



**MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES**



A NEW MULTI-OBJECTIVE MODEL FOR INTEGRATED SUPPLY CHAIN SCHEDULING PROBLEM IN FLEXIBLE JOB SHOP ENVIRONMENT

ZEYNEP CEYLAN

Ph.D. THESIS

Department of Industrial Engineering

Thesis Supervisor

Assoc. Prof. Dr. Serol BULKAN

Thesis CO- Supervisor

Prof. Dr. Hakan TOZAN

ISTANBUL, 2018



**MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES**



A NEW MULTI-OBJECTIVE MODEL FOR INTEGRATED SUPPLY CHAIN SCHEDULING PROBLEM IN FLEXIBLE JOB SHOP ENVIRONMENT

ZEYNEP CEYLAN

(724414005)

Ph.D. THESIS

Department of Industrial Engineering

Thesis Supervisor

Assoc. Prof. Dr. Serol BULKAN

Thesis CO- Supervisor

Prof. Dr. Hakan TOZAN

ISTANBUL, 2018

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

Zeynep CEYLAN, a Doctor of Philosophy student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended her thesis entitled “**A New Multi-Objective Model for Integrated Supply Chain Scheduling Problem in Flexible Job Shop Environment**”, on September 11, 2018 and has been found to be satisfactory by the jury members.

Jury Members

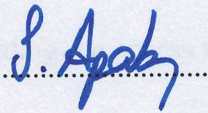
Assoc. Prof. Dr. Serol BULKAN

Marmara University 

Assoc. Prof. Dr. Gülfem TUZKAYA

Marmara University 

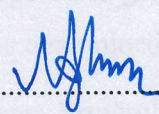
Assoc. Prof. Dr. Sinan APAK

Maltepe University 

Assist. Prof. Dr. Adnan ÇORUM

Bahçeşehir University 

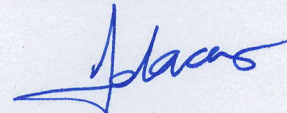
Assist. Prof. Dr. M. Batuhan AYHAN

Marmara University 

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Zeynep CEYLAN be granted the degree of Doctor of Philosophy in Department of Industrial Engineering, Industrial Engineering Program on 2018/24-06, 2018. (Resolution no: 2018/24-06,...).

Director of the Institute
Prof. Dr. Bülent EKİCİ



ACKNOWLEDGMENTS

At first, I must express my gratitude towards my advisor, Assoc. Prof. Dr. Serol BULKAN who guided, supported and helped me whenever I need. His knowledge consistently motivated me throughout my research work. I sincerely appreciate his valuable comments and time spent to assist and answer any of my research questions. It was though his patience, understanding and kindness that I completed my thesis.

I would also like to thank to my co-advisor Prof. Dr. Hakan TOZAN for his help, guidance and constructive suggestions and comments during the thesis study.

I am also grateful to thesis committee members Assoc. Prof. Dr. Gülfem TUZKAYA and Assist. Prof. Dr. Adnan ÇORUM, for their valuable contributions and their suggestions, guiding ideas and constructive approach to me.

Also, I would like to thank the Ondokuz Mayıs University Industrial Engineering Department staff for their support and friendship. I wish to thank to Assist. Prof. Ahad FOROUGHI specifically whom I did a lot of brain-storming and shared our ideas.

I would like to express my gratitude to GAMS Corporation for providing me access to GAMS optimization software during this study.

I wish to express my love and gratitude to my beloved family; for their support, understanding and endless love, through my life and my studies.

I want to thank my husband, Selim CEYLAN, for his invaluable support and guidance during my whole life. His advices always made me strong and kept me determined.

This thesis is dedicated to my treasure and meaning of my life, my son, Ertuğrul Kaan CEYLAN who showed me the purest, unrequited and great love in world. His big smile and laughs always gave me power when I was tired. I would not have finished this thesis without his priceless hugs.

SEPTEMBER, 2018

ZEYNEP CEYLAN

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENT	i
TABLE OF CONTENTS	ii
ÖZET	vi
ABSTRACT	vii
CLAIM FOR ORIGINALITY	viii
SYMBOLS	ix
ABBREVIATIONS	xi
LIST OF FIGURES	xiv
LIST OF TABLES	xvi
CHAPTER 1: INTRODUCTION	1
1.1. Supply Chain Scheduling	1
1.2. Importance of the Problem	4
1.3. Outline of the Thesis	5
CHAPTER 2: INTEGRATED SUPPLY CHAIN SCHEDULING MODELS AND LITERATURE REVIEW	6
2.1. General Description of Supply Chain Scheduling Problems	7
2.2. Classification of Supply Chain Scheduling Models	7
2.2.1. Order size	8
2.2.2. Number of customers	8
2.2.3. Vehicle characteristics	9
2.2.4. Objective function	10

2.2.4.1. <i>Time-based objective functions</i>	10
2.2.4.2. <i>Cost-based objective functions</i>	10
2.2.4.3. <i>Revenue-based objective functions</i>	11
2.2.5. Machine configuration	11
2.2.6. Product lifespan	17
2.2.7. Modeling methods and solution approaches	18
2.3. Literature Survey	19
CHAPTER 3: PROPOSED MATHEMATICAL MODEL	30
3.1. Model Assumptions	32
3.2. Indices, Parameters and Variables	32
3.2.1. Indices and sets	32
3.2.2. Parameters	33
3.2.3. Decision variables	33
3.3. Model Formulation	34
3.3.1. Objective Functions	34
3.3.1.1. <i>Total cost</i>	34
3.3.1.2. <i>Total C_{max}</i>	35
3.3.2. Constraints	36
3.4. Linearization of Proposed Model	40
3.4.1. Linearization the product of a binary and a continuous variable	41
3.4.2. Linearization the product of two binary variables	41
CHAPTER 4: MATERIAL AND METHODS	47
4.1. Multi-Objective Optimization	48
4.1.1. LP metric/global criterion method	50

4.1.2. Goal attainment	51
4.1.3. Weighted sum method	52
4.1.4. Lexicographic goal programming	54
4.1.5. Min-max method	56
4.1.6. Non-dominated sorting genetic algorithm-II (NSGA-II)	56
CHAPTER 5: TESTING AND COMPUTATIONAL EXPERIMENTS	61
5.1. Dataset	62
5.2. Small and Medium Sized Test Problems	62
5.2.1. Test Problem 1	63
5.2.1.1. <i>Individual optimization results for test problem 1</i>	63
5.2.1.2. <i>Normalization of objective functions</i>	64
5.2.1.3. <i>Weighted sum method results for test problem 1</i>	65
5.2.1.4. <i>Filtering/displaced ideal solution method</i>	66
5.2.1.5. <i>Goal attainment method results for test problem 1</i>	67
5.2.1.6. <i>NSGA-II results for test problem 1</i>	68
5.2.1.7. <i>Comparison of solution methods</i>	73
5.2.2. Test problem 2	79
5.2.3. Test problem 3	82
5.2.4. Test problem 4	85
5.2.5. Test problem 5	88
5.2.6. Test problem 6	91
5.2.7. Test problem 7	94
5.3. Large Sized Test Problems	99

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH DIRECTIONS	107
6.1. Conclusions	107
6.2. Future Research Directions	109
REFERENCES	110
APPENDIX	120
CURRICULUM VITAE	122



ÖZET

ESNEK İŞ ATÖLYESİ ORTAMINDA ENTEGRE TEDARİK ZİNCİRİ ÇİZELGELEME PROBLEMİ İÇİN YENİ ÇOK AMAÇLI BİR MODEL

Bu tezde, Tedarik Zinciri Çizelgeleme (TZÇ) problemi için yeni entegre çok amaçlı bir matematiksel model geliştirilmiştir. Üretim tesisi, depo ve nihai müşteriler içeren Tedarik Zinciri Ağı Tasarımı (TZAT) için çok ürünlü, çok periyotlu ve çok kademeli bir yapı geliştirilmiştir. Üretim tesisinin ürünleri esnek atölye tipi çizelgeleme ortamında üreteceği varsayılmıştır. Ürünler, bir dizi operasyondan oluşmakta ve nihai ürünü tamamlamak için üretim tesisinde bulunan çok fonksiyonlu makineler setinde işlenmektedir. Nihai ürünler, müşterilerin taleplerini karşılamak üzere depolara teslim edilmektedir. Ayrıca, müşteri taleplerinin yerine getirilmemesi durumunda, her dönem için bekleyen sipariş ortaya çıkmaktadır.

Problem kısıtlı çok amaçlı Karma Tamsayılı Doğrusal Programlama (KTDP) modeli olarak formüle edilmiştir. İlk amaç fonksiyonu, tüm periyotlardaki toplam tedarik zinciri maliyetlerini en aza indirmeyi ve ikinci amaç fonksiyonu ise tüm periyotlardaki yayılma alanlarının toplamını en aza indirmeyi amaçlamaktadır. Önerilen KTDP modelinin performansını değerlendirmek için çeşitli boyutlarda rastgele test problemleri oluşturulmuştur. Farklı Çok Amaçlı Karar Verme Yöntemleri (ÇAKY) ve Pareto yöntemine dayanan bir meta-sezgisel yaklaşım olan "*baskılanamayan sıralamalı genetik algoritma-II* (NSGA-II)" yöntemi modeli çözmek için kullanılmıştır. Ayrıca, en iyi çözüm yöntemini belirlemek için Filtreleme/Yer Değiştirme İdeal Çözüm yöntemi uygulanmıştır. Son olarak, önerilen yöntemin uygulanabilirliğini göstermek için bazı sayısal örnekler sunulmuştur.

EYLÜL, 2018

ZEYNEP CEYLAN

ABSTRACT

A NEW MULTI-OBJECTIVE MODEL FOR INTEGRATED SUPPLY CHAIN SCHEDULING PROBLEM IN FLEXIBLE JOB SHOP ENVIRONMENT

In this thesis, a new integrated multi-objective mathematical model for Supply Chain Scheduling (SCS) problem was developed. Multi-product, multi-period and multi-echelon structure was developed for Supply Chain Network Design (SCND) including manufacturing facility, warehouses, and final customers. It was assumed the manufacturing facility produces the products in flexible job-shop scheduling (F-JSS) environment. Each product consists of a sequence of operation and processed on a set of multi-functional machines in manufacturing facility to complete final product. The finished products are delivered to the warehouses in order to meet the demands of the customers. Moreover, in case customers' demands are not fulfilled, the shortage in terms of backorder can occur in each period.

The problem is formulated as a constrained multi-objective Mixed Integer Linear Programming (MILP) model. The first objective function aims to minimize the total supply chain costs and the second objective function aims to minimize the total makespan in all periods. Various sizes of test problems are randomly generated to evaluate the performance of the proposed MILP model. Different Multi-Objective Decision Making Methods (MODM) and the *"Non-dominated Sorting Genetic Algorithm-II (NSGA-II)"* method, which is a meta-heuristic approach based on Pareto method are used for solving the model. Furthermore, Filtering/Displaced Ideal Solution (DIS) method is applied to determine the best solution method. Finally, some numerical examples are presented to illustrate the applicability of the proposed method.

SEPTEMBER, 2018

ZEYNEP CEYLAN

CLAIM FOR ORIGINALITY

Supply chain and scheduling are the main parts of the production planning system which have a considerable influence on the productivity of the production systems. A supply chain comprises all interactions between suppliers, manufacturers, distributors, and customers. Scheduling is the process of arrangement, control and optimization of work and workloads in a production or manufacturing process. Thus, it is very important to determine that supply chain and scheduling are not independent and they should be integrated and identified simultaneously.

The main contribution and novelty of this thesis is presenting an efficient and useful mathematical model for integrated supply chain scheduling problem to the literature. A new multi-product, multi-period, multi-level and multi-objective model has been proposed and tested. Moreover, its results have been compared with different multi-objective optimization solution techniques.

A flexible job shop scheduling (F-JSS) model was integrated in manufacturing part of supply chain structure to minimize the makespan in all periods. However, in most of multi-level supply chain problems studied in the literature, the completion times of the manufacturing process are generated with some random numbers and included in the model as a parameter. In this thesis, F-JSS was integrated in the center of the entire Supply Chain Network (SCN) and thus obtained through multi-objective model.

In the literature, there are various methods ranging from classical methods to evolutionary algorithms to solve multi-objective optimization problems. Each method meets different expectations of decision makers in different features. However, the comparison and application of different methods in previous supply chain scheduling studies are not considered in the literature. In this thesis, five classical methods and an evolutionary algorithm were successfully applied to solve supply chain scheduling problem and results were compared.

September, 2018

Assoc. Prof. Dr. Serol BULKAN

Zeynep CEYLAN

SYMBOLS

C_{max}	: Makespan
D_j	: Delivery time
D_{max}	: Maximum delivery time
F_{max}	: Maximum flow time
L_{max}	: Maximum lateness
$M_{i,j}$	: Machine set for operation j of job i
$O_{i,j}$	: Operation j of job i
P_0	: Initial population
T_j	: Tardiness
$f_i(x)$	: i_{th} objective function
f_i^{max}	: Maximum value of the i_{th} objective function
f_i^{min}	: Minimum value of the i_{th} objective function
$g_i(x)$	: i_{th} inequality constraint function
$h_i(x)$	: i_{th} equality constraint function
k_q	: Normalization Constant
n_i	: Number of operations of job i
n_{pop}	: Number of population
p_c	: Crossover fraction
p_i	: Processing time of job i
p_m	: Mutation fraction
w_i	: Weight value for the i_{th} objective function
w_i^*	: Normalizing weight factor for the i_{th} objective function
x_i	: i_{th} decision variable
z^{ref}	: Goal vector
c	: Customer
h	: Operation
L	: Priority level
m	: Machine
N	: Dimension

p	: Product
P	: Crossing point
t	: Time period
U	: Number of tardy jobs
v	: Vehicle
w	: Warehouse
X	: Decision variable vector
X^*	: Design vector
α	: Deviation
Γ	: Feasible region



ABBREVIATIONS

3PLs	:	Third Party Logistics
ABC	:	Artificial Bee Colony
ACO	:	Ant Colony Optimization
ASA	:	Adaptive Search Approach
B&B	:	Branch and Bound
B&C	:	Branch and Cut
BCO	:	Bee Colony Optimization
CPU	:	Central Processing Unit
DIS	:	Displaced Ideal Solution
DM	:	Decision Maker
DP	:	Dynamic Programming
EAs	:	Evolutionary Algorithms
ECM	:	Elastic Constraint Method
ES	:	Exact Solution
F-FSS	:	Flexible Flow Shop Scheduling
F-JSS	:	Flexible Job Shop Scheduling
FSS	:	Flow Shop Scheduling
GA	:	Genetic Algorithm
GAM	:	Goal Attainment Method
GAMS	:	General Algebraic Modeling System
GCM	:	Global Criterion Method
GP	:	Goal Programming
GSA	:	Gravitational Search Algorithm

ICA	:	Imperialist Competitive Algorithm
ILP	:	Integer Linear Programming
ISCS	:	Integrated Supply Chain Scheduling
JSS	:	Job Shop Scheduling
LGP	:	Lexicographic Goal Programming
LNS	:	Large Neighborhood Search
LP	:	Linear Programming
LS	:	Local Search
MA	:	Memetic Algorithm
MATLAB	:	Matrix Laboratory
MILP	:	Mixed Integer Linear Programming
MINLP	:	Mixed Integer Nonlinear Programming
MODM	:	Multi Objective Decision Making
MOEAs	:	Multi Objective Evolutionary Algorithms
MOOPs	:	Multi Objective Optimization Problems
MOPSO	:	Multi Objective Particle Swarm Optimization
NP-hard	:	Non-deterministic Polynomial-time Hard
NS	:	Neighborhood Search
NSGA-II	:	Non-dominated Sorting Genetic Algorithm-II
OA	:	Order Acceptance
PESA-II	:	Pareto Envelope-based Selection Algorithm-II
PSO	:	Particle Swarm Optimization
RGGA	:	Reference Group Genetic Algorithm
RTT	:	Round Trip Transportation
SA	:	Simulated Annealing

SC	:	Supply Chain
SCIP	:	Solving Constraint Integer Programs
SCM	:	Supply Chain Management
SCN	:	Supply Chain Network
SCND	:	Supply Chain Network Design
SPEA-II	:	Strength Pareto Evolutionary Algorithm-II
WGA	:	Weighted Genetic Algorithm
WSM	:	Weighted Sum Method



LIST OF FIGURES

	PAGE
Figure 1.1. The Layout of Supply Chain Scheduling	3
Figure 2.1. The Classification of ISCS Models	8
Figure 2.2. Schematic of the Flexible Job Shop Problem (Bożek and Wysocki, 2015)	14
Figure 3.1. Integrated Supply Chain Scheduling in Flexible Job Shop Environment	31
Figure 4.1. Pareto Points for a Bi-Objective Optimization Problem	48
Figure 4.2. Goal Attainment Method for Optimization of a Bi-objective Problem	52
Figure 4.3. Solution with Weighted Sum Method for a) Convex b) Non-convex Solution Set	54
Figure 4.4. NSGA-II Algorithm Steps	58
Figure 4.5. Algorithm of Fast Non-Dominated Sorting in NSGA-II (Kaucic, 2008)	60
Figure 5.1. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 1	64
Figure 5.2. Generated Pareto Solutions according to a) NSGA-II method and b) WSM & GAM	72
Figure 5.3. Total Cost Values (Z_1) Obtained by All Methods for Test Problem 1	75
Figure 5.4. Total C_{max} Values (Z_2) Obtained by All Methods for Test Problem 1	75
Figure 5.5. The Gantt Chart of Test Problem 1	78

Figure 5.6. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 2	79
Figure 5.7. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 3	82
Figure 5.8. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 4	85
Figure 5.9. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 5	88
Figure 5.10. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 6	91
Figure 5.11. a) Total Cost and b) Total $C_{\max}$ Results Obtained by Individual Optimization for Test Problem 7	94
Figure 5.12. The Average Objective Function Values of a) Z_1 and b) Z_2 for Small and Medium Size Test Problems	97
Figure 5.13. Box plot of the Average CPU-Times Obtained Employing All Methods	98
Figure 5.14. The Average Objective Function Values of a) Z_1 and b) Z_2 for Large Size Test Problems	105
Figure 5.15. Box plot for Average Direct Distance Values of MODM Methods	106

LIST OF TABLES

	PAGE
Table 2.1. Several F-JSS Mathematical Models in Literature	16
Table 2.2. Supply Chain Scheduling Studies Based on the New Classification	20
Table 2.3. Goals, Objectives and Performance Measures of the Reviewed Studies	22
Table 3.1. The Conditions between X_{mht}^p , $X_{mh't}^{p'}$ and $(X_{mht}^p * X_{mh't}^{p'})$	42
Table 5.1. The Values of Parameters Based on Uniform Distributions	62
Table 5.2. The Generated Small and Medium Sized Problems Instances	63
Table 5.3. Weighted Sum Method with Different Weights for Test Problem 1	66
Table 5.4. Determining the Best Weight of WSM for Test Problem 1	67
Table 5.5. Goal Attainment with Different Weights for Test Problem 1	67
Table 5.6. Determining the Best Weight of GAM for Test Problem 1	68
Table 5.7. Objective Function Values and CPU time	69
Table 5.8. The Normalized Outputs of NSGA-II	70
Table 5.9. The Results Obtained by the Solution Methods for Test Problem 1	73
Table 5.10. Selection of Best Solution Method for Test Problem 1	74
Table 5.11. Total Amount of Processed Products on Machines	77
Table 5.12. Product Flow from Manufacturing Facility toward Warehouse	77
Table 5.13. Product Flow from Warehouse toward Customers	77

Table 5.14. Determining the Best Weight of WSM for Test Problem 2	80
Table 5.15. Determining the Best Weight of GAM for Test Problem 2	80
Table 5.16. The Results Obtained by the Solution Methods for Test Problem 2	81
Table 5.17. Selection of Best Method for Test Problem 2	81
Table 5.18. Determining the Best Weight of WSM for Test Problem 3	83
Table 5.19. Determining the Best Weight of GAM for Test Problem 3	83
Table 5.20. The Results Obtained by the Solution Methods for Test Problem 3	84
Table 5.21. The Results Obtained by the Solution Methods for Test Problem 3	84
Table 5.22. Determining the Best Weight of WSM for Test Problem 4	86
Table 5.23. Determining the Best Weight of GAM for Test Problem 4	86
Table 5.24. The Results Obtained by the Solution Methods for Test Problem 4	87
Table 5.25. Selection of Best Method for Test Problem 4	87
Table 5.26. Determining the Best Weight of WSM for Test Problem 5	89
Table 5.27. Determining the Best Weight of GAM for Test Problem 5	89
Table 5.28. The Results Obtained by the Solution Methods for Test Problem 5	90
Table 5.29. Selection of Best Solution Method for Test Problem 5	90
Table 5.30. Determining the Best Weight of WSM for Test Problem 6	92
Table 5.31. Determining the Best Weight of GAM for Test Problem 6	92
Table 5.32. The Results Obtained by the Solution Methods for Test Problem 6	93
Table 5.33. Selection of Best Method for Test Problem 6	93
Table 5.34. Determining the Best Weight of WSM for Test Problem 7	95

Table 5.35. Determining the Best Weight of GAM for Test Problem 7	95
Table 5.36. The Results Obtained by the Solution Methods for Test Problem 7	96
Table 5.37. Selection of best method for Test Problem 7	96
Table 5.38. The Generated Large Sized Problems Instances	99
Table 5.39. The Objective Functions Values Obtained by Individual Optimization	100
Table 5.40. Weighted Sum Method with Different Weights for Large Sized Test Problems	100
Table 5.41. GAM Results with Different Weights for Large Sized Test Problems	101
Table 5.42. The Results Obtained by LP metric/Global Criterion, Mini-max Goal and Lexicographic Programming Methods for Large Sized Test Problems	102
Table 5.43. The Results Obtained by the all MODM for Large Sized Test Problems	103

CHAPTER 1

INTRODUCTION

1.1. Supply Chain Scheduling

In general, Supply Chain (SC) not only involves transportation of the needed product from one location to another, but it also covers the entire production activities of the product, the procurement, outsourcing and distribution of the necessary raw materials and semi-finished products. While production activities involve sequencing and scheduling of customer orders at the manufacturing facility (or plant). The most important issue in Supply Chain Management (SCM) is the coordination and cooperation of all the stakeholders (suppliers, intermediaries, wholesalers, third party service providers, manufacturers and customers) in the supply network.

Due to global competition, companies, suppliers, producers, distributors and customers have realized that coordination is one of the key success factors in achieving efficient and effective SCM (Chen, 2010; Çalışkan, 2014; Pirim et al., 2014). For this reason, the coordination of production and distribution decisions such as production planning, distribution activities planning, supplier selection, and etc. in recent years has become one of the current issues in supply chain management studies for the last few years.

It is vital for the companies to be able to respond quickly to the needs of each customer in today's competitive environment. The fact that these requests are met by the business requires that not only the operation of the enterprise itself but also every activity of the supply chain is timely and of good quality. In order to reach high performance in a supply chain, it is essential to combine these functions and coordinate them together (Lawrence et al., 2010; Sarrafha et al., 2015; Sawik, 2016).

The studies that examine the detailed scheduling problem in the integrated supply, production-distribution models at the operational level are quite new. Hall and Potts (2003) have suggested Integrated Supply Chain Scheduling (ISCS) as a comprehensive tool for cooperation. Figure 1.1 illustrates a general layout of ISCS problem. The main

objective of ISCS problems is to obtain a suitable sequence of jobs, batches, machines, and etc. by considering different objective functions such as minimizing the total cost of the supply chain network (Hassanzadeh et al., 2016). Because, the classical scheduling models deal with the production scheduling problem without considering the total cost such as delivering cost, inventory cost, purchasing cost and etc. However, ISCS models provides a suitable sequence of jobs, machines, batches, etc. considering optimization of revenue, cost and customer service levels together in the supply chain.

In this thesis, a new bi-objective mathematical model for the ISCS problem was developed. It was assumed the manufacturing facilities produce the products in a F-JSS environment. Multi-product, multi-period and multi-echelon supply chain scheduling network was designed. The supply chain network including manufacturers, warehouses, and final customers was proposed.

The problem can be described as follows: The customer orders products from the manufacturer in a specific time period. Each product has a sequence of operations and are processed on a set of machines to complete product. The manufacturer processes the products on a set of multi-purpose machines in facility to find the optimum sequence. The finished products are delivered to the warehouses to be stored in order to meet the demands of the customers. Also, if customers' demands are not met, there may be shortage in terms of backorder for each period.

The proposed model has two objectives; (1) minimization of the total supply chain costs and (2) determination a proper schedule on machines that minimizes the makespan in all periods. However, these objectives may conflict with each other. This means that the minimization of makespan can cause increase in the total supply chain cost. The problem was first formulated as constrained bi-objective mixed integer nonlinear programming (MINLP) model. The MINLP model includes many nonlinear terms in both the objective and the constraints that greatly increase the computational complexity for solving the problem. Thus, in order to reduce the computational complexity, linearization techniques were applied to model.

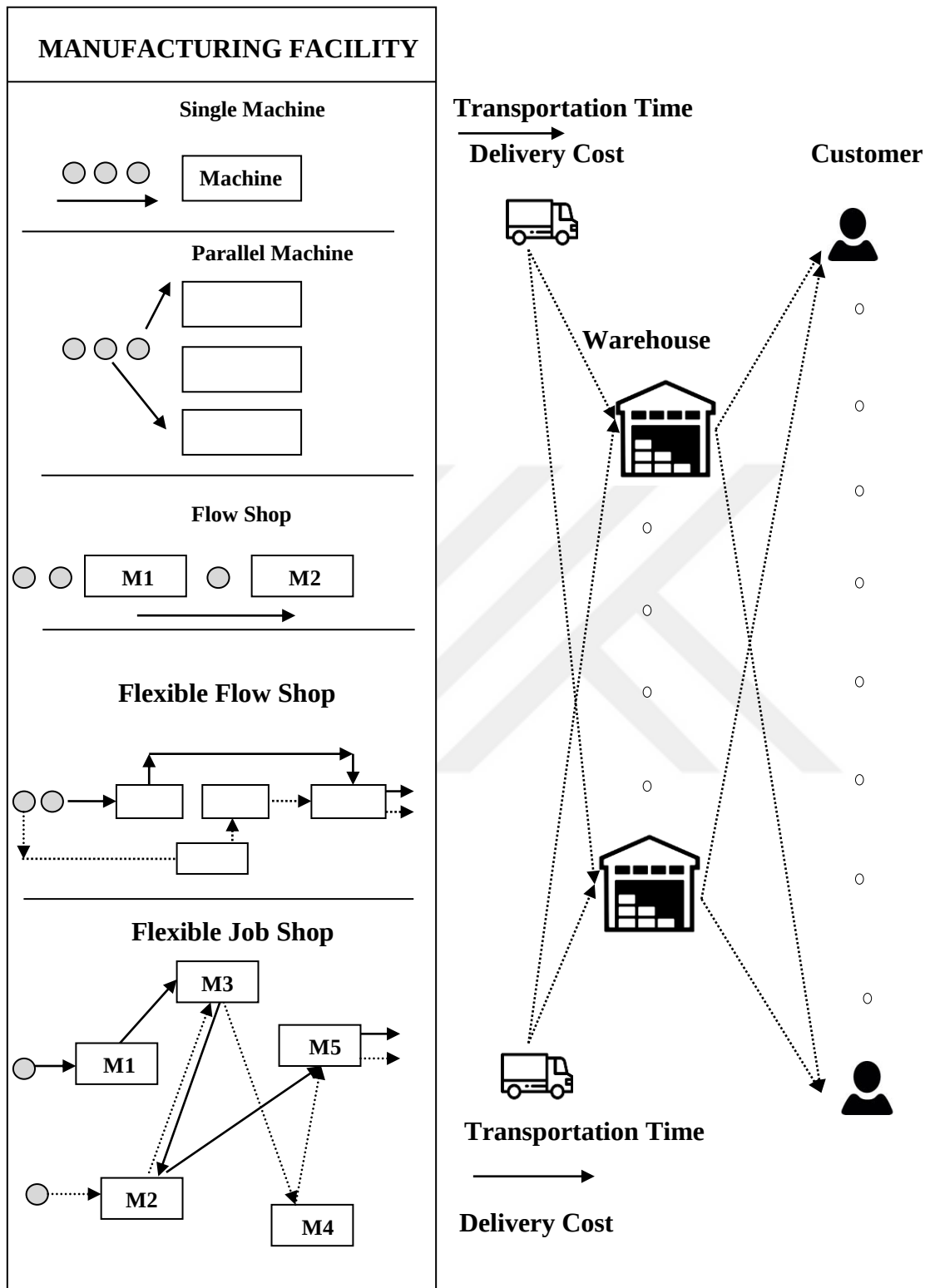


Figure 1.1. The Layout of Supply Chain Scheduling

1.2. Importance of the Problem

The integration of supply chain and scheduling is a complex optimization problem, so recently, researchers have placed increasing attention on to solve it. In the literature, there are extensive studies that has been focused on integrated scheduling models. The number of different types of problems encountered in many practical applications is increasing. In the literature, a large number of studies are carried out to formulate and solve supply chain scheduling problems. The majority of these efforts are mainly focused on single machine configuration and parallel machine configuration. Only a few authors have considered other general and complex configuration including flow shop and job shop scheduling (Sahraeian et al., 2017).

The main purpose of this Ph.D. thesis is to integrate the multi-echelon supply chain network and scheduling in F-JSS environment. In the majority of multi-echelon SCN problems in the literature, some predetermined numbers are used for the completion times of the manufacturing process. Also, these numbers are inserted into the model as input parameter, such as setup and process time, demand, and transportation costs. However, in this thesis the level of manufacturing was modeled as a F-JSS in the center of the entire supply chain network. In other words, the makespan was obtained through the proposed model. This leads difficulty for solving this model with integer variables due to the non-convex structure. Because, the required memory and computation time increase exponentially when the dimension (variables and constraints) of the problem is increased. The classical flexible job shop problem has proven to be strongly NP-hard (Non-deterministic Polynomial-time hard). Therefore, integrated supply chain scheduling problem in a flexible job shop environment is also NP-hard in strong sense.

The proposed bi- objective model minimizes both time and cost objectives simultaneously. In the literature, two different types of solution methods are used to solve multi-objective optimization models: (1) Multi-objective Optimization Techniques and (2) Multi-objective Decision Making (MODM) methods

Multi-objective optimization methods give a number of feasible solutions which are called Pareto sets. These Pareto optimal solutions offer various options to the decision maker. Unfortunately, these sets are often too large, it can become hard to select an optimal Pareto solution from the Pareto frontier. In these situations, MODM methods

have an important role for solving the multi-objective optimization problems (MOOPs) with different perspective which minimizes the deviation of each objective function from its individual optimal point (Pasandideh et al., 2015; Khalilpourazari and Khalilpourazary, 2017; Khalilpourazari and Khamseh, 2017). These methods use priority or weights with respect to information obtained from the decision maker and provide the decision maker with solutions.

In this thesis, five MODM methods including *Weighted Sum*, *LP Metric/Global Criterion*, *Min-Max*, *Goal Programming* and *Goal Attainment* and a Pareto-based meta-heuristic algorithm called *Non-dominated Sorting Genetic Algorithm-II (NSGA-II)* were applied for solving the proposed model. As a result, different solution sets were obtained with such techniques for the model. Thus, different preferences can be compared according to the importance of each objective function.

1.3. Outline of the Thesis

- In the following Chapter 2 gives general description by considering the research articles in the literature on ISCS problems. In this chapter, existing literature on ISCS models were reviewed and summarized by introducing the new classification scheme to help identify deficiencies in this area.
- Chapter 3, the problem description was provided and detailed formulations related to the mathematical model were explained.
- Chapter 4 contains all methods and techniques of study.
- In Chapter 5, the mathematical model presented in Chapter 3 was tested and computational experiments were provided for comparing the performance of the solution methods described in Chapter 4. In this chapter, the data used in this thesis was explained briefly.
- Finally, the thesis was concluded in Chapter 6. Moreover, some possible future research directions were stated.

CHAPTER 2

INTEGRATED SUPPLY CHAIN SCHEDULING MODELS AND LITERATURE REVIEW

The integration of supply chain and scheduling is a hard optimization problem. So recently, researchers have placed increasing attention on to solve it. In the literature, there are extensive studies that has been done on integrated scheduling models. Because, the different types of problems arise in many practical applications. As the scope of supply chain scheduling includes various problems, there is a need for an extensive classification on studies in detail.

