	1	50004.00	0.43	0.4	49790.00	0	0
0010	48417.00	0	0	48718.00	0.622	1	48417.00	0	0	48417.00	0	0
0010	49970.00	0	0	49971.00	0.002	0.01	49970.00	0	0	49970.00	0	0
0010	48137.00	0	0	48680.00	1.128	1	48480.00	0.71	0.63	48137.00	0	0
0010	48782.00	0	0	48861.00	0.162	0.26	49086.00	0.62	1	48782.00	0	0
0010	47030.00	0	0	47030.00	0	0	47030.00	0	0	47030.00	0	0
0010	48099.00	0	0	48449.00	0.728	1	48179.00	0.17	0.23	48099.00	0	0
0010	48462.00	0	0	48462.00	0	0	48764.00	0.62	1	48462.00	0	0
0010	47817.00	0	0	47829.00	0.025	0.63	47817.00	0	0	47817.00	0	0
0010	47540.00	0.04	0.22	47535.00	0.032	0.17	47535.00	0.03	0.17	47540.00	0.04	0.2
0011	66638.00	0.13	0.02	70826.00	6.42	1	67658.00	1.66	0.26	66638.00	0.13	0
0011	65312.00	0.32	0.06	68667.00	5.473	1	67474.00	3.64	0.67	65312.00	0.32	0.1
0011	66440.00	0	0	67957.00	2.283	1	67358.00	1.38	0.61	66440.00	0	0
0011	62347.00	0	0	63711.00	2.188	0.98	63741.00	2.24	1	62347.00	0	0
0011	64100.00	0.11	0.02	68086.00	6.333	1	65412.00	2.16	0.34	64100.00	0.11	0
0011	59910.00	0.18	0.04	62224.00	4.047	1	61906.00	3.51	0.87	59910.00	0.18	0
0011	60699.00	0.24	0.05	63614.00	5.052	1	61258.00	1.16	0.23	60699.00	0.24	0
0011	63225.00	0	0	68007.00	7.563	1	64627.00	2.22	0.29	63225.00	0	0
0011	61962.00	0	0	63279.00	2.125	0.54	64404.00	3.94	1	61962.00	0	0
0011	64244.00	0.17	0.04	66860.00	4.252	1	66856.00	4.25	1	64244.00	0.17	0
0100	190092.00	5.23	1	182637.00	1.106	0.21	182637.00	1.11	0.21	190092.00	5.23	1
0100	200447.00	0	0	200447.00	0	0	200447.00	0	0	200608.00	0.08	0.1
0100	283984.00	0	0	284435.00	0.159	0.61	283984.00	0	0	284721.00	0.26	1
0100	158046.00	1.24	0.88	157022.00	0.584	0.41	157022.00	0.58	0.41	158046.00	1.24	0.9
0100	181513.00	0	0	181513.00	0	0	181513.00	0	0	181513.00	0	0
0100	189148.00	5.57	1	179164.00	0	0	179164.00	0	0	189148.00	5.57	1
0100	150595.00	3.26	1	145916.00	0.053	0.02	145916.00	0.05	0.02	150595.00	3.26	1
0100	186014.00	1.81	0.76	182708.00	0	0	182708.00	0	0	186014.00	1.81	0.8
0100	213655.00	4.31	1	204832.00	0	0	204832.00	0	0	213655.00	4.31	1
0100	299104.00	0	0	299473.00	0.123	0.55	299104.00	0	0	299777.00	0.23	1
0101	809293.00	9.53	1	747724.00	1.194	0.13	747724.00	1.19	0.13	765956.00	3.66	0.4
0101	901667.00	0	0	901667.00	0	0	901667.00	0	0	901828.00	0.02	0.1
0101	1620484.0	0	0	1620935.0	0.028	0.61	1620484.00	0	0	1621221.0	0.05	1
0101	607441.00	4.63	1	588788.00	1.415	0.31	588788.00	1.42	0.31	607487.00	4.64	1
0101	810349.00	0	0	810349.00	0	0	810349.00	0	0	810349.00	0	0
0101	1024506.0	10.6	1	926214.00	0.004	0	926214.00	0	0	956904.00	3.32	0.3
0101	607772.00	9.48	1	558749.00	0.648	0.07	558749.00	0.65	0.07	593091.00	6.83	0.7
0101	869900.00	4.25	1	834408.00	0	0	834408.00	0	0	853596.00	2.3	0.5
0101	1167685.0	7.29	1	1088302.0	0	0	1088302.00	0	0	1088302.0	0	0
0101	1724704.0	0	0	1725073.0	0.021	0.55	1724704.00	0	0	1725377.0	0.04	1

