A HEURISTIC APPROACH FOR THE SINGLE MACHINE SCHEDULING TARDINESS PROBLEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAFFET İLKER ÖZBAKIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2011

Approval of the thesis:

A HEURISTIC APPROACH FOR THE SINGLE MACHINE SCHEDULING TARDINESS PROBLEMS

submitted by **SAFFET İLKER ÖZBAKIR** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Sinan Kayalıgil ____________________
Head of Department, **Industrial Engineering**

Prof. Dr. Ömer Kırca ____________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Meral Azizoğlu ____________________
Industrial Engineering Dept., METU

Prof. Dr. Ömer Kırca ____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. İsmail Serdar Bakal ____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Sinan Gürel ____________________
Industrial Engineering Dept., METU

Assist. Prof. Dr. Ferda Can Çetinkaya ____________________
Industrial Engineering Dept., Çankaya University

Date: 14.09.2011

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Saffet İlker Özbakır

Signature :

ABSTRACT

A HEURISTIC APPROACH FOR THE SINGLE MACHINE SCHEDULING TARDINESS PROBLEMS

Özbakır, Saffet İlker
M.Sc., Department of Industrial Engineering
Supervisor : Prof. Dr. Ömer Kırca

September 2011, 102 pages

In this thesis, we study the single machine scheduling problem. Our general aim is to schedule a set of jobs to the machine with a goal to minimize tardiness value. The problem is studied for two objectives: minimizing total tardiness value and minimizing total weighted tardiness value.

Solving optimally this problem is difficult, because both of the total tardiness problem and total weighted tardiness problem are NP-hard problems. Therefore, we construct a heuristic procedure for this problem. Our heuristic procedure is divided to two parts: construction part and improvement part. The construction heuristic is based on grouping the jobs, solving these groups and then fixing some particular number of jobs. Moreover, we used three type improvement heuristics. These are sliding forward method, sliding backward method and pairwise interchange method.

Computational results are reported for problem size = 20, 40, 50 and 100 at total tardiness problem and for problem size = 20 and 40 at total weighted tardiness problem. Experiments are designed in order to investigate the effect of three factors which are problem size, tardiness factor and relative range of due dates on computational difficulties of the problems. Computational results show that the heuristic proposed in this thesis is robust to changes at these factors.

Keywords: Single machine, scheduling, tardiness, weighted tardiness

ÖZ

TEK MAKİNE TAKVİMLEME GECİKME PROBLEMLERİ İÇİN BİR SEZGİSEL YAKLAŞIM

Özbakır, Saffet İlker
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ömer Kırca

Eylül 2011, 102 sayfa

Bu tezde tek makine takvimleme problem üzerine çalışıldı. Bu problemde genel amaç bir iş setini gecikme değerini enazlayacak şekilde makineye takvimlemektir. Problem iki hedef için çalışıldı: toplam gecikme değerini enazlamak ve toplam ağırlıklı gecikme değerini enazlamak.

Toplam gecikme ve ağırlıklı gecikme problemlerinin ikisinin de NP-zor problemler olmalarından dolayı, bu problemi en iyi şekilde çözmek oldukça zor. Bu yüzden, bu problem için bir sezgisel yaklaşım prosedürü geliştirildi. Sezgisel yaklaşım prosedürü iki bölümden oluşmaktadır: yapı kısmı ve geliştirme kısmı. Sezgisel yaklaşımın yapı kısmı işleri gruplamaya, bu grupları çözmeye ve sonra belirli sayıda işin sabitlenmesine dayanıyor. Bununla birlikte, sezgisel yaklaşımın geliştirme kısmı için üç metot kullanıldı. Bunlar ileriye doğru kaydırma metodu, geriye doğru kaydırma metodu ve ikili değiştirme metodudur.

İşlemler sonuçlar toplam gecikme probleminde problem büyüklüğü = 20, 40, 50 ve 100 için; toplam ağırlıklı gecikme probleminde de problem büyüklüğü = 20 ve 40 için rapor edildi. Deneyler, üç faktörün (problem büyüklüğü, gecikme faktörü ve teslim tarihininin göreceli genişliği) problemin işlemsel zorluğu üzerindeki etkilerini araştırmak için tasarlandı. İşlemsel sonuçlar bu tezde sunulan sezgisel yaklaşımın bu faktörlerdeki değişimlere dayanıklı olduğunu gösteriyor.

Anahtar Kelimeler: Tek makine, takvimleme, gecikme, ağırlıklı gecikme

To my family

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Prof. Ömer Kırca for his guidance and contributions. His vision, systematic thinking and knowledge contributed this study very much.

I would like to thank to TÜBİTAK (The Scientific and Technological Research Council of Turkey) for the generous support through my graduate studies.

I am grateful to my friends for their companionship and support throughout this study.

I would like to thank to my father, Tuncel Özbakır and to my mother, Kamile Özbakır for their love, support and encouragement in every stage of my life. I know that my achievements make them happier than me.

İlker

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Scheduling is the process of allocating resources between a set of tasks. There are tasks to be scheduled and there are particular resources to perform these tasks. In production scheduling, these tasks are jobs waiting to be processed and resources are machines. Scheduling problems can be categorized by specifying the resource configuration. A problem may contain one machine or several machines. If it contains one machine, jobs are likely to be single stage, while multiple-machine problems generally involve jobs with multiple stages. In this thesis, single machine scheduling problem is studied. In this problem, there is only one machine in order to process jobs.

Sometimes in order to completely understand a complex system, it is necessary to understand its parts. Single machine problem generally is a part of a larger scheduling problem. In some situations, it may be possible to solve the imbedded single machine problem independently and then to incorporate the result into the larger problem. For example, there may be a bottleneck stage in multiple operation processes and single machine analysis can be a reasonable approach to this problem.

Baker (1995) states the fact that in order to evaluate the solutions of single machine scheduling, there are different decision-making goals. These goals can be classified as *turnaround, timeliness* and *throughput*. Turnaround measures the completion time of a task. Timeliness measures the conformance of a particular task's completion to a given due date. Finally, throughput measures the amount of work completed during a fixed period of time. In this thesis, we deal with timeliness type decision-making goal. In this thesis total tardiness and total weighted tardiness are used as performance measures. Tardiness is based on "meeting job due dates" criteria. Tardiness penalizes the jobs by the amount that their completion times exceed their due dates. The difficulty of dealing with tardiness measure is the fact that tardiness is not a linear function of completion time.

It is shown that single machine total tardiness problem (Du and Leung, 1990) and single machine total weighted tardiness problem (Lawler, 1977 and Lenstra et al., 1977) are NP-Hard problems. Therefore, it may not be possible to find optimal solutions with available techniques at polynomial time. As a result of this, for problems beyond a certain size, it might be better to use heuristic solution procedures that have a more modest computational requirement but do not guarantee optimality.

In this thesis, a heuristic is developed to solve the single machine total tardiness and weighted tardiness problems as close as to optimal solution. A construction heuristic and improvement heuristic are proposed for this problem. The construction heuristic consists of grouping the jobs, solving these groups and fixing some particular number of jobs. These operations continue dynamically until no job to be scheduled remains. In addition to this construction heuristic, improvement heuristic methods are proposed in order to improve the solution of the construction heuristic. There are three improvement heuristic methods in this thesis which are sliding forward, sliding backward and pairwise interchange methods.

The rest of the thesis is organized as follows:

In Chapter 2, the general properties of the single machine total tardiness and weighted tardiness problems and previous researches about these problems are mentioned briefly. Characteristics of the problems are expressed by representing assumptions about the single machine tardiness problem. Furthermore, notation used in this thesis is given and the MIP (Mixed Integer Programming) models of both problems are examined in detail in this chapter. In addition to these, construction and improvement heuristics and quality of these heuristics are searched on literature.

Some studies about the application of genetic algorithms on the single machine tardiness problem are examined in this chapter.

In Chapter 3, the heuristic procedure which is proposed for the single machine tardiness problem in this thesis is expressed. The heuristic can be classified as construction heuristic and improvement heuristic. In this chapter, all details of these heuristics are mentioned step by step.

Chapter 4 reports the computational results of the heuristic for single machine total tardiness problem with 20, 40, 50 and 100 jobs and for single machine total weighted tardiness problem for 20 and 40 jobs. Moreover, comparison of results of the heuristic and the other construction heuristics is given. Also the effects of the parameters such as tardiness factor and relative range of due dates on the solution are discussed.

Finally, in Chapter 5, main conclusions are presented and some possible extensions are discussed.

CHAPTER 2

LITERATURE REVIEW

In this section, general properties of the single machine total tardiness and weighted tardiness problem and previous researches about these topics are examined.

2.1 SINGLE MACHINE TOTAL TARDINESS PROBLEM

In this problem, there is only one machine and it is used to process N jobs. The problem is to schedule a set of N jobs to the machine with a goal to minimize total tardiness value. There are some assumptions about this problem:

1. All jobs are independent from each other.

2. Setup times of the jobs are independent of job sequence and are included in processing times.

3. Job descriptors are deterministic and known in advance.

4. All jobs are available for processing at time 0.

5. The machine can process only one job at a time.

6. No preemption is allowed. The processing of a job cannot be interrupted.

7. The machine is continuously available.

Tardiness of a job is defined as $t_i = \max\{0, c_i - d_i\}$, where c_i is the completion time of job i and d_i is the due date of job i. According to Koulamas (1994), there is no benefit gained from the completing jobs early and the delay penalty is proportional to the delay according to tardiness criterion.

The MIP (Mixed Integer Programming) model of the single machine tardiness problem is given at below.

Sets:

$\boldsymbol{i}$	job
$\boldsymbol{j}$	position

Parameters:

$\boldsymbol{p_i}$	process time of job i
$\boldsymbol{d_i}$	due date of job i

Decision variables:

$$\boldsymbol{x_{ij}} = \begin{cases} 1, & \text{if job } i \text{ is scheduled to position } j \\ 0, & \text{if job } i \text{ is not scheduled to position } j \end{cases}$$

$\boldsymbol{c_j}$	completion time of the job which is scheduled to position j
$\boldsymbol{t_j}$	tardiness value of the job which is scheduled to position j

Objective function:

$$\text{Minimize} \quad \sum_{j=1}^{N} t_j \tag{2.1}$$

Objective is to minimize total tardiness value.

Constraints:

$$\sum_{i=1}^{N} x_{ij} = 1 \quad , \quad \forall j \tag{2.2}$$

Constraint 2.2 provides that only one job can be scheduled to a position.

$$\sum_{j=1}^{N} x_{ij} = 1 \quad , \quad \forall i \tag{2.3}$$

Constraint 2.3 provides that a job can be scheduled to only one position.

$$c_j = c_{j-1} + \sum_{i=1}^{N} x_{ij} * p_i \quad , \quad \forall j \tag{2.4}$$

Constraint 2.4 provides that the completion time of position *j* depends on the completion time of the job at previous position and the process time of the job assigned to position *j*.

$$t_j \geq c_j - \sum_{i=1}^{N} x_{ij} * d_i \quad , \quad \forall j \tag{2.5}$$

Constraint 2.5 provides that if right hand side of the constraint is bigger than zero, then tardiness of position *j* would be equal that value. Because, t_j does not take a higher value than right hand side of the constraint due to this is a minimization problem. If R.H.S of the constraint is negative, tardiness of position *j* would be zero because of the constraint 2.8.

$$x_{ij} \in \{0,1\} \quad , \quad \forall i, j \qquad (2.6)$$

Constraint 2.6 provides that x values can only take value of 0 or 1.

$$c_j \geq 0 \ , \quad \forall j \qquad (2.7)$$

Constraint 2.7 provides that completion time of position j should be higher than or equal to zero.

$$t_j \geq 0 \ , \quad \forall j \qquad (2.8)$$

Constraint 2.8 provides that tardiness value cannot take a negative value. It should be at least zero.

There are a lot of studies about the single machine total tardiness problem. It was first presented by Conway, Maxwell and Miller (1967). According to them, an optimal solution exists for the single machine total tardiness problem in which no job is preempted. Therefore, in order to find the optimal sequence, all combinations of the ordering of N jobs should be tried. One of them would give the optimal result. However, for large size problems, this is too difficult.

Many researches about the single machine total tardiness problem are about the development of heuristic procedures to this problem. The reason of this is the fact that single machine total tardiness problem is an NP-hard problem. This was shown by Du and Leung (1990). They claim that the problem cannot be solved in polynomial time. The computation time of the problem grows exponentially when the problem size increases. Therefore, large size problems cannot be solved optimality.

Exact algorithms for the single machine total tardiness problem are surveyed. One of the most efficient exact algorithms for this problem is developed by Potts and Van Wassenhove (1982). Their algorithm is developed by embedding the decomposition principle into a branch and bound algorithm. This algorithm can solve problems with up to 100 jobs.

There are many construction and improvement heuristic algorithms in order to solve single machine total tardiness problem. In order to select the construction heuristics which are used in this thesis, a general survey about the heuristics is done.

Koulamas (1994) presented a study about the total tardiness problem in all aspects. Only single machine total tardiness part is examined because we study about this topic. He analyzed the construction heuristics about the single machine total tardiness problem and then evaluated them. The construction heuristics which are examined in this study are the simplest construction heuristics which are shortest process time (SPT) and earliest due date (EDD), Montagne heuristic, modified due date (MDD) heuristic, cost over time (COVERT) heuristic, the apparent urgency (AU) heuristic and Panwalker, Smith and Koulamas (PSK) heuristic. Moreover, he analyzed some local search methods which seek improved solutions to a problem by searching in the neighborhood of an incumbent solution. These are also examined for our study, because while developing our improvement heuristics, these heuristics are used as source of inspiration. The local search methods which are examined by Koulamas (1994) are the simplest local search method which is adjacent pair-wise interchange (API), the net benefit of relocation (NBR) heuristic and two hybrid construction-local search heuristics which are Wilkerson-Irwin (WI) heuristic and traffic priority index (TPI) heuristic. Moreover, Koulamas (1994) compared some of these heuristics with each other. There is a comparison of the performance of API, NBR, TPI, WI and PSK heuristics in this study. This comparison is shown in Table 2.1.

The reason why Koulamas chose specifically PSK heuristic to compare with the local search methods is PSK heuristic was developed by him, Panwalker and Smith in 1993. According to Table 2.1, Koulamas (1994) claims that PSK heuristic performs better than the other tested heuristics. It is normal that PSK heuristic has lowest average CPU time, because it is a construction heuristic.

Table 2.1 Comparison of heuristics (Koulamas (1994))

Heuristic	Average Deviation (%)	Maximum Deviation (%)	Number Exact (out of 125)	Average CPU Time (Sec.)
API	0.64	12.48	76	4.12
NBR	2.4	24.2	27	0.97
TPI	1.02	12.24	66	0.12
WI	1.14	12.53	55	0.39
PSK	0.46	12.4	87	0.01

Russell and Holsenback (1997) presented an evaluation of leading heuristics for the single machine tardiness problem. They generally emphasize on two heuristics which are Panwalker, Smith and Koulamas (PSK) heuristic and the net benefit of relocation (NBR) heuristic. They try these heuristics on problems with 50 jobs. According to their study NBR heuristic gives better results than PSK heuristic.

Nyirenda (2001) described the relationship between the modified due date (MDD) heuristic and Wilkerson-Irwin (WI) heuristic for the single machine total tardiness problem. He shows that MDD heuristic and WI heuristic are strongly related in the sense that both are based on the same local optimality condition for a pair of adjacent jobs. As a result of this, adjacent pair-wise interchange (API) method cannot improve the sequence generated by these heuristics. He tries these heuristics on problems with 50 jobs and 100 jobs. In both problem sizes, these two heuristic give same tardiness values.

Fry, Vicens, Macleod and Fernandez (1989) developed a heuristic solution procedure to minimize mean tardiness or equivalently minimizing total tardiness on a single machine. Their heuristic utilizes the adjacent pair-wise interchange (API) method. According to simple API method, beginning from the first position of the initial solution, all adjacent jobs are switched until an improvement on tardiness occurs. When a pair of jobs to be switched is found, the switch is made and the search for a favorable switch begins again at the first position in sequence. However, it is high possibility to obtain a local optimum solution. Therefore, they improve API method. There are three operation types in their heuristic. At first one, API begins at the first position in sequence and proceeds front to back. In second strategy, API begins at the last position in sequence and proceeds back to the front. And finally, API procedure used in third strategy evaluates all adjacent pairs of jobs before switching. The adjacent job pair which gives the maximum improvement is identified and switched. In all strategies, after an adjacent pair has been switched, the procedure starts again until no improvement occurs. They compare this heuristic with Wilkerson-Irwin (WI) heuristic. They show that their heuristic gives better tardiness values than WI heuristic.

In addition to construction and improvement heuristics, there are some meta-heuristics which are applied to the single machine total tardiness problem. Meta-heuristics are used for combinatorial optimization in which an optimal solution is sought over a discrete search-space. Some meta-heuristics are genetic algorithm, simulated annealing, ant colony optimization and tabu search.

Bauer, Bullnheimer, Hartl and Strauss (1999) adapted ant colony optimization to the single machine total tardiness problem. Ant colony optimization models a nature-based, multi-agent process in order to solve hard combinatorial optimization problems. They try this method on problems with 50 jobs. Their method gives optimal solutions for 124 problems among 125 problems. Also, it gives optimal solutions for 609 problems among 625 problems.

Laguna, Barnes and Glover (1990) developed tabu search methods for the single machine total tardiness problem. Tabu search increases the performance of a local search method by using memory structures. After a potential solution is determined, it is marked as tabu and that solution is not visited repeatedly by the algorithm. They adapted this method to the single machine total tardiness problem. They try the method on the problems with 20, 25, 30 and 35 jobs. Generally, the method gives good results.

Despite meta-heuristics give efficient solutions for the single machine total tardiness problem, they are not used in the heuristic proposed in this thesis because of the high computational effort.

2.2 SINGLE MACHINE TOTAL WEIGHTED TARDINESS PROBLEM

Total weighted tardiness problem is a generalization of the total tardiness problem. It is assumed that all weights are 1 at total tardiness problem. But at total weighted tardiness problem, all jobs have different weights.

All assumptions which are expressed for total tardiness problem are also valid for single machine total weighted tardiness problem. At this problem, weighted tardiness is defined as $t_i = \max\{0, w_i * (c_i - d_i)\}$. w_i is the weight of job *i*.

MIP model of this problem is given at below. Only differences from MIP model of single machine total tardiness problem which is given at section 2.1 are shown.

Parameters:

w_i weight of job i

Other parameters are same with model of total tardiness problem.

Decision variables:

r_{ij} tardiness value of job i which is scheduled to position j

Other decision variables are same with model of total tardiness problem.

Objective function:

$$\text{Minimize } \sum_{i=1}^{N}\sum_{j=1}^{N} w_i * r_{ij} \qquad (2.9)$$

Objective is to minimize total weighted tardiness value.

Constraints:

Seven constraints of the MIP model of the total tardiness problem are also valid at this model. In addition, there are two extra constraints for total weighted tardiness problem.

$$r_{ij} \geq t_j - M*(1-x_{ij}) \quad , \ \forall i, j \qquad (2.10)$$

At this constraint, M represents a big number. Constraint 2.10 provides that if x_{ij}=0 which means job i is not scheduled to position j, r_{ij} value must be zero. When x_{ij}=0, the constraint would be $r_{ij} \geq t_j - M$, and because M is a big number, M has a higher value than t_j value. As a result of this, right hand side of the constraint would be negative. r_{ij} should has a non-negative value (this situation will be shown at next constraint). Therefore, r_{ij} must equal to zero, because this is minimization problem and by looking at the objective function, it can be said that r value should be as small as possible. On the other hand, when x_{ij}=1, which means job i is scheduled to position j, the constraint would be $r_{ij} \geq t_j$. This means that r_{ij} should equal to t_j value which is calculated at constraint 2.8 given at model of total tardiness problem.

$$r_{ij} \geq 0 \quad , \ \forall i, j \qquad (2.11)$$

Constraint 2.11 provides that r_{ij} value cannot take a negative value.

