CUSTOMER ORDER SCHEDULING ON TWO IDENTICAL PARALLEL MACHINES WITH JOB SETUP TIMES

THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
NİHAN ÇAĞLAYAN

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

SEPTEMBER 2017

Title of Thesis: **Customer Order Scheduling on Two Identical Parallel Machines with Job Setup Times**

Submitted by **Nihan ÇAĞLAYAN**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Can ÇOĞUN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Assist. Prof. Dr. Abdül Kadir GÖRÜR	Assoc. Prof. Dr. Ferda Can ÇETİNKAYA
Co-supervisor	Supervisor

Examination Date: 14.09.2017

Examining Committee Members

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA	(Çankaya Univ.)
Assoc. Prof. Dr. Alp ERTEM	(Çankaya Univ.)
Assist. Prof. Dr. Sakine BATUN	(METU)

STATEMENT OF NON-PLAGIARISM

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 : Nihan ÇAĞLAYAN

Signature :

Date : 14.09.2017

ABSTRACT

CUSTOMER ORDER SCHEDULING ON TWO IDENTICAL PARALLEL MACHINES WITH JOB SETUP TIMES

ÇAĞLAYAN, Nihan

M.S., Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Ferda Can ÇETİNKAYA

Co-Supervisor: Assist. Prof. Dr. Abdül Kadir GÖRÜR

September 2017, 48 pages

Across all countries, manufacturers seek to adapt the best strategies to provide the highest qualities of services with lowest costs. For that matter, researchers have tried to develop better shop structures that in fact influenced and even can be controlled by a single machine, parallel machines and flow shop matters alongside with setup considerations. They especially put forwarded several approaches to scheduling by focusing on making certain alterations on setup timings of job assignments to achieve the best time saving and eventually the lowest cost. Studies on setup times or costs showed that, running the grouped jobs with the same or similar setup needs dramatically reduces both the setup times and costs. This procedure can also be classified as group technology and customer order scheduling (COS). The focus of this thesis is to provide an alternative scheduling for customer orders that may contain one or more than one job lots within two identical parallel machines. The common belief of any customer order is very straightforward which in fact requires its orders' to be proceed at the same time with a prompt attitude so that all job lot can be received with a well synchronization. The completion time of the last set of each customer order also indicates the completed duration of the customer order.

Obviously, each job batch requires some certain setup arrangements unless setup needs of current job assignment match with upcoming assignment's setup needs. This study is an attempt to suggest a more time saving and low costing schedule by grouping and running customer orders with same setup requirements at most appropriate route via two identical parallel machines so as to reduce makespan time to present customers their orders in best optimized way. The existing problem has more than one variety; e.g., customer orders may contain more than one job assignment or orders may have different setup arrangements and times, etc. In such complex cases, it is fair to say that the problem is strongly NP-hard. MILP is practiced to solve optimal to small sized problems whereas a constructive algorithm is conducted to handle medium and large sized problems in order to get optimal and/or near-optimal solutions. GAMS, the optimization software for mathematical programming model is used to get optimum results. The heuristic algorithm is coded by computer language C++. In result of computational experiments, it was found that the mathematical model is inadequate to cover or may even fail to acquire solutions for especially medium and large sized problems.

Keywords: Scheduling, Customer Order Scheduling, Group Scheduling, Makespan, Identical Parallel Machines

ÖZ

KURULUM SÜRELİ İKİ ÖZDEŞ PARALEL MAKİNADA MÜŞTERİ SİPARİŞİ ÇİZELGELEMESİ

ÇAĞLAYAN, Nihan

Yüksek Lisans, Endüstri Mühendisliği Anabilim Dalı

Tez Yöneticisi: Doç. Dr. Ferda Can ÇETİNKAYA

Ortak Tez Yöneticisi: Yrd. Doç. Dr. Abdül Kadir GÖRÜR

Eylül 2017, 48 sayfa

Üreticiler, müşterilerine en kaliteli hizmeti minumum bütçeyle sunabilmek adına zaman içinde çeşitli stratejiler geliştirmeyi hedeflemişlerdir. Bunu temin edebilmek için de araştırmacılar, süreç içerisinde kurum yapısında baskın rol oynayan tek makina, paralel makina ve akış tipi problemleri kurulum hususlarıyla birlikte ele alarak daha iyi bir kurum yapısı oluşturmaya çalışmışlardır. Gelen işlerin kurulum sürelerine odaklanıp çeşitli iyileştirmeler yapılarak en iyi zaman tasarrufu ve böylelikle düşük bütçeye ulaşabilmek için çizelgelemeye farklı yaklaşımlar getirmişlerdir. Kurulum süresi ya da maliyeti üzerine yapılan araştırmalar göstermiştir ki, aynı ya da benzer kurulum ihtiyaçlarına sahip işleri gruplandırmak büyük ölçüde kurulum süresini ve maliyetini azaltmaktadır. Bu işlem, grup teknolojisi ve müşteri sipariş çizelgelemesi olarak da sınıflandırılabilir. Bu çalışma, bir ya da birden fazla içeriği olan müşteri siparişleri için bir çizelgeleme oluşturmaya odaklanmıştır. Her müşterinin hizmet sağlayıcıdan ortak talebi siparişlerinin

tamamının aynı zamanda üretilmesini ve aynı zamanda bitirilip toplu olarak kendilerine ulaştırılmasıdır. Son iş grubunun tamamlanma süresi aynı zamanda müşteri siparişinin de tamamlanma süresi anlamına gelir. O esnada işlenen mevcut iş ile hemen sonra devam ettirilecek iş için kurulum ihtiyaçları aynı olmadığı sürece elbette her iş grubu için farklı kurulum ayarları gerekebilir. Bu araştırma, aynı özelliklere sahip paralel makinalarla, farklı kurulum ayarları gerektirmeyen işleri gruplayarak arzu edilen zaman tasarrufu ve düşük bütçeye ulaşmak için, bütün siparişlerin tamamlanma süresini azaltıp müşteriye siparişlerini optimum şekilde ulaştırmayı sağlayacak bir çizelgeleme çalışmasıdır. Mevcut problemde birden fazla çeşitlilik vardır; örneğin, müşteri siparişlerinin bir ya da daha fazla çeşide sahip olması veya farklı kurulum sürelerine/ayarlamalara ihtiyaç duyması gibi. Bu gibi karmaşık durumlarda, problemin NP-hard şeklinde sınıflandırılması doğru olur. Küçük ölçekli problemlerin optimal çözümü için matematiksel model kullanılırken, orta ya da büyük ölçekli problemlerin en iyiye yakın çözümüne ya da en iyi çözümüne ulaşabilmek için sezgisel algoritma kullanıldı. Ayrıca, uygulanan sezgisel algoritmanın güvenilirliğini ve sürenin ideal kullanımını görebilmek için sayısal deney setleri oluşturuldu. Matematiksel programlama modelini optimum sonuçlara ulaşarak çözebilmek için GAMS adı verilen bir paket program kullanıldı. Sezgisel algoritma ise bilgisayar dillerinden olan C++ ile kodlandı. Yapılan sayısal deney setleri sonucunda, matematiksel modelin özellikle büyük ölçekli problemleri bütünüyle çözebilme ya da çözme konusunda yetersiz kaldığı görüldü. Bunun yanı sıra, bulgular, sezgisel algoritmanın optimum ya da optimuma yakın sonuçlara ulaşmada ve zamanı en uygun şekilde kullanmada daha iyi bir işleyişe sahip olduğunu gösterdi.

Anahtar Kelimeler: Çizelgeleme, Müşteri Sipariş Çizelgelemesi, Grup Çizelgelemesi, Siparişlerin Tamamlanma Süresi, Özdeş Paralel Makinalar

ACKNOWLEDGEMENTS

First of all I would like to thank my advisors Assoc. Prof. Dr. Ferda Can ÇETİNKAYA and Assist. Prof. Dr. Abdül Kadir GÖRÜR, for their permanent assistance, guidance and steadfastness during this study.

I would also like to thank my examining committee members Asst. Prof. Dr. Sakine BATUN and Assoc. Prof. Dr. Alp ERTEM for their precious opinions and contributions to this study.

I am thankful to my parents especially my mother and brother in love Kadir Can Dilber for their support and sharing my stress for particularly during thesis study.

I am also so grateful to my lovely friend Hale AKKOCAOĞLU and Betül ÖZTÜRK for supporting and sharing my tension during all stages in my thesis term.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM iii
ABSTRACT iv
ÖZ vi
ACKNOWLEDGEMENTS viii
TABLE OF CONTENTS ix
LIST OF FIGURES xi
LIST OF TABLES xii
LIST OF ABBREVIATIONS xiv
CHAPTER 1 1
1 INTRODUCTION 1
CHAPTER 2 3
2 PROBLEM DEFINITION AND PRELIMINARY SOLUTIONS 3
2.1 Problem Statement 3
2.2 Some Observations 7
2.3 Some Structural Properties for the Optimal Schedules 7
2.4 Some Special Cases Solvable by Priority Rules 8
CHAPTER 3 10
3 LITERATURE REVIEW 10
3.1 Group Scheduling 12
3.2 Customer Order Scheduling 13
CHAPTER 4 16

4 SOLUTION APPROACHES 16
4.1 Mathematical Model of the Problem 16
4.2 Shortest Path Network 19
4.3 Heuristic Algorithm 22
CHAPTER 5 33
5 COMPUTATIONAL EXPERIMENTS 33
5.1 Computational Setting for Test Problems 33
5.2 Performance Measures 34
5.3 Discussion of the Results 35
5.3.1 Performance of the MILP Model 35
5.3.2 Performance of the Heuristic Algorithms 38
CHAPTER 6 47
6 CONCLUSIONS AND PROPOSALS FOR FUTURE RESEARCH 47
REFERENCES 49
CURRICULUM VITAE 54

LIST OF FIGURES

FIGURES

Figure 1 A feasible schedule of customer orders and jobs on the machines 4
Figure 2 Shortest Path Network of the Machine 1 5
Figure 3 Shortest Path Network of the Machine 2 5
Figure 4 The feasible schedule of machines with setup saving 6
Figure 5 The classification of scheduling problem 12
Figure 6 Shortest Path Network of Machine 1 for Example 1 20
Figure 7 Shortest Part Network of Machine 2 for Example 1 21
Figure 8 Order allocation to the machines 26
Figure 9 Shortest Path Network for Machine 1 27
Figure 10 Shortest Path Network for Machine 2 29
Figure 11 Arc distances for the shortest path network on Machine 2 30
Figure 12 The makespan minimization sequence of the machines 31
Figure 13 Dissimilarity between the average percent deviations for the customer orders 42
Figure 14 Average percent deviations when K=10 53
Figure 15 Average percent deviations when K=15 53

LIST OF TABLES

TABLES

Table 1 Processing times and setup times of each jobs in Example 1 4
Table 2 The arc distances in the shortest path network for Machine 1 in Example 1.. .... **Hata! Yer işareti tanımlanmamış.**
Table 3 The arc distances in the shortest path network for Machine 2 in Example 1. 6
Table 4 The flow costs of Example 1 for Machine 1 22
Table 5 The flow costs of Example 1 for Machine 2 22
Table 6 Setup and Processing Times of the Jobs 25
Table 7 List of Jobs to be processed and Total Time of the Customer Orders 25
Table 8 Arc distances for the shortest path network on Machine 1 28
Table 9 Performance of mathematical model.......**Hata! Yer işareti tanımlanmamış.**
Table 10 Number of problems with the optimum results obtained from mathematical model.. 38
Table 11 The number of problems with the optimum results obtained from the mathematical model and heuristic algorithm ....**Hata! Yer işareti tanımlanmamış.**
Table 12 The number of optimum and the best integer results obtained from the mathematical model for VARIABLE Case.. 43
Table 13 The number of optimum and best integer results obtained from mathematical model for constant case 40
Tablo 14 Average percent deviations of the heuristic algorithm from optimal results for VARIABLE case 41
Tablo 15 Average percent of the heuristic algorithm from optimal results for CONSTANT case.. 43
Tablo 16 Average percent deviation of the heuristic algorithm from the best integer results for CONSTANT case 44

Tablo 17 Percent deviations of the heuristic when K=20 44
Tablo 18 General total percent deviations of the heuristic algorithm from the optimal solutions for CONSTANT and VARIABLE case 45
Tablo 19 General total percent deviations of the heuristic algorithm from the optimal solutions for CONSTANT and VARIABLE case.......................................46

LIST OF ABBREVIATIONS

COS	Customer Order Scheduling
NP	Nondeterministic Polynomial-time
MILP	Mixed Integer Linear Programming
TT	Total Time
LTT	Longest Total Time
LPT	Longest Processing Time
K	Number of customer orders
N	Number of different jobs
VARIABLE or VAR	Number of jobs within each customer order is variable, and determined by DU [1,*N*]
CONSTANT or CNST	Number of jobs within each customer order is constant (fixed), and determined by DU [2, *N*-1]
SHORT	Processing times are determined by DU [1,10]
LONG	Processing times are determined by DU [100,200]
LL	Setup times have low mean and low variance, and are determined by DU [25,35]
LH	S Setup times have low mean and high variance, and are determined by DU [10,50]
HL	Setup times have high mean and low variance, and are determined by DU [55,65]

HH	Setup times have high mean and high variance, and are determined by DU [40,80]

(Akkocaoğlu,2014)

CHAPTER 1

INTRODUCTION

This paper exclusively evaluates the customer orders scheduling on two parallel machines with setup times and aims to suggest a better understanding and dealing with makespan time. In plain sight, we grouped the collections of jobs (customer orders) with similar or same setup needs so as to access the best time usage (makespan) and so achieve the best low-cost.