Table C.3: Computational results for SPT, FFD, MFFD, EGI for n=100

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0110	99708.00	0	0	100728.00	1.023	1	100241.00	0.53	0.52	99708.00	0	0
0110	97103.00	0.03	0.05	97727.00	0.676	1	97674.00	0.62	0.92	97103.00	0.03	0
0110	100634.00	0.06	0.12	100634.00	0.06	0.12	100634.00	0.06	0.12	100634.00	0.06	0.1
0110	96936.00	0.06	0.05	98106.00	1.267	1	97761.00	0.91	0.72	96936.00	0.06	0
0110	97624.00	0.07	0.11	97686.00	0.136	0.2	98208.00	0.67	1	97624.00	0.07	0.1
0110	94742.00	0	0	94742.00	0	0	94742.00	0	0	94742.00	0	0
0110	96835.00	0	0	97722.00	0.916	1	97244.00	0.42	0.46	96835.00	0	0
0110	97562.00	0	0	97562.00	0	0	98136.00	0.59	1	97562.00	0	0
0110	96630.00	0.02	0.04	96630.00	0.018	0.04	96630.00	0.02	0.04	96630.00	0.02	0
0110	96160.00	0	0	96160.00	0	0	96160.00	0	0	96160.00	0	0
0111	133296.00	0	0	141555.00	6.196	1	135868.00	1.93	0.31	133296.00	0	0
0111	130893.00	0.48	0.09	137517.00	5.564	1	133834.00	2.74	0.49	130893.00	0.48	0.1
0111	134384.00	0.44	0.15	136286.00	1.863	0.63	137747.00	2.96	1	134384.00	0.44	0.1
0111	125836.00	0.4	0.13	129175.00	3.068	1	126661.00	1.06	0.35	125836.00	0.4	0.1
0111	128482.00	0.48	0.07	136294.00	6.594	1	130763.00	2.27	0.34	128482.00	0.48	0.1
0111	120387.00	0.16	0.04	124889.00	3.904	1	124278.00	3.4	0.87	120387.00	0.16	0
0111	122140.00	0.03	0.01	128460.00	5.209	1	124200.00	1.72	0.33	122140.00	0.03	0
0111	127886.00	0.19	0.03	136429.00	6.887	1	130621.00	2.34	0.34	127886.00	0.19	0
0111	125785.00	0.37	0.09	130607.00	4.217	1	129312.00	3.18	0.75	125785.00	0.37	0.1
0111	130144.00	0.24	0.06	134990.00	3.975	1	134803.00	3.83	0.96	130144.00	0.24	0.1
1000	76727.00	3.05	0.39	80220.00	7.741	1	79720.00	7.07	0.91	76727.00	3.05	0.4
1000	75425.00	3.53	1	74771.00	2.637	0.75	73589.00	1.01	0.29	73809.00	1.32	0.4
1000	107541.00	3.91	0.3	116983.00	13.03	1	111656.00	7.89	0.61	107571.00	3.94	0.3
1000	61883.00	2.22	0.9	62038.00	2.478	1	61411.00	1.44	0.58	61883.00	2.22	0.9
1000	72466.00	2.84	0.99	72043.00	2.242	0.78	70795.00	0.47	0.16	72466.00	2.84	1
1000	64927.00	1.28	0.21	67954.00	6.006	1	67732.00	5.66	0.94	64927.00	1.28	0.2
1000	59065.00	1.9	0.52	60101.00	3.69	1	59857.00	3.27	0.89	59065.00	1.9	0.5
1000	73597.00	4.2	1	71451.00	1.16	0.28	70632.00	0	0	73597.00	4.2	1
1000	76326.00	3.32	0.39	80111.00	8.441	1	80043.00	8.35	0.99	76326.00	3.32	0.4
1000	92143.00	4.67	0.31	101392.00	15.18	1	101306.00	15.1	0.99	90744.00	3.08	0.2
1001	318287.00	6.85	0.9	320460.00	7.582	1	318512.00	6.93	0.91	302004.00	1.39	0.2
1001	328808.00	8.59	1	312386.00	3.171	0.37	306605.00	1.26	0.15	312029.00	3.05	0.4
1001	570907.00	7.62	0.93	574046.00	8.209	1	545608.00	2.85	0.35	549041.00	3.5	0.4
1001	230591.00	5.84	1	226865.00	4.133	0.71	224435.00	3.02	0.52	224677.00	3.13	0.5
1001	317074.00	8.69	1	298338.00	2.271	0.26	291883.00	0.06	0.01	297388.00	1.95	0.2
1001	331729.00	5.08	1	323242.00	2.388	0.47	322450.00	2.14	0.42	315702.00	0	0
1001	229552.00	4.4	1	225657.00	2.628	0.6	223560.00	1.67	0.38	223751.00	1.76	0.4
1001	332849.00	9.85	1	307319.00	1.425	0.14	303000.00	0	0	317884.00	4.91	0.5
1001	389004.00	6.58	1	383799.00	5.157	0.78	383151.00	4.98	0.76	369042.00	1.11	0.2
1001	489127.00	7.54	0.67	506332.00	11.33	1	505838.00	11.2	0.99	485156.00	6.67	0.6
1010	41129.00	0	0	41129.00	0	0	41129.00	0	0	41129.00	0	0
1010	37583.00	0	0	37783.00	0.532	1	37583.00	0	0	37583.00	0	0
1010	43767.00	0	0	43767.00	0	0	43767.00	0	0	43767.00	0	0
1010	38333.00	0	0	38429.00	0.25	1	38333.00	0	0	38333.00	0	0
1010	38922.00	0	0	38966.00	0.113	0.42	38922.00	0	0	38922.00	0	0
1010	35569.00	0	0	35630.00	0.171	1	35569.00	0	0	35569.00	0	0
1010	39115.00	0.03	0.06	39289.00	0.473	1	39115.00	0.03	0.06	39115.00	0.03	0.1
1010	39069.00	0	0	39069.00	0	0	39069.00	0	0	39069.00	0	0
1010	37914.00	0	0	37914.00	0	0	37914.00	0	0	37914.00	0	0
1010	35143.00	0	0	35143.00	0	0	35143.00	0	0	35143.00	0	0

Table C.4: Computational results for SPT, FFD, MFFD, EGI for n=100 (continued)

