

**T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**YÜKSEK LİSANS
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ PROGRAMI**

**(s,S) STOK KONTROL SİSTEMİNE İLİŞKİN
PARAMETRELERİN SİMÜLASYON TABANLI
OPTİMİZASYONU**

Eren YEŞİL

**Danışman
Dr. Öğr. Üyesi Özgür ESKİ**

**İkinci Danışman
Dr. Öğr. Üyesi Özlem UZUN ARAZ**

MANİSA-2024

Eren
YEŞİL

(s,S) STOK KONTROL SİSTEMİNE İLİŞKİN PARAMETRELERİN
SİMÜLASYON TABANLI OPTİMİZASYONU

2024

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Ana Bilim Dalında akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını, tamamen kendi çalışmam olduğunu, her alıntıya kaynak gösterdiğimi, tezin yazımında akademik ve etik kurallara aykırı herhangi bir yapay zeka ve program kullanmadığımı beyan ederim.

Eren YEŞİL



ÖZET

Yüksek Lisans Tezi

(s,S) Stok Kontrol Sistemine İlişkin Parametrelerin Simülasyon Tabanlı Optimizasyonu

Eren YEŞİL

Manisa Celal Bayar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Özgür ESKİ

İkinci Danışman: Dr. Öğr. Üyesi Özlem UZUN ARAZ

Üretim yapan firmalarda stoklar firmanın maliyetlerinde önemli bir etkiye sahiptir. Firmalar karlılıklarını artırmak, maliyetlerini azaltmak için kendi kısıtlarına uygun stok kontrol politikası belirleyebilirler. Firmalar etkili stok kontrolü ile üretimlerini aksatmadan ve aşırı stok tutmadan üretimlerini devam ettirmenin yanı sıra maliyetlerini azaltabilir, maliyet raporları için düzenli bir şekilde stok dengesini sağlayabilirler. Bu çalışmada üretim yapan bir işletmede, tek kalemli stok kontrol modeli simülasyon yöntemi ile stokastik olarak modellenmiştir. Simülasyon programıyla entegre çalışan genetik algoritma, benzetimli tavlama algoritması ve simülasyon optimizasyonu aracı kullanarak optimum toplam stok maliyetini veren stok kontrol parametreleri belirlenmeye çalışılmıştır ve sonuçlar karşılaştırılmıştır. Genetik algoritma ve benzetimli tavlama algoritmalarında kullanılan parametreler algoritmanın etkin çalışması için önem arz etmektedir. Bu nedenle genetik algoritma ve benzetimli tavlama algoritmalarının performansını geliştirebilmek için parametreler ile bir dizi deney yapılmıştır. Stok maliyetinin farklı durumlar altında nasıl etkilendiğinin incelenebilmesi için program sipariş miktarı, tedarik süresi ve siparişlerin gelişler arası süresi faktörlerinin farklı seviyeleri için çalıştırılmış ve elde edilen toplam stok maliyeti değerleri karşılaştırılmıştır. Sonrasında bu faktörlerin etkisinin incelenebilmesi için karşılaştırmalı deneyler yapılmıştır.

Anahtar Kelimeler: Stok Kontrol, Simülasyon, Simülasyon Optimizasyonu, Genetik Algoritma, Benzetimli Tavlama

2024, 131 sayfa

ABSTRACT

M.Sc. Thesis

Simulation-Based Optimization of Parameters in (s,S) Inventory Control System

Eren YEŞİL

**Manisa Celal Bayar University
Graduate School of Education
Department of Industrial Engineering**

Supervisor: Assist. Prof. Dr. Özgür ESKİ

Co-Supervisor: Assist. Prof. Dr. Özlem UZUN ARAZ

Inventories in manufacturing companies have a significant impact on the costs of the company. Companies can determine an inventory control policy appropriate to their constraints in order to increase their profitability and reduce their costs. With effective stock control, companies can continue their production without interruption and without holding excessive stock, reduce their costs, and regularly balance their stock for cost reports. In this study, a single-item inventory control model was stochastically modelled with the simulation method in a manufacturing company. Using a genetic algorithm, simulated annealing algorithm, and simulation optimisation tool integrated with the simulation program, the inventory control parameters that give the optimum total inventory cost were tried to be determined, and the results were compared. The parameters used in genetic algorithms and simulated annealing algorithms are essential for the effective operation of the algorithm. Therefore, a series of experiments were conducted with parameters to improve the performance of genetic algorithm and simulated annealing algorithms. In order to examine how inventory costs are affected under different conditions, the program was run for varying levels of order quantity, lead time and time between arrivals, and the total inventory cost values obtained were compared. Afterwards, comparative experiments were conducted to examine the effects of these factors.

Keywords: Inventory Control, Simulation, Simulation Optimization, Genetic Algorithms, Simulated Annealing.

2024, 131 pages

TEŞEKKÜR

Araştırmalarım için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli danışman hocalarım Dr. Öğr. Üyesi Özgür ESKİ ve Dr. Öğr. Üyesi Özlem UZUN ARAZ'a teşekkürlerimi sunarım.

Ayrıca yardımlarından dolayı Dr. Öğr. Üyesi M. Emre ŞENOL'a teşekkürlerimi iletmek isterim.

Beni maddi ve manevi olarak destekleyen, sadece bu süreçte değil, her zaman yanımda olduğunu hissettiğim değerli aileme sonsuz sevgi ve saygılarımı sunarım.

Eren YEŞİL
Manisa, 2024



SİMGELER VE KISALTMALAR DİZİNİ

a	Anlamlılık Düzeyi
B	Ertelenmiş Sipariş (Backorder)
c	Birim Maliyet
Ç	Çevrim Süresi
d	Yıllık Talep
DT	Deterministik
ESM	Ekonomik Sipariş Miktarı
f	Uygunluk Fonksiyonu
GA	Genetik Algoritma (Genetic Algorithm)
h	Elde Bulundurma Maliyeti
I₀	Sipariş Edilmiş Henüz Ulaşmamış Ürün Miktarı
I_L	Eldeki Stok Miktarı
I_p	Stok seviyesi (Inventory Position)
i	Yıllık Faiz Oranı
K	Sabit Maliyet (Sipariş Maliyeti)
N	Kromozom Sayısı
n	Simülasyondaki Çevrim Sayısı
p(i)	i. Bireyin Seçim Olasılığı
P_c	Çaprazlama Olasılığı
P_m	Mutasyon Olasılığı
Q	Sipariş Miktarı
R	İnceleme Aralığı
r	Rastgele (Random) Sayı
s	Yeniden Sipariş Verme Noktası
S	Maksimum Stok Seviyesi
SA	Benzetimli Tavlama (Simulated Annealing)
ST	Stokastik
T	Sıcaklık
t	İterasyon Sayısı
T_k	k'nci iterasyondaki sıcaklık değeri
TSM	Toplam Stok Maliyeti

v	Değişken Maliyet
vb.	Ve benzeri
w	Örnek Yolun Boyutu
α	Soğutma Oranı
ΔE	İki Çözümün Amaç Fonksiyonundaki Değişim



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1 Pareto eğrisi [37].....	19
Şekil 3.2 Ekonomik sipariş miktarı modeli [50]	25
Şekil 3.3 Maliyet değişimi grafiği [50].....	26
Şekil 3.4 (s,Q) sistemi [50]	30
Şekil 3.5 (s,S) sistemi [50].....	31
Şekil 3.6 (R,S) sistemi [50].....	33
Şekil 3.7 (R,s,S) sistemi [50]	34
Şekil 5.1 Genetik algoritma çalışma prensibi [67].....	43
Şekil 5.2 Çaprazlama operatörleri [76].....	49
Şekil 5.3 Benzetimli tavlama algoritmasının yapısı [96].....	52
Şekil 6.1 Sipariş miktarları için zaman serisi grafiği	58
Şekil 6.2 Sipariş miktarları için otokorelasyon grafiği	59
Şekil 6.3 Sipariş miktarı dağılım grafiği.....	60
Şekil 6.4 Tedarik süreleri için zaman serisi grafiği	61
Şekil 6.5 Tedarik süreleri için otokorelasyon grafiği.....	62
Şekil 6.6 Tedarik süresi dağılım grafiği.....	63
Şekil 6.7 Model değişkenleri	66
Şekil 6.8 Modelin istatistiksel çıktı değerleri	67
Şekil 6.9 (s,S) Stok kontrol politikası arena modeli ekran görüntüsü	69
Şekil 6.10 OptQuest ile simülasyon tabanlı optimizasyon sonuçları.....	71
Şekil 6.11 OptQuest iterasyon maliyet grafiği.....	72
Şekil 6.12 GA ve SA tabanlı simülasyon optimizasyonu yönteminin akış şeması [86]	73
Şekil 6.13 Kromozom yapısı.....	74
Şekil 6.14 Çok noktalı çaprazlama yöntemi	76
Şekil 6.15 Mutasyon	77
Şekil 6.16 Rastgele başlangıç popülasyonunun oluşturulması	78
Şekil 6.17 Rulet çarkı yöntemi ile seçilen bireyler	78
Şekil 6.18 Çok noktalı çaprazlama işlemi.....	79
Şekil 6.19 Mutasyon işlemi	79
Şekil 6.20 Popülasyon büyüklükleri için maliyet karşılaştırılması.....	82
Şekil 6.21 (1904, 1997) parametreleri için jenerasyon-maliyet grafiği	82
Şekil 6.22 Çaprazlama olasılıkları için maliyet karşılaştırılması	85
Şekil 6.23 (1904,1997) parametreleri için jenerasyon-maliyet grafiği	86
Şekil 6.24 Mutasyon olasılıkları için maliyet karşılaştırılması.....	89
Şekil 6.25 (1905, 1988) parametreleri için jenerasyon-maliyet grafiği	90
Şekil 6.26 Benzetimli tavlama algoritması için başlangıç çözümü	94
Şekil 6.27 Komşu çözüm	94
Şekil 6.28 Başlangıç sıcaklıkları için maliyet karşılaştırılması	97
Şekil 6.29 (1907, 1997) parametreleri için iterasyon-maliyet grafiği.....	98
Şekil 6.30 Soğutma oranı için maliyet karşılaştırılması	101
Şekil 6.31 (1903, 1996) parametreleri için iterasyon-maliyet grafiği.....	101
Şekil 6.32 Soğutma oranları için maliyet karşılaştırılması	104
Şekil 6.33 (1904, 1997) parametreleri için iterasyon-maliyet grafiği.....	104
Şekil 6.34 Optimum çözümlerin mevcut durumla karşılaştırma grafiği.....	106
Şekil 6.35 Düşük, orta ve yüksek sipariş miktarı büyüklükleri için yöntemlerin karşılaştırılması	108

Şekil 6.36 Uzun, orta, kısa tedarik süresi büyüklükleri için yöntemlerin karşılaştırılması	110
Şekil 6.37 Siparişlerin kısa, orta ve uzun gelişler arası süresi büyüklükleri için yöntemlerin karşılaştırılması.....	113
Şekil 6.38 Anova çıktısı.....	115
Şekil 6.39 Maliyet etkileşim grafiği.....	116



TABLolar DİZİNİ

	Sayfa
Tablo 2.1 Matematiksel modeller kullanılarak yapılan çalışmalar	6
Tablo 2.2 Sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu veya simülasyon modelleme kullanılarak yapılan çalışmalar	11
Tablo 3.1 Stok kontrol politikalarının karşılaştırılması [53]	34
Tablo 5.1 Genetik algoritma terimleri [68]	44
Tablo 5.2 Tavlama işlemi ile optimizasyon probleminin eşleşmesi [83]	53
Tablo 6.1 80'lik popülasyon büyüklüğü için elde edilen sonuçlar	80
Tablo 6.2 100'lük popülasyon büyüklüğü için elde edilen sonuçlar	81
Tablo 6.3 120'lik popülasyon büyüklüğü için elde edilen sonuçlar	81
Tablo 6.4 Popülasyon büyüklükleri için maliyet karşılaştırması	81
Tablo 6.5 0,70'lik çaprazlama olasılığı için elde edilen sonuçlar	83
Tablo 6.6 0,75'lik çaprazlama olasılığı için elde edilen sonuçlar	84
Tablo 6.7 0,80'lik çaprazlama olasılığı için elde edilen sonuçlar	84
Tablo 6.8 0,85'lik çaprazlama olasılığı için elde edilen sonuçlar	85
Tablo 6.9 Çaprazlama olasılıkları için maliyet karşılaştırması	85
Tablo 6.10 0,001'lik mutasyon olasılığı için elde edilen sonuçlar	87
Tablo 6.11 0,03'lük mutasyon olasılığı için elde edilen sonuçlar	87
Tablo 6.12 0,06'lük mutasyon olasılığı için elde edilen sonuçlar	88
Tablo 6.13 0,01'lik mutasyon olasılığı için elde edilen sonuçlar	88
Tablo 6.14 Mutasyon olasılıkları için maliyet karşılaştırması	89
Tablo 6.15 5.000 başlangıç sıcaklığı için elde edilen sonuçlar	96
Tablo 6.16 7.000 başlangıç sıcaklığı için elde edilen sonuçlar	96
Tablo 6.17 10.000 başlangıç sıcaklığı için elde edilen sonuçlar	97
Tablo 6.18 Başlangıç sıcaklıkları için maliyet karşılaştırması	97
Tablo 6.19 0,4'lük soğutma oranı için elde edilen sonuçlar	99
Tablo 6.20 0,7'lik soğutma oranı için elde edilen sonuçlar	99
Tablo 6.21 0,85'lik soğutma oranı için elde edilen sonuçlar	100
Tablo 6.22 Soğutma oranları için maliyet karşılaştırması	100
Tablo 6.23 500'lük iterasyon büyüklüğü için elde edilen sonuçlar	102
Tablo 6.24 1.000'lik iterasyon büyüklüğü için elde edilen sonuçlar	103
Tablo 6.25 1.500'lük iterasyon büyüklüğü için elde edilen sonuçlar	103
Tablo 6.26 İterasyon büyüklükleri için maliyet karşılaştırması	103
Tablo 6.27 Optimum çözümler ile bulunan ve mevcut durumdaki stok maliyeti ...	105
Tablo 6.28 Sipariş miktarı büyüklüğü, yöntem ve maliyet tablosu	109
Tablo 6.29 Tedarik süresi büyüklüğü, yöntem ve maliyet tablosu	111
Tablo 6.30 Siparişlerin gelişler arası süresi büyüklüğü, yöntem ve maliyet tablosu	114

İÇİNDEKİLER

ÖZET.....	I
ABSTRACT	II
TEŞEKKÜR.....	III
SİMGELER VE KISALTMALAR DİZİNİ	IV
ŞEKİLLER DİZİNİ.....	VI
TABLOLAR DİZİNİ	VIII
İÇİNDEKİLER	IX
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	4
2.1 Matematiksel Modeller Kullanılarak Yapılan Çalışmalar	4
2.2 Sezgisel veya Metasezgisel Algoritma Tabanlı Simülasyon Optimizasyonu veya Simülasyon Modelleme Kullanılarak Yapılan Çalışmalar.....	7
3. STOK YÖNETİMİ	15
3.1 Stokların Sınıflandırılması	17
3.1.1 Stok Türüne Göre Sınıflandırma.....	17
3.1.2 Stok İşlevine Göre Sınıflandırma.....	18
3.1.3 ABC Analizine Göre Sınıflandırma.....	18
3.2 Stok Maliyetleri.....	20
3.2.1 Elde Bulundurma Maliyeti.....	20
3.2.2 Sipariş Verme ve Üretim Hazırlık Maliyeti	21
3.2.3 Elde Bulundurmama Maliyeti	22
3.3 Stok Kontrolü	22
3.3.1 Deterministik Stok Kontrol	24
3.3.2 Stokastik Stok Kontrol	27
3.3.2.1 Sürekli Gözden Geçirme Sistemleri.....	28
3.3.2.1.1 (s,Q) Sistemi.....	29
3.3.2.1.2 (s,S) Sistemi	30
3.3.2.2 Periyodik Gözden Geçirme Sistemleri.....	31
3.3.2.2.1 (R,S) Sistemi	32
3.3.2.2.2 (R,s,S) Sistemi.....	33
4. SİMÜLASYON VE STOK KONTROLÜ.....	36
4.1 Simülasyon Yönteminin Çeşitleri	36
4.2 Simülasyon Yöntemi Kullanmanın Avantajları ve Dezavantajları.....	37
4.3 Stok Kontrolünde Simülasyon	38
5. SİMÜLASYON OPTİMİZASYONU	40
5.1 Simülasyon Optimizasyonu için Klasik Yaklaşımlar	40

5.2 Simülasyon Optimizasyonu için Metasezgisel Yaklaşımlar	41
5.2.1 Genetik Algoritma.....	42
5.2.1.1 Parametre Kodlama.....	44
5.2.1.2 Başlangıç Popülasyonunun Oluşturulması.....	45
5.2.1.3 Uygunluk Fonksiyonu.....	45
5.2.1.4 Genetik Operatörler.....	45
5.2.1.4.1 Seçim.....	45
5.2.1.4.2 Çaprazlama.....	48
5.2.1.4.3 Mutasyon.....	49
5.2.1.5 Durdurma Kriteri.....	50
5.2.2 Benzetimli Tavlama	51
5.2.2.1 Komşuluk Belirleme	53
5.2.2.2 Başlangıç Sıcaklığı Belirleme	53
5.2.2.3 Soğutma Planı	54
5.2.2.4 Kabul Fonksiyonu	54
5.2.2.5 Durdurma Kriteri.....	55
6. STOK KONTROL PARAMETRELERİNİN BELİRLENMESİ İÇİN GENETİK ALGORİTMA VE BENZETİMLİ TAVLAMA TABANLI BİR SİMÜLASYON OPTİMİZASYON UYGULAMASI.....	56
6.1 Problem Tanımı ve Varsayımlar	56
6.2 Girdi Analizi.....	58
6.3 Maliyet Hesaplamaları	64
6.4 Simülasyon Modelinin Oluşturulması	66
6.5 Arena OptQuest ile Simülasyon Tabanlı Optimizasyon	70
6.6. Genetik Algoritma ve Benzetimli Tavlama Algoritması Tabanlı Simülasyon Optimizasyonu Yönteminin Oluşturulması.....	72
6.7 Genetik Algoritma Tabanlı Simülasyon Optimizasyonu Algoritmasının Kurulması.....	73
6.7.1 Parametrelerin Belirlenmesi.....	74
6.7.1.1 Kromozom Temsili	74
6.7.1.2 Başlangıç Popülasyonu Oluşturma ve Popülasyon Büyüklüğü	74
6.7.1.3 Uygunluk Fonksiyonu.....	75
6.7.1.4 Seçim Operatörü.....	75
6.7.1.5 Çaprazlama Operatörü	76
6.7.1.6 Mutasyon Operatörü.....	77
6.7.1.7 Sonlandırma Operatörü	77
6.7.2 Genetik Algoritma ile İlgili Sayısal Örnek	78
6.7.3 Genetik Algoritma Operatörlerinin Çözüm Performansına Etkisi.....	79

6.7.3.1 Popülasyon Büyüklüğü	80
6.7.3.2 Çaprazlama Operatörü	83
6.7.3.3 Mutasyon Operatörü.....	87
6.8 Benzetimli Tavlama Tabanlı Simülasyon Optimizasyonu Algoritmasının Kurulması.....	91
6.8.1 Benzetimli Tavlama Algoritması Parametreleri.....	91
6.8.1.1 Başlangıç Sıcaklığı.....	92
6.8.1.2 Soğutma Planı	92
6.8.1.3 Kabul Fonksiyonu	93
6.8.1.4 Durdurma Kriteri.....	94
6.8.2 Benzetimli Tavlama Algoritması ile İlgili Sayısal Örnek.....	94
6.8.3 Benzetimli Tavlama Algoritması Operatörlerinin Çözüm Performansına Etkisi	95
6.8.3.1 Başlangıç Sıcaklığı.....	95
6.8.3.2 Soğutma Oranı Parametresi.....	99
6.8.3.3 Durdurma Kriteri.....	102
6.9 Genetik Algoritma, Benzetimli Tavlama Algoritması ve OptQuest ile Bulunan Sonuçların ve Mevcut Durumun Karşılaştırılması.....	105
6.10 Faktör Analizi.....	107
6.10.1 Sipariş Miktarı.....	107
6.10.2 Tedarik Süresi	110
6.10.3 Siparişlerin Gelişler Arası Süresi	112
6.11 Sipariş Miktarı, Tedarik süresi ve Siparişlerin Gelişler Arası Süresi Etki Analizi	115
7. SONUÇLAR VE ÖNERİLER	118
KAYNAKÇA.....	120
EKLER.....	128
EK 6.1 Sipariş Miktarları İçin Rassal Örneklem Tablosu	128
EK 6.2 Tedarik Süreleri İçin Rassal Örneklem Tablosu.....	129
EK 6.3 OptQuest'te Kontrol Seçim Ekranı.....	130
EK 6.4 OptQuest'teki Kısıtlar Penceresi (1).....	130
EK 6.5 OptQuest'teki Kısıtlar Penceresi (2).....	131
EK 6.6 OptQuest'teki Amaç ve Gereksinim Seçim Penceresi	131

1. GİRİŞ

İşletmeler üretimin devamlılığının ve verimliliğinin sağlanabilmesi, müşteri siparişlerinin zamanında karşılanabilmesi amacıyla stok yönetimi kullanmaktadır. Stok yönetimi en az maliyetle yeterli stoğu tutmak amacıyla ürünlerin en uygun maliyetle tedarik edilmesi ve bulundurulmasını sağlayan yönetim faaliyetidir. Stok yönetimi, müşteri taleplerini karşılamak ve üretim süreçlerinin kesintisiz devam etmesini sağlamak için kritik öneme sahiptir. Günümüz iş dünyasında, etkili stok yönetimi, işletmelerin rekabet gücünü koruyabilmeleri ve sürdürülebilir olmaları açısından önemli bir rol oynamaktadır. Bu nedenle, stok seviyelerinin optimum düzeyde tutulması, maliyetlerin minimize edilmesi ve hizmet kalitesinin artırılması açısından büyük bir etkiye sahiptir. Etkin stok yönetimi ile stok kontrolleri doğru şekilde uygulanarak, tüketici beklentileri ve müşteri taleplerinin uygun şekilde karşılanması sonucunda müşteri memnuniyeti artmaktadır. Bu durum işletmelere rekabet avantajı sağlamak ve itibar kazandırmaktadır.

Değişen müşteri talepleri ve davranışları, tedarikçi ilişkileri ve mesafesi, mevcut sermaye, firmanın ekonomik durumu ve depo kapasitesi belirlenecek stok yönetimini doğrudan etkilemektedir. Bu nedenle stok yönetimi her işletmede farklılık göstermektedir ve her işletme kendi ihtiyaçlarına göre etkin bir stok kontrol yönetimi geliştirmelidir. İşletmeler kendilerine uygun bir stok kontrol yöntemi uygulamadıklarında, müşteri memnuniyetinin düşmesine ve finansal açıdan önemli olan siparişlerin karşılanamamasına veya fazla stok tutulmasına neden olmaktadır. Ayrıca çoğu işletmenin sahip olduğu stok maliyetlerini en aza indirmek önemli sorunlardandır. Belirtilen nedenlerden dolayı her işletme kendi yapısına uygun bir stok kontrol yöntemi saptamalıdır.

Stok kontrolü için literatürde çeşitli yöntemler bulunmaktadır. Genel olarak, stok kontrolü talebin veya tedarik süresinin kesin olarak bilindiği deterministik stok kontrolü ve talebin veya tedarik süresinin rastgele kabul edildiği, çoğu kez bir olasılık dağılımı yardımıyla ifade edilebildiği stokastik stok kontrolü olmak üzere 2 başlık altında incelenmektedir. Stokastik stok kontrolü talebin ve tedarik süresinin rastgele belirlenmesi nedeniyle gerçek hayat problemlerine daha uygundur.

Stok kontrol modelinin geliştirilmesi için çoğunlukla kullanılan yaklaşımlar matematiksel programlama ve simülasyon modellemesidir. Matematiksel programlama, matematiksel modellerin geliştirilmesi için doğrusal programlama, dinamik programlama, hedef programlama vb. temellere dayanmaktadır [1]. Ancak gerçek hayat stok kontrol modelleri için matematiksel model geliştirme, modelin gerçekçiliğinden uzak varsayımlar belirlenmesini ve sadeleştirme yapılmasını gerektirmektedir

Stok kontrol yönetimi problemlerinde simülasyon modellemenin kullanılması, matematiksel modellemeye göre uygun bir alternatiftir. Çünkü simülasyon modelleme ile analitik forma getirilmesi zor, çok değişkenli, doğrusal olmayan değişkenler tanımlanabilmektedir ve simülasyon daha gerçekçi sonuçlar üretmektedir. Ancak simülasyon tek başına bir optimizasyon tekniği değildir. Optimizasyon probleminin çözümü için simülasyon modeliyle optimizasyon aracının entegre edilmesi gereklidir. Optimizasyon aracının performansı, modelin karmaşıklığı ve kullanılan parametrelere bağlı olarak değişkenlik gösterebilir. Bu nedenle, farklı optimizasyon araçlarının sonuçlarının karşılaştırılması, hangi yöntemin belirli koşullar altında daha etkili olduğunu belirlemek için önem arz etmektedir.

Bu çalışmanın genel amacı, toplam stok maliyetini minimize eden optimum (s,S) değerlerinin bulunmasıdır. Bu amaç doğrultusunda işletmenin stok kontrol sistemini açıklayan ayrıntılı bir simülasyon modeli geliştirilmiştir. Simülasyon optimizasyonu aracıyla, geliştirilen stok kontrol modelinin optimum maliyeti veren parametreleri hesaplanmıştır. Daha sonra stok kontrol politikasının optimizasyonu için modele genetik algoritma (GA) ve benzetimli tavlama algoritması (SA) entegre edilmiştir. Geliştirilen GA'nın performansını iyileştirmek için GA parametrelerine (yani popülasyonun büyüklüğü, çaprazlama olasılığı ve mutasyon olasılığı) uygun değerleri belirlemek için denemeler yapılmıştır. Geliştirilen diğer algoritma SA'nın performansını iyileştirmek için SA parametrelerine (başlangıç sıcaklığı, soğutma oranı ve durdurma kriteri) uygun değerleri belirlemek için denemeler yapılmıştır. Her üç çözüm yöntemi ile bulunan sonuçlar karşılaştırılmış ve çözüm yöntemlerinin performansı incelenmiştir. Ayrıca çalışmada tedarik süresi, sipariş miktarı ve siparişlerin gelişler arası süresi faktörlerinin farklı seviyelerinin toplam stok maliyeti üzerindeki etkisi GA, SA, OptQuest sonuçları ile karşılaştırılarak incelenmiştir. Bu

incelemeler, işletmenin stok yönetiminde daha uygun kararlar almasına yardımcı olacak bilgiler sunmaktadır. Örneğin, uzun tedarik sürelerinin maliyet üzerindeki olumsuz etkilerini minimize etmek için stok seviyelerinin nasıl ayarlanması gerektiği veya büyük siparişlerin stok maliyetlerini nasıl etkilediği gibi sorulara yanıt aranmıştır.

Bu tezde 7 bölüm bulunmaktadır. İlk bölümde çalışmada incelenecek olan konuya giriş yapılmıştır. İkinci bölümde literatür araştırmasına yer verilmiştir ve bu çalışmanın literatüre katkısı belirtilmiştir. Üçüncü bölümde stoklar, stok çeşitleri ve maliyetleri, deterministik ve stokastik stok kontrolleri ve çeşitleri açıklanmıştır. Dördüncü bölümde simülasyon yöntemi açıklanmış, simülasyon yönteminin stok kontrolünde uygulanmasının uygunluğu belirtilmiştir. Beşinci bölümde simülasyon optimizasyonu konusu anlatılmış ve çalışmada kullanılan genetik algoritma ve benzetimli tavlama algoritması detaylı olarak anlatılmıştır. Altıncı bölümde ise gerçek bir işletmenin verileri kullanılarak model simülasyon yöntemi ile modellenmiş ve simülasyon optimizasyonu yaklaşımı ile çözülmüştür. Son olarak yedinci bölümde çalışmanın sonuçları yorumlanmış, bulgular özetlenmiş ve gelecek çalışma konularına değinilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Stok kontrol yönetimi, işletmelerin maliyetlerini azaltmada büyük bir potansiyele sahip olması nedeniyle araştırmacılar tarafından dikkat çekici olmuştur. Optimum stok kontrol parametrelerini belirlemek ve stok maliyetini minimize etmek amacıyla matematiksel modeller kullanarak çalışmalar yapılmıştır. İşletmelerin uyguladığı stok kontrol politikalarının, karmaşık gerçek hayat problemlerine uyarlanabilmesi için simülasyon modelleme ve aynı zamanda sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu kullanıldığı optimizasyon çalışmaları mevcuttur.

Bu tezde, literatüre ait incelenen çalışmalar (s,S) stok kontrol politikasında optimum stok kontrol parametrelerinin bulunmasına yönelik matematiksel model ile yapılan çalışmalar ve gerçek hayat problemleri için daha gerçekçi ve anlamlı çözümler sunan sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu veya simülasyon modelleme ile yapılan çalışmalar olarak iki başlık altında incelenmiştir.

2.1 Matematiksel Modeller Kullanılarak Yapılan Çalışmalar

Bu tezde uygulanan yöntem metasezgisel algoritma tabanlı simülasyon optimizasyonu yöntemi olduğu için ve literatürde matematiksel modeller kullanarak yapılan çok fazla çalışma bulunması nedeniyle literatürde, 2019 ve 2024 yılları arasındaki, matematiksel model kullanılarak sürekli gözden geçirme (s,S) stok kontrol politikasında optimum stok kontrol parametrelerinin bulunmasına yönelik çalışmalar incelenmiştir. Çalışmalar ulusal ve uluslararası veri tabanlarında araştırılmıştır. Araştırılan yıllar arasında yapılan çalışmalarda literatür araştırmasında da görüldüğü üzere çoğunlukla Markov zincirleri kullanılmıştır.

Matematiksel modeller kullanılarak optimum (s,S) parametrelerinin belirlenmesi konusunda yapılan çalışmalar Tablo 2.1’de özetlenmiştir.

Nair ve Jose [2], farklı hizmet düzeylerine sahip iki (s,S) üretim stok sistemi analiz etmişlerdir. Çalışmada kullanılan stok sisteminde müşterilerin sistemden kayıp yaşamalarını en aza indirmek için stok seviyesi s seviyesine düştüğünde, hizmet

azaltılmış bir hızda verilmiştir. Stok modelinin optimal s ve S değerlerini elde etmişlerdir. Çalışmalarında Markov zincirleri uygulamışlardır.

Zhang ve arkadaşları [3], fırsatçı bakım eşiği, güç üretim kayıplarını azaltmak için rüzgar hızına bağlı dinamik bir değer olan yeniden sipariş seviyesi s ve maksimum stok seviyesi S ile (s,S) stok kontrol politikası uygulamışlardır. Çalışmada zaman serilerini oluşturmak ve bakım süresini elde etmek için Markov zincir modeli kullanmışlardır.

Rabta [4], kayıp satışlar ve anında yenileme ile tek ürünlü sabit (s,S) stok modelini ele almıştır. Markov zincirindeki durağan dağılım için bir üst sınır önermiştir. Çalışmada stok sisteminin durağan özellikleri üzerindeki bozunumların etkisini tahmin etmiştir.

Ko [5], bozulabilir ürünlere ve yeniden talep edilme olasılığına sahip bir (s,S) sürekli stok modeli önermiştir. Çalışmada belirtilen sistemin maliyetlerini minimize etmek s ve S 'nin optimum değerlerini sayısal olarak belirleyebilmek için önerilen sistem Markov zincirleri olarak ele alınmıştır.

Selvakumar ve arkadaşları [6], (s,S) tipi stok kontrol politikası benimsenen kesikli zaman hizmet tesisi sisteminde optimal politikasının belirlenmesi için Markov Karar Süreci yöntemini uygulamışlardır.

Wei ve Yao [7], her bir siparişin sipariş maliyeti, sipariş miktarına bağlı olan sürekli zamanlı bir stok sistemini incelemişlerdir. Çalışmalarında bu ergodik stok kontrol problemi için (s,S) politikasının küresel optimalliğini göstermişlerdir.

Guijarro ve arkadaşları [8], kayıp satış ve kesikli talep durumunda sürekli gözden geçirme (s,S) politikasıyla yönetilen bir stok kontrol sisteminde kayıp satış durumunda hizmet düzeylerinin tam hesaplanması için bir yöntem sunmuşlardır. Çalışmada önerilen yöntem ile hizmet düzeyi belirlendiğinde, taban stok seviyesinin yanlış hesaplanması eksikliğini aşacağını belirtmişlerdir.

Shi ve arkadaşları [9], çalışmasında beklenmeyen varlık arızaları altında bakım ve yedek parça envanteri yönetim problemi çözümü için bir stokastik tam sayılı programlama modeli önermişlerdir.

Barron [10], çalışmasında raf ömründe belirsizlik içeren bir (s,S) sürekli gözden geçirme stok politikası ile bozulabilir stok modelinde ortalama toplam maliyeti en aza indiren optimal parametreleri belirlemeye çalışmıştır. Çalışmasında raf ömrü ve gecikmeye bağlı maliyetler ihmal edildiğinde %25'e kadar bir kayba yol açabileceğini ifade etmiştir.

Özkar [11], müşteriler arasında öncelik kısıtı olan iki tür müşteri ve her müşteri türü için farklı ürün içeren stok sistemini iki ayrı ürün için incelemiştir. Çalışmasında matris geometrik yöntemi kullanarak kararlı durum analizi tekniğini uygulamıştır. Kar fonksiyonunu maksimize etmek amacıyla her ürün için optimum stok politikası belirlemişlerdir.

Dissa ve Ushakumari [12], rastgele ortak ömre sahip her bir mal için rastgele teslim süresi, pozitif hizmet süresi ve iki farklı müşteri sınıfı bulunan bir hizmet tesisi içinde iki ürünlü (s,S) stok kontrol sistemini, öncelikli müşteriler için kapasitesi sınırlı bir ara stok tutma durumu altında incelemiştir. Çalışmalarında (s,S) stok kontrol sistemi parametrelerinin optimum değerini bulmak için markov zincirleri kullanmışlardır.

Tablo 2.1 Matematiksel modeller kullanılarak yapılan çalışmalar

Yıl	Çalışmanın Yazarı	Çalışma Başlığı	ST	DT	Yöntem ve Araçlar
2019	Nair, S. S. Jose, K. P.	Solution to production inventory systems with orbit, buffer and different service rates	✓		Markov zincirleri
2019	Zhang, C. Gao, W. Yang, T. Guo, S.	Opportunistic maintenance strategy for wind turbines considering weather conditions and spare parts inventory management	✓		Markov zincirleri

2019	Rabta, B.	Stationary Distribution and Perturbation Bounds for a Stochastic Inventory Model.	✓		Markov zincirleri
2020	Ko, S. S.	A nonhomogeneous quasi-birth-death process approach for an (s,S) policy for a perishable inventory system with retrial demands	✓		Markov zincirleri Level-dependent quasi-birth-death süreci
2020	Selvakumar, C. Maheswari, P. Elango, C.	Discrete markov decision process-inventory control in a service facility system.	✓		Markov zincirleri
2021	Wei, B. Yao, D.	Ergodic inventory control with diffusion demand and general ordering costs	✓		Markov zincirleri
2022	Guijarro, E. Babiloni, E. Cardós, M.	On the estimation of the fill rate for the continuous (s,S) inventory system for the lost sales context	✓		Markov zincirleri
2022	Barron, Y.	A replenishment inventory model with a stock-dependent demand and age-stock-dependent cost functions in a random environment.	✓		Markov zincirleri
2023	Shi, J. Rozas, H. Yildirim, M. Gebrael, N.	A stochastic programming model for jointly optimizing maintenance and spare parts inventory for IoT applications	✓		Karışık tamsayı programlama
2023	Ozkar, S.	Two-commodity queueing-inventory system with phase-type distribution of service times	✓		Kararlı durum analizi
2023	Dissa, S Ushakumari, P. V.	Two commodity perishable queueing inventory system with random common lifetime of commodities and positive lead time	✓		Markov zincirleri

2.2 Sezgisel veya Metasezgisel Algoritma Tabanlı Simülasyon Optimizasyonu veya Simülasyon Modelleme Kullanılarak Yapılan Çalışmalar

Bu başlık altında, literatüre ait sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu veya simülasyon modelleme kullanılarak, sürekli gözden geçirme (s,S) stok kontrol politikasında optimum stok kontrol parametrelerinin

bulunmasına yönelik çalışmalar incelenmiştir. Ulusal ve uluslararası veri tabanlarında, 1984 ve 2024 yılları arasında yapılan çalışmalar araştırılmıştır.

Sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu veya simülasyon modelleme kullanılarak optimum (s,S) parametrelerinin belirlenmesi konusunda yapılan çalışmalar Tablo 2.2’de özetlenmiştir.

Federgruen ve Zipkin [13], standart varsayımlar altında (durağan veri, iyi davranış gösteren bir dönemlik maliyetler, kesikli talep ve ortalama maliyet kriteri) (s,S) stok kontrol politikasının optimal parametrelerini hesaplamak için bir algoritma sunmuşlardır. Önerilen algoritmanın iteratif bir süreç olduğunu ve başlangıçta herhangi bir mevcut (s,S) politikası ile başlayarak sonlu sayıda iterasyonla optimal bir politikaya yakınsadığını ifade etmişlerdir.

Zheng ve Federgruen [14], stok sistemlerinde hem periyodik inceleme hem de sürekli (s,S) stok kontrol politikalarını optimize etmek için yeni bir algoritma sunmuşlardır. Çalışmalarında geliştirilen algoritmanın basit ve anlaşılması kolay olduğunu ifade etmişlerdir.

Chang-hua ve You-hua [15], stokastik envanter optimizasyonu için, sistem simülasyonu kullanmışlardır. Çalışmalarında genetik algoritma, ayırık olay sistemi simülasyonu ile birleştirilmiş ve stok kontrol değişkenlerinin seçimi için genetik algoritma kullanmışlardır. (s,S) stok kontrol sisteminin sonuçlarına göre, önerilen çözümün uygulanabilir ve etkili olduğunu ifade etmişlerdir.

Eltawil ve Elnagar [16], sürekli gözden geçirme (s,S) stok kontrol sisteminin bir simülasyon modelini sunmuşlardır. Yeniden sipariş seviyesi (s) ve maksimum sipariş seviyesi (S) miktarlarını optimize etmek ve toplam stok maliyetini minimize etmek için OptQuest aracını kullanmışlardır.

Biles ve arkadaşları [17], Kriging yöntemi kullanarak (s,S) stok kontrol sistemini incelemiş, s ve S ’nin optimal değerlerini bulmayı amaçlamışlardır. Çalışmalarında Kriging yönteminin stokastik stok kontrol sisteminde, kısıtlı optimizasyon problemlerini çözmek için fırsatlar sunduğunu belirtmişlerdir.

Kleijnen ve Wan [18], kısıtlı optimizasyon probleminin simülasyonunu, stokastik teslim süresine ve hizmet seviyesi kısıtlamasına sahip (s,S) stok kontrol sisteminde incelemişlerdir. Çalışmalarında incelenen stok kontrol sistemini simülasyon yöntemi ile modellemişlerdir. Stok kontrol parametrelerinin optimal değerleri OptQuest kullanılarak bulunmuştur. OptQuest'in stokastik kısıtlı optimizasyon problemini çözmede ve optimuma yakın bir çözümü etkili bir şekilde bulma konusunda iyi olduğunu vurgulamışlardır.

İlgın [1], çalışmasında bakım politikalarına bağlı olarak yedek parça stoklarının (s,S) stok kontrol politikası ile optimizasyonu üzerine çalışmıştır. Genetik algoritma parametrelerinin, benzetimli tavlama ile bulunduğu hibrit genetik algoritma ile simülasyon optimizasyonu çalışması yapmıştır.