Customers demand to attain their orders the best possible way. Thus, researchers seek and present to provide a satisfying model on stated matter. Obtaining an alternative machine obviously would shorten the total completion time however applying the most appropriate setup time scheduling is also important. A given customer order may contain more than one job assignments and may the request receiver have to accommodate all batch at the same time. Handling such cases with single machine is possible since it is much straight forward however when more than one job with different requirements takes place, a more complex and comprehensive method urges.

In addition to this, scheduling is crucial and is fundamental element on running given operations. For today's manufacturers and service providers, creating both the setup time and the costs plays a massive role on in terms of prompt delivery of customer orders with desired quality. In fact, it is schedule that creates the body of batch sizes, batch production durations and timings, setup needs and allocations of jobs to each machine. Thus, optimizing the schedules accordance to customer and job needs is a main course. On behalf of producing such schedules, it is very important to have a solid knowledge on setup time. Allahverdi and Soroush (2008), describes setup time as needed time to prepare the required resources to make a task. The advantage of reducing setup time can be sorted as increasing the production, reducing the expenses, rapid changeover, enhancing profitability and etc. The significance and advantage of including the setup times in scheduling survey has been searched since mid-1960s.

The purpose of this study is to get an optimum schedule by using two identical parallel machines so as to have the best time reduction in makespan while assigning customer orders. The enhanced developed model and the drawbacks along the model will be stated on upcoming chapters. Since the problem is very broad, the need of making assumptions on fields like; one machine can manufacture one job at a time, also preemption is not permitted, all order is distinct etc. arose in order to achieve convenient results. An in-depth analysis had to be conducted as our problem is classified as NP-Hard. In the interest of dealing with the problem among required assumptions, the researcher showed an approach to solve the case by using specific method called the constructive heuristic algorithm.

A heuristic algorithm is created as an alternative method to cover commonly known mathematical models' solutions more effectively and gives priority to speed to avoid time waste and high costs by disregarding optimality, precision or completeness to some extent. Even though, this problem could be handled with mathematical model entirely, it would be much more time consuming and would be more expensive. In addition, the heuristic algorithm has much better solutions for the large-sized problems, it is more convenient to manage such cases.

The evaluation process of the problem as this paper aims to discuss is as follows; Chapter 2 intends to define the problem with all aspects and offers an optimal schedule for makespan minimization with certain structural features. Chapter 3 investigates the literature about customer order scheduling and group scheduling. The solution approaches are in Chapter 4. In Chapter 5, we inspect the effectiveness of the mathematical model and the suggested heuristic algorithm that are used to form and back up our findings.

CHAPTER 2

PROBLEM DEFINITION AND PRELIMINARY SOLUTIONS

Here the problem is defined under some considerations and examined with the help of some properties of the optimal schedules for makespan minimization problems.

2.1 Problem Statement

There are different customers who create a set of orders to be manufactured on two identical parallel machines. In the problem settings there are two identical parallel machines are considered in this problem. In each order there can be one or more jobs which create a set of jobs that are also ready at time zero. All jobs in job lots have a fixed processing time and a fixed setup time. In addition, the sequence is independent therefore an independent setup is needed before processing for each job. A setup is required before processing procedure of the first job lot of a customer however if the beginning job is not the same as the last job lot of the immediately preceding customer order, there is no need for the setup. We observe two identical parallel machines problem to achieve a schedule of the customer orders along with their job lots minimizing makespan of the customer orders. In order to deliver each job lot at the same time to the customer all job lots should be processed. Additionally, a machine cannot run multiple jobs at a time rather can only process a job at a given time and also preemption cannot be utilized. In this work, it is assumed that every customer order composes of distinct jobs with onetime only processing and aligned sequence. Additionally, several identical jobs are not allowed in the same customer order. To simplify, we demonstrate the problem by following numerical example before continuing with our analysis.

Example 1 Consider a simple problem instance that has four customer orders. Order 1 has Job 1 and Job 3, Order 2 has Job 2 and Job 4, Order 3 has Job 1, Job 2, and Job

4, and Order 4 has solely Job 2. The processing times and setup times of these jobs are given below.

Table 1 Processing and setup times of all jobs in Example

Job	1	2	3	4
Setup time	2	1	4	3
Processing time	5	7	2	4

A feasible schedule of these four customer orders and four different jobs is for Machine 1, Order 3 and Order 4, for Machine 2 Order 2 and Order 1 which are demonstrated in the Figure 1 below however these orders are scheduled without considering the setup times.

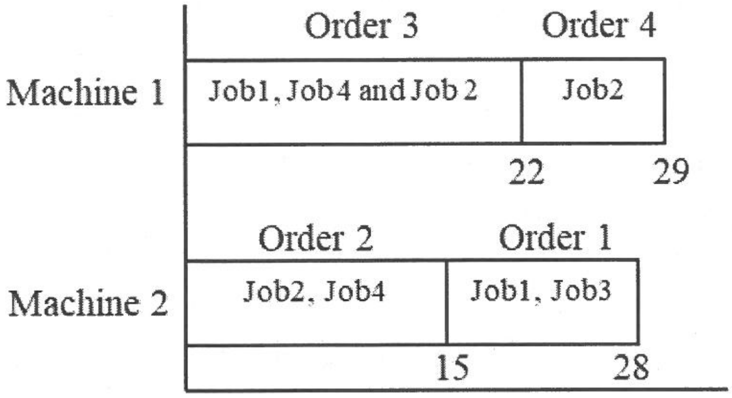

Figure 1 A feasible schedule of customer orders and jobs on the machines

Figure 1 exhibits the order schedules of the machines. The makespan of this schedule is 30 however this assignment does not have setup savings in calculation. When this consideration is thought, setup saving, while the orders are being assigned, the jobs of the orders should be considered for process order which means that which job is first or last processed in the same order must be noticed. Therefore the whole possible job positions can be thought, and the position possibilities of the jobs can be demonstrated in the following networks in Figures 2 and 3, the distance tables of the each machine is given below. In this problem the nodes are demonstrated as (k, O_i, J_j) which means that the matches of order-position assignment and indicates the last job to be processed in a given customer order which is explained in detail in Chapter 4.

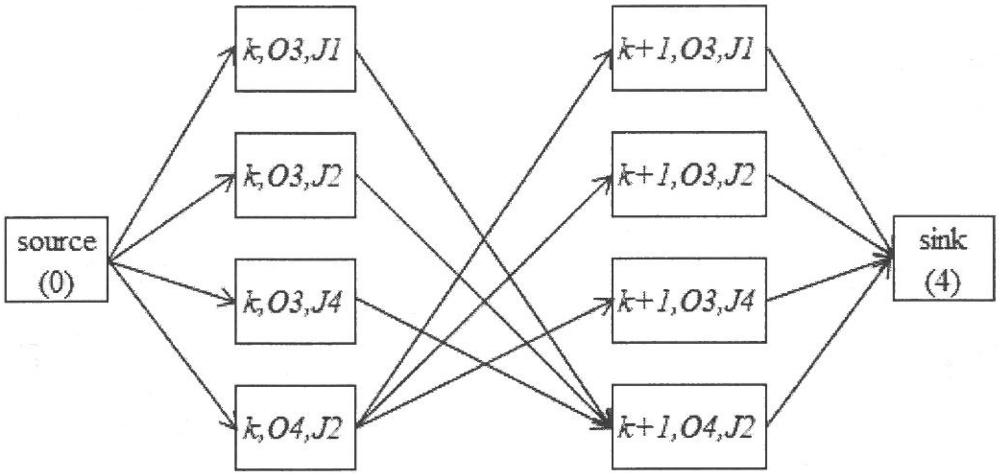

Figure 2 Shortest Path Network of the Machine 1

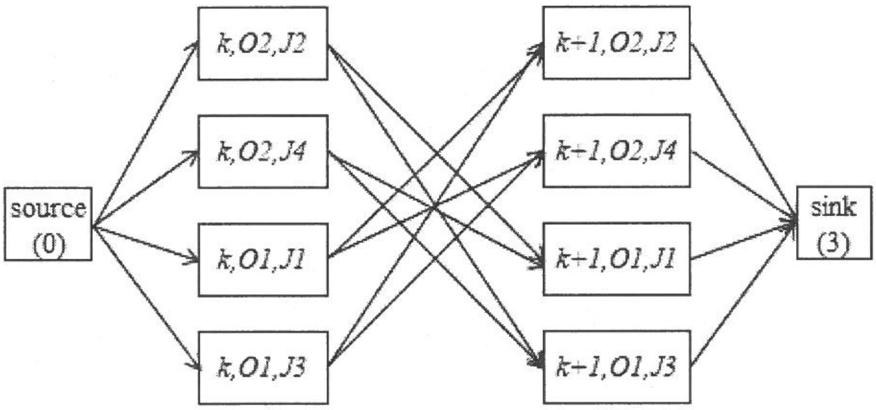

Figure 3 Shortest Path Network of the Machine 2

Figure 2 and Figure 3 demonstrate the shortest path network for the machines (Machine 1 and Machine 2). In Table 2, the arc distances (costs) are given for the shortest path network for Machine 1.

Table 2 The arc distances in the shortest path network for Machine 1 in Example 1

from \ to	*k+1,O3,J1*	*k+1,O3,J2*	*k+1,O3,J4*	*k+1,O4,J2*
k,O3,J1	-	-	-	8
k,O3,J2	-	-	-	8
k,O3,J4	-	-	-	8
k,O4,J2	22	21	22	-

Table 3 The arc distances in the shortest path network for Machine 2 in Example 1

from \ to	*k+1,O2,J2*	*k+1,O2,J4*	*k+1,O1,J1*	*k+1,O1,J3*
k,O2,J2	-	-	13	13
k,O2,J4	-	-	13	13
k,O1,J1	15	15	-	-
k,O1,J3	15	15	-	-

In the light of this information, the new schedules of the machines are shown in the following figure:

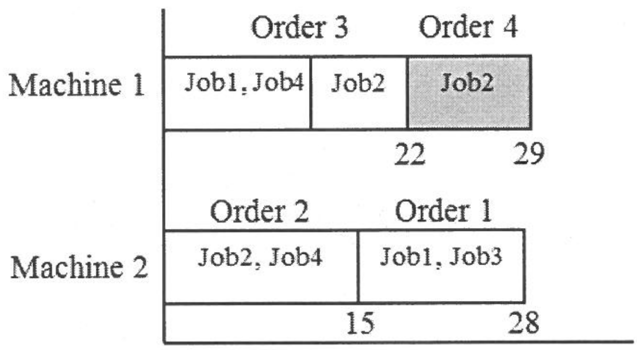

Figure 4 The feasible schedule of machines with setup saving

A feasible schedule of four customer orders for two identical parallel machines is demonstrated in Figure 4. As it is seen in this figure the makespan value is 29 time units. According to this figure, it is clear that there is no need for additional setup for job 2 for Order 4 due to the reason that the previous order's last job which is Order 3 is the same as the following order's job which is Job 2 of Order 4. On the other hand, in machine 2 whichever jobs of the order is processed first or last it does not end up with different completion time hence there is no common job for both Order 2 and Order 1. Consequently $C_{\max}$ is 29 for this example.

2.2 Some Observations

Let's explain some descriptions before debating our examinations.

Description 1 *Total time* (TT_i) is the summation of setup times and processing times of the all jobs in the each customer order O_i , i.e. $TT_i = \sum_{J_j \in O_i} (s_j + p_j)$.

Description 2 *The Longest Processing Time rule* (LPT) allows jobs to be ordered according to their longest total processing time to their smallest total processing time.

It is, however, possible to simplify the problem if setup times are disregarded while processing the customer orders with same or different job orders. This also avoids possible makespan minimization problems as the sequence of customer order progress would eventually ends up with the same objective value. In addition, since setup times are eliminated from the customer order progress the total completion of the customer orders only indicate only the process of each order.