	SPT			FFD			MFFD			EGI		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
1011	52349.00	0	0	52349.00	0	0	54105.00	3.35	1	52349.00	0	0
1011	47219.00	0	0	47857.00	1.351	0.37	48921.00	3.6	1	47219.00	0	0
1011	55991.00	0	0	55991.00	0	0	55991.00	0	0	55991.00	0	0
1011	46901.00	0	0	55296.00	17.9	1	48113.00	2.58	0.14	46901.00	0	0
1011	48722.00	0.4	0.1	48570.00	0.082	0.02	50400.00	3.85	1	48722.00	0.4	0.1
1011	43042.00	0	0	43103.00	0.142	0.04	44553.00	3.51	1	43042.00	0	0
1011	47152.00	0.39	0.13	47497.00	1.122	0.39	48338.00	2.91	1	47152.00	0.39	0.1
1011	48657.00	0	0	48657.00	0	0	50869.00	4.55	1	48657.00	0	0
1011	46788.00	0	0	46788.00	0	0	48497.00	3.65	1	46788.00	0	0
1011	44527.00	0.4	0.13	44527.00	0.399	0.13	45752.00	3.16	1	44527.00	0.4	0.1
1100	150593.00	2.26	0.4	155661.00	5.701	1	154365.00	4.82	0.85	150593.00	2.26	0.4
1100	145348.00	2.96	0.47	150107.00	6.33	1	146519.00	3.79	0.6	145348.00	2.96	0.5
1100	220956.00	8.77	0.58	233887.00	15.14	1	222867.00	9.72	0.64	217424.00	7.04	0.5
1100	122137.00	2.39	0.82	121011.00	1.446	0.5	121895.00	2.19	0.75	122137.00	2.39	0.8
1100	141041.00	3.46	0.71	142961.00	4.869	1	136323.00	0	0	141041.00	3.46	0.7
1100	130553.00	3.45	0.76	131902.00	4.517	1	131803.00	4.44	0.98	130553.00	3.45	0.8
1100	117319.00	0.99	0.32	118556.00	2.057	0.67	119732.00	3.07	1	117319.00	0.99	0.3
1100	142845.00	2.65	1	139160.00	0	0	139225.00	0.05	0.02	142845.00	2.65	1
1100	156478.00	4.82	0.95	156843.00	5.066	1	156643.00	4.93	0.97	156478.00	4.82	1
1100	181722.00	3.28	0.27	197624.00	12.32	1	197653.00	12.3	1	178962.00	1.71	0.1
1101	618542.00	3.88	1	616362.00	3.512	0.91	610536.00	2.53	0.65	595451.00	0	0
1101	626565.00	5.42	0.81	628763.00	5.787	0.86	611500.00	2.88	0.43	606735.00	2.08	0.3
1101	1208714.0	14.9	1	1164515.0	10.67	0.72	1105033.00	5.02	0.34	1181542.0	12.3	0.8
1101	455197.00	6.21	1	433491.00	1.141	0.18	443763.00	3.54	0.57	444320.00	3.67	0.6
1101	619346.00	10.2	1	594131.00	5.721	0.56	561978.00	0	0	587335.00	4.51	0.4
1101	677624.00	10.9	1	629595.00	3.026	0.28	628706.00	2.88	0.26	653522.00	6.94	0.6
1101	449365.00	4.53	1	434456.00	1.062	0.23	439844.00	2.32	0.51	435822.00	1.38	0.3
1101	642075.00	7.28	1	598529.00	0	0	599489.00	0.16	0.02	613439.00	2.49	0.3
1101	825580.00	9.84	1	753750.00	0.28	0.03	751997.00	0.05	0	777522.00	3.44	0.3
1101	975942.00	4.7	0.68	996044.00	6.853	0.99	996493.00	6.9	1	938596.00	0.69	0.1
1110	81713.00	0	0	81936.00	0.273	0.85	81713.00	0	0	81713.00	0	0
1110	73598.00	0	0	73598.00	0	0	73598.00	0	0	73598.00	0	0
1110	86371.00	0	0	86861.00	0.567	1	86371.00	0	0	86371.00	0	0
1110	75451.00	0.05	0.03	75451.00	0.05	0.03	76695.00	1.7	1	75451.00	0.05	0
1110	76346.00	0	0	76346.00	0	0	76346.00	0	0	76346.00	0	0
1110	70317.00	0.01	0.01	70796.00	0.69	1	70317.00	0.01	0.01	70317.00	0.01	0
1110	77631.00	0	0	77641.00	0.013	0.18	77631.00	0	0	77631.00	0	0
1110	76966.00	0	0	76998.00	0.042	1	76966.00	0	0	76966.00	0	0
1110	74578.00	0	0	74715.00	0.184	1	74578.00	0	0	74578.00	0	0
1110	68933.00	0	0	69161.00	0.331	1	68933.00	0	0	68933.00	0	0
1111	103910.00	0.19	0.05	104133.00	0.408	0.11	107435.00	3.59	1	103910.00	0.19	0.1
1111	92540.00	0	0	110452.00	19.36	1	95075.00	2.74	0.14	92540.00	0	0
1111	110011.00	0	0	110895.00	0.804	1	110011.00	0	0	110011.00	0	0
1111	92251.00	0.49	0.03	107224.00	16.8	1	93915.00	2.31	0.14	92251.00	0.49	0
1111	94976.00	0.34	0.12	94976.00	0.338	0.12	97434.00	2.93	1	94976.00	0.34	0.1
1111	85032.00	0.39	0.12	85184.00	0.573	0.18	87364.00	3.15	1	85032.00	0.39	0.1
1111	93075.00	0	0	93085.00	0.011	0	95731.00	2.85	1	93075.00	0	0
1111	95155.00	0	0	95187.00	0.034	0.01	99240.00	4.29	1	95155.00	0	0
1111	92120.00	0.39	0.1	91899.00	0.148	0.04	95396.00	3.96	1	92120.00	0.39	0.1
1111	86573.00	0	0	87221.00	0.749	0.3	88767.00	2.53	1	86573.00	0	0
AVG	231895.88	2.41	0.35	227173.13	2.40	0.51	225707.02	1.80	0.39	227233.18	1.52	0.29