Li ve Wan [19], stokastik teslim süresi ve hizmet seviyesi kısıtlarına sahip (s,S) stok kontrol politikasında optimal stok politikasını bulmaya çalışmışlardır. Çalışmalarında (s,S) stok kontrol sistemini simülasyon programında modelleyip, optimizasyon aracı olarak OptQuest kullanmışlardır. Ayrıca optimal stok parametrelerinin tahmini için Brute Force algoritmasını kullanmışlardır. Çalışmalarında stokastik kısıtlı optimizasyon problemlerinde OptQuest aracının etkili olduğu vurgulanmıştır.

Kurtuluş [20], yedek parça (s,S) stok kontrol politikasında Wagner sezgiselini kullanarak stok maliyetini bulmaya çalışmış ve optimal maliyetlerle karşılaştırarak, Wagner sezgiselinin kalitesini belirlemeyi amaçlamıştır. Çalışmasında Wagner sezgiselinin anlaşılması ve uygulanması kolay bir sezgisel olduğunu vurgulamıştır.

Attanayake ve arkadaşları [21], sağlık sektörüne yönelik bir envanter optimizasyon sistemi geliştirmeyi amaçlamışlardır. Stokastik talep ve teslimat süresi dağılımları altında (s,S) sürekli gözden geçirme stok kontrol politikası için stok maliyetlerini hesaplamak ve minimize etmek amacıyla simülasyon modellemesi kullanmışlardır.

Shahi ve Pulkki [22], çalışmalarında simülasyon tabanlı bir optimizasyon tedarik zinciri modeli geliştirmişlerdir. Çalışmalarında, üretim biriminin stokastik

talep altında bilgi ve malzemelerin iki yönlü akışını entegre etmesi için simülasyon modeli kullanmışlardır. Toplam stok maliyetini en aza indiren optimum (s,S) stok kontrol politikasını bulmak için simülasyon optimizasyon aracı kullanmışlardır.

Al-Fandi ve arkadaşları [23], çalışmasında zaman zaman bazı ilaçlarda stok tükenmesi veya aşırı stok durumu yaşanan bir tıbbi depo envanter yönetimini Arena programı kullanarak simülasyon modeli ile incelemişlerdir. Çalışmada mevcut stok durumunu düzenlemek için sürekli gözden geçirme (s,S) stok kontrol politikasını öneren yeni bir sistem ile güncellemişlerdir. Arena OptQuest aracı ile toplam maliyeti minimize edecek, optimum stok kontrol parametrelerini belirleyecek bir optimizasyon problemi formüle etmişlerdir.

Deng ve arkadaşları [24], rastgele tedarik süreleri ve müşteri taleplerine göre (s,S) stok kontrol politikasını iyileştirmeyi amaçlamışlardır. Modelin karmaşıklığı nedeniyle bu modelin hesaplanması ve çeşitli ortamlarda yedek parça sorunlarının optimize edilmesi için, parçacık sürü optimizasyonu kullanmışlardır.

Yuna ve arkadaşları [25], aralıklı talep yapısına sahip minimum maksimum stok kontrol politikasında (s,S) stok maliyeti ve elde bulundurmama maliyetini dengelemek için matematiksel bir model önermişlerdir ancak önerilen modele uygun bir çözüm elde edemedikleri için ilgili modele uygun bir uygunluk fonksiyonu önermişlerdir. Çalışmalarında önerdikleri uygunluk fonksiyonun çözümü için GA ve parçacık sürü optimizasyonu yöntemlerini kullanmışlardır.

Dbouk ve arkadaşları [26], petrol, gaz, teknoloji ve otomotiv sektörlerinde ekonomideki dalgalanmaları hesaba katmak için dinamik yeniden sipariş politikasına sahip bir (s,S) stok kontrol modeli geliştirmiştir. (s,S) değerlerinin optimizasyonu genetik algoritma kullanılarak gerçekleştirilmiştir. Markov özelliğine sahip talebin bulunduğu stok kontrol sistemleri ile ilgili mevcut literatürün nispeten karmaşık matematiksel modeller ve karmaşık stok politikaları sunduğunu belirtmişlerdir. Bu politikaların uygulanmasındaki dezavantajlar arasında karmaşıklık ve uygulanabilirliğin yer aldığını ifade etmişlerdir. Çalışmalarında, matematiksel karmaşıklığı basitleştirdiklerini vurgulamışlardır.

Güçdemir ve Taşoğlu [27], yedek parça stok yönetiminin ana amacının zamanında ve maliyet etkin bir şekilde yedek parça temin etmek olduğunu, yüksek teknoloji endüstrilerinde ürün yapılarının sürekli değişken doğası nedeniyle doğrudan yedek parça yerine, kullanılacak parçaların dönüştürülmesi gerektiğini ifade etmiştir. Çalışmalarında yerine geçme stratejileri ele alınan yedek parça (s,S) stok kontrol politikasının parametrelerini benzetimli tavlama algoritması ile optimize etmişler ve toplam maliyeti minimize etmişlerdir.

Tablo 2.2 Sezgisel veya metasezgisel algoritma tabanlı simülasyon optimizasyonu veya simülasyon modelleme kullanılarak yapılan çalışmalar

Yıl	Çalışmanın Yazarı	Çalışma Başlığı	ST	DT	Yöntem	Araçlar
1984	Federgruen, A. Zipkin, P.	An efficient algorithm for computing optimal (s,S) policies		✓	Optimal (s,S) politikası belirlemek için geliştirilen algoritma	-
1991	Zheng, Y. S.	Finding optimal (s,S) policies is about as simple as evaluating a single policy	✓		Optimal (s,S) politikası belirlemek için geliştirilen algoritma	-
2006	Chang-hua, J. I. A. N. G. You-hua, H. U.	Simulation and optimization of stochastic (s,S) inventory system based on genetic algorithm	✓		Genetik algoritma tabanlı simülasyon optimizasyonu	-
2007	Eltawil, A. B. Elnagar, G. R.	Simulation optimization of an (s,S) inventory control system with random demand sizes, demand arrivals, and lead times	✓		Simülasyon Optimizasyonu	Arena simülasyon programı OptQuest aracı
2007	Biles, W. E. Kleijnen, J. P. Van Beers, W.C. Van Nieuwenhuyse, I.	Kriging metamodeling in constrained simulation optimization: an explorative study	✓		Kriging yöntemi kısıtlı simülasyon optimizasyonu	Arena simülasyon programı Excel Solver

2007	Kleijnen, J. P. Wan, J.	Optimization of simulated systems: OptQuest and alternatives	✓		Simülasyon optimizasyonu Brute Force algoritması	Arena simülasyon programı OptQuest aracı
2007	İlgin, M. A.	Joint optimization of spare parts inventory and maintenance policies using hybrid genetic algorithms	✓		Genetik algoritma Benzetimli tavlama Hibrit algoritma	Arena simülasyon programı
2008	Li, L. Wan, J.	Simulation for Constrained Optimization of Inventory System by Using Arena and OptQuest	✓		Brute Force algoritması	Arena simülasyon programı OptQuest aracı
2012	Kurtulus, I. S.	A Test of Wagner's Heuristics for the Spare Parts Inventory Control Problem	✓		Wagner algoritması	-
2014	Attanayake, N. Kashef, R. F. Andrea, T.	A simulation model for a continuous review inventory policy for healthcare systems.	✓		Simülasyon modelleme	Simul8 simülasyon programı
2015	Shahi, S. Pulkki, R.	A simulation-based optimization approach to integrated inventory management of a sawlog supply chain with demand uncertainty	✓		Simülasyon optimizasyonu	AnyLogic simülasyon programı OptQuest aracı
2019	Al-Fandi, L. M. Obaid, A. A. B. Alfailakawi, B. I. Alsubaieci, H. A. Khudhair, S. A.	A simulation study to determine the parameters of medicine inventory policy	✓		Simülasyon optimizasyonu	Arena simülasyon programı OptQuest aracı
2021	Deng, H. Shi, Q. Wang, Y.	A Joint Optimization model of (s,S) inventory and supply strategy using an improved PSO-based algorithm	✓		Parçacık sürü optimizasyonu	-

2023	Yuna, F. Erkayman, B. Yılmaz, M.	Inventory control model for intermittent demand: a comparison of metaheuristics		✓	Genetik algoritma Parçacık sürü optimizasyonu	-
2023	Dbouk, W. Tarhini, H. Nasr, W.	Re-ordering policies for inventory systems with a fluctuating economic environment–Using economic descriptors to model the demand process	✓		Genetik algoritma	-
2024	Güçdemir, H. Taşoğlu, G.	Part transformation-based spare parts inventory control model for the high-tech industries	✓		Benzetimli tavlama	Arena simülasyon programı

Literatürde stok kontrol yönetimi ile ilgili çok fazla çalışma bulunmaktadır. Ancak simülasyon, sezgisel ve metasezgisel çözüm yöntemlerinin kullanıldığı daha az sayıda çalışma mevcuttur. Literatür araştırmasında görüldüğü üzere yapılan çalışmalarda daha çok matematiksel modelleme üzerinde durulmuştur ve daha fazla sayıda varsayım altında incelemeler yapılmıştır. Fakat günümüzde, matematiksel modelleme karmaşık ve büyük ölçekli sistemler için beklentileri tam olarak karşılayamamaktır. Optimizasyon tabanlı yaklaşımların uygulanabilmesi ve etkin olabilmesi için genellikle pek çok sadeleştirme ve varsayım gereklidir [28]. Bu nedenle gerçek hayat problemleri stokastik ve daha karmaşık olduğu için simülasyon, sezgisel ve metasezgisel çözüm yöntemleriyle ilgili daha çok çaba sarf etmemiz gereklidir. Dolayısıyla, bu tezde stok kontrol parametrelerinin belirlenmesi için günümüz gerçek yaşam problemlerinde daha gerçekçi sonuçlar veren genetik algoritma ve benzetimli tavlama tabanlı bir simülasyon optimizasyonu yöntemi kullanılarak gerçek bir problem üzerine yoğunlaşmıştır.

Bu çalışma ile literatüre katkı yapılacak olan en önemli noktalar aşağıdaki şekilde sıralanabilir:

- Bu çalışma ile malzeme planlama yöneticilerine stokastik bir sistemde bir stok kontrol politikasının uygulanması, toplam stok maliyetini minimize eden stok kontrol değerlerinin bulunması ve stok maliyetini etkileyen parametrelerin analiz edilmesi için yol gösterilmesi amaçlanmıştır. Bu amaçla sistemde kullanılacak girdiler analiz edilmiş, stok kontrol sistemi simülasyon modelleme ile modellenmiş, simülasyon modelinin optimizasyonu için metasezgisel algoritma tabanlı simülasyon optimizasyonu yöntemleri uygulanmış ve sonuçlar karşılaştırılmıştır. Daha sonrasında sistem farklı faktör büyüklükleri altında incelenmiş, hangi faktörlerin toplam maliyete etkisi olduğu araştırılmıştır. Bu çalışma gerçek hayat stok kontrol problemlerinin, daha az varsayım altında ve daha gerçekçi şekilde stokastik simülasyon modellemesi yapılması, simülasyon programıyla entegre çalışan metasezgisel algoritma tabanlı simülasyon optimizasyon yöntemi kullanılması ve sistemin durum senaryolarının incelenmesi nedeniyle diğer çalışmalardan ayrılmaktadır.
- Bu çalışmada, bir gerçek hayat probleminden alınan gerçek veriler kullanılmıştır. Bu nedenle incelenen stok kontrol modelinin gerçek hayat problemlerindeki etkisini görebilmemizi sağlayacak çalışmalardan biri olarak literatüre katkı sağlayacaktır.

3. STOK YÖNETİMİ

İngilizce’de "inventory" olarak ifade edilen terimin Türkçe karşılığı "envanter"dir. Stok terimi, işletmelerin üretim süreçlerinde kullanılan malzemeleri ve ürünleri ifade ederken, envanter terimi genellikle muhasebe açısından kullanılır. Envanter işletmelerin mallarının ve varlıklarının belirli bir dönemdeki fiziksel sayımını ifade eder [29].

Stoklar, ticari işletmeler için işletme sermayesi yapısında stratejik bir konumda bulunur. Genellikle işletmenin en büyük varlık bileşenini oluştururlar ve işletme sermayesi büyük ölçüde stok cirosuna dayanır. Dolayısıyla, stokların karı maksimize etmede önemli bir rol oynaması ve dönen varlıklar arasında kritik bir yere sahip olması son derece doğaldır [30].

Stok, işletmelerin üretim süreçlerinde veya satış faaliyetlerinde kullanmak üzere ellerinde bulundurdıkları, depoladıkları veya işledikleri her türlü malzeme veya ürünü ifade eder. Bu malzemeler genellikle işletmenin faaliyetlerini sürdürebilmesi için gerekli olan kaynakları temsil eder. Stok, işletmelerin operasyonel süreçlerinde kritik bir rol oynar; çünkü doğru stok seviyeleri, üretim ve hizmet kalitesini sağlamaya yardımcı olurken, aynı zamanda müşteri taleplerini karşılamak için gereken esnekliği sağlar [31].

Stoklar, üretim sistemlerinde hayati bir role sahiptir ve üretim sürekliliği için kaçınılmazdır. Üretim sistemlerinde stoklar, hammaddelerden bitmiş ürünlere kadar geniş bir yelpazeyi kapsar ve bu stokların etkin yönetimi, işletmenin operasyonel verimliliğini artırır. Bu stoklar genellikle hammaddeler, yarı mamuller ve nihai ürünler olarak sınıflandırılır ve üretim sürecinin farklı aşamalarında stratejik bir rol oynarlar. İyi bir stok yönetimi, üretim süreçlerinde sorunsuz bir akış sağlar ve müşteri taleplerinin zamanında karşılanmasını sağlar [32].

Stokların varlığı, üretim sistemlerinin esnekliğini ve adaptasyon yeteneğini artırır. Üretim hattında beklenmedik duraklamalar veya ani talep artışları meydana geldiğinde stokların kritik bir rolü vardır. Bu stoklar, üretim sürecinin kesintisiz olarak devam etmesini sağlar ve tedarik zincirindeki belirsizlikleri en aza indirir. Stokların

etkin yönetimi, işletmenin müşteri memnuniyetini yüksek tutmasına ve rekabet gücünü korumasına yardımcı olur [33].

Stok yönetiminin etkin bir şekilde yapılması, işletmelerin maliyet kontrolünü sağlar. Stok maliyetleri genellikle elde bulundurma maliyetleri, sipariş verme maliyetleri ve elde bulundurmama maliyetleri olarak üç temel kategoride incelenir. Bu maliyetlerin doğru yönetimi, işletmenin rekabet avantajını sürdürmesine ve karlılığını korumasına yardımcı olur. Dolayısıyla, işletmelerin sürdürülebilir büyümesine önemli katkı sağlar [34].

İşletmelerin stok bulundurma sebepleri aşağıdaki gibi belirtilmiştir [35]:

- **Talepteki Dalgalanmalar:** İşletmeler, müşteri memnuniyetini sağlamak ve kayıp satış durumlarını önlemek için stok tutarlar. Bu şekilde, beklenmedik talep artışları veya tedarik sorunları karşısında esneklik sağlar.
- **Tahmin edilebilirlik:** Stoklar, işletmelerin belirli bir dönemde ne kadar malzeme ve parçaya ihtiyaç duyduklarını belirlemelerine yardımcı olur. Bu da üretim süreçlerini verimli bir şekilde yönetmelerini sağlar.
- **Tedarik Güvenliği:** Elde bulunan stoklar, tedarikçilerden zamanında mal alınamadığında veya belirli bir hammaddenin temini zorlaştığında müşteri taleplerini karşılamaya devam edilmesine olanak sağlar.
- **Fiyat Dalgalanmalarına Karşı Koruma:** Büyük miktarlarda stok alımları, fiyat dalgalanmalarından korunmayı sağlar. Bu, işletmelerin maliyetlerini daha öngörülebilir hale getirir.
- **Düşük Sipariş Maliyetleri:** Büyük miktarlarda seyrek alımlar, küçük miktarlarda sık alımlara göre daha az maliyetlidir. Bu da işletmelerin toplam maliyetlerini düşürür.

- **Miktar İndirimleri:** Büyük miktarlarda stok alımları yapıldığında genellikle miktar indirimleri elde edilir. Bu, birim başına maliyetleri düşürür ve işletmeye tasarruf sağlar.

3.1 Stokların Sınıflandırılması

İşletmede etkin stok yönetimi için stok kalemlerinin doğru şekilde sınıflandırılması kritiktir. Bu sınıflandırma, stokların yönetimini kolaylaştırır ve doğru stratejilerin belirlenmesine yardımcı olur. Ayrıca, stokların sınıflandırılması, işletmenin operasyonel verimliliği artırmasına olanak tanır [30]. Farklı stok sınıflandırmaları mevcuttur. Bu sınıflandırmaların başlıcaları aşağıda verilmiştir.

3.1.1 Stok Türüne Göre Sınıflandırma

Üretim planlama ve kontrol, tedarik, satış ve maliyet muhasebesi bölümleri için uygun görülen stok sınıflandırması şu şekildedir [29]:

1. **Hammaddeler:** İşletmeye giren ve üzerinde işlem yaparak değer kazandırılan tüm varlıkları kapsar.
2. **Yarı Mamuller:** Üretim birimleri arasındaki ara depolarda tutulan ve üzerlerinde işlemler tamamlanmadığı için henüz mamül özelliği kazanmamış varlıklardır.
3. **Mamuller:** Müşteriye teslim edilmek üzere son ürün depolarında tutulan, üzerinde yapılması gereken tüm işlemlerin tamamlandığı varlıklardır. Bu varlıklar son ürün deposunda hareketsiz durdukları için takip ve kontrolü daha kolaydır.
4. **Hazır Parçalar:** Son ürünün bir bölümünü oluşturan ve genellikle dışarıdan temin edilen, somun, civata gibi çok kullanılan basit parçalar veya kompleks ürünlere monte edilen motor gibi karmaşık parçaları içerir.

5. **Yardımcı Malzemeler:** Üründe doğrudan kullanılmayan, tamir parçaları, kesme sıvısı, makine yağı gibi malzemeleri kapsar.

3.1.2 Stok İşlevine Göre Sınıflandırma

Stoklar işlevine göre aşağıdaki şekilde sınıflandırılmaktadır [35].

1. **Güvenlik Stoğu:** Talep dalgalanmalarına karşı firmaların her zaman ellerinde bulundurdukları belirli miktardaki stoktur. Bu miktar, geçmiş dönem verilerine ve gelecekteki büyük talep değişikliklerine göre belirlenir
2. **Tahmin Stoğu:** Dönemsel talep dalgalanmalarına veya planlı üretim duraksamalarına karşı hazırlık amacıyla elde tutulan stoktur.
3. **Transit Stoğu:** Tedarik zinciri içinde hareket halinde bulunan stok türüdür.
4. **Çevrim Stoğu:** Küçük fakat daha sık müşteri taleplerini karşılayabilmek için normalden daha fazla miktarda elde tutulan stoktur.

3.1.3 ABC Analizine Göre Sınıflandırma

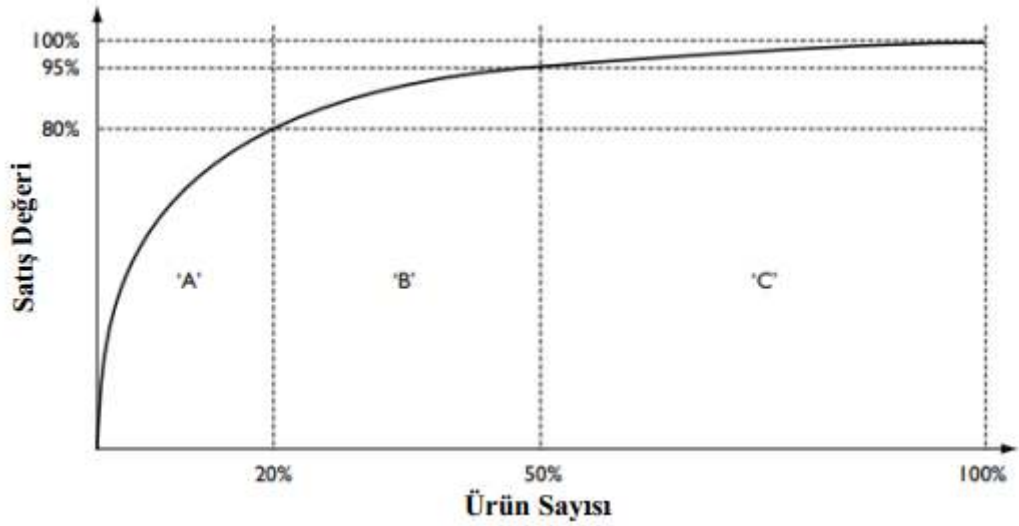
ABC analizi, stokları önem derecelerine göre kategorize eder ve bu sınıflandırma, stok yönetiminde kaynakların verimli kullanımını sağlar. A sınıfı stoklar, toplam stok değerinin büyük bölümünü oluşturur ve özenle yönetilmelidir. Bu stoklar, işletme gelirlerinin önemli bir kısmını oluşturduğu için kritik önemdedir. B sınıfı stoklar, orta seviyede önem taşır ve daha az sıkı denetime tabi tutulur, C sınıfı stoklar ise toplam stok değerinin küçük bir kısmını kapsar ve daha esnek bir yönetim yaklaşımı gerektirir [36].

A sınıfı stokların yönetiminde, sıkı denetim ve ayrıntılı izleme büyük önem taşır. Yüksek değerli ve stratejik ürünler olduklarından, talep tahminleri ve stok seviyeleri detaylı izlenmelidir. A sınıfı stokların optimal seviyede tutulması, işletmenin maliyetlerini kontrol altında tutmasına ve müşteri memnuniyetini artırmasına yardımcı olur [36].

B sınıfı stoklar, genellikle daha düşük değerli ürünlerden oluşur ve yönetimleri daha az sıkı kontrol gerektirir. Bu stokların izlenmesi ve yönetimi, A sınıfı stoklara kıyasla daha az maliyetli ve daha az karmaşıktır. B sınıfı stokların etkin bir şekilde yönetilmesi, işletmenin genel stok maliyetlerini düşürür ve stok yönetiminin verimliliğini artırır [38].

C sınıfı stoklar, düşük değerli ve genellikle büyük miktarlarda bulundurulanan ürünlerden oluşur. Bu stokların yönetimi, daha az sıkı kontrol gerektirir ve basit izleme yöntemleriyle yeterli olur. C sınıfı stokların verimli yönetimi için düşük maliyetli stratejiler tercih edilmelidir. Bu stokların doğru yönetilmesi, stok maliyetlerini azaltırken işletmenin kaynaklarını daha kritik alanlara yönlendirmesine imkan tanır. Bu sayede, işletme genel stok yönetiminde daha fazla verimlilik ve maliyet kontrolü sağlayabilir [32].

Ürünlerin yüzde 20'sinin yüzde 80'ini temsil ettiğini gösteren tipik Pareto Eğrisi Şekil 3.1'de verilmiştir.



Şekil 3.1 Pareto eğrisi [37]

Stokları sınıflandırırken aşırı derecede ayrıntıya inmek, yöneticilerin gereksiz ayrıntılarla boğulmasına ve sağlıklı kararlar almasını zorlaştırabilir. Öte yandan, stokları çok az sayıda başlık altında toplamak ise sınıflandırmanın faydalarını

azaltabilir. Bu nedenle, stoklar yöneticilere gerekli bilgileri sağlayacak şekilde dengeli ve uygun bir sınıflandırma ile ayrılmalıdır [39].

3.2 Stok Maliyetleri

Stok maliyetlerinin etkin yönetimi, işletmenin rekabet avantajını korumasına ve finansal performansını optimize etmesine katkı sağlar. Bu maliyetleri yönetmek için uygulanan stratejiler arasında stok devir hızını artırmak, optimal stok seviyelerini belirlemek ve tedarik süreçlerini iyileştirmek yer alır. Etkili stok maliyeti yönetimi, işletmenin sürdürülebilir büyümesini ve başarısını destekler [40].

Stok maliyetleri, işletmelerin stok yönetiminde göz önünde bulundurması gereken kritik unsurlardan biridir. Bu maliyetler genellikle elde bulundurma maliyeti, sipariş verme maliyeti ve elde bulundurmama maliyeti olarak üç ana kategoriye ayrılır. Bu maliyetlerin etkin bir şekilde yönetilmesi, işletmenin rekabet avantajını sürdürmesine ve karlılığını artırmasına yardımcı olur [36].

3.2.1 Elde Bulundurma Maliyeti

İşletmeler, stok bulundurmak için finansman sağlamak durumunda kalabilir; bu durum genellikle ya fırsat maliyetiyle ya da faiz maliyetiyle ilişkilendirilen bir maliyet gerektirir [30].

Genel olarak çoğu şirket, sürdürülebilir kârlılık sağlayabilmek için bireylerden daha yüksek getiri oranları elde etmek zorundadır. Alternatif yatırımın fırsat maliyeti, iç getiri oranı, varlık getirisi ve diğer engellerle ilişkilendirilen standart muhasebe önlemleriyle bağlantılıdır (ancak bunlar aynı değildir). Bir yatırımı cazip hale getirecek minimum faiz oranı firma için önemlidir. Alternatif yatırımın fırsat maliyeti için doğru faiz oranını belirlemek genellikle zordur; bu değer, firmanın muhasebe departmanı tarafından tahmin edilir ve muhasebe önlemlerinin bir karışımını içerir. Bu bağlamda, elde bulundurma maliyetinin bu bileşenini ifade etmek için genellikle sermaye maliyeti terimi kullanılır. Elde bulundurma maliyeti aşağıda listelenen dört bileşenin toplamından oluşan bir faiz olarak dikkate alınabilir [41]:

1. Sermaye maliyeti
2. Vergiler ve sigorta
3. Depolama maliyeti
4. Kırılma ve bozulma

Örneğin sermaye maliyeti %28, vergiler ve sigorta %2, depolama maliyeti %6, kırılma ve bozulma %1 olarak belirlendiğinde toplam faiz ücreti dört bileşenin toplamı olarak %37 bulunur. Bu değerler için bir yıllık dönemde stoklara yatırılan her bir TL için 37 kuruş ücret alınmaktadır. Ancak, stoklar genellikle TL cinsinden değil, birim olarak ölçüldüğü için, yıllık olarak birim başına TL cinsinden ifade etmek uygun olabilir. Burada "c" bir birim stoğun TL değerini, "i" yıllık faiz oranını ve "h" birim başına TL cinsinden elde bulundurma maliyetini temsil eder.

$$h = i * c \quad (3.1)$$

Dolayısıyla bu örnekte, değeri 180TL olduğunu varsaydığımız bir ürünün yıllık elde bulundurma maliyeti $h=(0,37)*(180\text{TL})=66,60\text{TL}$ olacaktır. Bu ürünün 300'ünü beş yıl boyunca elimizde tutsaydık, beş yıl boyunca toplam elde bulundurma maliyeti $(5)*(300)*(66.60) = 99,900 \text{ TL}$ şeklinde hesaplanacaktır [40].

3.2.2 Sipariş Verme ve Üretim Hazırlık Maliyeti

Her işletme üretim sürecine başlarken, üretimin gerçekleştirilmesi için gerekli olan sipariş ve hazırlık işlemlerini yapar. Bu işlemler genellikle sabit maliyetler gerektirir ve bir kez tamamlandıktan sonra, belirli bir malın üretim sürecinde ek maliyetler ortaya çıkmaz [42].

Çoğu uygulamada, sipariş maliyeti genellikle iki bileşen içerir: sabit ve değişken bileşenler. Sabit maliyet (K), sipariş boyutundan bağımsız olarak ortaya çıkar (sıfır olmadığı sürece). Değişken maliyet v ise birim başına gerçekleşir. Ayrıca K olarak da hazırlık maliyeti ve v olarak da orantılı sipariş maliyeti olarak adlandırılabilir. $V(x)$ olarak tanımlanan bu ifade, x birim siparişin (veya üretimin) maliyetini belirtir [41].

$$V(x) = \begin{cases} 0 & x = 0, \\ K + vx & x > 0. \end{cases} \quad (3.2)$$

3.2.3 Elde Bulundurmama Maliyeti

Elde bulundurmama maliyeti, aynı zamanda stoksuzluk veya kıtlık maliyeti olarak da bilinir ve talebi karşılamak için yeterli stok bulunmadığında ortaya çıkan maliyeti ifade eder. Eğer gelen talep stoklardan karşılanamazsa, stoklar yenilenene kadar müşteri talebi ya geciktirilir (backorder) ya da karşılanamayıp kaybedilir (lost sale). Bu iki durumun hangisinin gerçekleşeceği, genellikle ürünün ve müşteri tabanının özelliklerine bağlıdır ve çoğu zaman bu iki durumun bir kombinasyonu yaşanır. Ertelenmiş talep durumunda, elde bulundurmama maliyetine ek olarak muhasebe veya gecikme maliyetleri ortaya çıkabilir. Kayıp satış durumunda ise, potansiyel kar kaybı yaşanır. Her iki senaryoda da müşteri memnuniyeti ve "iyi niyet" kayıpları meydana gelir. İtibar kaybını elde bulundurmama maliyetinin bir parçası olarak hesaplamak ise pratikte oldukça zor olabilir [41].

Elde bulundurmama maliyeti, yeterli stok bulunmadığında oluşan tüm maliyetleri kapsar. Bu maliyetler arasında üretim kesintileri, müşteri siparişlerinin karşılanamaması ve müşteri memnuniyetsizliği gibi unsurlar bulunur [36]. Üretim duraklamaları, stok yetersizliği durumunda oluşan maliyetlerin başında gelir. Bu maliyetler, üretim hattının durması, iş gücü kaybı ve gecikme cezalarını içerir. Üretim duraklamalarının önlenmesi, etkili stok yönetimi ile doğrudan ilişkilidir [43].

3.3 Stok Kontrolü

Stok kontrol problemleri, yöneticilerin genellikle stok yenileme siparişi ne kadar olmalı ve stok yenileme siparişi ne zaman verilmeli sorularını cevaplandırması gereklidir. Bu kararlar, genellikle müşteri hizmet düzeyini en üst düzeye çıkartırken maliyetleri en aza indirmeyi amaçlayan stok kontrol problemlerinin temel unsurlarıdır. Her işletme, büyüklüğüne, yönetim politikalarına, üretim tipine, mali kaynaklarına ve diğer çeşitli faktörlere göre kendi stok kontrol sistemini oluşturur [44].

Stok kontrolü için literatürde çeşitli yöntemler bulunmaktadır. Genel olarak, stok kontrolü talebin veya tedarik süresinin kesin olarak bilindiği deterministik stok kontrolü ve talebin veya tedarik süresinin rastgele kabul edildiği, çoğu kez bir olasılık dağılımı yardımıyla ifade edilebildiği stokastik stok kontrolü olmak üzere 2 başlık altında ele alınmaktadır.

Stok kontrol sisteminin temel amacı, aşağıdaki üç sorunu çözmektir:

1. Stok durumu ne sıklıkla kontrol edilmelidir?
2. Sipariş ne zaman verilmelidir?
3. Siparişin miktarı ne kadar olmalıdır?

Deterministik talep koşullarında, ilk sorun önemsizdir çünkü stok durumunu herhangi bir anda bilmek, makul bir süre boyunca her an hesaplanabilmesini sağlar. Ayrıca, deterministik talep altında ikinci sorun, stok seviyesini önceden belirlenmiş bir değere (genellikle sifıra) ulaştığında sipariş verilerek çözülür. Son olarak, deterministik talep altında, ekonomik sipariş miktarı veya onun varyasyonları veya dağıtım prosedürlerinden biri kullanılır. Olasılık talebi koşullarında ise, yanıtlar daha karmaşıktır. İlk noktada, stok durumunu belirlemek için kaynak (işgücü, bilgisayar zamanı vb.) gereklidir. Stok durumu ne kadar az sıklıkta kontrol edilirse, beklenmeyen talep değişikliklerine karşı sistemi korumak için gereken süre de o kadar uzun olur. İkinci sorunun yanıtı, fazla stok bulundurmanın maliyetleri ile yetersiz müşteri hizmeti sağlamanın maliyetleri arasındaki dengeye dayanır. Üçüncü sorunun yanıtlanmasında ise, temel ekonomik sipariş miktarının belirlenmesinde tartışılan faktörler geçerlidir, ancak yönetimin belirlediği bazı hizmet kriterlerine göre bir etkileşim vardır; "Ne zaman yenilenmeli?" sorusu, kullanılan yenileme miktarından etkilenebilir. Bu üç temel soruya yanıt verebilmek için yöneticilerin çeşitli unsurları belirlemesi gerekmektedir. Bu unsurları, yöneticilerin stok politikalarını sistematik bir şekilde oluşturabilmeleri için şu dört soru şeklinde sıralayabiliriz:

1. İlgili ürün ne kadar önemlidir?
2. Stok durumu sürekli mi yoksa periyodik olarak mı kontrol edilmelidir?
3. Stok politikası ne şekilde yapılandırılmalıdır?
4. Hangi spesifik maliyet veya hizmet hedefleri belirlenmelidir?

Yöneticiler, öncelikle değerlendirdikleri ürünün firma için ne kadar kritik olduğunu belirlemelidir, birçok firma ürünlerini A, B veya C kategorilerine ayırmaktadır [45].

Deterministik modeller 1915’li yıllarda Ford W. Harris aracılığıyla Ekonomik Sipariş Miktarı modeli bulunarak literatüre girmiştir [108]. Bu model 1934 yılında Wilson tarafından geliştirilmiştir. Bu model ile başlayan stok modelleri daha sonrasında çeşitlendirilmiştir. Deterministik modellerde talep ve/veya tedarik süresi bellidir ve değişmemektedir [46]. Stokastik (olasılıklı) stok modelleri ise ilk olarak 1951 yılında Arrow, Harris ve Marschak tarafından incelendiği görülmüştür. Olasılıklı stok modellerinde talep ve/veya temin süresi bilgileri belirsizdir [47].

Bu çalışmada stokastik stok kontrol modelleri üzerine çalışılmasına rağmen, mevcut modellerin temeli olan deterministik stok kontrol modellerine de kısaca değinilecektir. Deterministik ve stokastik stok kontrol modelleri ile ilgili bilgiler aşağıdaki bölümlerde verilmiştir.

3.3.1 Deterministik Stok Kontrol

Deterministik stok kontrol modellerinde en temel varsayım, talebin sabit ve bilindiği durumdur. Bu iki farklı varsayım gerçekte şunlardır: Birincisi, talebin değişmeyeceği ve sabit kalacağı, diğeri ise talebin önceden tahmin edilebileceği varsayımdır. Basit ekonomik sipariş miktarı modeli, sabit ve bilinen talep üzerine kurulmuştur [48].

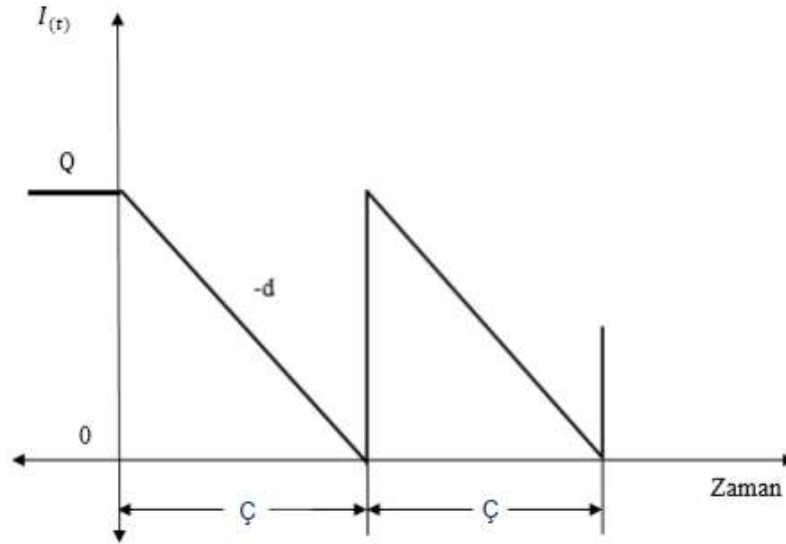
Klasik stok kontrol modelleri, sipariş miktarlarını ve zamanlarını son derece idealize edilmiş koşullar altında belirlemek için kullanılır ve şu varsayımlara dayanmaktadır:

- Sadece tek bir ürün için geçerlidir.
- Yıllık talep miktarları bilinir.
- Tüketim oranı sabittir.
- Tedarik süresi göz ardı edilir.

- Her sipariş, tek seferde yapılan teslimatlarla gelir.
- Miktar indirimi uygulanmaz.
- Stoksuz kalma durumu kabul edilmez.

Bu model, gerçek dünyada uygulanması zor olan birçok varsayıma dayanmasına rağmen, stok yönetiminin temel modeli olarak kullanılır. Bunun nedeni, stok yönetimiyle ilgili kavramları net bir şekilde ortaya koymasındır [49].

Toplam sipariş maliyeti optimize eden sipariş miktarına ekonomik sipariş miktarı denir. Şekil 3.2’de Ekonomik sipariş miktarı modelinin stok geometrisi verilmiştir.



Şekil 3.2 Ekonomik sipariş miktarı modeli [50]

Bu modelin parametreleri aşağıda verilmiştir:

- d : Yıllık talep (adet/yıl)
- Q : Sipariş miktarı (adet)
- K : Sipariş maliyeti (TL/sipariş)
- h : Elde bulundurma maliyeti (TL/adet/yıl)
- $\bar{C}$: Çevrim süresi

Bu gösterimleri kullanarak toplam stok maliyetini (TSM) veren denklem, yıllık toplam sipariş miktarı,

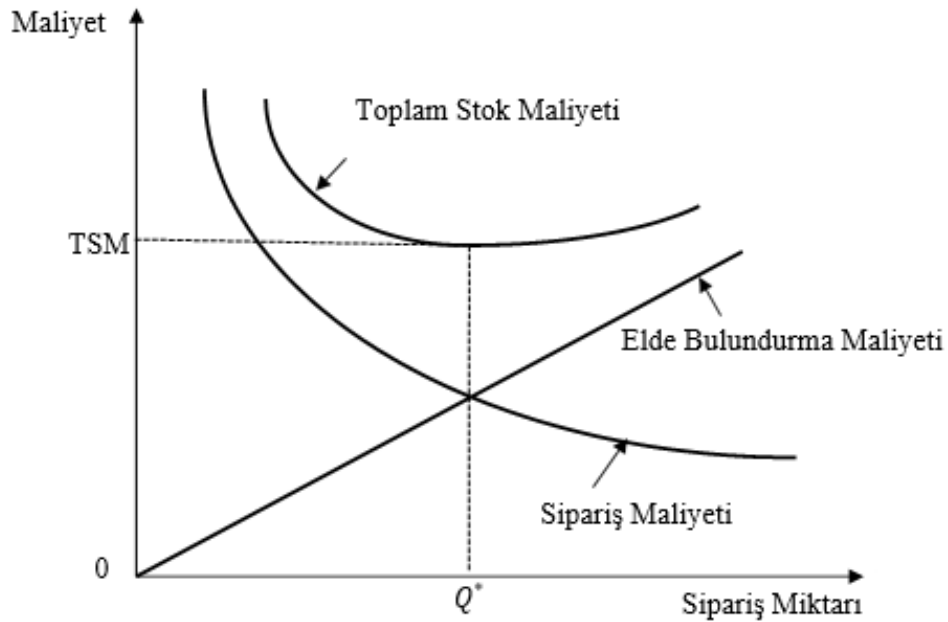
$$K \cdot \left(\frac{d}{Q}\right) \quad (3.3)$$

ve yıllık elde bulundurma maliyeti, ortalama stok $(Q/2)$ olduğundan,

$$h \cdot \left(\frac{Q}{2}\right) \quad (3.4)$$

$$TSM = k \cdot \left(\frac{d}{Q}\right) + h \cdot \left(\frac{Q}{2}\right) \quad (3.5)$$

Toplam stok maliyeti (TSM), iki maliyetin toplamı şeklinde yazılır. Amaç TSM'yi minimum yapan Q değerini bulmaktır. Şekil 3.3'den görüldüğü üzere toplam stok maliyeti diğer iki maliyetin kesiştiği noktada Q^* minimum değerini alır.