2.3 Some Structural Properties for the Optimal Schedules

Here some structural properties for the optimal schedules which are going to be used for the improvement of the result for the makespan minimization problems.

Property 1 For the makespan minimization problem there exists an optimal schedule without an idle times on both machines.

Proof If there is an idle time on the parallel machines, the following customer orders together with jobs might be changeded to the left on the machines without raising the objective of the valid schedule.

The set of customer orders which is demonstrated as O is able to be separated into two disjoint sets which are O' and O'' ($O = O' \cup O''$ and $O' \cap O'' = \varnothing$) and the set is created from customer orders which have not similar job with other orders, and its supplement set O'' is created from the rest of the customer orders.

2.4 Some Special Cases Solvable by Priority Rules

Case 1: In each customer order there is only one job, and whole jobs are dissimilar.

Theorem 1: *If there is one job for every customer order which is distinct then at any time each sequence of orders is optimal for each machine.*

Proof If there is only one job in each order, and also these are dissimilar, at that time a setup is indicated before any job is able to be processed. For both Machine 1 and Machine 2 the orders are sequenced according to their summation of setup and processing time of the job. The makespan time is calculated as $C_{\max} = \max\{C_{\max}(m_1), C_{\max}(m_2)\}$.

Case 2: There are two dissimilar jobs in every customer order which are available in every customer orders.

Theorem 2: *If every customer order has two dissimilar jobs (such as job1(J1) and job2 (J2)) which are available in each customer orders, at that time job with the minimum (maximum)setup time is processed first position (second position) of the each customer order in the optimal schedule.*

Proof Here each customer order has two dissimilar jobs, there is no importance of the sequence of orders, the only thing changes is that which order is going to be processed in which machine (Machine 1 or Machine 2). Hence, these jobs are common for the each customer orders. Whether or not *J1* or *J2* is processed in the first position ought to be decided at first due to the reason that if the first job of the first customer order is *J1*, the second order of the first job should be *J2* because there is no need to be prepared to setup therefore there is a setup saving for the second order of the first job. This procedure is repeating for both machines. The result of this schedule is as follows for the both machines;

$$\text{Schedule 1 for the Machine 1}: O_1(J_1 - J_2) - O_3(J_2 - J_1) - O_5(J_1 - J_2) - \ldots$$

$$\text{Schedule 1 for the Machine 2}: O_2(J_1 - J_2) - O_4(J_2 - J_1) - O_6(J_1 - J_2) - \ldots$$

Likewise, if the initial job of the initial customer order is *J1*, the second order of the first job should be *J2* because there is no need to be prepared to setup therefore there is a setup saving for the second order of the first job, again this procedure is repeating for both machines. The result of this schedule is as follows for the both machines;

$$\text{Schedule 2 for the Machine 1: } O_2(J_1 - J_2) - O_4(J_2 - J_1) - O_6(J_1 - J_2) - ...$$

$$\text{Schedule 2 for the Machine 2: } O_1(J_1 - J_2) - O_3(J_2 - J_1) - O_5(J_1 - J_2) - ...$$

Also this logic can be considered for the job exchanging. That is, the first schedule can be changes as follows;

$$\text{Schedule 1 for the Machine 1: } O_1(J_2 - J_1) - O_3(J_1 - J_2) - O_5(J_2 - J_1) - ...$$

$$\text{Schedule 1 for the Machine 2: } O_2(J_2 - J_1) - O_4(J_1 - J_2) - O_6(J_2 - J_1) - ...$$

As it can be proved that for both schedules the result of the makespan value is equal even if the makespan time of the machine and the starting order number change. The most significant thing in these schedules is which job's setup time has greater setup time value. Hence if the job having greater setup time value is the last job of the first order, then the second order of the first job's setup time saving is going to be bigger.

CHAPTER 3

LITERATURE REVIEW

In production fields the scheduling problems are classified in accordance with shop structure, job types and performance measures. There are different shop structures in scheduling researches yet especially job shop, single machine, flow shop and parallel machine issues are investigated in the literatures. Efficiency measures which are noticed in these studies vary relying on the shop structure. For example, an ascending number of surveys regarding that intend to diminish the maximum lateness, count of tardy jobs, maximum tardiness, completion time, makespan and so on. Hereby distinct performance measures and characteristics are taken in consideration in accordance with implementations in production system.

Recently, diversity of the problems has increased in the count of studies. The plurality of scheduling studies supposes setup as a part of the processing time or inconsiderable. This influences the examination quality for lots of implementations which need obvious behavior of setup. Such implementations, paired with the appearance of production concepts like the group technology and time based competition.

Setup covers work to arrange the machine, order or process for parts or cycle. This contains positioning operation in process material, achieving the devices, cleaning up, adjusting the devices and etc. In the research for long time the setup time / cost has been regarded negligible, neglected or thought as a part of the processing time for the subject of setup time. As this might be confirmed for some scheduling problems, lots of other conditions need for divisible setup time / cost thought. For divisible setup time / cost two different problem types are available. In the first one, setup solely relies on the job to be processed hereby this is called as sequence independent, on the other hand setup relies on both job to be processed and immediately preceding job hereby this is named sequence dependent. For example Bitran and Gilbert (1989) address a widespread practical case of m machine sequencing problem, and the aim is minimizing the total setup times, and the number of large setups. In our study as it is mentioned before the setup times of the job to be processed and immediately preceding job are related each other therefore here sequence dependent is considered. In 1999 firstly Allahverdi *et al.* prepared a review paper for involving the setup considerations. In years the importance of setup times has been searched in various studies. In 2008 Allahverdi *et al.* widen the survey on scheduling problems with an apart setup times / costs, and it is a review of scheduling study which includes setup consideration. For minimizing the setup times or costs different scheduling policies have been improved by searchers. There is a helpful procedure for reducing setup time even setup time is sequence dependent or not by *batching* which is basically grouping the jobs with same content in one lot. For this reason, there has been considerable concern in scheduling problems involving batching matter. Researches with setup times understandably show the benefit of batching and technology supposal. The problem to be considered can be divided in to two groups as group scheduling and customer order scheduling. For parallel machine problems the most related study to our subject on customer order scheduling and group scheduling is reviewed shortly in this chapter.

According to these considerations scheduling can be categorized as the following figure for our study. Therefore our problem can be classified as makespan minimization of parallel machines sequence dependent problem.

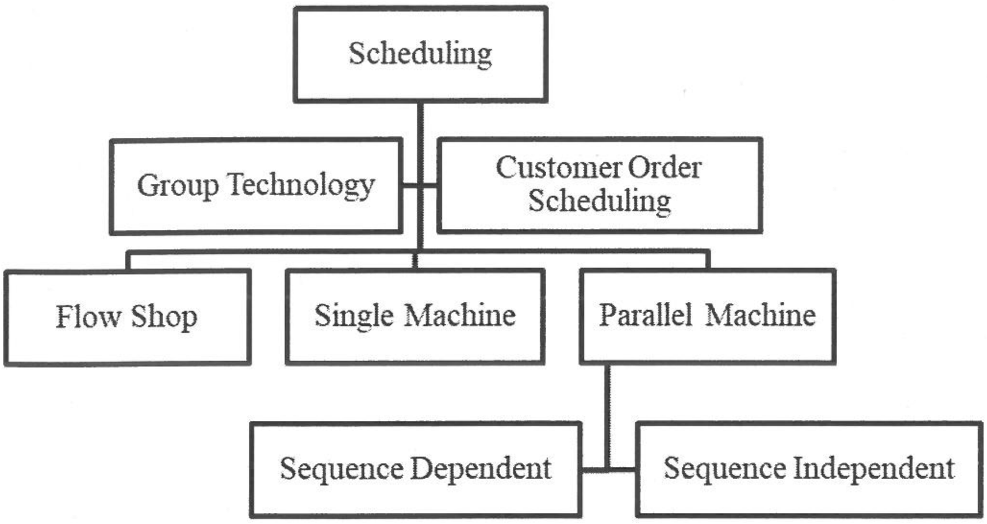

Figure 5 The classification of scheduling problem

3.1 Group Scheduling

The evaluation of the first scheduling consideration, group scheduling, is review however the illustration of group technology ought to be explained first. *Group technology* is defined as an attitude to producing and engineering management which searches to overcome the adequacy of great volume production by using resemblances of various products and operations in the production or implementation.

Brucker and others considers batch scheduling with deadlines on parallel machines in 1998. There are *G* groups consisting identical several jobs on m parallel machines. To find splitting of groups in to batches and batche scheduling on the machines have to be made in the problem. Sequence dependent machine setup time is supposed immediately before batch of group is produced. The aim is to reach schedule which is feasible in terms of deadlines.

Single machine scheduling problems is considered by Mingbao and Shijie (2005) in group technology. The aim of this paper is minimizing the makespan and total weighted completion time, hence any optimal condition and heuristic algorithms are proposed, and proving that total weighted completion time is NP hard in the problem.

Shen *et al.* address scheduling problem on parallel machine, and the aim is minimizing the total weighted completion time in which the product families are included. Also major setup happens when the processing the jobs in different families, and in addition the sequence dependencies are considered. Batching jobs of the same family can be implemented so that avoiding redundant setup.

3.2 Customer Order Scheduling

Other classification of the scheduling, *Customer Order Scheduling*, is other relevant field to our problem environment in prospect. Yang *et al.* (2005) consider parallel machines scheduling problem, and the aim is that the sum of completion times of customer order is minimized. Here the structure of the jobs in the batches is prespecified, and also no setup time exists between distinct jobs or distinct bathes. For the parallel machine case they analyze two heuristic algorithms that are depend on basic scheduling rules for he large problems, and the solutions of the methods are close to zero.

Yang (2005) presents the complexness of different problems with distinct types of objectives, job limitations in customer orders scheduling problem with multiple machines. The problem is distinct from many other batch scheduling problem due to the reason that the objective function is related with the completion time of batch on behalf of each jobs' completion times.

Minimizing the maximum lateness is considered for customer order scheduling problem by Su *et al.* (2013). The customer order problem in which jobs are scheduled on the parallel machines and send in batches is studies in this paper. Three heuristics which depends on simple scheduling rules are suggested. The aim of this study is to develop heuristics and examine their worst cases bounds. In first heuristic (H1) the batch earliest due date (EDD) dispatching rule is approached to decide the batch sequence, and LPT rule is performed to decide the job sequence in each batch. In second heuristic (H2) the batch EDD dispatching rule is performed to decide the batch sequence however the order of jobs in every batch is arbitrary. In the last heuristic (H3) again the batch EDD dispatching rule is applied to decide the batch sequence however the LPT rule is then applied to appoint jobs to the first available machine. At the end H1 and H3 are compared and the average deviation for these heuristics from the lower bound is 0.724% and 0.722%. The H2is omitted for brevity because it is similar to H3.

In 2005 Leung *et al.* prepared about order scheduling in an environment. Here there are n orders and m machines in parallel and each machine can produce one specific product type. The researchers demonstrate that minimization of the total completion time when $m \geq 3$ makes the problem strongly NP hard. The main point is the sequence where the various orders are processed on the different machines.

Jia and Mason (2008) investigate into the identical parallel machine which includes various orders per job scheduling problem whose aim is to minimize total weighted completion time. They examine two types of processing fields which are parallel and serial processing. Mixed integer program is suggested first and heuristic algorithm solution attitudes are improved for three main ways of the scheduling problem under search: assignment of job to machine, order assignment to job, and order sequencing.

Leung *et al.* (2008) are developed a great interest the scheduling orders problem for various product types in plant with a number of parallel machines. Every order needs for significant amounts of several varied product types which are able to produce simultaneously. In paper there are two facts; first one is each product type is able to be manufactured on one and only machine that is assigned to that product, all machines are flexible and identical for other case, and each product type is able to manufactured by any of the machine. Moreover, setup time is needed; every order includes release date and weight. In the research the total weighted completion time of the orders try to minimize. For the first case the heuristic algorithms cover different priority rules along with two LP based algorithms. For the flexible case the heuristic algorithms regarded as two classes which are the consecutive and the dynamic two phase heuristic.

Single machine is regarded to reduce makespan and total transportation cost of the orders in which the jobs are from diverse sets within dependent class setup times, and customer orders result from only one job from every classes by Gupta *et al.* (1997). They suggest constructive polynomial time algorithms for two hierarchic problems.

In our study the problem is about customer orders scheduling on two identical parallel machines with setup times. According to the classification of the scheduling problem our problem can be considered as customer order scheduling on two identical parallel machines with sequence independent setup times. As we have seen from the literature review, the problem that we have has not been studied before.