This chapter presents a comprehensive literature review on the ISCS problems. Similarities and differences of ISCS problems were discussed. The various studies in the literature were summarized and classified into several different classes. The purpose of this chapter is to provide a literature review about ISCS models in the literature to help noticing deficiencies in this area.

The section of this chapter was organized as follows: Section 2.1 gives general description by considering the research articles in the literature on ISCS problems. Section 2.2 presents new classification of ISCS models. In this section, the studies were categorized by considering the features such as number of customers (single or multiple), product lifespan (limited or unlimited), order sizes (equal or general), vehicle characteristics (limited/sufficient and homogeneous/heterogeneous), machine configurations and number of objective function (single or multi objective). Furthermore, problem characteristics of mathematical models and the solution approaches were also discussed. Because of this thesis main scope is *Flexible Job Shop Scheduling*, in this section the comprehensive information about F-JSS were given in detail. Section 2.3 is the background and review of the literature on ISCS studies based on the new classification.

2.1. General Description of Supply Chain Scheduling Problems

In this section, general structures and characteristics of ISCS problems were described. In the following, a common ISCS problem was described and associated parameters and notations were defined (Chen, 2010). The ISCS problem is typically described as follows: At the beginning of the planning horizon, a manufacturer with one or more facilities takes m orders ($M = 1, 2, \dots, m$) from c ($c \geq 1$) customers ($C = 1, 2, \dots, c$) which may be established in different locations. $M_i \subseteq M$ is the subset of the orders placed of customer i , and $m_i = |M_i|$ for $i \in C$, where $M = M_1 \cup \dots \cup M_c$, and $m_1 + \dots + m_c = m$. The orders are processed and then delivered individually or in batches to the customers by $v \geq 1$ vehicles which can be vans, trucks and air flights. Each order $j \in M$ is processed by single or multiple machines according to the specific machine configuration. In this configuration if only one machine is needed the processing time is represented as p_j , while if multiple machines are needed the processing time is represented as p_{ij} .

2.2. Classification of Supply Chain Scheduling Models

In this section, a brief information about components of ISCS models was given. In Figure 2.1, the classes which form the ISCS models were shown. Each class gives a brief information about number of customers (single or multiple), vehicle characteristics (limited/sufficient and homogeneous/heterogeneous), order sizes (equal or general), machine configurations, product shelf life (limited or unlimited) and number of objective function (single or multi objective), respectively. Furthermore, other details such as optimality properties, modeling method and solution approaches of the problems also mentioned.

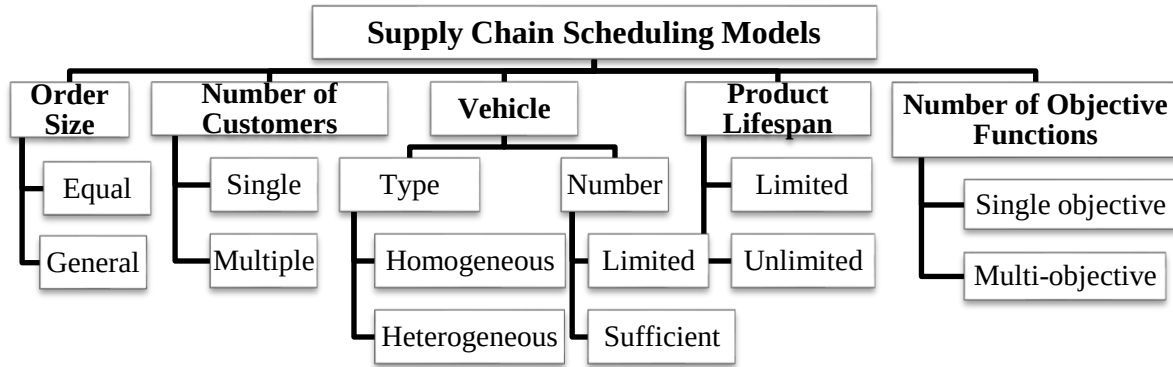


Figure 2.1. The Classification of ISCS Models

2.2.1. Order size

The manufacturer processes the orders first and then deliver the finished orders to the customers individually or in batches. Each order includes a set of products. For example, different orders may have varying order sizes and the complexity of different products may be different. In Existing ISCS models in the literature, order sizes are divided into two different class: (1) models where all the orders have the same (equal) size; and (2) models where the orders have generally. Available studies in literature show that the majority of the ISCS problems have equal size orders (Hall and Pots, 2003; Steiner and Zhang, 2009; Hamidinia et al., 2012; Mazdeh and Rostami, 2014; Ahmedizar and Farhadi, 2015; Hassanzadeh et al., 2016; Marandi, 2017). Because, the modeling ISCS problems with general order sizes is more complex than problems with equal order sizes.

2.2.2. Number of customers

It is assumed that there are one or more customers in the studies and each customer is located in different places. In some situations, orders from customers in the same location can be combined for delivery. However, vast majority of the previous studies reported in literature are mainly focused on the situation of multiple-customers (Chen and Vairaktarakis, 2005; Mazdeh et al., 2011; Rasti Barzoki and Hejazi, 2013; Gao et al., 2015; Cheng et al., 2017; Rostami et al., 2018).

Three possible situations under this section are as follows:

- Single customer ($c = 1$)
- Multiple customers ($c; c \geq 2$)
- n customers, means that each order belongs to a different customer

2.2.3. Vehicle characteristics

When an order is produced and completed, it should be delivered to the its customer by vehicles (e.g., vans, air flights, and trucks and etc.). Because, generally manufacturers and customers are located in different locations. Thus, for each completed order, the decision maker should decide when it will be delivered, which vehicle will be loaded and if there are multiple customers, the route through which the transportation system will be travelled.

The vehicles may be of different types and capacities. Most of ISCS studies in the literature assume either single vehicle or sufficient (unlimited) number of vehicles. However, the number of vehicles available is multiple, but limited in many world applications. Such problems are more difficult than single vehicle or sufficient vehicle problems. Moreover, vehicle capacity may be *limited* (i.e., can carry only a certain amount of orders in a shipment) or *unlimited* (i.e., can carry any number of orders in one shipment).

The vehicles can be *homogeneous* or *heterogeneous*. If all the vehicles have the same capacity, speed or same fixed and variable cost values, these problems are called as homogeneous. Otherwise, if one or more features are different for the vehicles, these types of problems are known as heterogeneous. Few studies in the literature have considered heterogeneous vehicles (Stecko and Zhao, 2007; Ullrich, 2013; Lee et al., 2014; Joo and Kim, 2017; Rafiei et al., 2018; Beheshtinia et al., 2018). Because, the modeling ISCS problems with heterogeneous vehicles is more complex than problems including homogeneous vehicles for delivery tasks. They are also known to be an NP-hard problem.

Furthermore, the most *recent studies* in ISCS problems are conducted on the third-party logistics (3PLs). 3PLs includes part or all of the company's supply chain and logistics operations being carried out by another companies. The usage of 3PLs can provide the

company more competitive by reducing the company's total cost and customer delivery times. Most 3PL problems assumes infinite number of vehicles available (Stecke and Zhao, 2007, Noroozi et al., 2017, Noroozi et al., 2018)

2.2.4. Objective function

Most of the available ISCS problems in the literature are concerned with following performance measurements: customer service level, sum of costs and total profit. These metrics can be listed as follow: (i) time-based, (ii) cost-based, and (iii) revenue-based objective functions.

2.2.4.1. Time-based objective functions

In ISCS problems, time-based measures are same with classical production scheduling problems. The common used *time-based performance metrics* are listed below:

- C_{max} = makespan
- L_{max} = maximum delivery lateness
- F_{max} = maximum flow time
- D_{max} = maximum delivery time
- $\sum D_j$ = total delivery time
- $\sum(w_j)D_j$ = total weighted delivery time
- $\sum(w_j)E_j$ = total weighted earliness
- $\sum(w_j)U_j$ = total weighted number of late orders,
- $\sum(w_j)F_j$ = total weighted flow time.
- $\sum(w_j)T_j$ = total weighted delivery tardiness and etc.

2.2.4.2. Cost-based objective functions

In this study, cost-based performance metrics can be discussed in sub-categories which are (i) total trip-based transportation cost (TC), (ii) total vehicle-based transportation cost (VC), and (iii) total production cost (PC), and others. TC is the total costs of all delivery trips where the cost of a delivery trip can consist of a fixed cost and a variable cost determined by the shipping path; i.e. total distance or the number of shipping orders. VC can be implemented in situations where a fixed cost has arisen regardless of

how many times a vehicle is turned off. PC shows the production cost of the orders, and this cost can be applied when the orders change according to the production schedule.

2.2.4.3. Revenue-based objective functions

Firms are profit-seeking organizations. Their main goal is to maximize the surplus of income over expenditures within certain limits. Different performance metrics are used in ISCS problems except of these (e.g., capacity utilization of distribution vehicles, quality of service, quality of product total satisfied demand and etc).

The problems are modeled as minimization if they have goals based on time and cost functions, or sum of these, and are modeled as maximization if they have revenue-based goals. It is always desired that the whole process has a short time, low cost or high profit. But, these objectives may conflict with each other. For example, having fewer vehicles at the delivery of goods may reduce the cost, but can also cause customers to extend the time to receive goods. For this reason, ISCS problem is a multi-objective optimization problem in real life.

Many ISCS problems in the literature are aimed at optimizing a single objective or multi-objectives together. In many real-world applications, it may be required to focus on multi-objectives at the same time. However, studies dealing with multi objectives are relatively low. There are three common approaches for solution of multi-objective ISCS problems: (i) *Constrained Optimization*, (ii) *Linear Combination*, and (iii) *Pareto Optimization*. Constrained optimization problem is a commonly used method in which one objective is minimized and the other is bounded. In the linear combination method, all objectives are aggregated into a single objective function. In the Pareto optimization method, independent objectives are combined to find a well-represented set of Pareto-optimal front. Because of complexity of this method, intelligent meta-heuristic approaches (evolutionary algorithms or swarm-based techniques) are used.

2.2.5. Machine configuration

A set of n orders is to be processed at a manufacturing facility. Depending on specific problem characteristics, the facility can include a single machine, a set of parallel machines, or a series of flow shop machines (regular flow shop, no-wait flow shop,

flexible flow shop, assembly flow shop), job shops and others.

Each of these machine configurations has been studied extensively in the literature on ISCS problems and modeled with different constraints, conditions and assumptions according to problem type. Moreover, studies with multi manufacturing facility (plant) which can be placed in different locations are frequently studied in recent years (Yılmaz and Pardalos, 2017; Rafiei et al., 2018; Beheshtinia et al., 2018). In ISCS models with multiple facilities, each order may be processed at different facilities and delivered together.

These classical machine structures are considered in below:

- *The single-machine scheduling configuration* is the most basic structure. In such a machine environment all orders are processed on a single machine.
- *Parallel-machine configuration* is similar to the single machine because it is a one-stage processing configuration. However, it includes m identical machines in parallel. Each order needs a single operation and may be processed on any one of the m machines.
- *Flow Shop Scheduling (FSS) configuration* is multi-stage production structures where there are m machines in series. In flow shop machine structure, each order has to be processed by all machines sequentially and all orders follow the same route.
- *Flexible Flow Shops Scheduling (F-FSS) configuration* is an extension of simple flow shops that have a special configuration that combines several properties of both the flow shop and the parallel machine scheduling.
- On the other hand, *Job Shop Scheduling (JSS) configurations* has a more general structure than the flow shop in which each order has its own predetermined route for visiting machines. In the JSSP, there are n jobs ($J_1, \dots, J_n$) with various sizes and each job requires to be scheduled on m ($m_1, \dots, m_m$) machines each job proceeds through a predetermined operational route

until all job operational requirements are satisfied. In the classical JSS problem, the process plan of a part includes the sequence of the machines visited by parts: there is an a priori assignment of operations to machines. Thus, the process plan is certain and process plan flexibility is not allowed (Roshanaei, 2012).

- *Flexible Job Shop Scheduling (F-JSS) configuration* is a generalization of the classical JSS which permits an operation to be processed on a given machine by from a given set. In other words, the main difference between JSP and FJSP is that the latter should choose a machine from a set of candidate machines to perform an operation (Türkyılmaz and Bulkan, 2015). Because of this thesis main scope is F-JSS, in this section how F-JSSP works were explained in detail.

P. Bruker and Schlie (1990) were among the first to address this problem. Flexible Job Shop Scheduling (F-JSS) problem is an extension of classical job shop scheduling problem in that operations are processed on any machine from a given machines sets. (Kim and Egbelu, 1999; Low and Wu; 2006; Demir and İşleyen, 2013; Chaudhry and Khan, 2016).

A general F-JSS problem may be formulated as follows:

- The set of m machines is represented as: $M = \{M_1, M_2, \dots, M_m\}$.
- There is a set of n jobs to be processed on m machines.
- Each job i has a sequence of n_i operations as: $(O_{i,1}, O_{i,2}, \dots, O_{i,n_i})$.
- In order to complete the job, all operations have to be performed. Each operation j of job i ($O_{i,j}$) processed on any among a set of available machines $M_{i,j}$.

Figure 2.2 shows a general schematic of the flexible job shop problem.

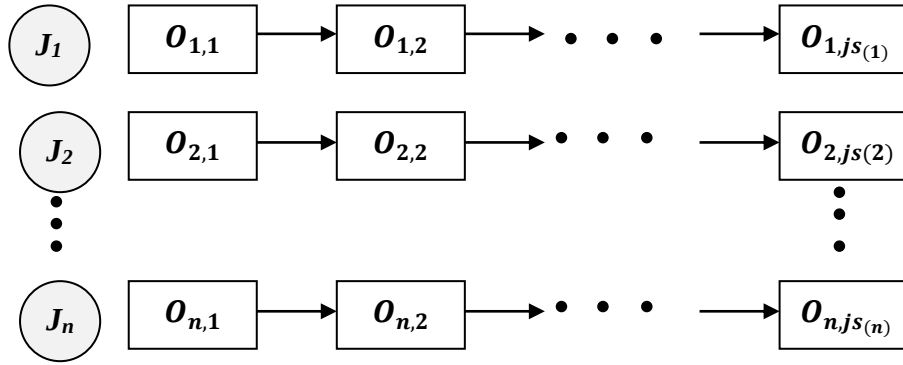


Figure 2.2. Schematic of the Flexible Job Shop Problem (Bożek and Wysocki, 2015)

The following assumptions are considered in a general F-JSS problem:

- All jobs and machines are available at beginning where time $t = 0$.
- All jobs are independent of each other.
- All jobs have equal priorities.
- A machine can only process one operation at a time.
- Jobs are independent from each other. There are no precedence constraints among the operations of different jobs.
- Each operation can be performed by only one machine at a time.
- Preemption of operations is not permitted. An operation cannot be stopped or interrupted during its processing.

The problem of scheduling jobs in F-JSP can be decomposed into two sub-problems: (i) the processing order of jobs on machines (sequencing) and (ii) assignment of each operations to a machine selected out of a set of capable machines. Since, F-JSS considers the machine assignment for each operation, it is more complex than classic JSS problem (Zhang et al., 2012). The F-JSS is a hard problem due to the expanding machine capabilities and increased product variety. Because of computational complexity, F-JSS is strongly NP-hard. (Choi and Choi, 2001; Fattahi et al., 2007).

For this reason, different solution methods and new algorithms have been presented to solve this problem. Due to complexity of F-JSS problem, many researchers have developed from various mathematical methods to different meta-heuristics such as evolutionary algorithms (Shokoui, 2018), tabu search algorithm (Imanipour, 2006;

Saidi-Mehrabad and Fattahi, 2007; Fattahi et al., 2007; Mousakhani 2103; Shen et al., 2018) genetic algorithm (Tamaki et al., 2001; Gao et al., 2006; Jalilvand-Nejad and Fattahi, 2015; Shokoui, 2018), simulated annealing (Fattahi et al., 2007; Jalilvand-Nejad and Fattahi, 2015), and so on. Table 2.1 shows an overview of F-JSP mathematical models in the literature.



Table 2.1. Several F-JSS Mathematical Models in Literature

Year	Author(s)	Modeling Method	Highlights	No of objective Function		Objectives	Solution Approaches
				Single	Multi		
(1997)	Liu and MacCarthy	MILP	Sequence dependent setup times Transportation times Limited Buffers	✓		Mean completion time, Makespan (Cmax), Maximum tardiness	Heuristic
(2001)	Tamaki et al.	MILP	Sequence independent setup times	✓		Cmax, Total tardiness	Genetic Algorithm
(2002)	Choi and Choi	MILP	Multiple process plans, Sequence dependent setup times	✓		Cmax	Local Search
(2006)	Gao et al.	MINLP	Preventive maintenance tasks	✓		Cmax, Maximal workload at any machine, Total workload overall machines	Local Search Hybrid Genetic Algorithm
(2006)	Imanipour	MINLP	Process planning and sequence flexibility Sequence dependent setup times	✓		Cmax	Tabu search, Greedy Neighborhood Search
(2007)	Saidi-Mehrabad and Fattahi	MILP	Sequence dependent setup times	✓		Cmax	Branch and Bound, Tabu Search
(2007)	Fattahi et al.	MILP	-	✓		Cmax	Branch and Bound, Tabu Search, Simulated annealing
(2010)	Ozguven et al.	MILP	Process plan flexibility	✓		Cmax	Branch and Bound method Heuristic Algorithms
(2013)	Mousakhani	MILP	Sequence dependent setup times	✓		Total Tardiness	Iterated local search, Neighborhood Search Algorithms, Tabu Search
(2015)	Jalilvand-Nejad and Fattahi	MILP	Sequence dependent setup times	✓		Delay costs, Setup costs and Holding cost	Simulated Annealing Genetic Algorithm
(2016)	Karimi et al.	MILP	With transportation times	✓		Cmax	Hybrid Imperialist Competitive Algorithm
(2018)	Shen et al.	MILP	Sequence dependent setup times	✓		Cmax	Tabu Search
(2018)	Shokoui	MILP	Precedence Constraints		✓	Cmax, Critical Machine Workload Machine, Total Workload	NSGA-II Weighted Genetic Algorithms (WGA)

2.2.6. Product lifespan

Another important feature that directly affects the complexity of the ISCS models is the lifespan of the products. In classic ISCS models, it is assumed that products can be stored and delivered indefinitely without loss of value. However, some types of products may deteriorate over time, or lose value due to changes in fashion or technology. These products have a very short life cycle. For example; after mixing the compulsory raw materials (water, cement, gravel, sand, etc.), the resultant concrete dough becomes solid and therefore useless in an hour. Similarly, after the production of plywood panels, some of the adhesive materials can lose power in seven days. These products should be delivered to the customer as soon as possible after production (Lacomme et al., 2018).

In all these cases, it is necessary to schedule the production and delivery of the finished product in an integrated manner. Studies on products that are quickly perishable in integrated supply chain scheduling problems are considered to be a new field of study in the literature.

There are various integrated production and distribution problems in the literature that consider rapidly perishable products such as fresh food, fruit, fast frozen cement, platelets (Devapriya, 2008; Çalışkan, 2014; Viergutz and Knust, 2014; Lee et al., 2014; Devapriya, 2017; Karaoğlu and Kesen, 2017; Marandi, 2017; Sağlam and Banerjee, 2018). In these models, there is generally a direct connection between production and distribution operations.

Due to the nature of the products that are quickly deteriorated, the production systems that produce these products adopt the *make-to-order* philosophy, not the *make-to-stock* philosophy. For this reason, in these type problems, there are zero or almost zero finished product stocks in the production facilities.

2.2.7. Modeling methods and solution approaches

The ISCS problems are extensively studied in the literature with various models and different objective functions and constraints. Several optimization techniques have been used in ISCS problems. Mathematical programming, especially Integer / Mixed Integer

Programming has become one of the most commonly applied approach in ISCS problems due to its sensitivity, flexibility and widespread modeling capability (Devapriya, 2008; Ullrich, 2013; Viergutz and Knust, 2014; Pei et al., 2014; Kang et al., 2016; Karaoğlu and Kesen, 2017; Gharaei and Jolai, 2018). The two most important enumerative methods: (i) Dynamic Programming (DP), (ii) Branch and Bound (abbreviated further on as B&B), Branch and Cut (B&C) are commonly used to solve especially small sized ISCS problems.

Moreover, the addition of new constraints and assumptions directly affect the dimension, formulation, complexity, computational time and solution approach of the model. The growth of the model's size causes to the difficulty of solving it with classic analytical methods. It is obvious that the ISCS problems are more complex than classical scheduling problems.

Batch delivery problems are strongly NP-hard because of combinatorial complexity. Therefore, the integrated ISCS problems are strongly NP-hard. Because of the difficulty of the problem, optimal benchmark for especially large sized problems cannot be computed. Because of the difficulty approximation algorithms, heuristics, meta-heuristics and hybrid meta-heuristics have been developed. Heuristics are useful and give simple solutions for large scale size problems (Stecke and Zhao, 2007; Rasti Barzoki and Hejazi, 2013; Gao et al., 2015). Meta heuristics such as classical (Simulated Annealing, Tabu Search), nature-inspired algorithms (Ant Colony Optimization, Bee Colony Optimization) and evolutionary algorithms (Genetic and Memetic) are developed for various ISCS problems. Each approach has different properties that make it suitable for application to specific problems. Genetic algorithms have been used in a wide variety of applications.

In recent years, the number of strategies proposed to solve complex optimization problems has increased significantly. For example, hybrid meta-heuristic which is combination of other optimization methods provides more efficient behavior and greater flexibility on complex problems (Hassanzadeh et al., 2016; Noorozi et al., 2017; Yılmaz and Pardalos, 2017; Gharaei and Jolai, 2018). The reason for this is that hybrid meta-heuristics use both their advantages and complementary strengths, for instance, more classical optimization techniques such as branch and bound or dynamic programming.

In this way, more complex but more realistic models can be formulated to obtain good approximate solutions in a reasonable run time.

2.3. Literature Survey

In this section, the studies on ISCS problems in the literature were classified in tabular forms. As the scope of ISCS involves different types of problems, it is important to examine each selected problem in detail in terms of proper timing and performance. Table 2.2 gives an overview about several ISCS studies based on the above classification. It also benchmarks the related literature and compares the existing problems. Table 2.3 presents the mentioned objectives in the ISCS models examined in this thesis.

The study of Hall and Potts (2003) is the first study of *Supply Chain Scheduling* in the literature. They integrated scheduling, batching and delivery on a single machine, under the assumption of the existence of batches along with different objectives such as number of late jobs (number of tardy jobs), sum of flow time, total delivery cost, and maximum lateness. They developed an efficient dynamic programming solution algorithm to minimize the above-mentioned goals. Chen and Vairaktarakis (2005) studied the integrated scheduling model of production and distribution operations in order to optimize the tradeoff between customer service level and distribution costs.

Table 2.2. Supply Chain Scheduling Studies Based on the New Classification

Year	Author(s)	Order Sizes		Number of customers		Number of objective func.		Product Lifetime		Vehicle Characteristics		Machine configuration	Modeling Method	Solution Approach
		Equal	General	Single	Multiple	Single	Multi	Unlimited	Limited	Type	Number			
2003	Hall and Potts	✓		✓	✓	✓		✓		Homogeneous	Sufficient	Single Machine	DP	DP Algorithms
2005	Chen and Vairaktarakis	✓			✓	✓		✓		Homogeneous	Sufficient	Single/Parallel Machine	DP	Exact and Heuristic Algorithms
2007	Stecke and Zhao	✓			✓	✓		✓		Heterogeneous	Sufficient	Single Machine	MIP	Heuristic
2008	Devapriya	✓			✓	✓			✓	Homogeneous	Sufficient	Single Facility	MIP	Exact, GA, MA
2009	Steiner and Zhang	✓			✓	✓		✓		-	-	Single Machine	DP	Approximation Algorithms
2011	Mazdeh et al.		✓		✓	✓		✓		-	-	Single Machine	MINLP	Exact, SA
2012	Hamidinia et al.	✓			✓	✓		✓		-	-	Single Machine	IP	GA
2013	Ullrich		✓		✓	✓		✓		Heterogeneous	Limited	Parallel Machine	MIP	GA
2013	Rasti Barzoki and Hejazi	✓			✓	✓		✓		Homogeneous	Sufficient	Single Machine	IP	Exact, B&B Heuristics
2014	Viergutz and Knust	✓			✓	✓			✓	Homogeneous	Limited	Single Facility	MIP	Exact, Heuristics
2014	Lee et al.		✓		✓	✓			✓	Heterogeneous	Limited	Parallel	MIP	Exact, LNS Algorithm
2014	Mazdeh and Rostami	✓		✓		✓		✓		Homogeneous	Sufficient	Flowshop	MIP	B&B, Heuristics
2014	Pei et al.		✓	✓		✓		✓		Homogeneous	Sufficient	Serial Machines	MIP	GSA, GA & PSO
2015	Gao et al.	✓			✓	✓		✓		Homogeneous	Limited	Single Machine	MIP	Exact, Heuristic
2015	Fan et al.	✓		✓		✓		✓		Homogeneous	Sufficient	Single Machine	DP	DP and Approximation Algorithms
2015	Ahmedizar and Farhadi	✓			✓	✓		✓		-	-	Single Machine	MIP	ICA
Modeling Methodology: IP: Integer programming, MILP: Mixed Integer Programming, MINLP: Mixed Integer Nonlinear Programming, DP: Dynamic Programming Solution Approach: Exact: Exact Solution, GA: Genetic Algorithm, PSO: Particle Swarm Optimization, GSA: Gravitational Search Algorithm, LNS: Large Neighborhood Search, SA: Simulated Annealing, B&B: Branch and Bound, ECM: Elastic Constraint Method, B&C: Branch and Cut, MA: Memetic Algorithm, ICA: Imperialist Competitive Algorithm														

Table 2.2. Supply Chain Scheduling Studies Based on the New Classification (Continuation)

Year	Author(s)	Order Sizes		Number of customers		Number of objective func.		Product Lifetime		Vehicle Characteristics		Machine configuration	Modeling Method	Solution Approach
		Equal	General	Single	Multiple	Single	Multi	Unlimited	Limited	Type	Number			
2015	Rostami et al.	✓		✓		✓		✓		Homogeneous	Sufficient	Single Machine	MIP	B&B, GA& PSO
2016	Kang et al.		✓		✓	✓		✓		Homogeneous	Limited	Single Facility	MIP	Exact, GA
2016	Assarzadegan and R.-Barzoki	✓			✓	✓		✓		Homogeneous	Sufficient	Single Machine	MIP / MINLP	GA, SA
2016	Hassanzadeh et al.	✓		✓			✓	✓		Homogeneous	Sufficient	Flow shop	MINLP	Hybrid Meta-heuristics
2017	Yılmaz and Pardalos		✓		✓	✓		✓		Homogeneous	Sufficient	Multi-Facility	MIP	Hybrid Meta-heuristics
2017	Joo and Kim		✓		✓	✓		✓		Heterogeneous	Limited	Unrelated Parallel Machines	MIP	Exact, Meta-heuristics
2017	Noroozi et al.		✓		✓	✓		✓		Heterogeneous	Limited	Single Machine	MIP	Hybrid Meta-heuristics
2017	Karaoğlu and Erhan		✓		✓	✓			✓	Homogeneous	Limited	Single Facility	MIP	B&C
2017	Devapriya et al.		✓		✓	✓			✓	Homogeneous	Sufficient	Single Facility	MIP	Exact, GA, MA
2017	Cheng et al.	✓	✓	✓		✓		✓		Homogeneous	Sufficient	Parallel Machine	MIP	Approximation Algorithms
2017	Kazemi et al.		✓		✓	✓		✓		-	-	Flow shop	MINLP	ICA
2017	Marandi	✓			✓	✓			✓	Heterogeneous	Limited	Flow shop	IP	Heuristic
2018	Noorozi et al.		✓		✓	✓		✓		Homogeneous	Sufficient	Single Machine	MIP	GA
2018	Gharaei and Jolai		✓		✓		✓	✓		Homogeneous	Sufficient	Parallel Machine	MIP	Hybrid Meta-heuristics
2018	Rostami et al.		✓	✓		✓		✓		Homogeneous	Limited	Single Machine	MIP	B&B, SA
2018	Rafiei et al.		✓		✓		✓	✓		Heterogeneous	Sufficient	Multi Facility	MIP	Exact, ECM
2018	Beheshtinia et al.		✓		✓	✓		✓		Heterogeneous	Limited	(Parallel) Multi-Facility	MIP	Hybrid GA
Modeling Methodology: IP: Integer programming, MILP: Mixed Integer Programming, MINLP: Mixed Integer Nonlinear Programming, DP: Dynamic Programming Solution Approach: Exact: Exact Solution, GA: Genetic Algorithm, PSO: Particle Swarm Optimization, GSA: Gravitational Search Algorithm, LNS: Large Neighborhood Search, SA: Simulated Annealing, B&B: Branch and Bound, ECM: Elastic Constraint Method, B&C: Branch and Cut, MA: Memetic Algorithm, ICA: Imperialist Competitive Algorithm														

Table 2.3. Goals, Objectives and Performance Measures of the Reviewed Studies

Author(s) / (Year)	Maximum Lateness	Maximum Tardiness	Total (Weighted) Tardiness	Total (Weighted) Earliness	Makespan	Average Lead Time	Total (Weighted) Completion Time	Delivery Time	Production Time	Number of Tardy Job	Flow Time	Customer Service Level	Service Span	Fixed Vehicle Cost	Staff Training Cost	Delivery Cost	Hiring Cost (Facility)	Raw Material Cost	Production Cost	Inventory Holding Cost	Earliness Cost	Tardiness Cost	Travel Cost	Service Level	Total Net Profit	Distribution Cost	Transportation Cost	Total Satisfied Demand	Number of Vehicle	Due Date Assignment Cost
Hall and Potts (2003)	✓									✓	✓					✓														
Chen and Vairaktarakis (2005)												✓														✓				
Stecke and Zhao (2007)																											✓			
Devapriya (2008)																	✓										✓			
Steiner and Zhang (2009)										✓						✓														
Mazdeh et al. (2011)										✓						✓														
Ullrich (2013)			✓																											
Hamidinia et al. (2012)			✓	✓												✓				✓										
R.-Barzoki and Hejazi (2013)										✓						✓													✓	
Viergutz and Knust (2014)										✓																	✓			
Lee et al. (2014)														✓					✓				✓							
Mazdeh and Rostami (2014)		✓														✓														
Pei et al. (2014)					✓																									
Gao et al. (2015)								✓	✓																					

Table 2.3. Goals, Objectives and Performance Measures of the Reviewed Studies

Author(s) / (Year)	Maximum Lateness	Maximum Tardiness	Total (Weighted) Tardiness	Total (Weighted) Earliness	Makespan	Average Lead Time	Total (Weighted) Completion Time	Delivery Time	Production Time	Number of Tardy Job	Flow Time	Customer Service Level	Service Span	Fixed Vehicle Cost	Staff Training Cost	Delivery Cost	Hiring Cost (Facility)	Raw Material Cost	Production Cost	Inventory Holding Cost	Earliness Cost	Tardiness Cost	Travel Cost	Service Level	Total Net Profit	Distribution Cost	Transportation Cost	Total Satisfied Demand	Number of Vehicle	Due Date Assignment Cost
Fan et al. (2015)								✓								✓														
Ahmedizar and Farhadi (2015)																✓				✓	✓	✓								
Rostami et al. (2015)		✓														✓														
Kang et al. (2016)																			✓								✓			
Assarzadegan&Barzoki (2016)		✓														✓														✓
Hassanzadeh et al. (2016)			✓	✓	✓					✓						✓				✓										
Yilmaz and Pardalos (2017)								✓																						
Joo and Kim (2017)					✓																									
Noorozi et al. (2017)																						✓		✓		✓				
Karaoğlu and Erhan (2017)					✓			✓								✓														
Devapriya et al. (2017)								✓																			✓	✓		
Chen et al. (2017)													✓																	
Kazemi et al. (2017)			✓													✓														
Marandi (2017)					✓																							✓	✓	
Noorozi et al. (2018)																✓						✓		✓				✓		
Gharaei and Jolai (2018)			✓																							✓				
Rostami et al. (2018)							✓	✓																						
Refiei et al. (2018)															✓		✓		✓	✓				✓			✓			
Beheshtinia et al. (2018)								✓																						

They investigated both single and parallel machine environment. They considered that there were unlimited of vehicles but the capacity of all vehicles was limited. They developed exact and heuristic algorithms to solve the problem. Stecké and Zhao (2007) developed a MIP model for solving of a commit to delivery business mode problem to minimize shipping cost. They provided an efficient and effective heuristic algorithm with polynomial time in order to find a near optimal solution for the NP-hard problem. They considered that transportation was done by 3PLs companies which involve heterogeneous vehicles.