Şekil 3.3 Maliyet değişimi grafiği [50]

$$K \cdot \frac{d}{q} = h \cdot \frac{Q}{2} \quad (3.6)$$

Eşitliğinde Q çözüldüğünde TSM' yi minimum yapan ekonomik sipariş miktarı (ESM) için aşağıdaki formül bulunur [50].

$$Q^* = \sqrt{\frac{2kd}{h}} \quad (3.7)$$

Ekonomik Sipariş Miktarı modeli, talep yapısının sabit ve belirli olduğu durumlarda etkili bir araç olarak kullanılmaktadır. Ancak, talep yapısının yüksek derecede değişken olduğu koşullarda bu model yetersiz kalmaktadır. Bununla birlikte, talebin belirli bir olasılıksal dağılıma sahip olduğu fakat değişkenliğin düşük seviyelerde seyrettiği durumlarda da ESM modeli uygulanabilir. ESM modeli, stok yönetiminde klasik bir yaklaşımdır ve belirli varsayımlar altında optimal sipariş miktarını hesaplar. Ancak, gerçek hayatta talep genellikle deterministik değildir ve bu nedenle stok yönetimi stratejileri de daha esnek olmalıdır. Talebin değişken olduğu ancak bu değişkenliğin büyük dalgalanmalar göstermediği durumlarda, ESM modeli hala geçerliliğini koruyabilir ve kullanılabilir. Bu tür durumlarda, ESM modeli, stok seviyelerini optimize etmek ve maliyetleri minimize etmek için uygun bir araç olabilir [51].

3.3.2 Stokastik Stok Kontrol

Talebin veya tedarik süresinin rastgele kabul edildiği, çoğu kez bir olasılık dağılımı yardımıyla ifade edilebildiği stok modeline stokastik veya olasılıklı stok denmektedir.

Talep stokastik olduğunda ilgili kavramların aşağıdaki şekilde kategorize edilmesi gereklidir:

- Eldeki Stok: Fiziksel olarak rafta bulunan ve asla negatif olamayacak stok miktarını ifade eder. Bu miktar, belirli bir müşteri talebinin doğrudan raftan karşılanıp karşılanamayacağını belirlemek için önemlidir.

- Net Stok: Eldeki stok miktarından ön siparişlerin (yani, bekleyen siparişlerin) çıkarılmasıyla hesaplanır. Bu miktar, bazı matematiksel çıkarımlarda kullanılır ve aşağıdaki önemli tanımın bir bileşenidir.
- Stok Seviyesi: Ertelenmiş talep (backorder) miktarından eldeki stok miktarı ile sipariş miktarı toplamının çıkarılmasıyla elde edilir.
- Ertelenmiş Talep: Stok olmadığı için henüz karşılanmayan taleplerdir. Daha önceden verilmiş sipariş geldiğinde karşılanabilir.
- Kayıp Satış (Lost Sales): Sipariş geldiğinde stok olmadığı için talepler karşılanamaz ve müşteri siparişini başka bir işletmeden karşılar.
- Güvenlik Stoğu: Yenileme gelmeden önceki net stoğun ortalama seviyesi olarak tanımlanır. Yenilemenin tam zamanında yapıldığını varsaydığımızda, güvenlik stoğu sıfır olurdu. Pozitif bir güvenlik stoğu ise, tedarik süresi boyunca beklenenden fazla talep olması durumunda bir tampon işlevi görür. Güvenlik stoğunun miktarı, stok tükendiğinde talebin nasıl karşılanacağına bağlı olarak değişir [45].

Stokastik stok kontrol sistemleri gözden geçirme şekline göre 2 başlıkta ele alınmıştır.

3.3.2.1 Sürekli Gözden Geçirme Sistemleri

Stok durumunun ne sıklıkla belirlenmesi gerektiği sorusunun cevabı, ardışık iki stok kontrolü arasındaki süreyi, yani inceleme aralığını (R) belirler. Sürekli inceleme, stok seviyesinin her an bilindiği aşırı bir durumdur. Satış noktası veri toplama sistemleri (elektronik tarayıcılar gibi) işlemlerin raporlanmasına izin vererek perakende düzeyinde büyük bir etkiye sahiptir [45].

Sürekli takip sistemlerinde, temel olarak (s,Q) ve (s,S) olarak bilinen iki stok politikası mevcuttur [52].

Sürekli incelemenin en büyük avantajı, aynı düzeyde müşteri hizmeti sağlamak için periyodik incelemeye göre daha az güvenlik stoğu gerektirmesi ve dolayısıyla daha düşük taşıma maliyetleri sunmasıdır. Bunun nedeni, güvenlik stoğunun gerekli olduğu sürenin periyodik incelemede daha uzun olmasıdır; çünkü stok seviyesinin, inceleme aralıkları arasında herhangi bir yeniden sipariş işlemi yapılamadan önemli ölçüde düşme ihtimali vardır [45].

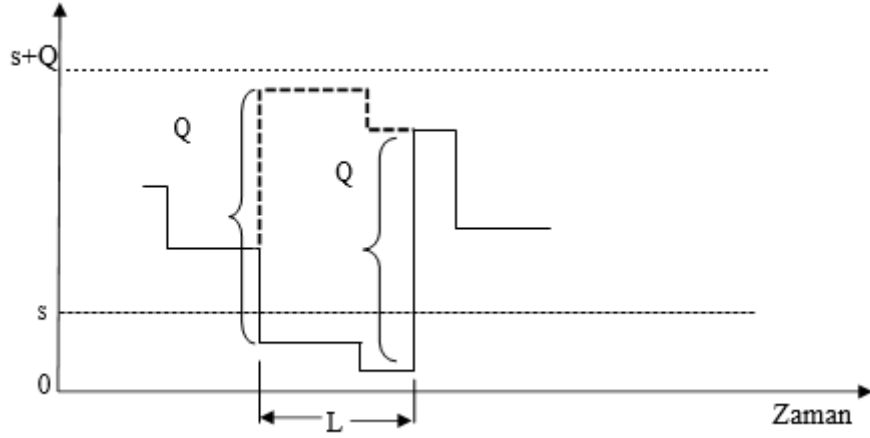
3.3.2.1.1 (s,Q) Sistemi

Bu, sürekli gözden geçirme sistemine dayanan bir stok yönetimidir (yani $R = 0$). Stok seviyeleri yeniden sipariş noktası s 'ye veya altına düştüğünde, sabit bir miktar sipariş verilir. Siparişlerin tetiklenmesi için net stok yerine stok pozisyonunun kullanıldığını belirtmek önemlidir. Stok pozisyonu, sipariş edilen ancak henüz teslim alınmamış malzemeleri de içerdiğinden, talebi doğru bir şekilde karşılar. Net stok kullanılsaydı, büyük bir sevkiyatın yarın yapılacak olması durumunda bile gereksiz yere bir sipariş daha verilebilirdi [45].

Stok seviyesi belirli bir sipariş noktası seviyesine (s) veya daha altına indiğinde, sabit bir miktarda (Q) sipariş verilir [52].

Sabit sipariş miktarı (s,Q) sisteminin avantajları şunlardır: stok görevlisinin anlaması oldukça kolaydır, hata olasılığı daha düşüktür ve tedarikçi için üretim gereksinimleri daha öngörülebilirdir. Ancak, (s,Q) sisteminin en büyük dezavantajı, bireysel işlemlerin büyük olduğu durumlarda etkili bir şekilde çalışamayabilmesidir. Özellikle, stok yenilemeyi tetikleyen işlem çok büyükse, Q miktarında bir yenileme, stok seviyesini yeniden sipariş noktasının üzerine çıkarmayabilir. Bu tür durumlarda, tamsayı katları şeklinde sipariş verilerek stok seviyesinin yeterince artırılması sağlanabilir [45].

Şekil 3.4'de (s,Q) sisteminin stok geometrisi verilmiştir.



Açıklama: - - - Stok seviyesi

Net stok miktarı ya da hem stok seviyesi ve net stok miktarı (eğer ikisi de eşitse)

Şekil 3.4 (s,Q) sistemi [50]

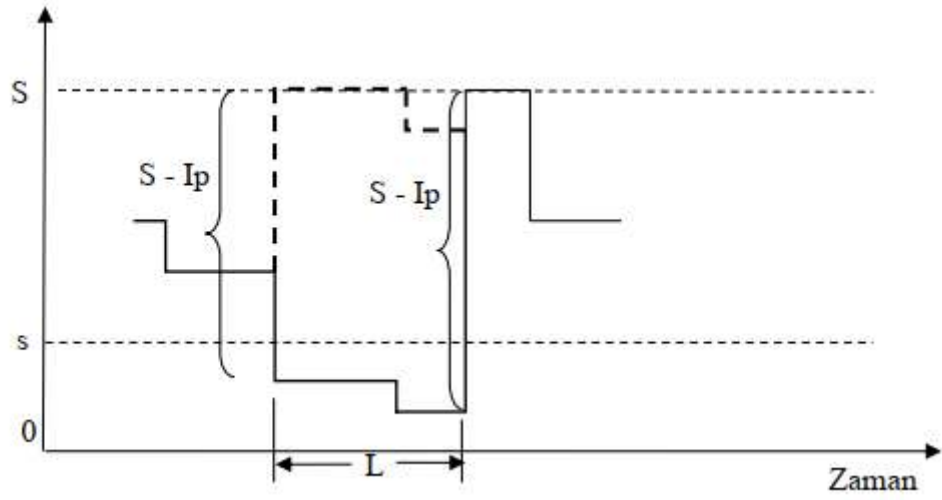
3.3.2.1.2 (s,S) Sistemi

Bu sistem de sürekli gözden geçirme esasına dayanır ve (s,Q) sisteminde olduğu gibi, stok seviyesi sipariş noktası s veya altına indiğinde yenileme yapılır. Ancak, (s,Q) sisteminden farklı olarak, stok seviyesini S sipariş seviyesine yükseltecek kadar değişken bir yenileme miktarı kullanılır. (s,S) sistemi genellikle min-maks sistemi olarak adlandırılır, çünkü stok pozisyonu, yeniden sipariş noktasının altındaki kısa süreli düşüşler dışında, her zaman minimum s değeri ile maksimum S değeri arasında kalır [45].

(s,S) sistemi stok seviyesi herhangi bir zamanda belirli bir seviyeye (s) veya altına düştüğünde, stok seviyesini belirli bir noktaya (S) yükseltmek için yeterli miktarda sipariş verilir [52].

Bu sistemin bir dezavantajı, sipariş miktarının değişken olmasıdır. Tedarikçiler daha sık hata yapabilir ve sabit bir sipariş miktarının getirdiği öngörülebilirliği tercih ederler [45].

Şekil 3.5'te (s,S) sisteminin stok geometrisi verilmiştir.



Açıklama: - - - Stok seviyesi

_____ Net stok miktarı ya da hem stok seviyesi ve net stok miktarı (eğer ikisi de eşitse)

Şekil 3.5 (s,S) sistemi [50]

3.3.2.2 Periyodik Gözden Geçirme Sistemleri

Periyodik incelemede, isminden de anlaşılacağı gibi, stok pozisyonu yalnızca her R zaman biriminde kontrol edilir; inceleme aralıkları arasında stok seviyesinin durumu hakkında önemli bir belirsizlik olabilir. Envanter kararları genellikle kapanış saatlerinden sonra bilgisayar tarafından yapılır. Bu nedenle, envanter, yeniden sipariş kararları açısından bakıldığında, inceleme süresi R genellikle bir gündür. Örneğin, bazı küçük firmalar envanteri incelemek ve satın alma kararlarını vermek için belirli bir kişiyi görevlendirir. Çoğu zaman, bu kişi diğer işleriyle o kadar meşguldür ki, bu görevleri yerine getirmek için haftada sadece bir öğleden sonrasını ayırabilir. Bu örneklerde görüldüğü gibi, fiziksel kısıtlamalar veya zaman sınırlamaları genellikle inceleme aralığını belirleyen faktörlerdir [45].

Periyodik takip sistemlerinde, temel olarak (R,S) ve (R,s,S) olarak bilinen iki stok politikası mevcuttur [52].

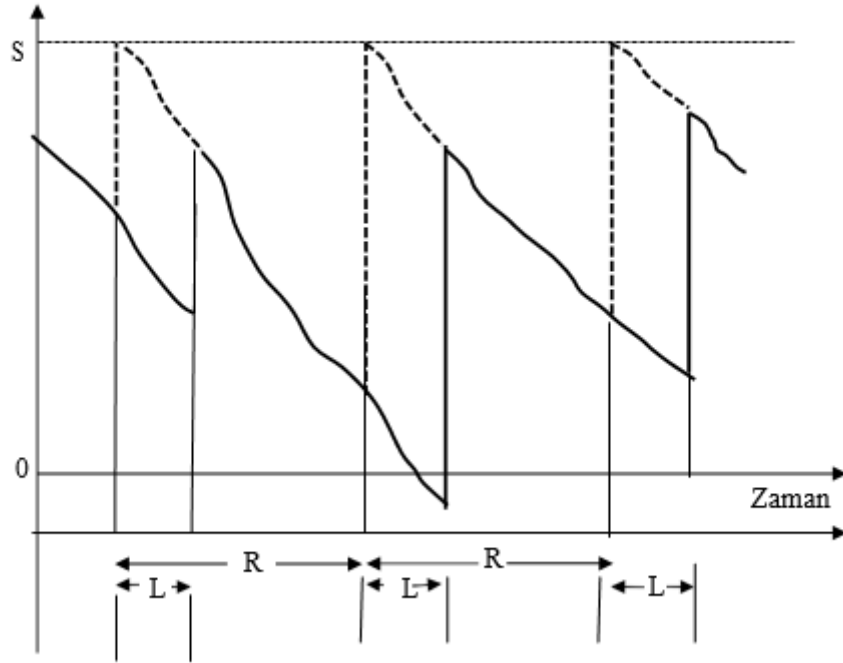
Ürünler aynı ekipman üzerinde üretilebilir, aynı tedarikçiden temin edilebilir veya aynı taşıma yöntemiyle gönderilebilir. Bu durumların her birinde, tedarik koordinasyonu avantajlı olabilir. Bu tür durumlarda, periyodik inceleme özellikle faydalıdır çünkü koordineli bir gruptaki tüm ürünler için aynı inceleme aralığı uygulanabilir. Periyodik gözden geçirme, ilgili personelin iş yükünü daha makul bir şekilde tahmin etmeye de olanak tanır. Buna karşın, sürekli gözden geçirme durumunda neredeyse her an sipariş yenileme kararı alınabilir, bu da iş yükünü daha az öngörülebilir hale getirir. Rastgele yerine düzenli (periyodik) bir model, genellikle malzeme yöneticileri için daha çekicidir. Sürekli gözden geçirmenin bir diğer dezavantajı, maliyetlerin ve hataların kontrol edilmesi açısından periyodik gözden geçirmeye kıyasla genellikle daha pahalı olmasıdır. Bu durum, özellikle birim zaman başına çok sayıda işlemin yapıldığı ürünlerin hızlıca listelenmesi gerektiğinde geçerlidir. Günümüzde, tek noktadan veri toplama sistemleri sayesinde maliyetler ve hatalar önemli ölçüde azaltılmıştır. Çok yavaş hareket eden ürünler için ise, sürekli gözden geçirme düşük gözden geçirme maliyetleri sağlar çünkü güncellemeler yalnızca işlem gerçekleştiğinde yapılır. Bununla birlikte, periyodik gözden geçirme, yavaş hareket eden ürünlerin bozulmasını (veya akışını) tespit etmede sürekli gözden geçirmeye göre daha etkili olabilir [45].

3.3.2.2.1 (R,S) Sistemi

Stok durumu, R süreli periyodik aralıklarla düzenli olarak gözden geçirilir. Gözden geçirme sırasında, stok seviyesini belirli bir düzeye (S) yükseltmek için yeterli miktarda sipariş verilir [52].

Zaman aralıklarıyla düzenli olarak yapılan periyodik inceleme sistemi, ilgili kalemlerin tedariklerinin daha iyi koordine edilmesini sağlar. Örneğin, uluslararası siparişlerde nakliye maliyetlerini kontrol altında tutmak için genellikle bir nakliye konteynerinin tamamen doldurulması gereklidir. Periyodik inceleme sistemi, bu koordinasyonu sağlayarak önemli tasarruflar sunabilir. Ayrıca, (R,S) sistemi, her R birimlik zaman diliminde stok seviyesini S seviyesine ayarlamak için düzenli bir fırsat sağlar; bu özellik, talep modelinin zamanla değişmesi durumunda çok değerlidir. Ancak (R,S) sisteminin taşıma maliyetleri, sürekli inceleme sistemlerine göre genellikle daha yüksektir [45].

Şekil 3.6'da (R,S) sisteminin stok geometrisi verilmiştir.



Açıklama: - - - Stok seviyesi

_____ Net stok miktarı ya da hem stok seviyesi ve net stok miktarı (eğer ikisi de eşitse)

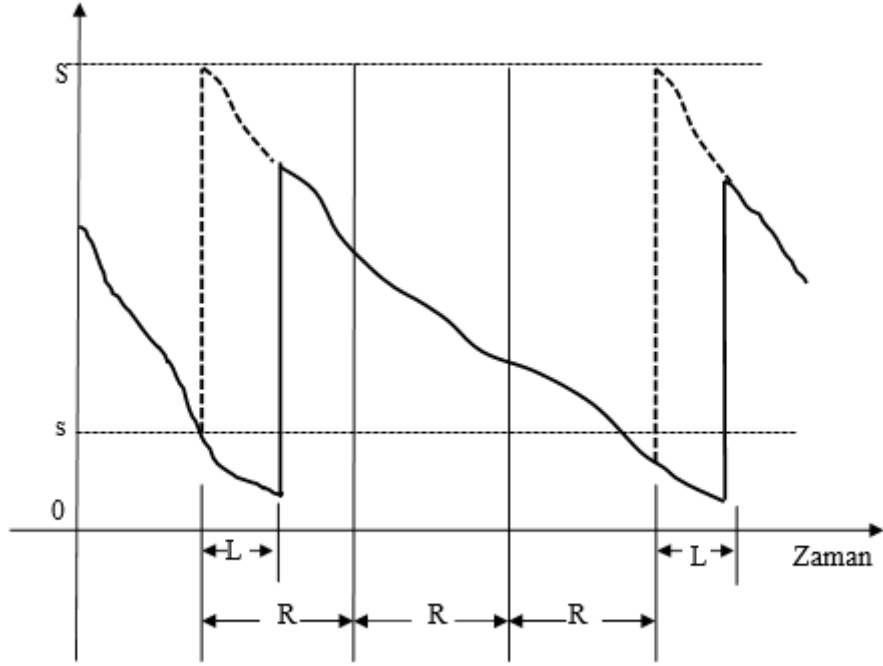
Şekil 3.6 (R,S) sistemi [50]

3.3.2.2.2 (R,s,S) Sistemi

Stok pozisyonu R uzunluğundaki belirli aralıklarla, düzenli olarak gözden geçirilir. Stok seviyesi s seviyesinin altına düştüğünde stok seviyesini S seviyesine getirmek için yeterli miktarda sipariş verilir. Stok seviyesi s'nin üzerindeyse sipariş verilmez. Bu strateji periyodik gözden geçirme (s,S) politikası olarak bilinir [51].

(s,S) sistemi, $R = 0$ olan ve (R,S) sistemi $s = S - 1$ olan özel durumdur. (R,s,S) sistemi ise (s,S) sisteminin periyodik bir versiyonudur. Ayrıca, (R,S) durumu $s = S - 1$ ile (s,S) sistemi periyodik bir uygulama olarak da kabul edilebilir [44].

Şekil 3.7'de (R,s,S) sisteminin stok geometrisi verilmiştir.



Açıklama: - - - Stok seviyesi

_____ Net stok miktarı ya da hem stok seviyesi ve net stok miktarı (eğer ikisi de eşitse)

Şekil 3.7 (R,s,S) sistemi [50]

Tablo 3.1’de çalışmada incelenen stok kontrol politikalarına ait karşılaştırmalar verilmiştir.

Tablo 3.1 Stok kontrol politikalarının karşılaştırılması [53]

Stok Kontrol Politikası	Gözden Geçirme Tipi	Yeniden Sipariş Noktası	Sipariş Miktarı	Sipariş Miktarı Tipi
(s,Q)	Sürekli	$I_p \leq s$	Q	Sabit
(s,S)	Sürekli	$I_p \leq s$	$Q = S - I_p$	Değişken
(R,S)	Periyodik	$I_p \leq S$	$Q = S - I_p$	Değişken
(R,s,S)	Periyodik	$I_p \leq s$	$Q = S - I_p$	Değişken

Verilen stok kontrol politikalarının uygulanmasındaki en büyük zorluklardan biri stok kontrol politikası parametrelerinin belirlenmesidir. Optimum stok kontrol parametrelerini belirlemek için deterministik sistemlerde ekonomik sipariş miktarı formülleri geliştirilmiştir, stokastik sistemlerde ise matematiksel modelleme, simülasyon modelleme, sezgisel veya metasezgisel algoritmalar kullanılmıştır. Bu tezde stokastik sürekli gözden geçirme (s,S) stok kontrol politikasının optimum stok kontrol parametrelerinin belirlenebilmesi için metasezgisel algoritma tabanlı simülasyon optimizasyonu çalışması yapılmıştır.



4. SİMÜLASYON VE STOK KONTROLÜ

Bazı basit modeller matematiksel yöntemler kullanılarak analitik çözümlerle değerlendirilebilir. Ancak, çoğu gerçek dünya sistemi çok karmaşıktır ve bu nedenle analitik olarak incelenemez. Bu tür karmaşık modeller, bilgisayar kullanılarak sayısal olarak değerlendirilir ve modelin istenen özelliklerini tahmin etmek için veri toplanır; bu sürece simülasyon denir. Simülasyon, gerçek dünyadaki sistemlerin veya süreçlerin bilgisayar ortamında bir modelinin oluşturulması ve bu model üzerinde farklı senaryoların denenmesi işlemidir. Bu yöntem, sistemin gerçek dünya koşullarındaki davranışlarını taklit ederek, çeşitli koşullardaki etkileri analiz etmeye olanak tanır. Özellikle teorik analizlerin veya pratik deneylerin uygulanabilir olmadığı durumlarda simülasyon oldukça kullanışlı bir araçtır [54].

Genel amaçlı programlama dilleri (FORTRAN, Visual Basic, C/C++, Java gibi) kullanarak simülasyon modelini sıfırdan uygulanabilir. Özel kütüphaneler olmadan simülasyon modellemek yüksek düzeyde programlama becerisi gerektirir. Simülasyonların tekrarlayıcı doğası, olayları sıralı olarak yönetme, rastgele sayılar üretme ve istatistik toplama gibi işlevler için bilgisayar kütüphanelerinin geliştirilmesini sağlamıştır [55].

Bilgisayarların hesaplama gücü ve depolama kapasitesi, özel simülasyon dillerinin geliştirilmesine olanak sağlamıştır. Bu diller, sürekli veya ayrık simülasyonlar için kullanılırken, bazıları hem sürekli hem de ayrık modelleme için kullanılabilir. Tüm simülasyon dilleri belirli standart programlama olanakları sunmaktadır, ancak bu olanakları kullanma şekli dilden dile değişmektedir. Bazı diller programlama odaklıdır (örneğin, SIMSCRIPT), bazı diller ise "sürükle ve bırak" ara yüzleri sunmaktadır (örneğin, ProModel, Arena) [55].

4.1 Simülasyon Yönteminin Çeşitleri

Kelton ve arkadaşları simülasyon modellerini 3 farklı boyutta sınıflandırmıştır [56]:

1. **Statik ve Dinamik Simülasyon:** Zamanın hiç bir rol oynamadığı modellere statik modeller denmektedir. Dinamik modellerde zamana göre farklı sonuçlar elde edilmektedir. Çoğu operasyonel model dinamik olup, Arena bu tür modeller için tasarlanmıştır.
2. **Sürekli ve Kesikli Simülasyon:** Kesikli bir modelde, değişim yalnızca belirli zaman noktalarında gerçekleşir yani sistem değişikliği belirli zaman periyotlarında yapılmaktadır. Sürekli bir modelde ise sistemin durumu zaman içinde sürekli olarak değişebilir. Aynı modelde hem kesikli hem sürekli değişkenler bulunabilir. Bu modellere karışık sürekli-kesikli modeller denmektedir.
3. **Deterministik ve Stokastik Simülasyon:** Simülasyon modeli herhangi bir olasılık içermiyorsa deterministik simülasyon modeli olarak tanımlanmaktadır. Simülasyon modelinde en az bir adet olasılık varsa stokastik simülasyon modeli olarak tanımlanmaktadır. Bir modelde hem deterministik hem de stokastik girişler bulunabilir. Arena, deterministik ve stokastik girişleri işleyebilir ve stokastik girişleri temsil etmek için birçok olasılık dağılımı ve süreç sunar.

4.2 Simülasyon Yöntemi Kullanmanın Avantajları ve Dezavantajları

Simülasyonun en büyük avantajı geleneksel yöntemlerle birçok kısıt altında çözülen veya problemin karmaşıklığı nedeniyle çözilemeyen gerçek hayat problemlerinin çözümünde kullanılabilmesidir [58]. Bunun haricinde simülasyon yöntemi kullanmanın bazı avantajları aşağıdaki gibi sıralanabilir [54]:

- Simülasyon, mevcut bir sistemin belirli işletim koşulları altında performansını tahmin etmeyi sağlamaktadır.
- Alternatif sistem tasarımları (veya bir sistem için alternatif işletim politikaları), hangi seçeneğin belirli bir gereksinimi en iyi şekilde karşıladığını görmek amacıyla simülasyon yöntemiyle karşılaştırılabilir.

- Simülasyon, deneysel koşullar üzerinde sistemin kendisiyle yapılan denemelere kıyasla çok daha iyi bir kontrol sağlamaktadır.
- Simülasyon, uzun süreli sistemleri kısa bir zamanda veya detaylı bir şekilde genişletilmiş zamanda incelemeyi mümkün kılar.

Simülasyon yöntemi kullanmanın bazı dezavantajları da vardır. Simülasyon yöntemi kullanmanın bazı dezavantajları aşağıdaki gibi sıralanabilir [54]:

- Simülasyon modelleri genellikle belirli bir sayıda alternatif sistem tasarımını karşılaştırmada daha etkilidir, ancak optimizasyon konusunda aynı derecede başarılı olmayabilirler. Öte yandan, uygun bir analitik model mevcutsa veya kolayca geliştirilebiliyorsa, bu model genellikle çeşitli giriş parametre setleri için modelin kesin gerçek özelliklerini sağlamada daha avantajlıdır.
- Simülasyon modellerinin geliştirilmesi genellikle pahalı ve zaman alıcıdır.
- Simülasyon çalışmasından elde edilen büyük veri hacmi veya gerçekçi animasyonların etkisi, sonuçlara gereğinden fazla güvenme eğilimi yaratabilir. Eğer model, incelenen sistemi geçerli bir şekilde temsil etmiyorsa, simülasyon sonuçları ne kadar etkileyici görünürse görünsün, gerçek sistem hakkında pek faydalı bilgi sağlamaz.

4.3 Stok Kontrolünde Simülasyon

Gerçek hayat stok kontrol problemleri karmaşık bir problem olarak ortaya çıkar. Dinamik yapıya sahip olan firmalarda belirsizlik ve gecikmeler stoktaki bazı ürünlerin yığılmasına veya ihtiyaç duyulan zamanda yeterli miktarda bulunmamasına neden olmaktadır. Bu sebeple firmaların rekabetçi koşullarda varlığını sürdürebilmesi için stok kontrol fonksiyonlarını rasyonel temellere bağlayarak yürütmesi gerekmektedir. Bir stok kontrol sisteminde talep ve tedarik süresi gibi kontrol dışı değişkenlerin optimum değerleri bulunmaya çalışılmaktadır. Optimal değerlerin belirlenmesi için firmaların gerçek sistemleri üzerinde tecrübeye dayalı olarak

denemeler yapması ve elde edilen sonuçlara göre yeni politikaların belirlenmesi maliyetli, tehlikeli ve uzun bir süreçtir. Bu nedenle gerçek sistemin modellenip, model üzerinden denemelerin yapılması ve kısa sürede sonuçların alınması gereklidir [57].

Yukarıda belirtilen sebeplerden dolayı bu tezde, karmaşık sistemlerin analizinde yaygın bir yöntem olan simülasyon modelleme kullanılmıştır [59].



5. SİMÜLASYON OPTİMİZASYONU

Bir simülasyon modelinin analizinde nihai amaç, temel çıktı performans ölçütünü optimize eden girdi faktörlerinin kombinasyonunu bulmaktır. Örneğin, kar veya maliyet gibi doğrudan ekonomik öneme sahip bir çıktıyı, girdi faktörlerinin tüm olası değerleri üzerinden maksimuma çıkarmak veya minimuma indirmek olabilir. Simülasyon optimizasyonu probleminin amaç fonksiyonu değeri simülasyon tarafından bulunmaktadır. Simülasyon optimizasyonu açısından simülasyon modeli giriş parametrelerini çıkış parametrelerine dönüştüren bir mekanizmadır [54]. Simülasyon optimizasyonu, simülasyon optimizasyonu için klasik yaklaşımlar ve simülasyon optimizasyonu için metasezgisel yaklaşımlar olarak incelenebilir [60, 61].

5.1 Simülasyon Optimizasyonu için Klasik Yaklaşımlar

Fu, 4 farklı ana simülasyon optimizasyonu yaklaşımı tanımlar:

- 1. Stokastik yaklaşım (gradyan tabanlı yaklaşımlar):** Stokastik yaklaşım algoritmaları, deterministik optimizasyonun gradyan arama yöntemlerini taklit etmeye çalışır. Bu yöntemde dayanan prosedürler, arama yönünü belirlemek için amaç fonksiyonunun eğimini tahmin etmeye odaklanır. Stokastik yaklaşım, en dik iniş gradyanı aramasıyla yakın bir ilişkiye sahip olduğundan, özellikle sürekli değişkenli problemler için uygundur [60].
- 2. Yanıt yüzeyi yöntemi:** Bu yöntem, metamodeller oluşturma ilkesine dayanır ancak daha yerelleştirilmiş bir yaklaşımla çalışır. "Yerel yanıt yüzeyi," arama yönünü belirlemek için kullanılır (örneğin, tahmini eğime doğru hareket etmek) ve bu süreç tekrarlanır. Başka bir deyişle, metamodeller amaç fonksiyonunu tüm çözüm uzayında karakterize etmeye çalışmaz; bunun yerine, araştırmanın hali hazırda incelenmiş olduğu yerel bölgeye odaklanır [60].
- 3. Rastgele arama algoritması:** Rastgele arama, sonsuz parametre uzayında çalışabilme yeteneğine sahiptir. Kontrol edilebilir faktörlerin her biri için belirlenen üst ve alt sınırlar, genel bir arama alanını tanımlar. Bu teknik, bu genel arama alanından rastgele noktalar seçerek işlemi gerçekleştirir [62].

4. Örnek yol optimizasyonu: Deterministik sürekli optimizasyon problemleri için geliştirilen bilgi ve deneyimden yararlanan bir metodolojidir. Buradaki temel amaç, w 'nin örnek yolun boyutu olduğu ve w rastgele değişkene dayalı olan deterministik bir fonksiyonu optimize etmektir. Simülasyon bağlamında, giriş faktörlerinin farklı değerleri üzerinden tepkiyi hesaplamak için ortak rastgele sayı yöntemi kullanılır; bu, aynı örnek yolunu sağlama amacı taşır. Örnek yol optimizasyonu, bulduğu tahmini optimal çözümün bir simülasyon modeliyle elde edilen örnek bir yol üzerinden oluşturulmuş deterministik bir çözüm olmasına dayanır. Genellikle, yaklaşık optimizasyon probleminin orijinal optimizasyon problemine yakın olabilmesi için w 'nin büyük olması gereklidir [60].

5.2 Simülasyon Optimizasyonu için Metasezgisel Yaklaşımlar

Metasezgisel algoritmalar, geniş bir yelpazedeki optimizasyon problemleri için etkili çözümler sunan optimizasyon teknikleridir. Diğer optimizasyon yöntemlerinden farklı olarak türevsizdirler ve arama uzayında türev hesaplamasına ihtiyaç duymazlar, bu da onları daha basit, esnek ve yerel optimumlardan kaçınma konusunda daha yetenekli kılar. Metasezgisel algoritmaların stokastik doğası, rastgele sonuçlar üreterek optimizasyon sürecini başlatmalarına olanak tanır, bu da erken yakınsamayı önleyerek arama alanını etkili bir şekilde incelemelerini sağlar. Keşif ve kar arasında denge kurarak, ilgi çekici bölgelerde detaylı aramalar yapar ve bu bölgelerde yerel aramalar gerçekleştirirler. Bu algoritmalar, çok yönlülük ve esneklikleri sayesinde mühendislik ve bilim gibi çeşitli alanlardaki optimizasyon problemlerine uyacak şekilde kolayca değiştirilebilir. Metasezgisel algoritmalar, optimizasyon problemlerini çözmeye yönelik farklı fikirlerden ilham alarak genel bir arama algoritması sınıfı oluşturur ve çeşitli problem türlerinde kullanılabilir. Metasezgiseller, karmaşık ve doğrusal olmayan sorunları ele alabilen, küresel optimumu bulma zorunluluğu olmadan iyi çözümler sunabilen bir optimizasyon algoritmaları sınıfıdır. Geleneksel optimizasyon tekniklerinden farklı olarak, amaç fonksiyonunu doğrusallaştırmaya veya türev ve gradyan kullanmaya gerek yoktur. Bu algoritmalar, gelişmiş stratejiler kullanarak yönetim, planlama, mimarlık, mühendislik, sağlık ve lojistik gibi alanlarda zor optimizasyon problemlerini etkili bir şekilde çözerler. Metasezgisel yöntemlerin

yaygın olarak kullanılması, zor problemleri çözmedeki başarısından kaynaklanmaktadır. Açık bir denklem tabanlı modelin geliştirilemediği durumlarda özellikle yararlıdırlar ve problem arama uzayını derinlemesine keşfetme yetenekleri sayesinde optimal çözümler bulma olasılıkları daha yüksektir. Yıllar içinde, evrime dayalı, doğadan ilham alan, fizik tabanlı ve stokastik algoritmalar geliştirilmiştir. Çoğu nüfus tabanlı olan bu algoritmalar, birçok çözümü aynı anda ele alarak optimal çözümlere ulaşmayı amaçlar. Gerçek dünya problemleri genellikle karmaşıklık, doğrusal olmama, birbiriyle çelişen hedefler ve çeşitli kısıtlamalarla karakterize edilir. Bu nedenle, bu tür problemler için en uygun çözümleri bulmak zor olabilir ve bazı durumlarda optimal çözümler mevcut olmayabilir [63].

Metasezgisel yöntemler amaç fonksiyonu hesaplamak için simülasyondan yardım almıştır. Bu yöntemler arasında genetik algoritma, benzetimli tavlama, tabu arama ve sinir ağları bulunur. Bu yöntemler genellikle deterministik bağlamda kombinatoriyal optimizasyon için tasarlanmış olup, kesin yakınsama garantileri yoktur. Ancak, simülasyon optimizasyonuna uygulandıklarında oldukça başarılı olmuşlardır [64].

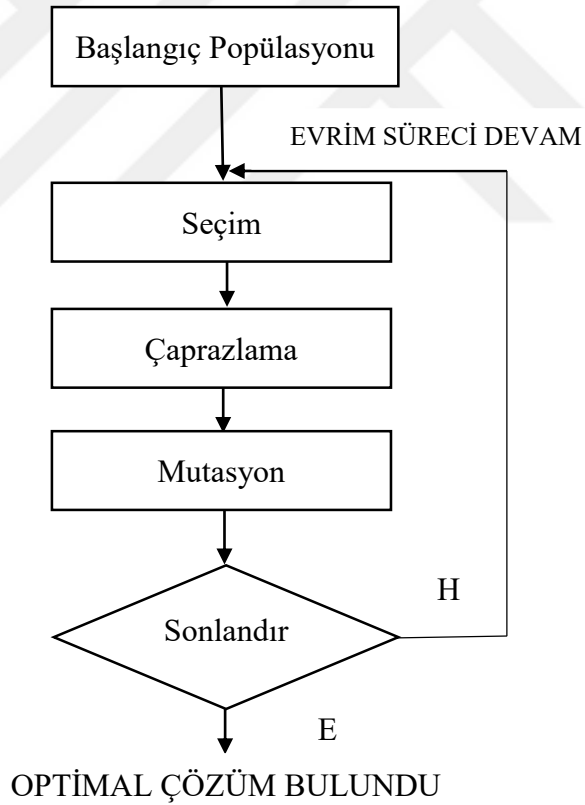
Bu çalışmada bir imalat sisteminin stok kontrol politikası optimizasyonu için genetik algoritma ve benzetimli tavlama algoritmalarının kullanıldığı bir simülasyon optimizasyonu yaklaşımı önerilmiştir. Sonrasında genetik algoritma ve benzetimli tavlama algoritmaları için sonuçlar karşılaştırılmıştır. Aşağıdaki bölümlerde genetik algoritma ve benzetimli tavlama algoritması hakkında detaylı bilgi sunulmaktadır.

5.2.1 Genetik Algoritma

Genetik algoritmalar, John Holland, meslektaşları ve Michigan Üniversitesi'ndeki öğrencileri tarafından geliştirilmiştir. Genetik algoritmalar, doğal seçim ve genetik mekaniklerine dayanan arama algoritmalarıdır. Bu algoritmalar, en uygun olanların hayatta kalması prensibini, yapılandırılmış ancak rastgele bilgi alışverişini kullanarak birleştirir. Her nesilde, eski neslin en uygun olanlarından parçalar kullanılarak yeni bir dizi yapay yaratık oluşturulur; zaman zaman yeni bir parça da deneme amaçlı eklenir. Rastgele olmalarına rağmen, genetik algoritmalar basit bir rastgele ilerleme değildir. Geçmiş bilgileri etkili bir şekilde kullanarak,

beklenen iyileştirilmiş performansla, yeni arama noktaları üzerinde spekülasyon yaparlar [65].

GA, belirli bir problem için potansiyel çözümleri temsil eden bireylerin (kromozomların) bir popülasyonu üzerinde çalışır. Her kromozoma, amaç (fitness) fonksiyonunun sonucuna göre bir uygunluk değeri atanır. Seçim mekanizması, yeni bir popülasyon oluşturulduğunda daha iyi amaç fonksiyonu değerine sahip bireyleri, daha kötü olanlardan daha sık üremeleri için tercih eder. Rekombinasyon (Recombination), ebeveyn bilgilerinin torunlarına geçirilmesinde karışmasına izin verirken, mutasyon popülasyonda yeniliklerin ortaya çıkmasını sağlar. Genellikle, başlangıç popülasyonu rastgele başlatılır ve evrim süreci önceden belirlenmiş bir iterasyon sayısından sonra durdurulur [66]. Şekil 5.1’de genetik algoritmanın çalışma prensibi verilmiştir.



Şekil 5.1 Genetik algoritma çalışma prensibi [67]

Tablo 5.1’de genetik algoritma terimlerinin açıklaması verilmiştir.

Tablo 5.1 Genetik algoritma terimleri [68]

Genetik Algoritma	Açıklama
Kromozom	Çözüm
Gen (Bit)	Özellik
Lokus	Özelliğin pozisyonu
Allel	Özelliğin değeri
Fenotip	Deşifre edilmiş çözüm
Genotip	Kodlanmış çözüm

Genetik algoritmaların yapısını anlayabilmek için aşağıdaki kavramların açıklanması faydalı olacaktır [1].