CHAPTER 4

SOLUTION APPROACHES

Here two identical parallel machines order scheduling problem is examined for minimizing makespan of the customer orders. The major choice is to decide the sequence of the orders in addition to this, the first and last job are considered for each order.

First MILP model is suggested to achieve the optimal or near optimal solutions for problem instances, after that according to the optimal results of the some polynomial time solvable cases are dedicated. The shortest path formulation is described by constructing a cost network, and finally the heuristic algorithm is proposed for this problem.

4.1 Mathematical Model of the Problem

The model given below is adapted from Akkocaoğlu (2014). The parameters, indices, sets and decision variables are used in our mathematical model are as follows:

Parameters, sets and indices

K Number of customer orders

i Index for customer orders ($i = 1,2,...,K$)

N Number of jobs

j Index for jobs ($j = 1,2,...,N$)

N_i Set of jobs in customer order O_i

D_{ij} $D_{ij} = 1$ If customer O_i has job J_j; or else, $D_{ij} = 0$

M The set of customer orders that have more than one job to be processed

p_j Processing time for job J_j

s_j Setup time for job J_j

m Index for machines ($m = 1,2$)

k The customer orders' position index in the sequence on a machine

($k = 1,2,...,K$)

Utilizing p_j and s_j, we calculate the total (summation of setup and processing) time of all jobs in customer order O_i and the setup time between two sequential jobs as

TT_i = Total (summation of setup and processing) time of all jobs in order = $\sum_{J_j \in O_i} (s_j + p_j)$

ST_{hj} = Setup time between jobs J_h and J_j (where $j \neq h$) = $\begin{cases} s_j & \text{if job } J_j \text{ immediately follows job } J_h \\ 0 & \text{otherwise} \end{cases}$

Decision variables

$$X_{ikm} = \begin{cases} 1 & \text{if customer order } O_i \text{ is assigned to position } k \text{ on machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$F_{ijkm} = \begin{cases} 1 & \text{if job } J_j \text{ is the beginning job in customer order } O_i \text{ assigned to position } k \text{ on machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$L_{ijkm} = \begin{cases} 1 & \text{if job } J_j \text{ is the ending job in customer order } O_i \text{ assigned to position } k \text{ on machine } m \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ihjlkm} = \begin{cases} 1 & \text{if both } L_{ijkm} \text{ and } F_{ijlk+1m} \text{ are equal to 1 on machine } m \text{ (i.e., last job of a customer order and the first job of the immediately following customer order are not same and } j \neq h) \\ 0 & \text{otherwise} \end{cases}$$

RT_{ikm} = Realized total (summation of setup and processing) time of customer order O_i appointed to position k on machine m

$C_{\max}$ = Time to complete all customer orders (makespan)

MILP model

Minimize $C_{\max}$ (1)

Subject to $\sum_{m=1}^{2}\sum_{k=1}^{K} X_{ikm} = 1$ for $i = 1,2,...,K$ (2)

$\sum_{i=1}^{K} X_{ikm} \leq 1$ for $k = 1,2,...,K$; $m = 1,2$ (3)

$\sum_{i=1}^{K} X_{ik+1m} \leq \sum_{i=1}^{K} X_{ikm}$ for $k = 1,2,...,K-1; m = 1,2$ (4)

$\sum_{i=1}^{K}\sum_{j \in N_i} F_{ijkm} = \sum_{i=1}^{K} X_{ikm}$ for $k = 1,2,...,K$; $m = 1,2$ (5)

$\sum_{i=1}^{K}\sum_{j \in N_i} L_{ijkm} = \sum_{i=1}^{K} X_{ikm}$ for $k = 1,2,...,K$; $m = 1,2$ (6)

$F_{ijkm} \leq D_{ij} X_{ikm}$ for $i = 1,2,..,K$; $j \in N_i$; $k = 1,2,...,K$; $m = 1,2$ (7)

$L_{ijkm} \leq D_{ij} X_{ikm}$ for $i = 1,2,..,K$; $j \in N_i$; $k = 1,2,...,K$; $m = 1,2$ (8)

$L_{ihkm} + F_{ijk+1m} - 1 \leq Y_{ihjlkm}$ for $i = 1,2,..,K$; $j \in N_i$; $h \in N_i$; $j \neq h$;

$l = 1,2,..,K; l \neq i; k = 1,2,...,K$;
$m = 1,2$ (9)

$F_{ijkm} + L_{ijkm} \leq 1$ for $i \in M$; $j \in N_i$; $k = 1,2,...,K$;
$m = 1,2$ (10)

$RT_{i1m} \geq TT_i X_{i1m}$ for $i = 1,2,...,K$; $m = 1,2$ (11)

$RT_{ikm} \geq TT_i X_{ikm} - \sum_{j=1}^{N} s_j F_{ijkm} + \sum_{h=1}^{N}\sum_{j=1}^{N}\sum_{l=1}^{K} ST_{hj} Y_{ihjlk-1m}$ for $i = 1,2,..,K$; $k \geq 2$
; $m = 1,2$ (12)

$C_{\max} \geq \sum_{i=1}^{K}\sum_{k=1}^{K} RT_{ikm}$ for $m = 1,2$ (13)

$$C_{\max} \geq 0,\ RT_{ikm} \geq 0 \qquad \text{for } \forall i,k,m \tag{14}$$

$$X_{ikm}, F_{ijkm}, L_{ijkm}, Y_{ihjlkm} \in \{0,1\} \quad \text{for } \forall h,i,j,k,l,m \tag{15}$$

The given MILP model above, the objective in (1) is to minimize the makespan which is equal to the maximum completion time of customer orders. Constraint set (2) ensures that each customer order is assigned to only one position among the available ones on the machine. Constraint set (3) ensures that each position in the sequence on a machine can be occupied by at most one customer order. Constraint set (4) guarantees that a position on a machine cannot be occupied if the previous position on the same machine is not occupied by a customer order. Constraint series (5) and (6) assure solely one job in every customer order can be operated as the first or last job in its own customer order on every machine, separately. Constraint sets (7) and (8) guarantee that a job is not able to be the beginning or ending job of a customer order allocated to a position for each machine if this customer order does not involve the job. Constraint set (9) fulfills the circumstance that no setup time is required before the processing of the first job of the order is identical as the ending job of the immediately preceding customer order on each machine. Constraint set (10) satisfies that each job in a customer order can be the first, immediate or last job of this customer order. Constraint sets (11) and (12) describe the realized total (summation of setup and processing) time of the customer orders assigned to the first and another position, respectively for each machine. Makespan is determined by the constraint set (13). Constraint sets (14) and (15) respond non-negativity and binary restrictions on the decision variables, respectively.

4.2 Shortest Path Network

The shortest path model is used to get a path for each machine in the proposed heuristic algorithm in this study. A group of nodes and arcs which are connecting certain pairs of nodes compose of the shortest path network. A shortest path between two nodes in the network is a path with minimum number of edges as long as the cost of each edge is the same. In the network a group of node covers:

- two vertices which are from dummy source (beginning) node demonstrated as 0, to the dummy sink (ending) node (K+1),

- node (k, O_i, J_j), $k = 1,...,K$, $i = 1,...,K$, $\forall J_j \in O_i$: This node clarifies, the matches of order-position assignments and indicates the last job to be processed in a given customer order.

For example, *(1,O3,J1)* means that Order 3 is allocated to the first position in the customer orders and the last processed job in this order is J1.

In the previous chapter there is an example (Example 1). In this example there are four customer orders, each order has different jobs that are Order1 has Job1, Order2 has Job 2 and Job 4, Order 3 has Job1, Job 2 and Job 4, and Order 4 has Job 2. Also the processing times and setup times of all jobs are given Table 1. According to this example shortest networks of the each machine are demonstrated in the following figures in order to explain the shortest path network of the machines.

With reference to the schedule of machines in Example 1 the following networks of machines are formed.

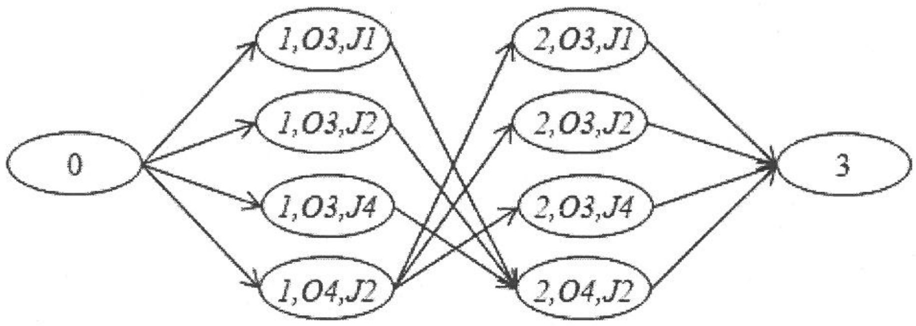

Figure 6 Shortest Path Network of Machine 1 for Example 1

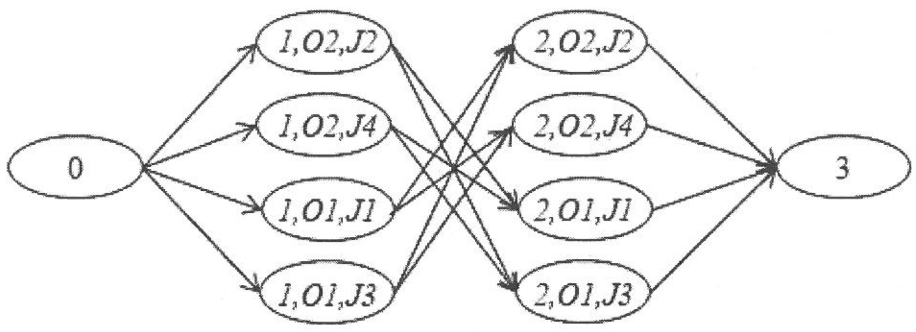

Figure 7 Shortest Part Network of Machine 2 for Example 1

Each position of the shortest path network, N_T which is the total number of jobs in whole orders and N_i is the number of the jobs of the order i are demonstrated $N_T = \sum_{i=1}^{K} N_i$. The oriented arc set is produced like this;

- The flow cost of the arc which is from the beginning node (0) to node $(1, O_i, J_r)$ is $\sum_{J_j \in O_i} (s_j + p_j)$.
- The flow cost $\sum_{J_j \in O_i} (s_j + p_j)$ of the arc which is from node (k, O_i, J_c) to the node $(k+1, O_l, J_c)$, in which $k = 1, \ldots, K-1$, $i \neq l$, and J_c is a shared job of orders O_i and O_l.
- The flow cost of an arc from node which is (k, O_i, J_r) to node $(k+1, O_l, J_v)$ is $\sum_{\substack{J_j \in O_l \\ j \neq r}} s_j + \sum_{J_j \in O_l} p_j$, in which $k = 1, \ldots, K-1$, $i \neq h$, $r \neq v$.
- The flow cost of an arc which is from node (K, O_i, J_r) to the ending node $(K+1)$ is zero.
- The maximum possible number of nodes is $(K \times N_T) + 2$.

In the following table the flow costs between the network nodes for the problem which is Example 1 for the Machine 1 and Machine 2 are demonstrated.

Table 4 The flow costs of Example 1 for Machine 1

to / from	*k+1,O3,J1*	*k+1,O3,J2*	*k+1,O3,J4*	*k+1,O4,J2*
k,O3,J1	-	-	-	8
k,O3,J2	-	-	-	8
k,O3,J4	-	-	-	8
k,O4,J2	22	21	22	-

Table 5 The flow costs of Example 1 for Machine 2

to / from	*k+1,O2,J2*	*k+1,O2,J4*	*k+1,O1,J1*	*k+1,O1,J3*
k,O2,J2	-	-	13	13
k,O2,J4	-	-	13	13
k,O1,J1	15	15	-	-
k,O1,J3	15	15	-	-

4.3 Heuristic Algorithm

Even if the complexity of the problem that we have is open, the problem is most likely NP-hard. To obtain the optimal result for the problem the mathematical model is constructed however for the large sized problems the optimal solution could not be handled in the limited time condition. Therefore, depending on the size of the problem and the CPU time we try to satisfy to compute approximate solutions. It is known that the algorithms usually find a solution which is close to optimality and also faster. Therefore we developed a fast algorithm that provides optimal solution or near-optimal solution acceptable computational times.