Devapriya (2008) focused on ISCS problem with one production facility and a large fleet size. The problem was modeled considering lifespan of the perishable products. The objective function of problem includes sum of two cost components. First is the variable transportation cost relevant to both plants. Second is the fixed hiring cost of both plants. Steiner and Zhang (2009) have addressed a two-level supply chain scheduling problem with a supplier and multiple customers. In their study, the orders are processed on a single machine and delivered in batches to customers. They proposed an polynomial time approximation algorithm to minimize the sum of the weighted number of late jobs and delivery cost. Mazdeh et al. (2011) developed a MINLP model to minimize the sum of the total weighted number of tardy jobs and delivery costs on the single machine for multiple customers. They used Simulated Annealing (SA) as a meta-heuristic method and compared its results with optimal solutions. Numerical results of study showed that the proposed SA algorithm requires low computation time in terms of the solution performance.

Hamidinia et al. (2012) proposed an effective genetic algorithm to minimize earliness, tardiness, delivery and inventory cost in a batch delivery system. The analysis results showed that the proposed genetic algorithm performed better than the classical genetic algorithm under non-batched systems. Rasti-Barzoki and Hajezi (2013) studied production and delivery scheduling integration problem. They proposed Integer Programming (IP) model and presented a heuristic and B&B algorithm for minimizing sum of the total weighed number of tardy jobs, the total due date assignment costs and delivery costs for multiple customer. The results of study demonstrated that the B&B was more efficient than CPLEX. Ullrich (2013) studied ISCS problem in which jobs are

processed on parallel machines with machine-dependent ready times. GA was proposed to solve MIP model. Heterogeneous vehicles with respect to their capacity and ready times were used for transportation of orders to the customers. They showed that the GA outperforms especially for small-size instances.

Viergutz and Knust (2014) studied and extended ISCS problem with lifetime (e.g., perishable or seasonal) constraints. In their problem, there is single product with a short lifetime which is produced at a single production plant with a limited production rate. The model was formulated as maximization problem such that the total satisfied demand is maximized. Lee et al. (2014) formulated a MIP model for a nuclear medicine production and delivery problem to minimize the total system cost including production costs, fixed vehicle costs and travel costs. They used a substance which has a limited half-life. They proposed a variant of a large neighborhood search (LNS) based algorithm and presented a benchmark data set. They showed that the proposed approach performs well on the benchmark instances. Mazdeh and Rostami (2014) extended ISCS problem for a two-machine flow-shop environment. They aimed to minimize maximum tardiness and delivery costs in a batched delivery system. MIP model was provided and solved using a B&B algorithm to obtain the optimum solution. Moreover, they also presented an upper bound (UB) heuristic with a quick processing time.

Pei et al. (2014) focused on a two-stage supply chain scheduling problem considering multiple manufacturers and different job sizes. The problem was formalized as a mixed integer programming model to minimize the makespan. A global optimization algorithm called Modified Gravitational Search Algorithm (MGSA) is developed to solve the problem. They evaluated effectiveness and efficiency performance of the proposed algorithm, MGSA, compared with PSO and GA. Based on the experimental results, they showed that MGSA has faster convergence speed in solving problems. Ghao et al. (2015) studied the no-wait condition between the production and distribution of each batch with limited vehicle capacity. They developed a heuristic with a guaranteed performance.

Fan et al. (2015) studied the objective of minimizing the sum of delivery time and total delivery cost caused by one vehicle without capacity constraint. They provided an approximation algorithm to solve the problem. They focused on non-availability single

machine constraint (machine can be unavailable due to regular preventive maintenance or unexpected breakdowns). Ahmadizar and Farhadi (2015) extended the problem of Hamidinia et al. (2012) by considering due windows and job release dates. They proposed a solution method with incorporating the dominance properties with an imperialist competitive algorithm. The computational experiments showed that the proposed hybrid algorithm can give optimal or near-optimal solutions in reasonable time.

Rostami et al. (2015) aimed to minimize the sum of maximum tardiness and delivery cost on the single machine with release times. They provided a branch and bound algorithm and two meta-heuristic methods for solving MIP model. The results of study showed that two meta-heuristic methods outperform on extremely large sizes within a reasonable time. Kang et al. (2016) constructed the ISCS problem as MIP model to minimize total production and transportation cost. Then, they proposed Genetic Algorithm (GA) to find near-optimal solutions. In the study, they emphasized that the proposed GA model was an effective and useful method. Assarzaghan and R.-Barzoki (2016) developed MINLP and MIP models to minimization of total costs which includes maximum tardiness, due date assignment and delivery costs. As the developed model is NP-hard, they used meta-heuristic algorithms, an Adaptive Genetic Algorithm (AGA) and a Parallel Simulated Annealing algorithm (PSA) especially large-size instances.

Hassanzadeh et al. (2016) addressed bi-objective production distribution flow shop scheduling problem. The first objective includes minimization function of makespan and total weighted tardiness. The aim of second objective function is to minimize of sum of total weighted earliness, total weighted number of tardy jobs, inventory costs and total delivery costs. They proposed two new hybrid meta-heuristics (HCMOPSO and HBNSGA-II) methods to solve the problem. The computational experiments of the study showed the proposed methods outperforms than NSGA-II, MOPSO, NSGA-III and ACO. Yılmaz and Pardalos (2017) extended study of Pei et al. (2014) by including multiple customers. They examined the supply chain scheduling problem with multiple manufacturers and multiple customers in the two-stage that are namely production stage and transportation stage. They developed a mathematical optimization model to minimize the average lead time of the batches. A hybrid Artificial Bee Colony (ABC)

and Simulated Annealing (SA) algorithm, namely HABCSA was developed and compared with GA and ABC. Numerical results of study showed that the HABCSA algorithm outperforms the ABC and the GA in terms of the solution performance.

Joo and Kim (2017) considered the ISCS problem with unrelated parallel machines, batches, and heterogeneous delivery truck. In order to minimize the makespan of the all process, they developed MIP model. They conducted computational experiments to evaluate and compare the performance of the proposed meta-heuristics on two different problem size. Noroozi et al. (2017) proposed a new approach of coordination in an integrated supply chain (ISC): coordinating *order acceptance* (OA) and *batch delivery* (BD) due to *round trip transportation* (RTT) and using *third-party logistics* (3PLs) vehicles. They proposed a MIP model and solved their model with hybrid evolutionary computation algorithms based on particle swarm optimization (PSO) and genetic algorithm (GA).

Karaoğlu and Erhan (2017) studied ISCS problem with limited shelf life product which is produced at single facility. They developed MIP model to minimize the time needed to produce and deliver goods to all customers. They created a model with production rate, product lifespan and vehicle (or truck) capacity constraints. They assessed the performance of the B&C algorithm. Results of study showed that B&C algorithm yielded lower and upper bounds that are so near to each other. Devapriya et al. (2017) focused on integrated production and distribution problem with perishable product. To minimize the total transportation cost, i.e. the cost of delivery time and the number of vehicles necessary to ensure all the demands, they formulated MIP model. They solved small size problem with CPLEX. In order to find approximate solutions for larger problems, they proposed heuristics. Cheng et al. (2017) studied integrated scheduling problem considering production and distribution for manufacturers. The problem involves parallel batch-processing plants and fixed vehicle capacity. In order to minimize service span to respond quickly to the customer, they proposed an approximation algorithm to solve MIP model.

Kazemi et al. (2017) investigated two-stage flow shop scheduling problem with multiple assembly machines. They presented a hybrid imperialist competitive algorithm (HICA). They proposed MINLP to minimize the sum of tardiness plus delivery costs.

Results of study demonstrated that HICA outperformed ICA, but had ICA less runtime than HICA. Marandi (2017) focused on a variation of ISCS problem that contains a short shelf-life product and flow shop scheduling decisions in a single plant. In order to minimize makespan and number of vehicles required to complete the distribution of the products to satisfy customer demand, integer linear programming (ILP) model was constructed. As the proposed model is NP-hard, a new graph-based heuristic method was proposed to efficiently solve the problem. Noroozi et al. (2018) proposed integrated production-distribution scheduling problem with order acceptance, batch direct delivery, and third-party logistics optimization. In order to maximize benefit, they developed two MIP models and solved them with adaptive search approach (ASA) and adaptive genetic algorithm (GA).

Gharaei and Jolai (2018) studied a parallel multi-machine scheduling problem with interfering jobs which belong to two sets of different agents. The first objective function aims to minimize total weighted tardiness of jobs and the second objective function aims to minimize total distribution cost simultaneously. Firstly, a MIP model is formulated for this problem. After that, multi-objective evolutionary algorithm based on decomposition are developed in order to solve the problem. The performance of algorithm has been compared with three common algorithms in the literature. The computational experiments of the study demonstrated that the developed algorithm outperforms than the other tested algorithms.

Rostami et al. (2018) investigated ISCS problem in which both machine deterioration and learning effects were consequently addressed. Single machine and capacitated vehicles were considered. They aimed to minimize the sum of weighted completion time and total delivery time. Results of study demonstrated that the suggested heuristic method had high efficiency to obtain the optimal solution and outperformed to solve large sizes of the problems at a short time. Rafiei et al. (2018) integrated production-distribution and transportation planning problem based on a four-echelon supply chain. Bi-objective MIP model was formulated followed by linearization of the nonlinear models. The first objective function aims to minimize total cost of supply chain and the second objective function aims to maximize service level. Adopted Elastic Constraint Method (ECM) method used to solve multi objective models.

Beheshtinia et al. (2018) studied integration of production scheduling and vehicle routing in multi-site manufacturing supply chain and multi-suppliers. They introduced Reference Group Genetic Algorithm (RGGA) to solve the ISCS problem. They tested the performance of the algorithm on real data taken from a drug manufacturer in Iran. The computational experiments demonstrated that the RGGA gave better results comparing the obtained results from the real case.

Literature survey showed that the majority of the previous works were about formulating and solving supply chain scheduling problems. These efforts were mainly focused on single machine and parallel machine configurations. Only a few authors considered other general and complex configuration including flow shop and job shop scheduling.

In this thesis, flexible job shop scheduling (F-JSS) model was integrated in manufacturing part of supply chain network. Moreover, a new multi-objective model was proposed and tested using different multi-objective optimization solution techniques. Therefore, different solution sets were created with such techniques for the model.

CHAPTER 3

PROPOSED MATHEMATICAL MODEL

The objective of this chapter is to describe the mathematical model about integrated supply chain scheduling problem in flexible job shop environment as shown in Figure 3.1. In this model, the flow of products is assumed in two stage. The first stage is delivering of products from facility to possible warehouses. In the second stage, products are transported from warehouses to possible customers. Facility (manufacturing plant) consists of multiple machines and each machine can operate a number of operations to produce the product in a flexible job-shop scheduling environment. Products have different number of ordered operations which are performed to complete final product.

Warehouses are responsible for the distribution of products to the customers and/or storing some of them for the next periods. A deterministic, discrete demand volume for every product is given by customers for pre-defined periods. Moreover, shortage in the form of backorder can occur in each period.

The presented problem includes deciding on sequencing operations, sizing lots and assigning machines to operations to simultaneously optimize multi-objective functions. There are two conflicting objectives of developed model; the first objective function is time and the other is cost type. All parameters and used decision variables were presented in the following sections. A mixed-integer linear programming (MILP) formulation of the studied problem was provided. Accordingly;

- The first objective function is minimizing *the total cost* of the supply chain system which includes;
 - Production cost of all products in facility.
 - The backorder cost, in case of customer demands in each period are not met.
 - The holding cost of inventory in manufacturing facility and each warehouse.

- The second objective function is minimizing *the total makespan* ($\sum C_{max}$) by scheduling each operation on the machines and taking the precedence constraints into account.

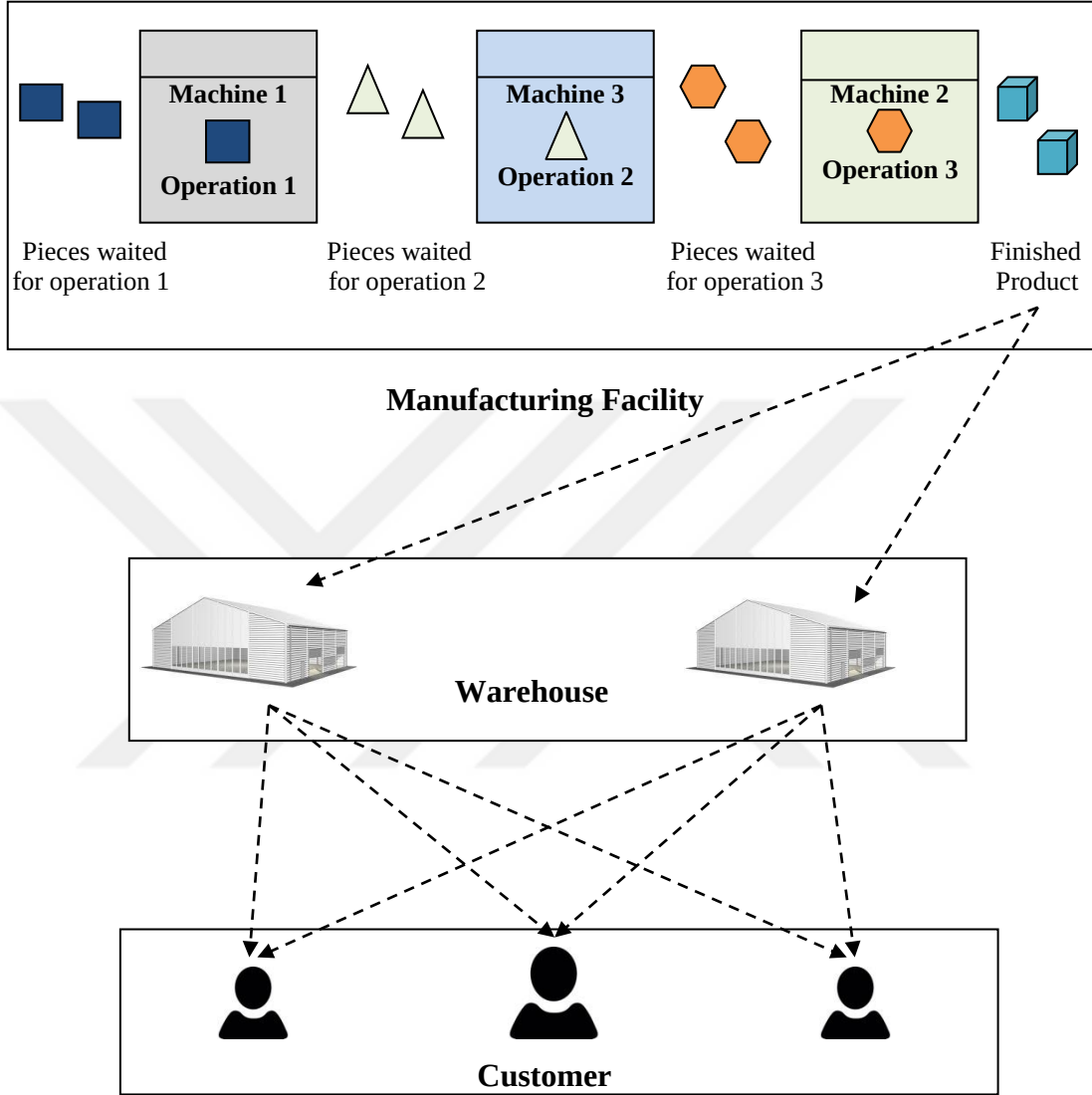


Figure 3.1. Integrated Supply Chain Scheduling in Flexible Job Shop Environment

The problem description was provided and detailed formulations related to the model were explained part by part. This chapter was organized as follows. Assumptions of the proposed model were presented in Section 3.1. In Section 3.2, sets, indices, parameters and decision variables used in the model were shown. In Section 3.3, a MINLP model was formed. Finally, linearization of proposed model was explained in Section 3.4.

3.1. Model Assumptions

The following assumptions are considered in the construction of the mathematical model:

1. There are multiple products.
2. All the parameters used in the model are deterministic.
3. Shortages are allowed.
4. Planning horizon is finite with T periods.
5. Products volumes are different.
6. There is storage limitation for manufacturing plant and warehouses.
7. Preemption of operations is not allowed.
8. A machine can perform different operations.
9. Each operation can be executed in several alternative machines.
10. All machines are continuously available from time zero.
11. Capacity of each machine is limited in each period.
12. Transportation time of products between machines are ignored.
13. All products are available at time zero.
14. Operating a product consumes machine capacity, which is limited.
15. There is no machine failure; therefore, machines are always available.
16. A backorder cost is occurred if customers' demands are not met.

3.2. Indices, Parameters and Variables

In this section, the notations including indices and sets, parameters, and decision variables used in the model were introduced as follows:

3.2.1. Indices and sets

m	Index used for a machine	$m = 1, \dots, M$
w	Index used for a warehouse	$w = 1, \dots, W$
p, p'	Index for a product	$p = 1, \dots, P$
c	Index for a customer	$c = 1, \dots, C$
t	Index used for a time period,	$t = 1, \dots, T$
h, h'	Index of operations	$h = 1, \dots, n(p) / n(p')$

3.2.2. Parameters

$n(p)$	Number of operations of product p
O_h^p	Operation h of product p
pc_{hmt}^p	Unit production cost of operation O_h^p on machine m in period t .
d_{ct}^p	Demand for product p by customer c in period t .
hc_t^p	Unit inventory holding cost of final product p in manufacturing facility in period t .
$hcost_{wt}^p$	Unit inventory holding cost for finished product p in warehouse w in period t .
sc_{ct}^p	Backorder cost per unit of the product p by supplying the demand of customer c in period t .
v_h^p	Volume of one unit of product p which has completed h_{th} operation.
ss_t	Total storage space capacity of manufacturing facility available in a period t .
$sspace_{wt}$	Total storage space capacity of warehouse w for storing products in time period t .
pt_{mht}^p	Processing time of operation O_h^p on machine m in period t .
mpt_{mt}	Available capacity of machine m in period t .
a_{mht}^p	$\begin{cases} 1, & \text{If operation } O_h^p \text{ can be operated on machine } m \text{ in period } t. \\ 0, & \text{otherwise} \end{cases}$
$bigM$	An arbitrary large positive number.

3.2.3. Decision variables

Q_{hmt}^p	Total quantity of the operation O_h^p operated on machine m in period t .
U_{mwt}^p	Quantity of product p which has completed final operation on machine m transferred from manufacturing facility to warehouse w in period t .

Y_{wct}^p	Quantity of product p which is transferred to customer c from warehouse w in period t.
I_t^p	Inventory of the of product p (which has completed final operation) in manufacturing facility in period t.
Inv_{wt}^p	Inventory of product p (which has completed final operation) in warehouse w in period t.
S_{ct}^p	Shortage quantity of product p of customer c in period t.
ST_{mht}^p	Start time of operation O_h^p on machine m in period t.
FT_{mht}^p	Finish time of operation O_h^p on machine m in period t.
$C_{max}(t)$	Makespan at period t.

- **Binary Variables**

$$x_{mht}^p = \begin{cases} 1, & \text{if machine } m \text{ is selected for operation } O_h^p \text{ in period } t. \\ 0, & \text{otherwise} \end{cases}$$

$$z_{mhh't}^{pp'} = \begin{cases} 1, & \text{if operation } O_h^p \text{ precedes operation } O_h^{p'} \text{ on machine } m \\ 0, & \text{otherwise} \end{cases}$$

3.3. Model Formulation

3.3.1. Objective functions

The presented model consists of two objectives: The first objective is minimizing total of cost in the supply chain. Total cost includes: (a) operation cost, (b) inventory holding cost of the products in manufacturing plant and warehouses, (c) the backorder cost of the product for customers. The second objective minimizes *total makespan in all periods*.

3.3.1.1. Total cost

Min Z_1 = Total Cost = Production cost + Inventory holding cost (*manufacturing facility and warehouse*) + Backorder cost

The components in the objective function can be calculated as following relations:

- **Production Cost**

$$\sum_{h=1}^{n(p)} \sum_{m=1}^M \sum_{t=1}^T \sum_{p=1}^P Q_{hmt}^p p c_{hmt}^p X_{mht}^p$$

- **Total Inventory holding cost (for manufacturing facility and warehouse)**

The total inventory holding cost during the planning period is the final product of the available stock quantity at the end of each period in manufacturing facility and warehouse.

$$\sum_{t=1}^T \sum_{p=1}^P I_t^p h c_t^p + \sum_{w=1}^W \sum_{t=1}^T \sum_{p=1}^P Inv_{wt}^p h cost_{wt}^p$$

- **Backorder cost**

The finished products are shipped to warehouses, and then delivered to customers. In case customers' demands are not met, there may be shortage in terms of backorder for each period.

$$\sum_{c=1}^C \sum_{t=1}^T \sum_{p=1}^P S_{ct}^p s c_{ct}^p$$

3.3.1.2. Total C_{max}

The second objective is to minimize the total makespan in all period.

$$\min Z_2 = \sum_{t=1}^T C_{max}(t)$$

3.3.2. Constraints

While considering the objectives, several constraints must be satisfied as the following:

- The constraint (1) makes certain that the total of work processing time on a machine m in period t should not exceed the machine capacity in that period.

$$\sum_{p=1}^P \sum_{h=1}^{n(p)} Q_{hmt}^p * X_{mht}^p * pt_{mht}^p \leq mpt_{mt}$$

$$m = 1, \dots, M; t = 1, \dots, T \quad (1)$$

- If operation O_h^p is not assigned to machine m in period t , the constraint (2) set the quantity of production on machine m equal to zero.

$$Q_{hmt}^p \leq bigM * X_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (2)$$

- The constraint in Equation (3) controls the quantity of operation O_h^p in each period.

$$\sum_{m=1}^M Q_{h-1mt}^p * X_{mh-1t}^p = \sum_{m=1}^M Q_{hmt}^p * X_{mht}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 2, \dots, n(p) \quad (3)$$

- Equation (4) make sure that operation O_h^p is assigned to only one machine in period t . As there is a possibility of an operation can be processed on alternative machines, a machine must be selected and the operation must be executed by the chosen machine.

$$\sum_{m=1}^M X_{mht}^p = 1$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 1, \dots, n(p) \quad (4)$$

- The constraint in Equation (5) is balance equation for the last operation of each product (finished product). It controls the inventory of product p in the end of period to

meet the demand.

$$I_t^p - I_{t-1}^p = \sum_{m=1}^M Q_{n(p)mt}^p * X_{mn(p)t}^p - \sum_{m=1}^M \sum_{w=1}^W U_{mwt}^p$$

$$t = 1, \dots, T; p = 1, \dots, P \quad (5)$$

- The constraint in Equation (6) is balance equation for the inventory for product p at warehouses.

$$Inv_{wt}^p - Inv_{wt-1}^p = \sum_{m=1}^M U_{mwt}^p - \sum_{c=1}^C Y_{wct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; w = 1, \dots, W \quad (6)$$

- Constraint (7) specifies that the total volume of inventory cannot exceed the total storage capacity of each warehouse in a period.

$$\sum_{p=1}^P Inv_{wt}^p * v_{n(p)}^p \leq sspace_{wt}$$

$$t = 1, \dots, T; w = 1, \dots, W \quad (7)$$

- Constraint (8) specifies that the total volume of inventory cannot exceed the total storage capacity of manufacturing plant in a period.

$$\sum_{p=1}^P I_t^p * v_{n(p)}^p \leq ss_t$$

$$t = 1, \dots, T \quad (8)$$

- Constraint (9) defines that the amount of each product shipped from warehouses to each customer in a time period cannot exceed demand of the customers.

$$\sum_{w=1}^W Y_{wct}^p \leq d_{ct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; c = 1, \dots, C \quad (9)$$

-The constraint in Equation (10) is balance equation for backorder level of the product p for customer c in a period.

$$S_{ct}^p - S_{ct-1}^p = d_{ct}^p - \sum_{w=1}^W Y_{wct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; c = 1, \dots, C \quad (10)$$

-The constraint in Equation (11) indicates that in the last period all the customers' demands are met and as a result, the customers' shortage in the last period will be zero.

$$S_{cT}^p = 0$$

$$p = 1, \dots, P; c = 1, \dots, C \quad (11)$$

-The constraint (12) enforces each product to follow a specified operation sequence.

$$FT_{mht}^p \geq ST_{mht}^p + pt_{mht}^p - \text{Big}M * (1 - X_{mht}^p)$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (12)$$

-The constraint (13) represents the precedence relationship among various operations of a product. In other words, the operation O_h^p is not started before the operation O_{h-1}^p has been completed.

$$\sum_{m=1}^M ST_{mh+1t}^p \geq \sum_{m=1}^M FT_{mht}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 1, \dots, n(p) - 1 \quad (13)$$

If operation O_h^p is not assigned to machine m , Constraint (14) sets the start and finish times of it on machine m equal to zero.

$$FT_{mht}^p + ST_{mht}^p \leq \text{Big}M * X_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (14)$$

- Constraint (15) defines the makespan time C_{max} .

$$C_{max_t} \geq \sum_{m=1}^M FT_{mn(p)t}^p$$

$$t = 1, \dots, T; p = 1, \dots, P \quad (15)$$

-Constraints (16) and (17) take care of the requirement that operation O_h^p and operation $O_{h'}^{p'}$ cannot be done at the same time on any machine.

$$ST_{mht}^p \geq FT_{mh't}^{p'} - bigM * (1 - X_{mht}^p) - bigM * (1 - X_{mh't}^{p'}) - bigM * (1 - Z_{mhh't}^{pp'})$$

$$m = 1, \dots, M; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p'); t = 1, \dots, T$$

$$O_h^p \neq O_{h'}^{p'} \quad (16)$$

$$ST_{mh't}^{p'} \geq FT_{mht}^p - bigM * (1 - X_{mht}^p) - bigM * (1 - X_{mh't}^{p'}) - bigM * (1 - Z_{mhh't}^{pp'})$$

$$m = 1, \dots, M; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p'); t = 1, \dots, T$$

$$O_h^p \neq O_{h'}^{p'} \quad (17)$$

Constraint 18 enforces that the machine for each O_h^p must be selected from the machine's alternatives of O_h^p .

$$X_{mht}^p \leq a_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (18)$$

Constraint 19 determines whether the disjunctive arcs of each possible operation pair $(O_h^p, O_{h'}^{p'})$ exist.

$$Z_{mh'ht}^{pp'} + Z_{mhh't}^{pp'} = X_{mht}^p * X_{mh't}^{p'}$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p')$$

$$O_h^p \neq O_{h'}^{p'} \quad (19)$$

Constraints (20), (21), and (22) indicate that the backorder and inventory levels of the products are zero at (t=0) time period zero.

$$S_{c0}^p = 0$$

$$p = 1, \dots, P; c = 1, \dots, C \quad (20)$$

$$I_0^p = 0$$

$$p = 1, \dots, P; h = 1, \dots, n(p) \quad (21)$$

$$Inv_{w0}^p = 0;$$

$$w = 1, \dots, W; p = 1, \dots, P \quad (22)$$

Constraints (23) and (24) represent the used decision variables in the model and their possible values.

$$X_{mh't}^{p'}, X_{mht}^p, Z_{mh'ht}^{p'p}, Z_{mhh't}^{pp'} = \{0, 1\}$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p') \quad (23)$$

$$Q_{hmt}^p, U_{mwt}^p, Y_{wct}^p, I_t^p, Inv_{wt}^p, S_{ct}^p, ST_{mht}^p, FT_{mht}^p, C_{\max_t}$$

$$m = 1, \dots, M; w = 1, \dots, W; p = 1, \dots, P; c = 1, \dots, C; h = 1, \dots, n(p); = 1, \dots, T \quad (24)$$

3.4. Linearization of Proposed Model

The proposed model includes many quadratic terms in both the objective and the constraints that greatly increase the computational complexity for solving the problem. To reduce the computational complexity, the model should be linearized. Thus, in order to analyze model more easily, exact linearization techniques were applied to transform it into linear forms by introducing additional variables and constraints. The linearization of these terms is discussed below.

3.4.1. Linearization the product of a binary and a continuous variable

Since the objective functions and constraints expressed by is non-linear ($Q_{hmt}^p * X_{mht}^p$), the following new continuous variable QX_{mht}^p was introduced to linearize model.

$$QX_{mht}^p = Q_{hmt}^p * X_{mht}^p$$

where Q_{hmt}^p is continuous variable and X_{mht}^p is binary variable. Moreover, additional constraints were formulated as below:

$$QX_{mht}^p \leq bigM * X_{mht}^p$$

$$QX_{mht}^p \leq Q_{hmt}^p$$

$$QX_{mht}^p \geq Q_{hmt}^p - bigM * (1 - X_{mht}^p)$$

3.4.2. Linearization the product of two binary variables

Since the constraint (19) expressed by is non-linear ($X_{mh't}^{p'} * X_{mht}^p$), the following new binary variable $XX_{mhh't}^{pp'}$ is introduced to linearize model.

$$XX_{mhh't}^{pp'} = X_{mht}^p * X_{mh't}^{p'}$$

where $X_{mh't}^{p'}$ and X_{mht}^p are binary variables. Moreover, additional constraints were formulated as below:

$$XX_{mhh't}^{pp'} \leq X_{mht}^p$$

$$XX_{mhh't}^{pp'} \leq X_{mh't}^{p'}$$

$$XX_{mhh't}^{pp'} \geq X_{mht}^p + X_{mh't}^{p'} - 1$$

Table 3.1 shows each possible operation pair ($O_h^p, O_{h'}^{p'}$) exist in model. If both operations are operated on the same machine m , then $X_{mht}^p * X_{mh't}^{p'} = 1$ and one of two 0–1

variables, $Z_{mh't}^{p'p}$ and $Z_{mhh't}^{pp'}$, should be equal 1. Otherwise $X_{mht}^p * X_{mh't}^{p'} = 0$, which means that there is no relationship between operations O_h^p and $O_{h'}^{p'}$, thus both $Z_{mh't}^{p'p}$ and $Z_{mhh't}^{pp'}$ equal 0.