Similar to total tardiness problem, single machine total weighted tardiness problem is also NP-hard problem. This is shown by Lawler (1977) and Lenstra, Rinooy Khan and Brucker (1977). Therefore, there are many heuristics and algorithms to solve total weighted tardiness problem in literature.

Potts and Van Wassenhove (1985) developed a branch and bound algorithm for the single machine total weighted tardiness problem. This algorithm can solve problems with up to 40 jobs optimally.

Volgenant and Teerhuis (1998) presented a study about the improved heuristics for the single machine total weighted tardiness problem. First of all, they solve the problem by using four known construction heuristics which are the apparent urgency heuristic (AU), the earliest due date heuristic (EDD), the greedy heuristic and the weighted shortest processing time heuristic (WSPT). Then, they improve the results of these heuristics by applying the priority rule of Rachamadugu (1987). They compare the results of heuristics before and after applying the priority rule for problem size (N) = 20, 40 and 80. They test the heuristics for different tardiness factor (TF) and range of the due dates (R) values. For the problems which process time of jobs are generated with U(1,10), the apparent urgency heuristic gives best results and the relative impact of the priority rule on the solutions decreases when the tardiness factor increases. Moreover, for the problems with process times ~U(1,100), the best solutions are given by greedy heuristic. After these tests, they comment the heuristics according to results. According to them;

* AU heuristic shows improved results applying the priority rule.

* EDD heuristic is best for problem samples with small tardiness factor values.

* On average, the greedy heuristic gives best or second best schedules both before and after the priority rule. However, it is the heuristic with the largest complexity and computing times are largest.

* WSPT heuristic is one of the fastest heuristics, but it is also the weakest heuristic.

Huegler and Vasko (1997) presented a study about a performance comparison of heuristics for the total weighted tardiness problem. They compare quick and dirty heuristics (EDD, SWPT and AU), descent method with zero interchanges heuristic (DESO) and dynamic programming based heuristic (DPBH). According to their findings, AU is the best construction heuristic.

Rachamadugu (1987) developed a local precedence relationship among adjacent jobs in an optimal sequence for the weighted tardiness problem. This rule;

$$\frac{w_{[i]}}{p_{[i]}} * \left(1 - \frac{(d_{[i]} - t - p_{[i]})^{+}}{p_{[i+1]}}\right) \geq \frac{w_{[i+1]}}{p_{[i+1]}} * \left(1 - \frac{(d_{[i+1]} - t - p_{[i+1]})^{+}}{p_{[i]}}\right) \qquad (2.12)$$

In this expression, [i] represents the index of the job in the i^{th} position, x^+ denotes max(0,x) and t is the start time for $J_{[i]}$.

In addition to these heuristics, there are studies about the applications of meta-heuristics to total weighted tardiness problem similar to total tardiness problem.

Besten, Stützle and Dorigo (2000) developed an ant colony optimization system for single machine total weighted tardiness problem. They show that ACO performs significantly better than most other previously proposed for total weighted tardiness problem. According to their results, ACO always finds the best-known solutions for the 100-job samples, whereas Tabu search algorithm can find 103 of best-known solutions among 125 problems.

Madureira (1999) presented a study about meta-heuristics for the single machine scheduling total weighted tardiness problem. She compared the performances of Random Local Search (RNDLS) and Tabu search algorithms. Both algorithms give good solutions, but Tabu search method gives better solutions.

Similar to total tardiness problem, meta-heuristics are not used in our heuristic for total weighted tardiness problem because of their computational efforts.

CHAPTER 3

HEURISTIC PROCEDURE

In this chapter, all details of the heuristic which is developed in order to solve the single machine total tardiness and weighted tardiness problem are presented. There is no difference at heuristic procedure for these two problems. There are two main parts of the heuristic procedure. First one is the construction part and the second one is improvement part. All steps of the heuristic procedure are shown in Figure 3.1.

First of all, some construction heuristics for the single machine tardiness problem are solved. Then, according to their tardiness value, these construction heuristics are weighted. By using these weights, all jobs in the problem take a priority value and jobs are ordered according to these priority values. As a result of this, a mixed solution is obtained. This mixed solution is used as an initial sequence to the construction heuristic proposed in this study.

The construction heuristic is based on grouping the jobs, solving these groups and then fixing some particular number of jobs. There are two parameters should be determined in this model. The first one is the number of jobs which are selected and grouped from the end, which is denoted by B. The other one is the number of jobs which are fixed to schedule after solving the group, which is denoted by b. First of all, last B jobs of the initial solution are selected, and these jobs are grouped. Then, this job group is solved optimally. From this optimal solution, last b jobs are selected and fixed to the schedule. After this, again last B jobs which are not scheduled are grouped and solved. This process continues dynamically until there is no job to be scheduled. As a result of this, the solution of the construction heuristic is obtained.

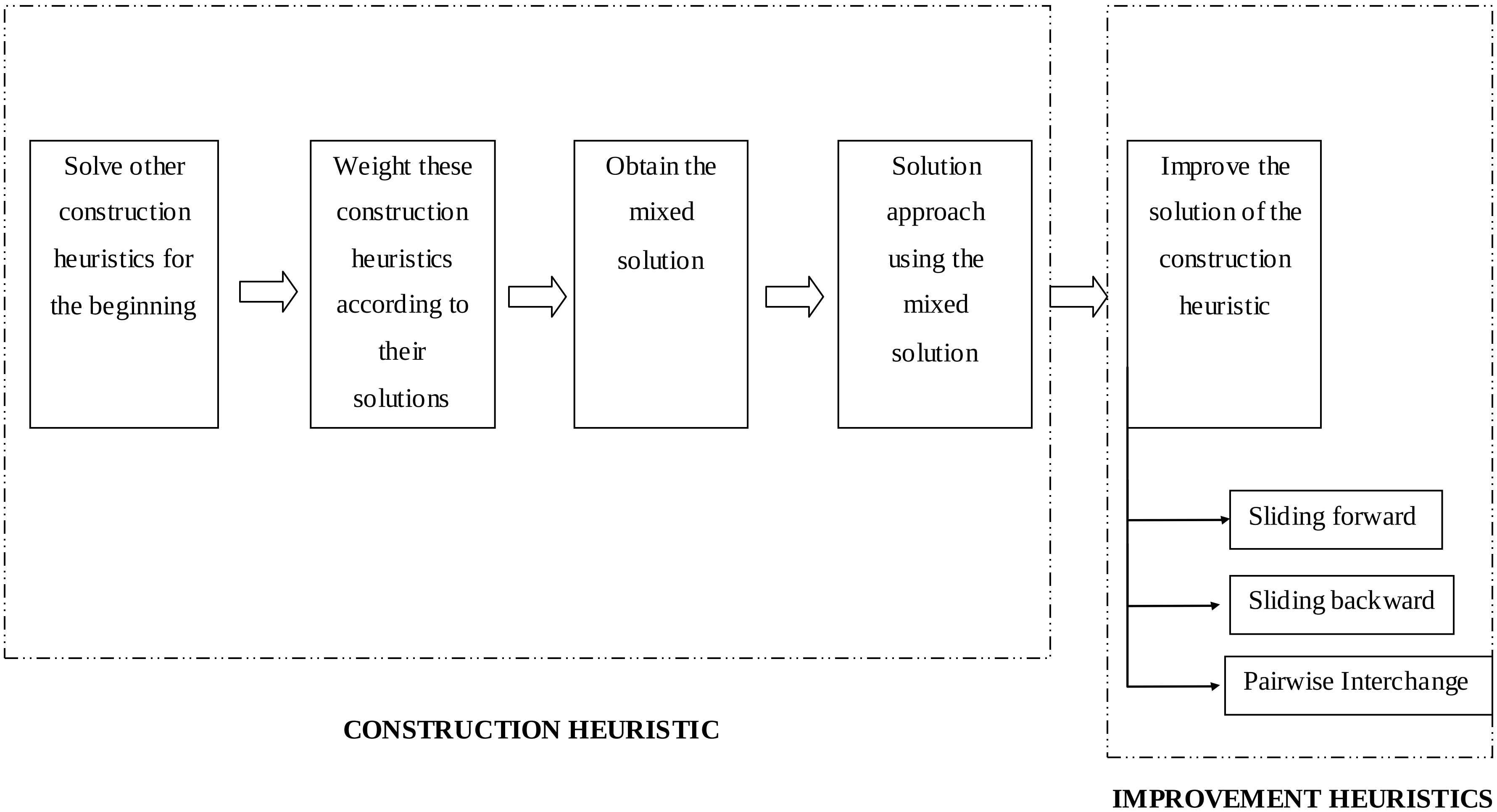

Figure 3.1 Steps of the heuristic

After the construction heuristic, this solution is improved by using some methods. The improvement heuristics used in this thesis are three types. They are sliding forward, sliding backward and pairwise interchange methods.

All details of these steps are explained at below sections.

3.1 CONSTRUCTION HEURISTIC

3.1.1 Other Construction Heuristics for the Beginning

3.1.1.1 Heuristics for total tardiness problem

At the beginning of the heuristic, some construction heuristics are used in order to obtain a good initial ordering of the jobs. These construction heuristics are selected taking into account their solution qualities and computational efforts. There are eight heuristics which are used for the beginning of the total tardiness problem.

* **Shortest Process Time (SPT) :** This heuristic sorts the jobs according to their process times in a non-decreasing order.

*** Earliest Due Date (EDD) :** This heuristic sorts the jobs according to their due dates in a non-decreasing order. EDD heuristic gives the optimal solution when the objective is to minimize maximum tardiness. Moreover, at total tardiness problem, if there is at most one tardy job at EDD sequence, it is the optimal solution.

*** Modified Due Date (MDD) :** This heuristic was developed by Baker and Bertrand (1982). It includes the dynamic implementation of EDD based on modified due dates.

$$MDD = \max(C + p_i, d_i) \tag{3.1}$$

In this formula, C represents the completion time of the last scheduled job. At each iteration, MDD value is computed for unscheduled jobs and the job which has smallest MDD value is scheduled. This process continues until there is no unscheduled job.

* **Apparent Urgency (AU) :** It was developed by Morton, Rachamadugu and Vepsalainen (1984). AU value of each job;

$$AU_i = (1/p_i) * \exp\{-\max[0, d_i - P(S) - p_i]/(k * \bar{p})\} \qquad (3.2)$$

In this formula, $\bar{p}$ is the average process time of all jobs and k is the so-called look-ahead parameter which is set according to the tightness of the due dates (*Koulamas, 1994*). If due dates of jobs are close together, a large k value should be used. On the other hand, k value should be small if the range of due dates is large. Finally, P(S) represents the total process time of jobs inside of the partial schedule S. ($P(S) = \sum_{i \in S} p_i$)

AU value is computed at each iteration, for all unscheduled jobs and then the job with highest AU value is scheduled to next position. Also, the scheduled job becomes a member of partial schedule S. These operations continue until there is no unscheduled job.

* **Panwalker, Smith and Koulamas (PSK) :** This heuristic was developed by Panwalker, Smith and Koulamas (1993). In this heuristic, the unscheduled jobs are kept in SPT order. If there are jobs with same process time, they are ordered according to EDD. The set of unscheduled jobs is named as U. Moreover, the set S includes the scheduled jobs in the order they are scheduled and C represents the completion time of last scheduled job in S. Steps of this heuristic are given briefly at "*Evaluation of leading heuristics for the single machine tardiness problem", (Russell and Holsenback , 1996)*:

Step 1: If U contains only one job, schedule it in the last position in S and go to step 9; else label the first job in U as the active job i

Step 2: If $C+p_i >= d_i$, go to step 8

Step 3: Select the next job in U as job j

Step 4: If $d_i <= C+p_j$, go to step 8

Step 5: If d_i<=d_j, go to step 7

Step 6: Job j now becomes the active job i. If this is the last job in U, go to step 8; else return to step 2

Step 7: If j is the last job in U, go to step 8; else return to step 3

Step 8: Remove job i from U and put it in the last position in S, increase C by p_i and return to step 1

Step 9: Calculate total tardiness for the sequence and terminate

*** <u>Cost Over Time (COVERT)</u> :** This heuristic is developed by Carroll (1965). In this heuristic, set S is the partial schedule with cardinality $|S|$ and B_i is the set of jobs which should precede job i in at least one optimal sequence according to Emmons' dominance conditions. Emmons' dominance conditions are explained at Appendix A. Let set E be the subset of the executable jobs such that $E = \{i : i \notin S, B_i \subseteq S\}$. For each job in E, priority index I_i , which estimates the probability that job i will be tardy if not scheduled next, is computed.

$$I_i = \begin{cases} 1, & \text{if } d_i \le P(S) + p_i \\ \dfrac{P(S \cup E) - d_i}{P(E) - p_i}, & \text{if } P(S) + p_i < d_i < P(S \cup E) \\ 0, & \text{if } P(S \cup E) \le d_i \end{cases}$$

$P(Q) = \sum_{i \in Q} p_i$ for any set of jobs Q. The job which has the highest I_i/p_i value is scheduled. These operations continue until all jobs are scheduled.

*** <u>Montagne</u> :** This heuristic is developed by Montagne (1969). Montagne heuristic orders the jobs in non-decreasing order of $p_j / (\sum_{i=1}^{n} p_i - d_j)$.

* **Hodgson's Algorithm :** Actually, Hodgson's Algorithm is used for the minimizing the number of tardy jobs in single machine tardiness problem. But this algorithm is also used in this thesis. Because, minimizing the number of tardy jobs can give the minimum total tardiness in some problems. Pinedo (2001) defines this algorithm. In this algorithm, E represents the set of early jobs and L represents the set of late jobs.

Step 1 : At the beginning, E contains all jobs and these jobs are ordered according to EDD. L does not have any job.

Step 2 : If no jobs in E are tardy, stop. Otherwise, identify the first tardy job in E. Let this job be the k^{th} job in E.

Step 3 : Identify the longest job among the first k jobs in sequence. Remove this job from E and place in L. Revise completion times and go to step 2.

3.1.1.2 Heuristics for total weighted tardiness problem

Similar to total tardiness problem, at total weighted tardiness problem, some well-known construction heuristics are chosen for the beginning. Solution qualities and computational efforts of the heuristics are considered while selecting them. There are six heuristics which are used in our heuristic for the weighted problem.

* **Shortest Weighted Process Time (SWPT) :** It is similar to shortest process time (SPT) heuristic. Only difference is the fact that SWPT sorts the jobs according to their (p_i/w_i) values in a non-decreasing order.

* **Earliest Due Date (EDD) :** It was explained at previous section.

* **Weighted Earliest Due Date (WEDD) :** This heuristic sorts the jobs according to their (d_i/w_i) values in a non-decreasing order.

* **Apparent Urgency (AU) :** This heuristic was explained at previous section, but there is a small difference for total weighted tardiness problem. AU value of each job for weighted problem;

$$AU_i = (w_i / p_i) * \exp\{-\max[0, d_i - P(S) - p_i]/(k * \bar{p})\} \qquad (3.3)$$

The rest of the heuristic is same with AU which is for total tardiness problem.

* **Montagne** : This heuristic was also explained at previous section. For the total weighted tardiness problem, this heuristic sorts the jobs in non-decreasing order of $p_j / w_j (\sum_{i=1}^{n} p_i - d_j)$.

* **The Greedy Heuristic** : This heuristic was developed by Fadlalla, Evans and Levy (1994) for total tardiness problem. In this thesis, the adaptation of this heuristic to total weighted tardiness problem is used. This adaptation is expressed briefly at *"Improved heuristics for the n-job single-machine weighted tardiness problem", (Volgenant and Teerhuis, 1998).*

M contains m_{ij} values which are;

$$m_{ij} = \begin{cases} w_i * \max\left(0, \sum_k p_k - d_i\right) + w_j * \max\left(0, \sum_k p_k - p_i - d_j\right), & \text{if } i \neq j \\ \infty, & \text{otherwise} \end{cases}$$

m_{ij} value represents the combined tardiness of jobs i and j with the pair (i,j) in the last position. This means that job i is the last job and job j is last but one job. These calculations are done for all combinations of n jobs.

After constructing matrix M, a binary matrix A is developed. A consists of a_{ij} values which are;

$$a_{ij} = \begin{cases} 1, & \text{if } m_{ij} \geq m_{ji} \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

$a_{ij}=0$ means that job j should be scheduled before job i.

Steps of the heuristic;

Step 0: Initialize k=n, L={1,2,…,n}, s(j)=0 and P(j)=∞ for $j \in L$

Step 1: Determine m_{ij} and a_{ij} for i, $j \in L$

Step 2: Compute P(i)= $\sum_{j \in L} a_{ij}$

Step 3: Select i for which P(i) = min{P(j); $j \in L$}

Step 4: Schedule job i, L=L\{i} and s(k)=i

Step 5: If k=1 stop; otherwise k = k-1 and go to step 1

3.1.2 Weighting the Heuristics

The construction heuristics, which are explained in previous section, have different characteristics from each other. When one of them gives the optimal result to total tardiness or weighted tardiness problem, the others can give results far from optimum. Therefore, there is need to weight these heuristics for each problem. Assigning a constant weight to each heuristic is not so meaningful. The reason of this is that solution quality of heuristics changes for all problems. That is why an algorithm is developed to assign weight to heuristics for each problem. The weight of each heuristic is determined as follows;

$$y_i = \left(\frac{1}{1 + \frac{x_i - x}{x}} \right)^{\alpha} \qquad (3.4)$$

In this formula, x represents the best tardiness value, which means the smallest value (because this is a minimization problem), among the tardiness values of construction heuristics and x_i is the tardiness value of the heuristic i. Moreover, α value means the degree of priority of the best heuristics. If α is high, weight range between the best heuristic and the other heuristics would be large. There are advantages and disadvantages of this situation. High α value increases the effect of the best heuristic and eliminates the effects of the heuristics which give bad results. However, there would be dominance of the best heuristic with a high α value and there could be some critical findings of the other heuristics. High α value disregards to other heuristics. As a result of this, some critical details could be passed over. On the other hand, low α value can easily detect some minor details from the solutions of other heuristics. However, in this situation, superfluous weights could be assigned to bad heuristics and this increases the effect of the bad heuristics to main heuristic. Therefore, α value should be adjusted by taking into account these situations.

α can take a value from 0 to ∞. This algorithm always assigns 1 to the best heuristic. If α is zero, all heuristics would weight equally and all of them are assigned 1. On the other hand, if α is ∞, all heuristics except the best one would be assigned 0 and there would be no effects of these heuristics. In order to determine the most appropriate α value, different α values were tried. This study is shown at Appendix B. As a result of this study, α=4 is determined the best α value. Therefore, all studies in this thesis are done by taking α as 4.

An example for total tardiness problem can be helpful in order to understand perfectly this weight assigning algorithm. For this, the example in Table 3.1 is used. This problem has 40 jobs to be scheduled to single machine. Process times and due dates of all jobs can be seen in Table 3.1.

Table 3.1 Sample problem

Job	p(i)	d(i)
1	21	418
2	7	529
3	19	460
4	37	388
5	16	360
6	33	467
7	4	465
8	14	397
9	18	363
10	5	409
11	32	330
12	40	426
13	1	399

Table 3.1 (Continued)

14	6	376
15	10	434
16	21	364
17	37	332
18	35	366
19	20	416
20	39	327
21	24	467
22	1	436
23	11	439
24	7	481
25	32	501
26	1	509
27	11	417
28	36	523
29	37	458
30	34	391
31	31	453
32	34	338
33	40	359
34	10	388
35	35	332
36	31	426
37	19	474
38	30	474
39	5	396
40	21	363

Total tardiness values of solutions of eight heuristics for total tardiness problem are shown in ascending order in Table 3.2.

Table 3.2 Total tardiness values of the heuristics for sample problem

AU	2683
MDD	2703
COVERT	2743
PSK	2938
MONTAGNE	3019
HODGSON	3135
SPT	3250
EDD	4477

In this problem, AU heuristic gives the best solution (2683). All weights will be determined according to this value. Calculated weights for all heuristics by taking α=4 are shown in Table 3.3.

Table 3.3 Weight of the heuristics for sample problem

AU	1
MDD	0,9707
COVERT	0,9153
PSK	0,6954
MONTAGNE	0,6237
HODGSON	0,5364
SPT	0,4644
EDD	0,1289

EDD solution is the worst solution among the heuristics and the difference between its solution and the best solution is too big. As a result of this, the weight assigned to EDD is the smallest (0.1289).