The heuristic algorithm that we proposed has four steps. In the first step, for each customer order the Longest Total Time (LTT) is calculated without setup savings. After calculating the LTT for each group allocation to the machines is realized. While assigning the orders to the machines, order with the longest total time has to dedicate to the machine which has the minimum total completion time until the all customer orders is allocated. This step is significant for assigning the customer orders to machines. According to the total times of the customer orders assignment is realized to the machines. After assignment is completed the sequence for each machine has been created. In order to achieve the path for the machines the next step is applied. In Step 2, the arc distance between nodes which represents the position of the jobs for each customer order is determined for each machine. Here, the most significant thing that should be thought that if the last processed job in the order is as the same as for the preceding first job of the order in the network, there is no additional setup time for the preceding first job of the order. In Step 3, the Shortest Path Model is solved for each machine. In the last step, according to the result of the Step 3, the completion time is calculated for each machine; then finally is decided. Therefore, the heuristic can be considered in two phases. In the first phase includes an assignment of orders to the machines. Second phase is solving each machine by shortest path. The stepwise specification of the suggested heuristic algorithm is as follows:

Step 1: Apply the LTT-list for the customer order allocations for the machines. (The initial schedule constitution is achieved without setup savings.)

i. The total time is calculated for every customer order O_i, the total time calculation is as follows;
$$TT_i = \sum_{J_j \in O_i} (s_j + p_j)$$

ii. According to the total times of the customer orders the allocation is realized to the machines. After assignment is completed, the sequence for each machine has been created.

Step 2: The arc distances for the shortest path network on machines are calculated.

Step 3: Solve the shortest path model for each machine.

i. The following simple integer LP model is used to solve to get the shortest path of the networks for the machines.

The following parameters, indices, sets and variables are utilized in this Shortest Path model:

k Index of customer orders ($k = 1,2,...,K$)

f , t indexes for the total number of jobs ($f = t = 1, 2,..., n$)

c_{ft} the cost flow from node f to node t

x_{ft} indicator variable for whether edge (f, t) is a part of shortest path

A_{tk} is a $N \times K$ binary matrix

$$A_{tk} = \begin{cases} 1 & \text{if customer order } k \text{ is on node } t \\ 0 & \text{otherwise} \end{cases}$$

Minimize $\sum_{f=0}^{n} \sum_{t=0}^{n} c_{ft}\, x_{ft}$ (1)

Subject to $\sum_{t=0}^{n} x_{ft} - \sum_{t=0}^{n} x_{tf} = \begin{Bmatrix} 1 & if & f = 0 \\ 0 & if & f \neq 0 \text{ or } K+1 \\ -1 & if & f = K+1 \end{Bmatrix}$ (2)

$$\sum_{f=0}^{n} \sum_{t=1}^{n} x_{ft} A_{tk} \leq 1 \quad \forall k \tag{3}$$

$$x_{ft} \in \{0,1\} \quad \forall f, t \tag{4}$$

The given shortest path model above, the objective function in (1) is to minimize the cost flow of the network. Constraint set (2) ensures the flow balance equation in the network. Constraint set (3) guarantees that each customer order on the node can be used once. Constraint set (4) responds binary restriction on the decision variable.

It is known that the model that is given above is completely unimodular, and the arc-node coincidence matrix related with the conservation equations. In addition to this there is at least one optimal result to LP relaxation of the mathematical model where almost every decision variables are integer such as $x_{ft} = 0$ or 1. To do this the LP relaxation is found by replacing with $x_{ft} \in \{0,1\}$ by $x_{ft} \geq 0$. Even if the shortest path model is a binary model, model can be solved as LP.

Step 4: Determine the $C_{max} = \max\{C_{max}(m_1), C_{max}(m_2)\}$

i. where $C_{max}(m_1)$ is the makespan of the machine 1, $C_{max}(m_2)$ is the makespan of the machine 2.

Example 2: There are five customer orders. In each customer orders there are a couple of jobs which are different from each other. The setup times and the processing times of this jobs and the total time of the customer orders are demonstrated in the following tables.

Table 6 Setup and Processing Times of the Jobs

	Job 1	Job 2	Job 3	Job 4	Job 5
Setup Time	25	28	31	30	31
Processing Time	6	6	8	1	5

These customer orders have different jobs, and in the following table which order has which job(s) and the total time of these orders are demonstrated.

Table 7 List of Jobs to be processed and Total Time of the Customer Orders

	Jobs to be processed	Total Time of Orders
Order 1	Job3, Job4, Job5	106
Order 2	Job1, Job2, Job3, Job4, Job5	171
Order 3	Job1, Job3, Job4, Job5	137
Order 4	Job1, Job3, Job5	106
Order 5	Job1, Job3	70

Step 1: Apply the LTT-list for the group allocations for the machines.

LTT-list: Order 2, Order 3, Order 1 or Order 4, Order 5

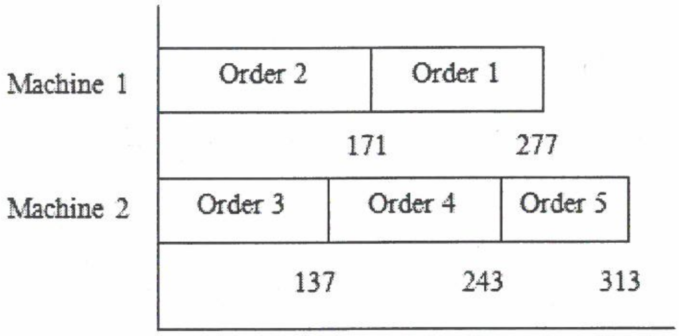

Figure 8 Order allocation to the machines

C_{max} : 313 (without setup savings)

Machine 1 schedule list: Order 2, and Order 1

Machine 2 schedule list: Order 3, Order 4, and Order 5

Step 2: For both machine the arc distances for both networks is decided.

According to these schedules the arc distances for both networks is given in the following figures.

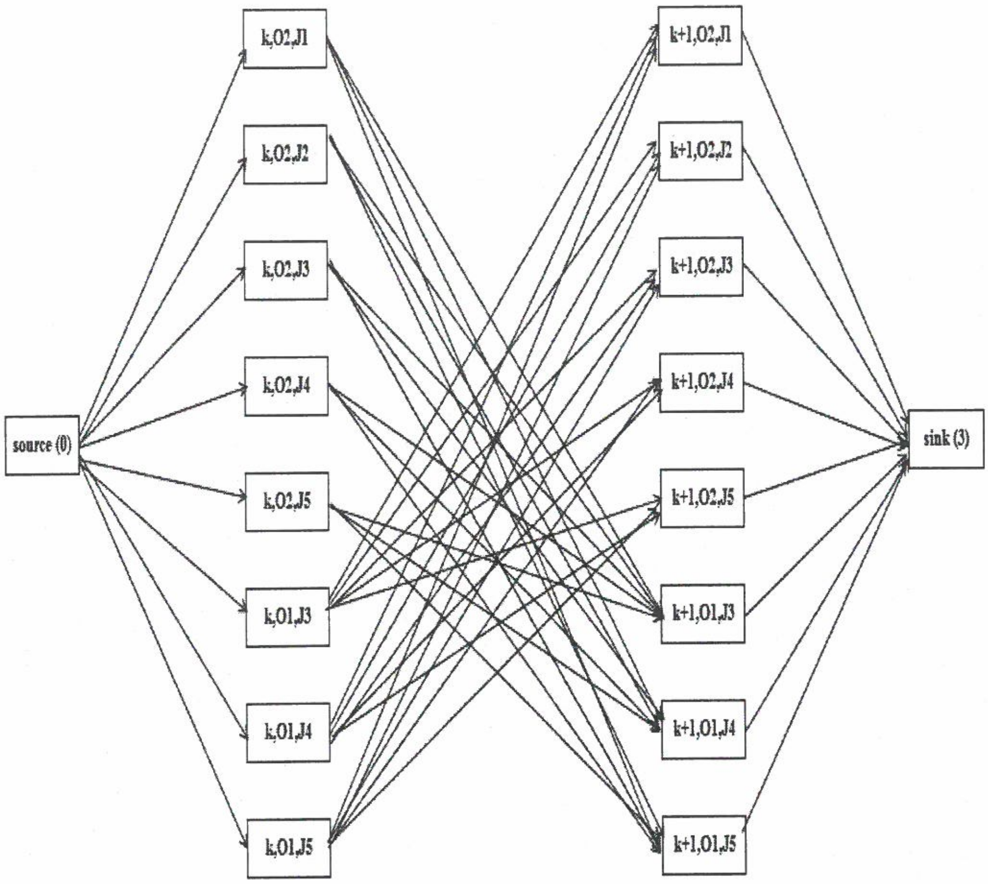

Figure 9 Shortest Path Network for Machine 1

As it is seen in the Table 8 each arc distances are calculated for Machine 1.

Table 8 Arc distances for the shortest path network on Machine 1

	source	k O2J1	k O2J3	k O2J4	k O2J2	k O2J5	k O1J4	k O1J5	k O1J3	k+1 O2J1	k+1 O2J3	k+1 O2J4	k+1 O2J2	k+1 O2J5	k+1 O1J4	k+1 O1J5	k+1 O1J3	sink
source	0	171	171	171	171	171	106	106	106	0	0	0	0	0	0	0	0	0
k O2J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	0
k O2J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75	106	0
k O2J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	76	76	0
k O2J2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	0
k O2J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	106	75	0
k O1J4	0	0	0	0	0	0	0	0	0	141	141	171	141	141	0	0	0	0
k O1J5	0	0	0	0	0	0	0	0	0	140	140	140	140	171	0	0	0	0
k O1J3	0	0	0	0	0	0	0	0	0	140	171	140	140	140	0	0	0	0
k+1 O2J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O2J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O2J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O2J2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O2J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O1J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O1J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O1J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sink	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