Table 3.1. The Conditions between X_{mht}^p , $X_{mh't}^{p'}$ and $(X_{mht}^p * X_{mh't}^{p'})$

X_{mht}^p	$X_{mh't}^{p'}$	$X_{mht}^p * X_{mh't}^{p'}$
1	1	1
1	0	0
0	1	0
0	0	0

Thus, the following mixed-integer linear programming model is derived.

$$\begin{aligned}
 \text{Minimize } Z_1 = & \sum_{h=1}^{n(p)} \sum_{m=1}^M \sum_{t=1}^T \sum_{p=1}^P QX_{hmt}^p p c_{hmt}^p + \sum_{t=1}^T \sum_{p=1}^P I_t^p h c_t^p \\
 & + \sum_{w=1}^W \sum_{t=1}^T \sum_{p=1}^P Inv_{n(p)wt}^p h cost_{n(p)wt}^p + \sum_{c=1}^C \sum_{t=1}^T \sum_{p=1}^P S_{ct}^p s c_{ct}^p \\
 \text{Minimize } Z_2 = & \sum_{t=1}^T C_{max}(t)
 \end{aligned}$$

Subject to;

$$\sum_{p=1}^P \sum_{h=1}^{n(p)} QX_{hmt}^p * p t_{mht}^p \leq m p t_{mt}$$

$$m = 1, \dots, M; t = 1, \dots, T \quad (1)$$

$$Q_{hmt}^p \leq bigM * X_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (2)$$

$$\sum_{m=1}^M QX_{h-1mt}^p = \sum_{m=1}^M QX_{hmt}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 2, \dots, n(p) \quad (3)$$

$$\sum_{m=1}^M X_{mht}^p = 1$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 1, \dots, n(p) \quad (4)$$

$$I_t^p - I_{t-1}^p = \sum_{m=1}^M QX_{n(p)mt}^p - \sum_{m=1}^M \sum_{w=1}^W U_{mwt}^p$$

$$t = 1, \dots, T; p = 1, \dots, P \quad (5)$$

$$Inv_{wt}^p - Inv_{wt-1}^p = \sum_{m=1}^M U_{mwt}^p - \sum_{c=1}^C Y_{wct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; w = 1, \dots, W \quad (6)$$

$$\sum_{p=1}^P Inv_{wt}^p * v_{n(p)}^p \leq sspace_{wt}$$

$$t = 1, \dots, T; w = 1, \dots, W \quad (7)$$

$$\sum_{p=1}^P I_t^p * v_{n(p)}^p \leq ss_t$$

$$t = 1, \dots, T \quad (8)$$

$$\sum_{w=1}^W Y_{wct}^p \leq d_{ct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; c = 1, \dots, C \quad (9)$$

$$S_{ct}^p - S_{ct-1}^p = d_{ct}^p - \sum_{w=1}^W Y_{wct}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; c = 1, \dots, C \quad (10)$$

$$S_{cT}^p = 0$$

$$p = 1, \dots, P; c = 1, \dots, C \quad (11)$$

$$FT_{mht}^p \geq ST_{mht}^p + pt_{mht}^p - \text{Big}M * (1 - X_{mht}^p)$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (12)$$

$$\sum_{m=1}^M ST_{mh+1t}^p \geq \sum_{m=1}^M FT_{mht}^p$$

$$t = 1, \dots, T; p = 1, \dots, P; h = 1, \dots, n(p) - 1 \quad (13)$$

$$FT_{mht}^p + ST_{mht}^p \leq \text{Big}M * X_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (14)$$

$$C_{\max_t} \geq \sum_{m=1}^M FT_{mn(p)t}^p$$

$$t = 1, \dots, T; p = 1, \dots, P \quad (15)$$

$$ST_{mht}^p \geq FT_{mh't}^{p'} - \text{big}M * (1 - X_{mht}^p) - \text{big}M * (1 - X_{mh't}^{p'}) - \text{big}M * (1 - Z_{mh'ht}^{p'p})$$

$$m = 1, \dots, M; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p'); t = 1, \dots, T$$

$$O_h^p \neq O_{h'}^{p'} \quad (16)$$

$$ST_{mh't}^{p'} \geq FT_{mht}^p - bigM * (1 - X_{mht}^p) - bigM * (1 - X_{mh't}^{p'}) - bigM * (1 - Z_{mhh't}^{pp'})$$

$$m = 1, \dots, M; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p'); t = 1, \dots, T$$

$$O_h^p \neq O_{h'}^{p'} \quad (17)$$

$$X_{mht}^p \leq a_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (18)$$

$$Z_{mh'ht}^{p'p} + Z_{mhh't}^{pp'} = XX_{mhh't}^{pp'}$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p')$$

$$O_h^p \neq O_{h'}^{p'} \quad (19)$$

$$S_{c0}^p = 0$$

$$p = 1, \dots, P; c = 1, \dots, C \quad (20)$$

$$I_0^p = 0$$

$$p = 1, \dots, P; h = 1, \dots, n(p) \quad (21)$$

$$Inv_{w0}^p = 0;$$

$$w = 1, \dots, W; p = 1, \dots, P \quad (22)$$

$$X_{mh't}^{p'}, X_{mht}^p, Z_{mh'ht}^{p'p}, Z_{mhh't}^{pp'}, XX_{mhh't}^{pp'} = \{0, 1\}$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p') \quad (23)$$

$$QX_{hmt}^p, Q_{hmt}^p, U_{mwt}^p, Y_{wct}^p, I_t^p, Inv_{wt}^p, S_{ct}^p, ST_{mht}^p, FT_{mht}^p, C_{\max_t}$$

$$m = 1, \dots, M; w = 1, \dots, W; p = 1, \dots, P; c = 1, \dots, C; h = 1, \dots, n(p); t = 1, \dots, T \quad (24)$$

- **Additional Constraints:**

$$QX_{mht}^p \leq bigM * X_{mht}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (25)$$

$$QX_{mht}^p \leq Q_{hmt}^p$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (26)$$

$$QX_{mht}^p \geq Q_{hmt}^p - bigM * (1 - X_{mht}^p)$$

$$m = 1, \dots, M; p = 1, \dots, P; h = 1, \dots, n(p); t = 1, \dots, T \quad (27)$$

$$XX_{mhh't}^{pp'} \leq X_{mht}^p$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p')$$

$$O_h^p \neq O_{h'}^{p'} \quad (28)$$

$$XX_{mhh't}^{pp'} \leq X_{mh't}^{p'}$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p')$$

$$O_h^p \neq O_{h'}^{p'} \quad (29)$$

$$XX_{mhh't}^{pp'} \geq X_{mht}^p + X_{mh't}^{p'} - 1$$

$$m = 1, \dots, M; t = 1, \dots, T; p, p' = 1, \dots, P; h = 1, \dots, n(p); h' = 1, \dots, n(p')$$

$$O_h^p \neq O_{h'}^{p'} \quad (30)$$

CHAPTER 4

MATERIAL AND METHODS

In this thesis, to deal with the proposed bi-objective supply chain scheduling problem, various multi objective optimization problems were applied. In the literature two main solution approaches are usually applied to solve MOOPs:

- (1) Multi-objective decision making (MODM) methods,
- (2) Multi-objective optimization techniques.

MODM techniques have an important role in solving MOOPs, which include more than one objective function. In these techniques priority or weight factors are used based-on information obtained from the decision makers and provide the decision maker with solutions. (Hwang and Masud, 2012; Pasandideh et al., 2015; Khalilpourazari and Khamseh, 2017).

Multi-objective optimization methods give a set of feasible solutions in the wider sense. These methods focus on a number of candidate solutions and offer various options to the decision maker. These sets are often too large, it can become hard to select an optimal solution.

In this thesis, various MODM methods were applied to solve proposed bi-objective mathematical model: *weighted sum*, *min-max*, *goal attainment*, *LP-metric/global criterion* and *goal programming*. Moreover, a Pareto-based meta-heuristic algorithm called *Non-dominated Sorting Genetic Algorithm-II (NSGA-II)* were applied. Thus, different solutions were obtained for the problem, since, the decision makers might have different preferences depending on the priorities of each objective function. Detailed descriptions of the used solution methods in this thesis will be explained in the next subsections.

4.1. Multi-Objective Optimization

In the single-objective optimization problems only a single optimal solution is provided. However, real world problems are more complex and generally there exist more than one or more objective for optimization. Actually, problems encountered in engineering involve complex assignments which require simultaneous optimization of various competing objectives. The multi-objective optimization approach is a widespread applied method to analyze and solve these types of difficult problems (Fallah and Honarparast, 2013).

The answer of MOOPs are a set of points known as *Pareto optimal set* that propose the trade-off solutions among conflicting objectives. Pareto-optimal can be defined as a decision/objective vector. The solution is valid if there is not another decision/objective vector that dominates solution vector. A feasible point can be defined as a solution to a multi-objective optimization problem, and is called Pareto optimal. At this point, there exists no other feasible point which improves one of the objectives without deteriorating at least one of the other objectives. However, in multi-objective optimization problem, usually there is not a single optimal solution but a set of optimal points. The set of these mathematically equivalent point is often referred to as the Pareto set or Pareto front (Niu et al. 2013; Donoso, 2016; Vo-Duy et al. 2017) .

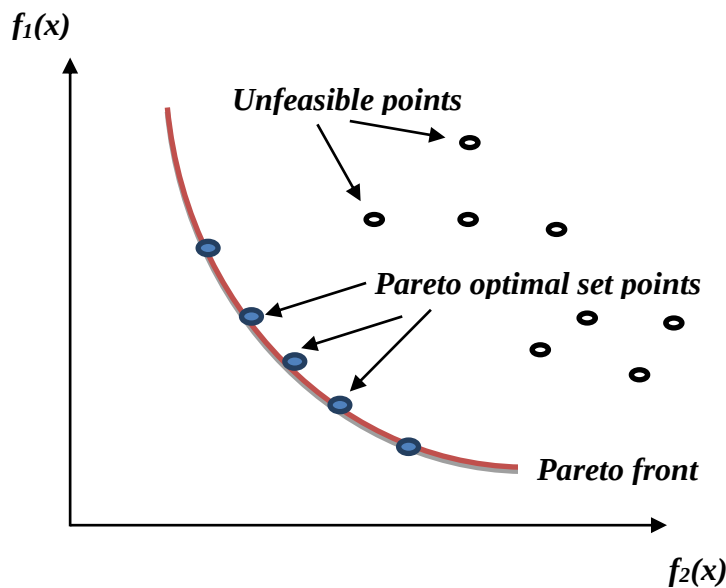


Figure 4.1. Pareto Points for a Bi-Objective Optimization Problem

The Figure 4.1 depicts a general Pareto concept for a bi-objective optimization problem. Pareto optimal points are located on the Pareto front. The remaining points represent unfeasible solutions.

MOOPs solutions aim to achieve to goals such as; (1) to maintain non-dominated points in the objective and decision space (2) to perform an algorithmic progress for the Pareto front (3) to supply a variety of points on the Pareto front and of Pareto optimal solutions and (4) to satisfy the decision maker by supplying enough and limited number of Pareto points to make a choice (Chiandussi et al. 2012).

The Pareto-optimal solution means that a target is best when other targets are stable. For this reason, it is impossible to achieve a better purpose than the best solution without destroying any purpose. The multi-objective optimization problem can be explained briefly as below:

Searching the design vector $X^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ which will meet the requirements for the m inequality constraints:

$$g_i(x) \geq 0, \quad i=1, 2, \dots, m \quad (4.1)$$

and the p equality constraints:

$$h_i(x) = 0, \quad i=1, 2, \dots, p \quad (4.2)$$

and will optimize the objective vector function:

$$F(X) = [f_1(x), f_2(x), \dots, f_i(x)]^T \quad (4.3)$$

where $f_i(x)$ is the i_{th} objective function, Γ is feasible region in the design space, $X = [x_1, x_2, \dots, x_n]^T$ is the decision variable vector, m and p are the number of inequality and equality constraint functions, respectively.

It can be said that among the set of all values satisfying Eq. (4.1), Eq. (4.2) and Eq. (4.3), the particular set $x_1^*, x_2^*, \dots, x_n^*$ that yields in the optimal values of all the objective functions, is named as Pareto-optimal solutions. Additionally, point X^* which belongs to feasible region in the design space (Γ) is known as Pareto optimal. For every $X^* \in \Gamma$, the following expression satisfies (Fallah and Honarparast, 2013):

$$\forall i \in \{1, 2, \dots, n\}: f(x) = f(x^*) \wedge \exists i \in \{1, 2, \dots, n\}: f_i(x) > f_i(x^*) \quad (4.4)$$

Eq. (4.4) shows that there will be no more feasible vector X which would reduce some criterion without resulting in a simultaneous raise in at least one of the other criteria.

4.1.1. LP metric/global criterion method

In multi-objective optimization, the Global Criterion Method (GCM) is involved to a solution group of methods in which none of the given prior knowledge or information is provided. The main benefit of this method is that the solution can be achieved without decision and there is no need to designer. In GCM, the distance between reference points and the suitable goal points has been minimized. However, the researcher must determine the reference point and measure to evaluate distances. The objective of this method is to predict the optimal ideal solution provided by deducing the prediction between the objective function values in the multi-objective model.

In this method, a multi-criteria optimization problem is converted into a scalar optimization problem by the aid of GCM. The function representing GCM have to be identified and a suitable solution have to be obtained so that the ideal solution (f_i^0) is close enough (Coello, 2000). This function can be shown as Eq. (4.5):

$$f(x) = \sum_{i=1}^k \left[\frac{f_i^0 - f_1(x)}{f_i^0} \right] \quad (4.5)$$

GCM can also be evaluated using a group of *LP* metrics which can be defined as:

$$L_p(f) = \left[\sum_{i=1}^k |f_i^0 - f_1(x)|^s \right]^{\frac{1}{s}}, 1 \leq s \leq \infty \quad (4.6)$$

In Eq. (4.6), “ s ” parameter can have different values, such as 1 or 2. However, other values for s can also be used, but the solution obtained after minimizing differs greatly depending on the values of s chosen.

For example, for $s = 1$ and $s = 2$ in Eq. (4.6) will yield to Eq. (4.7) and Eq. (4.8);

$$L_1(f) = \sum_{i=1}^k |f_i^0 - f_1(x)| \quad (4.7)$$

$$L_2(f) = \left[\sum_{i=1}^k (f_i^0 - f_1(x))^2 \right]^{1/2} \quad (4.8)$$

It is important to notice that minimization of $L_2(f)$ is equivalent to the minimization of the distance between the value of the function and the ideal solution. However, it is recommended to apply relative distances instead of studying with distance in an absolute sense (Oliveira and Saramago, 2010).

Therefore, Eq. (4.6) can be rearranged to give Eq. (4.9) as following:

$$L_p(f) = \left[\sum_{i=1}^k \left| \frac{f_i^0 - f_1(x)}{f_i^0} \right|^s \right]^{\frac{1}{s}}, 1 \leq s \leq \infty \quad (4.9)$$

Thus, values for all normalized functions are limited between the interval [0,1].

4.1.2. Goal attainment

In Goal Attainment Method (GAM), decision maker must provide a goal vector z^{ref} . Also, a vector of weights must be provided by decision maker in the form of $w = [w_1, w_2, \dots, w_k]$ which establish a contacts desired goals with the relative under or over attainment.

In order to find the ideal solution x^* , the following problem is solved:

Minimize α

subject to;

$$z^{ref} + \alpha \cdot w_i \geq f_i(x); \quad i = 1, \dots, k, \quad (4.10)$$

$$x \in \Gamma$$

Here α is a weighted deviation which is independent in sign. Also, weights $(w_1, w_2, \dots, w_k)$ are normalized which yields to Eq.(4.11);

$$\sum_{i=1}^k |w_i| = 1 \quad (4.11)$$

The mechanism of the way of goal attainment method operates is depicted in Figure 4.2.

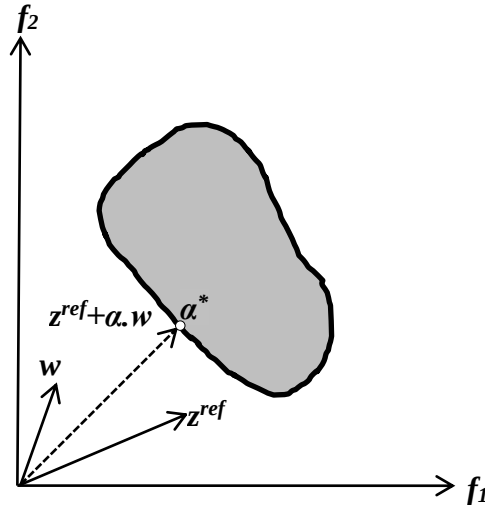


Figure 4.2. Goal Attainment Method for Optimization of a Bi-objective Problem

If some $w_i = 0$, ($i = 1, \dots, k$), it means that the maximum limit of objectives $f_i(x)$ is z^{ref} . It can be shown that every optimal Pareto solution can be formed by alternations in the weights, where $w_i \geq 0$ ($i = 1, \dots, k$). This situation is valid for non-convex problems, too (Coello et al., 2007).

The vector z^{ref} is presented by the decision maker goal, who also determines the direction of w . As depicted in Eq. (4.10), with given vectors w and z^{ref} , the direction of the vector $z^{ref} + \alpha \cdot w$ can be defined. Eq. (4.10) defines the searching for a reasonable point on this vector in objective space which is nearest to the origin. It is obvious that the optimal solution of Eq. (4.10) is the first point at which “ $z^{ref} + \alpha \cdot w$ ” intersects the feasible region in the objective space (Anghileri et al., 2005).

This point of intersection is critical. The existence of intersection means that Pareto optimal solution can be obtained. Decision maker can also decide whether the goals are attainable or not by the aid of optimum value of α . For instance, $\alpha < 0$ indicates that the goal of the decision maker is attainable and a better solution is then to be provided. Contrarily, for $\alpha > 0$ the goal is unapproachable (Coello, 2000).

4.1.3. Weighted sum method

Weighted Sum Method creates a single-objective model by weighing the i number of objective functions by giving a weight to each of them (Donoso, 2016). In this method, the multi-objective model can be presented as given in the below:

Optimize [minimize/maximize]

$$F'(X) = \sum_{i=1}^n w_i f_i(X) \quad (4.12)$$

subject to;

$$H(X) = 0 \quad (4.13)$$

$$G(X) \geq 0 \quad (4.14)$$

$$0 \leq w_i \leq 1, i = \{1, \dots, n\} \quad (4.15)$$

$$\sum_{i=1}^n w_i = 1 \quad (4.16)$$

As seen from the equations each of the functions are multiplied with their weight value (w_i) which must be in the range between 0 and 1. Also, it is obvious that the total of weights of the functions must equal to 1. The fitness function $F'(X)$ obtained from the sum is a linear combination of the functions $f_i(X)$.

The weighted sum method finds the points of the optimal Pareto front that includes all possible optimal solutions. For example, for two objective functions situation we would have the Eq. (4.17) as given in the following;

$$F'(X) = w_1 * f_1(X) + w_2 * f_2(X) \quad (4.17)$$

The main concept lying behind the weighted sum method is assigning values to w_1 and w_2 to locate two lines that tangentially in touch with the set of solutions (K). The crossing point shown as “ P ” of the two lines L_1 and L_2 can be attributed as optimal solution depending on the weights (w_1, w_2) which is associated with both the optimal Pareto front in the multi-objective system. When the set of solutions (K) is convex, optimal Pareto front may be found by altering the values of weight vector W (w_1 and w_2) (Figure 4.3a). Thus, another P point would be located with new weights which also belong to the optimal Pareto Front.

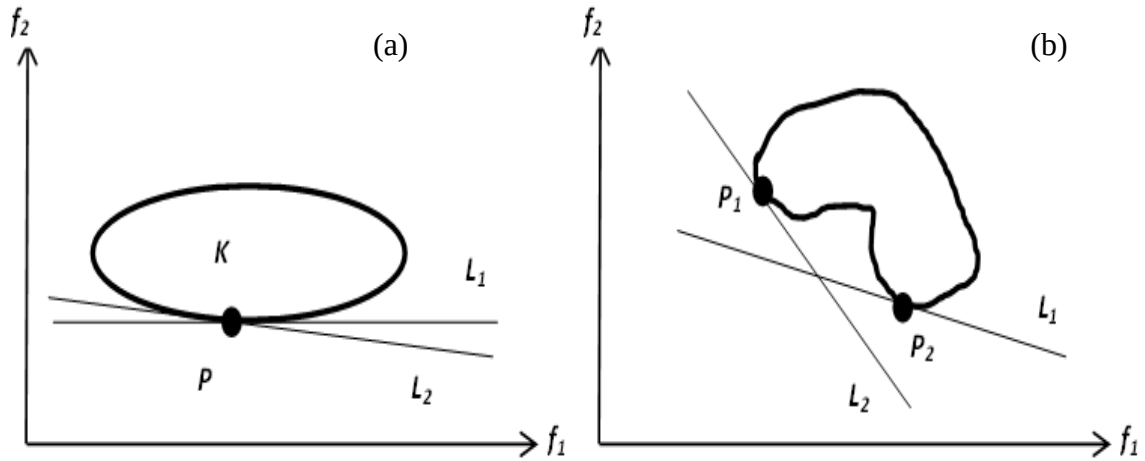


Figure 4.3. Solution with Weighted Sum Method for a) Convex b) Non-convex
Solution Set

Solution set K can also be non-convex as depicted in Figure (4.3b) where an optimal Pareto cannot be achieved by a linear combination. In a convex set of solutions, probability of being found would be $p > 0$ for all elements of the optimal Pareto front. However, in non-convex sets of solutions, this is “0” for just certain elements of the optimal Pareto front.

Another important issue for weighted sum method is the ranges of the functions which may have different magnitudes. The variation in ranges becomes critical when differences have high values due to the function having the higher values will suppress in the result. Thus, normalization of the values of all objective functions must be done (Donoso, 2016).

4.1.4. Lexicographic goal programming

In this solution method, several goals are classified into various levels of priorities. Additionally, in Lexicographic Goal Programming (LGP) goals with a lower-level priority is extremely more important than a goal of a higher-level priority. Therefore, it is critical to perform the goals of first level priority before paying attention to goals of second level of priority. It is suggested by some researchers that although this approach is some critics, it is the most practical way to assess these goals. A series of sequential goal programming problems are formulated and solved by this approach.

First, only the objectives of the first level priority and related constraints are considered and solved in the formulation of the problem of the goal programming problem. In case of presence of more than one solution to the problem, another goal programming problem with second-level priority goals have been formulated.

In this situation, the main goal is to obtain a minimum deviation between the targets of the second level priority. In addition, the objectives of the first priority level is used as a hard constraints, so that the resulting solution is not incompatible with the objectives of the first priority level. This process proceeds in turn with the goals of other high-level priorities. If one of the target programming problems results in a single solution, the process is terminated. When this occurs, all subsequent targets of higher priority level are known as meaningless and useless targets. However, as in other target programming methods, this method is also depends on the aim of users (Deb, 1999).

To formulate a generic lexical target program mathematically, the number of priority levels (L) is defined by the corresponding index $i = 1, \dots, L$. At present, all priorities are a function of a subset of the undesired deviation parameters; $h_1(n_p)$. This gives the formulation depicted as follows:

$$\text{Let Min } a = [h_1(\underline{n}, \underline{p}), h_2(\underline{n}, \underline{p}), \dots, h_L(\underline{n}, \underline{p})] \quad (4.18)$$

subject to;

$$f_q(\underline{x}) + n_q - p_q = b_q \quad (4.19)$$

$$\underline{x} \in \Gamma$$

$$n_q, p_q \geq 0 \quad q = 1, \dots, Q \quad (4.20)$$

Here, each $h_L(n,p)$ contains a number of undesired deviational variables. The exact nature of $h_L(n,p)$ based on the nature of the goal program to be created. However, the assumption of linear and separable structure will form the following equation;

$$h_l(\underline{n}, \underline{p}) = \sum_{q=1}^Q \left(\frac{u_q^l n_q}{k_q} + \frac{v_q^l p_q}{k_q} \right) \quad (4.21)$$

In Eq. (4.21), u_q^l is the preferred weight associated with the minimization of n_q in the l_{th} priority level. v_q^l is the preferential weight associated with the minimization of p_q in the l_{th} priority level.

The preferred weights are applied to model the priority of the minimization of the corresponding deviational variables. In the absence of a deviational variable in a particular importance level then its weight for that level is set equal to zero. Also, deviational variables that are evaluated as insignificant can be assigned as a preferential weight of zero. k_q is the normalization constant associated with the q_{th} goal. These constants are required in order to scale goals with the same measurement units (Jones et al., 2010).

4.1.5. Min-max method

The min-max solution method was proposed by Lightner and Director (1981). This method is applied to find a feasible solution which minimizes the its distance from the ideal solution as the representation of a decision maker's all preferences. The min-max method is capable to find effient solutions whether the problem is convex or non-convex (Augusto et al., 2012).

This method can be formulized algebraically as below:

$$Min (Max[w_i^* | f_i - f_i^0 |]) \quad (4.21)$$

subject to;

$$x \in \Gamma \quad (4.22)$$

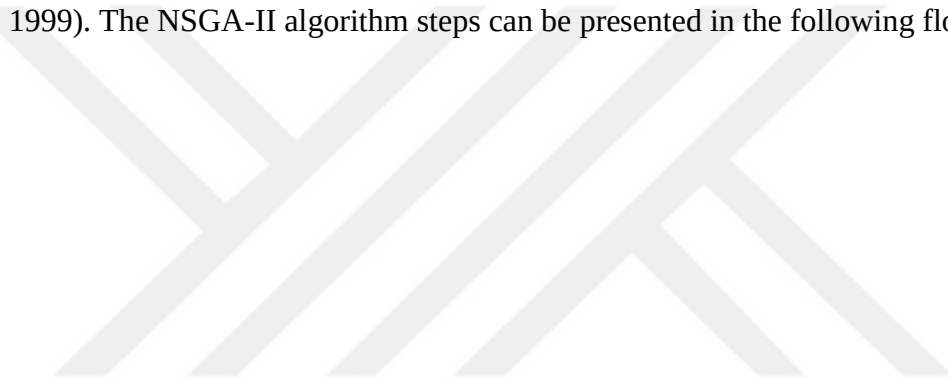
where, w_i^* is normalizing weight factor for the i_{th} objective function, f_i is the i_{th} objective function, f_i^0 is ideal solution obtained by the individual optimization.

4.1.6. Non-dominated sorting genetic algorithm-II (NSGA-II)

Genetic algorithms start with a series of randomly generated sequences (initial population) represented as chromosomes (or individuals) of team clusters of the same length. The gens in the chromosomes keep data and values through indexes and stimuli that represent a possible problem.

Genetic algorithms use a fitness function, assigning a set of values that each individual determines the closeness of each individual's optimal solution to evaluate each individual. To approach optimal resolution, several genetic operators are applied to the initial population: the selection selects the best ranked chromosomal fractions as the next generation parent from the current population and runs them on the crossing pairs. It provides a generation of offspring (child chromosomes) from the union of the parents' genetic information. The mutation genetically changes a single gene to improve the criteria to be optimized (Rodriguez et al., 2015).

NSGA-II developed by Deb et al. based on the Pareto approach, has been verified as an efficient algorithm to solve multipurpose optimization problems (Pareto-optimal set) (Deb, 1999). The NSGA-II algorithm steps can be presented in the following flow chart:



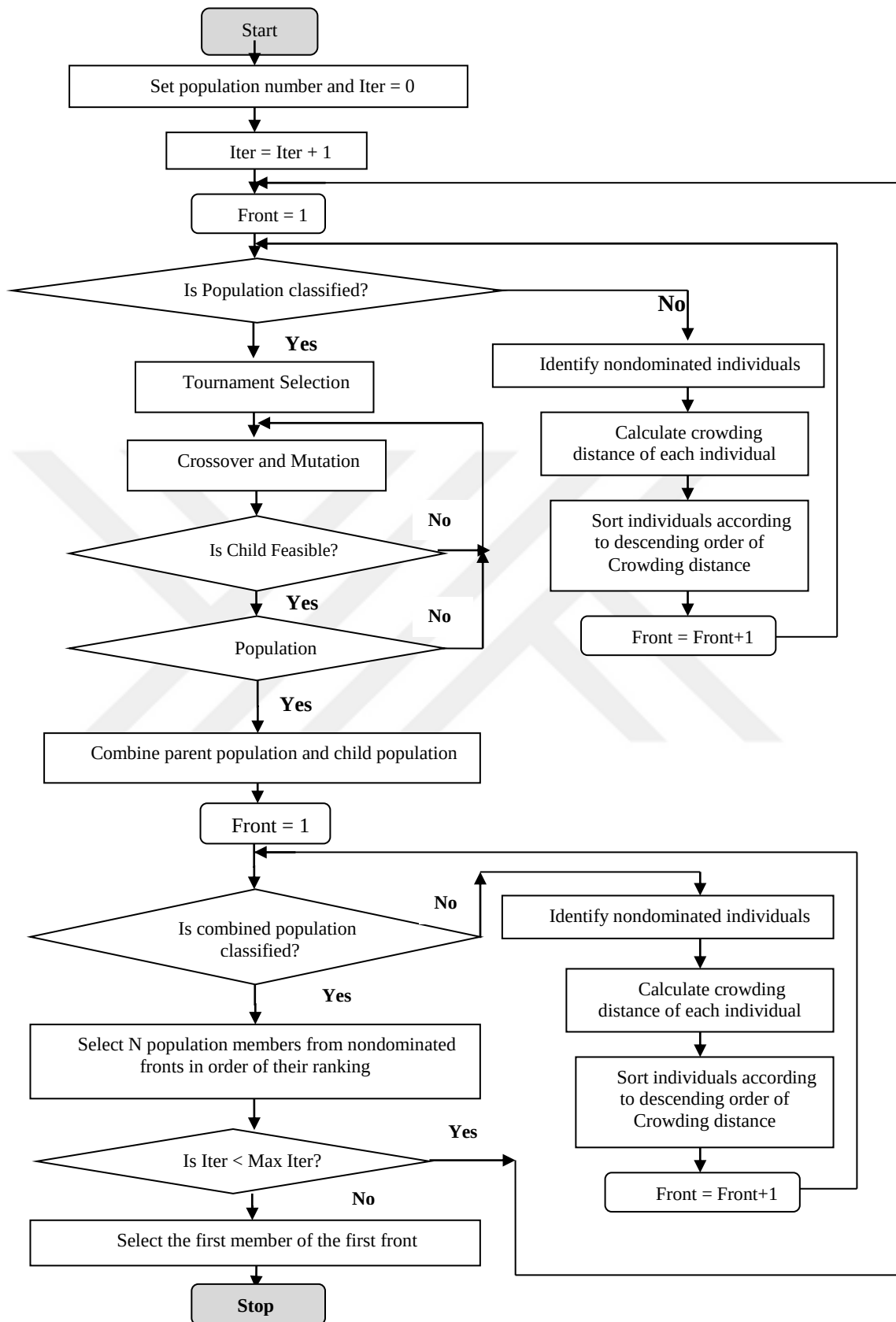


Figure 4.4. NSGA-II Algorithm Steps

Unlike classical mathematical programming methods, it has been proven that evolutionary algorithms are appropriate for multi-objective optimization problems due to their capability and capacity to obtain a solution set in a single study. In addition, Evolutionary Algorithms (EAs) are less sensitive than Pareto front, which consist of Pareto optimal solutions.

Goldberg (1989) proposed a multi-objective evolutionary algorithm using the concept of domination (Goldberg, 1989). This algorithm is an improved NSGA version generated from the well-known genetic algorithm and the non-dominant sequencing concept. Among the evolutionary algorithms, NSGA-II is a convenient method to meet the objectives of multi-objective optimization. NSGA-II uses the crowd comparison operator, which regulates both Pareto sovereignty and population density (Fallah and Honarparast, 2013; Vo-Duy et al., 2017).

NSGA-II is a powerful method in terms of efficiency and easy implementation which has attracted a great deal of interest from researchers. In recent years, the NSGA-II has been improved and widely used in design optimization of various problems and applications (Liu, Song and Wang, 2017). The general algorithm for NSGA-II can be outlined as below:

1. Produce an initial population P_0 in N dimensions.
2. The crowd comparison operator creates a child population Q_t using binary tournament selection based on transition and mutation in the main population (P_t), where "t" specifies the generation number. The child population and the main population then form the entire population R_t .
3. Apply a ranking approximation that does not dominate the R_t of the entire population in order to identify the different non-dominant facades of objective functions.
4. Form a new N parent ($P_t + 1$) using the acquired fronts (F_i).
5. Continue to the process till the maximum number of iterations is achieved (Vo-Duy et al., 2017).

Figure 4.5 represents the algorithm of Fast Non-Dominated Sorting in NSGA-II. The application of fast non-dominated sorting process for population P_t yields a list of the non-dominated fronts F_i .

fast-non-dominated-sort (P)

for each $p \in P$

$S_p = 0$

$n_p = 0$

for each $q \in P$

if $(p \prec q)$ then

If p dominates q

$S_p = S_p \cup \{q\}$

Add q to the set of solutions dominated by p

Else if $(q \prec p)$ then

$n_p = n_p + 1$

Increment the domination counter of p

if $n_p = 0$ then

p belongs to the first front

$p_{rank} = 1$

$F_1 = F_1 \cup \{p\}$

$i = 1$

Start the front counter

while $F_i \neq 0$

$Q = 0$

Used to store the members of the front

for each $p \in F_i$

for each $q \in S_p$

$n_q = n_q + 1$

if $n_q = 0$ then q belongs to the next front

$q_{rank} = i + 1$

$Q = Q \cup \{q\}$

$i = i + 1$

$F_i = Q$

Figure 4.5. Algorithm of Fast Non-Dominated Sorting in NSGA-II (Kaucic, 2008).

As shown in Figure 4.5, NSGA-II depends on the definition of the parameters required for the algorithm, such as population size, band number, probability of transition and the probability of mutation (Deb et al., 2002). For this reason, it is important to perform a parameter calibration for each problem examined.

CHAPTER 5

TESTING AND COMPUTATIONAL EXPERIMENTS

In this section, the mathematical model presented in Chapter 3 was tested for two reasons, first for its functionality to be proven, and second the outcomes and trade-offs between the conflicting objectives to be analyzed. Thus, computational experiments were performed to compare and evaluate the performances of the mentioned solution methods in Chapter 4.

Firstly, depending on the uniform distribution in pre-specified intervals, whole model parameters were arbitrarily created. Then, different sizes 14 test problems with different combinations were generated randomly. The generated problems were classified in terms of the number of product (p), number of machines (m), number of warehouse (w), number of customer (c), and number of periods (t), and number of operations of each product $O(p,h)$. For instance, the (2:1:1:2:2:3) test problem shows 2-machine, 1-warehouse, 1-customer, 2-product, 2-periods with max 3-operations, problem case.