Table C.5: Computational results for SPT, FFD, MFFD, EGI for n=100 (continued)

v-p-tl-tc	Knap Flowtime	d1	d2	2Bin Flowtime	d1	d2	GAPS Flowtime	d1	d2
0000	93090	3.09	0.60	93077.00	3.08	0.6	90297.00	0	0
0000	98796	1.76	1.00	97089.00	0	0	97089.00	0	0
0000	141702	0.00	0.00	141702.00	0	0	141702.00	0	0
0000	77364	0.00	0.00	77641.00	0.36	0.3	77420.00	0.072	0.07
0000	90667	1.53	1.00	89302.00	0	0	89302.00	0	0
0000	87767	0.04	0.00	94234.00	7.41	0.9	87730.00	0	0
0000	75497	3.99	0.88	73647.00	1.44	0.3	72603.00	0	0
0000	93416	2.34	1.00	91756.00	0.52	0.2	91284.00	0	0
0000	94812	0.00	0.00	103380.00	9.04	0.9	94812.00	0	0
0000	149054	0.00	0.00	149054.00	0	0	149054.00	0	0
0001	384654	4.90	0.46	389009.00	6.09	0.6	366695.00	0	0
0001	430031	0.40	1.00	428324.00	0	0	428324.00	0	0
0001	809952	0.00	0.00	809952.00	0	0	809952.00	0	0
0001	287469	0.39	0.10	293989.00	2.67	0.7	286350.00	0	0
0001	385573	0.36	1.00	384208.00	0	0	384208.00	0	0
0001	449975	0.01	0.00	513786.00	14.2	0.9	449938.00	0	0
0001	293537	7.57	0.54	286700.00	5.07	0.4	272876.00	0	0
0001	419266	0.51	0.15	418819.00	0.4	0.1	417134.00	0	0
0001	492594	0.00	0.00	573117.00	16.3	0.9	492594.00	0	0
0001	861854	0.00	0.00	861854.00	0	0	861854.00	0	0
0010	49861	0.14	0.13	49790.00	0	0	49790.00	0	0
0010	48519	0.21	0.34	48417.00	0	0	48417.00	0	0
0010	50052	0.16	1.00	49970.00	0	0	49970.00	0	0
0010	48224	0.18	0.16	48137.00	0	0	48137.00	0	0
0010	49014	0.48	0.76	48782.00	0	0	48782.00	0	0
0010	47198	0.36	1.00	47030.00	0	0	47030.00	0	0
0010	48256	0.33	0.45	48099.00	0	0	48099.00	0	0
0010	48549	0.18	0.29	48462.00	0	0	48462.00	0	0
0010	47836	0.04	1.00	47817.00	0	0	47817.00	0	0
0010	47610	0.19	1.00	47540.00	0.04	0.2	47520.00	0	0
0011	66553	0.00	0.00	66638.00	0.13	0	66638.00	0.128	0.02
0011	65104	0.00	0.00	65129.00	0.04	0	65259.00	0.238	0.04
0011	66522	0.12	0.05	66440.00	0	0	66440.00	0	0
0011	62434	0.14	0.06	62347.00	0	0	62347.00	0	0
0011	64194	0.25	0.04	64100.00	0.11	0	64031.00	0	0
0011	59966	0.27	0.07	59804.00	0	0	59865.00	0.102	0.03
0011	60736	0.30	0.06	60555.00	0	0	60699.00	0.238	0.05
0011	63312	0.14	0.02	63225.00	0	0	63225.00	0	0
0011	61981	0.03	0.01	61962.00	0	0	61962.00	0	0
0011	64170	0.06	0.01	64244.00	0.17	0	64133.00	0	0
0100	188762	4.50	0.86	184981.00	2.4	0.5	180639.00	0	0
0100	203621	1.58	1.00	200447.00	0	0	200447.00	0	0
0100	283984	0.00	0.00	283984.00	0	0	283984.00	0	0
0100	158316	1.41	1.00	156958.00	0.54	0.4	156110.00	0	0
0100	184113	1.43	1.00	181513.00	0	0	181513.00	0	0
0100	180618	0.81	0.15	189148.00	5.57	1	179164.00	0	0
0100	150145	2.95	0.91	148450.00	1.79	0.5	145839.00	0	0
0100	187078	2.39	1.00	183447.00	0.4	0.2	182708.00	0	0
0100	204832	0.00	0.00	213655.00	4.31	1	204832.00	0	0
0100	299104	0.00	0.00	299104.00	0	0	299104.00	0	0
0101	788059	6.65	0.70	777415.00	5.21	0.5	738899.00	0	0
0101	904841	0.35	1.00	901667.00	0	0	901667.00	0	0
0101	1620484	0.00	0.00	1620484.00	0	0	1620484.0	0	0
0101	590371	1.69	0.36	596769.00	2.79	0.6	580572.00	0	0
0101	812949	0.32	1.00	810349.00	0	0	810349.00	0	0
0101	941271	1.63	0.15	1024506.00	10.6	1	926178.00	0	0
0101	576956	3.93	0.41	580715.00	4.61	0.5	555149.00	0	0
0101	838778	0.52	0.12	836450.00	0.24	0.1	834408.00	0	0
0101	1088302	0.00	0.00	1167685.00	7.29	1	1088302.0	0	0
0101	1724704	0.00	0.00	1724704.00	0	0	1724704.0	0	0