3.1.3 Obtaining the Mixed Solution

In this step of the heuristic, calculated weights for the other construction heuristics in previous section are used to compute the priority value of all the jobs. First of all, the weights of each position in each heuristic are calculated by using the heuristic weights. This calculation for N-job problem is done as follows for heuristic i:

1^{st} position: $1*y_i$

2^{nd} position: $2*y_i$

.

.

n^{th} position: $n*y_i$

.

.

N^{th} position: $N*y_i$

y_i values are the heuristic weights which are calculated in the previous section. These calculations are done for each heuristic separately. Calculated weights for each position in each heuristic for the problem which is started in the previous section are shown in Appendix C.

After this operation, job priorities are calculated by using these weights. For this, job sequences of all heuristics are needed (Appendix D shows the job sequences for all heuristics for the sample problem). By using these sequences and the weights, priorities for all jobs are calculated separately according to;

$$\text{Priority of job j} = \sum_i n_{ij*} y_i \tag{3.5}$$

In this formula, n_{ij} represents the position of job j in heuristic i and y_i represents the weight of heuristic i, which are calculated in Section 3.1.2. How these calculations are done is shown for one specific job (Job 1) from the sample problem in Table 3.4. Moreover, Table 3.5 shows the calculated priorities for all jobs in sample problem.

The job with small priority value means that that job should be scheduled earlier, it should not be scheduled to last positions. Therefore, if jobs are sorted according to their priority values in non-decreasing order, an efficient mixed solution would be obtained. The obtained mixed solution for the sample problem is shown in Table 3.6.

Table 3.4 Priority calculation of Job 1 for sample problem

	n_{i1}	y_i	Priority (n_{i1}*y_i)
MDD	30	0.9707	29.121
AU	30	1	30
PSK	31	0.6954	21.557
COVERT	30	0.9153	27.459
SPT	20	0.4644	9.288
EDD	22	0.1289	2.8358
HODGSON	15	0.5364	8.046
MONTAGNE	21	0.6237	13.098
TOTAL PRIORITY ($\sum_{i=1}^{8} n_{i1} * y_i$)			**141.6**

Table 3.5 Priorities of all jobs for sample problem

Job	Priority	Job	Priority	Job	Priority
1	141.6264	**15**	97.0140	**29**	205.3416
2	133.8751	**16**	64.7916	**30**	137.1909
3	127.8787	**17**	64.4070	**31**	165.8557
4	132.2416	**18**	96.3579	**32**	68.7505
5	40.1205	**19**	128.1978	**33**	85.3106
6	182.1531	**20**	60.0833	**34**	69.5620
7	103.6532	**21**	154.7095	**35**	66.1814
8	92.8701	**22**	81.3508	**36**	163.5177
9	45.4355	**23**	106.6743	**37**	136.8441
10	77.1734	**24**	119.1696	**38**	165.6701
11	35.8207	**25**	182.0884	**39**	68.8757
12	209.1452	**26**	116.7488	**40**	58.5446
13	50.5072	**27**	95.1606		
14	49.7704	**28**	202.4809		

Table 3.6 Initial solution for sample problem

Position	Job
1	11
2	5
3	9
4	14
5	13
6	40
7	20
8	17
9	16
10	35
11	32
12	39
13	34
14	10
15	22
16	33
17	8
18	27
19	18
20	15
21	7
22	23
23	26
24	24
25	3
26	19
27	4
28	2
29	37
30	30
31	1
32	21
33	36
34	38
35	31
36	25
37	6
38	28
39	29
40	12

3.1.4 Solution Approach

All applications made in previous three sections are done in order to find an initial mixed solution to heuristic. At the final part of the section 3.1.3, a mixed solution was obtained. Now this solution is used as an initial solution to heuristic.

Solution approach developed in this thesis is based on grouping a particular number of jobs which is denoted by B, solving these groups and fixing a particular number of jobs, which is denoted by b. Solution algorithm is started from the end of the initial solution. From the end, a particular number of jobs are selected and then, they are solved in GAMS. As a result of this, an optimum solution for that group is obtained. Based on this GAMS solution, a particular number of jobs (b) from the end of that optimum schedule are fixed and then, a new group is developed by taking jobs that are not fixed at previous iteration and the next jobs. These iterations continue dynamically until there is no job to be scheduled.

In order to determine B value, efficiency and computational time of the heuristics are considered. If B is too big, there would be much more alternative jobs to be fixed in that iteration. Therefore, solution quality of the heuristic would increase. However, in this situation, computation time would also increase. Much more time would be spent to solve each group and total computational time of all groups would be too high. On the other hand, if B is small, computational times of each iteration would be small. However, in this situation, there would be many iterations and therefore, total computational time of all groups would be high like the situation of big B value. There should be a balance point for B value. In this thesis, B=10 is used for the number of jobs to be grouped at each iteration. Moreover, a b value (number of jobs to be fixed at each iteration) should be determined. In this thesis, for b value, half value of B is taken. Therefore, b=5 is used for the number of the jobs to be fixed at each iteration.

Steps of the heuristic are expressed as follows for N-job problem;

***Step 1*:** There are three sets in the heuristic. The first one is set S which contains the scheduled jobs in the heuristic. Initially, there is no job in S, i.e. it is empty. The other set is I. Set I represents the jobs which will be solved in that iteration. Initially, set I consists of the jobs which are scheduled to last B=10 positions at the initial solution which is found at Section 3.1.3. This means that the jobs from the (N-9)th position to Nth position at the initial solution are inside of set I at the beginning. The final set is U which contains the jobs which are not inside of set S or set I. At the beginning, set U consists of the jobs which are scheduled to first (N-10) positions at the initial solution. There is one more parameter, H. H represents the total process time of the jobs which are inside of set U. This means that at the beginning, H equals to summation of process times of jobs which are scheduled to between 1st position and (N-10)th position at initial solution.

$$H = \sum_{i \in U} p_i \tag{3.6}$$

H is used as the beginning time of the problem for that iteration.

***Step 2*:** If the problem is total tardiness problem, solve the following MIP model (main structure, definitions of parameters, decision variables and constraints were given at Chapter 2):

Objective Function… Min $\sum_{j=1}^{B=10} t_j$ (3.7)

Constraints…

$$\sum_{j=1}^{10} x_{ij} = 1 \quad , \quad \forall i \in I \tag{3.8}$$

$$\sum_{i \in I} x_{ij} = 1 \quad , \quad \forall j \tag{3.9}$$

$$c_{j=1} = H + \sum_{i \in I} p_i * x_{i,j=1} \tag{3.10}$$

$$c_j = c_{j-1} + \sum_{i \in I} p_i * x_{ij} \quad , \forall j \neq 1 \tag{3.11}$$

$$t_j \geq c_j - \sum_{i \in I} d_i * x_{ij} \quad , \quad \forall j \tag{3.12}$$

$$x_{ij} \in \{0,1\} \quad , \quad \forall i \in I \text{ , } \forall j \qquad (3.13)$$

$$c_j \geq 0 \quad , \quad \forall j \qquad (3.14)$$

$$t_j \geq 0 \quad , \quad \forall j \qquad (3.15)$$

If this is weighted tardiness problem, solve the following MIP model:

Objective Function…

$$\text{Min} \quad \sum_{i \in I} \sum_{j=1}^{B=10} w_i * r_{ij} \qquad (3.16)$$

Constraints…

$$\sum_{j=1}^{10} x_{ij} = 1 \quad , \quad \forall i \in I \qquad (3.17)$$

$$\sum_{i \in I} x_{ij} = 1 \quad , \quad \forall j \qquad (3.18)$$

$$c_{j=1} = H + \sum_{i \in I} p_i * x_{i,j=1} \qquad (3.19)$$

$$c_j = c_{j-1} + \sum_{i \in I} p_i * x_{ij} \quad , \forall j \neq 1 \qquad (3.20)$$

$$t_j \geq c_j - \sum_{i \in I} d_i * x_{ij} \quad , \quad \forall j \qquad (3.21)$$

$$r_{ij} \geq t_j - M * (1 - x_{ij}) \text{ , } \quad \forall i \in I \text{ , } \forall j \quad (3.22)$$

$$r_{ij} \in \{0,1\} \quad , \quad \forall i \in I \text{ , } \forall j \qquad (3.23)$$

$$x_{ij} \in \{0,1\} \quad , \quad \forall i \in I \text{ , } \forall j \qquad (3.24)$$

$$c_j \geq 0 \quad , \quad \forall j \qquad (3.25)$$

$$t_j \geq 0 \quad , \quad \forall j \qquad (3.26)$$

***Step 3*:** If set U is empty, remove all jobs from set I and put into set S with same order of MIP solution in step 2 and STOP.

If set U is not empty, remove five jobs from I which are sequenced to last 5 positions at MIP solution in step 2 and put them into S to last five empty positions in order of MIP solution.

Remove five jobs from end of set U and put these five jobs to set I. Update H value according to new set U. Go to step 2.

Now, let solve the sample problem in order to understand exactly how this solution algorithm works.

Iteration 1: In this problem, N=40. At the beginning, set S is empty, set I consists of the last 10 jobs of the initial solution (Table 3.6) and set U consists of the first 30 jobs of the initial solution.

S = Ø

I = {1, 21, 36, 38, 31, 25, 6, 28, 29, 12}

U = {11, 5, 9, 14, 13, 40, 20, 17, 16, 35, 32, 39, 34, 10, 22, 33, 8, 27, 18, 15, 7, 23, 26, 24, 3, 19, 4, 2, 37, 30}

$$H = \sum_{i \in U} p_i \Longrightarrow \quad H = 550$$

MIP solution of this iteration and jobs which are scheduled at set S are shown in Table 3.7.

Table 3.7 a) MIP solution at iteration 1   b) scheduled job at set S at iteration 1

Position	**Job**
1	1
2	21
3	38
4	31
5	36
6	25
7	6
8	28
9	29
10	12

a

Position	**Job**
31	
32	
33	
34	
35	
36	25
37	6
38	28
39	29
40	12

b

Iteration 2: In this iteration, the jobs which are put into S are removed from set I and last five jobs of set U are removed from U and they are put into set S. Updated sets and H value;

S = {25, 6, 28, 29, 12}

I = {19, 4, 2, 37, 30, 1, 21, 38, 31, 36}

U = {11, 5, 9, 14, 13, 40, 20, 17, 16, 35, 32, 39, 34, 10, 22, 33, 8, 27, 18, 15, 7, 23, 26, 24, 3}

H = 433

MIP solution according to these sets and jobs which are scheduled at set S are shown in Table 3.8.

Table 3.8 a) MIP solution at iteration 2   b) scheduled job at set S at iteration 2

Position	**Job**
1	19
2	37
3	1
4	21
5	2
6	38
7	31
8	36
9	30
10	4

a

Position	**Job**
26	
27	
28	
29	
30	
31	38
32	31
33	36
34	30
35	4

b

Iteration 3: Updated sets and H value;

S = {38, 31, 36, 30, 4, 25, 6, 28, 29, 12}

I = {7, 23, 26, 24, 3, 19, 37, 1, 21, 2}

U = {11, 5, 9, 14, 13, 40, 20, 17, 16, 35, 32, 39, 34, 10, 22, 33, 8, 27, 18, 15}

H = 391

MIP solution according to these sets and jobs which are scheduled at set S are shown in Table 3.9.

Table 3.9 a) MIP solution at iteration 3 b) scheduled job at set S at iteration 3

Position	Job
1	19
2	1
3	23
4	3
5	7
6	24
7	37
8	26
9	21
10	2

a

Position	Job
21	
22	
23	
24	
25	
26	24
27	37
28	26
29	21
30	2

b

Iteration 4: Updated sets and H value;

S = {24, 37, 26, 21, 2, 38, 31, 36, 30, 4, 25, 6, 28, 29, 12}

I = {33, 8, 27, 18, 15, 19, 1, 23, 3, 7}

U = {11, 5, 9, 14, 13, 40, 20, 17, 16, 35, 32, 39, 34, 10, 22}

H = 281

MIP solution according to these sets and jobs which are scheduled at set S are shown in Table 3.10.

Table 3.10 a) MIP solution at iteration 4   b) scheduled job at set S at iteration 4

Position	Job
1	33
2	18
3	8
4	19
5	27
6	1
7	15
8	23
9	3
10	7

a

Position	Job
16	
17	
18	
19	
20	
21	1
22	15
23	23
24	3
25	7

b

Iteration 5: Updated sets and H value;

S = {1, 15, 23, 3, 7, 24, 37, 26, 21, 2, 38, 31, 36, 30, 4, 25, 6, 28, 29, 12}

I = {32, 39, 34, 10, 22, 33, 18, 8, 19, 27}

U = {11, 5, 9, 14, 13, 40, 20, 17, 16, 35}

H = 226

MIP solution according to these sets and jobs which are scheduled at set S are shown in Table 3.11.

Table 3.11 a) MIP solution at iteration 5 b) scheduled job at set S at iteration 5

Position	Job
1	32
2	39
3	10
4	34
5	18
6	33
7	19
8	22
9	8
10	27

a

Position	Job
11	
12	
13	
14	
15	
16	33
17	19
18	22
19	8
20	27

b

Iteration 6: Updated sets and H value;

S = {33, 19, 22, 8, 27, 1, 15, 23, 3, 7, 24, 37, 26, 21, 2, 38, 31, 36, 30, 4, 25, 6, 28, 29, 12}

I = {40, 20, 17, 16, 35, 32, 39, 10, 34, 18}

U = {11, 5, 9, 14, 13}

H = 73

MIP solution according to these sets and jobs which are scheduled at set S are shown in Table 3.12.

Table 3.12 a) MIP solution at iteration 6   b) scheduled job at set S at iteration 6

Position	**Job**
1	39
2	10
3	34
4	18
5	35
6	16
7	17
8	20
9	40
10	32

a

Position	**Job**
6	
7	
8	
9	
10	
11	16
12	17
13	20
14	40
15	32

b

Iteration 7: Updated sets and H value;

S = {16, 17, 20, 40, 32, 33, 19, 22, 8, 27, 1, 15, 23, 3, 7, 24, 37, 26, 21, 2, 38, 31, 36, 30, 4, 25, 6, 28, 29, 12}

I = {11, 5, 9, 14, 13, 39, 10, 34, 18, 35}

U = Ø

H = 0

Because U is empty, all jobs in I are put into S in order of MIP solution. This is shown in Table 3.13.

Table 3.13 a) MIP solution at iteration 7 b) scheduled job at set S at iteration 7

Position	Job
1	18
2	35
3	13
4	14
5	34
6	9
7	5
8	11
9	10
10	39

a

Position	Job
1	18
2	35
3	13
4	14
5	34
6	9
7	5
8	11
9	10
10	39

b

Iteration 7 was the final iteration, because there are no more jobs to be scheduled. All jobs are scheduled. Final solution is shown in Table 3.14.

Table 3.14 Final solution of the construction heuristic for sample problem

Position	Job	p(i)	d(i)
1	18	35	366
2	35	35	332
3	13	1	399
4	14	6	376
5	34	10	388
6	9	18	363
7	5	16	360
8	11	32	330
9	10	5	409
10	39	5	396
11	16	21	364
12	17	37	332
13	20	39	327
14	40	21	363

Table 3.14 (Continued)

15	32	34	338
16	33	40	359
17	19	20	416
18	22	1	436
19	8	14	397
20	27	11	417
21	1	21	418
22	15	10	434
23	23	11	439
24	3	19	460
25	7	4	465
26	24	7	481
27	37	19	474
28	26	1	509
29	21	24	467
30	2	7	529
31	38	30	474
32	31	31	453
33	36	31	426
34	30	34	391
35	4	37	388
36	25	32	501
37	6	33	467
38	28	36	523
39	29	37	458
40	12	40	426

Total tardiness value of this final sequence is 2613. Comparison of tardiness values of this construction heuristic with other beginning heuristics is shown in Table 3.15.

Table 3.15 Total tardiness values of our construction heuristic and other heuristics

CONS. HEU.	**2613**
AU	2683
MDD	2703
COVERT	2743
PSK	2938
MONTAGNE	3019
HODGSON	3135
SPT	3250
EDD	4477

This construction heuristic gives 2.6% ((2683-2613)/2683) better solution from the best heuristic (AU) for this sample problem.

The construction heuristic ends here. Next section explains the improvement heuristics.

3.2 IMPROVEMENT HEURISTICS

There are three improvement heuristic methods which are applied to solution of construction heuristic: Sliding forward, sliding backward, pairwise interchange.

3.2.1 Sliding Forward

Sliding forward method put jobs into forward positions without affecting so much the other jobs' positions. According to this algorithm, by considering that $a<b$, (a, b) means that the job at position a is scheduled to position b and the jobs between position a and position b are scheduled to one position back to their original positions. The positions of other jobs do not change. A small example about sliding

forward is shown in Table 3.16 for 5-job problem. At this example sliding forward of the job at position 1 to position 5 is shown.

Table 3.16 Example of sliding forward

1	2
2	3
3	4
4	5
5	1

This improvement method is applied to all possible options. For N-job problem, there are (N-1) + (N-2) + …. + 2 + 1 options. This expression equals to $\frac{N*(N-1)}{2}$. This is the number of sliding forward operations.

3.2.2 Sliding Backward

In this method, the opposite move of the sliding forward is done. Sliding backward method put jobs into backward positions without affecting so much the other jobs' positions. According to this algorithm, by considering that b>a, (b, a) means that the job at position b is scheduled to position a and the jobs between position b and position a are scheduled to one position forward to their original positions. The positions of other jobs do not change. A small example about sliding forward is shown in Table 3.17 for 5-job problem. At this example sliding backward of the job at position 5 to position 1 is shown.

Table 3.17 Example of sliding backward

1
2
3
4
5

5
1
2
3
4

Like the sliding forward method, sliding backward is applied to $\frac{N*(N-1)}{2}$ possible options.

3.2.3 Pairwise Interchange

Pairwise interchange method effects the positions of only two jobs on the contrary of sliding forward and backward. One job goes to a forward position, the job at that forward position goes to the empty backward position. According to this algorithm, by considering that a<b, (a, b) means that the job at position a is scheduled to position b and the job at position b is scheduled to position a. The positions of other jobs do not change. A small example about pairwise interchange is shown in Table 3.18 for 5-job problem. At this example pairwise interchange of the jobs at position 5 and position 1 is shown.

Table 3.18 Example of pairwise interchange

1
2
3
4
5

5
2
3
4
1

Like the other methods, there are $\frac{N*(N-1)}{2}$ possible options at pairwise interchange method and pairwise interchange method is applied to all these options.

3.2.4 Application of Improvement Heuristics

The improvement heuristics were defined at previous three sections. Now in this section, the applications of these three improvement heuristics are explained.

First of all, these methods are applied to all possible options. The operations which provide an improvement on the tardiness of the problem are collected. And then, the operation which gives the maximum improvement is selected and applied to the problem. Moreover, the other operations which give the improvement and independent from the previously selected operation are applied to the problem.

After all possible improvements are applied to the problem, improvement heuristics are applied to the new solution. This continues until there is no more improvement and that solution is accepted as the final solution.

Now, let apply these improvement heuristics to the sample problem. The solution of the construction heuristic was given in Table 3.14 and improvement heuristics are applied to this solution.

All operations which improve the tardiness value are shown at below. Sliding forward operations are shown in Table 3.19, sliding backward operations are shown in Table 3.20 and finally, pairwise interchange operations are shown in Table 3.21.