- Parametre kodlama
- Başlangıç popülasyonunun oluşturulması
- Uygunluk fonksiyonu
- Genetik operatörlerin tanımı ve uygulanması
- Sonlandırma kriteri

5.2.1.1 Parametre Kodlama

GA araştırma uzayındaki değerler ile işlem yapmaz, üreme işlemlerinin gerçekleştirilebilmesi için kodlanmış diziler üzerinden işlem yapmaktadır [69]. GA orijinal formülasyonu ikili kodlamaya dayanmaktadır. İkili kodlamada, her kromozom 0 veya 1 bitlerinden oluşan bir dizedir. Bu kodlama türü, az sayıda allel ile bile çok sayıda olası kromozom sağlar. Ancak, bu kodlama birçok problem için doğal olmayabilir ve bazen çaprazlama ve/veya mutasyondan sonra düzeltmeler yapılması gerekebilir. İkili kodlamaya alternatif bir yöntem, gri kodlamadır. Gri kodlaması, her ardışık sayının sadece bir bit farklı olduğu ikili kodlamaya benzer. Doğrudan değer kodlaması ise gerçel sayılar gibi karmaşık değerlerin kullanıldığı problemlerde kullanılabilir. Bu tür problemler için ikili kodlamanın kullanımı zordur. Değer kodlamasında her kromozom, probleme bağlı olarak gerçel sayılar, karakterler veya diğer nesneler gibi değerlerin bir dizisidir. Permütasyon kodlaması ise her

kromozomun bir dizideki bir pozisyonu temsil eden sayılardan oluşan bir dizedir. Permütasyon kodlaması sıralama problemleri için kullanışlıdır. Ağaç kodlaması genellikle programları veya ifadeleri evrimleştirmek için kullanılır, yani genetik programlama için. Ağaç kodlamasında her kromozom, bir programlama dilindeki fonksiyonlar veya komutlar gibi nesnelerin bir ağacıdır [1].

5.2.1.2 Başlangıç Popülasyonunun Oluşturulması

Rastgele başlangıç popülasyonu oluşturmak, genetik algoritmalarda kullanılan en yaygın yöntemidir. Bu yöntem, çözüm uzayının geniş bir bölgesini keşfetmek için başlangıçta geniş bir çözüm çeşitliliği sağlar [70]. Başka bir yöntem, her birey için bir dizi başlatma işlemi gerçekleştirmek ve ardından en yüksek performans gösteren değerleri seçmektir. Alternatif olarak, yaklaşık çözümleri bulmak amacıyla diğer yöntemlerle tahminler yapılabilir ve algoritma bu noktalardan başlatılabilir [71].

5.2.1.3 Uygunluk Fonksiyonu

Genetik algoritmada, her aşamada uygunluk kriteri popülasyona uygulanır. Bu kriter, ya daha ileri hesaplamalar için seçilmesi gereken popülasyon kısmını değerlendirmek ya da algoritmayı durdurmak için kullanılır. Uygunluk fonksiyonu, uygulamaya özeldir ve optimum sonuçları elde etmek için uygulamanın gereksinimlerine göre farklı kısıtlamaları içerir. Genetik algoritmanın çalışma prensibi, uygunluk fonksiyonuna dayanmaktadır [72].

5.2.1.4 Genetik Operatörler

Kromozomlardaki genler üç ana operatör olan seçim, çaprazlama ve mutasyon operatörü tarafından şekillendirilebilir.

5.2.1.4.1 Seçim

Seçim, algoritmayı çözüme yönlendiren bileşendir ve yüksek uygunluk değerine sahip bireyleri düşük uygunluk değerine sahip olanlara tercih eder. Seçim işlemi deterministik olabilir, ancak çoğu uygulamada rastgele bileşenler içerir [69].

Uygulamada çok çeşitli seçim yöntemi bulunmaktadır. Literatürde bulunan ve çoğunlukla kullanılan seçim mekanizmaları genellikle altı ana sınıfa ayrılır [73]:

- Rassal Seçim
- Rulet Seçimi
- Beklenen Değer Seçimi
- Sıralı Seçim
- Turnuva Seçimi
- Elitizm Seçimi

1. Rassal Seçim: Popülasyondaki ebeveynlerin sayısı kadar, popülasyondaki kromozomlardan rastgele seçim yapar [74].

2. Rulet Seçim: Genetik algoritmalarda rulet tekerleği seçim yöntemi, bireylerin uygunluk fonksiyonuna göre seçilerek kopyalanmasını sağlar. Bu yöntemde, her bireyin seçim olasılığı uygunluk fonksiyonuna (f) bağlıdır. Uygunluk fonksiyonu, geçerlilik, karlılık gibi ölçülmesi istenen herhangi bir unsur olabilir. Mevcut popülasyondaki bireyler, rulet çemberinde uygunluklarına göre büyüklüğü ayarlanmış dilimler olarak düşünülebilir. Rulet tekerleği seçiminde, her bireyin seçim olasılığı $p(i)$, bireyin uygunluk değeri f_i 'nin toplam uygunluk değerine oranı olarak hesaplanır. Bu oran, bireyin rulet tekerleğinde kapladığı dilimin büyüklüğünü belirler. Bireyin seçim olasılığı $p(i) = f(i) / \sum_{j=1}^n f_j$ şeklinde hesaplanır. Rulet seçimi algoritması aşağıdaki şekildedir:

Adım 1: N tane birey içeren popülasyonda t zamanındaki nesil için uygunluk değerlerini f_i ($i=1,...,N$) ve tüm popülasyonun toplam uygunluk değerini $f_{toplam} = \sum_{j=1}^N f_j$ hesapla.

Adım 2: $[0,1]$ aralığında rastgele bir sayı $rand()$ yarat ve $r = rand() \times f_{toplam}$ hesapla.

Adım 3: $\sum_{j=1}^k f_j \geq r$ şeklinde en az değerli k 'yı elde et ve k . bireyi $t+1$ zamanındaki nesil için seç.

Adım 4: Seçilen birey sayısı N 'e gelene kadar 2. ve 3. adımları tekrar et [73].

3. **Beklenen Değer Seçimi:** S_n alt bireylerinin, bir sonraki nesilde beklenen sayısını belirlemek için bir yöntem tanımlamaktadır. Bu yaklaşım, N_n 'nin tam ve ondalık kısımlarını kullanarak, kaç tane alt bireyin bir sonraki nesilde korunacağını belirler.

$$N_n = f(S_n) / \sum_{n=1}^N f(S_n) * N \quad (5.1)$$

Burada, $f(S_n)$ S_n bireyinin uygunluğunu (fitness) temsil eder. N_n 'nin tam kısmı sayısı kadar popülasyon bir sonraki nesilde korunur [75].

4. **Sıralı Seçim:** Rulet tekerliği seçim yöntemi, uygunluklar (fitness) çok fazla farklılık gösterdiğinde sorunlar yaşayabilir. Örneğin, en iyi kromozom uygunluğu, tüm rulet çarkının %90'ını oluşturuyorsa, diğer kromozomların seçilme şansı çok az olacaktır. Sıralama seçimi (Rank seçim), önce popülasyonu uygunluğa göre sıralar ve sonra her kromozom bu sıralamadan bir uygunluk değeri alır. En kötü kromozom 1 uygunluk değeri alırken, ikinci en kötü 2 alır ve bu şekilde devam eder; en iyi kromozom ise popülasyondaki kromozom sayısı olan N kadar uygunluk değeri alır. Bu sayede tüm kromozomların seçilme şansı olur. Ancak bu yöntem daha yavaş bir yakınsamaya yol açabilir çünkü en iyi kromozomlar diğerlerinden çok fazla farklılık göstermez [76].

5. **Turnuva Seçimi:** Bu seçim şeması, popülasyondan rastgele bazı bireyler seçilerek ve bu grubun en iyi bireyinin ara popülasyona kopyalanmasıyla uygulanır; bu işlem, çiftleşme havuzu tamamlanana kadar tekrarlanır. Turnuvalar genellikle iki birey arasında yapılır. Daha büyük turnuvalar da, popülasyon boyutuna kıyasla çok büyük olmamak kaydıyla, farklı grup boyutlarıyla kullanılabilir. Turnuva seçimi, popülasyonun sıralanması gerekmeyeceği için çok verimli bir şekilde uygulanabilir. Turnuva prosedürü, uygunlukları yeniden eşlemeden çiftleşme havuzunu seçer. Turnuva boyutu ayarlanarak seçim baskısı istenildiği gibi artırılabilir veya azaltılabilir. Daha

büyük turnuvalar, seçim baskısını artırma etkisine sahiptir, çünkü ortalamanın altındaki bireylerin bir yarışmayı kazanma şansı pek yoktur [77].

6. **Elitizm Seçimi:** Yeni popülasyon oluştururken çaprazlama ve mutasyon kullanıldığında, en iyi kromozomu kaybetme olasılığımız yüksektir. Elitizm, en iyi kromozomun (veya birkaç en iyi kromozomun) yeni popülasyona ilk olarak kopyalanmasını sağlayan bir yöntemdir. Geri kalan işlemler klasik yöntemle yapılır. Elitizm, en iyi bulunan çözümün kaybedilmesini önlediği için genetik algoritmanın performansını çok hızlı bir şekilde artırabilir [76].

5.2.1.4.2 Çaprazlama

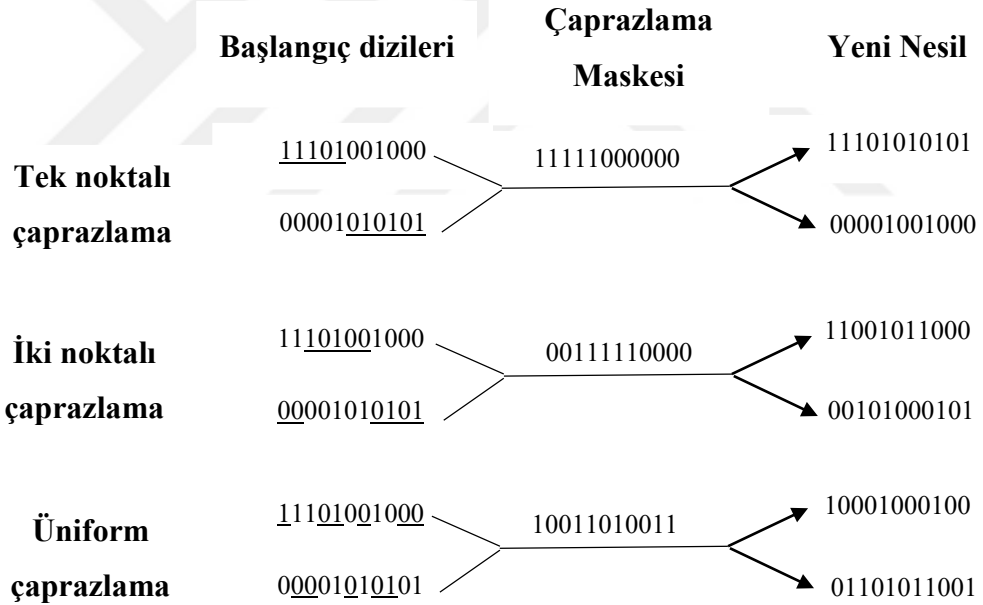
GA sonraki nesillerin oluşturulması, mevcut popülasyonun seçilen üyelerini yeniden birleştiren ve mutasyona uğratan bir dizi operatör tarafından belirlenir. Çaprazlama operatörü, iki ebeveyn dizisinden iki yeni yavru üretir; her ebeveyn den seçilen bitleri kopyalayarak yeni bireyler oluşturur. Her yavrudaki i konumundaki bit, ebeveynlerden birinin i konumundaki bitinden kopyalanır. Hangi ebeveynin i konumundaki biti vereceği, çaprazlama maskesi adı verilen ek bir dizi tarafından belirlenir. Çaprazlama operatörünün tek noktalı çaprazlama, iki noktalı çaprazlama ve üniform (düzgün) çaprazlama olarak üç türü vardır [76].

1. **Tek Noktalı Çaprazlama:** Tek noktalı çaprazlama, iki ebeveyni alır ve ebeveynlerin bölüneceği noktayı rastgele seçer. Seçilen noktadan sonraki kısımlar, iki ebeveyn arasında değiştirilerek iki çocuk oluşturulur [1].
2. **İki Noktalı Çaprazlama:** Genotipin genlerini yeniden birleştirmenin daha karmaşık bir yolu, çok noktalı çaprazlama tekniği kullanmaktır. En yaygın çok noktalı çaprazlama tekniği, iki noktalı çaprazlamadır. İki noktalı çaprazlamada, bir kromozom içinde rastgele iki çaprazlama noktası seçilir ve bu noktalar arasındaki iki ebeveyn kromozomu değiştirilerek iki yeni yavru üretilir [1].
3. **Üniform Çaprazlama:** Üniform çaprazlama, iki ebeveyn den alınan bitleri rastgele ve eşit şekilde birleştirir. Bu durumda, çaprazlama maskesi, her biti rastgele ve birbirinden bağımsız olarak seçilen rastgele bir bit dizisi olarak

oluşturulur [75]. Çok noktalı çaprazlama ile üniform çaprazlama arasındaki tek farklılık üniform çaprazlamada maske adı verilen ayrı bir dizinin kullanılmasıdır [78].

Şekil 5.2’de çaprazlama operatörleri için örnek verilmiştir.

Çaprazlama olasılığı, çaprazlama operatörünün uygulanma sıklığını kontrol eder. Her yeni popülasyonda, $P_c * N$ (popülasyon büyüklüğü) yapı çaprazlama işlemine tabi tutulur. Çaprazlama olasılığı ne kadar yüksekse, yeni yapılar popülasyona o kadar hızlı bir şekilde dahil edilir. Ancak, eğer çaprazlama olasılığı çok yüksekse, yüksek performanslı yapılar seçim iyileştirmeleri üretilmeden önce hızla elenebilir. Çaprazlama olasılığı çok düşükse, keşif oranının düşük olması nedeniyle arama durağanlaşabilir [79].



Şekil 5.2 Çaprazlama operatörleri [76]

5.2.1.4.3 Mutasyon

Yeniden kombinasyon operatörlerinin iki ebeveynin parçalarını birleştirerek yeni nesiller ürettiği gibi, ikinci bir operatör türü de tek bir ebeveyninden yeni nesiller üretir. Özellikle, mutasyon operatörü bit dizisinde küçük rastgele değişiklikler

yaparak, rastgele seçilen bir bitin değerini değiştirir. Mutasyon genellikle çaprazlamadan sonra gerçekleştirilir [76].

Mutasyon, genetik algoritmalarda arka plan operatörüdür. Temel amacı, kaybolan allelleri (bit değerleri) popülasyona yeniden kazandırmaktır. Bu mekanizma olmadan, GA şans eseri, belirli genlerin tüm popülasyonda ortak bir bit değerine erken yakınsaması nedeniyle arama alanının umut verici bölgelerini istemeden dışlayabilir. Mutasyon genellikle şu şekilde uygulanır: Her bireyin her biti, belirlenmiş bir olasılıkla (P_m) ters çevrilir. Mutasyon olasılığı için yaygın değerler [0,01, 0,001] aralığındadır [80].

Mutasyon, popülasyonun değişkenliğini artıran ikincil bir arama operatörüdür. Seçimden sonra, yeni popülasyondaki her yapının her bit pozisyonu, mutasyon olasılığı (P_m) ile eşit olasılıkla rastgele bir değişime uğrar. Sonuç olarak, her nesilde yaklaşık olarak $P_m * \text{popülasyon büyüklüğü} * \text{kromozom uzunluğu}$ kadar mutasyon meydana gelir. Düşük seviyede mutasyon, herhangi bir bit pozisyonunun tüm popülasyonda tek bir değere sonsuza dek yakınsamasını sağlar. Yüksek seviyede mutasyon ise temelde rastgele bir arama sağlar [79].

5.2.1.5 Durdurma Kriteri

Sonlandırma operatörü, genetik algoritmanın aramaya devam edip etmeyeceğini veya durdurup durdurmayacağını belirlemek için kullanılan bir işlemdir. Her nesil oluşturulduktan sonra genetik algoritma belirlenen sonlandırma kriterini kontrol ederek aramaya devam eder veya sonlandırır. Sonlandırma ile ilgili çeşitli yöntemler vardır başlıcaları aşağıda verilmiştir [73]:

1. **Nesil sayısı:** Evrim süreci, problem çözücü tarafından belirlenen maksimum nesil sayısına ulaştığında algoritma arama işlemini sonlandırır.
2. **Evrin Süresi:** Algoritmanın arama süresi, problem çözücü tarafından belirlenen maksimum süreyi aştığında işlem sonlanır.

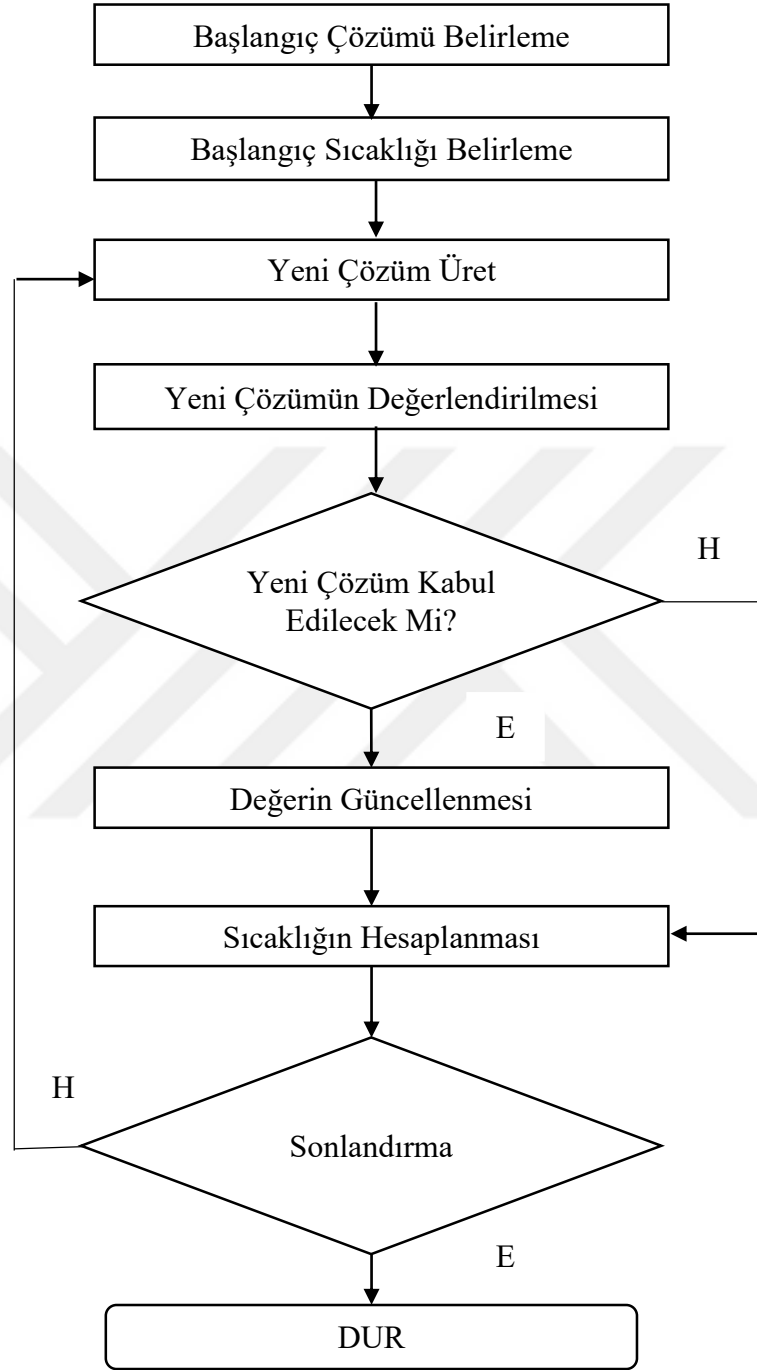
3. **Uygunluk Eşiği:** Minimizasyon problemlerinde, mevcut popülasyondaki en uygun değere sahip bireyin uygunluk değeri problem çözücü tarafından belirlenen eşiğin altına düşerse; maksimizasyon problemlerinde ise en uygun değere sahip bireyin uygunluk değeri belirlenen eşiğin üstüne çıkarsa algoritma arama işlemini sonlandırır.
4. **Popülasyon Yakınsaması:** Evrim süreci, popülasyon yakınsamaya başladığında sona erer.
5. **Gen Yakınsaması:** Kromozomlardaki genlerin değeri, problem çözücü tarafından belirlenen yüzdeye kadar yakınsamaya başladığında evrim süreci sona erer.

5.2.2 Benzetimli Tavlama

Benzetimli Tavlama (SA), Monte Carlo simülasyon tekniklerinden ilham alarak karmaşık kombinatoriyel optimizasyon problemlerini çözmek için kullanılan bir yöntemdir. "Benzetimli Tavlama" ismi, bir maddenin ısıdıktan sonra yavaşça soğutularak kusurların azaltılması ve en düşük enerji durumuna ulaşılması fiziksel tavlama süreciyle yapılan benzerliğe dayanır. SA, 1983 yılında Kirkpatrick ve arkadaşları tarafından bir optimizasyon tekniği olarak tanıtılmıştır. Bu yöntem, bir fiziksel sistemin soğumasının davranışını simüle ederek çalışır. Sıcaklık parametresi, algoritmanın ilerledikçe kötü çözümleri kabul etme olasılığını kontrol eder. Başlangıçta SA, daha fazla keşif yapabilmesi için (kötü çözümleri kabul ederek) yerel optimumlardan kaçınmayı sağlar. Zamanla sıcaklık azaldıkça (iterasyonlar ilerledikçe), algoritma çözüm aralıklarını daraltarak (sadece daha iyi çözümleri kabul ederek) düşük enerji durumuna ve donma noktasına doğru sabit hale gelir [81].

Benzetimli Tavlama süreci, optimize edilen sistemi önce yüksek bir etkin sıcaklıkta "eritmek", ardından sıcaklığı yavaş aşamalarda düşürerek sistemin "dondugu" ve daha fazla değişiklik olmadığı noktaya kadar indirmeyi içerir. Her sıcaklıkta, simülasyonun sistemin kararlı bir duruma ulaşması için yeterince uzun sürmesi gerekir [82].

Şekil 5.3’de SA çalışma prensibi verilmiştir. Tablo 5.2’de tavlama işlemindeki terimlerin optimizasyon problemindeki eşleşmesi verilmiştir.



Şekil 5.3 Benzetimli tavlama algoritmasının yapısı [96]

Tablo 5.2 Tavlama işlemi ile optimizasyon probleminin eşleşmesi [83]

Fiziksel Sistem	Kombinatorial Optimizasyon Sistemi
Sistemin Durumları	Uygun Çözümleri
Sistemin Enerjisi	Amaç Fonksiyon Değeri
Sıcaklık	Kontrol Parametresi
Denge Durumu	Optimal Çözüm

Benzetimli tavlama algoritmasının yapısını anlayabilmek için aşağıdaki kavramların açıklanması faydalı olacaktır.

- Komşuluk belirleme
- Başlangıç sıcaklığı belirleme
- Soğutma planı
- Kabul fonksiyonu
- Durdurma kriteri

5.2.2.1 Komşuluk Belirleme

Benzetimli Tavlama (SA) algoritmasında başlangıç çözümü elde edildikten sonra, bir komşu çözüm oluşturulur. Bu komşu çözüm, mevcut çözümde yapılan değişikliklerle ya rastgele yöntemlerle ya da önceden belirlenmiş kurallara göre üretilir [84]. Bu çalışmada komşu çözüm rassal olarak oluşturulmuştur.

5.2.2.2 Başlangıç Sıcaklığı Belirleme

Benzetimli Tavlama algoritmasında, başlangıç sıcaklığı yeterince büyük seçilmelidir ki tüm olası çözümler değerlendirilebilsin. Bu şekilde, algoritmanın başlangıç safhasında geniş bir arama yapabilmesi sağlanır. Başlangıçta çözümlerin kabul olasılığı bire çok yakın olmalıdır, böylece başlangıç çözümlerinin çoğu kabul edilir. Algoritma ilerledikçe ve sıcaklık düştükçe, çözümlerin kabul olasılığı sıfıra yakın olur. Bu, amaç fonksiyonunda iyileştirme sağlamayan yeni çözümlerin kabul edilmemesini sağlar [85].

5.2.2.3 Soğutma Planı

Sıcaklık, bir önceki çözüme göre daha kötü olan bir çözümün kabul edilme olasılığının hesaplanmasında kullanılır. Benzetimli tavlama algoritmasında, soğutma işlemi yavaş yavaş gerçekleştirilmelidir. Bu amaçla, bir sıcaklık azaltma fonksiyonu kullanılır. Literatürde farklı sıcaklık azaltma fonksiyonları önerilmiştir. Aşağıda bu fonksiyonlardan dört örnek verilmiştir [85]:

- | | |
|-------------------------|------------------------------|
| 1. Aritmetik fonksiyon | $T_k = T_{k-1} - C_{te}$ |
| 2. Geometrik fonksiyon | $T_k = T_{k-1} * \alpha$ |
| 3. Ters fonksiyon | $T_k = C_{te} / (1+k)$ |
| 4. Logaritmik fonksiyon | $T_k = C_{te} / (\log(1+k))$ |

Bu çalışmada geometrik fonksiyon kullanılmıştır.

5.2.2.4 Kabul Fonksiyonu

Eğer ΔE değeri ile hesaplanan kabul koşulu sağlanıyorsa, mevcut çözüm yerel komşulukla elde edilen sonuç olarak kabul edilir. Yani, komşu çözümün amaç fonksiyonu değeri ile mevcut çözümün değeri arasındaki fark, belirli bir kabul eşiği değerinden küçük veya eşitse, yeni çözüm kabul edilir. Ancak bu koşul sağlanmazsa, yani ΔE değeri kabul eşiğinden büyükse, algoritma kötü bir çözümün kabul edilme olasılığını hesaplamak için Boltzmann dağılımını kullanır. Bu dağılım, sıcaklık parametresine ve ΔE 'ye bağlı olarak kötü çözümlerin kabul edilme olasılığını belirler. Sıcaklık yüksek olduğunda kötü çözümlerin kabul edilme olasılığı artar, ancak sıcaklık düştükçe bu olasılık azalır [83].

$$P(\Delta E, T) = e^{-\frac{f(x') - f(x)}{T}} \quad (5.2)$$

Formüldeki T kontrol parametresi olan sıcaklığı ifade etmektedir.

5.2.2.5 Durdurma Kriteri

Durdurma kriteri için literatürde kullanılan çeşitli testler vardır. Bunlar arasında, algoritmanın belirli bir maksimum iterasyon sayısına ulaştığında, belirli bir deneme sayısı için kabul edilen çözümlerin sayısına ulaştığında ve önceden belirlenen bir son sıcaklık değerine ulaştığında durdurulması bulunur [85].



6. STOK KONTROL PARAMETRELERİNİN BELİRLENMESİ İÇİN GENETİK ALGORİTMA VE BENZETİMLİ TAVLAMA TABANLI BİR SİMÜLASYON OPTİMİZASYON UYGULAMASI

6.1 Problem Tanımı ve Varsayımlar

Tez kapsamında, Manisa’da beyaz eşya plastik enjeksiyon sektöründe faaliyet gösteren üretici bir firmada, montaj prosesinde kullanılan görsel bir parça, stok kontrol açısından ele alınmıştır. Firma ele alınan görsel parça için sürekli (s,S) stok kontrol politikası uygulamaktadır fakat stok kontrol sisteminin parametreleri planlama yöneticisinin tecrübelerine dayalı bir şekilde belirlendiği için yüksek talep durumunda stoksuz kalma, düşük talep durumunda ise stok yığılması durumu yaşanabilmektedir. Bu nedenle bu tezde firmanın uyguladığı stok kontrol politikasının geliştirilmesi, optimum (s,S) değerlerinin bulunması ve toplam stok maliyetinin minimize edilmesi amaçlanmaktadır.

Firma müşteriden gelen siparişe göre onaylı tedarikçiye sipariş açmakta, ürün tedarikçinin üretim kısıtlarına göre farklı sürelerde tedarik edilmektedir. Tedarik edilen ürün çeşitli işlemlerden geçirildikten sonra montaj işlemi yapılarak müşteriye sevk edilmektedir. Ele alınan ürün lazer kesim, serigrafi, dış kaplama uygulaması ve yapısal yapışkan yüzey kaplama gibi çeşitli prosesten geçtikten sonra montaj işlemi yapılmaktadır. Ürün çeşitli proseslerden geçtikten sonra montaj prosesine geçmesi nedeniyle, montaj hattının ve müşterinin siparişlerinin aksamaması açısından kritik bir öneme sahiptir.

Mevcut durumda, müşteriden her gün sipariş gelmektedir. Planlama mühendisi, stok seviyesi, güvenlik stoğu noktası (s) olarak belirlenen 750 adet altına düştüğünde stok pozisyonuna göre belirlenen maksimum stok seviyesi noktası (S) 1500 adete kadar sipariş vermektedir. Stok seviyesi eldeki stok miktarı, tedarikçiye açılan teslim alınmayan sipariş miktarı toplamından, ertelenmiş talebin miktarının çıkarılmasıyla hesaplanmaktadır. Stok seviyesi formülü aşağıda verilmiştir.

$$I_p = I_L + I_0 - \text{Ertelenmiş Talep (Backorder)} \quad (6.1)$$

Problemin yapısını daha iyi anlamak için içerdği kısıtların ve varsayımların tanımlanması önemli olacaktır.

- Mevcut durumda yeniden sipariş verme noktası (s) 750 adet, maksimum stok seviyesi (S) 1500 adet olarak firmanın planlama yöneticisi tarafından belirlenmiştir.
- Firma politikası nedeniyle yeniden sipariş verme noktasının (s) 500 ve 2000 adet arasında olması istenmektedir.
- Firma politikası nedeniyle maksimum stok seviyesinin (S) 1000 ve 2000 adet arasında olması istenmektedir.
- Müşteri tarafından ürünün bir adet onaylı tedarikçisi bulunmaktadır.

Bu tezde optimum (s,S) değerlerinin bulunması ve toplam stok maliyetinin minimize edilmesi amacıyla metasezgisel algoritma tabanlı simülasyon optimizasyon yaklaşımı önerilmiştir. Bu nedenle firmanın uyguladığı (s,S) stok kontrol politikası simülasyon modelleme programı olan Arena Rockwell Simulation 14.0 programı ile stokastik olarak modellenmiştir. Arena programının “OptQuest” optimizasyon aracı kullanılarak, optimum stok kontrol parametreleri ve toplam stok maliyeti elde edilmiştir. Daha sonra genetik algoritma ve benzetimli tavlama algoritması teknik bir programlama dili olan Matlab programında kodlanmıştır. Simülasyon programıyla entegre çalışan genetik algoritma ve benzetimli tavlama algoritmalarıyla optimum stok kontrol parametreleri ve toplam stok maliyeti elde edilmiştir. Genetik algoritma ve benzetimli tavlama algoritması ile elde edilen sonuçlar OptQuest optimizasyon aracı kullanılarak bulunan sonuçla karşılaştırılarak en iyi sonucu veren çözüm yöntemi belirlenmiştir.

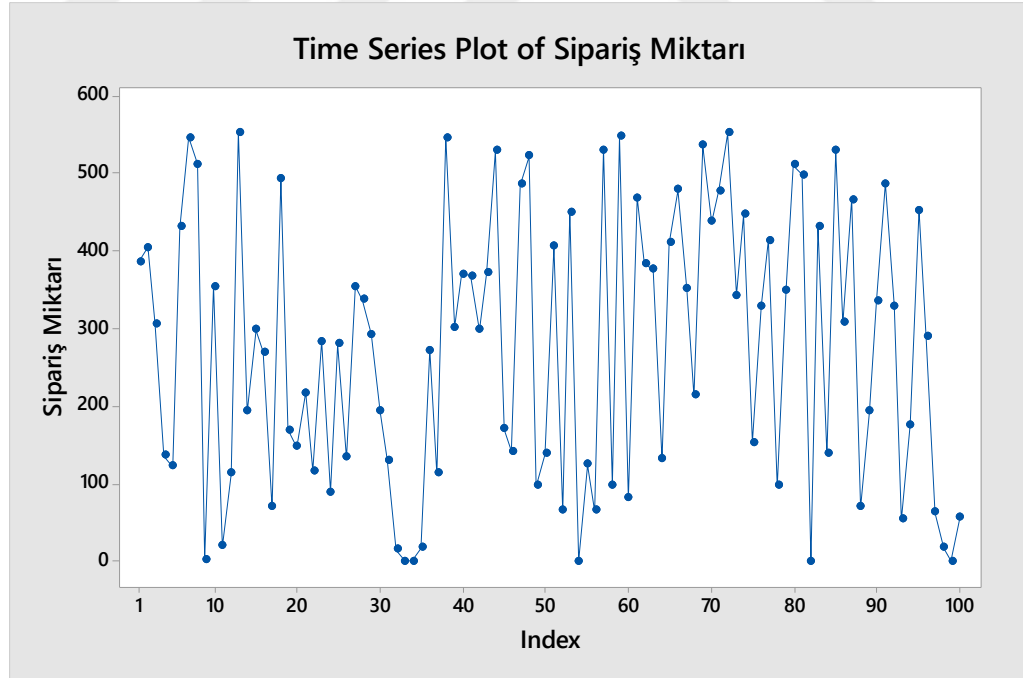
Simülasyon modelinin oluşturulabilmesi için stokastik verilerin dağılımının bulunması gerekmektedir. Bu nedenle girdi analizi yapılmıştır.

6.2 Girdi Analizi

Manisa'daki bir üretici firmanın veri seti incelenerek çözüm stratejisi detaylı bir şekilde sunulacaktır. Gizlilik anlaşması gereği, ilgili firmanın gerçek verileri modelde yaklaşık veriler olarak kullanılacaktır. Stokastik girdi verilerinin modellenmesi için yapılan analizler bu bölümde verilmiştir.

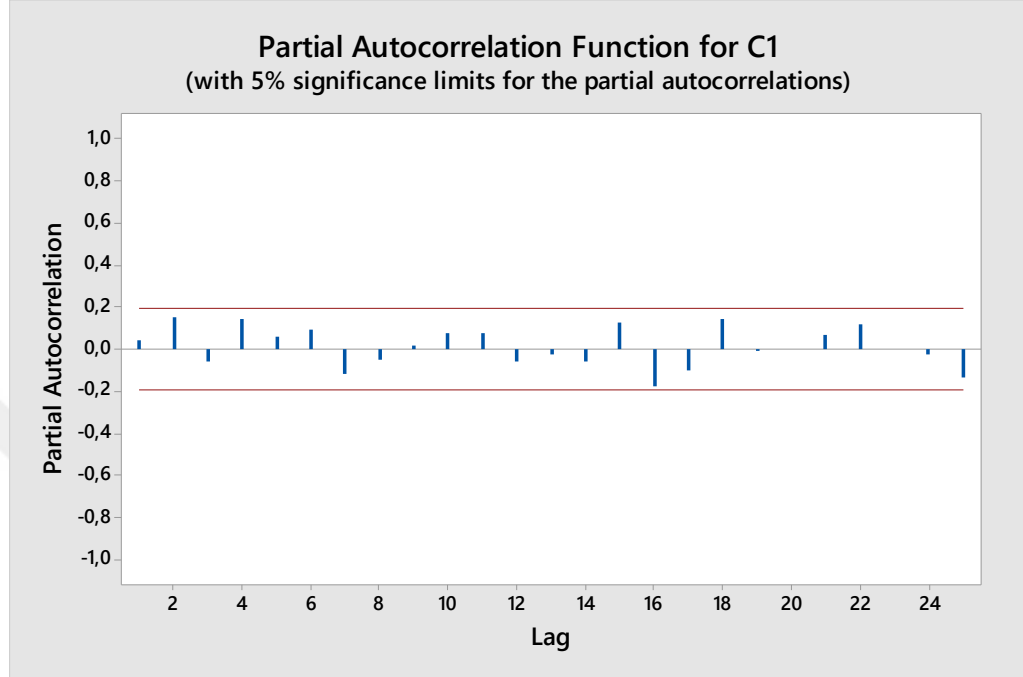
Son bir yıl içerisindeki müşteri siparişlerine göre yüz günlük rassal bir örneklem seçilerek günlere göre talep miktarları belirlenmiştir. Sipariş miktarına ilişkin alınan rassal örneklem tablosu Ek 6.1'de yer almaktadır.

Ek 6.1'de verilen örnekleme göre sipariş miktarları için zaman serisi ve otokorelasyon grafikleri istatistiksel bir yazılım olan Minitab programı ile incelenmiştir. Sipariş miktarı için örneklemin en uygun hangi dağılıma ait olduğu Arena programının "Input Analyzer" aracı ile belirlenmiştir. İlgili dağılımın simülasyonda kullanılabilirliği istatistiksel olarak test edilmiştir.



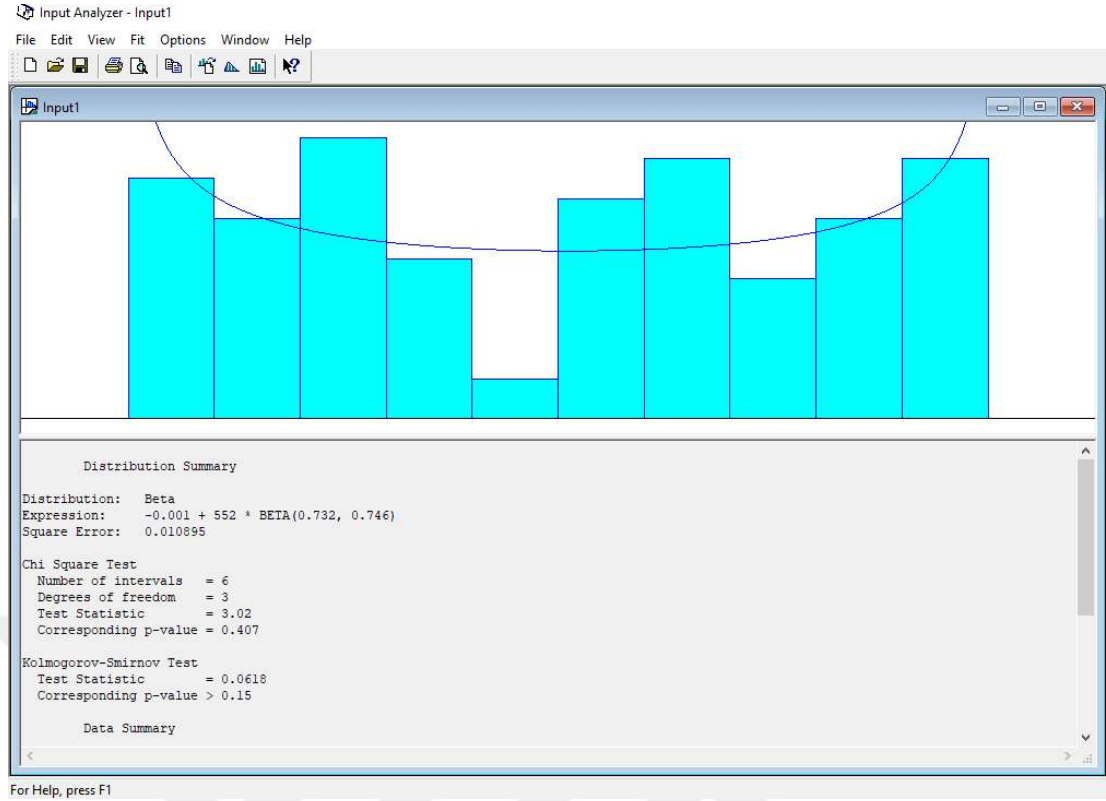
Şekil 6.1 Sipariş miktarları için zaman serisi grafiği

Sipariş miktarları örnekleme için zaman serisi grafiği Şekil 6.1’de verilmiştir. Zaman serisi grafiğinde döngüsel bir desen veya eğilim gözlemlenmediği için örneklem zamana göre durağandır.



Şekil 6.2 Sipariş miktarları için otokorelasyon grafiği

Şekil 6.2’de sipariş miktarları örnekleme için otokorelasyon grafiği yer almaktadır. Otokorelasyon grafiğinde, gecikme otokorelasyonları %95’lik güven düzeyini aşmadığı için örneklem verileri arasında bir otokorelasyon yoktur ve örneklem verileri birbirinden bağımsızdır. Örneklem verileri, zaman serisi grafiğine bakıldığında durağan ve otokorelasyon grafiğine bakıldığında birbirinden bağımsız olduğu gözlemlendiği için rassal bir örneklemdir.