Table C.6: Computational results for Knap, 2Bin, GAPS algorithms for n=100

	Knap			2Bin			GAPS		
v-p-tl-tc	Flowtime	d1	d2	Flowtime	d1	d2	Flowtime	d1	d2
0110	100018	0.31	0.30	99708.00	0	0	99708.00	0	0
0110	97501	0.44	0.66	97103.00	0.03	0	97071.00	0	0
0110	101093	0.52	1.00	100634.00	0.06	0.1	100574.00	0	0
0110	97092	0.22	0.17	96936.00	0.06	0	96879.00	0	0
0110	97950	0.41	0.61	97624.00	0.07	0.1	97553.00	0	0
0110	94840	0.10	1.00	94742.00	0	0	94742.00	0	0
0110	96974	0.14	0.16	96835.00	0	0	96835.00	0	0
0110	98024	0.47	0.80	97562.00	0	0	97562.00	0	0
0110	97003	0.40	1.00	96630.00	0.02	0	96613.00	0	0
0110	96611	0.47	1.00	96160.00	0	0	96160.00	0	0
0111	133606	0.23	0.04	133296.00	0	0	133296.00	0	0
0111	130981	0.55	0.10	130550.00	0.22	0	130269.00	0	0
0111	134843	0.78	0.27	133837.00	0.03	0	133793.00	0	0
0111	125703	0.30	0.10	125330.00	0	0	125836.00	0.404	0.13
0111	128530	0.52	0.08	128146.00	0.22	0	127863.00	0	0
0111	120485	0.24	0.06	120288.00	0.08	0	120197.00	0	0
0111	122279	0.15	0.03	122100.00	0	0	122140.00	0.033	0.01
0111	128082	0.35	0.05	127638.00	0	0	127861.00	0.175	0.03
0111	125668	0.28	0.07	125773.00	0.36	0.1	125322.00	0	0
0111	130307	0.37	0.09	129829.00	0	0	129977.00	0.114	0.03
1000	76393	2.60	0.34	75301.00	1.13	0.1	74456.00	0	0
1000	74531	2.31	0.65	74603.00	2.41	0.7	72850.00	0	0
1000	103792	0.29	0.02	103493.00	0	0	103820.00	0.316	0.02
1000	61739	1.98	0.80	61453.00	1.51	0.6	60538.00	0	0
1000	72481	2.86	1.00	70522.00	0.08	0	70463.00	0	0
1000	64223	0.19	0.03	64367.00	0.41	0.1	64104.00	0	0
1000	58628	1.15	0.31	58392.00	0.74	0.2	57962.00	0	0
1000	71096	0.66	0.16	71513.00	1.25	0.3	71451.00	1.16	0.28
1000	73875	0.00	0.00	74631.00	1.02	0.1	75261.00	1.876	0.22
1000	88030	0.00	0.00	90781.00	3.13	0.2	90558.00	2.872	0.19
1001	309813	4.01	0.53	308438.00	3.55	0.5	297875.00	0	0
1001	318278	5.12	0.60	311702.00	2.94	0.3	302786.00	0	0
1001	538317	1.47	0.18	530496.00	0	0	536184.00	1.072	0.13
1001	225347	3.44	0.59	221544.00	1.69	0.3	217860.00	0	0
1001	309837	6.21	0.71	299521.00	2.68	0.3	291714.00	0	0
1001	322916	2.29	0.45	321625.00	1.88	0.4	318907.00	1.015	0.2
1001	221591	0.78	0.18	221909.00	0.92	0.2	219879.00	0	0
1001	306660	1.21	0.12	310655.00	2.53	0.3	307319.00	1.425	0.14
1001	364977	0.00	0.00	373215.00	2.26	0.3	380296.00	4.197	0.64
1001	454822	0.00	0.00	475259.00	4.49	0.4	464176.00	2.057	0.18
1010	41138	0.02	1.00	41129.00	0	0	41129.00	0	0
1010	37739	0.42	0.78	37583.00	0	0	37583.00	0	0
1010	43836	0.16	1.00	43767.00	0	0	43767.00	0	0
1010	38347	0.04	0.15	38333.00	0	0	38333.00	0	0
1010	39028	0.27	1.00	38922.00	0	0	38922.00	0	0
1010	35615	0.13	0.75	35569.00	0	0	35569.00	0	0
1010	39149	0.12	0.24	39104.00	0	0	39106.00	0.005	0.01
1010	39166	0.25	1.00	39069.00	0	0	39069.00	0	0
1010	37926	0.03	1.00	37914.00	0	0	37914.00	0	0
1010	35203	0.17	1.00	35143.00	0	0	35143.00	0	0