Table 3.19 Sliding forward operations which improve the construction heuristic's solution

No	First Position	Second Position	Improvement Value
1	18	21	1
2	18	22	1
3	35	36	5
4	35	37	9
5	35	38	10
6	35	39	10
7	35	40	7

Table 3.20 Sliding backward operations which improve the construction heuristic's solution

No	First Position	Second Position	Improvement Value
1	36	31	3
2	36	32	5
3	36	33	6
4	37	33	2
5	36	34	7
6	37	34	4
7	36	35	5
8	37	35	3

Table 3.21 Pairwise interchange operations which improve the construction heuristic's solution

No	First Position	Second Position	Improvement Value
1	16	34	9
2	34	36	4
3	35	36	5
4	34	37	3
5	35	37	8
6	35	38	3

Now, the operation which has maximum improvement value is selected. There are two sliding forward operations which decrease tardiness by 10. One of them is (35, 38) and (35, 39). Because they cannot be applied together, only one of them should be selected. Operation (35, 38) is selected to be applied to construction heuristic solution. From the other operations there are only three operations which are independent from the selected operation which is sliding forward (35, 38). These are sliding forward operations (18, 21), (18, 22) and pairwise interchange operation (16, 34). The pairwise interchange (16, 34) has the maximum improvement value (9) among these operations. Therefore, it is also selected to be applied to construction heuristic solution. There are no more operations which are independent from these two operations. These are shown briefly in Table 3.22.

Table 3.22 Operations applied to the construction heuristic's solution

No	Type	First Position	Second Position	Improvement Value
1	Sli. Forward	35	38	10
2	Pairwise Interchange	16	34	9

These improvement operations are applied to the construction heuristic solution. The revised solution is given in Table 3.23.

Table 3.23 Revised solution

Position	Job	p(I)	d(I)
1	18	35	366
2	35	35	332
3	13	1	399
4	14	6	376
5	34	10	388
6	9	18	363
7	5	16	360
8	11	32	330
9	10	5	409
10	39	5	396
11	16	21	364
12	17	37	332
13	20	39	327
14	40	21	363

Table 3.23 (Continued)

15	32	34	338
16	30	34	391
17	19	20	416
18	22	1	436
19	8	14	397
20	27	11	417
21	1	21	418
22	15	10	434
23	23	11	439
24	3	19	460
25	7	4	465
26	24	7	481
27	37	19	474
28	26	1	509
29	21	24	467
30	2	7	529
31	38	30	474
32	31	31	453
33	36	31	426
34	33	40	359
35	25	32	501
36	6	33	467
37	28	36	523
38	4	37	388
39	29	37	458
40	12	40	426

The new tardiness value is 2613 – (10 + 9) = 2594. Now all improvement operations are tried again on the new solution given in Table 3.23. All operations which improve the new tardiness value are shown at below. Sliding forward operations are shown in Table 3.24, sliding backward operations are shown in Table 3.25 and finally, pairwise interchange operations are shown in Table 3.26.

Table 3.24 Sliding forward operations which improve the revised solution

No	First Position	Second Position	Improvement Value
1	26	27	2
2	26	28	1
3	34	35	8
4	34	36	15
5	34	37	19
6	34	38	22
7	34	39	25
8	34	40	25

Table 3.25 Sliding backward operations which improve the revised solution

No	First Position	Second Position	Improvement Value
1	27	26	2
2	35	31	4
3	35	32	6
4	36	32	2
5	35	33	7
6	36	33	4
7	35	34	8
8	36	34	6

Table 3.26 Pairwise interchange operations which improve the revised solution

No	First Position	Second Position	Improvement Value
1	26	27	2
2	34	35	8
3	34	36	14
4	34	37	12
5	34	38	12
6	34	39	15

The maximum improvement (25) is given by two operations: sliding forward (34, 39) and (34, 40). Only one of them should be chosen and operation (34, 39) is selected to be applied. From the other operations there are four operations which are independent from the selected operation which is sliding forward (34, 39). These are sliding forward operations (26, 27), (26, 28), sliding backward operation (27, 26) and pairwise interchange operation (26, 27). Actually, all (26, 27) operations are same. Therefore it can be said that there are two alternatives. The (26, 27) operation gives better improvement (2) than the other (1). Therefore, it is also selected to be applied to revised solution. There are no more operations which are independent from these two operations. These are shown briefly in Table 3.27.

Table 3.27 Operations applied to the revised solution

No	Type	First Position	Second Position	Improvement Value
1	Sli. Forward	34	39	25
2	Sli. Forward	26	27	2

These improvement operations are applied to solution in Table 3.23. The new revised solution is given in Table 3.28.

Table 3.28 New revised solution

Position	Job	p(i)	d(i)
1	18	35	366
2	35	35	332
3	13	1	399
4	14	6	376
5	34	10	388
6	9	18	363
7	5	16	360
8	11	32	330
9	10	5	409
10	39	5	396
11	16	21	364
12	17	37	332
13	20	39	327
14	40	21	363
15	32	34	338
16	30	34	391
17	19	20	416
18	22	1	436
19	8	14	397
20	27	11	417
21	1	21	418
22	15	10	434
23	23	11	439
24	3	19	460
25	7	4	465
26	37	19	474
27	24	7	481
28	26	1	509
29	21	24	467
30	2	7	529
31	38	30	474
32	31	31	453
33	36	31	426
34	25	32	501
35	6	33	467
36	28	36	523
37	4	37	388
38	29	37	458
39	33	40	359
40	12	40	426

The new tardiness value obtained after these operations is 2594 – (25 + 2) = 2567. Again, all improvement operations are tried on this new solution given in Table 3.28. But no operation improves the tardiness value. Therefore, it can be said that this is the final solution.

Before, the tardiness value of the construction heuristic was compared with the solutions of the other heuristics. Now, the solution of the improvement heuristics compares with the other solutions. Additionally, the optimum solution of this sample problem is added to comparison in this step. When this sample problem is solved at GAMS, the optimum solution is given as 2567. The comparison all solutions are given in Table 3.29.

Table 3.29 Optimal value, solutions of our heuristics and other heuristics for sample problem

OPTIMUM	**2567**
CONS. + IMP. HEU.	**2567**
CONS. HEU.	2613
AU	2683
MDD	2703
COVERT	2743
PSK	2938
MONTAGNE	3019
HODGSON	3135
SPT	3250
EDD	4477

As it can be seen, at the end of the combination of the construction heuristic and the improvement heuristics, the optimum solution is found.

The heuristic ends here. Now, the results of the heuristic will be examined in the next chapter.

CHAPTER 4

COMPUTATIONAL RESULTS

4.1 COMPUTATIONAL RESULTS FOR THE SINGLE MACHINE TOTAL TARDINESS PROBLEM

4.1.1 Design of the Experiment

The construction heuristic and the improvement heuristics were developed in previous chapter. Now, the effectiveness of the heuristic for total tardiness problem will be evaluated in this section. For this purpose, some test problems are generated. For each problem, there are two generated parameters for each job:

* Process time

* Due date

Process time for each job is generated from a discrete uniform distribution between 1 and N which is the size of the problem.

$$p_i \sim U(1, N) \tag{4.1}$$

After generating the process times, the total process time (TP) is calculated in order to generate due date values.

$$TP = \sum_{i=1}^{N} p_i \tag{4.2}$$

After this, integer due date for each job is generated from the uniform distribution;

$$d_i \sim U[TP*(1-\tau-\frac{R}{2}),TP*(1-\tau+\frac{R}{2})] \quad (4.3)$$

In this formula, τ means the tardiness factor and R means the relative range of due dates of the problem. These values determine the hardness of the problem. High τ and R values increase the hardness of the problem.

Number of job (N), τ and R are taken three values in this thesis for total tardiness problem:

N = {20, 40, 50, 100}

τ = {0.25, 0.5, 0.75}

R = {0.25, 0.5, 0.75}

For each N value, there are nine sets which are developed from the combinations of the τ and R values and 10 sample problems are solved for each set. This means that 90 problems are tested for each problem size. At total, there are 360 problems to be tested.

For each N value, results are examined separately.

4.1.2 Results for N=20

The advantage of N=20 is the comparing the results with the optimal solution easily. The reason of this is that the problem is easily solved when the problem size equals 20. 10 problems from each 9 sets are solved at GAMS and the solutions of these problems are compared with the results of the heuristic. In Table 4.1, the detailed comparison of the heuristic with the other heuristics is given for each set.

Table 4.1 Detailed comparison of the heuristic for N=20

		MDD	**AU**	**PSK**	**COVERT**	**SPT**	**EDD**	**HODGSON**	**MONTAGNE**	**CONS. HEU.**	**CONS. + IMP.**
T	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	8	5	2	8	0	1	0	2	10	10
0.25	0.5	9	9	7	9	0	8	2	4	10	10
0.25	0.75	10	8	9	9	0	9	8	0	10	10
0.5	0.25	1	0	0	0	0	0	0	0	7	9
0.5	0.5	5	2	0	2	0	0	0	0	8	10
0.5	0.75	5	1	0	1	0	0	0	0	7	10
0.75	0.25	1	1	0	0	0	0	0	0	7	10
0.75	0.5	6	1	0	0	0	0	0	0	7	9
0.75	0.75	8	1	0	0	0	0	0	0	9	9
TOTAL		**53**	**28**	**18**	**29**	**0**	**18**	**10**	**6**	**75**	**87**

In Table 4.1, for each heuristic number of problems which that heuristic gives the optimal solution is given for each set. As it can be seen in Table 4.1, the optimal solution of 75 problems among 90 problems is found by only using the construction heuristic. When the improvement heuristics are used addition to construction heuristic, 12 more problems' optimal solution is found. At the rest of the three problems, optimal solution cannot be found. Moreover, at total, it can be said that MDD heuristic gives the best solutions among the other construction heuristics. MDD gives the optimum tardy value for 53 problems among 90 problems. Generally, it can be said that the heuristics give good result when tardy factor (τ) is small. But as τ increases, the efficiency of these heuristics decreases. On the other hand, our heuristic also gives good solutions at problems with big τ value.

In Table 4.2, the deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values are given for each set.

Table 4.2 Deviations from the optimal result for N=20

		CONS. HEU.	**CONS. + IMP HEU.**
T	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	0.000	0.000
0.25	0.5	0.000	0.000
0.25	0.75	0.000	0.000
0.5	0.25	0.835	0.053
0.5	0.5	0.890	0.000
0.5	0.75	0.700	0.000
0.75	0.25	0.129	0.000
0.75	0.5	0.143	0.029
0.75	0.75	0.278	0.278
Average		**0.331**	**0.040**

As it is said before, there are 3 problems which the construction heuristic + the improvement heuristics do not give the optimal result. By looking the Table 4.2, these 3 problems increase the average deviation from the optimal result to 0.04%. And the average deviation of the construction heuristic is 0.331% from the optimal results.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic is shown in Table 4.3.

Table 4.3 Improvement values for N=20

		CONS. HEU.	CONS. + IMP HEU.
T	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	1.211	1.211
0.25	0.5	1.538	1.538
0.25	0.75	0.000	0.000
0.5	0.25	2.571	3.311
0.5	0.5	0.751	1.593
0.5	0.75	1.155	1.809
0.75	0.25	0.598	0.724
0.75	0.5	0.216	0.330
0.75	0.75	0.010	0.010
Average		**0.894**	**1.170**

As it can be seen in Table 4.3, construction heuristic improves the best of the other construction heuristics by 0.894% by itself and construction heuristics + improvement heuristics improve by 1.17%.

At Appendix E, all results are shown for each problem.

4.1.3 Results for N=40

When problem size is 40, the problem becomes harder than N=20, but optimal solutions of many problem can be obtained by using GAMS in order to compare with the results of our heuristic. However, some problems cannot be solved when N=40. GAMS give an approximate solution, not exact optimum solution. At total tardiness problem, for N=40 and other problem sizes, the problem is run at most 1500 seconds (25 minutes) at GAMS. Because, it is observed that if GAMS cannot find the optimum solution until 25^{th} minute, generally GAMS cannot reach to exact optimum solution after this time. In this situation, the optimal value is assumed the minimum value of the approximate solution obtained from GAMS, the solutions of other construction heuristics and the solution of our heuristic. Table 4.4 shows the detailed results of the construction heuristic and the improvement heuristics for 90 problems with N=40.

As it can be seen in Table 4.4, the optimal solution of 49 problems among 90 problems is found by only using the construction heuristic. When the improvement heuristics are used addition to construction heuristic, 29 more problems' optimal solution is found. At total, optimal solution of 12 problems cannot be found. Furthermore, by looking Table 4.4, it can be said that the construction heuristic can give the optimal result by itself at problems with low τ value. When $\tau = 0.25$, the construction heuristic gives optimal result at 29 of 30 problems. However, at high τ values, the construction heuristic is not so efficient by itself. It gives optimal result at 20 of 60 problems. The improvement heuristics should be applied to these problems. When improvement heuristic is applied to the solution of the construction heuristic, 28 more of these 60 problems are solved optimally.

Table 4.4 Detailed comparison of the heuristic for N=40

		MDD	AU	PSK	COVERT	SPT	EDD	HODGSON	MONTAGNE	CONS. HEU.	CONS. + IMP.
τ	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	3	1	0	1	0	0	0	0	10	10
0.25	0.5	9	7	5	8	0	5	1	3	9	10
0.25	0.75	10	10	10	10	0	10	10	0	10	10
0.5	0.25	0	0	0	0	0	0	0	0	3	9
0.5	0.5	3	0	0	0	0	0	0	0	4	7
0.5	0.75	6	0	0	0	0	0	0	0	6	9
0.75	0.25	2	0	0	0	0	0	0	0	2	5
0.75	0.5	3	0	0	0	0	0	0	0	1	8
0.75	0.75	3	1	0	0	0	0	0	0	4	10
TOTAL		**39**	**19**	**15**	**19**	**0**	**15**	**11**	**3**	**49**	**78**

In Table 4.5, the deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values are given for each set for N=40.

Table 4.5 Deviations from the optimal result for N=40

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	0.000	0.000
0.25	0.5	2.400	0.000
0.25	0.75	0.000	0.000
0.5	0.25	1.917	0.040
0.5	0.5	0.823	0.102
0.5	0.75	0.856	0.019
0.75	0.25	1.183	0.113
0.75	0.5	1.179	0.046
0.75	0.75	0.156	0.000
Average		**0.946**	**0.036**

According to Table 4.5, the construction heuristic gives results 0.946% far from the optimal solution by itself. When the improvement heuristics are applied to construction heuristic, this deviation is decreased to 0.036%.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic is shown in Table 4.6.

Table 4.6 Improvement values for N=40

		CONS. HEU.	CONS. + IMP HEU.
T	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	3.564	3.564
0.25	0.5	0.313	2.188
0.25	0.75	0.000	0.000
0.5	0.25	0.918	2.673
0.5	0.5	1.136	1.812
0.5	0.75	0.404	1.213
0.75	0.25	0.467	1.501
0.75	0.5	0.010	1.115
0.75	0.75	0.481	0.635
Average		**0.810**	**1.633**

As it can be seen in Table 4.6, construction heuristic improves the best of the other construction heuristics by 0.81% by itself and construction heuristics + improvement heuristics improve by 1.633%.

At Appendix F, each problem's results are shown.

4.1.4 Results for N=50

When the problem size is 50, obtaining optimum results by using GAMS is more difficult than low size problems. Therefore, like some problems with N=40, when optimal solution cannot be obtained from GAMS, the optimal value is assumed the minimum value of the approximate solution obtained from GAMS, the solutions of other construction heuristics and the solution of our heuristic. Table 4.7 shows the detailed results of the construction heuristic and the improvement heuristics for 90 problems with N=50.

Table 4.7 Detailed comparison of the heuristic for N=50

		MDD	AU	PSK	COVERT	SPT	EDD	HODGSON	MONTAGNE	CONS. HEU.	CONS. + IMP.
τ	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	4	1	0	1	0	0	0	0	6	10
0.25	0.5	10	6	8	10	0	8	0	3	10	10
0.25	0.75	10	10	10	10	0	10	10	0	10	10
0.5	0.25	0	0	0	0	0	0	0	0	1	5
0.5	0.5	4	0	0	0	0	0	0	0	5	7
0.5	0.75	3	0	0	0	0	0	0	0	4	8
0.75	0.25	0	0	0	0	0	0	0	0	1	7
0.75	0.5	2	0	0	0	0	0	0	0	2	10
0.75	0.75	4	0	0	0	0	0	0	0	3	9
TOTAL		**37**	**17**	**18**	**21**	**0**	**18**	**10**	**3**	**42**	**76**

As it can be seen in Table 4.7, the optimal solution of 42 problems among 90 problems is found by only using the construction heuristic. When the improvement heuristics are used addition to construction heuristic, 34 more problems' optimal solution is found.

Same situation for N=40 is also valid for the problems with N=50. At low τ value, construction heuristic gives good results by itself, but at problems with high τ value, the construction heuristic should be improved by improvement heuristics.

In Table 4.8, the deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values are given for each set for N=50.

Table 4.8 Deviations from the optimal result for N=50

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	2.145	0.000
0.25	0.5	0.000	0.000
0.25	0.75	0.000	0.000
0.5	0.25	2.225	0.256
0.5	0.5	0.754	0.098
0.5	0.75	1.792	0.183
0.75	0.25	2.633	0.043
0.75	0.5	0.726	0.000
0.75	0.75	0.292	0.017
Average		**1.174**	**0.066**

As it can be seen from the Table 4.8, the construction heuristic gives results 1.174% far from the optimal solution by itself. When the improvement heuristics are applied to construction heuristic, this deviation is decreased to 0.066%.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic for N=50 is shown in Table 4.9.

Table 4.9 Improvement values for N=50

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	3.478	5.401
0.25	0.5	0.000	0.000
0.25	0.75	0.000	0.000
0.5	0.25	0.333	2.223
0.5	0.5	0.501	1.135
0.5	0.75	-0.121	1.428
0.75	0.25	0.015	2.521
0.75	0.5	-0.179	0.541
0.75	0.75	-0.162	0.113
Average		**0.429**	**1.485**

According to Table 4.9, in some sets the construction heuristic does not improve the best solution of the other construction heuristics. There is a negative improvement value in three sets. These sets have τ and R values. However, when improvement heuristics are applied to the solution of the construction heuristic in these sets, positive improvement values are occurred. Moreover at average, construction heuristic improves the best of the other construction heuristics by 0.429% by itself and construction heuristics + improvement heuristics improve by 1.485%.

All results of each problem are shown at Appendix G.

4.1.5 Results for N=100

The problems with N=100 are so hard problems. Therefore, it is too difficult to obtain optimal result of the problems by solving these problems at GAMS except the problems with low τ and R values. As a result of this situation, for many problems optimal value is assumed the minimum value of the approximate solution obtained from GAMS, the solutions of other construction heuristics and the solution of our heuristic. In Table 4.10, the detailed comparison of the heuristic with the other heuristics is given for N=100.

According to Table 4.10, the optimal solution of 22 problems among 90 problems is found by only using the construction heuristic. When the improvement heuristics are used addition to construction heuristic, 40 more problems' optimal solution is found. At total, optimal solution of 28 problems cannot be found.

Moreover, Table 4.10 shows that MDD heuristic gives better results than the construction heuristic. At general, MDD finds optimal solution of 28 problems among 90, whereas the construction heuristic finds optimal solution of 22 problems. Therefore, it can be said that the efficiency of the construction heuristic by itself is not so high for the problems with big size. But when improvement heuristics are applied to the construction heuristic's solution, 40 more problems are optimally solved.

Table 4.10 Detailed comparison of the heuristic for N=100

		MDD	**AU**	**PSK**	**COVERT**	**SPT**	**EDD**	**HODGSON**	**MONTAGNE**	**CONS. HEU.**	**CONS. + IMP.**
T	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	1	0	0	0	0	0	0	0	0	3
0.25	0.5	9	3	8	9	0	8	0	3	10	10
0.25	0.75	10	10	10	10	0	10	10	0	10	10
0.5	0.25	0	0	0	0	0	0	0	0	0	1
0.5	0.5	0	0	0	0	0	0	0	0	0	9
0.5	0.75	1	0	0	0	0	0	0	0	2	9
0.75	0.25	0	0	0	0	0	0	0	0	0	6
0.75	0.5	2	0	0	0	0	0	0	0	0	8
0.75	0.75	5	0	0	0	0	0	0	0	0	6
TOTAL		**28**	**13**	**18**	**19**	**0**	**18**	**10**	**3**	**22**	**62**

In Table 4.11, the deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values are given for each set for N=100.