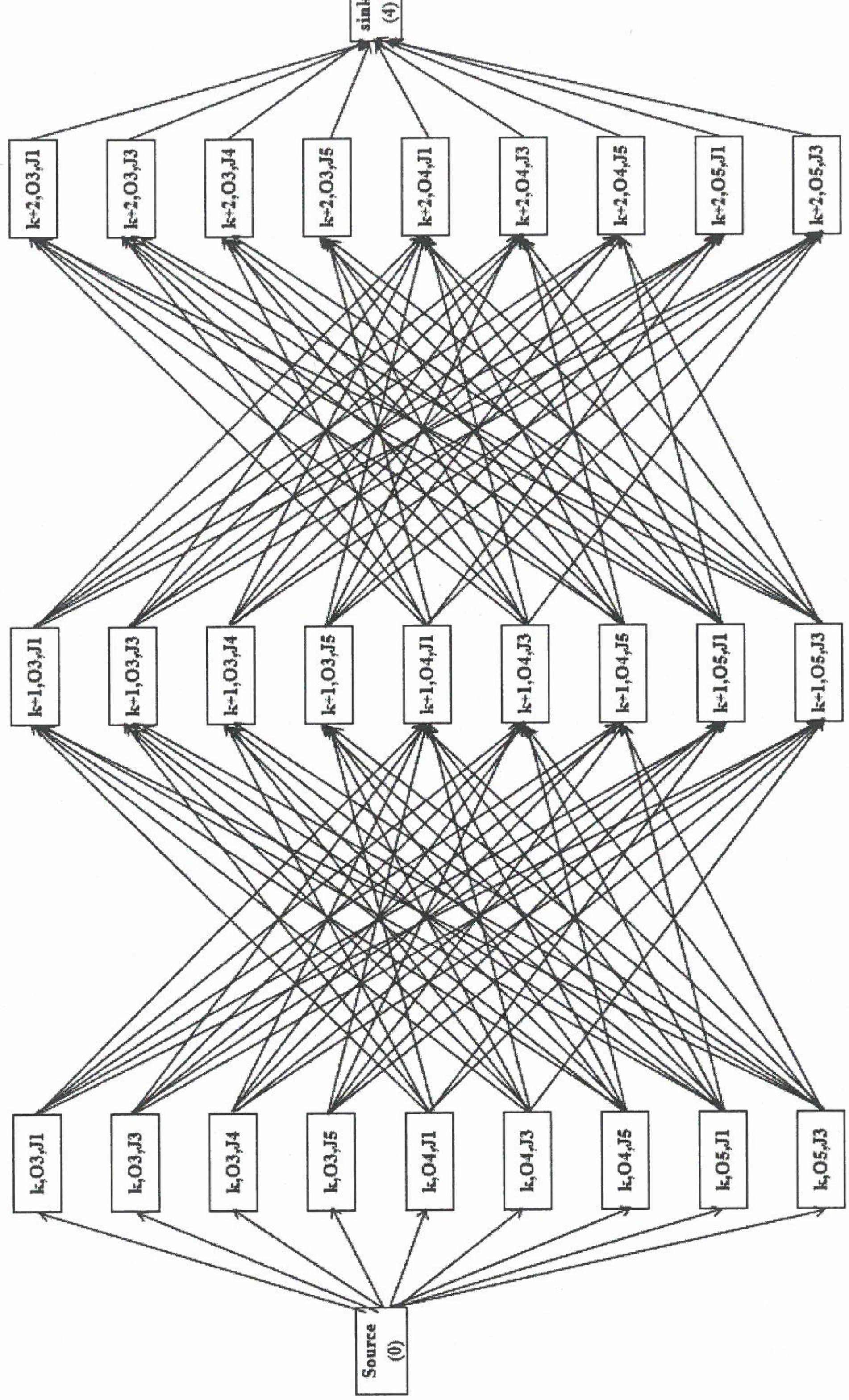

Figure 10 Shortest Path Network for Machine 2

	source	kO3J5	kO3J4	kO3J1	kO3J3	kO4J1	kO4J3	kO4J5	kO5J3	kO5J1	k+1 O3J5	k+1 O3J4	k+1 O3J1	k+1 O3J3	k+1 O4J1	k+1 O4J3	k+1 O4J5	k+1 O5J3	k+1 O5J1	k+2 O3J5	k+2 O3J4	k+2 O3J1	k+2 O3J3	k+2 O4J1	k+2 O4J3	k+2 O4J5	k+2 O5J3	k+2 O5J1	sink
source	0	137	137	137	137	106	106	106	70	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k O3J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75	106	70	70	0	0	0	0	0	0	0	0	0	0
k O3J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	70	70	0	0	0	0	0	0	0	0	0	0
k O3J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	81	81	45	70	0	0	0	0	0	0	0	0	0	0
k O3J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	106	75	70	39	0	0	0	0	0	0	0	0	0	0
k O4J1	0	0	0	0	0	0	0	0	0	0	112	112	137	112	0	0	0	45	70	0	0	0	0	0	0	0	0	0	0
k O4J3	0	0	0	0	0	0	0	0	0	0	106	106	106	137	0	0	0	70	39	0	0	0	0	0	0	0	0	0	0
k O4J5	0	0	0	0	0	0	0	0	0	0	137	106	106	106	0	0	0	70	70	0	0	0	0	0	0	0	0	0	0
k O5J3	0	0	0	0	0	0	0	0	0	0	106	106	106	137	75	106	75	0	0	0	0	0	0	0	0	0	0	0	0
k O5J1	0	0	0	0	0	0	0	0	0	0	112	112	137	112	106	81	81	0	0	0	0	0	0	0	0	0	0	0	0
k+1 O3J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75	106	70	70	0
k+1 O3J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	70	70	0
k+1 O3J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	81	81	45	70	0
k+1 O3J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	106	75	70	39	0
k+1 O4J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	112	112	137	112	0	0	0	45	70	0
k+1 O4J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	137	0	0	0	70	39	0
k+1 O4J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137	106	106	106	0	0	0	70	70	0
k+1 O5J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	106	106	137	75	106	75	0	0	0
k+1 O5J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	112	112	137	112	106	81	81	0	0	0
k+2 O3J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O3J4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O3J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O3J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O4J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O4J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O4J5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O5J3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k+2 O5J1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sink	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11 Arc distances for the shortest path network on Machine 2

As it is seen in the Figure 11 each arc distances are calculated one by one for Machine 2.

Step 3: Solve the shortest path model for each machine according to the arc distances in the networks for each machines.

The shortest path for Machine 1: $1, O_1, J_5 \rightarrow 2, O_2, J_1$

The completion time of Machine 1: 246

The shortest path for Machine 2: $1, O_4, J_5 \rightarrow 2, O_3, J_3 \rightarrow 3, O_5, J_1$

The completion time of Machine 2: 251

The final sequences of the machines are demonstrated in the following figures:

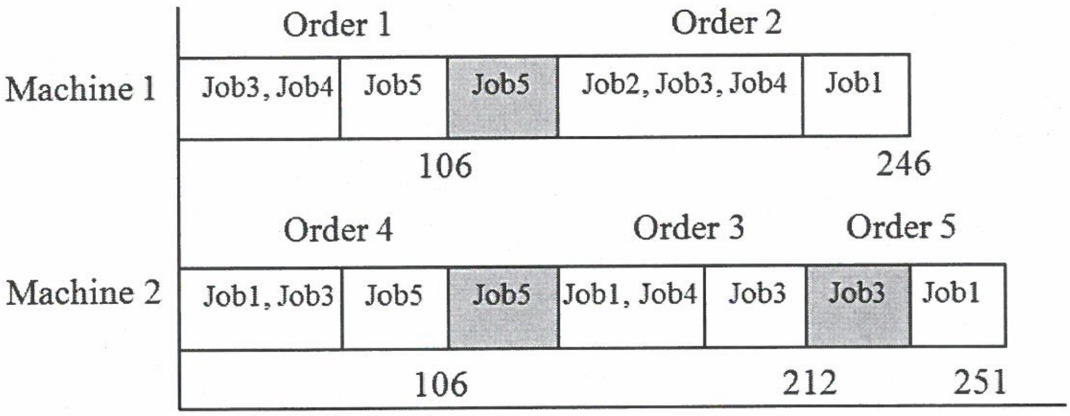

Figure 12 The makespan minimization sequence of the machines

As it seen in the Figure 7, for Machine 1 in Order 1 Job 5 is the last processed job and in Order 2 Job 5 is the first processed job which means that there is setup savings for the Order 2. Also for Machine 2 in Order 4 Job 5 is the last processed job and in Order 3 Job 5 is the first processed job which means that there is setup savings for the Order 3, in addition to this Order 3 and Order 5 have similar situation for setup savings.

Step 4: Determine the $C_{\max} = \max\{C_{\max}(m_1), C_{\max}(m_2)\}$

$C_{\max} = \max\{C_{\max}(m_1) = 246, C_{\max}(m_2) = 251\} = 251$

This is the result of the heuristic algorithm. When the heuristic algorithm result and the MILP model result are considered for this example, makespans are the same in both methods, and the customer orders allocation for both machines are also the same.

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, the computational experiments are explained in detail for measuring the effectiveness and the efficiency of the MILP model and the suggested constructive algorithm for the makespan minimization problem. MILP model is coded by GAMS 24.2.2 and solved by CPLEX solver, and the suggested heuristic algorithm is coded in C++ language. The computer which is Intel Core i7 16 GB RAM under Windows 8 operating system managed the whole computational experiments in this study.

5.1 Computational Setting for Test Problems

The parameter settings used in this study were generated by Akkocaoğlu (2014). The designed numeric equivalences of parameters that serve in experimental methods of this study stated below;

1. *Number of customer orders* (K): 5, 10, 15, 20
2. *Number of job types* (N): 5, 10, 15, 20
3. *Number of jobs within each customer order*: They are generated from four discrete uniform distributions

   Variable: DU[1, N]

   Constant: DU[2, N–1]

4. *Processing times*: Short: DU[1, 10]

   Long: DU[100, 200]

5. *Setup times*: Low mean-low variance: DU[25, 35]

Low mean-high variance: DU[10, 50]

High mean-low variance: DU[55, 65]

High mean-high variance: DU[40, 80]

Five problem instances are created for each potential combination of the parameters, therefore the whole number of problems is 1280 experimented. The makespan minimization problem approached by mathematical model, in particular, a software package called GAMS supported by CPLEX solver which enabled us to choose between nodes and also filter resources according to demands. In respect of these options CPLEX conduct with 2048 MB allocation memory, double parallel threads, taking best advantage of best estimate search as node choice policy and forceful branching. In addition to this restriction on the number of iterations and number of nodes to be searched are 10^9. The problem instances are analyzed depending on these circumstances and we restrict the runtime of the CPLEX for getting the optimal results for every problem instance to 10,800 sec.

5.2 Performance Measures

With the help of CPLEX there are two types of solution for MILP model which are the best integer solution wanted one and the best non-integer solution where some of variables are non-integer. The acquired best non-integer result is suited to the best integer solution if it is so then the optimal solution is obtained by MILP model, otherwise solution optimality cannot be sure. The makespan time which is obtained by heuristic algorithm and the makespan time of the optimum solution are compared in terms of the problem with optimal solutions. On the other hand the makespan time which is achieved by heuristic algorithm and the makespan time of the best integer solution are compared for the best integer solutions results of the problems.

MILP model achieves the optimum solution. In order to test the success of the heuristic algorithm, the percentages of deviations from optimum makespan times that derived from heuristic algorithm calculations are studied. Let PD^O is the percent deviation which is computed by

$$PD^O = \frac{M^H - M^O}{M^O} \times 100$$

where M^H = makespan of the solution obtained by the heuristic algorithm, and

M^O = makespan of the optimal solution gained by the MILP model.

On the other hand, the solutions that is not assured the optimal solution by the MILP model however best integer solution exists, the percent deviation of the makespan time should be calculated which is acquired by heuristic algorithm from the makespan time of the best integer result. Let's say the percent deviation is demonstrated as PD^B which is able to be calculated by

$$PD^B = \frac{M^H - M^B}{M^B} \times 100$$

M^B = makespan of the best integer solution gained by the MILP model.

When the result of the percent deviation of the best integer solutions obtained from heuristic algorithm is negative, the results are assumed as zero. The efficiency evaluation of the MILP model and the heuristic algorithm is the computed times which is supposed to solve the problem. The computed time of the heuristic algorithm was not evaluated due to the reason that it was quite smaller than the measured time of the MILP model. With increasing the number of customer order and job types the computational time of the solving the problem rises up however the computational time is again quite smaller.

5.3 Discussion of the Results

The solution approaches of the performances are argued in this part. The efficiency of the MILP model is discussed firstly and thereafter the efficiency of the heuristic algorithm is examined.

5.3.1 Performance of the MILP Model

In Table 9 the efficiency of the MILP model is demonstrated. According to this table whole problem instances with five customer order can be solved optimally. On the other hand, when the customer order is ten, seventy five percent of the instances cannot be solved optimally, and also 309 and 312 problem instances could not be solved for the set of problems which have fifteen and twenty orders separately.

Table 9 Performance of mathematical model

NUMBER OF CUSTOMER ORDERS	TOTAL NUMBER OF PROBLEM INSTANCES CONSIDERED	MILP		
		NUMBER OF TIMES OPTIMUM INTEGER SOLUTION IS OBTAINED	NUMBER OF TIMES BEST INTEGER SOLUTION IS OBTAINED	NUMBER OF UNSOLVED PROBLEM INSTANCES
5	320	320	0	0
10	320	79	241	0
15	320	11	309	0
20	320	8	312	0

For various combinations of the processing and setup times the particular analysis on the number of optimal results and the best integer results which are acquired by MILP model are shown in Table 9 in detail.

In order to analyze the performance of the MILP model the quality of non-optimal results should be examined. It is known that MILP model results with a gap which is in among with the solution found and the best possible. On account of this, for the purpose of demonstrating the percent difference of integer solutions from the theoretic optimum the gap rates are controlled to demonstrate. Especially for some problem instances, plenty of iterations are realized and integer solutions got closer range to the theoretical optimum for every iteration. Yet GAMS is restricted due to the time limitation before gaining the optimum result. However, this president is the best case due to the reason that completing 3 hour time limitation after that the gap rates get very close to zero. On the other side in the worst case analysis, solution of the problem instances operation (branching) is restricted because of the memory errors which show up after some iteration are ended. For the worst case analysis the problem instances whose result branching is restricted because of memory faults which happen after some iterations are finished. Here the highest gap result that denotes the worst case is obtained as 8. 28%.

Table 10 Number of problems with the optimum results obtained from mathematical model

NUMBER OF CUSTOMER	NUMBER OF JOBS	TOTAL NUMBER OF PROBLEM INSTANCE CONSIDERED	RANDOM															
			LOW								HIGH							
			LL		LH		HL		HH		LL		LH		HL		HH	
			NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL
5	5	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	10	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	15	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	20	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		320	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
10	5	80	4		1		2	3	1	1	2	3	3	4	3	3	3	5
	10	80		1	4	1			1	1	3	2	5	1	2	2		2
	15	80								1			3	1	2	1		
	20	80									1		3	1		1	2	
		320	4	1	5	1	2	3	2	3	6	5	14	7	7	7	5	7
15	5	80									2				4			3
	10	80																
	15	80																2
	20	80																
		320									2				4			5
20	5	80											1	1	1	1	1	1
	10	80										1						
	15	80																
	20	80															1	
		320										1	1	1	1	1	2	1

5.3.2 Performance of the Heuristic Algorithms

Here, the influence of problem variables changes on the heuristic algorithm performance is negotiated. In order to comprehend the comparison tables in the subsequent tables the abbreviations which are demonstrated in Appendix A. The number of optimally solved problems by mathematical model and constructive heuristic algorithm is demonstrated in the following Table 11.