Objective-function values (Z_1 and Z_2) and CPU time were obtained and compared for each test problem in order to choose the best in terms of total cost, total C_{max} , and their required CPU time at the same time. The proposed MILP model was solved in Generalized Algebraic Modeling System (GAMS) 25.0 software. Furthermore, NSGA-II solutions were obtained from MATLAB R2017a. The codes were executed on a personal computer with core (TM) i5-CPU 2.70 GHz, RAM 8.00 GB.

The classical flexible job shop problem has proven to be strongly NP-hard (Non-deterministic Polynomial-time hard). Therefore, integrated supply chain and scheduling problem in a flexible job shop environment is also NP-hard in strong sense. The main problem in practical usage is the size and complexity of the proposed model, even for relatively small sized problems. Because, if the problem size becomes large, the number of constraints and variables increases rapidly. In this problem, the computational time depends on number of customers, warehouses, time periods and especially numbers of

machines, products and operations. Since, solution methods fail to solve large-size instances within a reasonable timeframe, the run time has been limited up 3600 second for each problem.

This section was organized as follows. Dataset used in the test problems were presented in Section 5.1. In Section 5.2, numerical results about the solution of the small and medium size test problems were given. In Section 5.3, numerical results about the solution of the large size test problems were given. Finally, discussion and comments were presented about all test problems results.

5.1. Dataset

The dataset that was used in this thesis was randomly generated based on uniform distribution as shown Table 5.1.

Table 5.1. The Values of Parameters Based on Uniform Distributions

Parameters	Intervals	Parameters	Intervals
pc_{hmt}^p	U(1,6)	ss_t	U(50,400)
d_{ct}^p	U(1,20)	$sspace_{wt}$	U(500,1200)
hc_t^p	U(2,5)	pt_{mht}^p	U(0.1,0.9)
$hcost_{wt}^p$	U(4,9)	mpt_{mt}	U(104,936)
sc_{ct}^p	U(10,50)	v_h^p	U(0.8,1.5)

Big M: A large positive number

5.2. Small and Medium Sized Test Problems

Seven test problems with small and medium sized were generated. Since there is no benchmark dataset available in the literature for this problem (supply chain scheduling in flexible environment), it is necessary to generate random instances for the problem in order to evaluate and compare the performance of the solution methods. These test problems were defined in Table 5.2.

Table 5.2. The Generated Small and Medium Sized Problems Instances

Problem No	Problem Size (m:w:c:t:p:h _{max})	No. of Machine (m)	No. of Warehouses (w)	No. of Customers (c)	No. of Periods (t)	No. of Products (p)	No. of Operations O(p,h)
Test 1	(2:1:1:2:2:3)	2	1	1	2	2	p(1)=2,p(2)=3
Test 2	(2:3:4:4:2:2)	2	3	4	4	2	P(1)=2,p(2)=2
Test 3	(3:2:6:3:2:3)	3	2	6	3	2	P(1)=3,p2=(3)
Test 4	(4:1:6:3:2:5)	4	1	6	3	2	P(1)=3, p(2)=5
Test 5	(3:2:2:3:3:4)	3	2	2	3	3	P(1)=2,p(2)=4, P(3)=4
Test 6	(3:3:4:4:3:5)	3	3	4	4	3	P(1)=3,p(2)=3, P(3)=5
Test 7	(4:2:2:4:2:6)	4	2	2	4	2	P(1)=4, p(2)=6

5.2.1. Test Problem 1

Test Problem 1 (2:1:1:2:2:3) involves two machines in manufacturing facility, one potential warehouse, one customer, two periods, two products each of which with two and three operations, respectively.

5.2.1.1. Individual optimization results for test problem 1

Individual optimization is used to find optimal values of each single objective problem in multi-objective problems. It is assumed that the optimal solution for each objective solution is an effective solution. Briefly, at first the solution of the model is performed primarily by focusing one of the objectives every time to reach an efficient solution. If the effective solutions provided from the first objective problem satisfy the second or more objective problems constraints, then ideal overall solution is achieved (Pasandideh et al., 2015). Figure 5.1 shows objective function results obtained by individual optimization for test problem 1.

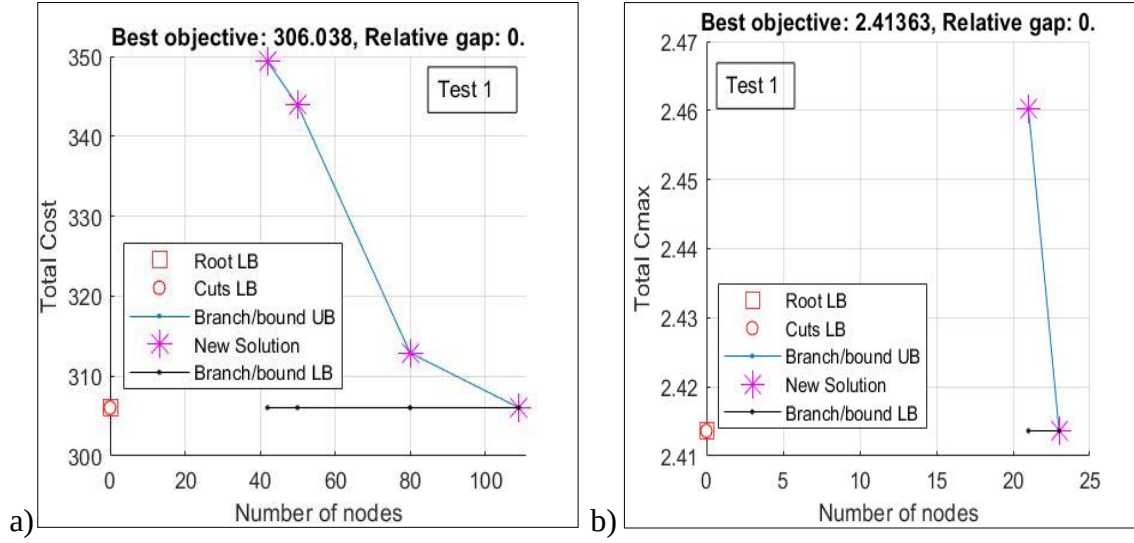


Figure 5.1. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 1

According to Figure 5.1, the objective function values $Z_1 = 306.038$ and $Z_2 = 2.414$ were obtained, respectively.

5.2.1.2. Normalization of objective functions

The objective functions with different units must be converted into dimensionless magnitudes in order to solve them simultaneously. Due to dimension inconsistencies, the sum of objective functions cannot be directly. Furthermore, fitness function is dominated by a single objective when the differences between the values of the objectives are large. Therefore, in all tests each objective function was normalized to achieve uniformity as follows:

$$\min \sum_{i=1}^n w_i \bar{f}_i(x) \quad (5.1)$$

where,

$$\bar{f}_i(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (5.2)$$

where w_i is the weight coefficients, $f_i^{\max}$ and $f_i^{\min}$ are maximum and minimum values of dimension i of each objective function.

The fitness function of model can then be expressed as follows:

$$F(x) = w1 * \frac{TCost(x) - TCost^{min}}{TCost^{max} - TCost^{min}} + w2 * \frac{TC_{max}(x) - TC_{max}^{min}}{TC_{max}^{max} - TC_{max}^{min}} \quad (5.3)$$

$$w1 + w2 = 1 \quad (5.4)$$

After Eq.(5.3) and Eq.(5.4) are solved, the range of each objective function will be between 0 and 1. The fitness function also stays between 0 and 1. According to results of *Test Problem 1*, $TCost^{max} = 371.898$ and $TCost^{min} = 306.038$, $TC_{max}^{max} = 2.933$ and $TC_{max}^{min} = 2.414$ were obtained, respectively. After the dimensionality removals and normalizations.

Eq. (5.5) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 306.038}{65.860} + w2 * \frac{TC_{max}(x) - 2.414}{0.519} \quad (5.5)$$

5.2.1.3. Weighted sum method results for test problem 1

In the Weighted Sum Method (WSM), all objectives are combined into a single objective by using a normalized weight vector. The Pareto solutions are obtained by resolving the problem using different weights. Considering that the result of using WSM varies with different weights, the aim of this experiment is to study the effect of applying different sets of weight values in the model and find the best set for solving the model. In this dissertation, four different weights (0.2, 0.8), (0.4, 0.6), (0.6, 0.8), and (0.8, 0.2) were used for combining objective functions. Results for each weight set applied to weighted sum method was given in Table 5.3.

Table 5.3. Weighted Sum Method with Different Weights for Test Problem 1

Problem Size	Weight		Average Total Cost (TCost)	Average Total C_{max} (TC_{max})	Average CPU Time (sec)
	w1	w2			
Test 1	0.8	0.2	314.546	2.658	1.202
	0.6	0.4	314.546	2.658	0.703
	0.4	0.6	334.013	2.536	0.857
	0.2	0.8	371.898	2.414	1.097

5.2.1.4. Filtering/displaced ideal solution method

Based on the results obtained from Table 5.3, it can be easily seen that none of the weight sets can optimize both objective functions simultaneously. For instance, weight set (0.2,0.8) performs best in optimizing TC_{max} (2.414) but, it has the worst performance in optimizing $TCost$ (371.898). Thus, the Filtering/Displaced Ideal Solution (DIS) method was applied to select the best solution set (Hwang and Masud, 1979).

In DIS method, F_i is a vector of the solution for each method and in this dissertation, it is obtained based on three criteria which are first objective function value and second objective function value, and average CPU time. The CPU time performance criteria is a significant factor to compare the multi-objective optimization techniques in term of time needed to solve the problem.

The values of objective 1 (*Total Cost*), objective 2 (*Total C_{max}*) and CPU time for each weight set obtained from Table 5.3 were listed in the rows in Table 5.4. The ideal solution (F^*) was first obtained by identifying the minimum value in the four columns for each row in Table 5.4, which are 314.546 for weighted average value of objective 1, 2.414 for weighted average value of objective 2, 0.703 sec for average of CPU time. Then, the values in F_i were normalized and the total distance of each criteria was calculated. Here, the direct distance means the deviation of current value from optimal value in criteria divided by optimal value Eq. (5.6).

$$Distance = \frac{F_m - F^*}{F^*} \quad (5.6)$$

As a result, a solution with a minimum total distance was selected. It was obvious that second column with weight sets (0.6,0.4) gets the minimum total distance for 0.101.

Hence, the results obtained by DIS method show that the weight sets (0.6, 0.4) is the best when compared to the result of other sets.

Table 5.4. Determining the Best Weight of WSM for Test Problem 1

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 1	Total Cost	314.546	314.546	334.013	371.898
	Total $C_{\max}$	2.658	2.658	2.536	2.414
	CPU Time(sec)	1.202	0.703	0.857	1.097
	Distance (TCost)	0.000	0.000	0.062	0.182
	Distance ($TC_{\max}$)	0.101	0.101	0.051	0.000
	Distance (CPU Time)	0.710	0.000	0.219	0.560
	Total Distance	0.811	0.101	0.331	0.743

5.2.1.5. Goal attainment method results for test problem 1

Goal Attainment Method (GAM) performs well when some parameters are set appropriately, such as weight and goal vectors. Because, the goal attainment method assigns a goal to each objective and try to get close to these goals.

In this thesis, goal vectors were obtained from individual optimization for each objective function. Result for each weight set applied to goal attainment method was given in Table 5.5.

Table 5.5. Goal Attainment with Different Weights for Test Problem 1

Problem Size	Weight		Average Total Cost (TCost)	Average Total $C_{\max}$ ($TC_{\max}$)	Average CPU Time (sec)
	w1	w2			
Test 1	0.8	0.2	352.431	2.535	1.152
	0.6	0.4	334.013	2.561	1.172
	0.4	0.6	314.546	2.658	0.949
	0.2	0.8	314.546	2.682	1.293

Table 5.6. Determining the Best Weight of GAM for Test Problem 1

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 1	Total Cost	352.431	334.013	314.546	314.546
	Total $C_{\max}$	2.535	2.561	2.658	2.682
	CPU Time(sec)	1.152	1.172	0.949	1.293
	Distance (TCost)	0.120	0.062	0.000	0.000
	Distance ($TC_{\max}$)	0.000	0.010	0.049	0.058
	Distance (CPU Time)	0.214	0.235	0.000	0.362
	Total Distance	0.334	0.307	0.049	0.420

It can be clearly seen from Table 5.6, third column with weight set (0.4,0.6) gets the minimum total distance for 0.049. Hence, the results obtained by applying the DIS method show that the weight set (0.4,0.6) is the best when compared to the result of other sets.

5.2.1.6. NSGA-II results for test problem 1

In this section, as mentioned in Chapter 4, a fast and elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) was used as an evolutionary method. For that reason, a NSGA-II program in MATLAB (NGPM 1.4) was used (Lin, 2011). The program is an implementation of NSGA-II proposed by Deb et al. (2002). The number of population (n_{pop}) and iteration were set as 60 and 100, respectively. Also, crossover (p_c) and mutation (m_c) coefficients were set as 0.7 and 0.3, respectively.

In order to compare and evaluate the performances of the NSGA-II with other methods, various optimization runs were carried out. The values for the first and the second objective function along with the required CPU time to achieve the solutions are given in Table 5.7. Since, NSGA-II searches large candidate solution spaces, 40 runs were conducted to achieve more reliable results (Pasandideh et al., 2013).

Table 5.7. Objective Function Values and CPU time

Run Number	NSGA-II Results		
	First Objective (TCost)	Second Objective (TC _{max})	CPU Time (sec)
1	306.038	2.916	114.470
2	314.546	2.791	112.201
3	321.890	2.679	115.944
4	371.897	2.417	116.183
5	309.036	2.901	110.362
6	306.038	2.916	116.276
7	321.892	2.702	114.449
8	306.038	2.931	112.173
9	330.299	2.601	112.792
10	326.615	2.668	117.804
11	310.783	2.769	109.098
12	306.038	2.887	111.331
13	321.340	2.681	122.912
14	315.023	2.721	119.159
15	306.153	2.895	117.432
16	306.038	2.932	109.811
17	308.451	2.836	117.162
18	330.301	2.599	118.933
19	343.606	2.594	110.577
20	320.659	2.757	106.564
21	306.038	2.984	117.852
22	352.490	2.515	119.233
23	314.546	2.767	121.930
24	343.679	2.578	106.511
25	314.224	2.787	108.150
26	324.542	2.686	118.981

Table 5.7. Objective Function Values and CPU time (Continuation)

Run Number	NSGA-II Results		
	First Objective (TCost)	Second Objective (TC _{max})	CPU Time (sec)
27	306.038	2.834	110.109
28	306.038	2.844	120.734
29	338.126	2.616	116.258
30	338.002	2.602	119.964
31	324.332	2.668	109.209
32	352.490	2.515	109.383
33	308.512	2.854	122.414
34	362.372	2.466	114.638
35	306.038	2.899	112.839
36	336.093	2.624	118.251
37	314.546	2.731	116.572
38	306.038	2.834	109.228
39	316.895	2.736	123.801
40	306.038	2.877	117.799

The DIS method was used to choose the best solution among the Pareto solutions. It can be seen clearly from Table 5.8, the 11th run is the best with a total distance of 0.185 according to the results of the other Pareto solutions.

Table 5.8. The Normalized Outputs of NSGA-II

Run Order	NSGA-II Results			
	Normalized Z ₁ (TCost)	Normalized Z ₂ (TC _{max})	Normalized CPU Time (sec)	Total Distance
1	0.000	0.206	0.075	0.281
2	0.028	0.155	0.053	0.236
3	0.052	0.108	0.089	0.249
4	0.215	0.000	0.091	0.306
5	0.010	0.200	0.036	0.246

Table 5.8. The Normalized Outputs of NSGA-II (Continuation)

Run Order	NSGA-II Results			
	Normalized Z ₁ (TCost)	Normalized Z ₂ (TC _{max})	Normalized CPU Time (sec)	Total Distance
6	0.000	0.206	0.092	0.298
7	0.052	0.118	0.075	0.244
8	0.000	0.213	0.053	0.266
9	0.079	0.076	0.059	0.214
10	0.067	0.104	0.106	0.277
11	0.016	0.146	0.024	0.185
12	0.000	0.194	0.045	0.240
13	0.050	0.109	0.154	0.313
14	0.029	0.126	0.119	0.274
15	0.000	0.198	0.103	0.301
16	0.000	0.213	0.031	0.244
17	0.008	0.173	0.100	0.281
18	0.079	0.075	0.117	0.271
19	0.123	0.073	0.038	0.234
20	0.048	0.141	0.000	0.189
21	0.000	0.235	0.106	0.341
22	0.152	0.041	0.119	0.312
23	0.028	0.145	0.145	0.317
24	0.123	0.067	0.000	0.190
25	0.027	0.153	0.015	0.195
26	0.060	0.111	0.117	0.289
27	0.000	0.173	0.034	0.206
28	0.000	0.177	0.134	0.310
29	0.105	0.082	0.092	0.279
30	0.104	0.077	0.126	0.307
31	0.060	0.104	0.025	0.189

Table 5.8. The Normalized Outputs of NSGA-II (Continuation)

Run Order	NSGA-II Results			
	Normalized Z_1 (TCost)	Normalized Z_2 (TC _{max})	Normalized CPU Time (sec)	Total Distance
32	0.152	0.041	0.027	0.219
33	0.008	0.181	0.149	0.338
34	0.184	0.020	0.076	0.281
35	0.000	0.199	0.059	0.259
36	0.098	0.086	0.110	0.294
37	0.028	0.130	0.094	0.252
38	0.000	0.173	0.026	0.198
39	0.035	0.132	0.162	0.330
40	0.000	0.190	0.106	0.296

Figure 5.2 shows Pareto solutions obtained by NSGA-II, Weighted Sum and Goal Attainment Method for test problem 1.

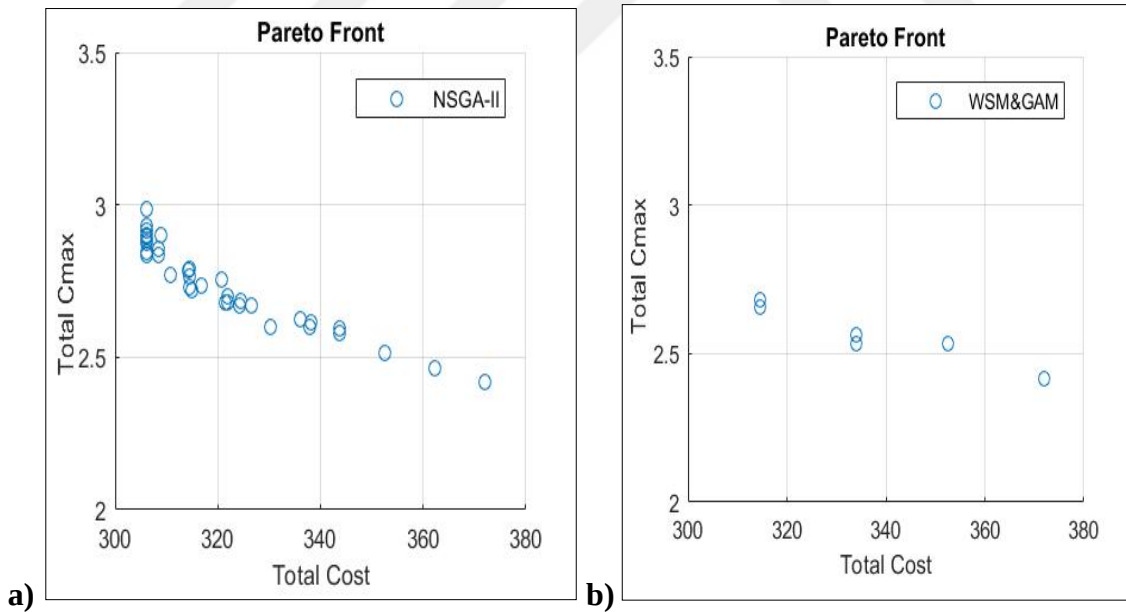


Figure 5.2. Generated Pareto Solutions according to a) NSGA-II method and b) WSM&GAM

5.2.1.7. Comparison of solution methods

After the determining best weights for *Goal Attainment* and *Weighted Sum Methods*, the three MODM methods including; the *LP Metrics/Global Criterion*, the *Mini-max*, *Lexicographic/Preemptive Goal Programming* were also applied to solve the test problems.

These solution methods give different views for the decision makers. For example, the LP Metrics searches for an ideal solution that is nearest to the optimum values of the objectives. The Mini-max method focuses on the weakest objective. The lexicography approach is used to optimize the objective with a higher priority at first, and then optimizes the second objective. In this dissertation, it was determined that total cost (first objective function) is more important than total makespan (second objective function) in all test problems. Results for all methods were given in Table 5.9.

Table 5.9. The Results Obtained by the Solution Methods for *Test Problem 1*

Problem Size	Methods	Total Cost (TCost)	Total C_{max} (TC_{max})	Average CPU Time (sec)
Test 1	Individual Optimization	306.038	2.414	0.640
	Weighted Sum	314.546	2.658	0.703
	LP Metric/Global Criterion	314.546	2.658	0.858
	Min-max	334.013	2.536	1.472
	Goal Attainment	314.546	2.658	0.949
	Lexicographic Goal Programming	334.029	2.536	0.851
	NSGA-II	310.783	2.769	109.098

Table 5.10. Selection of Best Solution Method for Test Problem 1

Problem Size	Methods	Distance (TCost)	Distance (TC_{max})	Distance (CPU time)	Direct Distance
Test 1	Weighted Sum	0.028	0.101	0.098	0.227
	LP Metric/Global Criterion	0.028	0.101	0.341	0.470
	Mini-max	0.091	0.051	1.300	1.442
	Goal Attainment	0.028	0.101	0.483	0.612
	Lexicographic Goal Programming	0.091	0.051	0.330	0.472
	NSGA-II	0.016	0.147	169.466	169.628

Figures 5.3 and 5.4 show the results of the first and second objective function values obtained from all methods, respectively.

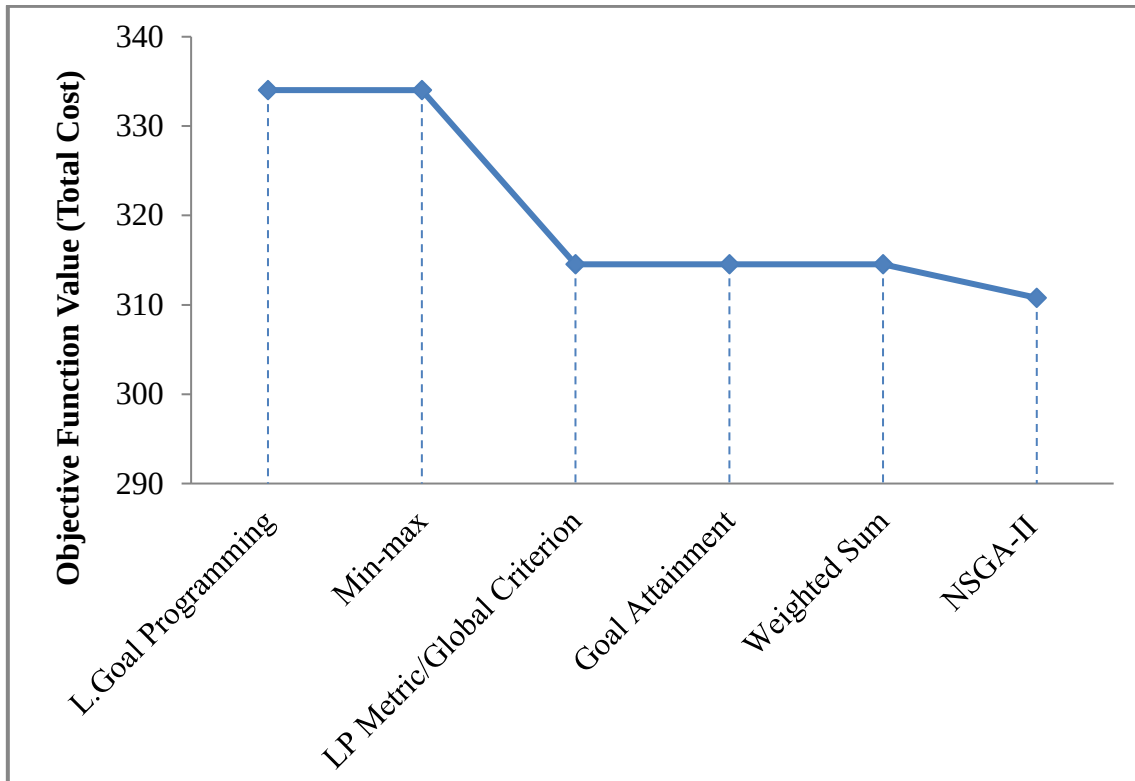


Figure 5.3. Total Cost Values (Z_1) Obtained by All Methods for Test Problem 1

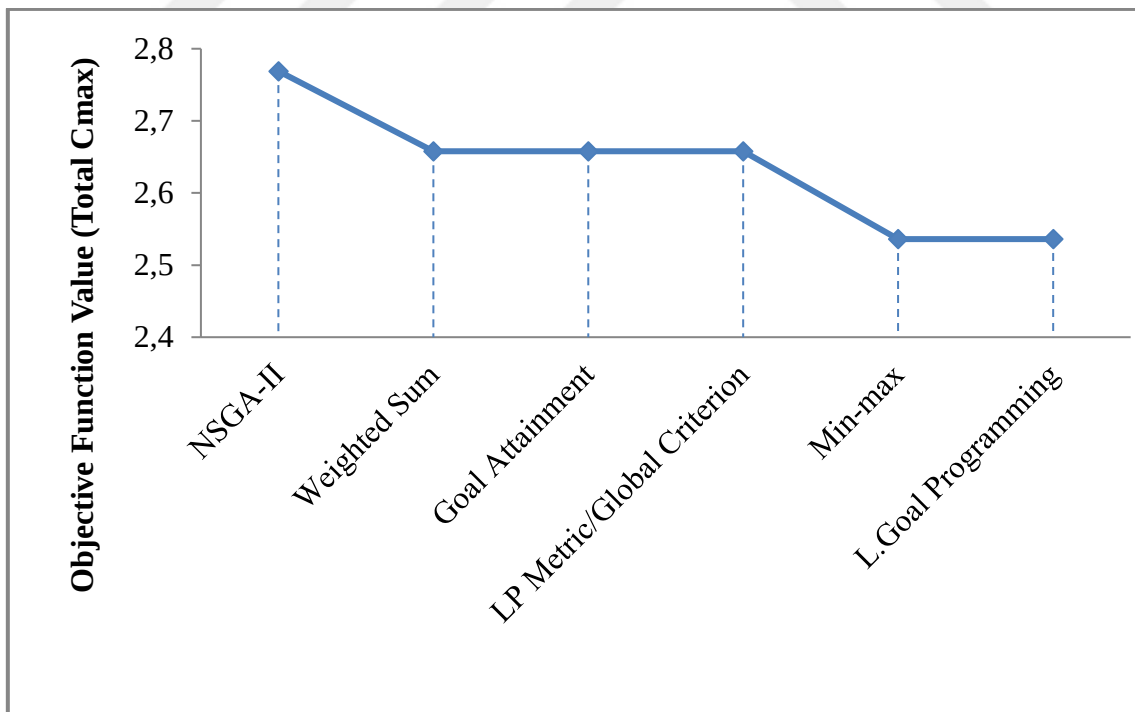


Figure 5.4. Total C_{max} Values (Z_2) Obtained by All Methods for Test Problem 1

The aim of this section is to compare the solution methods according to different point of views. Based on the results obtained Table 5.9, it is obvious that none of the problem-solving methods can optimize both objective functions simultaneously. For instance;

- NSGA-II performs best in optimizing Total Cost (310.783), however it has the worst performance in optimizing the Total C_{max} (2.769).
- It is obvious that the Lexicographic Goal Programming and Mini-max method have the best performance on optimizing Total C_{max} , however they have the worst performance in optimizing the Total Cost (334.029 and 334.013, respectively).
- The CPU time of NSGA-II is not close to other methods. It is very time-consuming method when compared to other methods (109.098 sec). Because, NSGA-II finds a greater number of non-dominated solutions and searches more regions of solutions space.
- Moreover, the Mini-max problem is also time-consuming method among MODM methods (1.472 sec). Since it is not linear, this method was solved by Solving Constraint Integer Programs (SCIP) which can also be used as MINLP solver in GAMS software.
- As shown Table 5.10, DIS method was applied to select best solution method in terms of all performance measures (Z_1 , Z_2 and CPU time). The results show that the *weighted sum method* is the best with (minimum) direct distance of 0.227.

These Pareto solutions provide the decision maker to select the most suitable solution. For example, if the first objective function is more important for the decision makers, the NSGA-II method can be chosen. On the other hand, if the second objective function is more important, Lexicographic Goal Programming or Mini-max methods can be chosen.

The detailed solutions for obtained by weighted sum method for test problem 1 were given in Table 5.11, 5.12, and 5.13.

Table 5.11. Total Amount of Processed Products on Machines

Period	Product 1				Product 2					
	Machine 1		Machine 2		Machine 1			Machine 2		
	Operation No		O(p,h)		O(p,h)			O(p,h)		
	1	2	1	2	1	2	3	1	2	3
t=1	12.193	0	0	12.193	0	16.787	0	16.787	0	16.787
t=2	0	0	5.385	5.385	14.336	14.336	14.336	0	0	0

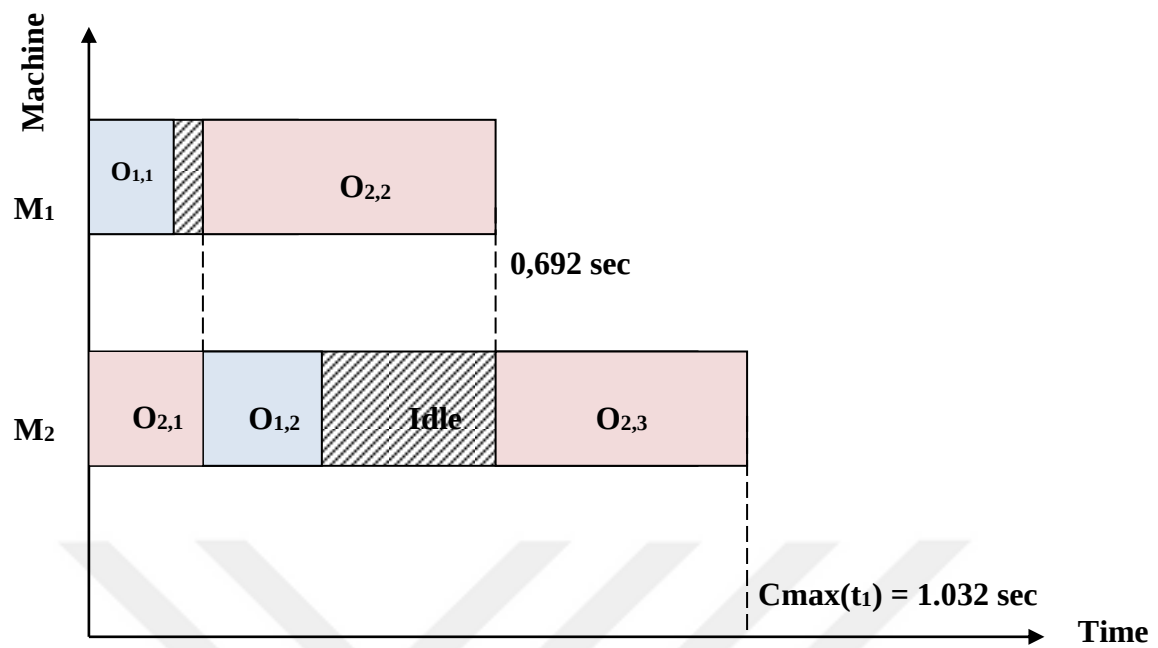
Table 5.12. Product Flow from Manufacturing Facility toward Warehouse

Period	Machine (from)	Product 1	Product 2
		(to) Warehouse	Warehouse
t=1	m=1	0	0
	m=2	12.193	16.787
t=2	m=1	0	14.336
	m=2	5.385	0

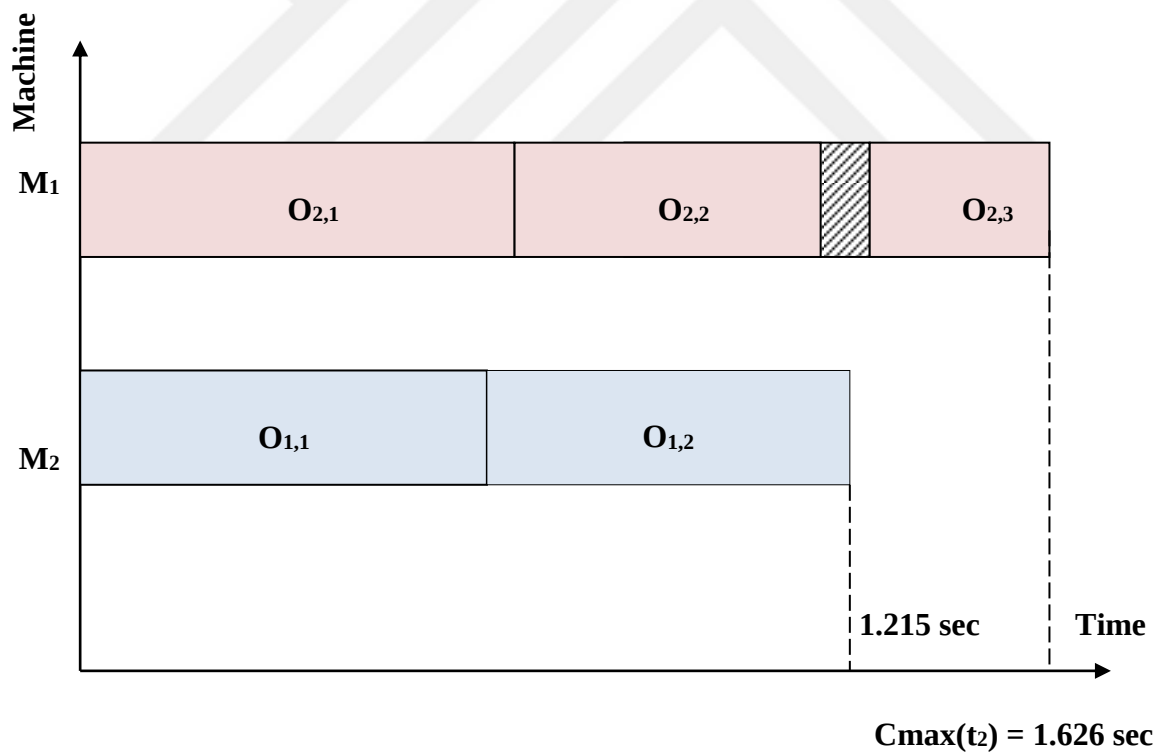
Table 5.13. Product Flow from Warehouse toward Customers

Warehouse (from)	Period	Product 1	Product 2
		(to) Customer	Customer
w=1	t=1	12.193	16.787
	t=2	5.385	14.336

The Gantt chart obtained by weighted sum method for test problem 1 was given in Figure 5.5.



a) Time Period (t=1)



b) Time Period (t=2)

Figure 5.5. The Gantt Chart of Test Problem 1

5.2.2. Test problem 2

Test Problem 2 (2:3:4:4:2:2) involves two machines in manufacturing facility, three potential warehouses, four customers, four periods, two products with two operations.