Table C.7: Computational results for Knap, 2Bin, GAPS algorithms for n=100 (continued)

v-p-tl-tc	Knap Flowtime	Knap d1	Knap d2	2Bin Flowtime	2Bin d1	2Bin d2	GAPS Flowtime	GAPS d1	GAPS d2
1011	52358	0.02	0.01	52349.00	0	0	52349.00	0	0
1011	47813	1.26	0.35	47219.00	0	0	47219.00	0	0
1011	56060	0.12	1.00	55991.00	0	0	55991.00	0	0
1011	46915	0.03	0.00	46901.00	0	0	46901.00	0	0
1011	48828	0.61	0.16	48541.00	0.02	0	48530.00	0	0
1011	43088	0.11	0.03	43042.00	0	0	43042.00	0	0
1011	47186	0.46	0.16	46970.00	0	0	46972.00	0.004	0
1011	48754	0.20	0.04	48657.00	0	0	48657.00	0	0
1011	46800	0.03	0.01	46788.00	0	0	46788.00	0	0
1011	44383	0.07	0.02	44360.00	0.02	0	44350.00	0	0
1100	150489	2.19	0.38	149583.00	1.57	0.3	147266.00	0	0
1100	147222	4.29	0.68	144215.00	2.16	0.3	141171.00	0	0
1100	203132	0.00	0.00	211042.00	3.89	0.3	209625.00	3.196	0.21
1100	122766	2.92	1.00	121919.00	2.21	0.8	119286.00	0	0
1100	140833	3.31	0.68	139072.00	2.02	0.4	137630.00	0.959	0.2
1100	126202	0.00	0.00	126243.00	0.03	0	126215.00	0.01	0
1100	118907	2.36	0.77	116166.00	0	0	116252.00	0.074	0.02
1100	142375	2.31	0.87	140935.00	1.28	0.5	139160.00	0	0
1100	149281	0.00	0.00	151500.00	1.49	0.3	149447.00	0.111	0.02
1100	177733	1.01	0.08	177888.00	1.1	0.1	175952.00	0	0
1101	609378	2.34	0.60	597374.00	0.32	0.1	597356.00	0.32	0.08
1101	634302	6.72	1.00	612345.00	3.02	0.5	594369.00	0	0
1101	1052202	0.00	0.00	1143033.00	8.63	0.6	1135340.0	7.901	0.53
1101	451206	5.27	0.85	451463.00	5.33	0.9	428600.00	0	0
1101	604153	7.50	0.74	588661.00	4.75	0.5	580081.00	3.221	0.32
1101	627820	2.74	0.25	637352.00	4.3	0.4	611101.00	0	0
1101	448847	4.41	0.97	437048.00	1.66	0.4	429891.00	0	0
1101	626899	4.74	0.65	620934.00	3.74	0.5	598529.00	0	0
1101	753943	0.31	0.03	767897.00	2.16	0.2	751646.00	0	0
1101	942133	1.07	0.15	940133.00	0.85	0.1	932166.00	0	0
1110	81975	0.32	1.00	81713.00	0	0	81713.00	0	0
1110	73819	0.30	1.00	73598.00	0	0	73598.00	0	0
1110	86371	0.00	0.00	86371.00	0	0	86371.00	0	0
1110	75526	0.15	0.09	75450.00	0.05	0	75413.00	0	0
1110	76542	0.26	1.00	76346.00	0	0	76346.00	0	0
1110	70450	0.20	0.29	70317.00	0.01	0	70311.00	0	0
1110	77687	0.07	1.00	77631.00	0	0	77631.00	0	0
1110	76990	0.03	0.75	76966.00	0	0	76966.00	0	0
1110	74595	0.02	0.12	74578.00	0	0	74578.00	0	0
1110	69012	0.11	0.35	68933.00	0	0	68933.00	0	0
1111	104172	0.45	0.12	103710.00	0	0	103910.00	0.193	0.05
1111	92761	0.24	0.01	92540.00	0	0	92540.00	0	0
1111	110011	0.00	0.00	110011.00	0	0	110011.00	0	0
1111	91906	0.12	0.01	91994.00	0.21	0	91799.00	0	0
1111	95172	0.55	0.19	94656.00	0	0	94681.00	0.026	0.01
1111	84838	0.16	0.05	84844.00	0.17	0.1	84699.00	0	0
1111	93482	0.44	0.15	93075.00	0	0	93075.00	0	0
1111	95179	0.03	0.01	95155.00	0	0	95155.00	0	0
1111	91779	0.02	0.00	91971.00	0.23	0.1	91763.00	0	0
1111	86652	0.09	0.04	86573.00	0	0	86573.00	0	0
AVG	225296.6	1.04	0.40	227439.29	1.25	0.1	223113.04	0.21	0.02