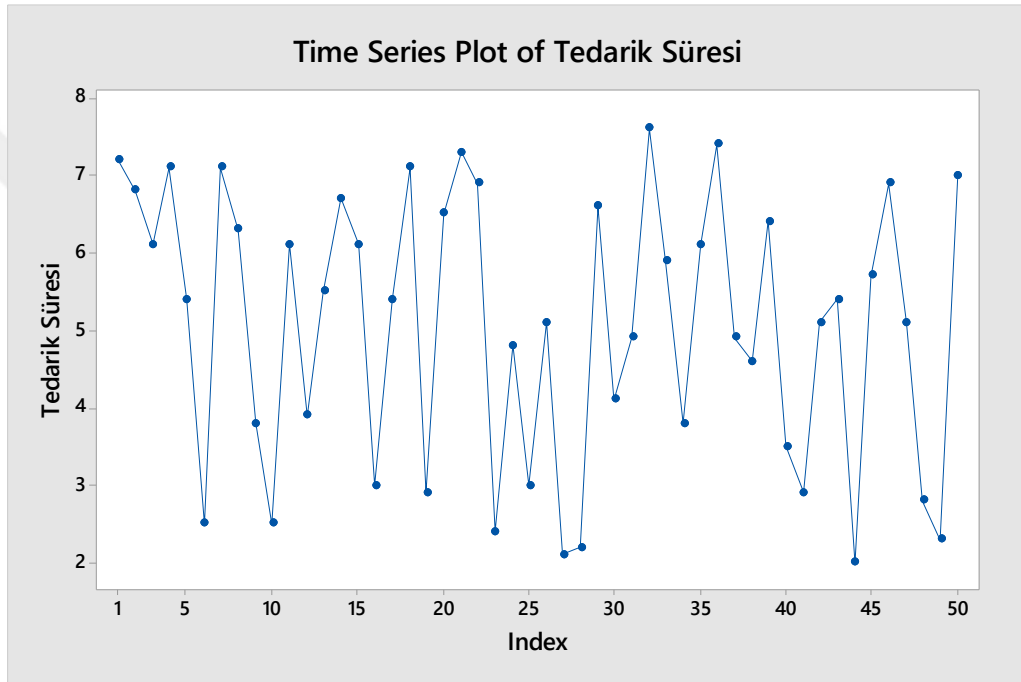
Şekil 6.3 Sipariş miktarı dağılım grafiği

Şekil 6.3’de görüldüğü üzere örneklem Arena programının “Input Analyzer” aracı ile analiz edildiğinde en uygun sipariş miktarı dağılımının $-0.001 + 552 * \text{BETA}(0.732, 0.746)$ şeklindeki BETA dağılımı olduğu görülmektedir. Programda belirtilen dağılım için $\alpha=0,05$ anlam düzeyinde ki-kare uygunluk testi gerçekleştirilmiştir Test için H_0 : Örneklem verileri $-0.001 + 552 * \text{BETA}(0.732, 0.746)$ dağılımına uygundur ve H_0 : Örneklem verileri $-0.001 + 552 * \text{BETA}(0.732, 0.746)$ dağılımına uygun değildir olarak 2 adet hipotez kurulmuştur. Ki-kare uygunluk testinde örneklem için P değeri 0,407 olarak elde edilmiştir. Buna göre $P \text{ değeri}=0,407 > \alpha=0,05$ olması nedeniyle H_0 hipotezinin kabul edilmemesi için istatistiksel bir ifade bulunmamıştır, H_0 hipotezi kabul edilmiştir ve örneklem verilerinin $-0.001 + 552 * \text{BETA}(0.732, 0.746)$ dağılımına uyduğu kabul edilmiştir.

Uygulamada ele alınan ürün müşteri tarafından belirlenen tek onaylı tedarikçiden satın alınmaktadır. Ürünün tedarik süresi, siparişlerin verilmesinden sonra tedarikçi stoklarına ve tedarikçinin üretim kapasitesine bağlı olarak değişkenlik gösterebilir. Bu sebeple daha önce verilmiş siparişlerden tedarik sürelerine ilişkin 50

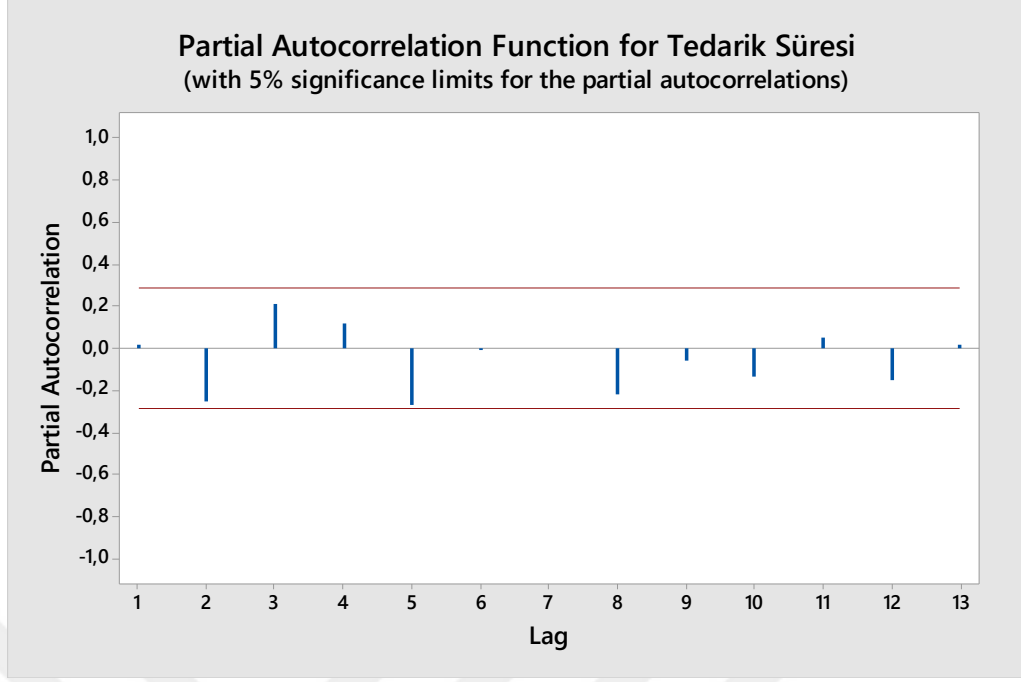
adet rassal örneklem alınmıştır. Ek 6.2’de tedarik sürelerine ilişkin alınan rassal örneklem tablosu verilmiştir.

Ek 6.2’de verilen örnekleme göre tedarik süreleri için zaman serisi ve otokorelasyon grafikleri istatistiksel bir yazılım olan Minitab programı ile incelenmiştir. Tedarik süreleri için örneklemin en uygun hangi dağılıma ait olduğu Arena programının “Input Analyzer” aracı ile belirlenmiştir. İlgili dağılımın simülasyonda kullanılabilirliği istatistiksel olarak test edilmiştir.



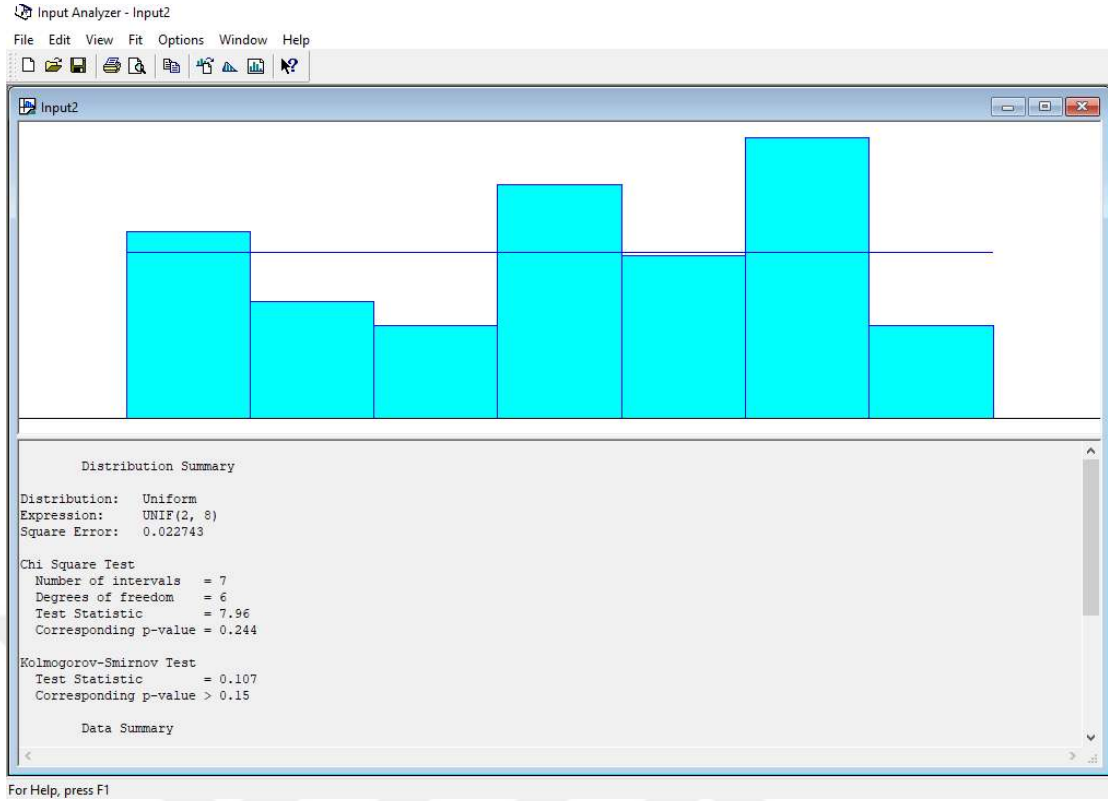
Şekil 6.4 Tedarik süreleri için zaman serisi grafiği

Tedarik süreleri örnekleme için zaman serisi grafiği Şekil 6.4’te verilmiştir. Zaman serisi grafiğinde döngüsel bir desen veya eğilim gözlemlenmediği için örneklem zamana göre durağandır.



Şekil 6.5 Tedarik süreleri için otokorelasyon grafiği

Şekil 6.5'te tedarik süreleri örnekleme için otokorelasyon grafiği yer almaktadır. Otokorelasyon grafiğinde, gecikme otokorelasyonları %95'lik güven düzeyini aşmadığı için örneklem verileri arasında bir otokorelasyon yoktur ve örneklem verileri birbirinden bağımsızdır. Örneklem verileri, zaman serisi grafiğine bakıldığında durağan ve otokorelasyon grafiğine bakıldığında birbirinden bağımsız olduğu gözlemlendiği için rassal bir örneklemdir.



Şekil 6.6 Tedarik süresi dağılım grafiği

Şekil 6.6’da görüldüğü üzere örneklem Arena programının “Input Analyzer” aracı ile analiz edildiğinde en uygun sipariş miktarı dağılımının UNIF (2, 8) şeklindeki UNIF dağılımı olduğu görülmektedir. Programda belirtilen dağılım için $\alpha=0,05$ anlam düzeyinde ki-kare uygunluk testi gerçekleştirilmiştir Test için H_0 : Örneklem verileri UNIF (2, 8) dağılımına uygundur ve H_0 : Örneklem verileri UNIF (2, 8) dağılımına uygun değildir olarak 2 adet hipotez kurulmuştur. Ki-kare uygunluk testinde örneklem için P değeri 0,244 olarak elde edilmiştir. Buna göre P değeri $=0,244 > \alpha=0,05$ olması nedeniyle H_0 hipotezinin kabul edilmemesi için istatistiksel bir ifade bulunmamıştır, H_0 hipotezi kabul edilmiştir ve örneklem verilerinin UNIF (2, 8) dağılımına uyduğu kabul edilmiştir.

Firmanın kontrolü dışındaki sipariş miktarı ve tedarik süresinin hangi dağılıma uyduğu belirlenmiştir. Firmanın stok kontrol politikası Arena programında modellenirken, bulunan dağılımlar kullanılarak stokastik bir stok kontrol modeli üzerinden çalışmaya olanak sağlanacaktır.

6.3 Maliyet Hesaplamaları

Firmanın optimum stok kontrol parametreleri bulunmaya çalışılırken, simülasyon modelinde sistemin çalışmasını belirleyecek değişkenlerin hesabı için stok maliyetlerine ihtiyaç duyulmaktadır. Firmanın sipariş verme maliyeti, elde bulundurma maliyeti, malzeme maliyeti, elde bulundurmama (stok tutmama) maliyeti olmak üzere 4 çeşit stok maliyeti bulunmaktadır. Maliyetlerin hesaplanmasında 2023 yılının ikinci dönem verilerinden yararlanılmıştır ve gizlilik anlaşması gereği yaklaşık değerler kullanılmıştır. Stok maliyetlerinin hesaplanması ile ilgili formüller aşağıdaki verilmiştir:

- Aylık sipariş verme maliyeti = aylık ortalama sipariş sayısı * sabit maliyet (115,91 TL/Sipariş)
- Aylık elde bulundurma maliyeti= aylık ortalama stok miktarı * aylık faiz (0,0235) * birim malzeme maliyeti (101,50 TL/Adet)
- Aylık malzeme maliyeti= aylık ortalama sipariş miktarı * birim malzeme maliyeti (101,50 TL/Adet)
- Aylık elde bulundurmama maliyeti= 45* aylık toplam ortalama ertelenmiş sipariş miktarı

Bir birim ürünü bekletme maliyeti 45 TL/Adet'dir.

- Aylık toplam stok maliyeti = aylık sipariş verme maliyeti + aylık elde bulundurma maliyeti + aylık malzeme maliyeti+ aylık elde bulundurmama maliyeti

Ürünün birim satın alma maliyeti 101,50 TL/Adet'dir.

Aylık faiz oranı piyasa koşulları dikkate alınarak firma yöneticileri tarafından 0,0235 olarak belirlenmiştir.

Sabit maliyet hesaplanırken sipariş verilen malzemenin tedarikiyle ilgili olarak geçen süre, bu süre için ödenen ücretler ve dokümantasyon ücreti göz önünde

bulundurulmuştur. Firmanın sipariş sürecinde görevli departmanlar: teslim-sevkiyat, planlama ve muhasebe departmanlarıdır. Departmanların malzeme tedarik sürecindeki görevleri aşağıda özetlenmiştir.

Planlama departmanı için: Stok seviyelerinin düzenli olarak takibi, stok seviyeleri ve müşterilerden gelen siparişlerin analizi doğrultusunda tedarikçiye firmaya sipariş talebi oluşturulması, tedarikçi ve müşteri ile iletişim halinde kalarak siparişin sorunsuz bir şekilde tamamlanması olarak özetlenebilir.

Teslim ve sevkiyat departmanı için: Malzemelerin tedariki için sevkiyat organizasyonu, gelen malzemelerin ve ürünlerin sipariş taleplerine uygunluğunun kontrol edilmesi, gelen ürünün depoya alınması ve stok sisteminin güncellenmesi olarak özetlenebilir.

Muhasebe departmanı için: İşletmenin finansal yükümlülüklerini yerine getirmek üzere tedarikçiye yapılan ödemelerin yönetilmesi, ürün ile ilgili finansal işlemlerin kaydedilmesi olarak özetlenebilir.

Siparişin yönetilmesi için planlama departmanı 25,02, teslim ve sevkiyat departmanı 15, muhasebe departmanı 15 dakika süre harcamaktadır. Sabit maliyet hesaplaması için çalışanlara verilen aylık 17.250 TL kullanılacaktır. Bir çalışan 30 iş günü boyunca günde 8 saat boyunca çalışmaktadır.

Bu bilgilerle dakikalık ücret elde edilir. $17.250/30/8/60=1,1979$ TL/Dakika

Planlama departmanı işçilik maliyeti: $25,02*1,1979=29,9714$ TL/Sipariş

Teslim ve sevkiyat departmanı işçilik maliyeti: $15*1,1979= 17,9685$ TL/Sipariş

Muhasebe departmanı işçilik maliyeti: $15*1,1979= 17,9685$ TL/Sipariş

Bu işlemler sonucunda bir kere sipariş verme sabit işçilik maliyeti 65,91 TL/Sipariş olarak hesaplanır. Bir kere sipariş verme dokümantasyon işlemi maliyeti

50 TL/Sipariş olarak hesaplanmıştır. Bir kere sipariş verme maliyeti 115,91 TL/Sipariş olarak hesaplanmıştır.

Stoksuz kalmama olasılığı = Simülasyondaki (n) çevrim boyunca stoksuz kalınmayan çevrim sayısı / Simülasyondaki toplam çevrim sayısı (n) olarak hesaplanmıştır.

6.4 Simülasyon Modelinin Oluşturulması

Temel girdi parametreleri belirlendikten sonra, firmanın uyguladığı stok kontrol politikası Arena simülasyon programında modellenmiştir. Modelin oluşturulmasında kullanılan sipariş miktarı ve tedarik süresi parametreleri deterministik değildir ve firmanın stok yönetim süreci belirsizliklere sahiptir. Bu nedenle stokastik bir stok yönetim modeli oluşturulmuştur. Modeli oluşturmak için Arena’da kullanılan değişkenler şekil 6.7’de verilmiştir.

Variable - Basic Process								
	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics
1	kucuks			Real	System		1 rows	☐
2	buyuks			Real	System		1 rows	☐
3	baslangicstok			Real	System		1 rows	☐
4	guncelstok			Real	System		1 rows	☐
5	Yoldaki Stok Miktarı			Real	System		0 rows	☐
6	Ertelenmiş Talep			Real	System		0 rows	☐
7	Eideki Stok Miktarı			Real	System		1 rows	☒
8	Stok Pozisyonu			Real	System		0 rows	☐
9	Satınalma Sipariş Miktar			Real	System		0 rows	☐
10	Anlık Sipariş Miktar			Real	System		0 rows	☐
11	Sipariş Sayısı			Real	System		0 rows	☐
12	Stoktan Karşılama Sipariş Sayısı			Real	System		0 rows	☐
13	Toplam Sipariş Miktarı			Real	System		0 rows	☐
14	Ertelenen Talep Miktarı			Real	System		0 rows	☒
15	Toplam Satınalma Sipariş Miktarı			Real	System		0 rows	☐
16	gelis suresi			Real	System		0 rows	☐
17	Toplam Satınalma Sipariş Sayısı			Real	System		0 rows	☐

Şekil 6.7 Model değişkenleri

Simülasyon modelinin oluşturulma aşamaları aşağıdaki gibi verilmiştir:

- Model, müşterinin ihtiyacını sipariş etmesiyle başlamaktadır.

- Müşteriden gelen sipariş miktarı girdi analizinde belirlenen olasılık dağılım fonksiyonu kullanılarak hesaplanır.
- Gelen talebin mevcut stok miktarı ile karşılanıp karşılanmadığı kontrol edilir. Eğer talep miktarı, mevcut stok miktarından düşük ise talep karşılanır. Aksi halde siparişin tamamı için ertelenmiş talep oluşur (Tam ertelenmiş talep).
- Talebin karşılanması veya ertelenmiş talep oluşması durumunda stok seviyesi güncellenir. Daha sonra kontrol edilir. Talep karşılandığında eğer stok seviyesi yeniden sipariş noktasının altındaysa satın alma sürecine geçilir. Aksi halde sistemden çıkış gerçekleşir. Ertelenmiş talep oluştuğunda stok seviyesi yeniden sipariş noktasının altındaysa satın alma sürecine geçilir.
- Satın alma kararı verildikten sonra ürünün tedarik süreci başlar. Tedarik süresi girdi analizinde belirlenen olasılık dağılım fonksiyonu kullanılarak hesaplanır. Sipariş edilen ürün tedarikçiden geldiğinde stok seviyesi güncellenir.
- Tedarikçiden gelen ürün miktarı ile stok seviyesi güncellendikten sonra, geciktirilmiş talep olup olmadığı kontrol edilir. Eğer eldeki stok ertelenen sipariştten büyükse önce ertelenen sipariş karşılanır, stok seviyesi tekrar güncellenir ve stok seviyesi yeniden sipariş noktasının altındaysa satın alma sürecine geçilir.

Şekil 6.8’de modelin maliyetlerinin hesaplanması için formüller verilmiştir. Maliyetler modelin içinde otomatik olarak hesaplanmaktadır.

Statistic - Advanced Process			
	Name	Type	Expression
1	Aylık Ortalama Malzeme Maliyeti	Output	TAVG(Aylık Verilen Siparis Miktarı)* 101.5
2	Cevrim Hizmet Duzeyi 1	Output	Stoktan Karsilanan Sip Sayisi / Siparis Sayisi
3	Aylık Ortalama Elde Bulundurma Maliyeti	Output	DAVG(Eldeki Stok Miktarı Value) * 0.235 * 101.5
4	Aylık Ortalama Toplam Siparis Maliyeti	Output	115.91*TAVG(Aylık Siparis Sayisi)
5	Aylık Ortalama Toplam Maliyet	Output	OVALUE(Aylık Ortalama Toplam Siparis Maliyeti) + OVALUE(Aylık Ortalama Malzeme Maliyeti) + OVALUE(Aylık Ortalama Elde Bulundurma Maliyeti) + OVALUE(Aylık Ortalama Backorder Maliyeti)
6	Aylık Ortalama Backorder Maliyeti	Output	DAVG(Ertelenen Talep Miktarı Value)*45

Şekil 6.8 Modelin istatistiksel çıktı değerleri

Şekil 6.9’da Arena programında modellenen (s,S) stok kontrol politikasının ekran görüntüsü verilmiştir.



Simülasyon modelinin varsayımları aşağıdaki şekilde verilmiştir:

- Talebin stoklardan karşılanamadığı durumlarda ertelenmiş talep olduğu varsayılır.
- Sipariş kararı vermede stok seviyesi kullanılır.
- Tedarik süresince talep yeniden sipariş noktasını aşarsa stoksuz kalma gerçekleşir.
- Tedarik edilen talep ile öncelikle varsa ertelenmiş talep karşılanır.
- Model 360 gün boyunca 10 replikasyon çalıştırılmıştır.

Model sonuçlarının teorik hesaplamalarla uygun olduğu görülmüştür. Modelin doğrulaması ve geçerlilik aşamaları yapıldıktan sonra simülasyon optimizasyonu aracı OptQuest ile optimum stok kontrol parametreleri ve minimum stok kontrol maliyeti belirlenmesi bölümüne geçilmiştir.

6.5 Arena OptQuest ile Simülasyon Tabanlı Optimizasyon

Arena simülasyon modeli kurulduktan sonra, ilk etapta Arena programının optimizasyon aracı olan OptQuest kullanılarak optimum stok kontrol parametreleri bulunmaya çalışılmıştır. OptQuest aracı kullanarak %80 stoksuz kalmama olasılığına göre ortalama maliyetin minimize edildiği optimum yeniden sipariş verme miktarı ve maksimum stok miktarı noktası belirlenmeye çalışılacaktır. Bu süreçte, belirlenen değişkenler şunlardır: yeniden sipariş verme miktarı ve maksimum stok miktarı. OptQuest, bu değişkenler tanımlandıktan sonra çalıştırılacak ve amaç fonksiyonuna uygun olan optimum kontrol değişkenlerini bulacaktır. Böylece, stoksuz kalmama olasılığını %80'in üstünde tutarken, ortalama maliyeti en aza indiren stok yönetim stratejisi belirlenecektir.

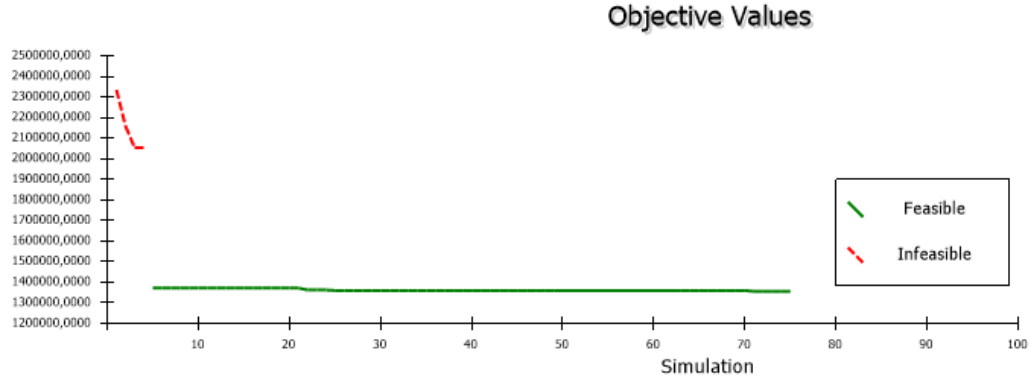
- Kontrol değişkenleri: kucuks, buyuks
- Amaç fonksiyonu: aylık ortalama toplam maliyet: minimize
- Kısıtlıyıcılar: kucuks<buyuks
- $100 \cdot \text{cevrim hizmet düzeyi} > 80$ (talebin stoklardan karşılanma olasılığı %80'in üzerinde)

OptQuest ile bulunan en iyi çözümlerin ekran görüntüsü Şekil 6.10’da verilmiştir. OptQuest aracı kullanılarak bulunan en iyi çözüm parametreleri, buyuks miktarı 2000, kucuks miktarı 1970 ile stok maliyeti 1.356.161,83 TL hesaplanmıştır. OptQuest ile çözüm süresi 00:04:05 olarak bulunmuştur.

Best Solutions						
Select	Simulation	Objective Value	Status	buyuks	kucuks	
☒	71	1,356162E+006	Feasible	2000	1970	
☐	35	1,360239E+006	Feasible	2000	1948	
☐	25	1,360362E+006	Feasible	2000	1949	
☐	26	1,361102E+006	Feasible	2000	1941	
☐	66	1,361940E+006	Feasible	2000	1897	
☐	30	1,362899E+006	Feasible	2000	1902	
☐	29	1,362899E+006	Feasible	2000	1913	
☐	22	1,363256E+006	Feasible	2000	1990	
☐	73	1,364230E+006	Feasible	2000	1901	
☐	46	1,364556E+006	Feasible	2000	1939	
☐	72	1,365399E+006	Feasible	2000	1974	
☐	68	1,366314E+006	Feasible	2000	1963	
☐	47	1,367834E+006	Feasible	2000	1947	
☐	42	1,369112E+006	Feasible	2000	1905	
☐	41	1,370440E+006	Feasible	2000	1955	
☐	5	1,372039E+006	Feasible	2000	2000	
☐	23	1,372402E+006	Feasible	2000	1987	
☐	21	1,372455E+006	Feasible	2000	1960	
☐	24	1,374101E+006	Feasible	2000	1950	
☐	9	1,374440E+006	Feasible	2000	1951	
☐	50	1,374732E+006	Feasible	2000	1931	

Şekil 6.10 OptQuest ile simülasyon tabanlı optimizasyon sonuçları

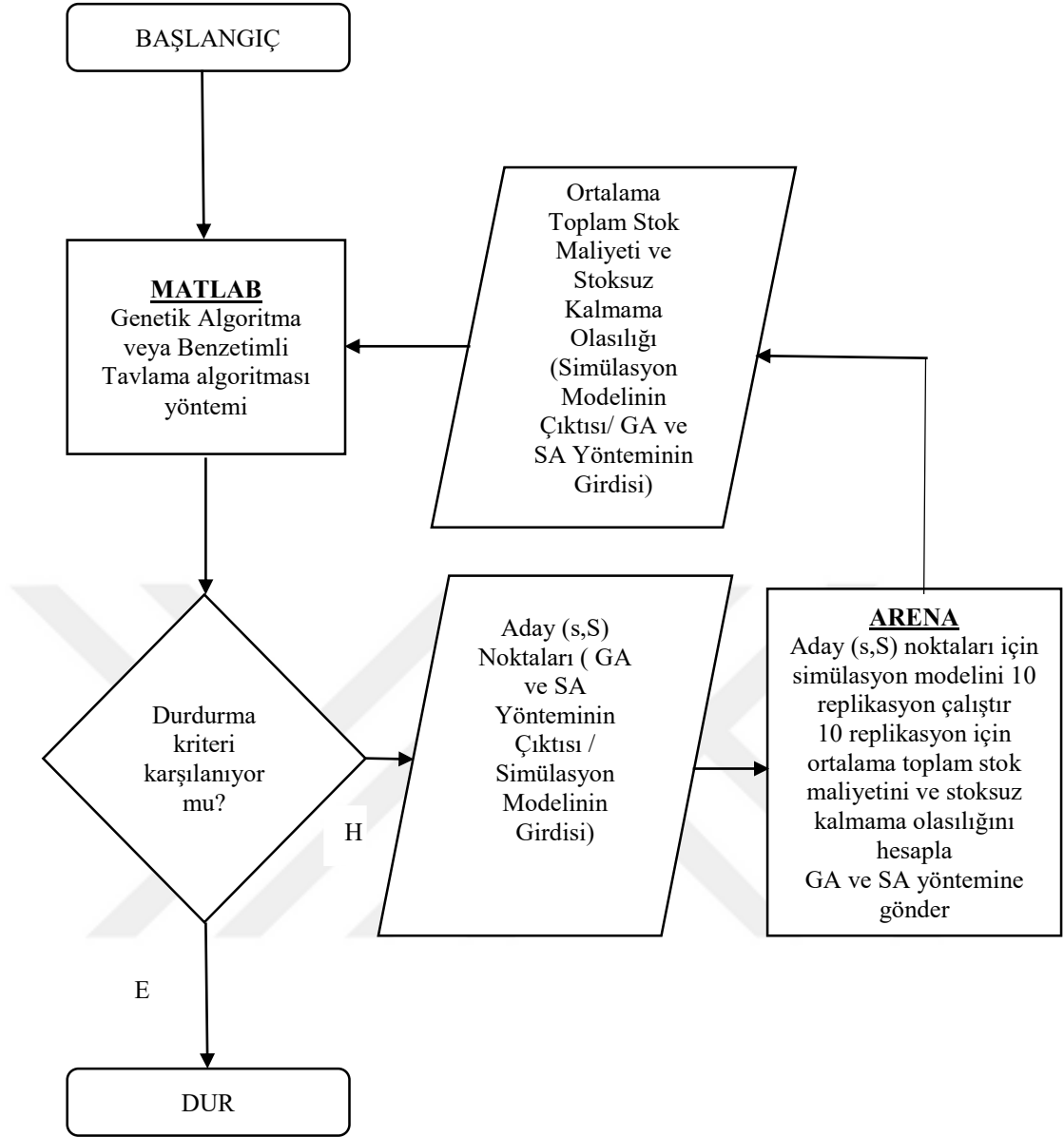
Değişkenler tanımlanıp OptQuest aracı çalıştırıldığında iterasyon ilerledikçe amaç fonksiyonu değerinin iyileşme grafiği Şekil 6.11’de görülmektedir. İterasyon ilerledikçe daha uygun sonuçlar elde edilerek optimum nokta tespit edilmiştir. Bu, kontrol değişkenlerinin doğru bir şekilde optimize edildiğini ve istenen hedefe ulaşıldığını göstermektedir.



Şekil 6.11 OptQuest iterasyon maliyet grafiği

6.6. Genetik Algoritma ve Benzetimli Tavlama Algoritması Tabanlı Simülasyon Optimizasyonu Yönteminin Oluşturulması

Önerilen çözüm yaklaşımında öncelikle GA ve SA aday (s,S) noktaları bulur ve bu noktalar simülasyon modeline girdi olarak gönderilir. GA ve SA tabanlı simülasyon optimizasyonu yöntemi Şekil 6.12’de ayrıntılı olarak gösterilmiştir. Bu şekle göre GA ve SA simülasyon modeline bir aday çözüm gönderdiğinde, simülasyon programı ile bu noktalar kullanılarak 10 replikasyon çalıştırılır. Ortalama toplam stok maliyeti ve stoksuz kalmama olasılığı bulunur. Daha sonra ortalama toplam stok maliyeti, amaç fonksiyonu değeri olarak simülasyon optimizasyonu yöntemine iletilir. Ayrıca stoksuz kalmama olasılığı da simülasyon optimizasyonu yöntemine gönderilir. Simülasyon optimizasyonu problemi tamamlandığında optimuma yakın (s,S) stok kontrol noktaları elde edilir. Özetle Arena programı amaç fonksiyonlarının ve stoksuz kalmama olasılıklarının hesaplanması için hesap makinesi olarak kullanılmıştır denilebilir.



Şekil 6.12 GA ve SA tabanlı simülasyon optimizasyonu yönteminin akış şeması [86]

Bundan sonraki bölümde simülasyon optimizasyonu için kodlanan genetik algoritma ve benzetimli tavlama algoritmalarına ilişkin kurulan modellere değinilecektir.

6.7 Genetik Algoritma Tabanlı Simülasyon Optimizasyonu Algoritmasının Kurulması

Bir problemin çözülebilmesi için genetik algoritma geliştirilmesi iki farklı karar içerir. İlk karar sorunun nasıl çözüleceği ile alakalıdır. Uygun çözüm aralığının

tanımı, uygunluk fonksiyonun biçimi ve bireylerin kromozomlar şeklinde temsil edilmesini içerir. İkinci karar genetik algorithmada kullanılacak parametreleri içerir. Başlangıç popülasyonu seçimi, çaprazlama, mutasyon ve sonlandırma kriterlerinin seçimidir [1].

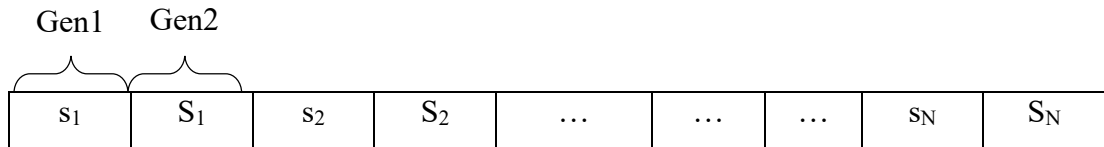
s ve S değerlerinin alt ve üst sınırları sırasıyla [500, 2000], [1000, 2000] olarak tanımlanmıştır. Simülasyon modeli 10 replikasyon boyunca çalıştırılmıştır.

İlerleyen bölümlerde ilk olarak, optimum stok kontrol parametrelerinin çözümü için geliştirilen genetik algorithmada kullanılan seçim, çaprazlama, mutasyon ve sonlandırma operatörleri açıklanmıştır. Sonrasında genetik algoritma operatörlerinin çözüm performansına etkisi için bulunan sonuçlar tartışılmıştır.

6.7.1 Parametrelerin Belirlenmesi

6.7.1.1 Kromozom Temsili

Bu çalışmada kromozomlar “gerçek değer temsili” kodlama yöntemiyle kodlanmıştır [87]. Parametre değerleri genlere gerçek değerleri ile doğrudan aktarılmıştır. GA tasarlanırken öncelikle sipariş edilen ürünün yeniden sipariş edilme noktası ve yeniden sipariş noktası miktarının altına düştüğünde belirlenen maksimum stok seviyesi kodlanmıştır. Yani genlerden biri yeniden sipariş verme noktasının olası bir değerini, genlerden diğeri maksimum stok seviyesinin olası bir değerini ifade etmektedir. Kromozomların yapısı Şekil 6.13’te verilmiştir.



Şekil 6.13 Kromozom yapısı

6.7.1.2 Başlangıç Popülasyonu Oluşturma ve Popülasyon Büyüklüğü

Popülasyon büyüklüğünün belirlenmesi, genetik algorithmada seçilmesi gerekli olan parametrelerden birisidir. Popülasyon genelde rastgele bir şekilde arama

alanından seçilen bireylerden oluşan problemi optimize etmek için aday çözümlerdir [89]. Popülasyon büyüklüğü ile çözüm grubunun büyüklüğü belirlenmektedir. Büyük çözüm grubu olması programın çok uzun süre çalışmasına neden olacak, küçük çözüm grubu olması ise yayılımı azaltarak optimum olmayan bir çözüme yakınsayacaktır. Bu sebeple çözüm uzayı seçimi algoritmanın performansını ve etkinliğini değiştirecektir. Genetik algoritmalarda popülasyonun büyüklüğü ile ilgili kesin bir aralık belirtilmemiştir. Fakat popülasyon büyüklüğü 100 ile 300 arasında seçilmesi önerilmektedir [87]. Literatürde genetik algoritmalarda popülasyon büyüklüğü 10 ile 160 arasında seçilmesi önerilmektedir [79]. Bu çalışmada 80, 100 ve 120 olmak üzere 3 farklı popülasyon büyüklüğü için sonuçlar değerlendirilmiştir. Genetik algoritmanın her iterasyonunda popülasyon sayısı sabit tutulmuştur. Başlangıç popülasyonu oluşturmak için farklı yöntemler uygulanmaktadır. Başlangıç çözümü genetik algoritmanın çoğu uygulamasında rastgele oluşturulur [88]. Bu çalışmada başlangıç çözümü rastgele oluşturulmuştur.

6.7.1.3 Uygunluk Fonksiyonu

Stok kontrol sistemlerinin amaç fonksiyonu toplam stok maliyetini minimize etmektir. Bu sebeple popülasyondaki bireylerin uygunluk değeri, bireylerin maliyeti olarak kullanılmıştır. Her birey için uygunluk değeri maliyet hesaplamalarında verilen toplam stok maliyeti formülüyle Arena programı vasıtasıyla hesaplanmaktadır.

6.7.1.4 Seçim Operatörü

Seçim operatörü, genetik algortmada seçilmesi gereken parametrelerden birisidir. Bireyin seçim olasılığı popülasyondaki bireylerin uygunluk (uyum) fonksiyonu aracılığıyla değerlendirilmektedir. Daha sonra gelecek nesillerin ebeveyni olarak kullanılacak bireyleri seçmek amacıyla bir seçim mekanizması kullanılır [90].

Uygulamalarda en yaygın kullanılan yöntem rulet çarkı yöntemidir [68]. Bu çalışmada en çok kullanılan rulet çarkı yöntemi seçim operatörü olarak belirlenmiştir.

6.7.1.5 Çaprazlama Operatörü

Farklı kromozomların genlerinin değiştirilmesi ile bazı durumlarda daha iyi çözümler bulunurken bazı durumlarda ise kötü sonuçlar elde edilebilir. Genetik algortmada çaprazlama işlemi ile kötü sonuçlu kromozomlardan iyi sonuçlar yaratılabilir [73].

Bu çalışmada çaprazlama operatörü olarak çok noktalı çaprazlama seçilmiştir. Diğer operatörlere göre genetik algoritma da daha etkin biçimde çözüm bulma potansiyeline sahiptir [78]. Çaprazlama işlemi süreci Şekil 6.14'te gösterilmiştir.

Çaprazlama olasılığı parametresi çaprazlanma operatörünün uygulanma sıklığını ifade etmektedir. Çaprazlama olasılığı yüksek olduğunda iyi çözümler iyileştirme yapılmadan daha hızlı bir şekilde atlanabilmektedir. Çaprazlama olasılığı çok düşük olduğunda arama oranının çok düşük olması nedeniyle genetik algoritma durabilmektedir. Çaprazlama olasılığının 0,25 ile 1 arasında 0,05'lik artışlar şeklinde seçilmesi önerilmektedir [79]. Bu çalışmada, 0,70 - 0,75 - 0,80 ve 0,85 olmak üzere 4 farklı çaprazlama değeri kullanılmıştır.

Çaprazlama Noktaları

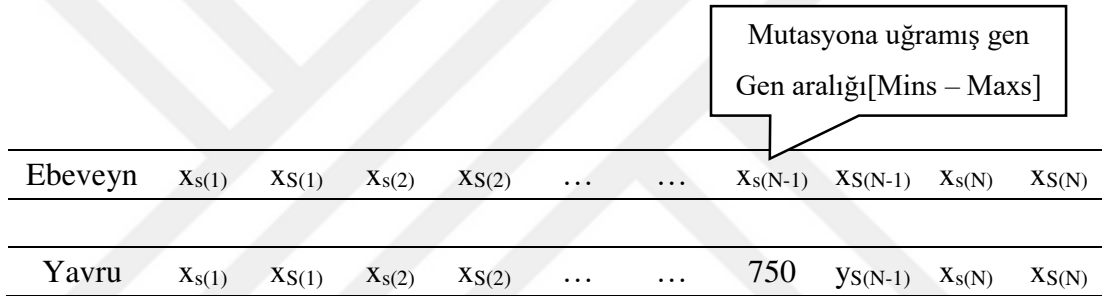
Ebeveyn 1	$x_{s(1)}$	$X_{S(1)}$	$x_{s(2)}$	$X_{S(2)}$	...	...	$x_{s(N-1)}$	$X_{S(N-1)}$	$x_{s(N)}$	$X_{S(N)}$
Ebeveyn 2	$y_{s(1)}$	$Y_{S(1)}$	$y_{s(2)}$	$Y_{S(2)}$	...	...	$y_{s(N-1)}$	$Y_{S(N-1)}$	$y_{s(N)}$	$Y_{S(N)}$
Yavru 1	$y_{s(1)}$	$X_{S(1)}$	$y_{s(2)}$	$X_{S(2)}$	...	...	$y_{s(N-1)}$	$X_{S(N-1)}$	$y_{s(N)}$	$X_{S(N)}$
Yavru 2	$x_{s(1)}$	$Y_{S(1)}$	$x_{s(2)}$	$Y_{S(2)}$	...	...	$x_{s(N-1)}$	$Y_{S(N-1)}$	$x_{s(N)}$	$Y_{S(N)}$

Şekil 6.14 Çok noktalı çaprazlama yöntemi

6.7.1.6 Mutasyon Operatörü

Oluşturulan popülasyonların tekdüze olmaması, çözüm uzayında farklı noktaları arama ve genetik algoritmanın yerel minimuma takılmasını önlemek amacıyla mutasyon operatörü kullanılır. Gerçek değer temsili kodlama yöntemi kullanılması nedeniyle mutasyon yapılacak gen için belirlenen aralıkta bir rastgele bir sayı yaratılır ve gen değeri bu sayı ile değiştirilir [91]. Mutasyon işlemi süreci Şekil 6.15'te gösterilmiştir.