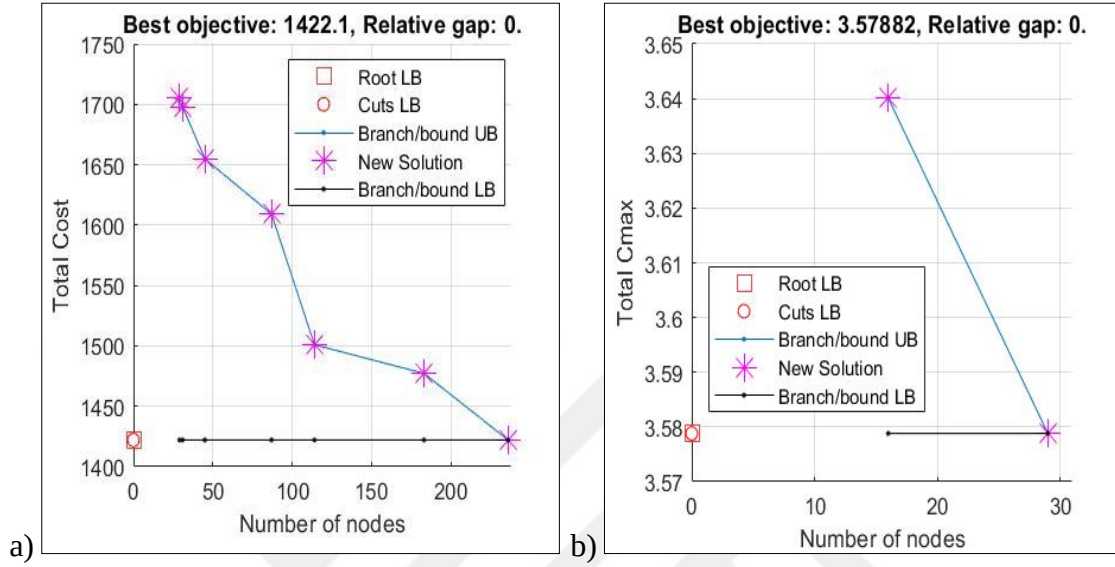


Figure 5.6. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 2

Figure 5.6 shows objective function results obtained by individual optimization for *test problem 2*. The objective function values $Z_1 = 1422.097$ and $Z_2 = 3.579$ were obtained, respectively. $TCost^{max} = 1845.007$ and $TCost^{min} = 1422.097$, $TCmax^{max} = 4.132$ and $TCmax^{min} = 3.579$ were calculated for dimensionality removals and normalizations.

Eq. (5.7) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 1422.097}{422.910} + w2 * \frac{TCmax(x) - 3.579}{0.553} \quad (5.7)$$

Results for each weight set applied to WSM were given in Table 5.14. It can be easily seen that fourth column with weight set (0.2,0.8) gets the minimum total distance for 0.297.

Table 5.14. Determining the Best Weight of WSM for Test Problem 2

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 2	Total Cost	1422.097	1467.862	1584.940	1845.007
	Total C _{max}	4.132	3.960	3.770	3.579
	CPU Time(sec)	1.092	1.047	1.112	0.843
	Distance (TC _{cost})	0.000	0.032	0.115	0.297
	Distance (TC _{max})	0.155	0.106	0.053	0.000
	Distance (CPU Time)	0.295	0.242	0.319	0.000
	Total Distance	0.450	0.381	0.487	0.297

Results for each weight set applied to goal attainment method was given in Table 5.15. According to results, goal attainment method performs best in optimizing *Total Cost* (1494.864), however it has the worst performance in optimizing the *Total C_{max}* (3.960).

Table 5.15. Determining the Best Weight of GAM for Test Problem 2

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 2	Total Cost	1811.247	1641.413	1550.432	1494.864
	Total C _{max}	3.706	3.770	3.831	3.960
	CPU Time(sec)	1.002	1.351	1.182	0.971
	Distance (TC _{cost})	0.212	0.098	0.037	0.000
	Distance (TC _{max})	0.000	0.017	0.034	0.069
	Distance (CPU Time)	0.032	0.391	0.217	0.000
	Total Distance	0.244	0.507	0.288	0.069

Based on the results obtained from Table 5.15, it can be easily seen that fourth column with weight set (0.2,0.8) gets the minimum total distance for 0.069. Results for all methods for test problem 2 were given in Table 5.16.

Table 5.16. The Results Obtained by the Solution Methods for Test Problem 2

Problem Size	Method	Total Cost (TCost)	Total $C_{\max}$ ($TC_{\max}$)	Average CPU Time (sec)
Test 2	Individual Optimization	1422.097	3.579	0.797
	Weighted Sum	1845.007	3.579	0.843
	LP Metric/Global Criterion	1584.940	3.770	1.152
	Mini-max	1548.383	3.833	2.362
	Goal Attainment	1494.864	3.960	0.971
	Lexicographic Goal Programming	1845.007	3.579	1.408
	NSGA-II	1432.836	4.109	185.777

DIS method was applied to select best solution method. The results of method selection were given in Table 5.17. The results show that the *weighted sum method* is the best with (minimum) direct distance of 0.355.

Table 5.17. Selection of Best Method for Test Problem 2

Problem Code	Methods	Distance z_1	Distance z_2	CPU time Distance	Direct Distance
Test 2	Weighted Sum	0.297	0.000	0.058	0.355
	LP Metric/Global Criterion	0.115	0.053	0.445	0.613
	Mini-max	0.089	0.071	1.964	2.123
	Goal Attainment	0.051	0.106	0.218	0.376
	Lexicographic Goal Programming	0.297	0.000	0.767	1.064
	NSGA-II	0.008	0.148	232.095	232.251

5.2.3. Test problem 3

Test Problem 3 (3:2:6:3:2:3) involves three machines in manufacturing facility, two potential warehouses, six customers, three periods, two products each of which with three operations.

As shown Figure 5.7, $TCost^{max} = 2603.275$ and $TCost^{min} = 1898.901$, $TC_{max}^{max} = 5.311$ and $TC_{max}^{min} = 3.654$ were obtained, respectively. After the dimensionality removals and normalizations.

Eq. (5.8) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 1898.901}{704.374} + w2 * \frac{TC_{max}(x) - 3.654}{1.657} \quad (5.8)$$

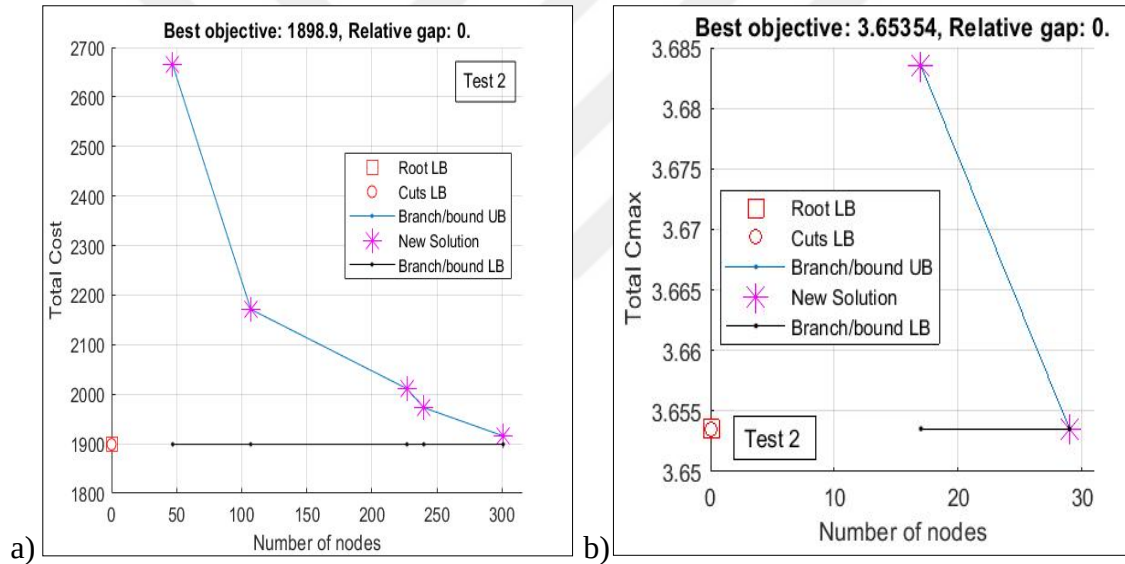


Figure 5.7. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 3

As seen from Table 5.18, it is clear that the third column with weight sets (0.4,0,6) has a minimum total distance of 0.208.

Table 5.18. Determining the Best Weight of WSM for Test Problem 3

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 3	Total Cost	1927.194	2091.047	2134.220	2603.275
	Total C _{max}	4.992	4.122	4.023	3.654
	CPU Time(sec)	1.496	1.662	1.289	1.820
	Distance (TCost)	0.000	0.085	0.107	0.351
	Distance (TC _{max})	0.366	0.128	0.101	0.000
	Distance (CPU Time)	0.161	0.289	0.000	0.412
	Total Distance	0.527	0.502	0.208	0.763

It can be clearly seen from Table 5.19, fourth column with weight set (0.2,0.8) gets the minimum total distance for 0.185. Hence, the results obtained by applying the DIS method show that the weight set (0.2,0.8) is the best when compared to the result of other sets.

Table 5.19. Determining the Best Weight of GAM for Test Problem 3

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 3	Total Cost	2245.139	2134.220	2091.047	1999.222
	Total C _{max}	3.881	4.023	4.332	4.598
	Distance (TCost)	2.952	2.542	3.503	1.841
	Distance (TCost)	0.123	0.068	0.046	0.000
	Distance (TC _{max})	0.000	0.037	0.116	0.185
	Distance (CPU Time)	0.603	0.381	0.903	0.000
	Total Distance	0.726	0.485	1.065	0.185

Results for all methods for Test Problem 3 were given in Table 5.20.

Table 5.20. The Results Obtained by the Solution Methods for Test Problem 3

Problem Size	Methods	Total Cost (TCost)	Total C _{max} (TC _{max})	Average CPU Time (sec)
Test 3	Individual Optimization	1898.901	3.654	1.098
	Weighted Sum	2134.220	4.023	1.289
	LP Metric/Global Criterion	2091.047	4.122	3.275
	Mini-max	2134.220	4.107	5.732
	Goal Attainment	1999.222	4.598	1.841
	Lexicographic Goal Programming	2245.453	3.881	3.184
	NSGA-II	1898.901	4.806	428.865

DIS method was applied again to select best solution method. The best method selection was given in Table 5.21. The results show that the *weighted sum method* is the best with (minimum) direct distance of 0.399.

Table 5.21. The Results Obtained by the Solution Methods for Test Problem 3

Problem Size	Methods	Distance (TCost)	Distance (TC _{max})	Distance (CPU time)	Direct Distance
Test 3	Weighted Sum	0.124	0.101	0.174	0.399
	LP Metric/Global Criterion	0.101	0.128	1.983	2.212
	Mini-max	0.124	0.124	4.220	4.468
	Goal Attainment	0.053	0.258	0.677	0.988
	Lexicographic Goal Programming	0.183	0.062	1.900	2.144
	NSGA-II	0.000	0.315	389.587	389.903

5.2.4. Test problem 4

Test Problem 4 (4:1:6:3:2:5) involves four machines in manufacturing facility, one potential warehouse, six customers, three periods, two products each of which with three and five operations, respectively.

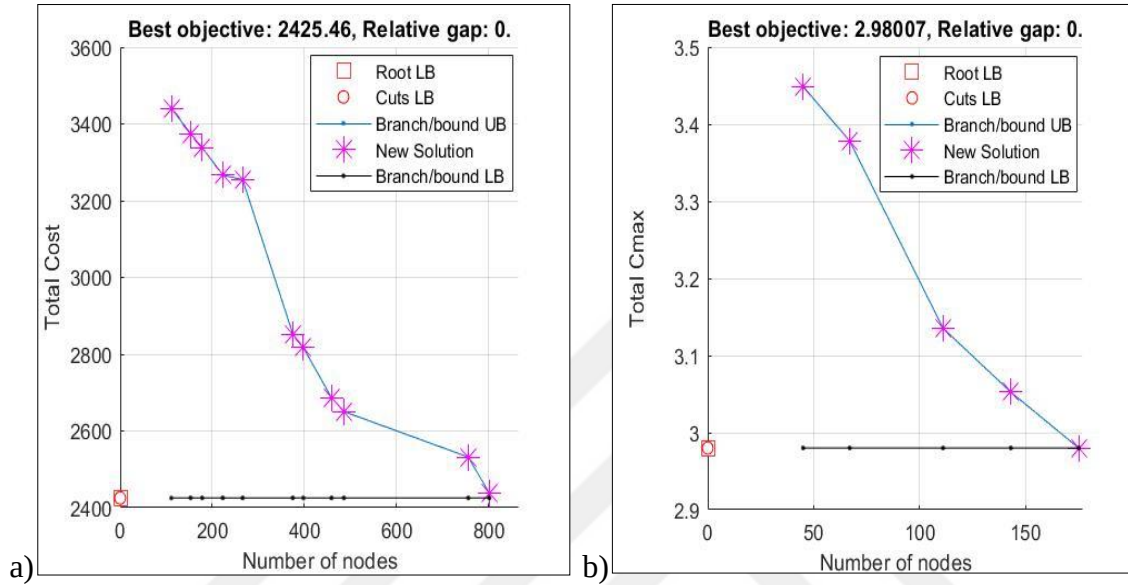


Figure 5.8. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 4

Figure 5.8 shows objective function results obtained by individual optimization for *test problem 4*. The objective function values $Z_1 = 2425.456$ and $Z_2 = 2.980$ were obtained, respectively. $TCost^{max} = 3915.609$ and $TCost^{min} = 2425.456$, $TCmax^{max} = 7.347$ and $TCmax^{min} = 2.980$ were calculated for dimensionality removals and normalizations.

Eq. (5.9) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 2425.456}{1490.153} + w2 * \frac{TCmax(x) - 2.980}{4.367} \quad (5.9)$$

Results for each weight set applied to WSM were given in Table 5.22. It can be easily seen that third column with weight set (0.4,0.6) gets the minimum total distance for 0.503.

Table 5.22. Determining the Best Weight of WSM for Test Problem 4

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 4	Total Cost	2455.613	2868.807	3173.653	3810.771
	Total C _{max}	6.809	4.644	3.696	3.054
	CPU Time(sec)	2.041	3.010	1.935	2.245
	Distance (TC _{cost})	0.000	0.168	0.292	0.552
	Distance (TC _{max})	1.230	0.521	0.210	0.000
	Distance (CPU Time)	0.055	0.556	0.000	0.160
	Total Distance	1.284	1.244	0.503	0.712

Results for each weight set applied to Goal Attainment method was given in Table 5.23. According to results, goal attainment method performs best in optimizing *Total Cost* (2646.151), however it has the worst performance in optimizing the *Total C_{max}* (5.677).

Table 5.23. Determining the Best Weight of GAM for Test Problem 4

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 4	Total Cost	3294.686	3021.470	2860.064	2646.151
	Total C _{max}	3.617	4.144	4.890	5.677
	CPU Time(sec)	6.782	7.744	7.480	7.821
	Distance (TC _{cost})	0.245	0.142	0.081	0.000
	Distance (TC _{max})	0.000	0.146	0.352	0.570
	Distance (CPU Time)	0.000	0.142	0.103	0.153
	Total Distance	0.245	0.429	0.536	0.723

Based on the results obtained from Table 5.23, it can be easily seen that first column with weight set (0.8,0.2) gets the minimum total distance for 0.245. Results for all methods for test problem 4 were given in Table 5.24.

Table 5.24. The Results Obtained by the Solution Methods for Test Problem 4

Problem Size	Method	Total Cost (TCost)	Total $C_{\max}$ ($TC_{\max}$)	Average CPU Time (sec)
Test 4	Individual Optimization	2425.456	2.980	1.112
	Weighted Sum	3173.653	3.696	1.935
	LP Metric/Global Criterion	3021.470	3.999	4.102
	Mini-max	3120.628	3.853	18.835
	Goal Attainment	3294.686	3.617	6.782
	Lexicographic Goal Programming	3197.355	3.661	6.636
	NSGA-II	2864.307	4.840	864.315

DIS method was applied to select best solution method. The results of method selection were given in Table 5.25. The results show that the *weighted sum method* is the best with (minimum) direct distance of 1.289.

Table 5.25. Selection of Best Method for Test Problem 4

Problem Size	Methods	Distance z_1	Distance z_2	CPU time Distance	Direct Distance
Test 4	Weighted Sum	0.308	0.240	0.740	1.289
	LP Metric/Global Criterion	0.246	0.342	2.689	3.277
	Mini-max	0.287	0.293	15.938	16.518
	Goal Attainment	0.358	0.214	5.099	5.671
	Lexicographic Goal Programming	0.318	0.229	4.968	5.514
	NSGA-II	0.181	0.624	776.262	777.070

5.2.5. Test problem 5

Test Problem 5 (3:2:2:3:3:4) involves three machines in manufacturing facility, two potential warehouses, two customers, three periods, three products each of which with two, four, and four operations, respectively.

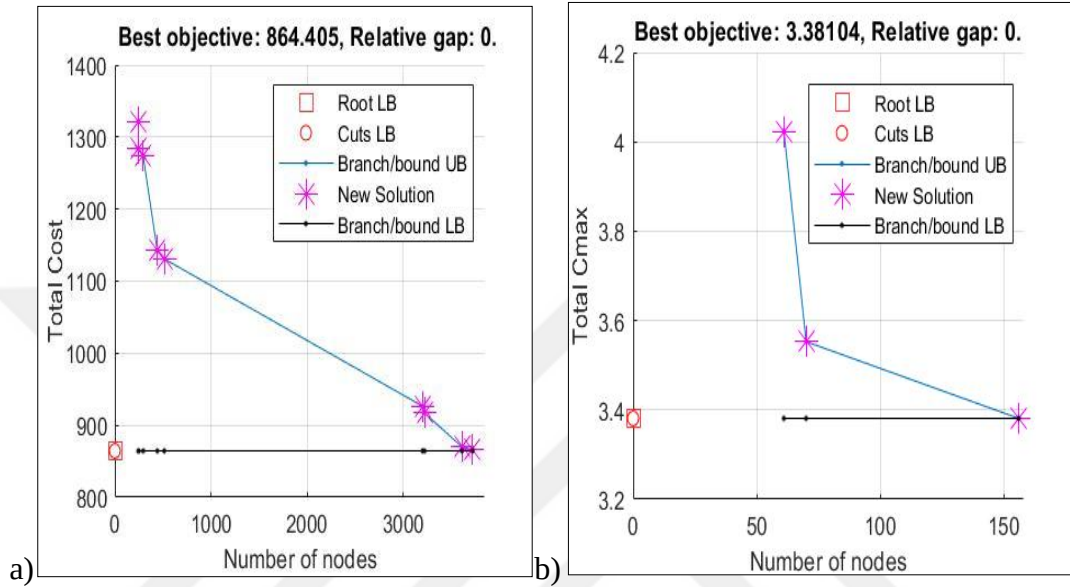


Figure 5.9. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 5

As shown Figure 5.9, the objective-function values $Z_1 = 864.405$ and $Z_2 = 3.381$ were obtained, respectively. $TCost^{max} = 1215.072$, $TCost^{min} = 864.405$, $TCmax^{max} = 5.778$ and $TCmax^{min} = 3.381$ were calculated for dimensionality removals and normalization.

Eq. (5.10) the fitness function changes as follows:

$$F(x) = w_1 * \frac{TCost(x) - 864.405}{350.667} + w_2 * \frac{TCmax(x) - 3.381}{2.397} \quad (5.10)$$

Results for each weight set applied to WSM were given in Table 5.26. It can be easily seen that third column with weight set (0.4,0.6) gets the minimum total distance (0.260).

Table 5.26. Determining the Best Weight of WSM for Test Problem 5

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 5	Total Cost	887.101	902.584	1055.012	1114.587
	Total $C_{\max}$	5.019	4.801	3.692	3.449
	CPU Time(sec)	66.934	81.731	52.104	64.170
	Distance (TCost)	0.000	0.017	0.189	0.256
	Distance (TC _{max})	0.455	0.392	0.070	0.000
	Distance (CPU Time)	0.285	0.569	0.000	0.232
	Total Distance	0.740	0.978	0.260	0.488

Based on the results obtained from goal attainment method in Table 5.27, it can be easily seen that second column with weight set (0.6,0.4) gets the minimum total distance (0.179).

Table 5.27. Determining the Best Weight of GAM for Test Problem 5

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 5	Total Cost	1055.012	1003.137	960.617	915.043
	Total $C_{\max}$	3.707	4.013	4.367	4.766
	CPU Time(sec)	82.355	53.878	60.415	64.863
	Distance (TCost)	0.153	0.096	0.050	0.000
	Distance (TC _{max})	0.000	0.083	0.178	0.286
	Distance (CPU Time)	0.529	0.000	0.121	0.204
	Total Distance	0.682	0.179	0.349	0.490

The results for all method were given in Table 5.28.

Table 5.28. The Results Obtained by the Solution Methods for *Test Problem 5*

Problem Size	Method	Total Cost (TC_{Cost})	Total C_{max} (TC_{max})	Average CPU Time (sec)
Test 5	Individual Optimization	864.405	3.381	15.324
	Weighted Sum	1055.012	3.692	52.104
	LP Metric/Global Criterion	1003.137	3.976	202.456
	Mini-max	1014.603	3.968	612.004
	Goal Attainment	1003.137	4.013	53.878
	Lexicographic Goal Programming	1114.431	3.451	68.033
	NSGA-II	864.470	5.749	1157.161

As shown from Table 5.29, the results show that the *weighted sum method* is the best with (minimum) direct distance of 2.713.

Table 5.29. Selection of Best Solution Method for *Test Problem 5*

Problem Size	Methods	Distance (TC_{Cost})	Distance (TC_{max})	Distance (CPU time)	Direct Distance
Test 5	Weighted Sum	0.221	0.092	2.400	2.713
	LP Metric/Global Criterion	0.160	0.176	12.212	12.548
	Mini-max	0.174	0.174	38.938	39.285
	Goal Attainment	0.160	0.187	2.516	2.863
	Lexicographic Goal Programming	0.289	0.021	3.440	3.750
	NSGA-II	0.000	0.700	74.513	75.213

5.2.6. Test problem 6

Test Problem 6 (3:3:4:4:3:5) involves three machines in manufacturing facility, three potential warehouses, four customers, four periods, three products each of which with three, three, and five operations, respectively.

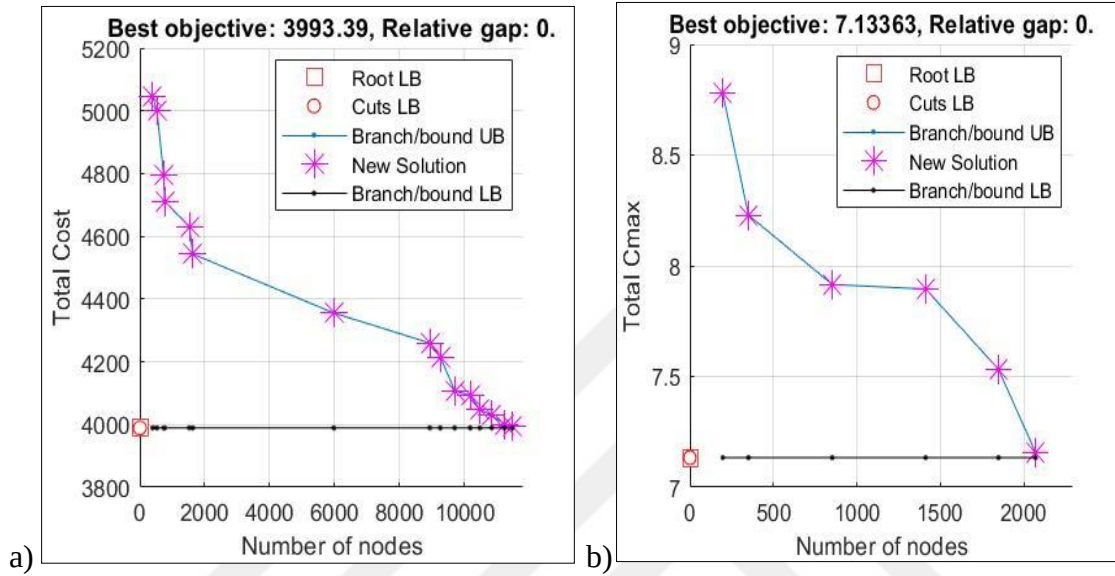


Figure 5.10. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 6

Figure 5.10 shows objective function results obtained by individual optimization for *test problem 6*. The objective function values $Z_1 = 3993.386$ and $Z_2 = 7.134$ were obtained, respectively. $TCost^{max} = 5441.260$ and $TCost^{min} = 3993.386$, $TCmax^{max} = 10.521$ and $TCmax^{min} = 7.134$ were calculated for dimensionality removals and normalizations.

Eq. (5.11) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 3993.386}{1447.874} + w2 * \frac{TCmax(x) - 7.134}{3.387} \quad (5.11)$$

Results for each weight set applied to WSM were given in Table 5.30. It can be easily seen that third column with weight set (0.4,0.6) gets the minimum total distance for 0.155.

Table 5.30. Determining the Best Weight of WSM for *Test Problem 6*

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 6	Total Cost	4022.181	4470.524	4567.214	4760.308
	Total C _{max}	9.878	7.611	7.398	7.260
	CPU Time(sec)	78.670	80.611	49.872	60.518
	Distance (TC _{Cost})	0.000	0.111	0.136	0.184
	Distance (TC _{max})	0.361	0.048	0.019	0.000
	Distance (CPU Time)	0.577	0.616	0.000	0.213
	Total Distance	0.938	0.776	0.155	0.397

Results for each weight set applied to goal attainment method was given in Table 5.31. According to results, GAM performs best in optimizing *Total Cost* (4191.606), however it has the worst performance in optimizing the *Total C_{max}* (8.989).

Table 5.31. Determining the Best Weight of GAM for *Test Problem 6*

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 6	Total Cost	4542.588	4453.294	4324.406	4191.606
	Total C _{max}	7.464	7.851	8.296	8.989
	CPU Time(sec)	268.513	344.831	298.786	321.147
	Distance (TC _{Cost})	0.084	0.062	0.032	0.000
	Distance (TC _{max})	0.000	0.052	0.111	0.204
	Distance (CPU Time)	0.000	0.284	0.113	0.196
	Total Distance	0.084	0.399	0.256	0.400

Based on the results obtained from Table 5.31, it can be easily seen that first column with weight set (0.8,0.2) gets the minimum total distance for 0.084. Results for all methods for Test Problem 6 were given in Table 5.32.

Table 5.32. The Results Obtained by the Solution Methods for Test Problem 6

Problem Size	Method	Total Cost (TCost)	Total $C_{\max}$ (TC _{max})	Average CPU Time (sec)
Test 6	Individual Optimization	3993.386	7.134	22.534
	Weighted Sum	4567.214	7.398	49.870
	LP Metric/Global Criterion	4517.788	7.476	80.857
	Mini-max	4426.576	7.909	525.875
	Goal Attainment	4542.588	7.464	268.513
	Lexicographic Goal Programming	4566.744	7.398	91.236
	NSGA-II	4512.002	7.842	2576.387

The results of method selection were given in Table 5.33. The results show that the *weighted sum method* is the best with (minimum) direct distance of 1.394.

Table 5.33. Selection of Best Method for Test Problem 6

Problem Code	Methods	Distance z_1	Distance z_2	CPU time Distance	Direct Distance
Test 6	Weighted Sum	0.144	0.037	1.213	1.394
	LP Metric/Global Criterion	0.131	0.048	2.588	2.767
	Mini-max	0.108	0.109	22.337	22.554
	Goal Attainment	0.138	0.046	10.916	11.100
	Lexicographic Goal Programming	0.144	0.037	3.049	3.229
	NSGA-II	0.130	0.099	113.333	113.562

5.2.7. Test problem 7

Test Problem 7 (4:2:2:4:2:6) involves four machines in manufacturing facility, two potential warehouses, two customers, four periods, two products each of which with four and six operations, respectively.

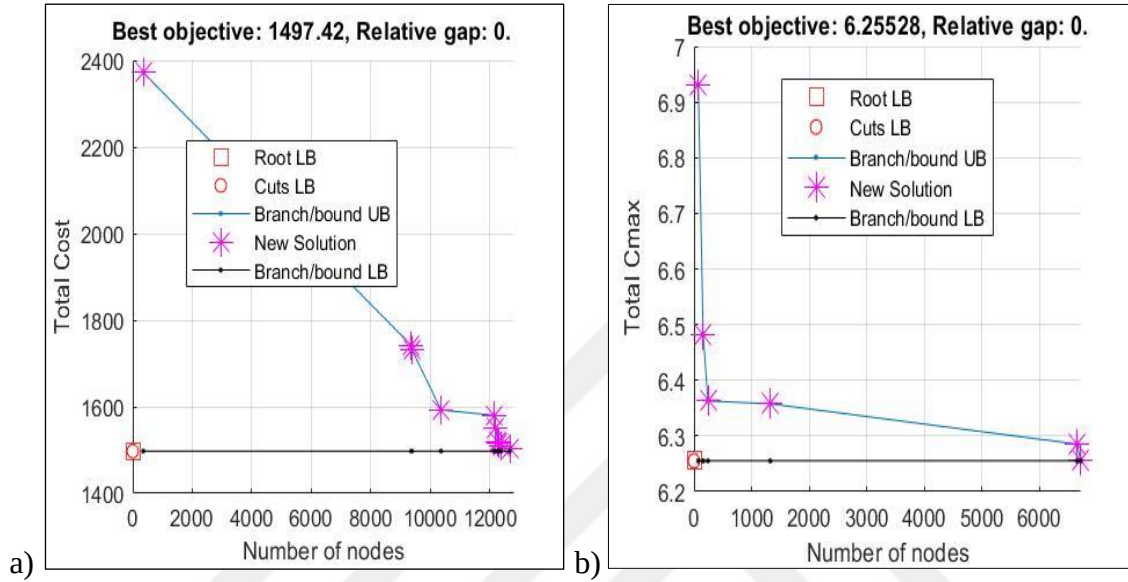


Figure 5.11. a) Total Cost and b) Total C_{max} Results Obtained by Individual Optimization for Test Problem 7

Figure 5.11 shows objective function results obtained by individual optimization for *test problem 7*. The objective function values $Z_1 = 1497.421$ and $Z_2 = 6.255$ were obtained, respectively. $TCost^{max} = 2269.173$ and $TCost^{min} = 1497.421$, $TCmax^{max} = 12.080$ and $TCmax^{min} = 6.255$ were calculated for dimensionality removals and normalizations.