Table 4.11 Deviations from the optimal result for N=100

		CONS. HEU.	CONS. + IMP HEU.
τ	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	1.768	0.847
0.25	0.5	0.000	0.000
0.25	0.75	0.000	0.000
0.5	0.25	3.107	0.464
0.5	0.5	2.656	0.009
0.5	0.75	1.950	0.094
0.75	0.25	3.129	0.090
0.75	0.5	0.516	0.011
0.75	0.75	0.261	0.028
Average		**1.488**	**0.171**

As it can be seen from the Table 4.11, the construction heuristic gives results 1.488% far from the optimal solution by itself. When the improvement heuristics are applied to construction heuristic, this deviation is decreased to 0.171%.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic for N=100 is shown in Table 4.12.

Table 4.12 Improvement values for N=100

		CONS. HEU.	CONS. + IMP HEU.
T	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	0,823	1,712
0.25	0.5	0,000	0,000
0.25	0.75	0,000	0,000
0.5	0.25	0,190	2,734
0.5	0.5	-0,132	2,435
0.5	0.75	-0,026	1,779
0.75	0.25	-0,961	2,011
0.75	0.5	-0,322	0,182
0.75	0.75	-0,202	0,031
Average		**-0,070**	**1,209**

According to Table 4.12, the construction heuristic has a negative improvement value at average for the problems with N=100. This is normal, because as explained before MDD heuristic gives better results than the construction heuristic for N=100. Therefore, when the results of the construction heuristic are compared with the best solution of the other construction heuristics (naturally, MDD is the best among these heuristics for many problems here), the construction heuristic's results are higher. Therefore, there is no improvement at average. However, when improvement heuristics are applied to the construction heuristic's solution, it improves the best of the other construction heuristics by 1.209%.

At Appendix H, all results of each problem are shown for N=100.

All results were given for four problem sizes in this section. The summary of the results is given in Table 4.13.

Table 4.13 Summary of the results (total tardiness problem)

	# of optimum solution among 90 problems		Ave. Deviation from optimum result (%)		Max. Deviation from optimum result (%)		Improvement of the best of the other cons. heu. (%)	
N	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**
20	75	87	0.331	0.040	4.814	2.784	0.894	1.170
40	49	78	0.946	0.036	24.000	0.618	0.810	1.633
50	42	76	1.174	0.066	8.591	1.390	0.429	1.485
100	22	62	1.488	0.171	5.522	3.102	-0.070	1.209

Finally, average computational times of our heuristic for each problem size are given in Table 4.14.

Table 4.14 Computational times (total tardiness problem)

N	Comp. Time for Cons. Heu. (min)	Comp. Time for Imp. Heu. (min)	Total Comp. Time (min)
20	03:22	01:19	04:41
40	06:51	02:22	09:13
50	08:33	03:21	11:54
100	17:58	05:47	23:45

4.2 COMPUTATIONAL RESULTS FOR THE SINGLE MACHINE TOTAL WEIGHTED TARDINESS PROBLEM

4.2.1 Design of the Experiment

In order to evaluate our heuristic for the single machine total weighted tardiness problem, problems which are generated for the single machine total tardiness problem are used. Only weight values are added to each problem. Weight for each job is generated from a discrete uniform distribution between 1 and 10.

$$w_i \sim U(1,10) \qquad (4.4)$$

The heuristic is tested for the single machine total weighted tardiness problem for problem sizes 20 and 40. Different from the total tardiness problem, the heuristic is not tested for problem sizes 50 and 100. Because, at these numbers of jobs, complexity of the problem is too high and therefore, reaching optimal solutions in order to compare with our heuristic's solution is too hard by using GAMS or any other solver programs. Therefore, a healthy evaluation for these numbers of jobs is impossible.

4.2.2 Results for N=20

Different from the total tardiness problem, solving total weighted tardiness problem for N=20 is not so easy. Optimal solutions for each problem can be obtained by using GAMS, and then these solutions were compared with the heuristic solution at total tardiness problem for problem size equals 20. But at total weighted tardiness problem, optimal solutions of many problems with high τ value cannot be reached. Therefore, similar to total tardiness problem, optimal value is assumed the minimum value of the approximate solution obtained from GAMS, the solutions of other construction heuristics and the solution of our heuristic at these problems. At total weighted tardiness problem the problem is run at most 1800 seconds (30 minutes) at GAMS. Because, it is observed that GAMS generally cannot reach to exact optimum solution, unless GAMS finds the optimum solution until 30^{th} minute. Table 4.15 shows the detailed comparison of the construction heuristic and the improvement heuristics with the other heuristics for 90 problems.

Table 4.15 Detailed comparison of the heuristic for N=20 (weighted tardiness problem)

		AU	**Greedy**	**EDD**	**WEDD**	**SWPT**	**Montagne**	**CONS. HEU.**	**CONS. + IMP. HEU.**
τ	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	0	5	0	0	0	0	10	10
0.25	0.5	2	6	5	0	0	1	10	10
0.25	0.75	5	8	8	0	0	0	9	9
0.5	0.25	1	1	0	0	0	0	6	8
0.5	0.5	0	2	0	0	0	0	6	7
0.5	0.75	0	1	0	0	0	0	6	7
0.75	0.25	0	1	0	0	0	0	6	8
0.75	0.5	0	2	0	0	0	0	7	7
0.75	0.75	0	1	0	0	0	0	8	10
TOTAL		**8**	**27**	**13**	**0**	**0**	**1**	**68**	**76**

By looking to Table 4.15, it can be seen that the optimal solutions of 68 problems among 90 problems are found by only using the construction heuristic. Moreover, optimal solutions of 8 more problems are acquired by adding improvement heuristics to construction heuristic. Moreover, according to Table 4.15, Greedy heuristic gives best results among the other construction heuristics. It finds 27 optimal solutions. Greedy heuristic works better at problems with τ=0.25 than problems with high τ values. Our heuristic gives good results at all sets.

Table 4.16 shows the deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values for each set for N=20.

Table 4.16 Deviations from the optimal result for N=20 (weighted tardiness problem)

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	0.000	0.000
0.25	0.5	0.000	0.000
0.25	0.75	0.606	0.606
0.5	0.25	0.995	0.218
0.5	0.5	2.635	2.449
0.5	0.75	2.594	2.456
0.75	0.25	0.911	0.177
0.75	0.5	0.313	0.313
0.75	0.75	0.129	0.000
Average		**0.909**	**0.691**

As it can be seen from the Table 4.16, the construction heuristic gives results 0.909% far from the optimal solution by itself. When the improvement heuristics are applied to construction heuristic, this deviation is decreased to 0.691%.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic for N=20 can be seen in Table 4.17.

Table 4.17 Improvement values for N=20 (weighted tardiness problem)

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	4.368	4.368
0.25	0.5	7.000	7.000
0.25	0.75	2.000	2.000
0.5	0.25	3.060	3.821
0.5	0.5	2.260	2.432
0.5	0.75	9.809	9.942
0.75	0.25	0.420	1.141
0.75	0.5	0.743	0.743
0.75	0.75	2.981	3.102
Average		**3.627**	**3.839**

Table 4.17 shows that construction heuristic improves the best of the other construction heuristics by 3.627% by itself and construction heuristics + improvement heuristics improve by 3.839%.

At Appendix I, each problem's results are shown.

4.2.3 Results for N=40

Table 4.18 shows the detailed results for the total weighted tardiness problem when problem size equals 40.

Table 4.18 Detailed comparison of the heuristic for N=40 (weighted tardiness problem)

		AU	**Greedy**	**EDD**	**WEDD**	**SWPT**	**Montagne**	**CONS. HEU.**	**CONS. + IMP. HEU.**
τ	**R**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**	**opt.**
0.25	0.25	0	0	0	0	0	0	4	7
0.25	0.5	0	7	2	0	0	0	9	9
0.25	0.75	10	10	10	0	0	0	10	10
0.5	0.25	0	0	0	0	0	0	1	9
0.5	0.5	0	0	0	0	0	0	2	8
0.5	0.75	0	0	0	0	0	0	4	8
0.75	0.25	1	0	0	0	0	0	0	6
0.75	0.5	1	0	0	0	0	0	3	9
0.75	0.75	0	3	0	0	0	0	5	10
TOTAL		**12**	**20**	**12**	**0**	**0**	**0**	**38**	**76**

According to Table 4.18, the construction heuristic finds optimal solution of 38 problems among 90 problems itself. This is approximately half of the findings of the construction heuristic for N=20, which is 68. The efficiency of the construction heuristic itself decreases as the problem size increases. On the other hand, when improvement heuristics are applied to the solution of the construction heuristic, 38 more problems' optimal solutions are found. There are totally 14 problems which our heuristic cannot find the optimal result. This number is same with N=20. Therefore, it can be said that when our construction heuristic and improvement heuristics works together, they are not affected so much from the problem size. But of course, this is valid only these numbers of problem size. Furthermore, Table 4.18 shows that the other construction heuristics do not give so good results. The best heuristic is Greedy heuristic, similar to N=20. It finds optimal solutions of 20 problems among 90. Our heuristic works good at all sets. The worst results are at problems with τ=0.75 and R=0.25. At that set, our heuristic finds 6 optimal solutions among 10.

The deviations of the construction heuristic and improvement heuristics' results from the optimum tardy values for each set for N=40 are shown in Table 4.19.

Table 4.19 Deviations from the optimal result for N=40 (weighted tardiness problem)

		CONS. HEU.	**CONS. + IMP HEU.**
τ	**R**	**dev. (%)**	**dev. (%)**
0.25	0.25	9.428	2.399
0.25	0.5	4.405	3.036
0.25	0.75	0.000	0.000
0.5	0.25	5.462	0.125
0.5	0.5	5.690	0.121
0.5	0.75	5.475	1.335
0.75	0.25	1.215	0.327
0.75	0.5	0.953	0.004
0.75	0.75	0.257	0.000
Average		**3.654**	**0.816**

Table 4.19 shows that the construction heuristic gives results 3.654% far from the optimal solution by itself. When the improvement heuristics are applied to construction heuristic, this deviation is decreased to 0.816%.

The improvement of the tardy values of the best of the other construction heuristics by our heuristic for N=40 can be seen in Table 4.20.

Table 4.20 Improvement values for N=40 (weighted tardiness problem)

		CONS. HEU.	**CONS. + IMP HEU.**
T	**R**	**imp. (%)**	**imp. (%)**
0.25	0.25	6.761	12.352
0.25	0.5	6.246	7.234
0.25	0.75	0.000	0.000
0.5	0.25	2.865	7.633
0.5	0.5	5.227	10.043
0.5	0.75	6.857	10.348
0.75	0.25	0.591	1.455
0.75	0.5	0.222	1.157
0.75	0.75	0.705	0.957
Average		**3.275**	**5.687**

Table 4.20 shows that construction heuristic improves the best of the other construction heuristics by 3.275% by itself and construction heuristics + improvement heuristics improve by 5.687%.

At Appendix J, all results of each problem are shown for weighted tardiness problem with N=40.

All results were given for two problem sizes at total weighted tardiness problem in this section. The summary of the results is given in Table 4.21.

Table 4.21 Summary of the results (weighted tardiness problem)

	# of optimum solution among 90 problems		**Ave. Deviation from optimum result (%)**		**Max. Deviation from optimum result (%)**		**Improvement of the best of the other cons. heu. (%)**	
N	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**	**Cons. Heu.**	**Cons. + Imp. Heu.**
20	68	76	0.909	0.691	14.452	14.452	3.627	3.839
40	38	76	3.654	0.816	44.047	30.357	3.275	5.687

Finally, average computational times of our heuristic for total weighted tardiness problem and each problem size are given in Table 4.22.

Table 4.22 Computational times (weighted tardiness problem)

N	**Comp. Time for Cons. Heu. (min)**	**Comp. Time for Imp. Heu. (min)**	**Total Comp. Time (min)**
20	03:34	01:14	04:48
40	07:20	02:58	10:18

CHAPTER 5

CONCLUSION

In this section, main conclusions of this thesis and possible extensions for future works are explained.

In this thesis, single machine total tardiness and weighted tardiness problems are studied. First of all, general properties and assumptions of the single machine tardiness problem are discussed. Because both single machine total tardiness problem and weighted tardiness problem are NP-hard problems, heuristic solution procedures are used. In order to solve these scheduling problems, a heuristic is proposed in this thesis.

The heuristic has same procedure for both total tardiness and weighted tardiness problems. Initially, some simple, well-known construction heuristics are solved and an initial schedule is obtained by combining the solutions of these heuristics. After this, a part of this solution is solved optimally by using a solver (GAMS). For this, initially last 10 jobs from the schedule are taken, and the optimal sequence of these jobs are found by taking the starting time as the summation of process times of the previous jobs. Then, from the optimal solution of these 10 jobs, last 5 jobs are taken and they are fixed. After this operation, the next 10 jobs which are not fixed 5 jobs from previous solution and last 5 jobs from the initial schedule which are not solved at previous sub-problem are taken, these jobs are solved optimally and again last 5 jobs of this optimal sequence are fixed. These operations continue dynamically until there is no job not scheduled. This part is the construction part of our heuristic. Also some improvement heuristics are proposed to get better the solution of the construction heuristic. These improvement heuristic methods are sliding forward, sliding backward and pairwise interchange methods. At sliding forward method, a job is placed to a next position in the schedule and the jobs between the previous position and the present position of that job are scheduled to one position down. Sliding backward method is opposite of the sliding forward method. At this method, a selected job is placed to a previous position in the schedule and the jobs between the previous position and the present position of that job are scheduled to one position up. Finally, at pairwise interchange method, positions of selected two jobs are changed. These methods are applied to all jobs. The operation which gives maximum improvement at tardiness value is selected and this is applied to the solution. Then, all methods are applied to revised solution again. This continues until there is no more improvement. These steps were defined step by step and also a sample problem was used as a numerical example.

The heuristic is tested for several problem sizes (N), tardy factors (τ) and due date ranges (R). For total tardiness problem, the heuristic is tested for N=20, 40, 50 and 100. But for weighted tardiness problem, N=20 and 40 are used. The reason of this is the fact that for large size problems, finding optimal solution or a close solution to optimal is too difficult. In order to compare our heuristic's solution, optimal solution of all problem or if finding exact optimum solution is not possible, near optimal solution of that problem are used. At total tardiness problem, when only our construction heuristic is run, from 90 problems, 75 problems' optimal solution is found for N=20; 49 problems' optimal solution is found for N=40; 42 problems' optimal solution is found for N=50; 22 problems' optimal solution is found for N=100. When improvement heuristics are run with the construction heuristic, 87 problems' optimal solution is found for N=20; 78 problems' optimal solution is found for N=40; 76 problems' optimal solution is found for N=50; 62 problems' optimal solution is found for N=100 at total tardiness problem. As it can be seen, construction heuristic is not so efficient for large size problems itself. However, when improvement and construction heuristic are run together, it also finds good results for large size problems. Moreover, the efficiency of the construction heuristic decreases when tardy factor increases. But when improvement and construction heuristic are run together, the heuristic is more robust to changes in tardy factor value. At weighted tardiness problem, our construction heuristic itself finds 68 problems' optimal solution for N=20 and 38 problem's optimal solution for N=40 among 90 problems. When improvement and construction heuristic are performed together, it finds 76 problems' optimal solution for both N=20 and N=40. The performance of the construction heuristic itself slightly decreases when problem size is increased from 20 to 40. However, when improvement and construction heuristic are run together, the performance does not change. Finally, it can be said that when our heuristic is compared with other heuristics, it gives much better results than the others.

Finally, the study in this thesis can be extended with several ways for future works. For example, in this thesis when a particular number of jobs (B) are grouped from the initial solution, this value is taken as 10. Also when a particular number of jobs (b) are fixed from the sub-solution, 5 is taken for b. At future works different combinations for B and b values can be tried. Moreover, some optimal algorithms such as dynamic programming and branch and bound methods can be used in order to solve groups optimally. Sometimes, finding optimal solution with GAMS takes a lot of time. Another extension can be the application of Emmons' Dominance Properties to the problem before solving it. This can provide a better start for the heuristic. Furthermore, this heuristic can be adapted to different scheduling problems such as single machine problem with jobs with different release times, single machine problem with jobs which have precedence constraints, one-stage parallel machines problem, flow shop problem or job shop problem at future researches.

REFERENCES

Baker, K. R., and Bertrand, J. W. M., (1982), "A Dynamic Priority Rule for Scheduling Against Due-Dates", Journal of Operations Management, 3, 37-42.

Baker, K. R., (1995), "Elements of Sequencing and Scheduling", Wiley, New York.

Bauer, A., Bullnheimer, B., Hartl, R. F., and Strauss, C., (1999), "Minimizing Total Tardiness on a Single Machine Using Ant Colony Optimization", Central European Journal, Issue: 5/99.

Carroll, D. C., (1965), "Heuristic Sequencing of Single and Multiple Components", Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Mass.

Conway, R. W., Maxwell, W. L., and Miller, L. W., (1967), "Theory of Scheduling", Addison-Wesley, Reading Mass.

Den Besten, M., Stützle, T., and Dorigo, M., (2000), "Ant Colony Optimization for the Total Weighted Tardiness Problem", In Proceedings of the Parallel Problem Solving from Nature Conference.

Du, J. and Leung J. Y. - T., (1990), "Minimizing Total Tardiness on One Machine is NP-Hard.", Mathematics of Operations Research 15, 483-495.

Emmons, H., (1969), "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness", Operations Research, 17, 701-715.

Fadlalla, A., Evans, J. R., and Levy, M. S., (1994), "A greedy heuristic for the mean tardiness sequencing problem", Computers & Operations Research, 21:329-36.

Fry, T. D., Vicens, L., Macleod, K., and Fernandez, S., (1989), "A Heuristic Solution Procedure to Minimize $\bar{T}$ on a Single Machine", Journal of the Operational Research Society, 40, 293-297.

Huegler, P. A., and Vasko, F. J., (1997), "A Performance Comparison of Heuristics for the Total Weighted Tardiness Problem", Computers ind. Engng, Vol. 32, No. 4, 753-767.

Koulamas, C., (1994), "The Total Tardiness Problem: Review and Extensions", Operations Research, Vol. 42, No. 6, 1025-1041.

Laguna, M., Barnes, J. W., and Glover, F. W., (1990), "Tabu search methods for a single machine scheduling problem", Journal of Intelligent Manufacturing, 2, 63-74.

Lawler, E. L., (1977), "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness", Annals of Discrete Mathematics, 1, 331-342.

Lenstra, J. K., Rinooy Khan, A. H. G. and Brucker, P., (1977), "Complexity of machine scheduling problems", Annals of Discrete Mathematics, 1, 343-363.

Madureira, A. M., (1999), "Meta-Heuristics for the Single-Machine Scheduling Total Weighted Tardiness Problem", Proceedings of the 1999 IEEE, International Symposium on Assembly and Task Planning, Porto-Portugal, 405-410.

Montagne, E. R., (1969), "Sequencing with Time Delay Costs", Industrial Engineering Research Bulletin, Arizona State University, Tucson.

Morton, T. E., Rachamadugu, R. M., and Vepsalainen, A., (1984), "Accurate Myopic Heuristics for Tardiness Scheduling", Working Paper, Carnegie Mellon University, Pittsburgh, Penn.

Nyirenda, J. C., (2001), "Relationship Between the Modified Due Date Rule and the Heuristic of Wilkerson and Irwin", Orion, Vol. 17, No. 1/2, 101-111.

Panwalker, S. S., Smith, M. L., and Koulamas, C. P., (1993), "A Heuristic for the Single Machine Tardiness Problem", European Journal of Operation Research, 70, 304-310.

Pinedo, M., (2001), "Scheduling: Theory, Algorithms, and Systems", Prentice Hall, New York.

Potts, C. N., and Van Wassenhove, L. N., (1982), "A Decomposition Algorithm for the Single Machine Total Tardiness Problem", Operations Research Letters 32, 177-181.

Potts, C. N., and Van Wassenhove, L. N., (1985), "A Branch and Bound Algorithm for the Total Weighted Tardiness Problem", Operations Research 33, 363-377.

Rachamadugu, R. V., (1987), "A note on the weighted tardiness problem", Operations Research, Vol. 35, No. 3, 450-452.

Russell, R. M., and Holsenback, J. E., (1997), "Evaluation of leading heuristics for the single machine tardiness problem", European Journal of Operation Research, 96, 538-545.