Mutasyon olasılığının genel olarak [0.01, 0.001] arasında olması önerilmektedir [79]. Bu çalışmada mutasyon olasılığı olarak 0,001 - 0,03 - 0,06 ve 0,01 olmak üzere 4 farklı mutasyon operatörü kullanılmıştır.



Şekil 6.15 Mutasyon

6.7.1.7 Sonlandırma Operatörü

Genetik algoritma sonlandırma operatörü koşulu sağlandığında durmaktadır. Genellikle durdurma koşulu iki faktöre bağlı olan iterasyon sayısı olarak belirlenebilir. İlk faktör evrim yapısının karmaşıklığıdır. Potansiyel çözüm sayısı arttıkça algoritmanın çalışma süresi uzamaktadır. Diğer faktör ise çözüm uzayının büyüklüğüdür. Çözüm uzayı büyük olursa daha az sayıda iterasyon ile optimal çözüme ulaşma olasılığı artmaktadır [92].

Genetik algoritma sonlandırma koşulu için bir diğer yaklaşım da iterasyon sayısı önceden belirlenmiş kabul edilebilir bir çözüme ulaşılması veya belirli sayıda tekrarın aşılıp aşılmasına bağlıdır. Kromozomlar ve kromozamlara bağlı maliyetler arasında bir fark kalmadığında iterasyon durdurulmaktadır [93].

Kromozomların uygunluk değerlerinin eşit olduğu durum yani jenerasyon arasında bir fark kalmadığı durumda durdurulmaktadır [94].

Bu problemde uygunluk değerlerinin 5 iterasyon boyunca eşit çıkması durumunda sonraki iterasyonlarda genellikle eşit olduğu gözlemlenmiştir. Bu nedenle bu çalışmada algoritmanın çözüm süresini optimal tutabilmek için uygunluk değeri 5 iterasyon üst üste eşit çıkması durumunda iterasyona durdurma koşulu eklenmiştir.

6.7.2 Genetik Algoritma ile İlgili Sayısal Örnek

Belirlenen sipariş miktarı ve maksimum sipariş noktası aralığında oluşturulan N adet rastgele bir başlangıç popülasyonu Şekil 6.16'da verilmiştir.

s_1	S_1	s_2	S_2			s_{N-1}	S_{N-1}	s_N	S_N
1720	1904	1665	1940	...	...	1688	1937	1796	1926

Şekil 6.16 Rastgele başlangıç popülasyonunun oluşturulması

Başlangıç popülasyonunda oluşturulan kromozomların uygunluk fonksiyonu değerlerine göre, rulet çarkı yöntemi ile çaprazlanacak ebeveynler seçilir. Şekil 6.17'de rulet çarkı yöntemiyle seçilen bireyler verilmiştir.

1720	1904	1858	1866	...	...	1688	1937	1834	1963
------	------	-------------	-------------	-----	-----	------	------	-------------	-------------

Şekil 6.17 Rulet çarkı yöntemi ile seçilen bireyler

Seçilen ebeveynlere yeniden sipariş noktası değerinin (s) maksimum sipariş miktarı değerinden (S) küçük olma koşulu ve çaprazlama olasılığına göre çok noktalı çaprazlama yöntemi ile çaprazlanma işlemi yapılır. Çaprazlanan ebeveynler ve oluşan yavrular Şekil 6.18'de verilmiştir.

Çaprazlama Noktaları

Ebeveyn 1	1720	1904	...	...	1688	1937
Ebeveyn 2	1858	1866	...	...	1834	1963
Yavru 1	1858	1904	...	...	1834	1937
Yavru 2	1720	1866	...	...	1688	1963

Şekil 6.18 Çok noktalı çaprazlama işlemi

Rulet çarkı yöntemi ile seçilen bireylere mutasyon olasılığına göre mutasyon işlemi yapılır. Şekil 6.19’da mutasyon işlemi gösterilmiştir.

Ebeveyn	1720	1904	1858	1866	...	...	1688	1937	1834	1963
Yavru	1720	1904	1858	1866	...	...	750	1937	1834	1963

Mutasyona uğramış gen
Gen aralığı[500 – 2000]

Şekil 6.19 Mutasyon işlemi

Bu işlemler GA sonlandırma koşulu sağlanana kadar devam etmektedir.

6.7.3 Genetik Algoritma Operatörlerinin Çözüm Performansına Etkisi

Genetik algoritma operatörleri ve genetik algoritma operatörlerinin oranı, genetik algoritma problemlerinde optimum veya optimuma yakın çözümlerin elde edilmesinde etkili olduğu için (performansa etkisi olduğu için), parametrelerin üzerinde denemeler yapılarak optimum değerleri bulunamasa da en uygun değerleri bulunmaya çalışılmıştır. Bu bölümde farklı genetik algoritma operatörlerinin çözümü nasıl etkilediği incelenmiştir. Genetik algoritmanın rassallığından kaynaklı değişkenliği önlemek amacıyla incelenen her parametre değeri için genetik algoritma

beş kere çalıştırılarak sonuçların ortalama değerleri de verilmiştir. Genetik algoritmanın rassallığına bağlı olarak çaprazlama ve mutasyon olasılığı değişkenlik göstermektedir.

Genetik algoritma kodu teknik bir programlama dili olan Matlab programında kodlanmıştır. Bütün çözümler AMD A8-4500M APU with Radeon (tm) HD Graphics 1.90 GHz işlemciye sahip bilgisayarda çalıştırılmıştır. Programın çalışma süresi saat, dakika ve saniye olarak verilmiştir.

6.7.3.1 Popülasyon Büyüklüğü

Popülasyon büyüklüğü etkinliğinin belirlenebilmesi için genetik algoritma parametrelerinden çaprazlama olasılığı 0,85 ve mutasyon olasılığı 0,01 olarak sabit tutulmuştur. Popülasyon büyüklüğü değiştirildiğinde etkilerinin incelenebilmesi için popülasyon büyüklüğü sabit tutulmamıştır. Bu çalışmada, 80, 100 ve 120 olmak üzere üç farklı popülasyon büyüklüğü kullanılmıştır.

Genetik algoritma popülasyon büyüklüğü 80 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.1’de verilmiştir.

Tablo 6.1 80’lik popülasyon büyüklüğü için elde edilen sonuçlar

Popülasyon Büyüklüğü: 80					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1903	1994	1.357.024,76	00:45:40	17
2. çözüm	1893	1981	1.373.680,31	00:43:04	14
3. çözüm	1867	1997	1.393.682,50	00:38:16	13
4. çözüm	1937	1997	1.363.701,72	00:40:50	13
5. çözüm	1988	1998	1.364.370,50	00:27:39	10
Ortalama			1.370.491,96	00:39:06	13,4

Genetik algoritma popülasyon büyüklüğü 100 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.2’de verilmiştir.

Tablo 6.2 100'lük popülasyon büyüklüğü için elde edilen sonuçlar

Popülasyon Büyüklüğü: 100					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1790	1988	1.394.071,47	01:08:08	20
2. çözüm	1939	1999	1.360.525,10	00:43:24	13
3. çözüm	1905	1997	1.350.595,62	00:40:05	12
4. çözüm	1902	1993	1.356.416,85	00:36:35	11
5. çözüm	1982	1989	1.361.726,29	00:43:12	12
Ortalama			1.364.667,07	00:46:17	13,6

Genetik algoritma popülasyon büyüklüğü 120 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.3'de verilmiştir.

Tablo 6.3 120'lik popülasyon büyüklüğü için elde edilen sonuçlar

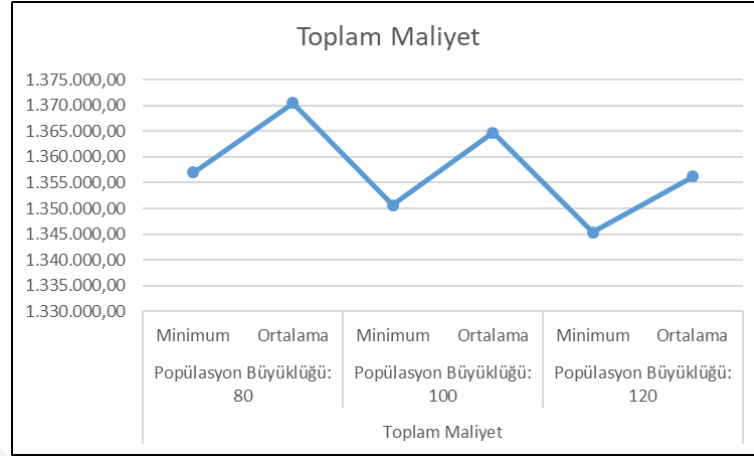
Popülasyon Büyüklüğü: 120					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1892	1985	1.360.084,71	01:20:09	18
2. çözüm	1900	1992	1.359.124,38	01:05:00	16
3. çözüm	1900	1991	1.360.806,51	01:13:16	18
4. çözüm	1904	1997	1.345.356,50	00:55:44	17
5. çözüm	1968	1996	1.355.497,53	01:06:19	14
Ortalama			1.356.173,93	01:08:06	16,6

Popülasyon büyüklüğü için toplam maliyet karşılaştırılması Tablo 6.4'de verilmiştir. Tabloda her popülasyon büyüklüğü için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.4 Popülasyon büyüklükleri için maliyet karşılaştırması

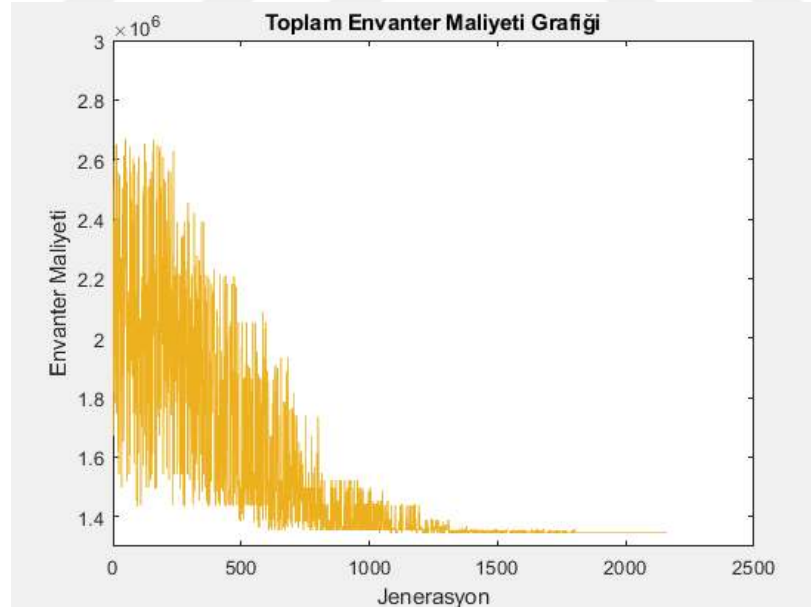
Toplam Maliyet					
Popülasyon Büyüklüğü: 80		Popülasyon Büyüklüğü: 100		Popülasyon Büyüklüğü: 120	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.357.024,76	1.370.491,96	1.350.595,62	1.364.667,07	1.345.356,50	1.356.173,93

Popülasyon büyüklüklerine göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.20’de verilmiştir.



Şekil 6.20 Popülasyon büyüklükleri için maliyet karşılaştırılması

Şekil 6.21’de popülasyon büyüklüğüne göre en düşük maliyeti veren (1904, 1997) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.21 (1904, 1997) parametreleri için jenerasyon-maliyet grafiği

Küçük popülasyonlar daha hızlı fakat daha az çeşitli sonuçlar vermektedir, büyük popülasyonların ise daha çeşitli ve kaliteli çözümler üretmektedir, ancak daha

fazla hesaplama gücü gerektirmektedir. Popülasyon büyüklüğünün, genetik algoritmanın çözüm kalitesi ve hesaplama süresi üzerinde önemli bir etkisi olduğunu görülmektedir [103].

Genetik algoritma popülasyon sayısı arttıkça daha iyi bir çözüme yakınsamaktadır. Bununla birlikte genetik algoritma popülasyon sayısı arttığında programın çalışma süresinin arttığı gözlenmektedir.

Yapılan çalışmada 120 popülasyon büyüklüğü en düşük maliyetli stok kontrol parametrelerini vermiştir.

6.7.3.2 Çaprazlama Operatörü

Çaprazlama operatörü olasılığının etkinliğinin belirlenebilmesi için genetik algoritma parametrelerinden mutasyon olasılığı 0,01 ve en iyi sonucu veren popülasyon büyüklüğü değeri 120 olarak sabit tutulmuştur. Çaprazlama olasılığı büyüklüğü değiştirildiğinde çözüm üzerindeki etkilerinin incelenebilmesi için çaprazlama olasılığı sabit tutulmamıştır. Bu çalışmada, 0,70 - 0,75 - 0,80 ve 0,85 olmak üzere 4 farklı çaprazlama olasılığı değeri kullanılmıştır.

Genetik algoritma çaprazlama olasılığı 0,70 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.5'te verilmiştir.

Tablo 6.5 0,70'lik çaprazlama olasılığı için elde edilen sonuçlar

P_c: 0,70					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1898	1982	1.392.050,43	01:04:43	15
2. çözüm	1889	1993	1.367.135,59	01:14:59	18
3. çözüm	1785	1989	1.389.618,86	01:00:04	15
4. çözüm	1871	1994	1.394.142,73	00:57:27	15
5. çözüm	1960	1996	1.368.129,30	00:45:28	11
Ortalama			1.382.215,38	01:00:32	14,8

Genetik aprazlama olasılıđı 0,75 olduđuunda bulunan maliyet, programın alıřma sũresi ve iterasyon sayısı Tablo 6.6’da verilmiřtir.

Tablo 6.6 0,75’lik aprazlama olasılıđı iin elde edilen sonular

P_c: 0,75					
	s	S	Maliyet	Programın alıřma Sũresi	İterasyon
1. özũm	1903	1995	1.355.579,72	01:09:31	16
2. özũm	1795	1994	1.379.423,82	01:01:41	15
3. özũm	1948	1997	1.375.259,85	01:11:08	17
4. özũm	1889	1982	1.363.908,01	01:10:07	17
5. özũm	1967	1996	1.360.685,96	00:58:16	14
Ortalama			1.366.971,47	01:07:16	15,8

Genetik algoritma aprazlama olasılıđı 0,80 olduđuunda bulunan maliyet, programın alıřma sũresi ve iterasyon sayısı Tablo 6.7’de verilmiřtir.

Tablo 6.7 0,80’lik aprazlama olasılıđı iin elde edilen sonular

P_c: 0,80					
	s	S	Maliyet	Programın alıřma Sũresi	İterasyon
1. özũm	1976	1982	1.363.947,07	01:16:12	18
2. özũm	1880	1999	1.383.407,73	00:54:32	13
3. özũm	1902	1995	1.350.124,44	01:10:11	17
4. özũm	1908	1996	1.360.053,10	01:21:25	19
5. özũm	1937	1997	1.363.701,72	00:55:36	17
Ortalama			1.364.246,81	01:07:35	16,8

Genetik algoritma aprazlama olasılıđı 0,85 olduđuunda bulunan maliyet, programın alıřma sũresi ve iterasyon sayısı Tablo 6.8’de verilmiřtir.

Tablo 6.8 0,85'lik çaprazlama olasılığı için elde edilen sonuçlar

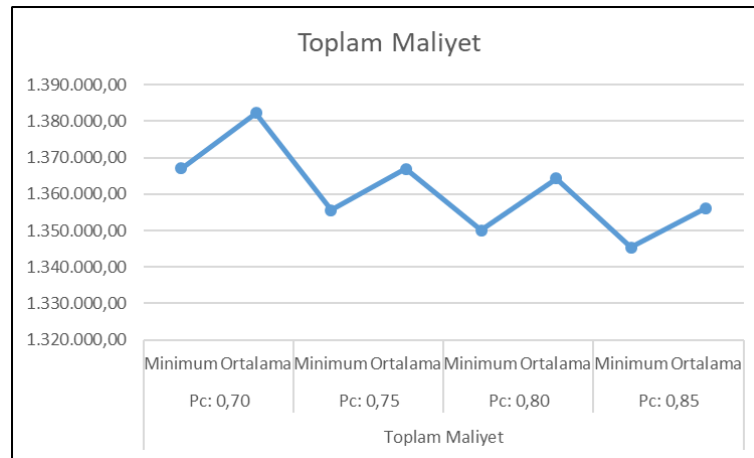
P_c: 0,85					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1892	1985	1.360.084,71	01:20:09	18
2. çözüm	1900	1992	1.359.124,38	01:05:00	16
3. çözüm	1900	1991	1.360.806,51	01:13:16	18
4. çözüm	1904	1997	1.345.356,50	00:55:44	17
5. çözüm	1968	1996	1.355.497,53	01:06:19	14
Ortalama			1.356.173,93	01:08:06	16,6

Çaprazlama olasılığı için toplam maliyet karşılaştırılması Tablo 6.9 verilmiştir. Tabloda her çaprazlama olasılığı için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.9 Çaprazlama olasılıkları için maliyet karşılaştırması

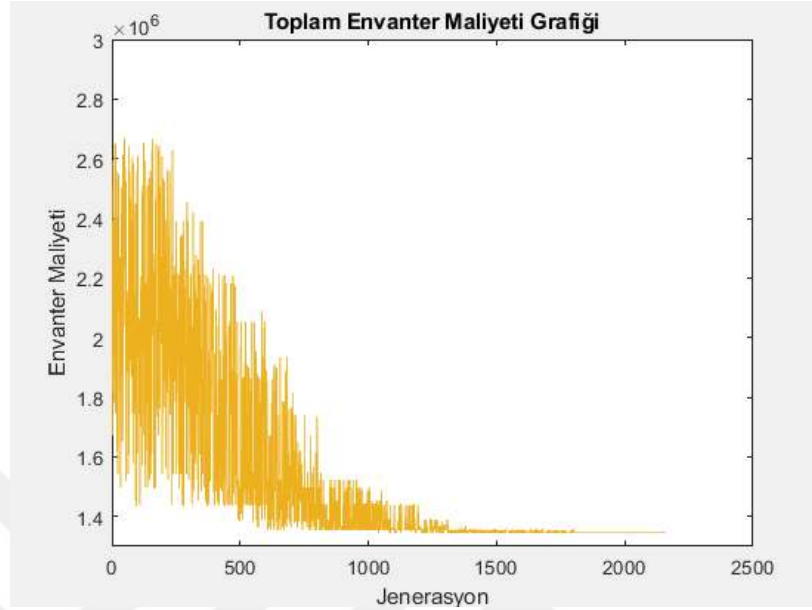
Toplam Maliyet							
P_c: 0,70		P_c: 0,75		P_c: 0,80		P_c: 0,85	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.367.135,59	1.382.215,38	1.355.579,72	1.366.971,47	1.350.124,44	1.364.246,81	1.345.356,50	1.356.173,93

Çaprazlama olasılıklarına göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.22'de verilmiştir.



Şekil 6.22 Çaprazlama olasılıkları için maliyet karşılaştırılması

Şekil 6.23'te çaprazlama olasılığına göre en düşük maliyeti veren (1904, 1997) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.23 (1904,1997) parametreleri için jenerasyon-maliyet grafiği

Optimal çaprazlama olasılığının, probleme ve popülasyon büyüklüğüne bağlı olarak değişmekle birlikte, genellikle $[0,25, 1]$ arasında olduğu belirtilmiştir. Optimal oranların belirlenmesinde deneme-yanılma yönteminin ve problemin doğasının dikkate alınması gerektiği ifade edilmiştir [79].

Genetik algoritma çaprazlama olasılığı arttıkça daha iyi bir çözüme yakınsadığı gözlemlenmektedir. Genetik algoritma çaprazlama olasılığı arttıkça çaprazlama işlemi daha yüksek olasılıkla yapıldığı için programın aynı iterasyon sayısında daha uzun süre çalışması beklenmektedir. Fakat genetik algoritmanın rassallığından dolayı çaprazlama ve mutasyon için farklı rassal işlemler yapıldığı için programın çalışma süresi değişkenlik göstermiştir.

Yapılan çalışmada 0,85 çaprazlama olasılığı en düşük maliyetli stok kontrol parametrelerini vermiştir.

6.7.3.3 Mutasyon Operatörü

Mutasyon operatörü olasılığının etkinliğinin belirlenebilmesi için genetik algoritma parametrelerinden en iyi sonucu veren popülasyon büyüklüğü değeri 120 ve en iyi sonucu veren çaprazlama olasılığı 0,85 olarak sabit tutulmuştur. Mutasyon olasılığı büyüklüğü değiştirildiğinde çözüm üzerindeki etkilerinin incelenebilmesi için mutasyon olasılığı sabit tutulmamıştır. Bu çalışmada, mutasyon olasılığı olarak 0,001 - 0,03 - 0,06 - 0,01 olmak üzere 4 farklı çaprazlama olasılığı değeri kullanılmıştır.

Genetik algoritma mutasyon olasılığı 0,001 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.10'da verilmiştir.

Tablo 6.10 0,001'lik mutasyon olasılığı için elde edilen sonuçlar

P_m: 0,001					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1882	1984	1.382.191,79	00:49:58	12
2. çözüm	1958	2000	1.361.639,88	01:11:52	17
3. çözüm	1893	1997	1.363.432,67	01:18:12	18
4. çözüm	1914	1999	1.365.716,09	01:36:18	22
5. çözüm	1980	1986	1.361.446,79	00:36:19	11
Ortalama			1.366.885,44	01:06:32	16

Genetik mutasyon olasılığı 0,03 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.11'de verilmiştir.

Tablo 6.11 0,03'lük mutasyon olasılığı için elde edilen sonuçlar

P_m: 0,03					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1906	1999	1.345.866,36	00:49:22	15
2. çözüm	1895	1988	1.355.550,90	01:05:42	15
3. çözüm	1982	1988	1.360.936,47	00:56:50	13
4. çözüm	1909	1998	1.352.048,85	01:17:57	18
5. çözüm	1896	1996	1.372.692,36	01:05:46	20
Ortalama			1.357.418,99	01:03:07	16,2

Genetik algoritma mutasyon olasılığı 0,06 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.12’de verilmiştir.

Tablo 6.12 0,06’lık mutasyon olasılığı için elde edilen sonuçlar

P_m: 0,06					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1973	2000	1.352.210,74	00:46:05	14
2. çözüm	1905	1998	1.344.728,96	00:56:05	17
3. çözüm	1903	1996	1.347.415,24	01:12:20	22
4. çözüm	1910	1998	1.356.667,51	01:02:47	19
5. çözüm	1904	1995	1.354.759,11	01:18:21	24
Ortalama			1.351.156,31	01:03:08	19,2

Genetik algoritma mutasyon olasılığı 0,01 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.13’te verilmiştir.

Tablo 6.13 0,01’lik mutasyon olasılığı için elde edilen sonuçlar

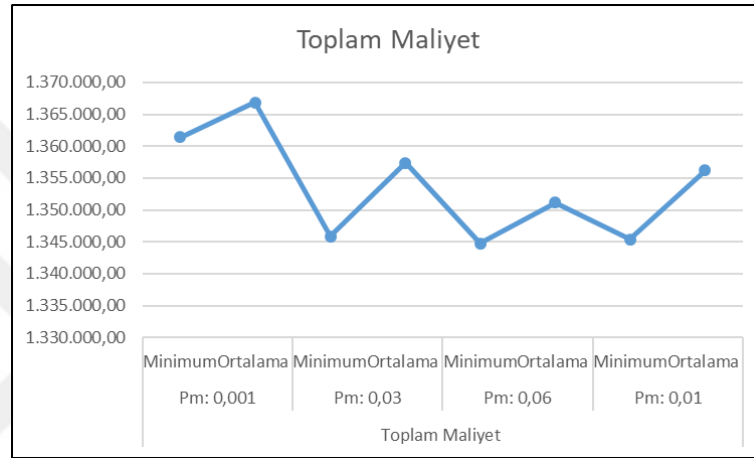
P_m: 0,01					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1892	1985	1.360.084,71	01:20:09	18
2. çözüm	1900	1992	1.359.124,38	01:05:00	16
3. çözüm	1900	1991	1.360.806,51	01:13:16	18
4. çözüm	1904	1997	1.345.356,50	00:55:44	17
5. çözüm	1968	1996	1.355.497,53	01:06:19	14
Ortalama			1.356.173,93	01:08:06	16,6

Mutasyon olasılığı için toplam maliyet karşılaştırılması Tablo 6.14’te verilmiştir. Tabloda her mutasyon olasılığı için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.14 Mutasyon olasılıkları için maliyet karşılaştırması

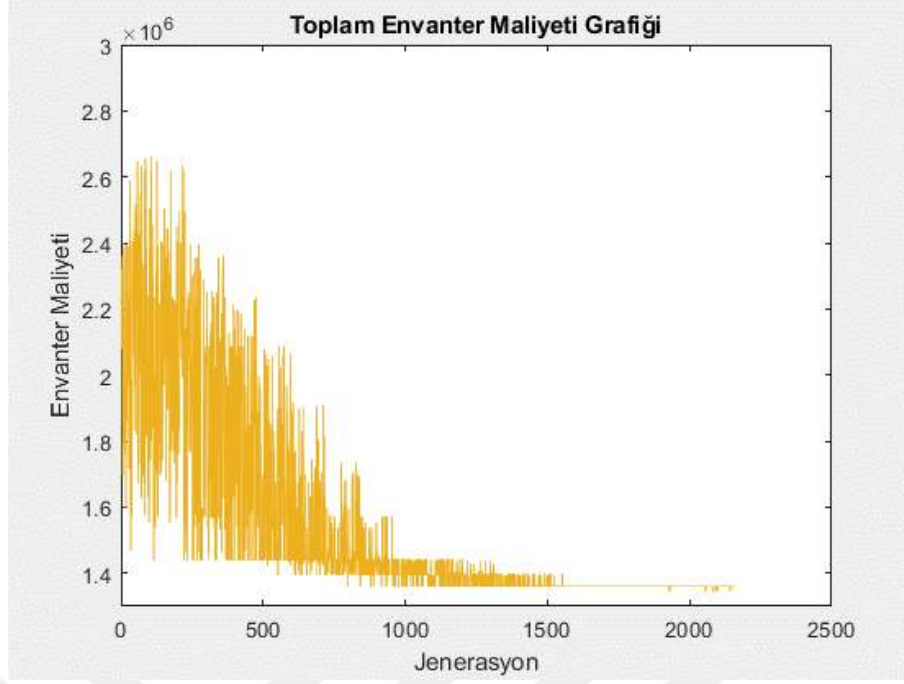
Toplam Maliyet							
P _m : 0,001		P _m : 0,03		P _m : 0,06		P _m : 0,01	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.361.446,79	1.366.885,44	1.345.866,36	1.357.418,99	1.344.728,96	1.351.156,31	1.345.356,50	1.356.173,93

Mutasyon olasılıklarına göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.24'te verilmiştir.



Şekil 6.24 Mutasyon olasılıkları için maliyet karşılaştırılması

Şekil 6.25'te mutasyon olasılığına göre en düşük maliyeti veren (1905, 1988) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.25 (1905, 1988) parametreleri için jenerasyon-maliyet grafiği

Düşük mutasyon olasılıklarının (genellikle %1-5) en uygun sonuçları verdiğini belirtilmiştir, çünkü bu oranlar çözüm alanını etkin bir şekilde araştırmayı sağlar ve genetik çeşitliliği korur. Mutasyon olasılığının düşük tutulmasının, algoritmanın genel performansını artırdığını ve çözüm kalitesini yükselttiği vurgulanmıştır [95].

Mutasyon olasılığının çok düşük olması herhangi bir genin sonsuza kadar tek bir değerde kalarak popülasyonun sabit bir değere yakınsamasına neden olabilir. Genetik algoritma mutasyon olasılığı azaldıkça sabit bir değere yakınsama olasılığı yükseldiği için iterasyonun daha erken biteceği varsayılmaktadır. Genetik algoritma rassallığından dolayı 0,01 mutasyon olasılığı için iterasyon daha kısa sürmüştür.

Yapılan çalışmada 0,06 mutasyon olasılığı en düşük maliyetli stok kontrol parametrelerini vermiştir.

6.8 Benzetimli Tavlama Tabanlı Simülasyon Optimizasyonu Algoritmasının Kurulması

Benzetimli tavlama algoritmasını kullanarak bir optimizasyon problemini çözerken, kontrol değişkenlerinin doğru temsili önemlidir, ancak yeni çözümlerin üretilme şekli de önemlidir. Çözüm oluşturucu, küçük rastgele değişiklikler yaparak ve tüm olası çözümlere ulaşılmasına izin vererek çözümün çeşitliliğini artırmalıdır. Bu, algoritmanın daha geniş bir arama alanını keşfetmesine ve daha iyi sonuçlar elde etmesine yardımcı olabilir [96].

İlerleyen bölümlerde ilk olarak optimum stok kontrol parametrelerinin çözümü için geliştirilen benzetimli tavlama algoritması için probleme özgü kullanılan parametreler verilmiştir. Sonrasında benzetimli tavlama operatörlerinin çözüm performansına etkisi için bulunan sonuçlar tartışılmıştır.

s ve S değerlerinin alt ve üst sınırları sırasıyla [500, 2000] , [1000, 2000] olarak tanımlanmıştır. Simülasyon modeli 10 replikasyon boyunca çalıştırılmıştır.

Bir sorunun çözümünde, çözümün kalitesini ölçmek için bir yöntem belirlemek önemlidir. Bu maliyet fonksiyonunun tanımlanması, çözülmeye çalışılan sorunu doğru şekilde temsil ettiğinden emin olunmalıdır. Her iterasyonda hesaplanacağı için, maliyet fonksiyonunun etkin bir şekilde hesaplanabilir olması da kritiktir. Mümkünse, maliyet fonksiyonu aynı değeri döndüren çok sayıda durumdan kaçınarak, aramanın yönünü de belirlemelidir. Bu, algoritmanın doğru yönde ilerlemesini sağlar ve daha etkili bir çözüm bulunmasına yardımcı olabilir [97]. Stok kontrol sistemlerinin amaç fonksiyonu toplam stok maliyetini minimize etmektir. Bu sebeple aday çözümlerin amaç fonksiyonu değeri maliyet olarak kullanılmıştır. Her aday çözüm için amaç fonksiyonu değeri maliyet hesaplamalarında verilen toplam stok maliyeti formülüyle Arena programı vasıtasıyla hesaplanmaktadır.

6.8.1 Benzetimli Tavlama Algoritması Parametreleri

Benzetimli tavlama algoritmasının uygulanabilmesi için başlangıç sıcaklığı belirleme, sıcaklık azaltma fonksiyonu, soğutma oranı ve algoritmayı durdurma kriteri

gibi kontrol parametrelerinin belirlenmesi gereklidir [98]. İlerleyen bölümlerde ilk olarak optimum stok kontrol parametrelerinin çözümü için geliştirilen SA için kullanılan başlangıç sıcaklığı, soğutma planı, kabul fonksiyonu ve sonlandırma kriteri açıklanmıştır. Sonrasında benzetimli tavlama operatörlerinin çözüm performansına etkisi için bulunan sonuçlar tartışılmıştır.

6.8.1.1 Başlangıç Sıcaklığı

Başlangıç sıcaklığı bir girdi parametresidir. Başlangıç çözümlerinin kabul edilme olasılığının yüksek olması için başlangıç sıcaklığının yeterince büyük olması önemlidir. Ancak, çok yüksek bir başlangıç sıcaklığı seçmek, algoritmanın çok uzun süre çalışmasına ve kötü bir performansa neden olabilir. Bu nedenle, başlangıç sıcaklığının dengeli bir şekilde seçilmesi, algoritmanın verimli bir şekilde çalışmasını sağlamak için önemlidir [98]. Başlangıç sıcaklığı, hemen hemen her komşu duruma geçişe izin verecek kadar yüksek olmalıdır. Aksi takdirde, sonuç çözümü başlangıç çözümü ile aynı (veya çok yakın) olabilir [97].

Bu çalışmada başlangıç sıcaklıkları problemin yapısına uygun olarak 5.000 - 7.000 - 10.000 olarak seçilmiştir.

6.8.1.2 Soğutma Planı

Soğutma planı belirlenen bir sıcaklık düşürme şekline göre her iterasyonda sıcaklığı azaltmaktadır. Sıcaklığı azaltma şekli algoritmanın başarısı açısından yüksek öneme sahiptir [97].

Literatürde farklı soğutma prensipleri kullanılmaktadır. En yaygın kullanılan geometrik soğutma planı aşağıdaki gibidir. Bu çalışmada aşağıdaki geometrik soğutma planı kullanılmıştır.

$$T_k = T_{k-1} * \alpha \quad (6.2)$$

Bu formülde T_k , k'nci iterasyondaki sıcaklık değerini, T_{k-1} , (k-1)'inci iterasyondaki sıcaklık değerini temsil etmektedir. α ise sabit bir orandır ve soğutma oranını temsil etmektedir [96].

Soğutma oranı (α), genellikle 0,85 ile 0,99 aralığında değer almaktadır [100]. Kirkpatrick çalışmasında soğutma oranını 0,8 ile 0,95 arasında kullanmıştır [82]. Bazı çalışmalarda soğutma oranının (0,1) arasında olabileceği belirtilmiştir [99]. Tabii ki, α 'nın değeri ne kadar yüksek olursa, sıcaklığın durma kriterine düşürülmesi o kadar uzun sürecektir [97]. Bu çalışmada algoritmanın soğutma oranı değeri, tüm aralıkları kapsayacak şekilde 0,4 - 0,7 - 0,85 olarak seçilmiştir.

6.8.1.3 Kabul Fonksiyonu

Hesaplama adımlarında, mevcut çözümün komşu çözümleri uygun komşuluk mekanizmaları kullanılarak üretilir. Oluşturulan komşu çözümün amacı hesaplanarak, mevcut çözümden daha iyi ise yeni çözüm olarak kabul edilir; aksi takdirde, belirlenen kritere göre kabul edilir veya reddedilir. Hesaplama adımlarının sonunda sıcaklık, belirli bir fonksiyona göre azaltılır [98].

Yeni bulunan komşu çözüm, önceki amaç fonksiyonundan daha iyi ise kabul edilir. Bu, minimizasyon problemi ise, en düşük değere bakılır ve arama devam eder. Ancak, komşu çözümlere bakarken daha kötü bir çözüm bulunursa, bu çözümün kabul olasılıklarına göre değerlendirilir. Çözümlerin kabul olasılıkları değerlendirildikten sonra, sıcaklık kademeli olarak azaltılarak arama işlemi komşu çözümler üzerinde devam eder [101]. Formüldeki $f(x') - f(x)$ daha kötü olan çözümün amaç fonksiyonu ile mevcut çözümün amaç fonksiyonu arasındaki fark değerini T ise o adımdaki sıcaklığı belirtmektedir. Sıcaklık azaldıkça kabul olasılığı da azalır, bu da çözümlerin optimal çözüme daha yakın hale gelmesini sağlar [102].

$$P(\Delta E, T) = e^{-\frac{f(x') - f(x)}{T}} \quad (6.3)$$

6.8.1.4 Durdurma Kriteri

Algoritmanın durdurulması için çeşitli yaklaşımlar mevcuttur. Bunlar arasında, belirlenen maksimum iterasyon sayısına ulaşılması, belirlenen en düşük sıcaklık seviyesine inilmesi ve istenen kriterleri sağlayan bir çözüme ulaşılması yer alır. Bu çeşitli durdurma kriterleri, algoritmanın başarılı bir şekilde tamamlanmasını sağlamak için kullanılabilir [25].

Bu çalışmada, bitirme koşulu olarak, önceden belirlenen bir maksimum iterasyon sayısına ulaşıldığında algoritma sonlanır ve bulunan en iyi çözüm sonuç olarak kabul edilir.

6.8.2 Benzetimli Tavlama Algoritması ile İlgili Sayısal Örnek

Belirlenen sipariş miktarı ve maksimum sipariş noktası aralığında rastgele oluşturulan bir başlangıç çözümü Şekil 6.26’da verilmiştir.

S ₁		S ₁	
1767		1966	

Şekil 6.26 Benzetimli tavlama algoritması için başlangıç çözümü

Sipariş miktarı ve maksimum sipariş noktası aralığına uygun bir komşu çözüm bulunur. Şekil 6.27’de komşu çözüm verilmiştir.

1744	1971
------	------

Şekil 6.27 Komşu çözüm

Komşu çözümün amaç fonksiyonu değeri mevcut çözümden daha iyiye komşu çözüm yeni çözüm olarak kaydedilerek, komşu çözümden devam edilir. Eğer komşu çözümün amaç fonksiyonu değeri, mevcut çözümün amaç fonksiyonu değerinden daha kötü ise, komşu çözümden devam edilip edilmeyeceğine bir olasılıkla karar verilir.

Mevcut çözümün amaç fonksiyonu değeri (1.410.596,76), komşu çözümün amaç fonksiyonu değerinden (1.433.891,00) büyük olduğu için kabul olasılığı (0,001) hesaplanır. Rasgele üretilen sayı kabul olasılığından küçük olduğunda komşu çözüm kabul edilir.

Bu işlemler sonlandırma koşulu sağlanana kadar devam eder.

6.8.3 Benzetimli Tavlama Algoritması Operatörlerinin Çözüm Performansına Etkisi

Benzetimli tavlama algoritması operatörleri, benzetimli tavlama algoritması problemlerinde optimum veya optimuma yakın çözümlerin elde edilmesinde etkilidir. Bu parametrelerin optimum değerleri bulunamasa da çeşitli denemeler yaparak uygun değerler bulunmaya çalışılmıştır. Bu bölümde farklı benzetimli tavlama algoritması operatörlerinin çözümü nasıl etkilediği incelenmiştir. Benzetimli tavlama algoritması rassallığından kaynaklı değişkenliği önlemek amacıyla incelenen her parametre değeri için benzetimli tavlama algoritması beş kere çalıştırılarak sonuçların ortalama değerleri de verilmiştir.

Benzetimli tavlama algoritması kodu teknik bir programlama dili olan Matlab programında kodlanmıştır. Bütün çözümler AMD A8-4500M APU with Radeon (tm) HD Graphics 1.90 GHz işlemciye sahip bilgisayarda çalıştırılmıştır. Programın çalışma süresi saat, dakika ve saniye olarak verilmiştir.

6.8.3.1 Başlangıç Sıcaklığı

Başlangıç sıcaklığının etkinliğinin belirlenebilmesi için benzetimli tavlama parametrelerinden soğutma oranı 0,85 olarak sabit tutulmuş ve 500 iterasyon boyunca çalıştırılmıştır. Başlangıç sıcaklığı değiştirildiğinde etkilerinin incelenebilmesi için başlangıç sıcaklığı sabit tutulmamıştır. Bu çalışmada, 5.000 - 7.000 - 10.000 olmak üzere üç farklı başlangıç sıcaklığı değeri kullanılmıştır.

Benzetimli tavlama başlangıç sıcaklığı 5.000 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.15'te verilmiştir.