Eq. (5.10) the fitness function changes as follows:

$$F(x) = w1 * \frac{TCost(x) - 1497.421}{771.752} + w2 * \frac{TCmax(x) - 6.255}{5.825} \quad (5.12)$$

Results for each weight set applied to WSM was given in Table 5.34. It can be easily seen that third column with weight set (0.4,0.6) get the minimum total distance for 0.295.

Table 5.34. Determining the Best Weight of WSM for Test Problem 7

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 7	Total Cost	1507.722	1692.395	1764.562	2045.059
	Total Cmax	11.097	7.748	7.121	6.333
	CPU Time(sec)	268.211	252.834	234.332	277.229
	Distance (TCost)	0.000	0.122	0.170	0.356
	Distance (TC _{max})	0.752	0.223	0.124	0.000
	Distance (CPU Time)	0.145	0.079	0.000	0.183
	Total Distance	0.897	0.425	0.295	0.539

According to Table 5.35, goal attainment method performs best in optimizing *Total Cost* (1597.711), however it has the worst performance in optimizing the *Total Cmax* (9.283).

Table 5.35. Determining the Best Weight of GAM for Test Problem 7

Problem Size	Criterion	Weight (w1, w2)			
		(0.8,0.2)	(0.6,0.4)	(0.4,0.6)	(0.2,0.8)
Test 7	Total Cost	1844.958	1747.09	1673.94	1597.711
	Total C _{max}	6.911	7.511	8.253	9.283
	CPU Time(sec)	306.340	341.564	363.621	331.219
	Distance (TCost)	0.155	0.093	0.048	0.000
	Distance (TC _{max})	0.000	0.087	0.194	0.343
	Distance (CPU Time)	0.000	0.115	0.187	0.081
	Total Distance	0.155	0.295	0.429	0.424

Based on the results obtained from Table 5.35, it can be easily seen that first column with weight set (0.8,0.2) gets the minimum total distance for 0.155. Results for all methods for test problem 7 were given in Table 5.36.

Table 5.36. The Results Obtained by the Solution Methods for Test Problem 7

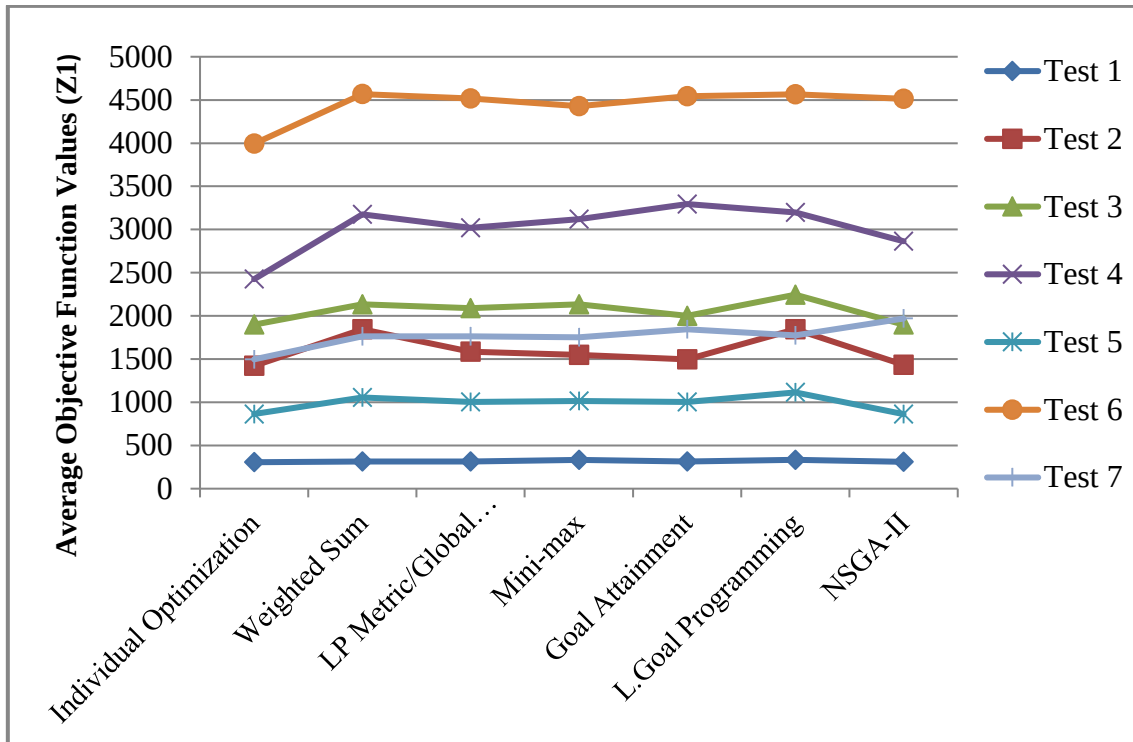
Problem Size	Method	Total Cost (TCost)	Total $C_{\max}$ (TC _{max})	Average CPU Time (sec)
Test 7	Individual Optimization	1497.421	6.255	32.344
	Weighted Sum	1764.562	7.121	234.332
	LP Metric/Global Criterion	1764.562	7.121	332.554
	Mini-max	1754.137	7.207	807.846
	Goal Attainment	1844.958	6.911	306.340
	Lexicographic Goal Programming	1776.023	7.071	295.689
	NSGA-II	1969.595	6.945	3572.023

The results of method selection were given in Table 5.37. The results show that the *weighted sum method* is the best with (minimum) direct distance of 6.562.

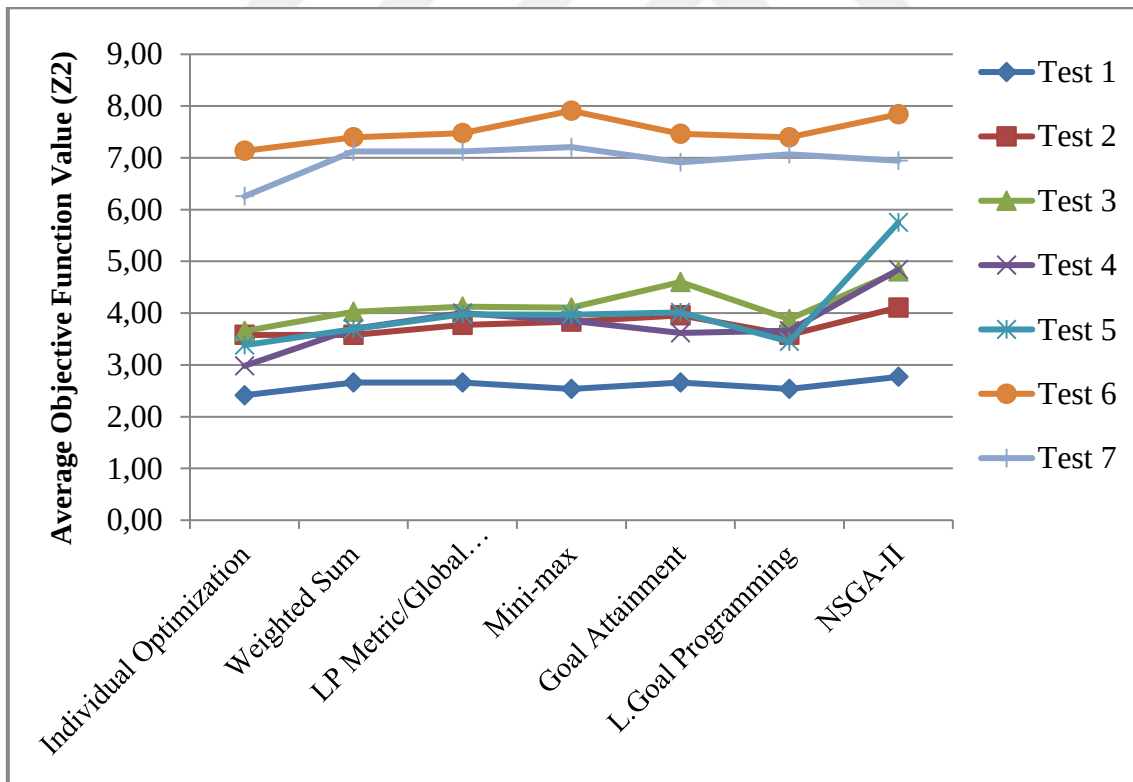
Table 5.37. Selection of best method for *Test Problem 7*

Problem Code	Methods	Distance z_1	Distance z_2	CPU time Distance	Direct Distance
Test 7	Weighted Sum	0.178	0.138	6.245	6.562
	LP Metric/Global Criterion	0.178	0.138	9.282	9.599
	Mini-max	0.171	0.152	23.977	24.3
	Goal Attainment	0.232	0.105	8.471	8.808
	Lexicographic Goal Programming	0.186	0.130	8.142	8.459
	NSGA-II	0.315	0.110	109.439	109.9

Figure 5.12 shows the average objective function values of Z_1 and Z_2 for small and medium size test problems obtained employing all the MODM methods. Figure 5.13 shows Box-plot of the average CPU times of all methods.



a)



b)

Figure 5.12. The Average Objective Function Values of a) Z_1 and b) Z_2 for Small and Medium Size Test Problems

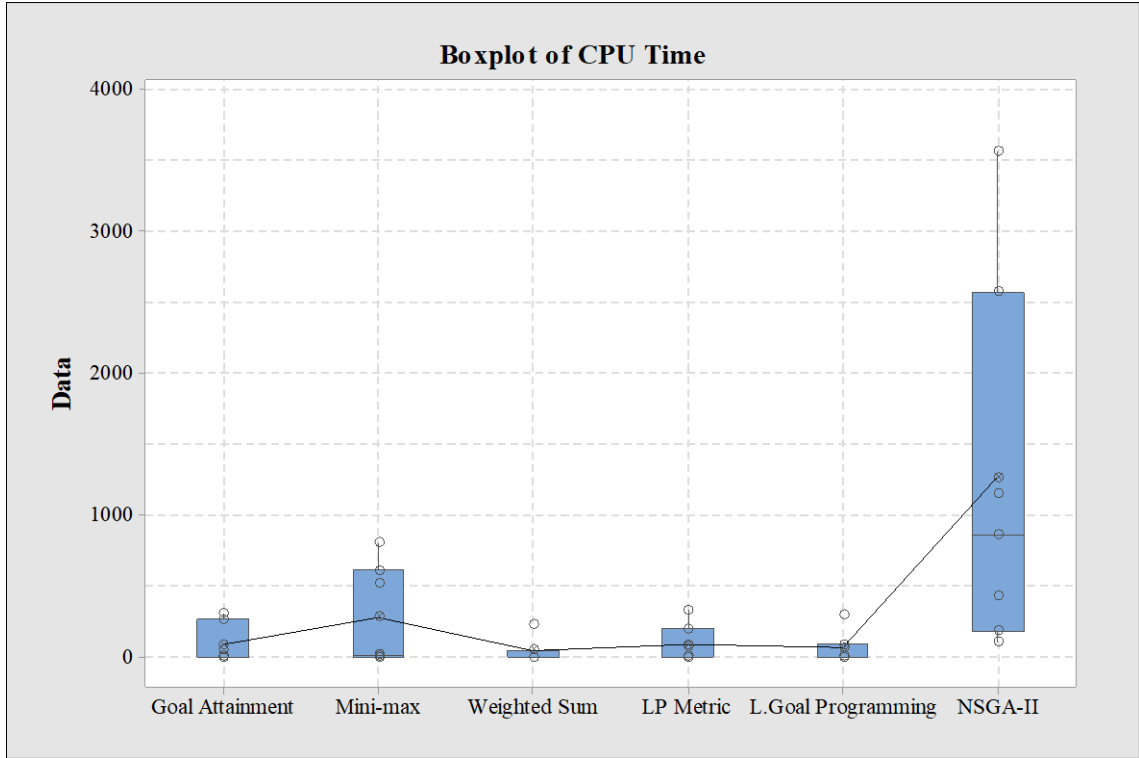


Figure 5.13. Boxplot of the Average CPU-Times Obtained Employing All Methods

As a conclusion, when the results of calculations and the performance of the methods are evaluated together, we can result such as follows:

- The results indicated that different Pareto solutions were obtained by the test problems using multi-objective solution methods, thus, the decision makers might have different preferences depending on priority sequence of objective functions.
- The computational results showed that NSGA-II method cannot perform well for minimization of the second objective function value in all test problems. Moreover, there are big differences between NSGA-II method and other solution methods in term of average CPU-time (average CPU time value for all test problems is about 1270.518 sec). Because, NSGA-II can search more solution space and thus has more potential to find a large number of non-dominated solutions. For this reason, this high CPU time is acceptable. Therefore, the NSGA-II method is inefficient in term of CPU-time criteria for solving large-size problems (Rabbani et al., 2017).

- The solution method with less CPU-time is preferred over other methods. So, weighted sum and lexicographic goal programming methods seem to have good performance regarding the CPU-time measure (average CPU time is about 48,725 and 66.720, respectively), since their computation times are significantly less comparing to other methods.
- Weighted Sum method has shown impressive performance on the small and medium size test problems to approximate the Pareto surface quickly. This method has found a solution with a good tradeoff between first and second objective functions and mean CPU-time.

5.3. Large Sized Test Problems

Another experiment was constructed for large-sized test problems depending on Table 5.38. Since, the NSGA-II method is inefficient in term of CPU-time measure for solving the problem, it is not used in this section. Moreover, the run time was limited up 3600 second for each problem and first and second objective function values were used as performance criterion for DIS method.

Table 5.38. The Generated Large Sized Problems Instances

Problem No	Problem Size (m:w:c:t:p:h _{max})	No. of Machine (m)	No. of Warehouses (w)	No. of Customers (c)	No. of Periods (t)	No. of Products (p)	No. of Operations O(p,h)
Test 8	(6:4:10:21:6:3)	6	4	10	21	6	p(1)=2,p(2)=2 p(3)=2,p(4)=3 p(1)=3,p(2)=3
Test 9	(6:3:10:7:4:6)	6	3	10	7	4	p(1)=4,p(2)=4 p(3)=6,p(4)=6
Test 10	(4:2:8:14:4:5)	4	2	8	14	4	p(1)=4,p(2)=4 p(3)=5,p(4)=5
Test 11	(6:3:6:21:4:6)	6	3	6	21	4	p(1)=3,p(2)=3 p(3)=6,p(4)=6
Test 12	(8:2:8:7:6:6)	8	2	8	7	6	p(1)=3,p(2)=3 p(1)=4,p(2)=4 p(3)=6,p(4)=6
Test 13	(4:4:12:14:5:5)	4	4	12	14	5	p(1)=2,p(2)=3 p(3)=4,p(4)=5 p(1)=5
Test 14	(6:3:12:21:5:8)	6	3	12	21	5	p(1)=2,p(2)=2 p(3)=4,p(4)=4 p(1)=8

Table 5.39 shows objective function results (Z_1 and Z_2) obtained by individual optimization for each test problem.

Table 5.39. The Objective Functions Values Obtained by Individual Optimization

Problem No	Problem Size	Total Cost (TCost)	Total $C_{\max}$ (TC _{max})
Test 8	(6:4:10:21:6:3)	55968.404	16.536
Test 9	(6:3:10:7:4:6)	24834.671	9.110
Test 10	(4:2:8:14:4:5)	41525.546	19.685
Test 11	(6:3:6:21:4:6)	41392.074	30.011
Test 12	(8:2:8:7:6:6)	21960.478	9.082
Test 13	(4:4:12:14:5:5)	66400.705	20.965
Test 14	(6:3:12:21:5:8)	92433.101	35.665

Results for each weight set applied to weighted sum method were given in Table 5.40.

Table 5.40. Weighted Sum Method with Different Weights for Large Sized Test Problems

Problem No	Objectives	Weights			
		[0.8,0.2]	[0.6,0.4]	[0.4,0.6]	[0.2,0.8]
Test 8	Z_1 (TCost)	57719.821	60904.783	66745.249	81222.831
	Z_2 (TC _{max})	30.804	25.808	21.631	17.573
Test 9	Z_1	25283.764	27504.898	30527.591	36823.395
	Z_2	18.415	14.034	11.614	9.472
Test 10	Z_1	42073.040	44269.170	48338.339	54464.382
	Z_2	31.093	26.275	22.537	20.154
Test 11	Z_1	42354.038	44396.243	48979.174	55827.980
	Z_2	55.851	46.560	37.354	32.002
Test 12	Z_1	22533.496	23802.312	26200.014	29630.932
	Z_2	17.261	13.899	11.125	9.659
Test 13	Z_1	67022.540	69529.052	74168.135	82377.268
	Z_2	31.829	26.942	23.857	21.137
Test 14	Z_1	94278.330	97101.822	103261.614	115367.188
	Z_2	60.313	50.634	43.685	37.976

DIS method was applied to select best weights for weighted sum and goal attainment methods. The results of weight selection were given in Appendix 5.1 and 5.2. Results for each weight set applied to goal attainment method were given in Table 5.41.

Table 5.41. GAM Results with Different Weights for Large Sized Test Problems

Problem No	Objectives	Weights			
		[0.8,0.2]	[0.6,0.4]	[0.4,0.6]	[0.2,0.8]
Test 8	Z1(Tcost)	72969.660	66926.984	62968.025	59765.096
	Z2(TCmax)	19.502	21.634	23.863	27.134
Test 9	Z1	31588.646	29955.427	28058.200	26604.750
	Z2	11.108	12.113	13.389	15.338
Test 10	Z1	50038.184	47244.293	45313.552	43569.329
	Z2	21.705	23.304	25.079	27.446
Test 11	Z1	51118.247	48263.979	45959.127	43848.828
	Z2	35.223	38.573	42.814	48.377
Test 12	Z1	27081.807	25452.328	24272.677	23207.377
	Z2	10.634	11.906	13.286	15.140
Test 13	Z1	76562.707	73374.774	70920.042	68863.165
	Z2	22.804	24.331	25.873	28.097
Test 14	Z1	108803.505	102763.246	99018.925	96004.164
	Z2	40.717	44.166	47.859	53.297

Results of LP metric/Global criterion, Mini-max and Lexicographic goal programming methods were given in Table 5.42.

Table 5.42. The Results Obtained by LP metric/Global Criterion, Mini-max and Lexicographic Goal Programming Methods for Large Sized Test Problems

Problem No	Objectives	Methods		
		LP Metric /G.Criterion	Mini-Max	L.Goal Programming
Test 8	Z1 (Tcost)	63571.994	83538.187	69784.896
	Z2 (TCmax)	23.376	24.682	20.351
Test 9	Z1	28130.058	35143.471	32377.158
	Z2	13.300	12.891	10.769
Test 10	Z1	46450.643	58790.522	49159.297
	Z2	23.869	27.869	22.073
Test 11	Z1	47490.892	56152.714	51629.829
	Z2	39.473	40.770	34.794
Test 12	Z1	25076.443	30984.073	26960.153
	Z2	12.198	12.801	10.640
Test 13	Z1	70791.941	87670.489	76546.843
	Z2	25.880	27.662	22.747
Test 14	Z1	99530.494	120565.419	105325.182
	Z2	47.116	43.621	41.134

As shown Table 5.43, DIS method was applied to select best solution method in terms of two performance criterion (Z_1 and Z_2). The results show that the Lexicographic Goal Programming is the best with (minimum) direct distance for all large sized test problems.

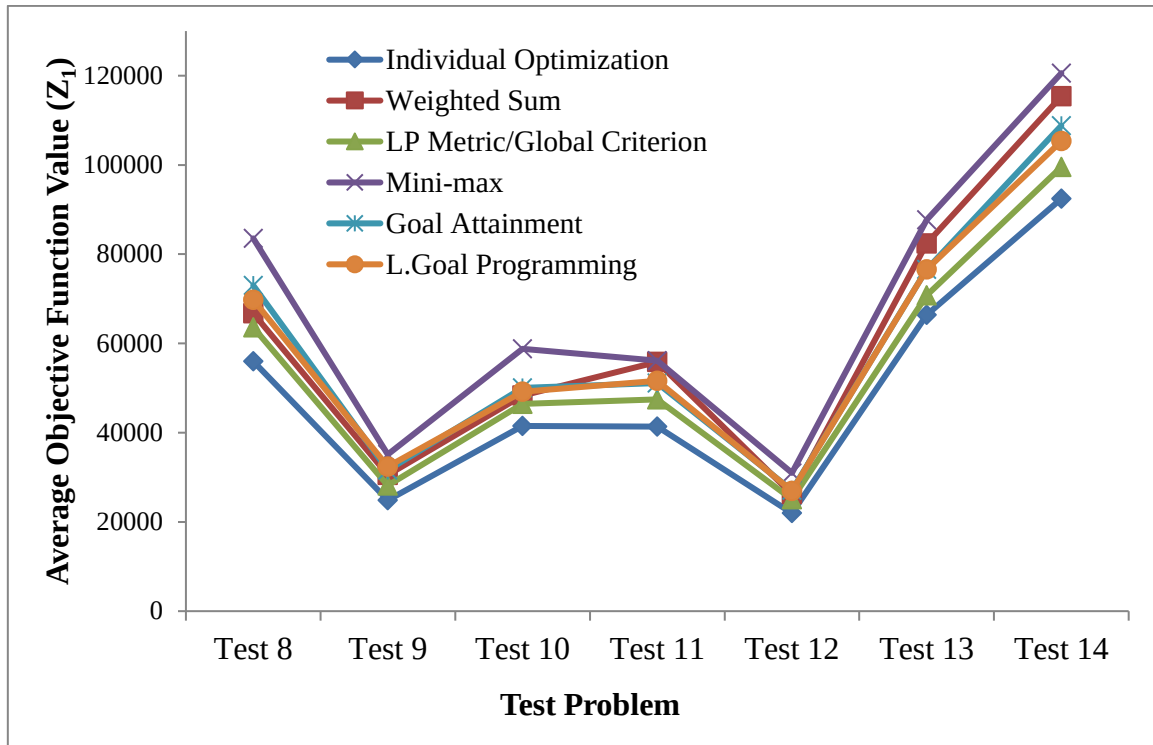
Table 5.43. The Results Obtained by the all MODM for Large Sized Test Problems

Problem No	Criterion	(TCost)	(TC _{max})	Distance z1	Distance z2	Direct Distance
8	Individual Optimization (A)	55968.404	16.536	0.000	0.000	0.000
	Weighted Sum (B)	66745.249	21.631	0.193	0.308	0.501
	LP Metric/Global Criterion (C)	63571.994	23.376	0.136	0.414	0.549
	Mini-max (D)	83538.187	24.682	0.493	0.493	0.985
	Goal Attainment (E)	72969.660	19.502	0.304	0.179	0.483
	L.Goal Programming (F)	69784.896	20.351	0.247	0.231	0.478
9	(A)	24834.671	9.110	0.000	0.000	0.000
	(B)	30527.591	11.614	0.229	0.275	0.504
	(C)	28130.058	13.300	0.133	0.460	0.593
	(D)	35143.471	12.891	0.415	0.415	0.830
	(E)	31588.646	11.108	0.272	0.219	0.491
	(F)	32377.158	10.769	0.304	0.182	0.486
10	(A)	41525.546	19.685	0.000	0.000	0.000
	(B)	48338.339	22.537	0.164	0.145	0.309
	(C)	46450.643	23.869	0.119	0.213	0.331
	(D)	58790.522	27.869	0.416	0.416	0.832
	(E)	50038.184	21.705	0.205	0.103	0.308
	(F)	49159.297	22.073	0.184	0.121	0.305

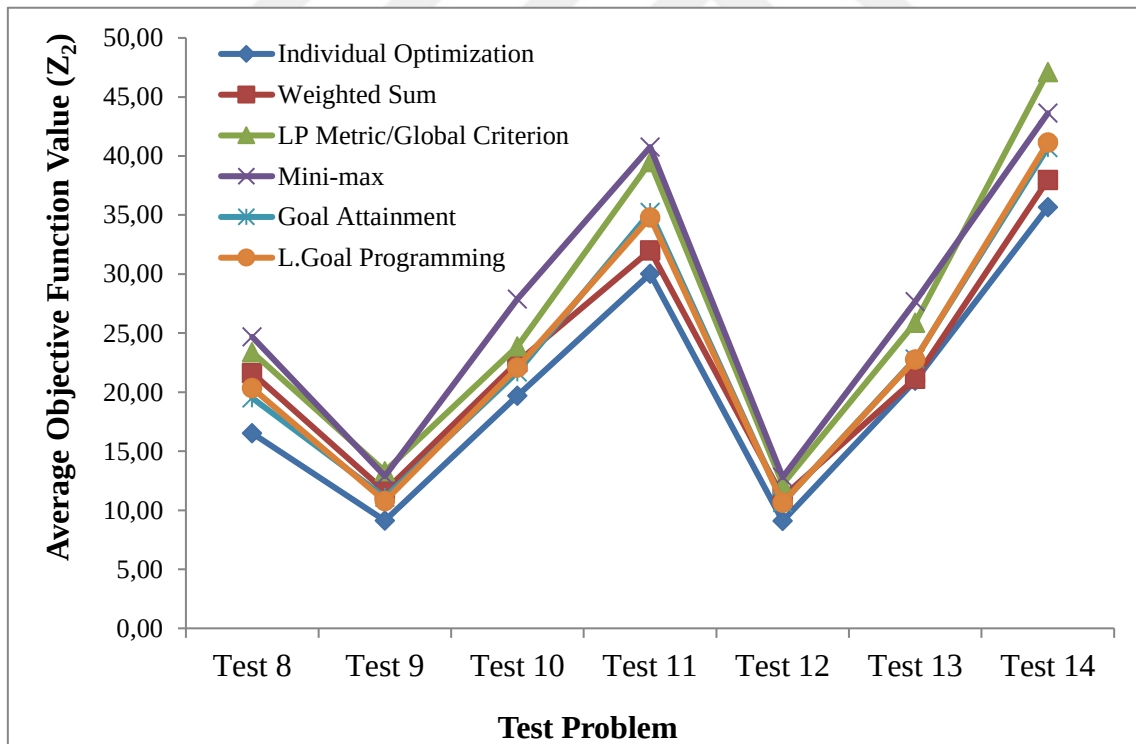
Table 5.43. The Results Obtained by the all MODM for Large Sized Test Problems
(Continuation)

Problem No	Criterion	(TCost)	(TC _{max})	Distance z1	Distance z2	Direct Distance
11	(A)	41392.074	30.011	0.000	0.000	0.000
	(B)	55827.980	32.002	0.349	0.066	0.415
	(C)	47490.892	39.473	0.147	0.315	0.463
	(D)	56152.714	40.770	0.357	0.359	0.715
	(E)	51118.247	35.223	0.235	0.174	0.409
	(F)	51629.829	34.794	0.247	0.159	0.407
12	(A)	21960.478	9.082	0.000	0.000	0.000
	(B)	26200.014	11.125	0.193	0.225	0.418
	(C)	25076.443	12.198	0.142	0.343	0.485
	(D)	30984.073	12.801	0.411	0.409	0.820
	(E)	27081.807	10.634	0.233	0.171	0.404
	(F)	26960.153	10.640	0.228	0.172	0.399
13	(A)	66400.705	20.965	0.000	0.000	0.000
	(B)	82377.268	21.137	0.241	0.008	0.249
	(C)	70791.941	25.880	0.066	0.234	0.301
	(D)	87670.489	27.662	0.320	0.319	0.640
	(E)	76562.707	22.804	0.153	0.088	0.241
	(F)	76546.843	22.747	0.153	0.085	0.238
14	(A)	92433.101	35.665	0.000	0.000	0.000
	(B)	115367.188	37.976	0.248	0.065	0.313
	(C)	99530.494	47.116	0.077	0.321	0.398
	(D)	120565.419	43.621	0.304	0.223	0.527
	(E)	108803.505	40.717	0.177	0.142	0.319
	(F)	105325.182	41.134	0.139	0.153	0.293

Fig. 5.14 shows the average objective function values of Z_1 and Z_2 for all large size test problems obtained employing all the MODM methods.



a)



b)

Figure 5.14. The Average Objective Function Values of a) Z_1 and b) Z_2 for Large Size Test Problems

Figure 5.15 shows for average direct distance values of MODM methods in large sized test problems

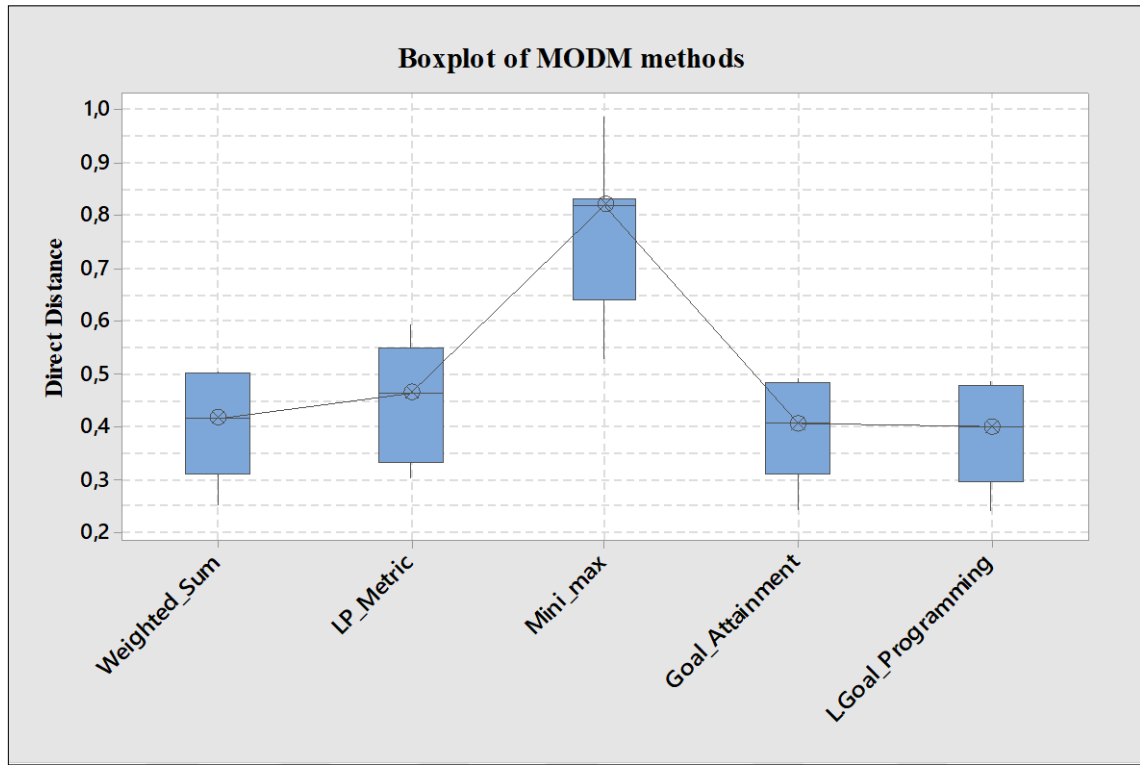


Figure 5.15. Boxplot for Average Direct Distance Values of MODM Methods

- It is obvious that from Table 5.43 and Figure 5.14, the *Lexicographic Goal Programming* method has the best performance on the large size test problems by presenting a solution which provides a satisfactory tradeoff between first and second objective functions value (Average direct distance value is about 0.372). Because, it has a great flexibility which provides the decision maker to easily incorporate many variations on goals and constraints.
- Goal Attainment and Weighted Sum Methods seem to have close performance to Lexicographic Goal Programming method. So, they can be used as alternative for solving large sized test problems (Average direct distance values are about 0.379 and 0.387, respectively).
- On the other hand, the Mini-max is not an effective method among the MODM methods for solving all size test problems (Average Direct Distance value is about 0.764).