Table C.8: Computational results for Knap, 2Bin, GAPS algorithms for n=100 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0000	0	0	0.02	1.58	3.86	4.92	11.38
0000	0	0.01	0.01	1.71	5.04	5.16	11.39
0000	0	0.01	0.01	1.9	6.18	4.92	17.46
0000	0	0	0.01	1.35	5.14	5.09	10.36
0000	0	0.01	0.01	1.52	5.01	4.91	11.48
0000	0	0.01	0.01	1.61	6.32	4.98	14.95
0000	0.01	0.01	0.01	1.49	3.63	4.95	11.39
0000	0	0.01	0.01	1.55	4.97	4.89	11.88
0000	0	0.01	0.01	1.87	6.05	5.1	15.11
0000	0	0.02	0.02	2.01	7.07	4.91	17.67
0001	0	0.01	0	1.31	3.74	4.99	11.41
0001	0.01	0.01	0	1.39	4.92	4.97	11.7
0001	0	0.02	0.02	1.7	5.95	4.79	17.75
0001	0	0.01	0	1.31	4.87	5.05	10.51
0001	0	0.01	0.01	1.36	4.92	4.86	11.24
0001	0	0.01	0.02	1.61	6.19	4.71	15.14
0001	0.01	0.01	0	1.4	3.63	4.99	11.23
0001	0.01	0.01	0.01	1.39	4.84	4.9	11.92
0001	0	0.01	0.02	1.38	6.26	4.97	14.95
0001	0	0.02	0.02	1.93	6.99	4.81	17.55
0010	0	0.01	0.01	1.16	0.45	4.73	11.04
0010	0	0	0.01	1.18	4.56	5.93	11.03
0010	0	0.01	0	1.21	0.91	7.1	18.03
0010	0.01	0	0.01	1.28	0.4	6.2	10.21
0010	0	0	0.01	1.15	4.65	2.26	11.18
0010	0	0.01	0	1.25	0.43	5.73	15.34
0010	0	0.01	0.01	1.08	0.49	5.71	11.12
0010	0	0	0.01	1.11	0.42	6.29	11.33
0010	0	0.01	0	1.13	0.44	6.07	15.81
0010	0.01	0.01	0.01	1.12	4.68	4456.65	17.97
0011	0	0.01	0.01	1.18	0.52	5.71	10.8
0011	0	0	0.01	1.09	4.65	6.1	10.7
0011	0.01	0	0.01	1.09	0.78	5.67	10.43
0011	0.01	0	0.01	1.1	0.38	6.34	10.72
0011	0	0.01	0.01	1.23	4.03	8.32	10.56
0011	0.01	0	0.01	1.23	0.44	5.95	10.4
0011	0.01	0	0.01	1.03	0.47	5.83	10.84
0011	0	0	0.01	1.14	0.47	5.46	11.04
0011	0	0.01	0	1.21	0.41	9.8	10.29
0011	0.01	0	0.01	1.1	4.2	19.85	10.36
0100	0	0	0	1.08	3.48	4.93	0.88
0100	0	0	0	1.16	4.36	4.64	0.97
0100	0	0.01	0	1.42	6.97	4.59	1.2
0100	0	0	0	1.38	5.22	4.78	0.82
0100	0	0	0	1.28	4.12	4.47	1.17
0100	0	0	0	1.34	7.13	4.32	1.21
0100	0	0	0	1.21	3.83	4.85	0.92
0100	0	0.01	0	1.21	4.87	4.27	1.27
0100	0	0	0	1.31	7.15	4.31	0.89
0100	0	0	0	1.35	7.12	4.83	0.96
0101	0	0.01	0	1.5	3.48	5.01	11.04
0101	0	0.01	0	1.48	4.36	5.03	11.03
0101	0	0.01	0	1.77	6.97	5.16	18.03
0101	0.01	0.01	0	1.42	5.22	5.13	10.21
0101	0.01	0.01	0	1.68	4.12	5.2	11.18
0101	0	0.01	0.01	1.45	7.13	5.08	15.34
0101	0.01	0	0.01	1.53	3.83	4.99	11.12
0101	0.01	0	0	1.65	4.87	4.93	11.33
0101	0.01	0.01	0	1.78	7.15	4.88	15.81
0101	0	0.02	0	2.14	7.12	4.95	17.97

Table C.9: Computation times for n=100

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
0110	0.01	0.01	0.01	1.5	1.47	21.99	10.34
0110	0.01	0.01	0.01	1.33	0.47	14.41	10.26
0110	0.01	0	0	1.4	0.97	7.09	9.76
0110	0	0	0	1.5	0.43	28.61	10.28
0110	0.01	0.01	0.01	1.5	6.51	1707.86	9.99
0110	0	0	0	1.56	0.43	7.1	9.95
0110	0	0	0	1.45	2.32	7.26	10.52
0110	0	0.01	0.01	1.49	0.45	16.24	10.65
0110	0	0	0	1.42	0.43	8.74	9.99
0110	0.01	0	0	1.31	3.48	37.44	10.06
0111	0	0.01	0	1.44	1.79	21.85	10.05
0111	0	0.01	0.01	4.46	0.48	59.34	9.92
0111	0	0	0.01	4.57	1	8.98	9.65
0111	0	0.01	0	1.57	0.46	31.58	10.46
0111	0	0	0.01	1.38	6.86	918.49	10.04
0111	0.01	0.01	0.01	1.49	0.48	7.03	10.11
0111	0	0.01	0	1.29	2.37	7.14	10.27
0111	0	0	0	1.42	0.42	10.67	10.29
0111	0	0.01	0.01	1.48	0.5	8.55	9.9
0111	0.01	0.01	0	1.4	3.44	30.78	9.91
1000	0.01	0.01	0.01	1.24	3.17	1.16	8.9
1000	0.01	0	0.01	1.22	2.84	1.14	9.24
1000	0.01	0.01	0.01	1.41	4.5	1.15	10.01
1000	0	0.01	0	1.36	1.96	1.34	8.78
1000	0	0	0.01	1.24	3.01	1.59	9.29
1000	0	0.01	0.01	1.48	4.08	1.03	9.78
1000	0.01	0.01	0	1.33	2.53	1.03	8.85
1000	0.01	0	0.01	1.43	3.08	1.18	9.45
1000	0	0.01	0.01	1.32	4.28	1.05	9.69
1000	0	0.01	0	1.35	3.53	0.96	9.89
1001	0	0.01	0	1.27	3.25	5.5	9.45
1001	0.01	0.01	0	1.39	2.75	5.84	9.36
1001	0	0.01	0.01	1.51	4.53	5.2	13.9
1001	0	0	0	1.54	2.13	6.99	9.39
1001	0.01	0.01	0	1.51	3.16	7.11	9.5
1001	0	0.01	0.01	1.4	4.08	5.5	10.54
1001	0	0.01	0.01	1.32	2.67	5.57	9.48
1001	0.01	0.01	0	1.32	2.95	5.39	9.55
1001	0	0	0.01	1.44	4.22	5.37	10.08
1001	0	0.01	0.01	1.54	3.43	5.24	10.39
1010	0.01	0.01	0.01	1.32	0.46	3	10.51
1010	0	0.01	0	1.44	0.4	1.5	10.8
1010	0.01	0.01	0.01	1.3	0.38	2.47	10.27
1010	0.01	0.01	0	1.22	0.8	4.24	11.06
1010	0.01	0.01	0	1.18	0.51	1.74	10.52
1010	0	0.01	0.01	1.48	0.47	6.42	10.68
1010	0	0.01	0	1.45	0.46	1.74	10.71
1010	0	0.01	0	1.42	0.38	1.9	10.72
1010	0.01	0.01	0.01	1.41	0.35	1.5	10.61
1010	0.01	0.01	0.01	1.3	0.12	6.88	10.57