Volgenant, A., and Teerhuis, E., (1998), "Improved heuristics for the n-job single-machine weighted tardiness problem", Computers & Operations Research, 26, 35-44.

APPENDIX A

EMMONS' DOMINANCE CONDITIONS

Emmons (1969) developed dominance conditions for the single machine tardiness problem. These conditions establish the relative order in which pairs of jobs are processed in an optimal schedule. At below two theorems of Emmons are given.

Some notations used by Emmons;

B_i = set of jobs sequenced before job i (J_i)

A_i = set of jobs sequenced after J_i

$$A_i' = \{J_i : i \notin A_i\}$$

Emmons' First Theorem:

For any two jobs J_i and J_j, if;

a) $p_i \leq p_j$

b) $d_i \leq \max(\sum_{k \in B_j} p_k + p_j, d_j)$,

then J_i precedes J_j ($J_i \leftarrow J_j$) in at least one optimal schedule. This means that $i \in B_j$ and $j \in A_i$. Emmons' first theorem gives us conditions under which a shorter job can be said to precede a longer one.

Emmons' Second Theorem:

For any two jobs J_i and J_j, if;

a) $p_i \leq p_j$

b) $d_i > \max(\sum_{k \in B_j} p_k + p_j, d_j)$

c) $d_i + p_i \geq \sum_{k \in A_j^{'}} p_k$,

then J_j precedes J_i ($J_j \leftarrow J_i$) in at least one optimal schedule. This means that $j \in B_i$ and $i \in A_j$. Emmons' second theorem gives necessary conditions for a longer job to precede a shorter one in an optimal schedule.

APPENDIX B

DETERMINING ALFA VALUE

A weight formula (Equation 3.4) is mentioned for the construction heuristics in section 3.1.2. According to this formula, the best heuristic which gives the smallest tardiness value takes the highest weight. α value in this formula determines the range between the maximum weight and minimum weight. When $\alpha = 0$, all construction heuristics take same weight, 1. On the other hand, when $\alpha = \infty$, the best heuristic takes value of 1 and all the other heuristics take value of 0. In order to determine the best α value, different α values are tried on the total tardiness problems with 20 jobs. This number of jobs is selected, because the problems are solved easier than the other problems with higher number of jobs.

For the total tardiness problems with 20 jobs different α values which are 0.5, 1, 2, 3, 4 and 5 are tried. Too big α values are not considered, because the solution depends more on only one heuristic when α increases. Therefore, the biggest α value tried is 5. These α values are tried on 90 sample problems with 20 jobs. Table B.1 shows the results for these α values. In Table B.1, number of problems which that α value gives the lowest tardiness value among all α values is given.

Table B.1 Number of problems which that α value gives the best result among all α values

A					
0.5	**1**	**2**	**3**	**4**	**5**
84	87	87	89	90	89

According to Table B.1, for all 90 problems, results of our heuristic with α = 4 gives best solution among the other solutions of our heuristic with different α values. Actually, other α values do not give bad solutions but α = 4 seems the best. Therefore, in this thesis, for our heuristic α = 4 is used.

APPENDIX C

WEIGHTS FOR EACH POSITION IN EACH HEURISTIC FOR SAMPLE PROBLEM

Table C.1 Weights for each position in each heuristic for sample problem

Position	MDD	AU	PSK	COVERT	SPT	EDD	HODGSON	MONTAGNE
1	0.9707	1	0.6955	0.9153	0.4645	0.1290	0.5365	0.6238
2	1.9415	2	1.3909	1.8307	0.9289	0.2580	1.0729	1.2476
3	2.9122	3	2.0864	2.7460	1.3934	0.3870	1.6094	1.8713
4	3.8829	4	2.7819	3.6613	1.8578	0.5159	2.1458	2.4951
5	4.8537	5	3.4773	4.5767	2.3223	0.6449	2.6823	3.1189
6	5.8244	6	4.1728	5.4920	2.7868	0.7739	3.2187	3.7427
7	6.7951	7	4.8683	6.4073	3.2512	0.9029	3.7552	4.3664
8	7.7658	8	5.5637	7.3227	3.7157	1.0319	4.2916	4.9902
9	8.7366	9	6.2592	8.2380	4.1801	1.1609	4.8281	5.6140
10	9.7073	10	6.9547	9.1533	4.6446	1.2898	5.3645	6.2378
11	10.6780	11	7.6501	10.0687	5.1091	1.4188	5.9010	6.8616
12	11.6488	12	8.3456	10.9840	5.5735	1.5478	6.4374	7.4853
13	12.6195	13	9.0411	11.8993	6.0380	1.6768	6.9739	8.1091
14	13.5902	14	9.7365	12.8147	6.5025	1.8058	7.5104	8.7329
15	14.5610	15	10.4320	13.7300	6.9669	1.9348	8.0468	9.3567
16	15.5317	16	11.1274	14.6453	7.4314	2.0637	8.5833	9.9805
17	16.5024	17	11.8229	15.5607	7.8958	2.1927	9.1197	10.6042
18	17.4731	18	12.5184	16.4760	8.3603	2.3217	9.6562	11.2280
19	18.4439	19	13.2138	17.3913	8.8248	2.4507	10.1926	11.8518
20	19.4146	20	13.9093	18.3067	9.2892	2.5797	10.7291	12.4756
21	20.3853	21	14.6048	19.2220	9.7537	2.7087	11.2655	13.0993
22	21.3561	22	15.3002	20.1373	10.2181	2.8376	11.8020	13.7231
23	22.3268	23	15.9957	21.0527	10.6826	2.9666	12.3384	14.3469
24	23.2975	24	16.6912	21.9680	11.1471	3.0956	12.8749	14.9707
25	24.2683	25	17.3866	22.8833	11.6115	3.2246	13.4114	15.5945
26	25.2390	26	18.0821	23.7987	12.0760	3.3536	13.9478	16.2182

Table C.1 (Continued)

27	26.2097	27	18.7776	24.7140	12.5404	3.4826	14.4843	16.8420
28	27.1804	28	19.4730	25.6293	13.0049	3.6115	15.0207	17.4658
29	28.1512	29	20.1685	26.5447	13.4694	3.7405	15.5572	18.0896
30	29.1219	30	20.8640	27.4600	13.9338	3.8695	16.0936	18.7133
31	30.0926	31	21.5594	28.3753	14.3983	3.9985	16.6301	19.3371
32	31.0634	32	22.2549	29.2907	14.8628	4.1275	17.1665	19.9609
33	32.0341	33	22.9504	30.2060	15.3272	4.2565	17.7030	20.5847
34	33.0048	34	23.6458	31.1213	15.7917	4.3854	18.2394	21.2085
35	33.9756	35	24.3413	32.0367	16.2561	4.5144	18.7759	21.8322
36	34.9463	36	25.0368	32.9520	16.7206	4.6434	19.3123	22.4560
37	35.9170	37	25.7322	33.8673	17.1851	4.7724	19.8488	23.0798
38	36.8877	38	26.4277	34.7827	17.6495	4.9014	20.3853	23.7036
39	37.8585	39	27.1232	35.6980	18.1140	5.0304	20.9217	24.3274
40	38.8292	40	27.8186	36.6133	18.5784	5.1593	21.4582	24.9511

APPENDIX D

JOB SEQUENCES OF THE HEURISTIC FOR SAMPLE PROBLEM

Table D.1 Job sequences of the heuristics for sample problem

Position	MDD	AU	PSK	COVERT	SPT	EDD	HODGSON	MONTAGNE
1	20	20	20	5	13	20	11	13
2	11	11	11	9	22	11	5	22
3	17	17	35	11	26	17	9	26
4	35	35	17	20	7	35	40	7
5	32	13	32	17	10	32	16	39
6	33	32	33	32	39	33	14	10
7	5	14	5	35	14	5	34	14
8	9	33	9	33	2	9	30	24
9	40	5	40	40	24	40	39	2
10	16	9	16	13	15	16	8	34
11	18	40	18	14	34	18	13	15
12	14	16	14	16	23	14	10	27
13	4	18	34	18	27	4	19	23
14	34	39	4	4	8	34	27	8
15	39	22	30	34	5	30	1	5
16	13	34	13	39	9	39	36	9
17	8	10	10	10	3	8	15	40
18	10	30	39	22	37	13	22	16
19	27	8	15	27	19	10	23	19
20	15	27	22	8	1	19	31	3
21	22	15	23	15	16	27	3	1
22	23	23	27	7	40	1	7	37
23	19	7	8	23	21	12	6	11
24	7	26	7	26	38	36	21	21
25	24	3	24	24	31	15	37	32

Table D.1 (Continued)

26	3	24	3	3	36	22	38	35
27	37	37	37	37	11	23	24	17
28	26	19	26	2	25	31	25	18
29	2	2	2	19	6	29	26	36
30	1	1	19	1	30	3	2	30
31	21	21	1	21	32	7	20	20
32	38	38	21	38	18	6	17	31
33	31	31	38	36	35	21	35	38
34	36	36	31	31	28	37	32	4
35	25	25	36	25	4	38	33	33
36	6	6	25	6	17	24	18	6
37	30	28	6	30	29	25	4	25
38	28	4	28	28	20	26	12	29
39	29	29	29	29	12	28	29	12
40	12	12	12	12	33	2	28	28

APPENDIX E

EXPERIMENTS FOR TOTAL TARDINESS PROBLEM (N=20)

Table E.1 Experiments for total tardiness problem (N=20)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Mdd	Au	Psk	Cov-ert	Spt	Edd	Hod-Gson	Mon-tag-ne
1	0,25	0,25	51	51	51	53	54	53	54	153	53	124	56
2	0,25	0,25	55	55	55	55	55	55	55	117	55	132	55
3	0,25	0,25	39	39	39	39	39	39	39	56	41	54	39
4	0,25	0,25	56	56	56	56	56	67	56	106	71	112	65
5	0,25	0,25	43	43	43	43	46	60	43	93	60	89	53
6	0,25	0,25	51	51	51	51	51	55	51	111	68	114	69
7	0,25	0,25	49	49	49	49	67	58	49	171	58	138	83
8	0,25	0,25	58	58	58	58	60	62	58	115	90	65	86
9	0,25	0,25	60	60	60	60	60	74	60	126	80	89	76
10	0,25	0,25	44	44	44	56	49	56	48	105	81	86	51
11	0,25	0,5	3	3	3	3	3	3	3	315	3	127	3
12	0,25	0,5	2	2	2	2	2	2	2	101	2	2	2
13	0,25	0,5	18	18	18	18	18	32	18	151	18	18	19
14	0,25	0,5	4	4	4	4	4	4	4	119	4	48	4
15	0,25	0,5	13	13	13	13	13	15	13	127	15	36	34
16	0,25	0,5	9	9	9	9	9	9	9	227	9	128	15
17	0,25	0,5	8	8	8	8	8	8	8	161	8	92	12
18	0,25	0,5	2	2	2	2	2	2	2	105	2	28	2
19	0,25	0,5	22	22	22	26	26	26	26	132	26	101	41
20	0,25	0,5	6	6	6	6	6	6	6	79	6	22	8
21	0,25	0,75	0	0	0	0	0	0	0	155	0	0	23
22	0,25	0,75	40	40	40	40	41	41	46	327	48	145	76
23	0,25	0,75	0	0	0	0	0	0	0	169	0	0	114
24	0,25	0,75	0	0	0	0	0	0	0	156	0	0	27
25	0,25	0,75	0	0	0	0	0	0	0	164	0	0	23
26	0,25	0,75	0	0	0	0	0	0	0	42	0	0	21
27	0,25	0,75	3	3	3	3	4	3	3	325	3	86	109
28	0,25	0,75	0	0	0	0	0	0	0	274	0	0	36
29	0,25	0,75	0	0	0	0	0	0	0	100	0	0	3
30	0,25	0,75	0	0	0	0	0	0	0	93	0	0	40
31	0,5	0,25	252	252	252	252	254	300	254	311	497	293	303
32	0,5	0,25	393	393	393	401	418	450	404	507	704	466	468
33	0,5	0,25	348	359	348	418	409	426	374	428	607	409	428
34	0,5	0,25	275	275	275	320	325	337	335	315	519	307	309
35	0,5	0,25	404	404	404	406	407	428	407	493	553	451	470
36	0,5	0,25	236	247	236	247	253	267	249	281	364	291	266
37	0,5	0,25	345	345	345	348	348	384	346	400	410	387	381
38	0,5	0,25	228	228	228	233	229	262	231	278	348	270	255
39	0,5	0,25	379	381	381	382	383	410	389	466	513	453	421
40	0,5	0,25	319	319	319	346	346	347	346	371	408	386	359
41	0,5	0,5	298	298	298	301	301	344	302	495	396	335	402

Table E.1 (Continued)

42	0,5	0,5	165	165	165	187	189	194	172	276	261	208	219
43	0,5	0,5	317	317	317	317	340	356	330	434	557	401	359
44	0,5	0,5	281	281	281	281	284	306	294	415	450	354	352
45	0,5	0,5	223	223	223	223	224	287	237	389	425	298	333
46	0,5	0,5	206	206	206	206	206	242	206	407	303	307	328
47	0,5	0,5	242	242	242	242	242	281	263	373	375	269	318
48	0,5	0,5	113	113	113	119	115	119	113	226	179	177	196
49	0,5	0,5	294	306	294	313	313	354	346	500	476	406	412
50	0,5	0,5	457	479	457	480	483	499	493	686	779	546	619
51	0,5	0,75	66	66	66	66	71	71	66	273	71	199	113
52	0,5	0,75	313	319	313	320	331	328	340	484	514	395	409
53	0,5	0,75	186	186	186	186	208	245	199	340	333	220	254
54	0,5	0,75	71	71	71	71	77	87	72	245	101	204	121
55	0,5	0,75	125	125	125	134	143	145	126	402	176	210	224
56	0,5	0,75	228	228	228	228	232	279	232	447	313	250	340
57	0,5	0,75	113	113	113	113	113	150	114	324	172	185	261
58	0,5	0,75	87	87	87	91	94	102	94	443	122	219	169
59	0,5	0,75	418	422	418	422	454	483	426	628	693	524	552
60	0,5	0,75	194	202	194	215	215	238	228	433	321	317	354
61	0,75	0,25	844	844	844	868	845	899	1013	894	1233	922	896
62	0,75	0,25	534	534	534	538	541	589	547	575	892	558	569
63	0,75	0,25	772	777	772	801	801	904	818	808	1448	882	809
64	0,75	0,25	620	620	620	625	630	715	666	640	866	714	649
65	0,75	0,25	559	559	559	584	562	648	595	623	1153	646	617
66	0,75	0,25	801	802	801	807	891	926	941	845	1325	940	851
67	0,75	0,25	771	775	771	775	783	796	790	811	1049	808	821
68	0,75	0,25	799	799	799	801	799	813	804	895	1250	948	873
69	0,75	0,25	599	599	599	625	600	631	627	655	892	694	654
70	0,75	0,25	841	841	841	841	849	901	928	922	1465	951	925
71	0,75	0,5	699	701	701	701	703	729	712	828	1159	802	810
72	0,75	0,5	506	506	506	506	506	534	511	612	768	592	613
73	0,75	0,5	634	634	634	634	717	737	733	730	941	732	675
74	0,75	0,5	940	940	940	940	944	965	962	1050	1163	1092	1037
75	0,75	0,5	734	734	734	734	735	793	747	826	1397	848	795
76	0,75	0,5	958	968	958	959	959	1012	1041	1074	1663	1097	1102
77	0,75	0,5	991	992	991	1011	1011	1026	1026	1164	1353	1175	1179
78	0,75	0,5	653	653	653	653	654	691	659	786	1220	742	788
79	0,75	0,5	650	650	650	658	682	699	668	737	1114	728	751
80	0,75	0,5	555	555	555	555	558	573	558	634	933	617	591
81	0,75	0,75	431	443	443	443	448	465	453	539	705	797	535
82	0,75	0,75	742	742	742	742	743	750	772	894	1100	1039	857
83	0,75	0,75	791	791	791	791	791	824	841	971	1136	1009	887
84	0,75	0,75	948	948	948	948	949	966	958	1004	1458	1136	1000
85	0,75	0,75	478	478	478	478	479	529	547	680	733	640	614
86	0,75	0,75	1321	1321	1321	1321	1323	1323	1330	1457	1531	1404	1436
87	0,75	0,75	779	779	779	779	780	796	891	871	1128	1076	877
88	0,75	0,75	419	419	419	419	423	465	437	509	666	491	473
89	0,75	0,75	823	823	823	823	825	853	833	1004	1187	1575	898
90	0,75	0,75	1005	1005	1005	1006	1006	1010	1014	1189	1402	1314	1160

APPENDIX F

EXPERIMENTS FOR TOTAL TARDINESS PROBLEM (N=40)

Table F.1 Experiments for total tardiness problem (N=40)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Mdd	Au	Psk	Cov-ert	Spt	Edd	Hod-Gson	Mon-tag-ne
1	0,25	0,25	208	208	208	242	242	242	233	392	304	390	275
2	0,25	0,25	407	407	407	416	428	425	454	860	709	749	517
3	0,25	0,25	256	256	256	263	270	350	270	398	369	344	330
4	0,25	0,25	260	260	260	260	267	264	265	779	330	617	394
5	0,25	0,25	218	218	218	218	218	222	218	507	426	396	264
6	0,25	0,25	233	233	233	233	252	281	237	685	359	415	358
7	0,25	0,25	228	228	228	249	240	251	253	595	403	466	298
8	0,25	0,25	245	245	245	260	246	267	261	507	399	289	286
9	0,25	0,25	322	322	322	370	370	372	370	792	392	536	401
10	0,25	0,25	173	173	173	176	176	186	176	398	217	285	183
11	0,25	0,5	1	1	1	1	1	1	1	783	1	192	1
12	0,25	0,5	10	10	10	10	10	10	10	1377	10	422	10
13	0,25	0,5	0	0	0	0	0	0	0	1342	0	0	0
14	0,25	0,5	56	56	56	56	64	61	64	877	61	113	129
15	0,25	0,5	66	66	66	66	66	76	66	1035	91	373	96
16	0,25	0,5	8	8	8	8	8	8	8	1217	8	395	12
17	0,25	0,5	63	63	63	63	63	70	63	788	70	414	69
18	0,25	0,5	100	124	100	128	135	156	133	800	226	229	184
19	0,25	0,5	39	39	39	39	39	70	39	770	76	292	47
20	0,25	0,5	9	9	9	9	19	9	9	626	9	329	19
21	0,25	0,75	0	0	0	0	0	0	0	1514	0	0	98
22	0,25	0,75	0	0	0	0	0	0	0	1682	0	0	1692
23	0,25	0,75	0	0	0	0	0	0	0	1170	0	0	250
24	0,25	0,75	0	0	0	0	0	0	0	255	0	0	9
25	0,25	0,75	0	0	0	0	0	0	0	2115	0	0	92
26	0,25	0,75	0	0	0	0	0	0	0	395	0	0	397
27	0,25	0,75	0	0	0	0	0	0	0	545	0	0	308
28	0,25	0,75	0	0	0	0	0	0	0	1316	0	0	760
29	0,25	0,75	0	0	0	0	0	0	0	1549	0	0	327
30	0,25	0,75	0	0	0	0	0	0	0	783	0	0	214
31	0,5	0,25	2728	2739	2739	2786	2787	2829	2777	3257	3761	3134	3144
32	0,5	0,25	1766	1807	1766	1809	1824	1873	1832	2089	3091	2020	2109
33	0,5	0,25	2009	2009	2009	2033	2015	2086	2045	2611	3112	2593	2248
34	0,5	0,25	1622	1629	1622	1689	1711	1839	1678	2045	3294	1939	2019
35	0,5	0,25	2083	2180	2083	2218	2239	2251	2239	2567	3269	2546	2392
36	0,5	0,25	2372	2372	2372	2399	2384	2399	2434	3114	2954	3002	2833
37	0,5	0,25	1197	1310	1197	1310	1338	1383	1369	1541	2776	1450	1467
38	0,5	0,25	2125	2125	2125	2235	2150	2277	2189	2668	3009	2371	2671
39	0,5	0,25	2567	2613	2567	2703	2641	2938	2743	3250	4477	3135	3019
40	0,5	0,25	2341	2344	2341	2396	2345	2449	2359	3030	3450	2838	2651