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

6.1. Conclusions

In previous studies, integrated supply chain scheduling problems are studied generally in a single and parallel machine environment. The number of studies about integrated supply chain scheduling problem in a multi-level, multi-product, multi-machine and flexible routes environment is very limited. This thesis is one of the few works solving multi-objective supply chain scheduling problem in flexible job shop configuration.

MODM methods, especially min-max, goal programming and goal attainment methods which are applied relatively low when compared with other methods in previous studies. In this thesis, five MODM methods and NSGA-II were compared and successfully applied to solve supply chain scheduling problem. These methods can be used for solving of other problems of simultaneous optimization of integrated supply chain scheduling problems.

The nature of the supply chain management decisions involves distribution of raw materials from suppliers, manufacturing of products at factories (facilities), distribution of finished products to customers via warehouses, and the storage of raw materials and finished products at factories and warehouses. In this thesis, a multi period, multi-echelon mathematical model was developed for a supply chain network including manufacturing facilities, warehouses and customers. A F-JSS in manufacturing level of the supply chain was integrated to increase the applicability.

The problem was formulated as a multi-objective MILP model. In the model, two objective functions were simultaneously optimized. One objective function is to minimize the total cost of the supply chain which includes the production cost, inventory holding cost in manufacturing plant and warehouses, and backorder cost. Shortage (backorder) was considered in each period to make the model closer to the real-life conditions. The other objective function was to minimize the makespan in all

periods. Different size test problems were generated to illustrate the applicability of the developed model.

Various methods were applied to compare the computational performance because there exist trade-offs between the objectives. At first, five MODM methods and which includes "*Weighted Sum, LP Metric/Global Criterion, Maxi-min, Goal Attainment, and Goal programming*" were used in order to solve the problem. Then, as a MOEA "*Non-dominated Sorting Genetic Algorithm-II (NSGA-II)*" was also applied. Filtering/Displaced Ideal Solution (DIS) method was also applied to compare the methods in terms of three criteria; which are total cost and total makespan time ($\sum C_{max}$), and average CPU time.

Given test problems results and performance of the methods, several conclusions were drawn as follows:

- Many Pareto solutions were obtained by resolving the problem using different methods. Each solution obtained by methods is an effective solution for the optimization problem and provides helpful information for decision makers to select the most suitable solution.
- The computational results showed that the MODM methods have smaller execution times than NSGA-II. Because, NSGA-II searches more regions of solutions area, this higher computational time is expected.
- The results obtained by applying the DIS method show that the weighted sum method was the best among all investigated methods for small and medium size test problems. Because, CPU run time was relatively low compared to other methods. Thus, this method can be preferred to solve large-scale problems in reasonable time.
- F-JSS problem is strongly NP-hard. Thus, due to increase in constraints and variables in the F-JSS problem, it is difficult to solve the problem which shows a higher level of complexity. Thus, CPU time raises exponentially by increasing the size of the test problems.
- Other experiments were constructed for large-sized test problems. Results from all applied methods were compared and Lexicographic Goal Programming method was selected as the best method depending on the mentioned criteria and its detailed

computational results were presented. The computational results of the model has shown that this model is more favorable in solving large-scale problems.

6.2. Future Research Directions

Future research may extend into two directions: (1) the perspective of problem and (2) the solution method.

From the perspective of problem, some of the possible future research directions can be stated as follows:

- In order to reflect the real-life situations more effectively, various parameters of the mathematical model can be evaluated as fuzzy.
- New studies on applications of stochastic behaviors can be conducted.
- Researchers can handle the problem considering of lost sales or backorder and lost sale together.
- Researchers can reformulate the model using additional constraints related to maintenance activities, sequence-dependent set-up times and costs and etc. These constraints may increase the difficulty of the problem. So, significant changes may require for solving problem.
- A vehicle routing problem can be combined with supply chain scheduling problem in order to make the model closer to the real-life conditions.
- In addition, reverse logistic can be incorporated to model. Thus, recycling products can be used in the production cycle as raw materials for manufacturers.

From the perspective of solution methods, some of the possible future research directions can be stated as follows:

- In order to solve the problem, one can consider other multi objective evolutionary algorithms such as, Pareto Envelope-based Selection Algorithm II (PESA-II), Multi-Objective Particle Swarm Optimization (MOPSO), Strength Pareto Evolutionary Algorithm 2 (SPEA2) and etc. to find Pareto solutions. Thus, different Pareto solution sets can be obtained for the model.

REFERENCES

- Ahmadizar, F., & Farhadi, S. (2015). Single-machine batch delivery scheduling with job release dates, due windows and earliness, tardiness, holding and delivery costs. *Computers & Operations Research*, 53, 194-205.
- Anghileri, M., Chirwa, E. C., Lanzi, L., & Mentuccia, F. (2005). An inverse approach to identify the constitutive model parameters for crashworthiness modelling of composite structures. *Composite structures*, 68(1), 65-74.
- Assarzadegan, P., & Rasti-Barzoki, M. (2016). Minimizing sum of the due date assignment costs, maximum tardiness and distribution costs in a supply chain scheduling problem. *Applied Soft Computing*, 47, 343-356.
- Augusto, O. B., Bennis, F., & Caro, S. (2012). A new method for decision making in multi-objective optimization problems. *Pesquisa Operacional*, 32(2), 331-369.
- Beheshtinia, M. A., Ghasemi, A., & Farokhnia, M. (2018). Supply chain scheduling and routing in multi-site manufacturing system (case study: a drug manufacturing company). *Journal of Modelling in Management*, 13(1), 27-49.
- Bożek, A., & Wysocki, M. (2015). Flexible job shop with continuous material flow. *International Journal of Production Research*, 53(4), 1273-1290.
- Brucker, P., & Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing*, 45(4), 369-375.
- Chaudhry, I. A., & Khan, A. A. (2016). A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3), 551-591.
- Chen, Z. L., & Vairaktarakis, G. L. (2005). Integrated scheduling of production and distribution operations. *Management Science*, 51(4), 614-628.

Cheng, B., Pei, J., Li, K., & Pardalos, P. M. (2017). Integrated scheduling of production and distribution for manufacturers with parallel batching facilities. *Optimization Letters*, 1-15.

Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, 63(5), 912-942.

Choi, I. C., & Choi, D. S. (2002). A local search algorithm for job shop scheduling problems with alternative operations and sequence-dependent setups. *Computers & Industrial Engineering*, 42(1), 43-58

Coello, C. A. (2000). An updated survey of GA-based multi-objective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2), 109-143.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (Vol. 5). New York: Springer.

Çalışkan, E. (2014). Çabuk Bozulabilen Ürünlerde Bütünleşik Tedarik Zinciri Çizelgeleme Problemi için Karışık Tamsayılı Matematiksel Model ve Sezgisel Çözümü, (Doctoral dissertation, Gazi University).

Deb, K. (1999). Solving goal programming problems using multi-objective genetic algorithms. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 1, pp. 77-84). IEEE.

Deb, K., Member, A., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A Fast and Elitist Multi-objective Genetic Algorithm : 6, 182–197.

Demir, Y., & İşleyen, S. K. (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, 37(3), 977-988.

Devapriya, P. (2008). *Optimal fleet size of an integrated production and distribution scheduling problem for a single perishable product* (Doctoral dissertation, Clemson University).

- Devapriya, P., Ferrell, W., & Geismar, N. (2017). Integrated production and distribution scheduling with a perishable product. *European Journal of Operational Research*, 259(3), 906-916.
- Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2008). Strategies for multiobjective genetic algorithm development: Application to optimal batch plant design in process systems engineering. *Computers & Industrial Engineering*, 54(3), 539-569.
- Donoso, Y., & Fabregat, R. (2016). *Multi-objective optimization in computer networks using metaheuristics*. Auerbach Publications.
- Fallah, N., & Honarparast, S. (2013). NSGA-II based multi-objective optimization in design of Pall friction dampers. *Journal of Constructional Steel Research*, 89, 75-85.
- Fan, J., Lu, X., & Liu, P. (2015). Integrated scheduling of production and delivery on a single machine with availability constraint. *Theoretical Computer Science*, 562, 581-589.
- Fattahi, P., Mehrabad, M. S., & Jolai, F. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *Journal of intelligent manufacturing*, 18(3), 331-342.
- Gao, J., Gen, M., & Sun, L. (2006). Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. *Journal of Intelligent Manufacturing*, 17(4), 493-507.
- Gharaei, A., & Jolai, F. (2018). A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. *Applied Soft Computing*.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.
- Hall, N. G., & Potts, C. N. (2003). Supply Chain Scheduling: Batching and Delivery. *Operations Research*, 51(4), 566-584.

Hamidinia, A., Khakabimamaghani, S., Mazdeh, M. M., & Jafari, M. (2012). A genetic algorithm for minimizing total tardiness/earliness of weighted jobs in a batched delivery system. *Computers & Industrial Engineering*, 62(1), 29-38.

Hassanzadeh, A., Rasti-Barzoki, M., & Khosroshahi, H. (2016). Two new meta-heuristics for a bi-objective supply chain scheduling problem in flow-shop environment. *Applied Soft Computing*, 49, 335-351.

Hwang, C. L., & Masud, A. S. M. (2012). *Multiple objective decision making—methods and applications: a state-of-the-art survey* (Vol. 164). Springer Science & Business Media.

Imanipour, N. (2006, October). Modeling & solving flexible job shop problem with sequence dependent setup times. In *Service Systems and Service Management, 2006 International Conference on* (Vol. 2, pp. 1205-1210). IEEE.

Jalilvand-Nejad, A., & Fattahi, P. (2015). A mathematical model and genetic algorithm to cyclic flexible job shop scheduling problem. *Journal of Intelligent Manufacturing*, 26(6), 1085-1098.

Jones, D., & Tamiz, M. (2010). Goal programming variants. In *Practical goal programming* (pp. 11-22). Springer, Boston, MA.

Joo, C. M., & Kim, B. S. (2017). Rule-based meta-heuristics for integrated scheduling of unrelated parallel machines, batches, and heterogeneous delivery trucks. *Applied Soft Computing*, 53, 457-476.

Kang, H. Y., Pearn, W. L., Chung, I. P., & Lee, A. H. (2016). An enhanced model for the integrated production and transportation problem in a multiple vehicles environment. *Soft Computing*, 20(4), 1415-1435.

Karaoğlu, İ., & Kesen, S. E. (2017). The coordinated production and transportation scheduling problem with a time-sensitive product: a branch-and-cut algorithm. *International Journal of Production Research*, 55(2), 536-557.

Karimi, S., Ardalan, Z., Naderi, B., & Mohammadi, M. (2017). Scheduling flexible job-shops with transportation times: Mathematical models and a hybrid imperialist competitive algorithm. *Applied Mathematical Modelling*, 41, 667-682.

Kaucic, M. (2008). PhD Thesis, Evolutionary computation for trading systems, Universita degli Studi di Trieste.

Kazemi, H., Mazdeh, M. M., & Rostami, M. (2017). The two-stage assembly flow-shop scheduling problem with batching and delivery. *Engineering Applications of Artificial Intelligence*, 63, 98-107.

Khalilpourazari, S., & Khalilpourazary, S. (2017). A Lexicographic Weighted Tchebycheff Approach for Multi-constrained Multi-Objective Optimization of the Surface Grinding Process. *Engineering Optimization*, 49(5), 878–895.

Khalilpourazari, S., & Khamseh, A. A. (2017). Bi-objective Emergency Blood Supply Chain Network Design in Earthquake Considering Earthquake Magnitude: A Comprehensive Study with Real World Application. *Annals of Operations Research*, 1-39.

Kim, K. H., & Egbelu, P. J. (1999). Scheduling in a production environment with multiple process plans per job. *International Journal of Production Research*, 37(12), 2725-2753.

Lacomme, P., Moukrim, A., Quilliot, A., & Vinot, M. (2018). Supply chain optimisation with both production and transportation integration: multiple vehicles for a single perishable product. *International Journal of Production Research*, 1-24.

Lawrence, K. D., Klimberg, R. K., & Miori, V. M. (Eds.). (2010). *The Supply Chain in Manufacturing, Distribution, and Transportation: Modeling, Optimization, and Applications*. CRC press.

Lee, J., Kim, B. I., Johnson, A. L., & Lee, K. (2014). The nuclear medicine production and delivery problem. *European Journal of Operational Research*, 236(2), 461-472

Lightner, M., & Director, S. (1981). Multiple criterion optimization for the design of electronic circuits. *IEEE Transactions on Circuits and Systems*, 28(3), 169-179.

Lin, S., 2011. A NSGA-II Program in Matlab (NGPM 1.4)

Liu, J., & MacCarthy, B. L. (1997). A global MILP model for FMS scheduling. *European journal of operational research*, 100(3), 441-453

Liu, Y., Song, S., & Wang, C. (2017). Multi-objective optimization on the shock absorber design for the lunar probe using nondominated sorting genetic algorithm II. *International Journal of Advanced Robotic Systems*, 14(4).

Low, C., Yip, Y., & Wu, T. H. (2006). Modelling and heuristics of FMS scheduling with multiple objectives. *Computers & operations research*, 33(3), 674-694.

Marandi, F. (2017). A new approach in graph-based integrated production and distribution scheduling for perishable products. *Journal of Quality Engineering and Production Optimization*, 2(1), 65-76.

Mazdeh, M. M., & Rostami, M. (2014). A branch-and-bound algorithm for two-machine flow-shop scheduling problems with batch delivery costs. *International Journal of Systems Science: Operations & Logistics*, 1(2), 94-104.

Mazdeh, M. M., Hamidinia, A., & Karamouzian, A. (2011). A mathematical model for weighted tardy jobs scheduling problem with a batched delivery system. *International Journal of Industrial Engineering Computations*, 2(3), 491-498.

Mousakhani, M. (2013). Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. *International Journal of Production Research*, 51(12), 3476-3487.

Nandi, A. K., Deb, K., Ganguly, S., & Datta, S. (2012). Investigating the role of metallic fillers in particulate reinforced flexible mould material composites using evolutionary algorithms. *Applied Soft Computing*, 12(1), 28-39.

- Niu, Y., Tan, H., Liu, Y., Wang, X., & Xu, T. (2013). The effect of particle size and heating rate on pyrolysis of waste capsicum stalks biomass. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 35(17), 1663-1669.
- Noroozi, A., Mazdeh, M. M., Heydari, M., & Rasti-Barzoki, M. (2018). Coordinating order acceptance and integrated production-distribution scheduling with batch delivery considering Third Party Logistics distribution. *Journal of Manufacturing Systems*, 46, 29-45.
- Noroozi, A., Mazdeh, M. M., Noghondarian, K., Rasti-Barzoki, M., & Heydari, M. (2017). Evolutionary computation algorithms to coordinating order acceptance and batch delivery for an integrated supply chain scheduling. *Computational and Applied Mathematics*, 1-51.
- Oliveira, L. S. D., & Saramago, S. F. (2010). Multiobjective optimization techniques applied to engineering problems. *Journal of the brazilian society of mechanical sciences and engineering*, 32(1), 94-105.
- Özgüven, C., Özbakır, L., & Yavuz, Y. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling*, 34(6), 1539-1548.
- Pasandideh, S. H. R., Niaki, S. T. A., & Asadi, K. (2015). Optimizing a Bi-objective Multi-Product Multi-Period Three Echelon Supply Chain Network with Warehouse Reliability. *Expert Systems with Applications*, 42(5), 2615–2623.
- Pasandideh, S. H. R., Niaki, S. T. A., & Sharafzadeh, S. (2013). Optimizing a bi-objective multi-product EPQ model with defective items, rework and limited orders: NSGA-II and MOPSO algorithms. *Journal of Manufacturing Systems*, 32(4), 764-770.
- Pei, J., Liu, X., Pardalos, P. M., Fan, W., Yang, S., & Wang, L. (2014). Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain. *The International Journal of Advanced Manufacturing Technology*, 70(1-4), 335-348.

Pirim, H., Al-Turki, U., & Yilbas, B. S. (2014). *Supply Chain Management and Optimization in Manufacturing*. Springer.

Rabbani, M., Farrokhi-Asl, H., & Asgarian, B. (2017). Solving a bi-objective location routing problem by a NSGA-II combined with clustering approach: application in waste collection problem. *Journal of Industrial Engineering International*, 13(1), 13-27.

Rafiei, H., Safaei, F., & Rabbani, M. (2018). Integrated Production-Distribution Planning Problem in a Competition-Based Four-Echelon Supply Chain. *Computers & Industrial Engineering*.

Rasti-Barzoki, M., & Hejazi, S. R. (2013). Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. *European Journal of Operational Research*, 228(2), 345-357.

Rodriguez, A., Ordóñez, A., Ordoñez, H., & Segovia, R. (2015). Adapting NSGA-II for Hierarchical Sensor Networks in the IoT. *Procedia Computer Science*, 61, 355-360.

Roshanaei, V. (2012). Mathematical modelling and optimization of flexible job shops scheduling problem,” M.S. Thesis, Industrial and Manufacturing Systems Engineering, Windsor Univ., Ontario, Canada.

Rostami, M., Kheirandish, O., & Ansari, N. (2015). Minimizing maximum tardiness and delivery costs with batch delivery and job release times. *Applied Mathematical Modelling*, 39(16), 4909-4927.

Rostami, M., Nikravesh, S., & Shahin, M. (2018). Minimizing total weighted completion and batch delivery times with machine deterioration and learning effect: a case study from wax production. *Operational Research*, 1-33.

Sağlam, Ü., & Banerjee, A. (2018). Integrated multiproduct batch production and truck shipment scheduling under different shipping policies. *Omega*, 74, 70-81.

Sahraeian, R., Rohaninejad, M., & Fadavi, M. (2017). A New model for integrated lot sizing and scheduling in flexible job shop problem. *Journal of Industrial and Systems Engineering*, 10(3), 72-91.

Saidi-Mehrabad, M., & Fattahi, P. (2007). Flexible job shop scheduling with tabu search algorithms. *The International Journal of Advanced Manufacturing Technology*, 32(5-6), 563-570.

Sarrafha, K., Rahmati, S. H. A., Niaki, S. T. A., & Zaretalab, A. (2015). A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA. *Computers & Operations Research*, 54, 35-51.

Sawik, T. (2016). Integrated supply, production and distribution scheduling under disruption risks. *Omega*, 62, 131-144.

Shen, L., Dauzère-Pérès, S., & Neufeld, J. S. (2018). Solving the flexible job shop scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, 265(2), 503-516.

Shokouhi, E. (2018). Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints. *Production & Manufacturing Research*, 6(1), 61-89.

Stecke, K. E., & Zhao, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing & Service Operations Management*, 9(2), 206-224.

Steiner, G., & Zhang, R. (2009). Approximation algorithms for minimizing the total weighted number of late jobs with late deliveries in two-level supply chains. *Journal of Scheduling*, 12(6), 565.

Tamaki, H., Ono, T., Murao, H., & Kitamura, S. (2001, October). Modeling and genetic solution of a class of flexible job shop scheduling problems. In *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on* (Vol. 2, pp. 343-350). IEEE.

- Türkyılmaz, A., & Bulkan, S. (2015). A hybrid algorithm for total tardiness minimisation in flexible job shop: genetic algorithm with parallel VNS execution. *International Journal of Production Research*, 53(6), 1832-1848.
- Ullrich, C. A. (2013). Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research*, 227(1), 152-165.
- Viergutz, C., & Knust, S. (2014). Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1), 293-318.
- Vo-Duy, T., Duong-Gia, D., Ho-Huu, V., Vu-Do, H. C., & Nguyen-Thoi, T. (2017). Multi-objective optimization of laminated composite beam structures using NSGA-II algorithm. *Composite Structures*, 168, 498-509.
- Wang, Q., Batta, R., & Szczerba, R. J. (2005). Sequencing the processing of incoming mail to match an outbound truck delivery schedule. *Computers & operations research*, 32(7), 1777-1791.
- Yılmaz, Ö. F., & Pardalos, P. M. (2017). Minimizing average lead time for the coordinated scheduling problem in a two-stage supply chain with multiple customers and multiple manufacturers. *Computers & Industrial Engineering*, 114, 244-257.
- Zhang, Q., Manier, H., & Manier, M. A. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. *Computers & Operations Research*, 39(7), 1713-1723.

APPENDIX

Appendix 5-1. Weighted Sum Method with Different Weights for Large Sized Test Problems

Problem No	Criterion	Weighted Sum Method (Weights)			
		[0.8,0.2]	[0.6,0.4]	[0.4,0.6]	[0.2,0.8]
8	Distance (T_{Cost})	0.000	0.055	0.156	0.407
	Distance (TC_{max})	0.753	0.469	0.231	0.000
	Total Distance	0.753	0.524	0.387	0.407
9	Distance (T_{Cost})	0.000	0.088	0.207	0.456
	Distance (TC_{max})	0.944	0.482	0.226	0.000
	Total Distance	0.944	0.569	0.434	0.456
10	Distance (T_{Cost})	0.000	0.052	0.149	0.295
	Distance (TC_{max})	0.543	0.304	0.118	0.000
	Total Distance	0.543	0.356	0.267	0.295
11	Distance (T_{Cost})	0.000	0.048	0.156	0.318
	Distance (TC_{max})	0.745	0.455	0.167	0.000
	Total Distance	0.745	0.503	0.324	0.318
12	Distance (T_{Cost})	0.000	0.056	0.163	0.315
	Distance (TC_{max})	0.787	0.439	0.152	0.000
	Total Distance	0.787	0.495	0.314	0.315
13	Distance (T_{Cost})	0.000	0.037	0.107	0.229
	Distance (TC_{max})	0.506	0.275	0.129	0.000
	Total Distance	0.506	0.312	0.235	0.229
14	Distance (T_{Cost})	0.000	0.030	0.095	0.224
	Distance (TC_{max})	0.588	0.333	0.150	0.000
	Total Distance	0.588	0.363	0.246	0.224

Appendix 5.2. Goal Attainment Method with Different Weights for Large Sized Test Problems

Problem No	Criterion	Goal Attainment Method (Weights)			
		[0.8,0.2]	[0.6,0.4]	[0.4,0.6]	[0.2,0.8]
8	Distance (T_{Cost})	0.221	0.120	0.054	0.000
	Distance (TC_{max})	0.000	0.109	0.224	0.391
	Total Distance	0.221	0.229	0.277	0.391
9	Distance (T_{Cost})	0.187	0.126	0.055	0.000
	Distance (TC_{max})	0.000	0.090	0.205	0.381
	Total Distance	0.187	0.216	0.260	0.381
10	Distance (T_{Cost})	0.148	0.084	0.040	0.000
	Distance (TC_{max})	0.000	0.074	0.155	0.265
	Total Distance	0.148	0.158	0.195	0.265
11	Distance (T_{Cost})	0.166	0.101	0.048	0.000
	Distance (TC_{max})	0.000	0.095	0.216	0.373
	Total Distance	0.166	0.196	0.264	0.373
12	Distance (T_{Cost})	0.167	0.097	0.046	0.000
	Distance (TC_{max})	0.000	0.120	0.249	0.424
	Total Distance	0.167	0.216	0.295	0.424
13	Distance (T_{Cost})	0.112	0.066	0.030	0.000
	Distance (TC_{max})	0.000	0.067	0.135	0.232
	Total Distance	0.112	0.132	0.164	0.232
14	Distance (T_{Cost})	0.133	0.070	0.031	0.000
	Distance (TC_{max})	0.000	0.085	0.175	0.309
	Total Distance	0.133	0.155	0.207	0.309

CURRICULUM VITAE

Name and Surname : Zeynep CEYLAN
Marital Status : Married with 1 children
Place and Date of Birth : Şebinkarahisar / Giresun– 06.06.1987
Foreign Language : English
E-Mail : zeynep.dokumaci@omu.edu.tr



Education:

- **Master (M.Sc):** Marmara University, 2014 (Turkey), *Thesis Title* : Shelf Arrangement via Data Mining Techniques and A Case Study; *Supervisor* : Prof. Dr. S. Ümit OKTAY FIRAT
- **Undergraduate (B.Sc) :** Gaziantep University (Turkey), 2011; *Thesis Title* : Application Of Technology Forecasting with Data Envelopment Analysis (TFDEA) in Textile Industry; *Supervisor*: Prof. Dr. Türkay DERELİ

Job Experience:

Year	University	Department	Academic Title
10/2012 –	Ondokuz Mayıs University	Industrial Engineering	Res. Assistant
04/2011–09/2012	Istanbul Kemerburgaz University	Industrial Engineering	Res. Assistant

Articles:

Papers Published in International Refereed Journal

1. CEYLAN, Z., GÜRSEV, S., & BULKAN, S. (2017). An Application of Data Mining in Individual Pension Savings and Investment System. *European Journal of Science and Technology*, (Special Issue), 7-11.
2. CEYLAN, Z., BULKAN, S., & TOZAN, H. (2017). Tek ve Çok Dönemli Envanter Kontrol Modelleri. *Journal of Engineering Sciences and Design*, 5(2), 441-455.
3. CEYLAN, Z., & Pekel, E. (2017). Comparison of Multi-Label Classification Methods for Prediagnosis of Cervical Cancer. *International Journal of Intelligent Systems and Applications in Engineering*, 5(4), 232-236.

4. CEYLAN, Z., & BULKAN, S. Drug Inventory Management of a Pharmacy using ABC and VED Analysis. *Journal of Health Technology Assessment*, 2(1), 13-18.
5. CEYLAN, Z., Gürsev, S., & BULKAN, S. İki Aşamalı Kümeleme Analizi ile Bireysel Emeklilik Sektöründe Müşteri Profiline Değerlendirilmesi. *International Journal of Informatics Technologies*, 10(4), 475-485.
6. CEYLAN, Z., & FIRAT, S. Ü. O. (2017). A New Drug-Shelf Arrangement for Reducing Medication Errors using Data Mining: A Case Study. *Süleyman Demirel University Journal of Natural and Applied Sciences*, 21(3), 774-781.
7. CEYLAN, S., Topcu, Y., & CEYLAN, Z. (2014). Thermal behaviour and kinetics of alga *Polysiphonia elongata* biomass during pyrolysis. *Bioresource Technology*, 171, 193-198.

Proceedings

International Conference Proceedings

1. CEYLAN ZEYNEP, CEYLAN SELİM, BULKAN SEROL (2018). Estimation of Aviation Fuel Demand of Turkey by MLR And Grey Model. *The International Conference on Materials Science, Mechanical and Automotive Engineerings and Techonology*, İzmir.
2. CEYLAN ZEYNEP, BULKAN SEROL (2017). Assessment of Medical Waste Management: A Case Study in Istanbul. *International Conference on Computational and Statistical Methods in Applied Sciences*, Samsun.
3. CEYLAN ZEYNEP, BULKAN SEROL, TOZAN HAKAN (2017). An Application of Multi-Period Multi-Product Production Planning Model in Automotive Industry. *International Conference on Computational and Statistical Methods in Applied Sciences*, Samsun.
4. CEYLAN ZEYNEP, BULKAN SEROL (2017). The Grey Prediction GM (1,1) Model for Forecasting of Transportation-Related Energy Demand of Turkey. *International Conference on Computational and Experimental Science and Engineering*, Antalya.
5. CEYLAN ZEYNEP, BULKAN SEROL, TOZAN HAKAN (2017). Comparison of Multi-Objective Decision-Making Methods for Bi-Objective Integrated Supply

- Chain Network in Automotive Industry. *International Conference Computational and Experimental Science and Engineering*, Antalya.
6. PEKEL EBRU, CEYLAN ZEYNEP (2017). A Hybrid Decision Tree - Genetic Algorithm (DT-GA) Study for Diagnosis of Kidney Disease. XVIII. *Proceeding of XVth International Symposium on Econometrics, Operations Research and Statistics*, Trabzon.
 7. PEKEL EBRU, CEYLAN ZEYNEP (2017). Performance of Machine Learning Algorithms for Diagnosing Chronic Kidney Disease. XVIII. *Proceeding of XVth International Symposium on Econometrics, Operations Research and Statistics*, Trabzon.
 8. CEYLAN ZEYNEP, BULKAN SEROL (2017). Optimization of Integrated Production/Distribution and Inventory Planning Model for a Factory in Automotive Industry. *International Conference on Research in Education and Science*, İzmir.
 9. CEYLAN ZEYNEP, CEYLAN SELİM, BULKAN SEROL (2017). Assessing the Accuracy of Artificial Intelligence and MLR Models in Estimation of Road Transportation Energy Demand in Turkey. *International Conference on Research in Education and Science*, İzmir.
 10. CEYLAN ZEYNEP, BULKAN SEROL (2017). Analysis of Road Based Air Pollutant Levels by Various Artificial Intelligence Techniques for Predicting Air Quality. *International Conference on Research in Education and Science*, İzmir.
 11. CEYLAN ZEYNEP, GÜRSEV SAMET, BULKAN SEROL (2016). An application of Data Mining in Individual Pension Savings and Investment System. *3rd International Conference on Computational and Experimental Science and Engineering*, Antalya.
 12. CEYLAN ZEYNEP, CEYLAN SELİM (2016). Application of Artificial Neural Network in Modeling Co-Pyrolysis of Waste Tire Pistachios Shell. *3rd International Conference Computational and Experimental Science and Engineering*, Antalya.
 13. CEYLAN SELİM, CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT (2015). Forecasting CO₂ emissions related with fossil fuel consumption in Turkey. *27th European Conference on Operational Research*, Glasgow.

14. CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT, ÖZGÖNENEL OKAN (2015). Response Surface Optimization of an Artificial Neural Network for Predicting the CO₂ Emissions from Electricity Generation. *27th European Conference on Operational Research*, Glasgow.
15. CEYLAN ZEYNEP, BULKAN SEROL (2015). Inventory Analysis of a Pharmacy. *International Conference on Value Chain Sustainability (ICOVACS)*, Istanbul.
16. CEYLAN ZEYNEP, ÜLKÜ İLAYDA, VAYVAY ÖZALP (2015). An Analysis of Statistical Control Charts with Fuzzy Set Theory. *International Conference on Value Chain Sustainability (ICOVACS)*, İstanbul.
17. CEYLAN ZEYNEP, CEYLAN SELİM, TOPCU YILDIRAY (2014). Thermal Behaviour and Kinetics of Macroalga Biomass During Pyrolysis. *13th Mediterranean Congress of Chemical Engineering*, Barcelona.
18. CEYLAN ZEYNEP, DEMİRCİ NESLİHAN, BASKAK MURAT, FIRAT SENİYE ÜMİT (2014). An Optimization Model for Reverse Distribution Network of a Company. *The 12th International Logistics and Supply Chain Conference*, Istanbul.
19. CEYLAN ZEYNEP, KARHAN ZEHRA, FIRAT SENİYE ÜMİT, ERGEN BURHAN (2014). A Comparative Study on Diabetes Disease Diagnosis using Data Mining Classification Techniques. *International Conference on Applied Informatics for Health and Life Sciences*, Izmir.
20. CEYLAN SELİM, TOPCU YILDIRAY, CEYLAN ZEYNEP (2014). Thermokinetic Analysis of Pyrolysis of Aquatic Biomass Lignite Coal and Their Blends. *13th Mediterranean Congress of Chemical Engineering*, Barcelona.
21. CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT (2014). Patient Profile Determination for a Pharmacy via Data Mining Classification Techniques. *Proceeding of XVth International Symposium on Econometrics, Operations Research and Statistics*, Isparta.

National Conference Proceedings

1. CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT (2014). Veri Madenciliği Tekniklerini Kullanarak İş Kazalarının İstatistiksel Analizi Gıda Sektöründe Uygulama. *The 34th National Operations Research and Industrial Engineering (ORIE) Congress*, Bursa.

2. CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT (2014). Veri Madenciliği Birliktelik Kuralı Tekniğı ile Raf Düzenleme. 3. *National Logistics and Supply Chain Conference*, Trabzon.
3. Neslihan Demirci, CEYLAN ZEYNEP, FIRAT SENİYE ÜMİT (2012). Bulanık Çok Ölçütlü Karar Yöntemleri ile Hastanelerde Risk Değerlendirmesi. 18. *National Ergonomics Congress*, Gaziantep.