Tablo 6.15 5.000 başlangıç sıcaklığı için elde edilen sonuçlar

T₀: 5.000					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1907	1997	1.349.681,78	00:18:32	500
2. çözüm	1800	2000	1.374.323,86	00:17:47	500
3. çözüm	1794	1994	1.378.917,50	00:20:43	500
4. çözüm	1906	1998	1.349.968,08	00:23:52	500
5. çözüm	1798	1998	1.376.465,61	00:19:49	500
Ortalama			1.365.871,37	00:20:09	500

Benzetimli tavlama başlangıç sıcaklığı 7.000 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.16’da verilmiştir.

Tablo 6.16 7.000 başlangıç sıcaklığı için elde edilen sonuçlar

T₀: 7.000					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1796	1999	1.379.365,79	00:20:38	500
2. çözüm	1796	1995	1.377.646,76	00:24:03	500
3. çözüm	1800	2000	1.374.323,86	00:19:41	500
4. çözüm	1797	1996	1.377.246,99	00:18:42	500
5. çözüm	1798	1998	1.376.465,61	00:18:58	500
Ortalama			1.377.009,80	00:20:24	500

Benzetimli tavlama başlangıç sıcaklığı 10.000 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.17’de verilmiştir.

Tablo 6.17 10.000 başlangıç sıcaklığı için elde edilen sonuçlar

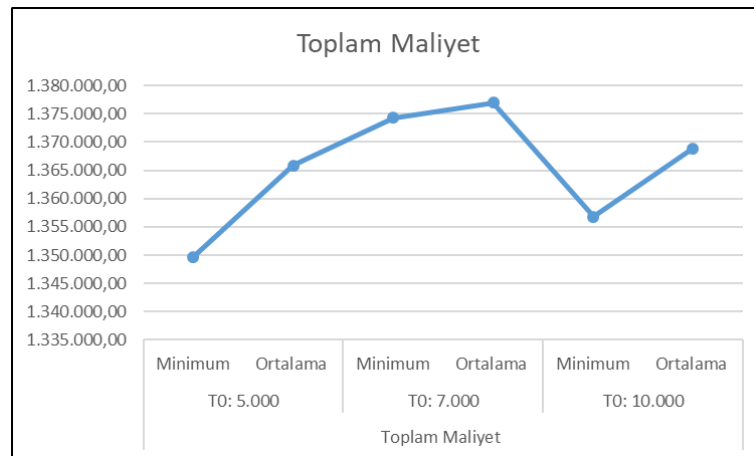
T₀: 10.000					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1799	1999	1.376.019,84	00:20:22	500
2. çözüm	1902	1994	1.357.723,81	00:18:32	500
3. çözüm	1801	2000	1.374.670,29	00:19:08	500
4. çözüm	1903	1993	1.356.791,21	00:17:36	500
5. çözüm	1835	1999	1.379.151,69	00:20:05	500
Ortalama			1.368.871,37	00:19:09	500

Başlangıç sıcaklığı için toplam maliyet karşılaştırılması Tablo 6.18’de verilmiştir. Tabloda her başlangıç sıcaklığı büyüklüğü için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.18 Başlangıç sıcaklıkları için maliyet karşılaştırması

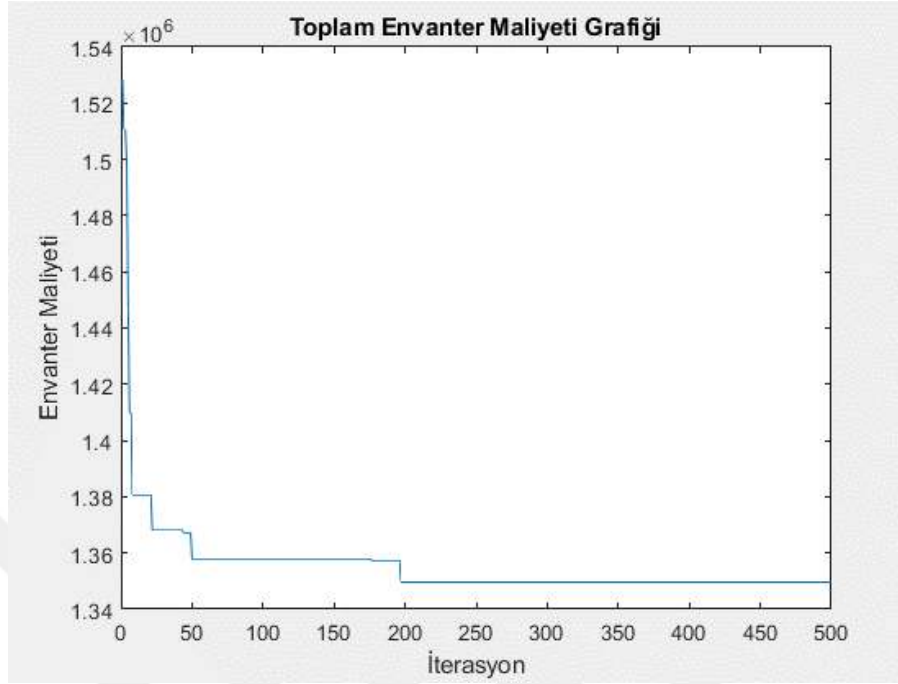
Toplam Maliyet					
T₀: 5.000		T₀: 7.000		T₀: 10.000	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.349.681,78	1.365.871,37	1.374.323,86	1.377.009,80	1.356.791,21	1.368.871,37

Başlangıç sıcaklığı değerlerine göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.28’de verilmiştir.



Şekil 6.28 Başlangıç sıcaklıkları için maliyet karşılaştırılması

Şekil 6.29’da başlangıç sıcaklıklarına göre en düşük maliyeti veren (1907, 1997) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.29 (1907, 1997) parametreleri için iterasyon-maliyet grafiği

Benzetimli tavlama algoritmasındaki başlangıç sıcaklığı, algoritmanın başlangıçta çözüm uzayını ne kadar geniş bir şekilde araştıracağını belirler. Yüksek bir başlangıç sıcaklığı, algoritmanın daha kötü çözümleri daha sık kabul etmesine olanak tanıyarak daha geniş bir araştırma yapılmasını sağlar ve yerel minimumlara takılma olasılığını azaltır. Eğer başlangıç sıcaklığı çok düşük olursa, algoritma erken bir aşamada alt-optimal bir çözüme sıkışabilir. Aşırı yüksek bir sıcaklık ise, algoritmanın optimal olmayan bölgeleri gereksiz yere araştırmasına neden olabilir ve hesaplama kaynaklarını israf edebilir [104].

Yapılan çalışmada 5.000 başlangıç sıcaklığı en düşük maliyetli stok kontrol parametrelerini vermiştir.

6.8.3.2 Soğutma Oranı Parametresi

Soğutma oranı etkinliğinin belirlenebilmesi için benzetimli tavlama parametrelerinden en iyi sonucu veren başlangıç sıcaklığı 5.000 sabit tutulmuş ve 500 iterasyon boyunca çalıştırılmıştır. Soğutma oranı parametresi değiştirildiğinde etkilerinin incelenebilmesi için soğutma oranı parametresi sabit tutulmamıştır. Bu çalışmada, 0,4 - 0,7 ve 0,85 olmak üzere 3 farklı soğutma oranı kullanılmıştır.

Benzetimli tavlama soğutma oranı 0,4 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.19’da verilmiştir.

Tablo 6.19 0,4’lük soğutma oranı için elde edilen sonuçlar

α: 0,4					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1800	2000	1.374.323,86	00:21:11	500
2. çözüm	1902	1995	1.350.124,44	00:23:41	500
3. çözüm	1800	1999	1.376.168,98	00:24:54	500
4. çözüm	1796	1996	1.376.740,66	00:23:51	500
5. çözüm	1905	1996	1.352.049,91	00:13:33	500
Ortalama			1.365.881,57	00:21:26	500

Benzetimli tavlama soğutma oranı 0,7 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.20’de verilmiştir.

Tablo 6.20 0,7’lik soğutma oranı için elde edilen sonuçlar

α: 0,7					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1903	1996	1.347.415,24	00:22:49	500
2. çözüm	1796	1996	1.376.740,66	00:24:25	500
3. çözüm	1797	1997	1.376.771,69	00:16:06	500
4. çözüm	1900	1993	1.350.302,42	00:16:29	500
5. çözüm	1796	1996	1.376.740,66	00:13:47	500
Ortalama			1.365.594,13	00:18:43	500

Benzetimli tavlama soğutma oranı 0,85 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.21’de verilmiştir.

Tablo 6.21 0,85’lik soğutma oranı için elde edilen sonuçlar

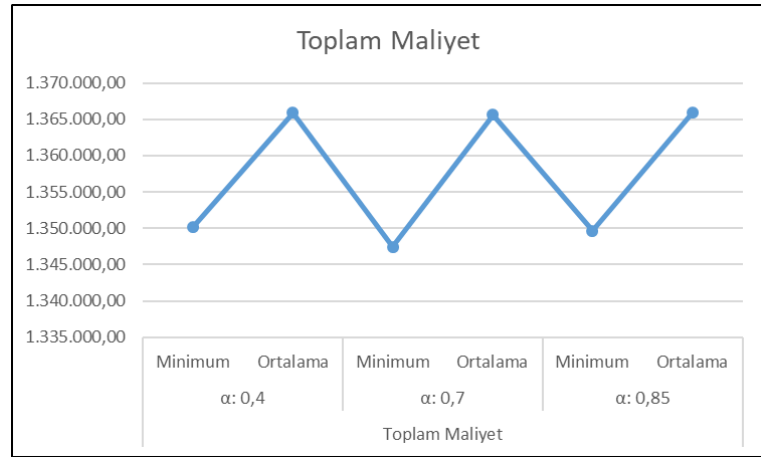
α: 0,85					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1907	1997	1.349.681,78	00:18:32	500
2. çözüm	1800	2000	1.374.323,86	00:17:47	500
3. çözüm	1794	1994	1.378.917,50	00:20:43	500
4. çözüm	1906	1998	1.349.968,08	00:23:52	500
5. çözüm	1798	1998	1.376.465,61	00:19:49	500
Ortalama			1.365.871,37	00:20:09	500

Soğutma oranı için toplam maliyet karşılaştırılması Tablo 6.22’de verilmiştir. Tabloda her soğutma oranı büyüklüğü için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.22 Soğutma oranları için maliyet karşılaştırması

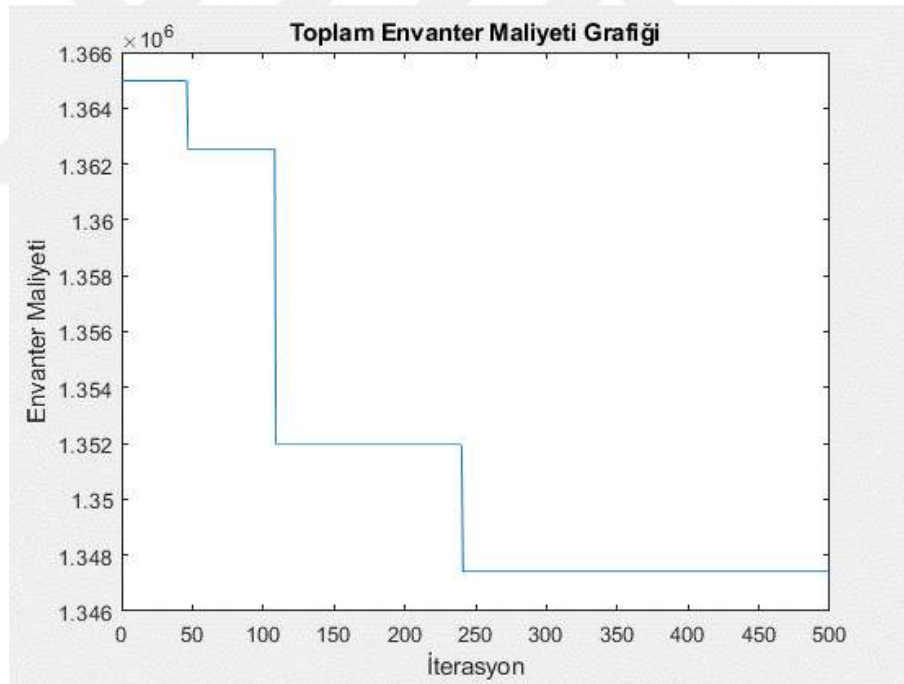
Toplam Maliyet					
α: 0,4		α: 0,7		α: 0,85	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.350.124,44	1.365.881,57	1.347.415,24	1.365.594,13	1.349.681,78	1.365.871,37

Soğutma oranlarına göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.30’da verilmiştir.



Şekil 6.30 Soğutma oranı için maliyet karşılaştırılması

Şekil 6.31’de soğutma oranlarına göre en düşük maliyeti veren (1903, 1996) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.31 (1903, 1996) parametreleri için iterasyon-maliyet grafiği

Soğutma oranı, sıcaklığın optimizasyon süreci boyunca ne kadar hızlı azaldığını kontrol eder. Yavaş bir soğutma oranı (yani her iterasyonda daha küçük bir azalma) çözüm uzayını daha kapsamlı bir şekilde araştırmaya olanak tanıyarak daha iyi bir nihai çözüme ulaşılmasını sağlar, ancak bu durum hesaplama süresini artırabilir. Hızlı bir soğutma oranı, daha hızlı bir yakınsama sağlar ancak yerel minimumlarda takılma riskini artırır. Dolayısıyla, optimal bir soğutma programı bulmak, yakınsama hızı ve çözüm kalitesi arasında denge sağlamak için kritik öneme sahiptir [104].

Yapılan çalışmada soğutma oranı 0,7 en düşük maliyetli stok kontrol parametrelerini vermiştir.

6.8.3.3 Durdurma Kriteri

İterasyon büyüklüğü etkinliğinin belirlenebilmesi için benzetimli tavlama parametrelerinden en iyi sonucu veren başlangıç sıcaklığı 5.000 ve en iyi sonucu veren soğutma oranı 0,7 sabit tutulmuştur. İterasyon büyüklüğü değiştirildiğinde etkilerinin incelenebilmesi için iterasyon büyüklüğü sabit tutulmamıştır. Bu çalışmada, 500 – 1000 - 1500 olmak üzere üç farklı iterasyon büyüklüğü kullanılmıştır.

Benzetimli tavlama iterasyon büyüklüğü 500 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.23'te verilmiştir.

Tablo 6.23 500'lük iterasyon büyüklüğü için elde edilen sonuçlar

İterasyon: 500					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1903	1996	1.347.415,24	00:22:49	500
2. çözüm	1796	1996	1.376.740,66	00:24:25	500
3. çözüm	1797	1997	1.376.771,69	00:16:06	500
4. çözüm	1900	1993	1.350.302,42	00:16:29	500
5. çözüm	1796	1996	1.376.740,66	00:13:47	500
Ortalama			1.365.594,13	00:18:43	500

Benzetimli tavlama iterasyon büyüklüğü 1.000 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.24'te verilmiştir.

Tablo 6.24 1.000’lik iterasyon büyüklüğü için elde edilen sonuçlar

İterasyon: 1.000					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1907	1997	1.349.681,78	00:37:29	1.000
2. çözüm	1801	2000	1.374.670,29	00:27:47	1.000
3. çözüm	1903	1996	1.347.415,24	00:39:57	1.000
4. çözüm	1800	2000	1.374.323,86	00:27:49	1.000
5. çözüm	1907	2000	1.346.028,91	00:28:33	1.000
Ortalama			1.358.424,02	00:32:19	1.000

Benzetimli tavlama iterasyon büyüklüğü 1.500 olduğunda bulunan maliyet, programın çalışma süresi ve iterasyon sayısı Tablo 6.25’te verilmiştir.

Tablo 6.25 1.500’lük iterasyon büyüklüğü için elde edilen sonuçlar

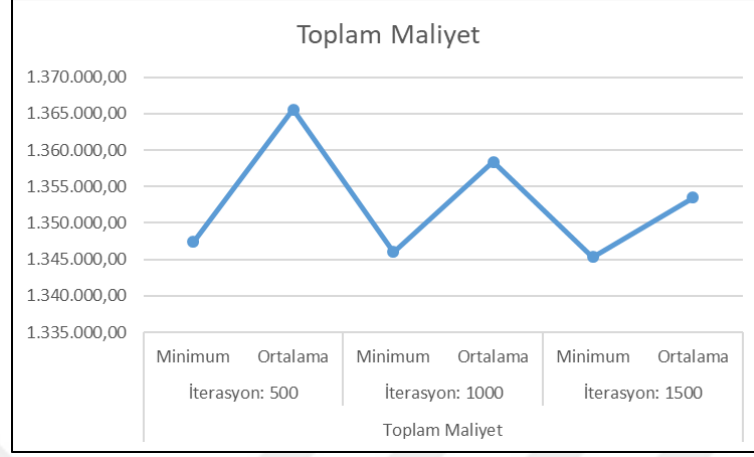
İterasyon: 1.500					
	s	S	Maliyet	Programın Çalışma Süresi	İterasyon
1. çözüm	1904	1997	1.345.356,50	01:00:30	1.500
2. çözüm	1905	1997	1.350.595,62	00:58:48	1.500
3. çözüm	1908	1998	1.349.054,24	00:57:27	1.500
4. çözüm	1798	1998	1.376.465,61	00:43:28	1.500
5. çözüm	1907	2000	1.346.028,91	00:58:32	1.500
Ortalama			1.353.500,18	00:55:45	1.500

İterasyon büyüklüğü için toplam maliyet karşılaştırılması Tablo 6.26’da verilmiştir. Tabloda her iterasyon büyüklüğü için 5 çözüm arasındaki en küçük değer ve 5 çözümün ortalaması verilmiştir.

Tablo 6.26 İterasyon büyüklükleri için maliyet karşılaştırması

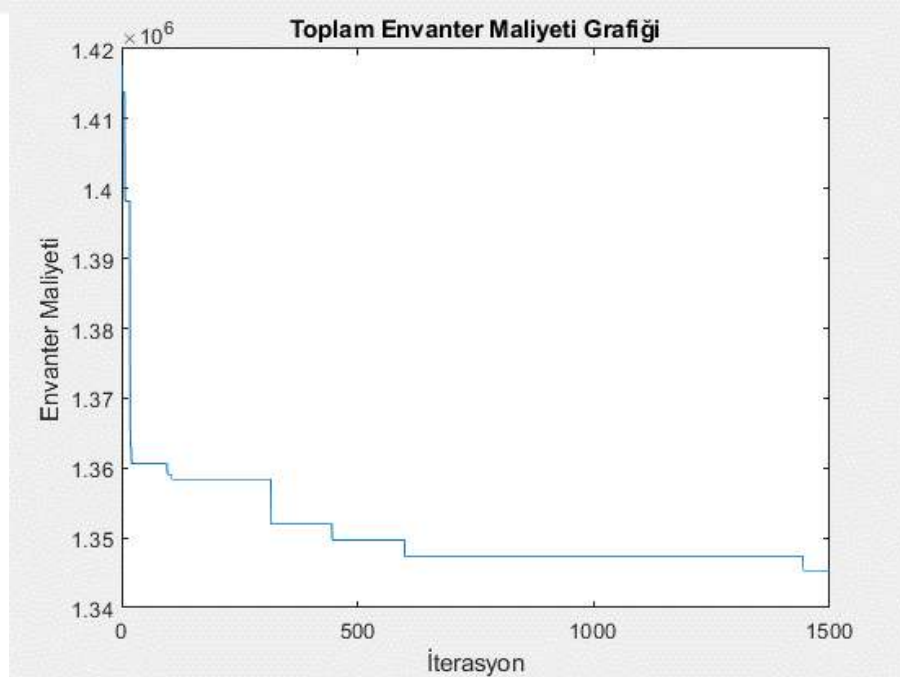
Toplam Maliyet					
İterasyon: 500		İterasyon: 1000		İterasyon: 1500	
Minimum	Ortalama	Minimum	Ortalama	Minimum	Ortalama
1.347.415,24	1.365.594,13	1.346.028,91	1.358.424,02	1.345.356,50	1.353.500,18

İterasyon büyüklüğüne göre toplam maliyetlerin karşılaştırma grafiği Şekil 6.32’de verilmiştir.



Şekil 6.32 Soğutma oranları için maliyet karşılaştırılması

Şekil 6.33’te iterasyon büyüklüğüne göre en düşük maliyeti veren (1904, 1997) parametreleri için hesaplanan maliyet grafiği verilmiştir.



Şekil 6.33 (1904, 1997) parametreleri için iterasyon-maliyet grafiği

İterasyon sayısı, her sıcaklık seviyesinde gerçekleştirilen adım sayısını belirler. Daha fazla iterasyon, algoritmanın her sıcaklık seviyesinde çözüm uzayını daha kapsamlı bir şekilde araştırmasına olanak tanıyarak global optimuma ulaşma şansını artırır. Ancak iterasyon sayısının artırılması, hesaplama yükünü de artırır. Bu nedenle, problemi ve mevcut hesaplama kaynaklarını dikkate alarak uygun bir denge sağlanmalıdır [104].

Yapılan çalışmada iterasyon sayısı 1.500 olduğunda en düşük maliyetli stok kontrol politikasını vermiştir.

6.9 Genetik Algoritma, Benzetimli Tavlama Algoritması ve OptQuest ile Bulunan Sonuçların ve Mevcut Durumun Karşılaştırılması

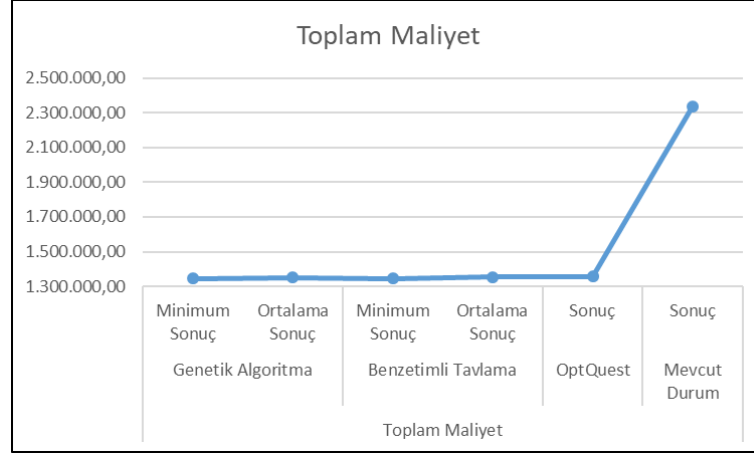
Optimizasyon işlemlerinin ardından 3 farklı çözüm yöntemi için optimum stok maliyetleri ve mevcut durumdaki stok maliyeti Tablo 6.27’de verilmiştir.

Tablo 6.27 Optimum çözümler ile bulunan ve mevcut durumdaki stok maliyeti

Toplam Maliyet					
Genetik Algoritma		Benzetimli Tavlama		OptQuest	Mevcut Durum
Minimum Sonuç	Ortalama Sonuç	Minimum Sonuç	Ortalama Sonuç	Sonuç	
1.344.728,96	1.351.156,31	1.345.356,50	1.353.500,18	1.356.161,83	2.335.222,32

OptQuest aracı kullanılarak önerilen çözüm ile toplam maliyetin yaklaşık %41,92 iyileştirilebileceği, benzetimli tavlama algoritması kullanılarak önerilen çözüm ile toplam maliyetin yaklaşık %42,03 iyileştirilebileceği, en iyi iyileştirmenin genetik algoritma kullanılarak yaklaşık %42,14 olacağı belirlenmiştir.

Optimum çözümlerin mevcut durumla karşılaştırma grafiği Şekil 6.34’te verilmiştir.



Şekil 6.34 Optimum çözümlerin mevcut durumla karşılaştırma grafiği

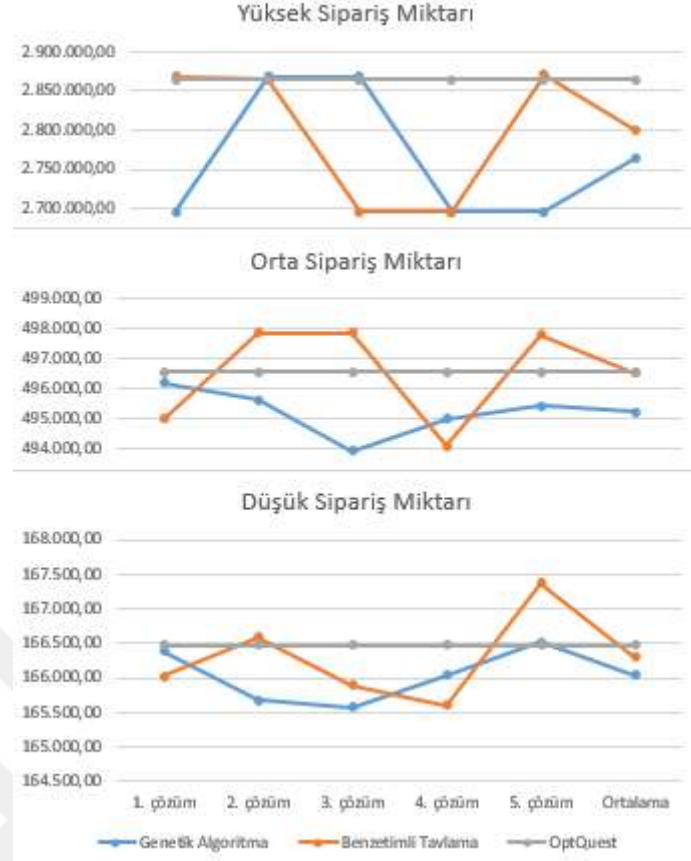
Genetik Algoritmalar, küresel arama kapasitesi ve yerel optimumlardan kaçınma yetenekleriyle bilinir. Bu algoritmalar, büyük ve karmaşık arama alanlarını etkili bir şekilde keşfetme yeteneğine sahiptirler ve çeşitli problem türlerine uygulanabilirler. Ancak, hesaplama maliyetlerinin yüksek olması ve uzun yakınsama süreleri gibi dezavantajları vardır. Yapılan araştırmalar, GA'ların optimal çözümleri bulmada genellikle diğer yöntemlerden daha başarılı olduğunu göstermektedir. Bu avantajlar, GA'ları karmaşık optimizasyon problemleri için çekici bir duruma getirir. Benzetimli Tavlama, basitliği ve yerel arama yetenekleri ile öne çıkar. Bu algoritma, başlangıçta daha kötü çözümleri kabul ederek yerel optimumlardan kaçınır ve zamanla kabul olasılığını azaltır. Ancak, yerel optimumlarda takılma olasılığı ve performansının soğuma programı ve başlangıç sıcaklığına duyarlılığı gibi dezavantajları vardır. Benzetimli tavlama, hesaplama açısından daha az maliyetli olup, iyi bir yerel aramanın önemli olduğu problemlerde etkili performans gösterir [105]. OptQuest, birden fazla optimizasyon algoritmasını (sinir ağları, tabu arama vb.) birleştiren hibrit bir optimizasyon aracıdır. Bu optimizasyon aracı, küresel ve yerel arama arasında denge kurarak en uygun çözümleri bulmak için çeşitli algoritmaların güçlü yönlerinden yararlanır [18]. OptQuest'in adaptif yapısı ve hibrit yaklaşımı, optimizasyon problemlerinde yüksek verimlilik ve etkililik sağlar [110].

6.10 Faktör Analizi

Bu bölümde, sipariş miktarı, tedarik süresi ve siparişlerin gelişler arası süresi faktörlerinin geliştirilen metodolojinin performansı üzerindeki etkileri incelenmiştir. Genetik algoritma ile çözülürken en iyi sonucu veren başlangıç popülasyonu büyüklüğü 120, çaprazlama olasılığı 0,85 ve mutasyon olasılığı 0,06 kullanılmıştır. Benzetimli tavlama algoritması ile çözülürken en iyi değeri veren başlangıç sıcaklığı 5.000, soğutma oranı 0,7, iterasyon büyüklüğü 1.500 kullanılmıştır. Hizmet düzeyinin %80'in üstünde olması kısıtı altında olurlu çözümü olmayan faktör büyüklükleri için hizmet düzeyi kısıtı kullanılmamıştır. Genetik algoritma ve benzetimli tavlama algoritması rassallığından kaynaklı değişkenliği önlemek amacıyla, her iki algoritma da incelenen her faktör büyüklüğü için beş kere çalıştırılmış ve beş çözümün ortalaması alınmıştır. OptQuest aracı ile bulunan çözümler grafikte lineer olarak gösterilmiştir.

6.10.1 Sipariş Miktarı

Bu bölümde incelenen stok kontrol modeline sipariş miktarı büyüklüğünün etkisi incelenmiştir. Mevcut durum sipariş miktarı $-0.001 + 552 * \text{BETA}(0.732, 0.746)$ dağılımına uygun olarak hesaplanmaktadır. Bu dağılımda normalleştirme faktörü değiştirildiğinde düşük, orta ve yüksek sipariş miktarları sırasıyla $-0.001 + 100 * \text{BETA}(0.732, 0.746)$, $-0.001 + 300 * \text{BETA}(0.732, 0.746)$ ve $-0.001 + 750 * \text{BETA}(0.732, 0.746)$ olarak belirlenmiştir. Bulunan çözümler ve ortalamalar Şekil 6.35'te verilmiştir.



Şekil 6.35 Düşük, orta ve yüksek sipariş miktarı büyüklükleri için yöntemlerin karşılaştırılması

Tablo 6.28’de genetik algoritma, benzetimli tavlama ve OptQuest yöntemlerinin düşük, orta ve yüksek sipariş miktarları için toplam maliyetlerini detaylı bir şekilde verilmiştir. Tablo 6.28 değerlendirilerek, her bir yöntem için maliyetlerin nasıl değiştiğini ve hangi yöntemlerin en uygun maliyetleri sunduğu analiz edilecektir.

Tablo 6.28 Sipariş miktarı büyüklüğü, yöntem ve maliyet tablosu

Sipariş Miktarı	Yöntem	Minimum Maliyet	Ortalama Maliyet
Düşük	Genetik Algoritma	165.579,19	166.043,02
Düşük	Benzetimli Tavlama	165.610,85	166.302,73
Düşük	OptQuest	166.480,98	-
Orta	Genetik Algoritma	493.945,01	495.255,78
Orta	Benzetimli Tavlama	494.100,10	496.531,91
Orta	OptQuest	496.578,55	-
Yüksek	Genetik Algoritma	2.695.898,96	2.765.283,32
Yüksek	Benzetimli Tavlama	2.696.197,16	2.799.325,11
Yüksek	OptQuest	2.864.729,13	-

Genetik algoritma, düşük, orta ve yüksek sipariş miktarlarında en düşük maliyetleri sunan yöntem olarak öne çıkmaktadır. Bu, genetik algoritmanın tüm sipariş büyüklüklerinde etkili ve maliyet açısından verimli olduğunu göstermektedir. Hem minimum hem de ortalama maliyetlerde genetik algoritmanın performansı dikkat çekicidir.

Benzetimli tavlama, genetik algoritmaya kıyasla genellikle daha yüksek maliyetlerle sonuçlanmaktadır. Ortalama maliyetlere bakıldığında düşük ve orta siparişlerde yakın bir performans göstermekle birlikte, yüksek siparişlerde genetik algoritmaya göre daha maliyetli olmaktadır.

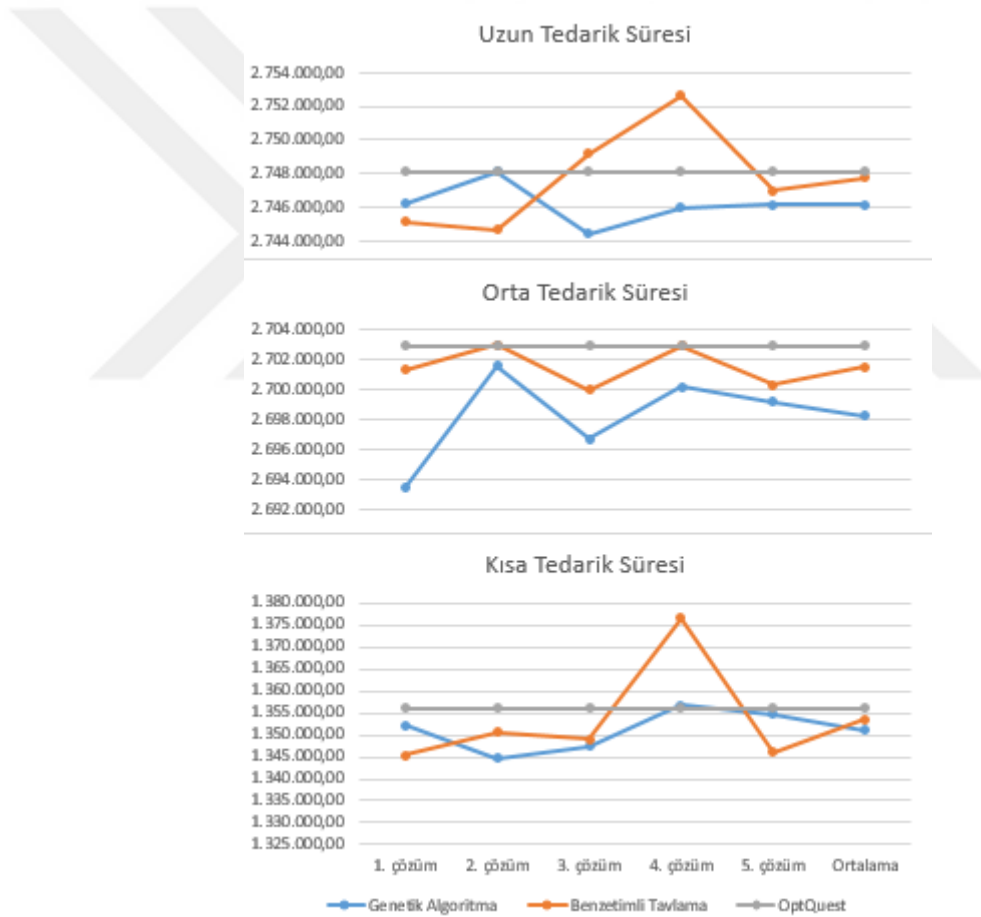
OptQuest, hem küçük hem de orta ve yüksek sipariş miktarlarında en yüksek maliyetleri sunmaktadır. Bu, OptQuest'in diğer yöntemlere kıyasla daha maliyetli olduğunu ve genellikle daha yüksek maliyetlerle sonuçlandığını göstermektedir.

Düşük sipariş miktarları, daha sık sipariş verilmesine neden olabilir, bu da işlem maliyetlerini artırır. Ayrıca, düşük sipariş miktarları, stok seviyesinin sık sık kritik düzeylere düşmesine yol açabilir, bu da tedarik zincirinde aksamalara neden olabilir. Düşük sipariş miktarlarının operasyonel etkileri genellikle artan sipariş maliyetleri ve stok tükenmesi riskinin artmasıdır [45]. Yüksek sipariş miktarları, daha seyrek sipariş gerektirir, bu da işlem maliyetlerini azaltabilir. Ancak, yüksek sipariş miktarları, fazla stok bulundurma maliyetlerine yol açabilir ve nakit akışını olumsuz

etkileyebilir. Aynı zamanda, yüksek miktarda sipariş vermek, ürünlerin bozulma riskini artırabilir ve depolama maliyetlerini yükseltebilir [41].

6.10.2 Tedarik Süresi

Bu bölümde stok kontrol modeline tedarik süresi uzunluğunun etkisi incelenmiştir. Mevcut durum tedarik süresi UNIF (2,8) dağılımına uygun olarak hesaplanmaktadır. Kısa, orta ve uzun tedarik süreleri sırasıyla UNIF (2,8), UNIF (8,14) ve UNIF (14,20) olarak belirlenmiştir. Bulunan çözümler ve ortalamalar Şekil 6.36’da verilmiştir.



Şekil 6.36 Uzun, orta, kısa tedarik süresi büyüklükleri için yöntemlerin karşılaştırılması

Tablo 6.29’da, genetik algoritma, benzetimli tavlama ve OptQuest yöntemlerinin kısa, orta ve uzun tedarik süreleri için maliyetleri detaylı bir şekilde

incelemektedir. Tablo 6.29 değerlendirilerek her bir yöntem için maliyetlerin nasıl değiştiğini ve hangi yöntemlerin en uygun maliyetleri sunduğu analiz edilecektir.

Tablo 6.29 Tedarik süresi büyüklüğü, yöntem ve maliyet tablosu

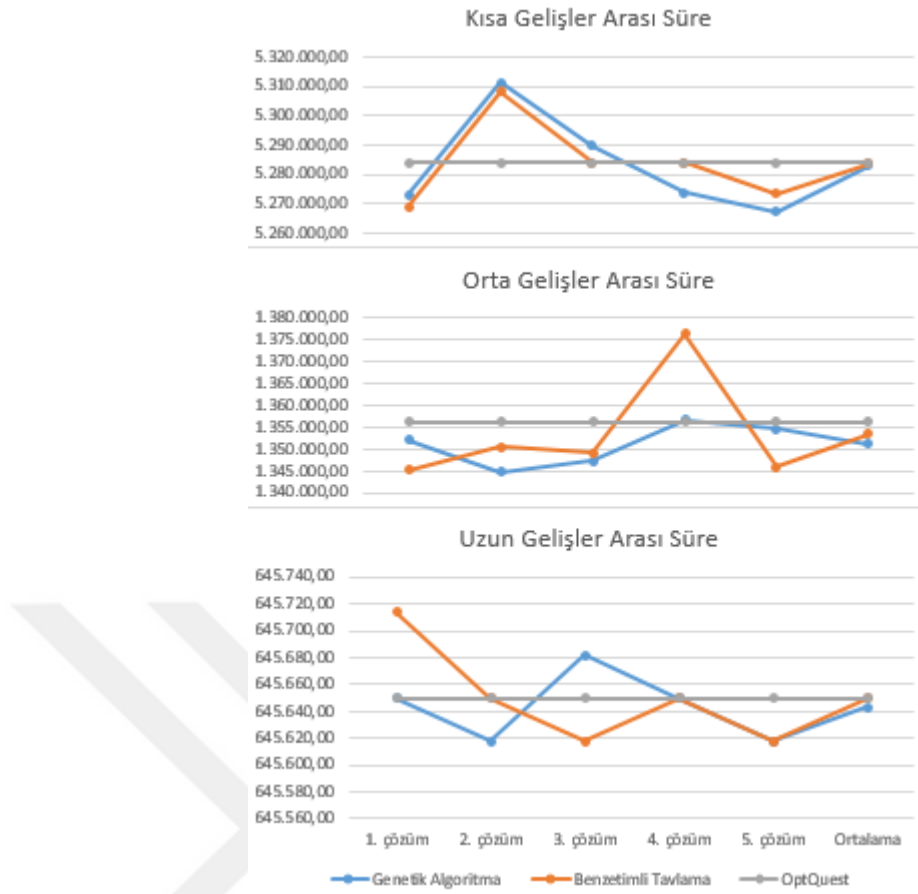
Tedarik Süresi	Yöntem	Minimum Maliyet	Ortalama Maliyet
Kısa	Genetik Algoritma	1.344.728,96	1.351.156,31
Kısa	Benzetimli Tavlama	1.345.356,50	1.353.500,18
Kısa	OptQuest	1.356.161,83	-
Orta	Genetik Algoritma	2.693.513,58	2.698.270,60
Orta	Benzetimli Tavlama	2.700.014,85	2.701.534,46
Orta	OptQuest	2.702.926,99	-
Uzun	Genetik Algoritma	2.744.437,91	2.746.186,76
Uzun	Benzetimli Tavlama	2.744.694,36	2.747.743,71
Uzun	OptQuest	2.748.136,15	-

Genetik Algoritma genel olarak en düşük maliyetleri bulduğu ve bu nedenle maliyet etkinliği en yüksek yöntem olarak öne çıktığı görülmektedir. Kısa tedarik sürelerinde genetik algoritma hem minimum hem de ortalama maliyet açısından en avantajlı sonuçları sunarken, benzetimli tavlama yöntemi minimum maliyet açısından genetik algoritmaya çok yakın sonuçlar verse de, ortalama maliyette biraz daha yüksek maliyetler bulmaktadır. OptQuest yöntemi ise en yüksek optimum maliyeti sunmaktadır. Orta tedarik sürelerinde, genetik algoritma, benzetimli tavlama ve OptQuest yönteminden daha düşük ortalama maliyetler bulması nedeniyle avantajlı olmaktadır. Uzun tedarik sürelerinde, Genetik algoritma en düşük minimum ve ortalama maliyetleri sağlamakta, Benzetimli Tavlama ise bu sonuçlara çok yakın ama daha yüksek maliyetler vermektedir. OptQuest en yüksek optimum maliyeti sunmaktadır. Tüm bu bulgular ışığında, maliyet etkinliği ve genel performans açısından Genetik Algoritma yöntemi, tedarik süresi büyüklüğü ne olursa olsun en uygun yöntem olarak öne çıkarken, Benzetimli Tavlama da makul bir alternatif olabilmektedir ancak maliyetlerin biraz daha yüksek olabileceği dikkate alınmalıdır. OptQuest yöntemi ise diğer yöntemlere göre daha yüksek maliyetler sunduğundan, maliyet etkinliği açısından daha az tercih edilebilir bir yöntem olarak değerlendirilmektedir.