Table F.1 (Continued)

41	0,5	0,5	1830	1834	1830	1898	1903	1916	1896	2995	2983	2443	2564
42	0,5	0,5	1405	1405	1405	1405	1425	1438	1409	2993	1985	2059	1794
43	0,5	0,5	1539	1631	1539	1648	1649	1650	1689	2794	2186	2182	2334
44	0,5	0,5	1283	1283	1283	1283	1373	1415	1329	2468	2468	1659	2055
45	0,5	0,5	1577	1593	1577	1663	1666	1707	1684	3223	2587	2597	2477
46	0,5	0,5	1329	1330	1330	1345	1428	1496	1438	2102	2633	1724	1962
47	0,5	0,5	1276	1282	1282	1282	1285	1354	1331	2546	2009	1889	1997
48	0,5	0,5	1256	1262	1262	1267	1281	1394	1328	2431	2164	1854	1853
49	0,5	0,5	1688	1688	1688	1711	1719	1809	1784	3180	3134	1965	2382
50	0,5	0,5	1435	1435	1435	1435	1460	1534	1488	2485	2037	2176	1916
51	0,5	0,75	**924**	924	924	924	983	999	963	3709	1388	1836	1909
52	0,5	0,75	353	353	353	353	386	426	396	1743	664	708	662
53	0,5	0,75	706	718	706	719	745	786	744	2865	1151	1242	1538
54	0,5	0,75	**901**	897	897	897	1009	1095	939	3645	1423	1903	1846
55	0,5	0,75	**1987**	1985	1980	2028	2040	2189	2040	4245	2753	2893	2816
56	0,5	0,75	**1864**	1937	1864	1927	1929	2030	2061	3800	3614	2183	2875
57	0,5	0,75	**1076**	1105	1078	1142	1150	1177	1131	3169	1607	1947	2115
58	0,5	0,75	826	826	826	826	828	893	877	2155	1254	1018	1136
59	0,5	0,75	546	546	546	546	615	659	552	2075	893	938	1197
60	0,5	0,75	**474**	474	474	474	482	503	475	2020	596	1135	924
61	0,75	0,25	4635	4691	4637	4639	4681	4766	4743	5085	8610	5085	4939
62	0,75	0,25	4579	4579	4579	4579	4603	4623	4717	4971	6639	5018	4953
63	0,75	0,25	6515	6542	6540	6683	6612	6709	6592	7064	9338	7155	7007
64	0,75	0,25	3440	3440	3440	3440	3457	3672	3531	3924	5052	3770	3862
65	0,75	0,25	6067	6158	6071	6358	6213	6470	6259	6546	10137	6692	6586
66	0,75	0,25	6428	6721	6428	6826	6833	7142	6816	6918	10657	7110	6917
67	0,75	0,25	5192	5259	5192	5442	5248	5450	5312	5602	7922	6004	5545
68	0,75	0,25	5354	5362	5354	5441	5400	5466	5415	5848	8697	5885	5794
69	0,75	0,25	5893	6003	5894	6065	6094	6450	6131	6316	8949	6482	6283
70	0,75	0,25	6626	6682	6667	6766	6838	6883	6825	7134	10883	7105	7142
71	0,75	0,5	**5389**	5493	5382	5590	5593	5661	5564	6388	9214	5976	6400
72	0,75	0,5	**5905**	5900	5894	5933	5936	6287	6301	6676	8534	6587	6643
73	0,75	0,5	**5133**	5140	5133	5150	5242	5338	5378	6287	8698	5835	5995
74	0,75	0,5	**5144**	5143	5143	5153	5164	5251	5448	6120	9260	5727	6077
75	0,75	0,5	**5278**	5512	5292	5535	5539	5580	5682	6352	9135	6332	5971
76	0,75	0,5	**4644**	4668	4644	4644	4652	5066	5086	5469	7820	5604	5186
77	0,75	0,5	**4461**	4475	4461	4565	4570	4756	4548	4997	7723	5289	4998
78	0,75	0,5	**5573**	5622	5570	5559	5739	5848	5925	6718	9948	6058	6601
79	0,75	0,5	**4382**	4480	4368	4368	4616	4656	4680	5387	9027	5060	5033
80	0,75	0,5	**5734**	5735	5705	5739	5747	6028	6239	6696	9249	6265	6472
81	0,75	0,75	**6245**	6220	6207	6276	6280	6345	6944	7557	10117	10626	7181
82	0,75	0,75	**3390**	3363	3353	3468	3472	3551	4969	4578	6636	4714	4322
83	0,75	0,75	**5318**	5304	5304	5306	5306	5355	5726	6376	8354	7142	6048
84	0,75	0,75	**4486**	4476	4472	4529	4539	4630	4904	5575	7484	5460	5233
85	0,75	0,75	**5340**	5323	5321	5321	5351	5518	5756	6471	8465	7090	6046
86	0,75	0,75	**5396**	5432	5386	5386	5414	5517	5501	6660	8448	7413	6257
87	0,75	0,75	**2945**	2921	2919	2934	2982	3166	3095	3745	5918	3964	3584
88	0,75	0,75	**5806**	5799	5799	5803	5831	5918	5869	7439	8639	8172	6474
89	0,75	0,75	**5066**	5034	5034	5034	5034	5216	5268	6693	7605	7385	6023
90	0,75	0,75	**5155**	5132	5132	5135	5161	5219	5530	6213	7909	8724	5874

* Values written bold in the column of optimal value are not exact optimum.

APPENDIX G

EXPERIMENTS FOR TOTAL TARDINESS PROBLEM (N=50)

Table G.1 Experiments for total tardiness problem (N=50)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Mdd	Au	Psk	Cov-ert	Spt	Edd	Hod-Gson	Mon-tag-ne
1	0,25	0,25	641	676	641	708	757	860	754	1257	1634	1012	905
2	0,25	0,25	504	504	504	504	517	526	530	755	658	715	539
3	0,25	0,25	538	539	538	635	627	666	627	1229	944	1034	649
4	0,25	0,25	570	570	570	570	570	640	585	1341	896	1087	654
5	0,25	0,25	359	359	359	373	375	510	367	985	644	783	609
6	0,25	0,25	453	453	453	453	457	494	453	1271	618	1051	483
7	0,25	0,25	536	536	536	589	578	613	584	1480	940	882	631
8	0,25	0,25	430	461	430	473	485	503	474	1195	726	907	534
9	0,25	0,25	582	632	582	681	660	703	675	1732	976	1316	918
10	0,25	0,25	587	587	587	587	603	631	592	1743	897	1123	843
11	0,25	0,5	12	12	12	12	12	12	12	2465	12	609	39
12	0,25	0,5	4	4	4	4	32	4	4	1103	4	567	70
13	0,25	0,5	9	9	9	9	19	9	9	1657	9	491	15
14	0,25	0,5	143	143	143	143	164	205	143	2555	205	1244	191
15	0,25	0,5	30	30	30	30	30	30	30	1821	30	476	32
16	0,25	0,5	29	29	29	29	29	29	29	1508	29	473	29
17	0,25	0,5	22	22	22	22	24	29	22	1545	29	338	34
18	0,25	0,5	15	15	15	15	15	15	15	959	15	227	70
19	0,25	0,5	1	1	1	1	1	1	1	2021	1	605	1
20	0,25	0,5	4	4	4	4	4	4	4	1918	4	452	4
21	0,25	0,75	0	0	0	0	0	0	0	3011	0	0	1474
22	0,25	0,75	0	0	0	0	0	0	0	1759	0	0	70
23	0,25	0,75	0	0	0	0	0	0	0	2026	0	0	272
24	0,25	0,75	0	0	0	0	0	0	0	1432	0	0	313
25	0,25	0,75	0	0	0	0	0	0	0	2951	0	0	892
26	0,25	0,75	0	0	0	0	0	0	0	4448	0	0	412
27	0,25	0,75	0	0	0	0	0	0	0	1115	0	0	736
28	0,25	0,75	0	0	0	0	0	0	0	1884	0	0	3
29	0,25	0,75	0	0	0	0	0	0	0	1248	0	0	1768
30	0,25	0,75	0	0	0	0	0	0	0	2990	0	0	207
31	0,5	0,25	3550	3796	3550	3771	3786	3886	3834	4412	8432	4033	4199
32	0,5	0,25	4718	4849	4718	4829	4847	5056	4929	5695	6550	5457	5366
33	0,5	0,25	3881	3898	3881	4043	4059	4148	3918	5160	5824	5042	4931
34	0,5	0,25	3846	3846	3846	3981	3906	4227	3951	4829	6397	4653	4448
35	0,5	0,25	4689	4698	4689	4911	4756	5000	4766	5816	7130	5620	5472
36	0,5	0,25	3864	4005	3873	3993	3996	4102	4055	4929	5110	4853	4695
37	0,5	0,25	3732	3757	3739	3773	3787	3807	3777	4818	5314	4484	4288
38	0,5	0,25	4702	4750	4750	4784	4759	4866	4837	5916	8247	5461	5669
39	0,5	0,25	3947	4049	3951	4046	4070	4331	4187	4854	7849	4535	4772
40	0,5	0,25	4336	4509	4380	4621	4654	5127	4550	5577	7266	5069	5360
41	0,5	0,5	2212	2216	2216	2212	2218	2364	2294	4519	3316	3097	3265
42	0,5	0,5	**2811**	2908	2809	2918	2935	3024	2960	5947	4262	4266	3913
43	0,5	0,5	**2926**	2976	2923	3067	3073	3132	3048	6881	4241	4674	4594
44	0,5	0,5	2006	2019	2019	2049	2040	2053	2080	3992	3411	3188	2843
45	0,5	0,5	**3558**	3554	3554	3586	3604	3636	3974	5326	5903	4274	4876
46	0,5	0,5	2467	2467	2467	2467	2471	2598	2633	4870	4340	3081	3922

Table G.1 (Continued)

47	0,5	0,5	**3861**	3914	3867	3901	3914	4105	3980	6400	7053	5093	5431
48	0,5	0,5	3300	3300	3300	3330	3331	3413	3415	6165	4388	5241	5185
49	0,5	0,5	2767	2767	2767	2767	2771	2947	2824	4562	3865	3367	3573
50	0,5	0,5	2244	2244	2244	2244	2251	2552	2380	5325	3919	3070	3777
51	0,5	0,75	**3346**	3288	3288	3362	3416	3602	3516	7699	6424	3823	5352
52	0,5	0,75	**1145**	1145	1145	1149	1194	1219	1187	5885	1609	2828	2546
53	0,5	0,75	**1606**	1685	1606	1676	1687	1718	1804	5238	2511	2658	3213
54	0,5	0,75	**1330**	1289	1270	1281	1305	1353	1355	6077	2042	3087	3174
55	0,5	0,75	**2067**	2060	2042	2014	2131	2297	2239	6023	3679	3342	3934
56	0,5	0,75	**1177**	1247	1176	1258	1262	1362	1236	5152	1916	1916	2711
57	0,5	0,75	**1932**	1751	1751	1751	1949	2012	2023	9252	2787	3785	4059
58	0,5	0,75	**3431**	3532	3446	3559	3594	3715	3714	8535	5281	4587	5547
59	0,5	0,75	**1836**	1818	1818	1819	1855	2061	1952	7983	2827	3317	4017
60	0,5	0,75	**2089**	2094	2089	2089	2123	2197	2301	5592	3225	2694	4009
61	0,75	0,25	9291	9427	9291	9455	9474	9932	9994	9995	14102	10256	9954
62	0,75	0,25	11736	12227	11736	12192	12222	12270	12328	12816	19039	13244	12759
63	0,75	0,25	11299	11299	11299	11573	11591	11618	11443	12201	16358	12380	12067
64	0,75	0,25	9441	9610	9463	9610	9719	9949	9735	10233	15690	10574	10266
65	0,75	0,25	12248	12251	12248	12438	12380	12513	12495	13326	21328	13269	13362
66	0,75	0,25	**8993**	9340	8987	9544	9469	9654	9365	9696	17665	10231	9676
67	0,75	0,25	11090	11619	11097	11539	11541	11693	11538	12180	19143	12100	12279
68	0,75	0,25	9149	9457	9161	9413	9327	9887	9443	9979	16768	10220	9995
69	0,75	0,25	9677	10101	9677	10101	10055	10393	10658	10573	16237	10901	10443
70	0,75	0,25	10901	11165	10901	11489	11426	11596	11179	11702	17746	12117	11752
71	0,75	0,5	**10186**	10162	10162	10213	10216	10446	10429	12470	16815	11945	11877
72	0,75	0,5	**9281**	9259	9237	9247	9257	9690	9676	11019	16972	9981	10694
73	0,75	0,5	**9077**	9214	9068	9090	9335	9493	9496	10914	14243	10116	10642
74	0,75	0,5	**8425**	8507	8364	8412	8414	8729	8774	10445	15473	9939	10056
75	0,75	0,5	**10352**	10376	10348	10399	10432	10750	10739	12380	19221	11683	12195
76	0,75	0,5	**9145**	9112	9088	9121	9122	9320	9208	10888	15874	11018	10632
77	0,75	0,5	**10120**	10069	10035	10035	10048	10304	10562	12177	19824	10900	12165
78	0,75	0,5	**11692**	11866	11645	11924	11930	12084	12011	13390	20710	12750	13385
79	0,75	0,5	**8966**	8954	8871	8942	9005	9411	9366	11095	14963	10264	10616
80	0,75	0,5	**8235**	8201	8201	8201	8224	8530	8365	9577	15553	9369	9531
81	0,75	0,75	**8614**	8533	8533	8560	8624	8757	8819	11034	15978	11961	10447
82	0,75	0,75	**8701**	8620	8611	8611	8671	8831	9015	11024	14420	11928	10555
83	0,75	0,75	**11322**	11346	11285	11292	11292	11483	11677	13311	16320	16820	12895
84	0,75	0,75	**13890**	13871	13871	13847	13850	14071	14122	16634	21253	16370	15829
85	0,75	0,75	**8289**	8360	8213	8217	8423	8553	8619	11198	14491	13200	10368
86	0,75	0,75	**9938**	9770	9767	9784	9821	10033	9955	13471	14364	14215	11818
87	0,75	0,75	**12357**	12050	12017	12050	12072	12294	12477	15537	18695	17130	14190
88	0,75	0,75	**11524**	11417	11417	11417	11484	11579	12034	14040	17412	19846	12800
89	0,75	0,75	**9058**	8916	8915	8915	8994	9083	9300	11491	15905	13841	10657
90	0,75	0,75	**11503**	11187	11187	11235	11243	11274	11512	14758	16342	15743	12718

* Values written bold in the column of optimal value are not exact optimum.

APPENDIX H

EXPERIMENTS FOR TOTAL TARDINESS PROBLEM (N=100)

Table H.1 Experiments for total tardiness problem (N=100)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Mdd	Au	Psk	Cov-ert	Spt	Edd	Hod-gson	Mon-tag-ne
1	0,25	0,25	3288	3305	3300	3625	3491	4245	3373	8767	5867	6989	5119
2	0,25	0,25	3402	3424	3402	3613	3590	3791	3465	8163	6729	6034	4390
3	0,25	0,25	3166	3203	3203	3267	3207	3443	3203	9096	4855	5743	4058
4	0,25	0,25	4882	5090	4985	4992	5131	5235	5103	14937	7720	11666	5591
5	0,25	0,25	4931	5152	5084	5355	5264	5459	5370	13313	8309	8553	6704
6	0,25	0,25	3930	3947	3944	4077	4164	4106	4036	10342	6288	6715	5121
7	0,25	0,25	4108	4144	4108	4401	4166	4504	4283	10809	6058	8372	5085
8	0,25	0,25	3481	3530	3517	3481	3768	3576	3571	10555	5876	8180	4598
9	0,25	0,25	3821	3952	3821	4006	4136	4092	4067	9121	6380	6728	4990
10	0,25	0,25	3020	3034	3030	3208	3179	3372	3102	9703	6150	6752	3961
11	0,25	0,5	6	6	6	6	7	6	6	6779	6	1770	7
12	0,25	0,5	12	12	12	12	12	12	12	9062	12	461	12
13	0,25	0,5	23	23	23	23	103	23	23	7758	23	777	55
14	0,25	0,5	2	2	2	2	28	2	2	12551	2	1957	10
15	0,25	0,5	136	136	136	136	169	167	151	10737	170	311	193
16	0,25	0,5	37	37	37	37	73	37	37	9249	37	695	41
17	0,25	0,5	28	28	28	28	33	28	28	8740	28	2024	33
18	0,25	0,5	3	3	3	3	3	3	3	12999	3	1999	3
19	0,25	0,5	15	15	15	15	15	15	15	11822	15	347	15
20	0,25	0,5	123	123	123	125	329	125	123	14585	135	763	648
21	0,25	0,75	0	0	0	0	0	0	0	17096	0	0	1332
22	0,25	0,75	0	0	0	0	0	0	0	11815	0	0	1837
23	0,25	0,75	0	0	0	0	0	0	0	8478	0	0	29
24	0,25	0,75	0	0	0	0	0	0	0	13451	0	0	1832
25	0,25	0,75	0	0	0	0	0	0	0	21018	0	0	1632
26	0,25	0,75	0	0	0	0	0	0	0	9584	0	0	539
27	0,25	0,75	0	0	0	0	0	0	0	17271	0	0	2970
28	0,25	0,75	0	0	0	0	0	0	0	17877	0	0	3202
29	0,25	0,75	0	0	0	0	0	0	0	12497	0	0	2181
30	0,25	0,75	0	0	0	0	0	0	0	20970	0	0	4741
31	0,5	0,25	30114	31322	30114	32185	31436	32836	31595	37960	45052	36110	36333
32	0,5	0,25	26947	27972	27279	28218	27992	28560	28199	34019	46506	30981	32221
33	0,5	0,25	36834	37638	37018	38291	37826	39267	38244	47645	59217	42896	43042
34	0,5	0,25	29610	29953	29724	30380	30037	30489	30218	38297	48209	35130	36393
35	0,5	0,25	**28917**	30166	28986	30780	30230	31107	30492	35949	50754	33487	33909
36	0,5	0,25	**29974**	30124	30109	30631	30561	31332	30725	39146	56668	35470	37264
37	0,5	0,25	27144	28250	27280	28376	28252	28669	28628	33964	45837	30752	32092
38	0,5	0,25	35112	36089	35372	35870	36064	35966	36169	46375	51842	42166	42108
39	0,5	0,25	33872	34797	33961	35177	34780	36432	35060	40646	54689	38696	39639
40	0,5	0,25	30211	31875	30310	31911	32057	33230	31781	37967	56679	34948	35072
41	0,5	0,5	**20433**	21536	20409	20967	21040	21564	22209	45671	33868	25208	34041
42	0,5	0,5	**16150**	16192	16149	16483	16294	16630	16803	34648	25592	22362	23185
43	0,5	0,5	**19486**	20036	19483	20035	20153	20302	20383	43804	31065	28572	30974
44	0,5	0,5	**18152**	18833	18136	18727	18983	19145	18954	42715	28358	29034	28304
45	0,5	0,5	**21390**	22221	21382	22629	22385	22811	22617	40191	38522	26489	31790
46	0,5	0,5	**20877**	21426	20825	21403	21584	21837	21641	38286	34627	28539	30767
47	0,5	0,5	**20189**	20282	20108	20444	20549	21205	21371	40843	35757	25089	31394
48	0,5	0,5	**19016**	19413	19034	19666	19476	19986	20107	41306	30008	25415	29756
49	0,5	0,5	**21252**	21302	21218	21261	21546	22078	21777	39848	38286	27577	32679
50	0,5	0,5	**18357**	19067	18345	19170	19028	19546	19280	33562	30117	27534	27374
51	0,5	0,75	**12680**	12570	12272	12618	12626	13576	12887	54608	20320	21024	29218
52	0,5	0,75	**11198**	11171	10975	11239	11287	11652	11667	45616	20460	18087	23918

Table H.1 (Continued)