Table C.10: Computation times for n=100 (continued)

v-p-tl-tc	SPT	FFD	MFFD	EGI	Knap	2bin	GAPS
1011	0	0.02	0.01	1.25	0.42	12.89	10.68
1011	0.01	0.01	0.01	1.43	0.38	6.55	10.92
1011	0	0.01	0	1.29	0.35	9.85	10.52
1011	0	0	0.01	1.38	0.76	17.13	11.2
1011	0	0.01	0.01	1.26	0.44	7.85	10.93
1011	0	0.01	0	1.49	0.45	25.02	10.81
1011	0.01	0	0	1.24	0.46	7.46	10.76
1011	0.01	0.01	0.01	1.37	0.35	7.7	10.68
1011	0	0.01	0	1.29	0.35	6.54	10.76
1011	0	0	0	1.33	1.1	27.78	10.98
1100	0.01	0	0.01	1.76	2.8	6.47	8.88
1100	0	0	0	1.56	3.16	5.8	8.89
1100	0	0.01	0.01	1.74	3.82	5.26	10.04
1100	0.01	0.01	0.01	1.64	3.68	6.93	8.5
1100	0	0	0	1.8	3.55	5.72	9.08
1100	0	0.01	0.01	2.19	4.12	5.88	9.46
1100	0	0.01	0.01	1.69	2.92	6.24	8.69
1100	0	0	0	1.75	3.06	5.58	9.15
1100	0	0.01	0.01	2.01	3.9	5.07	9.57
1100	0.01	0.01	0.01	1.81	3.43	5.51	9.8
1101	0	0.01	0	1.38	3.05	6.35	9.4
1101	0	0.01	0.01	1.54	3.18	5.54	9.54
1101	0.02	0.01	0.01	1.5	3.79	7.21	10.54
1101	0.01	0.01	0.01	1.4	3.65	5.37	9.04
1101	0	0	0	1.3	3.56	5.88	9.5
1101	0	0.01	0.01	1.51	4.14	6.23	10.07
1101	0.01	0	0	1.44	2.94	5.28	9.23
1101	0.01	0.01	0.01	1.5	2.96	5.16	9.75
1101	0.01	0.01	0	1.79	4.05	5.48	10.09
1101	0.01	0.01	0.01	1.5	3.38	4.91	10.12
1110	0	0.02	0.01	1.43	0.6	1.72	10.73
1110	0	0.01	0	1.22	0.39	1.85	10.96
1110	0	0	0.01	1.31	0.72	2.4	10.29
1110	0	0.01	0.01	1.32	0.59	3.89	10.94
1110	0	0.01	0	1.31	0.94	1.72	10.65
1110	0	0	0.01	1.35	0.53	2.58	10.86
1110	0	0.01	0.01	1.54	0.37	20.03	11.05
1110	0	0.01	0	1.41	0.48	8.45	10.56
1110	0.01	0	0.01	1.43	1.46	2.93	10.7
1110	0.01	0	0.01	1.37	0.64	2.08	10.95
1111	0	0.01	0.01	1.05	0.61	2.39	0.83
1111	0.01	0.01	0.01	1.26	0.41	9.33	0.91
1111	0	0.01	0	1.25	0.73	15	0.85
1111	0	0.01	0.01	1.33	0.61	6.79	0.79
1111	0	0.01	0.01	1.29	0.87	10.15	0.85
1111	0	0.01	0	1.14	0.52	70.99	0.83
1111	0	0.01	0	1.31	0.32	29.44	0.94
1111	0	0.01	0.01	1.33	0.53	11.4	0.81
1111	0.01	0.01	0	1.22	1.59	7.9	0.89
1111	0.01	0.01	0	1.22	0.65	9.78	0.91
AVG	0.003438	0.00725	0.006	1.449438	2.755375	51.92663	9.795438

Table C.11: Computation times for n=100 (continued)

VITA

Evrim Didem Güneş was born on December 11, 1975 in K.Maraş, Turkey. She attended Atatürk Anadolu Lisesi and Özel Büyük Lise in Ankara, Turkey. She has graduated with honors from the department of Industrial Engineering, Bilkent University, in 1997. In September 1997, she joined the department of Industrial Engineering at Bilkent University as a research assistant. From that time to the present, she worked with Dr. Selim Aktürk for her graduate study at the same department.