Table 11 The number of problems with the optimum results obtained from the mathematical model and heuristic algorithm

K	NUMBER OF PROBLEMS	MILP		ALGORITHM
		OPTIMAL	BEST INTEGER	OPTIMAL
5	320	320	0	115
10	320	79	241	29
15	320	11	309	5
20	320	8	312	3

As it is seen in Table 10 the whole problem instances when the customer order is five can be solved optimally. Yet when the number of customer order is increasing, it is clearly mentioned in the table the number of optimally solved problems is decreasing from 79 to 8 for the problem sets which have ten, fifteen, and twenty customer orders separately.

Some of problem instances are solved optimally by mathematical model, the rest of them have the best integer solutions. Therefore, to make a comparison these two different situations should be considered separately. To do so, in the following tables are composed. Table 11 and Table 12 demonstrate the number of optimum and the best integer results obtained from the mathematical model for VARIABLE and CONSTANT Case are demonstrated in detail.

Table 12 The number of optimum and the best integer results obtained from the mathematical model for VARIABLE Case

NUMBER OF CUSTOMER	NUMBER OF JOBS	VARIABLE															
		SHORT								LONG							
		LL		LH		HL		HH		LL		LH		HL		HH	
		optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best
5	5	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	10	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	15	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	20	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
10	5	4	1	0	5	1	4	0	5	2	3	3	2	1	4	1	4
	10	0	5	0	5	0	5	0	5	0	5	0	5	1	4	1	4
	15	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	20	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
15	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	10	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	15	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	20	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
20	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	10	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	15	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	20	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5

Table 13 The number of optimum and best integer results obtained from mathematical model for constant case

NUMBER OF CUSTOMER	NUMBER OF JOBS	CONSTANT															
		SHORT								LONG							
		LL		LH		HL		HH		LL		LH		HL		HH	
		optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best	optimum	best
5	5	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	10	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	15	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
	20	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0
10	5	2	3	3	2	3	2	4	1	3	2	4	1	3	2	5	0
	10	3	2	2	3	5	0	1	4	2	3	2	3	0	5	2	3
	15	0	5	0	5	3	2	1	4	2	3	0	5	0	5	1	4
	20	1	4	0	5	1	4	1	4	1	4	1	4	2	3	0	5
15	5	2	3	0	5	0	5	0	5	4	1	0	5	0	5	3	2
	10	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	15	0	5	0	5	0	5	0	5	0	5	0	5	0	5	2	3
	20	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
20	5	0	5	0	5	1	4	1	4	1	4	1	4	1	4	1	4
	10	0	5	1	4	0	5	0	5	0	5	0	5	0	5	0	5
	15	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
	20	0	5	0	5	0	5	0	5	0	5	0	5	1	4	0	5

Five problem instances are produced and the averages of percent deviations of these five instances are gotten for every combination of test parameters. Table 13 demonstrates the percent deviations of heuristic algorithm from the optimal results for 640 instances in which the number of jobs of every customer order is variable. For example, the average percent deviation of these five problem instances is computed as 1.037, this value figures that there are five customer orders, there are five jobs in every order, the processing times of the jobs are short, mean and variance of job setup time is low, and the number of job through every customer order is variable.

Table 14 Average percent deviations of the heuristic algorithm from optimal results for VARIABLE case

Number of Customer Orders	Number of Jobs	VARIABLE										
		SHORT					LONG					Total Average
		LL	LH	HL	HH	Average	LL	LH	HL	HH	Average	
5	5	1.04	10.09	1.49	1.04	3.42	2.80	0.85	3.02	2.22	2.22	2.82
	10	3.63	0.00	1.84	1.36	1.71	0.98	2.11	1.81	0.19	1.27	1.49
	15	1.82	1.84	3.76	0.92	2.08	1.88	1.69	0.00	1.48	1.26	1.67
	20	1.75	2.54	1.13	0.27	1.42	1.78	1.47	0.37	3.21	1.71	1.57
AVG_5		2.06	3.62	2.06	0.90	2.16	1.86	1.53	1.30	1.78	1.62	1.89
10	5	4.98		0.45		2.72	3.01	1.43	1.71	3.36	2.38	2.55
	10								3.08	2.10	2.59	2.59
	15											
	20											
AVG_10		4.98		0.45		2.72	3.01	1.43	2.39	2.73	2.48	2.60
AVG_TOTAL		3.52	3.62	1.25	0.90	2.44	2.43	1.48	1.85	2.25	2.05	2.24

The empty blank cells in Table 13 are not able to be calculated for involved problem sets because of the lack of memory for not existing MILP results for these problem sets where number of jobs in every order is variable. In the previous Table 12 the detail of the number of optimal results are given in detail.

Table 13 demonstrates that the average percent deviations of the results which are achieved by heuristic algorithm, and when the processing times are short which is demonstrated as SHORT and or when the processing times are long which is demonstrated as LONG.

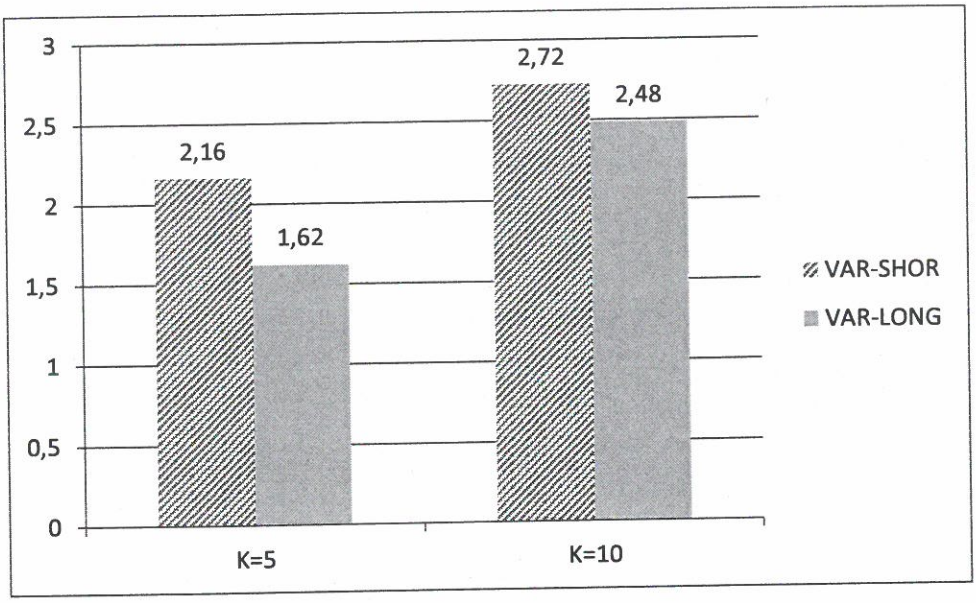

Figure 13 Dissimilarity between the average percent deviations for the customer orders

For every number of customer orders, Figure 13 shows the dissimilarity between the averages of LONG and SHORT. According to these figure percent deviations of LONG for five and ten number of orders is less than the percent the percent deviations of SHORT case. This circumstance is hoped due to the reason that in LONG case problem instances the processing time are slightly bigger than setup times. For this reason the impact of setup reduction is minor important than SHORT case by reason of the processing times are small relative to setup times. In such cases where customer orders have constant number of job assignments, the logic stated above, is more convenient to apply.

As can be seen similar logic from the previous Table 13, it is established for the problem instances where the count of jobs customer orders is obtained as constant. It is clear that in Table 14, too much of jobs of problem instances are not able to be calculated when the customer orders are fifteen and twenty.

Table 15 Average percent of the heuristic algorithm from optimal results for CONSTANT case

NUMBER OF CUSTOMER	NUMBER OF JOB	CONSTANT											General Total Average
		SHORT					LONG					Total Average	
		LL	LH	HL	HH	Average	LL	LH	HL	HH	Average		
5	5	8,26	2,18	9,62	2,88	5,73	1,31	1,71	1,10	1,20	1,33	3,53	4,27
	10	1,06	4,37	4,43	1,72	2,89	0,73	1,50	1,01	2,00	1,31	2,10	6,05
	15	0,66	5,25	0,69	1,94	2,14	2,65	2,66	3,43	2,93	2,92	2,53	8,76
	20	0,76	1,36	0,28	0,56	0,74	1,49	0,63	1,79	1,79	1,42	1,08	10,54
AVG_5		2,69	3,29	3,75	1,77	2,88	1,54	1,62	1,83	1,98	1,75	2,31	2,31
10	5	0,12	2,28	4,88	0,75	2,01	0,08	0,94	0,00	0,98	0,50	1,26	3,13
	10	0,64	17,65	3,22	0,00	5,38	2,48	0,37	0,03	6,33	2,30	3,84	6,92
	15			1,97	4,59	3,28	0,04			0,02	0,03	1,66	8,33
	20	0,00		15,13	0,00	5,04	0,01	0,00	0,00		0,00	2,52	11,26
AVG_10		0,25	9,97	6,30	1,34	3,93	0,65	0,44	0,01	2,44	0,71	2,32	2,32
15	5	0,00					0,18			0,08	0,13	0,13	2,56
	10									0,00	0,00	0,00	7,50
	15												
	20												
AVG_15		0,00					0,18			0,04	0,06	0,06	0,06
20	5			0,00	1,09	0,55	0,00	0,04	0,02	0,10	0,04	0,29	2,65
	10		0,00			0,00						0,00	5,00
	15												
	20								0,00		0,00	0,00	10,00
AVG_20			0,00	0,00	1,09	0,36	0,00	0,04	0,01	0,10	0,02	0,19	0,19
AVG_TOTAL		0,98	4,42	3,35	1,40	2,39	0,59	0,70	0,62	1,14	0,63	1,22	1,22

Similar demonstration as previous table, Table 14 demonstrates the percent deviations of heuristic algorithm results which are obtained from optimal solutions for constant case by MILP model. As it seen in the table when the number of customer orders is fifteen and twenty, and also in every customer orders the number of jobs is constant, the empty cells shows that they could not be solved optimally as mentioned before. Thus, for these problem instances which fall into fifteen and twenty customer orders, the analysis of heuristic algorithm's performance which depends on the percent deviation from the optimal results could not be adequate to examine the results and guides the wrong solutions. Therefore, the best integer results which is acquired by MILP model ought to be compared by the time optimal result does not appear. Table 15 indicates the percent deviations of the heuristic results from the best integer results which are acquired by MILP model.

Table 16 Average percent deviation of the heuristic algorithm from the best integer results for CONSTANT case

NUMBER OF CUSTOMER	NUMBER OF JOBS	CONSTANT											General Total Average
		SHORT					LONG					Total Average	
		LL	LH	HL	HH	Average	LL	LH	HL	HH	Average		
10	5	0,17	1,32	0,00	0,00	0,37	0,04	0,02	0,00		0,02	0,20	2,60
	10	0,74	0,05		0,68	0,49	0,44	0,01	0,15	0,18	0,19	0,34	5,17
	15	2,42	0,43	0,21	0,08	0,79	0,31	0,60	2,37	0,84	1,03	0,91	7,95
	20	1,46	0,24	0,02	0,10	0,45	0,27	0,25	0,04	0,08	0,16	0,31	10,15
AVG_10		1,20	0,51	0,08	0,21	0,53	0,26	0,22	0,64	0,37	0,35	0,44	0,44
15	5	2,58	7,03	3,55	6,43	4,90	4,44	3,50	1,56	2,23	2,93	3,91	4,46
	10	1,71	1,20	0,84	10,15	3,48	2,68	3,68	3,25	2,52	3,03	3,25	6,63
	15	0,74	2,49	0,74	0,81	1,20	2,07	2,48	1,25	1,91	1,93	1,56	8,28
	20	0,43	0,88	0,30	0,24	0,46	1,61	1,38	1,12	2,05	1,54	1,00	10,50
AVG_15		1,37	2,90	1,36	4,41	2,51	2,70	2,76	1,79	2,18	2,36	2,43	2,43
20	5	1,56	5,39	0,32	1,63	2,22	0,08	1,69	0,11	0,45	0,58	1,40	3,20
	10	0,01	0,01	0,01	3,29	0,83	0,00	0,00	0,00	0,00	0,00	0,42	5,21
	15	1,14	2,08	2,97	0,01	1,55	0,00	0,00	0,00	0,00	0,00	0,77	7,89
	20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	10,00
AVG_20		0,68	1,87	0,82	1,23	1,15	0,02	0,42	0,03	0,11	0,15	0,65	0,65
AVG_TOTAL		1,08	1,76	0,75	1,95	1,39	1,00	1,13	0,82	0,89	0,95	1,17	1,17

The achieved percent deviations for the case where the processing times of the jobs are short are considerably smaller than results achieved for the case where the processing times of the jobs are long, because for the setup reduction the total completion time plays critical role when the processing times are slightly less than the setup times. The obtained results by heuristic algorithm are compared with the best integer results which are achieved from MILP model, and demonstrated in Appendix A. The detail of the best integer solutions according to Table 15 in Appendix B for customer orders ten and fifteen are demonstrated in figures.