53	0,5	0,75	**6488**	6549	6549	6550	6643	7116	6952	42907	10428	16402	18049
54	0,5	0,75	**10087**	10078	9802	10045	10108	10331	11099	45421	14550	17227	23377
55	0,5	0,75	**14689**	13523	13523	13527	13594	14423	14419	45629	23516	20464	27975
56	0,5	0,75	**5330**	5280	5280	5280	5321	5677	5498	36285	8968	14979	15154
57	0,5	0,75	**21590**	20569	20090	20584	20690	21309	22255	55428	37435	28319	35281
58	0,5	0,75	**14199**	13490	13209	13257	13574	14807	14794	52682	23859	20277	28552
59	0,5	0,75	**17346**	17183	16674	17294	17361	18351	18821	55439	27043	26052	33589
60	0,5	0,75	**12930**	13413	12902	13424	13471	13756	13946	53290	20780	21684	26340
61	0,75	0,25	**73055**	75547	73032	75937	75066	76108	74769	78919	123583	80136	78259
62	0,75	0,25	**82819**	85708	83036	85769	86200	87269	86087	89237	137147	91533	89489
63	0,75	0,25	**80984**	84007	80975	83307	83627	84506	85327	89868	128793	91672	88506
64	0,75	0,25	**87261**	90599	87655	91254	89834	91667	91734	93559	140303	94617	93134
65	0,75	0,25	**79533**	81796	79512	81472	81411	82478	82837	87435	132529	87830	87160
66	0,75	0,25	**85277**	87021	85427	86556	86866	87259	88953	94233	136040	92458	94262
67	0,75	0,25	**69914**	72519	69903	72311	70606	72898	72822	77086	125504	77722	75857
68	0,75	0,25	**88260**	89500	88270	89407	89548	90014	90028	97560	139836	97015	96117
69	0,75	0,25	**94676**	96439	94671	95830	96142	96364	96425	105972	152681	105085	104313
70	0,75	0,25	91970	96435	91970	96423	94094	98621	96691	96573	142650	100187	96946
71	0,75	0,5	**78150**	77209	77034	77238	77203	77898	78647	91891	137483	83264	87800
72	0,75	0,5	**67860**	68065	67634	67959	68046	68519	69313	83006	108197	75211	78340
73	0,75	0,5	**83213**	82185	81965	82024	82124	82905	82988	101115	132058	94911	96213
74	0,75	0,5	**62253**	61247	61020	61120	61199	62080	62822	78276	107505	68553	74758
75	0,75	0,5	**64592**	64005	63911	64027	64087	64901	64346	80247	122109	74232	76747
76	0,75	0,5	**76500**	76189	75598	75742	75773	76974	78820	89223	130760	82555	88524
77	0,75	0,5	**72069**	71295	71258	71250	71308	71925	72865	91046	123700	79628	88105
78	0,75	0,5	**54003**	53952	53290	53605	54035	54769	55476	66839	104356	62056	64610
79	0,75	0,5	**79638**	79269	78744	78670	78744	80850	80643	97406	130418	86758	94222
80	0,75	0,5	**68561**	68218	67771	67794	67887	69388	69905	81986	112084	74917	78640
81	0,75	0,75	**85302**	82447	82292	82316	82437	82981	83191	102096	139820	106512	96867
82	0,75	0,75	**86045**	82052	81940	81907	82067	82780	82992	104856	140246	109101	97740
83	0,75	0,75	**55093**	54019	53843	53750	53816	55030	55355	73106	78807	80550	67448
84	0,75	0,75	**94265**	90102	89719	89703	89879	90436	93037	119591	146423	111332	109489
85	0,75	0,75	**72843**	70003	69994	70080	70211	70843	71038	97141	115667	92131	87277
86	0,75	0,75	**64809**	61843	61600	61710	61815	62526	63142	82016	116939	79553	76982
87	0,75	0,75	**92252**	89056	88938	88938	88967	89584	90619	109830	137238	114952	100723
88	0,75	0,75	**80488**	77597	77402	77367	77485	78093	80388	98224	136186	104544	91007
89	0,75	0,75	**76653**	74058	73919	73958	74019	74542	75743	94779	125265	96797	87694
90	0,75	0,75	**79912**	76029	75819	75974	76212	76783	77076	100623	129191	103078	91703

* Values written bold in the column of optimal value are not exact optimum.

APPENDIX I

EXPERIMENTS FOR TOTAL WEIGHTED TARDINESS PROBLEM (N=20)

Table I.1 Experiments for total weighted tardiness problem (N=20)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Au	Greedy	Edd	WEdd	Swpt	Mont-Agne
1	0,25	0,25	190	190	190	235	190	470	369	400	307
2	0,25	0,25	165	165	165	167	169	221	258	371	173
3	0,25	0,25	112	112	112	129	129	239	288	253	200
4	0,25	0,25	170	170	170	189	178	477	375	319	186
5	0,25	0,25	174	174	174	194	218	452	252	263	237
6	0,25	0,25	283	283	283	340	331	470	784	694	474
7	0,25	0,25	66	66	66	126	66	143	181	192	147
8	0,25	0,25	172	172	172	187	172	441	297	353	187
9	0,25	0,25	96	96	96	132	96	498	262	216	196
10	0,25	0,25	48	48	48	72	48	433	99	126	90
11	0,25	0,5	9	9	9	25	9	9	131	553	12
12	0,25	0,5	14	14	14	30	14	14	71	136	14
13	0,25	0,5	29	29	29	33	29	162	144	190	93
14	0,25	0,5	20	20	20	20	20	24	122	212	36
15	0,25	0,5	54	54	54	108	128	117	243	357	182
16	0,25	0,5	27	27	27	89	74	27	601	1386	187
17	0,25	0,5	24	24	24	54	24	24	679	691	168
18	0,25	0,5	8	8	8	31	28	10	142	124	13
19	0,25	0,5	53	53	53	53	53	72	164	268	61
20	0,25	0,5	12	12	12	35	35	12	189	57	22
21	0,25	0,75	0	0	0	8	0	0	251	381	94
22	0,25	0,75	66	70	70	85	70	254	344	862	316
23	0,25	0,75	0	0	0	0	0	0	256	608	337
24	0,25	0,75	0	0	0	0	0	0	398	596	231
25	0,25	0,75	0	0	0	0	0	0	604	926	262
26	0,25	0,75	0	0	0	0	0	0	331	601	220
27	0,25	0,75	12	12	12	32	200	15	768	1227	418
28	0,25	0,75	0	0	0	72	0	0	458	649	351
29	0,25	0,75	0	0	0	0	0	0	23	279	93
30	0,25	0,75	0	0	0	7	0	0	326	376	159
31	0,5	0,25	839	839	839	884	839	3711	1357	965	965
32	0,5	0,25	1124	1124	1124	1204	1204	4358	1650	1509	1522
33	0,5	0,25	**668**	668	668	742	774	2811	952	864	798
34	0,5	0,25	**986**	986	986	1073	1169	2699	1313	1344	1226
35	0,5	0,25	**1595**	1620	1620	1698	1736	3355	2236	1778	1730
36	0,5	0,25	**1038**	1038	1038	1038	1078	1920	1547	1276	1250
37	0,5	0,25	750	765	750	775	775	2461	902	846	835
38	0,5	0,25	644	648	648	648	652	2136	811	735	670
39	0,5	0,25	1820	1925	1820	1852	2079	3069	2999	2277	2188
40	0,5	0,25	707	707	707	760	736	2384	939	798	804

Table I.1 (Continued)

41	0,5	0,5	548	548	548	576	561	1946	918	1127	962
42	0,5	0,5	276	315	315	319	315	1064	523	683	470
43	0,5	0,5	**1099**	1099	1099	1142	1116	2944	1495	1550	1210
44	0,5	0,5	939	957	957	967	968	2897	2080	1707	1300
45	0,5	0,5	610	610	610	707	610	2374	1159	992	900
46	0,5	0,5	655	655	655	882	655	1680	1043	1095	1032
47	0,5	0,5	1223	1223	1223	1317	1353	2996	2375	2177	1598
48	0,5	0,5	213	231	231	253	231	464	551	640	414
49	0,5	0,5	863	879	863	1051	930	2040	1334	1498	1131
50	0,5	0,5	**1542**	1542	1542	1625	1639	4118	1934	2282	1938
51	0,5	0,75	174	174	174	213	221	459	461	1116	375
52	0,5	0,75	897	897	897	999	897	2888	1552	2070	1570
53	0,5	0,75	558	558	558	597	569	1770	829	1155	970
54	0,5	0,75	429	491	491	530	585	841	1434	1830	734
55	0,5	0,75	442	442	442	491	751	986	1192	1244	919
56	0,5	0,75	461	498	498	644	598	1541	1282	1431	911
57	0,5	0,75	295	295	295	403	370	803	865	1170	849
58	0,5	0,75	227	227	227	282	431	608	1036	1647	553
59	0,5	0,75	**2537**	2572	2537	2625	2774	4802	4127	4212	3614
60	0,5	0,75	384	392	392	498	400	1866	831	1413	1012
61	0,75	0,25	**3346**	3447	3343	3388	3466	6214	4303	3643	3620
62	0,75	0,25	**2223**	2223	2223	2229	2471	4699	3584	2511	2529
63	0,75	0,25	**3119**	3251	3119	3190	3280	7708	4707	3375	3361
64	0,75	0,25	**2180**	2180	2180	2220	2220	4462	2854	2408	2408
65	0,75	0,25	**2159**	2159	2159	2175	2355	6952	3056	2523	2498
66	0,75	0,25	**3068**	3105	3105	3112	3105	7554	4105	3309	3264
67	0,75	0,25	**3191**	3209	3209	3270	3338	5644	4035	3519	3457
68	0,75	0,25	**3455**	3455	3455	3585	3560	7554	4754	4073	3950
69	0,75	0,25	**2509**	2509	2509	2541	2509	5424	3753	2695	2701
70	0,75	0,25	**3402**	3402	3402	3410	3508	7201	4615	3607	3615
71	0,75	0,5	**3094**	3094	3094	3221	3094	7114	5322	3957	3505
72	0,75	0,5	**3217**	3222	3222	3262	3282	5343	5046	3930	3773
73	0,75	0,5	**2170**	2230	2230	2182	2325	3931	2901	2548	2468
74	0,75	0,5	**3655**	3663	3663	3767	3712	7715	5389	4233	4188
75	0,75	0,5	**4028**	4011	4011	4132	4071	8821	7850	4367	4460
76	0,75	0,5	**2813**	2813	2813	2862	2813	7941	4563	3538	3472
77	0,75	0,5	**4288**	4288	4288	4381	4345	7312	5490	5320	4988
78	0,75	0,5	**1821**	1821	1821	1999	1833	5726	3101	2477	2444
79	0,75	0,5	**2116**	2116	2116	2186	2186	6873	2663	2883	2509
80	0,75	0,5	**2654**	2654	2654	2719	2666	5430	5151	3114	2823
81	0,75	0,75	**1251**	1228	1220	1268	1277	3669	1780	1906	1781
82	0,75	0,75	**1172**	1172	1172	1338	1293	5357	2473	1957	1705
83	0,75	0,75	**2551**	2547	2547	2565	2708	5092	4179	3194	2968
84	0,75	0,75	**3506**	3506	3506	3620	3506	7647	6488	3860	3743
85	0,75	0,75	**1892**	1895	1883	2076	2070	4860	3141	2993	2661
86	0,75	0,75	**3792**	3792	3792	3950	3837	7514	5553	5114	4559
87	0,75	0,75	**5097**	5097	5097	5161	5168	7781	7140	5989	5959
88	0,75	0,75	**2030**	2030	2030	2150	2080	3177	2853	2370	2256
89	0,75	0,75	**3606**	3606	3606	3671	3713	7231	4918	4703	4157
90	0,75	0,75	**5995**	5995	5995	6090	6130	9751	8961	7341	6690

* Values written bold in the column of optimal value are not exact optimum.

APPENDIX J

EXPERIMENTS FOR TOTAL WEIGHTED TARDINESS PROBLEM (N=40)

Table J.1 Experiments for total weighted tardiness problem (N=40)

No	τ	R	Opt	Cons. Heu.	Cons. + Imp. Heu.	Au	Greedy	Edd	WEdd	Swpt	Mont-Agne
1	0,25	0,25	356	359	359	401	359	1498	673	607	608
2	0,25	0,25	**1314**	1399	1274	1314	1990	4332	3077	2405	2280
3	0,25	0,25	604	665	665	686	665	2723	1704	1837	1101
4	0,25	0,25	508	718	508	913	718	2181	1760	1246	928
5	0,25	0,25	571	571	571	800	769	2842	1655	1409	793
6	0,25	0,25	582	582	582	700	756	2478	1567	1334	900
7	0,25	0,25	371	442	371	414	442	1614	1071	995	863
8	0,25	0,25	1024	1024	1024	1269	1627	1749	2774	1671	1268
9	0,25	0,25	**659**	745	745	798	863	2629	1618	1437	1148
10	0,25	0,25	795	795	795	933	1118	1680	2636	1816	907
11	0,25	0,5	8	8	8	60	8	8	2292	2342	185
12	0,25	0,5	25	25	25	84	77	50	3098	5156	421
13	0,25	0,5	0	0	0	15	0	0	3130	4033	247
14	0,25	0,5	168	242	219	233	242	379	772	1127	792
15	0,25	0,5	252	252	252	412	252	597	3282	2664	1088
16	0,25	0,5	22	22	22	40	22	24	1739	2132	475
17	0,25	0,5	123	123	123	147	295	282	1234	1169	202
18	0,25	0,5	256	256	256	406	256	1570	1797	2986	1268
19	0,25	0,5	47	47	47	124	47	200	1140	830	333
20	0,25	0,5	31	31	31	44	31	32	1496	2692	141
21	0,25	0,75	0	0	0	0	0	0	1279	1967	1006
22	0,25	0,75	0	0	0	0	0	0	1240	3190	2654
23	0,25	0,75	0	0	0	0	0	0	2115	3849	1302
24	0,25	0,75	0	0	0	0	0	0	738	929	164
25	0,25	0,75	0	0	0	0	0	0	3399	5781	1614
26	0,25	0,75	0	0	0	0	0	0	1309	1698	1270
27	0,25	0,75	0	0	0	0	0	0	933	1132	1017
28	0,25	0,75	0	0	0	0	0	0	464	2655	3476
29	0,25	0,75	0	0	0	0	0	0	1846	3098	1795
30	0,25	0,75	0	0	0	0	0	0	1811	3716	1644
31	0,5	0,25	**10288**	10350	10288	10645	11165	24330	16092	13116	12197
32	0,5	0,25	**5899**	6231	5854	6487	6231	18088	9917	7188	7025
33	0,5	0,25	**4188**	3968	3968	4008	4583	14607	6957	5502	5186
34	0,5	0,25	**5640**	5942	5603	5970	6174	19709	9045	7098	6821
35	0,5	0,25	**4493**	4684	4549	5169	4753	14740	6570	5537	5107
36	0,5	0,25	**6100**	6389	6100	7202	6769	16676	9638	7890	7003
37	0,5	0,25	**4017**	3996	3961	4351	4153	12984	7308	4989	4955

Table J.1 (Continued)

38	0,5	0,25	**6357**	6836	6327	8073	7398	19214	11950	9060	8039
39	0,5	0,25	**6028**	6487	5830	7455	7185	21585	9546	8176	7586
40	0,5	0,25	**4822**	5397	4804	5202	5539	21371	9174	8000	6385
41	0,5	0,5	**10070**	10196	10167	11734	11242	19732	17430	15904	13103
42	0,5	0,5	**2945**	2977	2700	3563	3076	11860	7358	10095	5778
43	0,5	0,5	**3772**	3833	3772	3860	4180	10090	6572	8440	6768
44	0,5	0,5	**2593**	2779	2584	2949	3452	13191	7296	8038	5271
45	0,5	0,5	**5686**	5802	5656	6771	8539	16921	12179	12481	8806
46	0,5	0,5	**4450**	4993	4280	5034	5597	17732	10145	9452	6912
47	0,5	0,5	**3319**	3811	3265	4547	3800	12091	7111	8560	6249
48	0,5	0,5	**2412**	2348	2348	2552	3364	12131	5216	5819	4657
49	0,5	0,5	**6374**	6055	6055	7322	6376	18848	12290	14750	9788
50	0,5	0,5	**4912**	4925	4924	5239	5207	11904	7513	10066	7143
51	0,5	0,75	**2987**	3134	2937	3610	3390	7973	7860	14208	7624
52	0,5	0,75	**697**	686	686	793	1305	2924	3141	4691	2878
53	0,5	0,75	**1488**	1654	1476	1964	1663	7415	4235	6946	4550
54	0,5	0,75	**2949**	2691	2691	3195	4042	8941	11429	15560	8905
55	0,5	0,75	**3592**	3207	3193	3526	3512	13355	8480	14250	7539
56	0,5	0,75	**6274**	6448	6377	6975	7179	20468	12199	15219	10051
57	0,5	0,75	**2768**	3187	3092	4010	3092	9787	8258	7678	5145
58	0,5	0,75	**1925**	1909	1909	2305	1925	7643	5666	9799	4067
59	0,5	0,75	**2381**	2575	2189	2775	3016	5485	6773	8062	5266
60	0,5	0,75	**1534**	1530	1530	1700	2843	3020	6170	8567	4829
61	0,75	0,25	**15527**	15440	15426	15682	15677	43837	21813	17289	17014
62	0,75	0,25	**22860**	23148	22860	22836	25029	42230	29960	26543	25086
63	0,75	0,25	**31584**	32016	31592	31807	32476	58723	41973	36608	35544
64	0,75	0,25	**13282**	13331	13282	13349	13697	28079	19701	15585	15435
65	0,75	0,25	**17687**	17907	17648	18698	18491	44853	25615	20798	20382
66	0,75	0,25	**26181**	26374	26261	26461	26661	55393	33563	27412	27503
67	0,75	0,25	**19892**	20315	19892	21929	21161	46333	28642	22712	22179
68	0,75	0,25	**20718**	20645	20615	20931	20922	50411	27177	23333	21325
69	0,75	0,25	**19772**	20366	20332	19836	20753	44727	26734	22465	22042
70	0,75	0,25	**27932**	28080	27671	28121	28396	61100	37144	31602	31107
71	0,75	0,5	**15797**	16128	15751	15745	17691	54667	28736	20585	19736
72	0,75	0,5	**21406**	21375	21375	21697	22479	52410	35985	26898	26381
73	0,75	0,5	**19392**	19470	19296	19785	19519	43943	27055	25427	23804
74	0,75	0,5	**22788**	22740	22737	23012	23851	50067	38580	27089	26014
75	0,75	0,5	**22761**	22659	22638	23184	22696	53091	38242	28355	27371
76	0,75	0,5	**20500**	20291	20291	20553	20440	40471	29813	24857	23054
77	0,75	0,5	**19231**	20120	19152	19578	20643	53887	36378	23209	22845
78	0,75	0,5	**25482**	25410	25154	25786	26333	53728	39601	30778	29790
79	0,75	0,5	**13056**	12909	12909	13154	13020	39583	23264	15944	15226
80	0,75	0,5	**20380**	19489	19485	19746	19910	49494	27394	26666	24104
81	0,75	0,75	**23784**	23418	23418	23834	23535	53055	47340	30501	27511
82	0,75	0,75	**12208**	12292	12024	12250	12320	34597	25509	18826	16420
83	0,75	0,75	**20012**	19655	19645	19941	19804	45411	34844	25695	23051
84	0,75	0,75	**20356**	20204	20184	20446	20184	43649	36174	26816	23507
85	0,75	0,75	**22748**	22645	22640	22893	22640	50608	39688	29719	25976
86	0,75	0,75	**14364**	14328	14328	14892	14717	43106	28273	19859	18452
87	0,75	0,75	**7297**	7195	7195	7640	7325	29992	11241	12230	10617
88	0,75	0,75	**26266**	26174	26131	26472	29157	51200	37342	38027	30352
89	0,75	0,75	**18912**	18700	18700	18852	18700	38849	28323	25465	23260
90	0,75	0,75	**22151**	22079	22079	22359	22239	44758	35762	28441	25901

* Values written bold in the column of optimal value are not exact optimum.