Table 17 Percent deviations of the heuristic when K=20

Number of Customer Orders	Number of Jobs	CON-SHORT	CON-LONG
20	5	2.22	1.40
	10	0.83	0.42
	15	1.55	0.77
	20	0.00	0.00

Table 16 shows the percent deviations of the heuristic algorithm for the CONSTANT case when the customer order is twenty. The attitude of heuristic algorithm is demonstrated for five customers which is relied on the number of jobs inside of every processing times and order.

Table 18 General total percent deviations of the heuristic algorithm from the optimal solutions for CONSTANT and VARIABLE case

NUMBER OF CUSTOMERS	NUMBER OF JOBS	General Total Average
5	5	3.18
	10	1.80
	15	2.10
	20	1.32
AVG_5		2.10
10	5	1.90
	10	3.21
	15	1.66
	20	2.52
AVG_10		2.46
15	5	0.13
	10	
	15	0.00
	20	
AVG_15		0.06
20	5	0.29
	10	0.00
	15	
	20	0.00
AVG_20		0.19
AVG_TOTAL		1.73

According to the average percent deviations of the heuristic algorithm from the optimal solutions for both CONSTANT and VARIABLE case the general average percent deviations are given in Table 17. In addition to this, also the average percent deviations from the best integer solutions for both CONSTANT and VARIABLE case the general average percent deviations are given in Table 18. In Table 17, the general percent deviation is 1.73 for all over problem instances, and in Table 18 the general percent deviation is 1.03 for all over problem instances.

Table 19 General total percent deviations of the heuristic algorithm from the best integer solutions for CONSTANT and VARIABLE case

NUMBER OF CUSTOMERS	NUMBER OF JOBS	General Total Average
10	5	0.70
	10	1.58
	15	2.39
	20	2.73
AVG_10		0.33
15	5	4.19
	10	3.78
	15	3.54
	20	3.75
AVG_15		2.55
20	5	1.09
	10	1.35
	15	2.07
	20	2.50
AVG_20		0.21
AVG_TOTAL		1.03

CHAPTER 6

CONCLUSIONS AND PROPOSALS FOR FUTURE RESEARCH

In this thesis, COS problem on two identical parallel machines with setup times is investigated. The assumption is that; jobs which are in the same customer order must be processed one after another, and the whole customer orders are delivered together. A setup is required before processing procedure of the first job lot of a customer however if the beginning job is not the same as the last job lot of the instantly preceding customer order, there is no need for the setup. The aim of this study is to minimize the makespan of the customer orders.

In our study we developed a mathematical programming model and a heuristic algorithm to provide an optimal and near optimal results respectively. It is observed that the suggested heuristic algorithm is developed for the makespan minimization problem in order to get satisfying solutions for solving the small, medium, and large sized problem instances and getting an optimal or near optimal results. The results shows that a standard MILP solver is not adequate to solve the especially medium, and large sized problems therefore developing the heuristic algorithm is a wise solution.

Customer order scheduling problems on two identical machines are not greatly studied before. For this reason there is a huge amount of subject residual for the future study. Various extensions can be considered for our study. The first one can be changing the problem characteristic. Various problem characteristics can be considered like ready times, precedence relations between the jobs, performance measures can be considered like maximum tardiness, the number of tardy orders, and so on. Secondly, metaheuristics could be helpful to develop the quality of the results which is achieved in our study. Thirdly, more complex machining environments like multiple stages or multiple parallel machines can be another future study topic.

REFERENCES

1. **Akkocaoğlu, H. (2014).** A new customer order scheduling problem on a single-machine with job setup times, Unpublished MS. Thesis, Çankaya University, Ankara.
2. **Allahverdi, A., & Soroush, H. (2008).** The significance of reducing setup times/setup costs. *European Journal Of Operational Research 187*, Pp. 978-984.
3. **Allahverdi, A., Gupta, J. N., & Aldowaisan, T. (1999).** A review of scheduling research involving setup considerations. *Omega, International Journal Of Management Science, 27* 219-239.
4. **Allahverdi, A., Ng, C., Cheng, T., & Kovalyov, M. Y. (2008).** A survey of scheduling problems with setup times or costs. *European Journal Of Operational Research 187*, 985-1032
5. **Brucker, P., Kovalyov, M. Y., Shafransky, Y. M., & Werner, F. (1998).** Batch scheduling with deadlines on parallel machines. *Annals of Operations Research, 83*, 23-40.
6. **Erel, E., & Ghosh, J. B. (2007).** Customer order scheduling on a single machine with family setup times: Complexity and algorithms. *Applied Mathematics and Computation, 185*(1), 11-18.
7. **Gupta, J. N., Ho, J. C., & Van Der Veen, J. A. (1997).** Single machine hierarchical scheduling with customer orders and multiple job classes. *Annals of Operations Research, 70*, 127-143.
8. **Janiak, A., Kovalyov, M. Y., & Portmann, M.C. (2005).** Single machine group scheduling with resource dependent setup and processing times. *European Journal of Operational Research, 162*, 112-121.
9. **Jia, J., & Mason, S. J. (2009).** Semiconductor manufacturing scheduling of jobs containing multiple orders on identical parallel machines. *International Journal of Production Research, 47*(10), 2565-2585.

10. **Leung, J. Y. T., Li, H., & Pinedo, M. (2005).** Order scheduling in an environment with dedicated resources in parallel. *Journal of Scheduling, 8*(5), 355-386.
11. **Leung, J. Y. T., Li, H., & Pinedo, M. (2008).** Scheduling orders on either dedicated or flexible machines in parallel to minimize total weighted completion time. *Annals of Operations Research, 159*(1), 107-123.
12. **Mingbao, C., & Shijie, S. (2005).** Two scheduling problems in group technology with deteriorating jobs. *Applied Mathematics-A Journal of Chinese Universities, 20*(2), 225-234.
13. **Shen, L., Mönch, L., & Buscher, U. (2014).** A simultaneous and iterative approach for parallel machine scheduling with sequence-dependent family setups. *Journal of Scheduling, 17*(5), 471-487.
14. **Su, L. H., Chen, P. S., & Chen, S. Y. (2013).** Scheduling on parallel machines to minimise maximum lateness for the customer order problem. International *Journal of Systems Science*, 44(5), 926-936.
15. **Yang, J. (2005).** The complexity of customer order scheduling problems on parallel machines. *Computers & Operations Research, 32*(7), 1921-1939.
16. **Yang, J., & Posner, M. E. (2005).** Scheduling parallel machines for the customer order problem. *Journal Of Scheduling*, 8(1), 49-74.

APPENDICES

APPENDIX A – COMPARISON TABLES

TABLE A. 1 - AVERAGE PERCENT DEVIATIONS OF ALGORITHM 4 FROM THE BEST-INTEGER SOLUTIONS

NUMBER OF CUSTOMER	NUMBER OF JOBS	TOTAL NUMBER OF PROBLEM INSTANCE CONSIDERED	RANDOM															
			LOW								HIGH							
			LL		LH		HL		HH		LL		LH		HL		HH	
			NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL	NUMBER OF SOLVED	NUMBER OF OPTIMAL
5	5	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	10	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	15	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	20	80	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		320	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
10	5	80	4		1		2	3	1	1	2	3	3	4	3	3	3	5
	10	80		1	4	1			1	1	3	2	5	1	2	2		2
	15	80								1			3	1	2	1		
	20	80									1		3	1		1	2	
		320	4	1	5	1	2	3	2	3	6	5	14	7	7	7	5	7
15	5	80									2				4			3
	10	80																
	15	80																2
	20	80																
		320									2				4			5
20	5	80											1	1	1	1	1	1
	10	80										1						
	15	80																
	20	80															1	
		320										1	1	1	1	1	2	1

APPENDIX B – AVERAGE PERCENT DEVIATIONS FROM THE BEST INTEGER SOLUTIONS FOR BOTH VARIABLE AND CONSTANT CASE

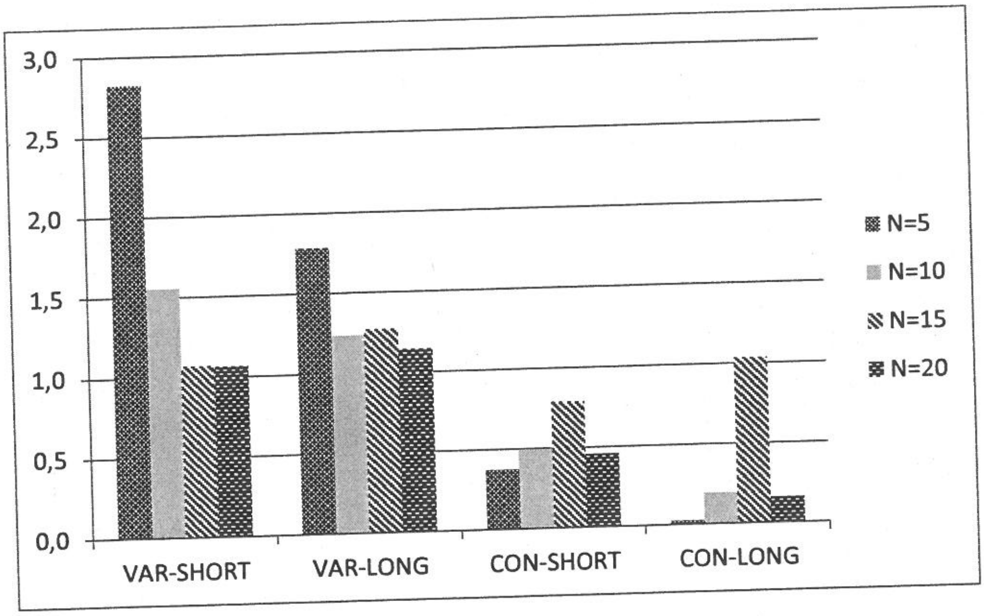

Figure 14 Average percent deviations when *K*=10

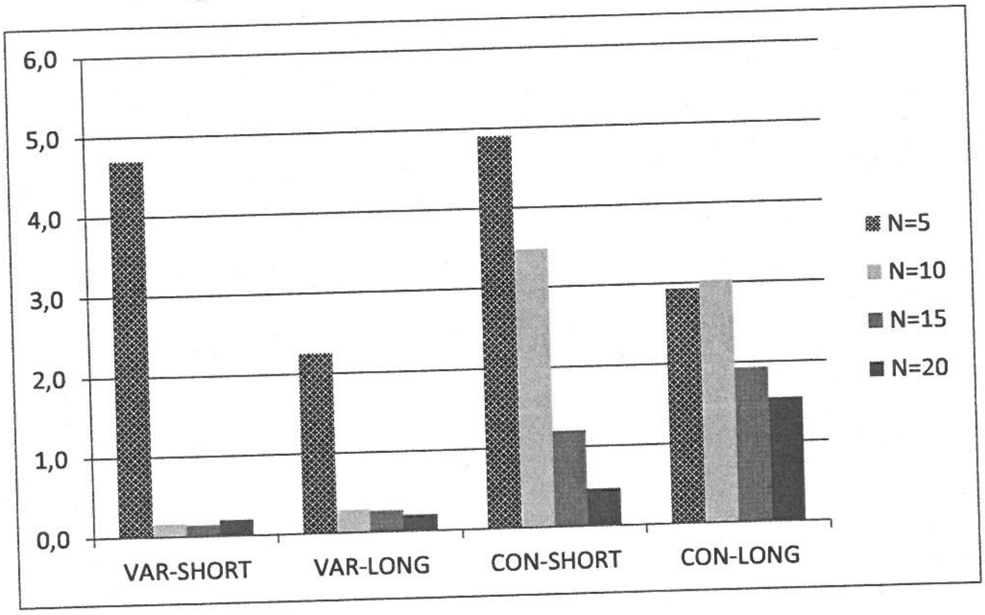

Figure 15 Average percent deviations when *K*=15

APPENDIX C

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Çağlayan, Nihan
Nationality: Turkish (TC)
Date and Place of Birth: 19 May 1987, Kırşehir
Marital Status: Single
Phone: +90 386 280 47 18
email: nihancaglayan@ahievran.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Çankaya University/ Industrial Engineering	2017
BS	Çankaya University/ Industrial Engineering	2011
High School	Mehmet Akif Ersoy High School	2006

WORK EXPERIENCE

Year	Place	Enrollment
2014-present	Ahi Evran University	Intructor
2011-2012	GENPOWER Generator	Production Planning Engineer
2009 July	Havelsan	Intern Engineering Student
2009 July	Aselsan	Intern Engineering Student
2008 July	Hidromek	Intern Engineering Student

FOREIGN LANGUAGES

Advanced English