Kısa tedarik süreleri, stok bulundurma maliyetlerini azaltır ve hızlı yanıt sağlar. Bu durum, güvenlik stoklarına olan ihtiyacı azaltır ve tedarik zincirinde daha fazla esneklik sağlar. Aynı zamanda müşteri memnuniyetini artırabilir. Kısa tedarik süreleri, genellikle müşteri hizmet seviyelerini artırmak için önemlidir [109]. Uzun tedarik süreleri, daha fazla güvenlik stoğu gerektirir, bu da stok bulundurma maliyetlerini artırır. Uzun tedarik süreleri, siparişlerin zamanında karşılanmaması riskini artırır ve tedarik zincirinde aksamalar yaratabilir. Bu durum, müşteri memnuniyetsizliğine ve satış kaybına yol açabilir. Uzun tedarik süreleri, tedarik zinciri planlamasında büyük zorluklar yaratır [107].

6.10.3 Siparişlerin Gelişler Arası Süresi

Bu bölümde stok kontrol modeline siparişlerin gelişler arası süresi büyüklüğünün etkisi incelenmiştir. Mevcut durumda sabit olarak her gün 1 sipariş gelmektedir. Siparişlerin kısa, orta ve uzun gelişler arası süresi büyüklükleri sırasıyla 0,5 günde 1 sipariş gelmesi, her gün 1 sipariş gelmesi ve 1,5 günde 1 sipariş gelmesi olarak belirlenmiştir. Bulunan çözümler ve ortalamalar Şekil 6.37’de verilmiştir.



Şekil 6.37 Siparişlerin kısa, orta ve uzun gelişler arası süresi büyüklükleri için yöntemlerin karşılaştırılması

Tablo 6.30’da, genetik algoritma, benzetimli tavlama ve OptQuest yöntemleri ile siparişlerin kısa, orta ve uzun gelişler arası süresi seviyeleri için maliyetler detaylı bir şekilde incelemektedir. Tablo 6.30 değerlendirilerek, her bir yöntem için maliyetlerin nasıl değiştiğini ve hangi yöntemlerin en uygun maliyetleri sunduğu analiz edilecektir.

Tablo 6.30 Siparişlerin gelişler arası süresi büyüklüğü, yöntem ve maliyet tablosu

Siparişlerin Gelişler Arası Süresi	Yöntem	Minimum Maliyet	Ortalama Maliyet
Kısa	Genetik Algoritma	5.267.336,08	5.283.100,22
Kısa	Benzetimli Tavlama	5.269.002,73	5.283.767,41
Kısa	OptQuest	5.284.149,36	-
Orta	Genetik Algoritma	1.344.728,96	1.351.156,31
Orta	Benzetimli Tavlama	1.345.356,50	1.353.500,18
Orta	OptQuest	1.356.161,83	-
Uzun	Genetik Algoritma	645.617,70	645.643,28
Uzun	Benzetimli Tavlama	645.617,70	645.649,67
Uzun	OptQuest	645.649,67	-

Genetik Algoritma, özellikle düşük maliyetler elde etmede daha başarılıdır. Ancak, her üç yöntemin de farklı büyüklükteki siparişlerin gelişler arası sürelerinde yakın performanslar sergilediği görülmektedir.

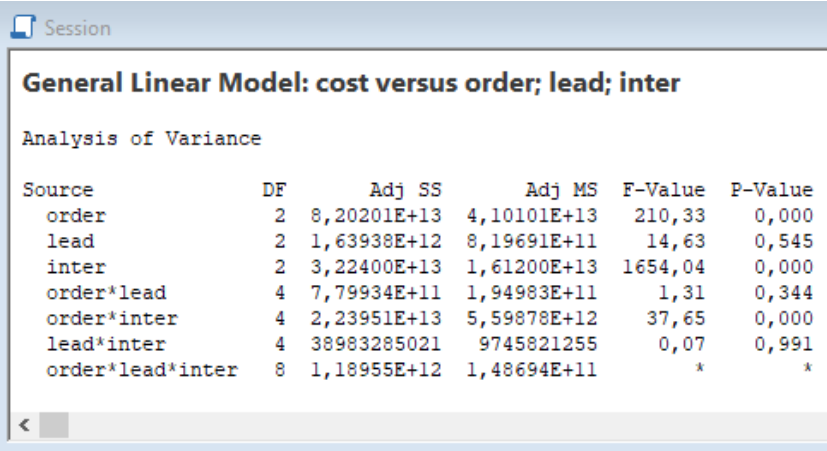
Genetik algoritma, kısa, orta ve uzun siparişlerin gelişler arası büyüklükleri için en düşük maliyetleri sunan yöntem olarak öne çıkmaktadır. Bu, genetik algoritmanın tüm siparişlerin gelişler arası süre büyüklüklerinde etkili ve maliyet açısından verimli olduğunu göstermektedir. Hem minimum hem de ortalama maliyetlerde genetik algoritma en iyi sonuçları vermiştir.

Benzetimli tavlama, genetik algoritmaya kıyasla genellikle daha yüksek maliyetlerle sonuçlanmaktadır. Sipariş gelişler arası büyüklüğü uzun olduğunda, ortalama maliyete bakıldığında OptQuest ile aynı sonucu vermesine rağmen minimum maliyetlere bakıldığında benzetimli tavlama daha avantajlı çözümler bulmuştur.

OptQuest, hem küçük hem de orta ve yüksek sipariş miktarlarında en yüksek maliyetleri sunmaktadır. Bu, OptQuest'in diğer yöntemlere kıyasla daha maliyetli olduğunu ve genellikle daha yüksek maliyetlerle sonuçlandığını göstermektedir.

6.11 Sipariş Miktarı, Tedarik süresi ve Siparişlerin Gelişler Arası Süresi Etki Analizi

Bu bölümde bütün sonuçlarda en iyi değeri veren genetik algoritma kullanılarak sipariş miktarı, tedarik süresi ve siparişlerin gelişler arası süresi etkilerinin incelenmesi için Minitab programında Anova analizi yapılmıştır. Bu bölümde genetik algoritma çalıştırılırken tüm çözümlerin olurlu olması için ve faktörlerin stok kontrol sistemine etkilerini incelemek için hizmet düzeyi kısıtı kullanılmamıştır. Şekil 6.38’de Anova analizi sonuçları yer almaktadır.



Session

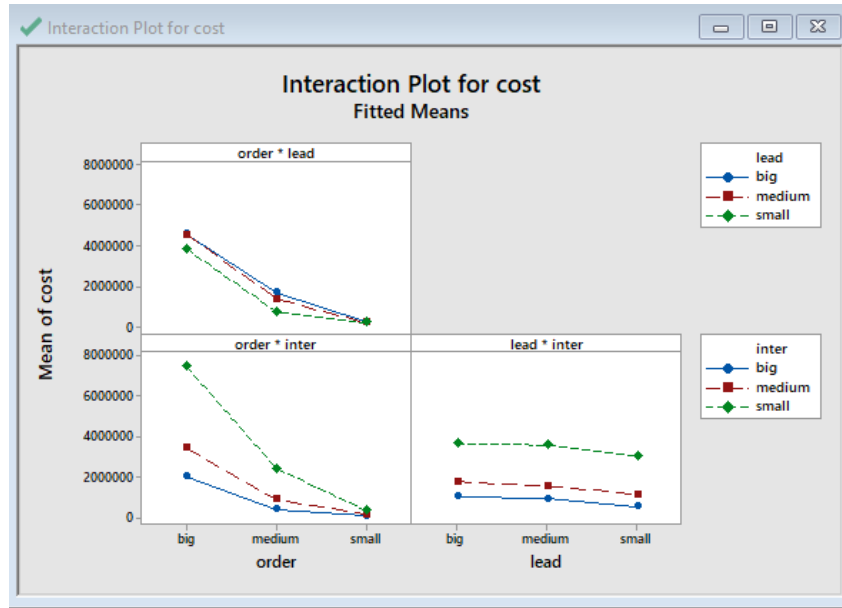
General Linear Model: cost versus order; lead; inter

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
order	2	8,20201E+13	4,10101E+13	210,33	0,000
lead	2	1,63938E+12	8,19691E+11	14,63	0,545
inter	2	3,22400E+13	1,61200E+13	1654,04	0,000
order*lead	4	7,79934E+11	1,94983E+11	1,31	0,344
order*inter	4	2,23951E+13	5,59878E+12	37,65	0,000
lead*inter	4	38983285021	9745821255	0,07	0,991
order*lead*inter	8	1,18955E+12	1,48694E+11	*	*

Şekil 6.38 Anova çıktısı

Şekil 6.39’da maliyet etkileşim grafiği verilmiştir.



Şekil 6.39 Maliyet etkileşim grafiği

Sipariş miktarı bağımlı değişken üzerinde istatistiksel olarak anlamlı bir etkiye sahiptir. P değeri 0,05'ten çok küçüktür, bu da sipariş miktarının bağımlı değişken üzerinde çok güçlü bir etkisi olduğunu gösterir. Yani, sipariş miktarındaki değişiklikler, maliyet üzerinde önemli bir fark yaratmaktadır. Tedarik süresi bağımlı değişken üzerinde anlamlı bir etkiye sahip değildir. P değeri 0,05'ten büyüktür, bu da tedarik süresinin bağımlı değişken üzerinde istatistiksel olarak anlamlı bir etkisi olmadığını gösterir. Yani, tedarik süresi değişiklikleri, maliyette önemli bir değişikliğe neden olmamaktadır. Siparişlerin gelişler arası süresi bağımlı değişken üzerinde istatistiksel olarak çok anlamlı bir etkiye sahiptir. P değeri 0,05'ten çok küçüktür, bu da siparişlerin gelişler arası süresinin bağımlı değişken üzerinde güçlü bir etkisi olduğunu gösterir. Siparişlerin gelişler arası süresindeki değişiklikler, maliyet üzerinde önemli bir fark yaratmaktadır.

Sipariş miktarı ve tedarik süresi etkileşimi bağımlı değişken üzerinde anlamlı bir etkiye sahiptir. P değeri 0,05'ten küçüktür, bu da bu iki değişkenin etkileşiminin bağımlı değişken üzerinde istatistiksel olarak anlamlı bir etkisi olduğunu gösterir. Yani, sipariş miktarı ve tedarik süresi birlikte düşünüldüğünde maliyet üzerinde anlamlı bir etkisi vardır. Sipariş miktarı ve siparişlerin gelişler arası süresi etkileşimi bağımlı değişken üzerinde çok anlamlı bir etkiye sahiptir. P değeri 0,05'ten çok

küçüktür, bu da bu iki değişkenin etkileşiminin bağımlı değişken üzerinde güçlü bir etkisi olduğunu gösterir. Sipariş miktarı ve siparişlerin gelişler arası süresi birlikte değerlendirildiğinde maliyet üzerinde önemli bir fark yaratmaktadır. Tedarik süresi ve siparişlerin gelişler arası süresi etkileşimi bağımlı değişken üzerinde anlamlı bir etkiye sahip değildir. P değeri 0,05'ten büyüktür, bu da bu iki değişkenin etkileşiminin bağımlı değişken üzerinde istatistiksel olarak anlamlı bir etkisi olmadığını gösterir. Yani, tedarik süresi ve siparişlerin gelişler arası süresi birlikte ele alındığında maliyet üzerinde anlamlı bir etkisi yoktur. Bu üçlü etkileşimin P değeri verilmediğinde, bu etkileşimin istatistiksel olarak anlamlı olmadığı varsayılabilir.

Bu sonuçlar, maliyetleri düşürmek veya optimize etmek amacıyla hangi değişkenlerin ve etkileşimlerin dikkate alınması gerektiği konusunda yol gösterici olabilir. Özellikle sipariş miktarı ve siparişlerin gelişler arası süresi değişkenlerine odaklanmak, maliyet yönetiminde önemli bir fark yaratabilir.

7. SONUÇLAR VE ÖNERİLER

Stoklar firmaların karlılıklarını etkileyen önemli faktörlerden biridir. Stokların yönetilmesi firmaların günümüz koşullarında ayakta kalabilmeleri için kritik konuların başında yer almaktadır. Firmalar stokların etkin yönetimiyle karlılıklarını arttırabilir, müşteri memnuniyetini sağlayabilir ve şirketler arasında yapılan sözleşmeler gereği, karşılanamayan müşteri talepleri nedeniyle tazminat ödemelerini engelleyebilir.

Her firmanın kendi satış politikaları, üretim biçimi, üretim kapasitesi ve tedarikçi firmanın uzaklığı ve üretim yapısı gibi farklı durumlara sahip olduğu için her firmanın stok yönetim politikası farklılık gösterir. Önemli olan firmanın bu koşulları değerlendirip kendi ihtiyaçlarına göre bir stok kontrol politikası geliştirebilmesidir. Firmaların belirledikleri stok kontrol yönteminden sonra stok maliyetini minimize eden ve yüksek müşteri hizmet düzeyini temel alan uygun stok kontrol parametrelerini belirlemesi gerekmektedir.

Simülasyon uygulamaları sayesinde sistemin değişen çevre şartlarına ve farklı değişkenlere bağlı olarak farklı koşullar altında ne tür sonuçlar verebileceği incelenebilmektedir. Firmanın uyguladığı ya da uygulayacağı stok kontrol politikası simülasyon modeliyle gerçek veriler kullanarak modellenabilir ve farklı koşullar altında analiz edilebilir. Simülasyon sayesinde bu işlemlerin yapılması firmalara ciddi bir süre ve maliyet kazancı sağlamaktadır. Bununla birlikte simülasyon tek başına bir çözüm tekniği değildir. Bu nedenle stok kontrol parametrelerinin belirlenmesinde simülasyon ve optimizasyonun entegre edilmesi gerekmektedir.

Bu çalışmada üretim yapan bir firmanın stok kontrol maliyetini minimize eden optimal stok kontrol politikası parametrelerinin bulunabilmesi için benzetimli tavlama ve genetik algoritma tabanlı bir simülasyon optimizasyonu yaklaşımı sunulmuştur. Firmanın uyguladığı stok kontrol politikasının bir simülasyon modeli geliştirilmiştir ve stok kontrol modelinin parametrelerini optimize etmek için genetik algoritma ve benzetimli tavlama algoritması entegre edilmiştir. Genetik algoritma ve benzetimli tavlama algoritmalarının en iyi parametrelerini belirlemek için parametre analizleri yapılmıştır. Ayrıca problem OptQuest aracıyla da çözülmüştür ve genetik algoritma, benzetimli tavlama algoritması ve OptQuest ile elde edilen sonuçlar karşılaştırılmıştır.

Modeldeki faktörlerin (sipariş miktarı, tedarik zamanı ve siparişlerin gelişler arası süresi) farklı seviyelerinin büyüklükleri belirlenerek, toplam stok maliyeti üzerindeki etkileri incelenebilmesi için deney tasarımı gerçekleştirilmiştir.

Tüm sonuçlarda en iyi sonucu veren genetik algoritma kullanılarak önerilen çözüm ile toplam maliyetin yaklaşık %42,14 oranında azaltılabileceği bulunmuştur. Siparişlerin gelişler arası süresi ve sipariş miktarı değişkenlerine odaklanıldığında maliyet yönetiminde önemli bir fark yaratılabilir.

Bu çalışmanın başka bir uzantısı olarak simülasyon tabanlı optimizasyon işleminin bir karar destek sistemi ile entegre edilmesi olabilir. Böylece programda kullanılan verilerin otomatik hale gelmesi sağlanabilir. Bu karar destek sistemi ile simülasyonun endüstride uygulaması basitleşecek ve güncel verilerle çalışma imkanı sağlanabilecektir.

Ayrıca optimum stok parametreleri, belirlenen uygunluk fonksiyonunun değerine bağlı olarak değiştirilerek dinamik bir stok sistemi oluşturulabilir. Böylece sipariş miktarının yoğun olduğu dönemler ve sipariş miktarının düştüğü dönemlerde farklı optimum noktalarla sipariş verilerek, stok maliyetini daha fazla düşürebilme potansiyeli sağlanabilir.

Önerilen yaklaşım optimizasyon konusunda iyi olmasına rağmen uzun bir süreç boyunca çalışmaktadır. Bunun temel nedeni, GA ve SA amaç fonksiyonunun her değerlendirmesinin, önceden tanımlanmış sayıda simülasyon modeli çalışmasının yürütülmesini gerektirmesidir. Önerilen algoritmanın verimliliğini artırmaya yönelik diğer bir yaklaşım, paralel bir algoritma tasarlamak ve temel algoritmanın görevini farklı işlemcilerle dağıtmak olabilir.

KAYNAKÇA

- [1] Ilgın, M. A. (2006). Joint optimization of spare parts inventory and maintenance policies using hybrid genetic algorithms. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı.Yüksek Lisans Tezi.
- [2] Nair, S. S., & Jose, K. P. (2019). Solution To Production Inventory Systems With Orbit, Buffer And Different Service Rates. Palestine Journal of Mathematics, 8(2).
- [3] Zhang, C., Gao, W., Yang, T., & Guo, S. (2019). Opportunistic maintenance strategy for wind turbines considering weather conditions and spare parts inventory management. Renewable Energy, 133, 703-711.
- [4] Rabta, B. (2019). Stationary Distribution and Perturbation Bounds for a Stochastic Inventory Model. Journal of Mathematical Sciences, 237, 722-729.
- [5] Ko, S. S. (2020). A nonhomogeneous quasi-birth-death process approach for an (s,S) policy for a perishable inventory system with retrial demands. Journal of Industrial & Management Optimization, 16(3).
- [6] Selvakumar, C., Maheswari, P., & Elango, C. (2020). Discrete markov decision process-inventory control in a service facility system. Revista Investigación Operacional, 41(6), 872-881.
- [7] Wei, B., & Yao, D. (2021). Ergodic inventory control with diffusion demand and general ordering costs. Operations Research Letters, 49(4), 578-585.
- [8] Guijarro, E., Babiloni, E., & Cardós, M. (2022). On the estimation of the fill rate for the continuous (s,S) inventory system for the lost sales context. Plos one, 17(2), e0263655.
- [9] Shi, J., Rozas, H., Yildirim, M., & Gebraeel, N. (2023). A stochastic programming model for jointly optimizing maintenance and spare parts inventory for IoT applications. IIE Transactions, 55(4), 419-431.
- [10] Barron, Y. (2022). A replenishment inventory model with a stock-dependent demand and age-stock-dependent cost functions in a random environment. Asia-Pacific Journal of Operational Research, 39(03), 2150035.
- [11] Ozkar, S. (2023). Two-commodity queueing-inventory system with phase-type distribution of service times. Annals of Operations Research, 331(2), 711-737.
- [12] Dissa, S., & Ushakumari, P. V. (2023). Two commodity perishable queueing inventory system with random common lifetime of commodities and positive lead time. Opsearch, 1-26.
- [13] Federgruen, A., & Zipkin, P. (1984). An efficient algorithm for computing optimal (s,S) policies. Operations research, 32(6), 1268-1285.
- [14] Zheng, Y. S., & Federgruen, A. (1991). Finding optimal (s,S) policies is about as simple as evaluating a single policy. Operations research, 39(4), 654-665.
- [15] Chang-hua, J. I. A. N. G., & You-hua, H. U. (2006). Simulation and Optimization of Stochastic (s,S) Inventory System Based on Genetic Algorithm (Chinese). Journal of East China Normal University (Natural Science), 2006(3), 71.

- [16] Elta wil, A. B., & Elnagar, G. R. (2007). Simulation optimization of an (s,S) inventory control system with random demand sizes, demand arrivals, and lead times. In 37th International Conference on Computers and Industrial Engineering (pp. 2420-2432).
- [17] Biles, W. E., Kleijnen, J. P., Van Beers, W. C., & Van Nieuwenhuyse, I. (2007, December). Kriging metamodeling in constrained simulation optimization: an explorative study. In 2007 Winter Simulation Conference (pp. 355-362). IEEE.
- [18] Kleijnen, J. P., & Wan, J. (2007). Optimization of simulated systems: OptQuest and alternatives. *Simulation Modelling Practice and Theory*, 15(3), 354-362.
- [19] Li, L., & Wan, J. (2008, December). Simulation for constrained optimization of inventory system by using arena and OptQuest. In 2008 International Conference on Computer Science and Software Engineering (Vol. 2, pp. 202-205). IEEE.
- [20] Kurtulus, I. S. (2012). A Test of Wagner's Heuristics for the Spare Parts Inventory Control Problem. *International Journal of Operations Research and Information Systems (IJORIS)*, 3(4), 88-100.
- [21] Attanayake, N., Kashef, R. F., & Andrea, T. (2014, May). A simulation model for a continuous review inventory policy for healthcare systems. In 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-6). IEEE.
- [22] Shahi, S., & Pulkki, R. (2015). A simulation-based optimization approach to integrated inventory management of a sawlog supply chain with demand uncertainty. *Canadian Journal of Forest Research*, 45(10), 1313-1326.
- [23] Al-Fandi, L. M., Obaid, A. A. B., Alfai lakawi, B. I., Alsubaiei, H. A., & Khudhair, S. A. (2019). A simulation study to determine the parameters of medicine inventory policy. *Proceedings of the Estonian Academy of Sciences*, 68(4).
- [24] Deng, H., Shi, Q., & Wang, Y. (2021). A Joint Optimization Model of (s,S) Inventory and Supply Strategy Using an Improved PSO-Based Algorithm. *Wireless Communications and Mobile Computing*, 2021(1), 7621692.
- [25] Yuna, F., Er kayman, B., & Yılmaz, M. (2023). Inventory control model for intermittent demand: a comparison of metaheuristics. *Soft Computing*, 27(10), 6487-6505.
- [26] Dbouk, W., Tarhini, H., & Nasr, W. (2023). Re-ordering policies for inventory systems with a fluctuating economic environment—Using economic descriptors to model the demand process. *Journal of the Operational Research Society*, 74(3), 860-872.
- [27] Güçdemir, H., & Taşoğlu, G. (2024). Part transformation-based spare parts inventory control model for the high-tech industries. *International Journal of Industrial Engineering Computations*, 15(1), 307-326.
- [28] Göçken, M., Boru, A., & Dosdoğru, A. T. (2018). İki Aşamalı tedarik zincirinde eş zamanlı stok kontrolünün ve tedarikçi seçiminin simülasyon optimizasyonu yaklaşımı ile analizi. *Karaelmas Fen ve Mühendislik Dergisi*, 8(1), 1-10.
- [29] Kobu, B. (2006). *Üretim Yönetimi*, 13. Baskı, İstanbul: Beta Basım.

- [30] Kaya, N. (2020). Stok Yönetimi, İksad Yayınevi: Erzurum.
- [31] Chase, R. B., Jacobs, F. R. & Aquilano, N. J. (2006). Operations management for competitive advantage. Mcgraw-Hill/İrwin. New York.
- [32] Krajewski, L. J., Ritzman, L. P. & Malhotra, M. K. (2022). Operations management: Processes and supply chains. Pearson. Kendallville in United States of America.
- [33] Vollmann, T., Berry, W. L., Whybark, D. C., & Jacobs, F. R. (2004). Manufacturing planning and control systems for supply chain management: the definitive guide for professionals. McGraw-Hill Professional, 5.
- [34] Schroeder, R., & Goldstein, S. M. (2018). Operations management in the supply chain. Total Supply Chain Management (Seventh Ed). McGraw-Hill Education. United States of America.
- [35] Max, M. (2003). Essentials of inventory management. American Management Association. 1601 Broadway, New York.
- [36] Silver, E. A., Pyke, D. F., & Thomas, D. J. (2016). Inventory and production management in supply chains. CRC press. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton.
- [37] Rushton, A., Croucher, P., & Baker, P. (2014). The handbook of logistics and distribution management: Understanding the supply chain. Kogan Page Publishers. India.
- [38] Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (1999). Designing and managing the supply chain: Concepts, strategies, and cases. New York: McGraw-hill. Fairfield.
- [39] Özgür, B. (2007). Envanter Yönetimi İçin Maliyet Parametrelerinin Hesaplanması: Otomotiv Sektöründe Bir Uygulama. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü İşletme Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi.
- [40] Slack, N., Chambers, S., & Johnston, R. (2010). Operations management. Pearson education. England.
- [41] Nahmias, S., & Olsen, T. L. (2015). Production and operations analysis. Waveland Press. United States of America.
- [42] Ertürk, M. (2017). İşletme biliminin temel ilkeleri. İstanbul: Beta Basım Yayım.
- [43] Bowersox, D. J., Closs, D. J., Cooper, M. B., & Bowersox, J. C. (2020). Supply chain logistics management. Mcgraw-hill. New York.
- [44] Behret, H. (2011). Üretim sistemlerinde bulanık tek dönemli stok kontrol modelleri. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Doktora Tezi.
- [45] Silver, E. A., Pyke, D. F., & Peterson, R. (1998). Inventory management and production planning and scheduling (Vol. 3, p. 30). New York: Wiley.

- [46] Yin, K. K., Liu, H., & Johnson, N. E. (2002). Markovian inventory policy with application to the paper industry. *Computers & chemical engineering*, 26(10), 1399-1413.
- [47] Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, 250-272.
- [48] Graves, S. C., Kan, A. R., & Zipkin, P. H. (Eds.). (1993). *Logistics of production and inventory* (Vol. 4). Elsevier. North Holland.
- [49] Winston, W. L. (2004). *Operations research: applications and algorithm*. Thomson Learning, Inc. Canada.
- [50] Aslantaş, M. (2019). Bir transformatör firmasında stok kontrol parametrelerinin simülasyon tabanlı optimizasyonu. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi.
- [51] Waters, D. (2021). *Logistics An Introduction to supply chain management*. Palgrave macmillan. Great Britain.
- [52] Blumenfeld, D. (2001). *Operations research calculations handbook*. CRC press. United States of America.
- [53] Guo, Y., Shi, Q., & Guo, C. (2022). Multi-Period Spare Parts Supply Chain Network Optimization under (T, s, S) Inventory Control Policy with Improved Dynamic Particle Swarm Optimization. *Electronics*, 11(21), 3454.
- [54] Law, A. M., Kelton, W. D., & Kelton, W. D. (2007). *Simulation modeling and analysis* (Vol. 3). New York: Mcgraw-hill.
- [55] Rossetti, M. D. (2016). *Simulation modeling and Arena*. John Wiley & Sons. United States of America.
- [56] Kelton, W. D., Sadowski, R. P., Zupick, N. B., (2015). *Simulation with Arena*. McGraw-Hill, Inc. United States of America.
- [57] Şirincan, S. (1992). Stok kontrol simülasyon sistemi. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi.
- [58] Kabadayı, N., & Özçakar, N. (2007). Seri tedarik zincirinde temel-stok seviyelerinin simülasyon temelli genetik algoritma ile çözülmesi. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Üretim Bilim Dalı. Yüksek Lisans Tezi.
- [59] Altioğ, T., & Melamed, B. (2010). *Simulation modeling and analysis with Arena*. Elsevier. United States of America.
- [60] April, Glover, & Laguna. (2003, December). Practical introduction to simulation optimization. In *Proceedings of the 2003 Winter Simulation Conference*, 2003. (Vol. 1, pp. 71-78). IEEE.
- [61] Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3), 192-215.
- [62] Smith, D. E. (1973). An empirical investigation of optimum-seeking in the computer simulation situation. *Operations Research*, 21(2), 475-497.

- [63] Tomar, V., Bansal, M., & Singh, P. (2024). Metaheuristic Algorithms for Optimization: A Brief Review. *Engineering Proceedings*, 59(1), 238.
- [64] Ólafsson, S., & Kim, J. (2002, December). Simulation optimization. In *Proceedings of the winter simulation conference* (Vol. 1, pp. 79-84). IEEE.
- [65] Goldberg, D. E. (1989). *Genetic Algorithms in Smith, Optimization, and Machine Learning*, Addison-Wesley Publishing Company Inc. United States of America.
- [66] Azzaro-Pantel, C., Bernal-Haro, L., Baudet, P., Domenech, S., & Pibouleau, L. (1998). A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. *Computers & chemical engineering*, 22(10), 1461-1481.
- [67] Grupe, F. H., & Jooste, S. (2004). Genetic algorithms: A business perspective. *Information management & computer security*, 12(3), 288-297.
- [68] Gen, M., & Cheng, R. (1999). *Genetic algorithms and engineering optimization* (Vol. 7). John Wiley & Sons. Canada.
- [69] Bodenhofer, U. (2003). *Genetic algorithms: theory and applications*. Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter, 2004.
- [70] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press. London, England.
- [71] Coley, D. A. (1999). *An introduction to genetic algorithms for scientists and engineers*. World Scientific Publishing Company. Singapore.
- [72] Kour, H., Sharma, P., & Abrol, P. (2015). Analysis of fitness function in genetic algorithms. *International Journal of Scientific and Technical Advancements*, 1(3), 87-89.
- [73] Vural, M. (2005). *Genetik algoritma yöntemi ile toplu üretim planlama*. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü İşletme Anabilim Dalı. Yüksek Lisans Tezi.
- [74] Altay, A. (2007). *Genetik algoritma ve bir uygulama*. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi.
- [75] Sakawa, M., & Kato, K. (2009). An Interactive Fuzzy Satisficing Method for Multiobjective Nonlinear Integer Programming Problems with Block-Angular Structures through Genetic Algorithms with Decomposition Procedures. *Advances in Operations Research*, 2009(1), 372548.
- [76] Haldurai, L., Madhubala, T., & Rajalakshmi, R. (2016). A study on genetic algorithm and its applications. *International Journal of computer sciences and Engineering*, 4(10), 139.
- [77] da Silva, A. P. A. (2002). *Tutorial Genetic Algorithms. Learning And Nonlinear Models*, SBRN, Brasil, 1, 43-58.
- [78] S. Bekiroğlu. (2003) *Genetik Algoritma ile Çelik Çerçevelerin Optimum Boyutlandırılması*. Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi.

- [79] Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on systems, man, and cybernetics*, 16(1), 122-128.
- [80] Biethahn, J., & Nissen, V. (Eds.). (1995). *Evolutionary algorithms in management applications*. Springer Science & Business Media.
- [81] Magoulas, G. D., Eldabi, T., & Paul, R. J. (2002, December). Global search strategies for simulation optimisation. In *Proceedings of the Winter Simulation Conference* (Vol. 2, pp. 1978-1985). IEEE.
- [82] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- [83] Erten, K. (2021). Uygunluk kısıtlı çok kaynaklı genelleştirilmiş atama problemi için bir tavlama simülasyonu algoritması. Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- [84] Gendreau, M., & Potvin, J. Y. (Eds.). (2010). *Handbook of metaheuristics* (Vol. 2, p. 9). New York: Springer.
- [85] Şahin R. (2004). Çok Kriterli Dinamik Tesisi Düzenleme Probleminin Tavlama Benzetimi ile Çözülmesi. Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı. Doktora Tezi.
- [86] Tasoglu, G. (2019). Solving Dynamic Berth Allocation And Quay Crane Scheduling Problem Simultaneously By Using Simulation Optimization. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Doktora Tezi.
- [87] Biroğul, S. (2005). Genetik algoritma yaklaşımıyla atölye çizelgeleme. Gazi Üniversitesi Fen Bilimleri Enstitüsü. Yüksek Lisans Tezi.
- [88] Onwubolu, G. C. (2002). *Emerging Optimization Techniques in Production Planning and Control*. Imperial College Press. Singapore.
- [89] Zbigniew, M. (1996). Genetic algorithms+ data structures= evolution programs. *Computational statistics*, 24, 372-373.
- [90] De Jong, K. (2017, July). Evolutionary computation: a unified approach. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 373-388).
- [91] Marzouk, M., & Moselhi, O. (2002). Simulation optimization for earthmoving operations using genetic algorithms. *Construction Management & Economics*, 20(6), 535-543.
- [92] Chan, F. T., Chung, S. H., & Wadhwa, S. (2005). A hybrid genetic algorithm for production and distribution. *Omega*, 33(4), 345-355.
- [93] Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley & Sons. New Jersey.
- [94] Buckles, B. P. (1992). *Genetic algorithms*. IEEE Computer Society Press. California.

- [95] Back, T. (1996). Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press. New York.
- [96] Busetti, F. (2003). Simulated annealing overview. World Wide Web URL [www. geocities. com/francorbusetti/saweb. pdf](http://www.geocities.com/francorbusetti/saweb.pdf), 4.
- [97] Kendall, G. (July 15, 2002). Simulated Annealing. Retrieved February 16, 2006, from www.cs.nott.ac.uk/~gmk/aim/notes/simulatedannealing.doc
- [98] Temiz, I. (2010). Çok Kriterli Permütasyon Akış Tipi Çizelgeleme Problemi için Bir Tavlama Simülasyonu Yaklaşımı. Cankaya University Journal of Science and Engineering, 7(2).
- [99] Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM computing surveys (CSUR), 35(3), 268-308.
- [100] Arenas, M. G., Laredo, J. L., Castillo, P. A., Sa, P. G., Mora, A. M., Prieto, A., & Mereño, J. J. (2010, July). Statistical analysis of the parameters of the simulated annealing algorithm. In IEEE Congress on Evolutionary Computation (pp. 1-8). IEEE.
- [101] Özcan, U., & Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. Computers & Industrial Engineering, 57(1), 217-227.
- [102] Manavizadeh, N., Hosseini, N. S., Rabbani, M., & Jolai, F. (2013). A Simulated Annealing algorithm for a mixed model assembly U-line balancing type-I problem considering human efficiency and Just-In-Time approach. Computers & industrial engineering, 64(2), 669-685.
- [103] Shapiro, J. (1999). Genetic algorithms in machine learning. In Advanced Course on Artificial Intelligence (pp. 146-168). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [104] Xue, H., Zhang, Z., Deng, S., Mu, L., Fu, J., & Diao, L. (2024). Improved Simulated Annealing Algorithm on the Design of Satellite Orbits for Common-View Laser Time Transfer. Remote Sensing, 16(3), 472.
- [105] Nasr, D., Dahr, R. E., Assaad, J., & Khatib, J. (2022). Comparative Analysis between Genetic Algorithm and Simulated Annealing-Based Frameworks for Optimal Sensor Placement and Structural Health Monitoring Purposes. Buildings, 12(9), 1383.
- [106] Chopra, S., & Meindl, P. (2007). Supply chain management. Strategy, planning & operation (pp. 265-275). Gabler.
- [107] Ballou, R. H., & Srivastava, S. K. (2007). Business logistics/supply chain management: planning, organizing, and controlling the supply chain. Pearson Education India.
- [108] Zipkin, P. H. (2000). Foundations of inventory management mcgraw-hill. Irwin, New York, USA.
- [109] Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2004). Managing the supply chain: the definitive guide for the business professional (No. 272468). Mcgraw-hill.

[110] Eskandari, H., Mahmoodi, E., Fallah, H., & Geiger, C. D. (2011, December). Performance analysis of commercial simulation-based optimization packages: OptQuest and Witness Optimizer. In Proceedings of the 2011 winter simulation conference (WSC) (pp. 2358-2368). IEEE.



EKLER

EK 6.1 Sipariş Miktarları İçin Rassal Örneklem Tablosu

Örneklem	Sipariş Miktarı	Örneklem	Sipariş Miktarı	Örneklem	Sipariş Miktarı	Örneklem	Sipariş Miktarı
1	386	26	133	51	405	76	328
2	404	27	353	52	64	77	413
3	305	28	336	53	449	78	97
4	135	29	292	54	0	79	349
5	122	30	193	55	124	80	510
6	430	31	130	56	64	81	497
7	545	32	14	57	529	82	0
8	511	33	0	58	98	83	430
9	2	34	0	59	546	84	138
10	354	35	16	60	80	85	528
11	19	36	271	61	468	86	307
12	113	37	112	62	383	87	465
13	552	38	545	63	375	88	69
14	192	39	301	64	131	89	193
15	297	40	369	65	409	90	335
16	269	41	367	66	478	91	486
17	69	42	297	67	350	92	328
18	492	43	371	68	213	93	53
19	169	44	528	69	535	94	175
20	147	45	170	70	437	95	452
21	216	46	141	71	477	96	288
22	116	47	485	72	551	97	62
23	283	48	521	73	342	98	16
24	88	49	96	74	446	99	0
25	279	50	138	75	152	100	55

EK 6.2 Tedarik Süreleri İçin Rassal Örneklem Tablosu

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

EK 6.3 OptQuest'te Kontrol Seçim Ekranı

Controls User Specified

User Specified Summary								
Included	Control /	Element Type	Type	Low Bound	Suggested Value	High Bound	Step	Description
☐	Anlik Siparis Miktar	Variable	Continuous	0	0	0	N/A	
☐	baslangicstok	Variable	Continuous	675000000	750000000	825000000	N/A	
☒	buyuks	Variable	Discrete	1000	1500	2000	1	
☐	Eldeki Stok Miktari	Variable	Continuous	675000000	750000000	825000000	N/A	
☐	Ertelenen Talep Miktari	Variable	Continuous	0	0	0	N/A	
☐	Ertelenmis Talep	Variable	Continuous	0	0	0	N/A	
☐	gelis suresi	Variable	Continuous	0	0	0	N/A	
☐	guncelstok	Variable	Continuous	675000000	750000000	825000000	N/A	
☒	kucuks	Variable	Discrete	500	750	2000	1	
☐	Satinalma Siparis Miktar	Variable	Continuous	0	0	0	N/A	
☐	Siparis Sayisi	Variable	Continuous	0	0	0	N/A	
☐	Stok Pozisyonu	Variable	Continuous	0	0	0	N/A	
☐	Stoktan Karsilanan Sip Sayisi	Variable	Continuous	0	0	0	N/A	
☐	Toplam Satinalma Siparis Miktar	Variable	Continuous	0	0	0	N/A	
☐	Toplam Satinalma Siparis Sayisi	Variable	Continuous	0	0	0	N/A	
☐	Toplam Siparis Miktari	Variable	Continuous	0	0	0	N/A	
☐	Yoldaki Stok Miktari	Variable	Continuous	0	0	0	N/A	

EK 6.4 OptQuest'teki Kısıtlar Penceresi (1)

OptQuest for Arena [C:\Users\win10\Desktop\tez\optimizasyon1.opt]

File Edit View Add Run Help

Controls

- User Specified
 - Anlik Siparis Miktar (Variable)
 - baslangicstok (Variable)
 - buyuks (Variable)
 - Eldeki Stok Miktari (Variable)
 - Ertelenen Talep Miktari (Variable)
 - Ertelenmis Talep (Variable)
 - gelis suresi (Variable)
 - guncelstok (Variable)
 - kucuks (Variable)
 - Satinalma Siparis Miktar (Variable)
 - Siparis Sayisi (Variable)
 - Stok Pozisyonu (Variable)
 - Stoktan Karsilanan Sip Sayisi (Variable)
 - Toplam Satinalma Siparis Miktari (Variable)
 - Toplam Satinalma Siparis Sayisi (Variable)
 - Toplam Siparis Miktari (Variable)
 - Yoldaki Stok Miktari (Variable)
- Responses
 - Entity
 - Queue
 - System
 - User Specified
- Constraints
 - New Constraint
 - New Constraint 2
- Objectives
 - New Objective

Controls

- buyuks (Variable)
- kucuks (Variable)

Responses

- Aylık Ortalama Toplam Maliyet (Output Value)
- Cevrim Hizmet Duzeyi 1 (Output Value)

Mathematical Functions:

- + * min
- / max
- sqrt log log10
- pow exp abs
- pi rand floor
- e fmod ceil
- sin cos tan
- sinh cosh tanh
- asin acos atan
- DtoR RtoD atan2
- <= = >=
- Sum All Controls

New Constraint

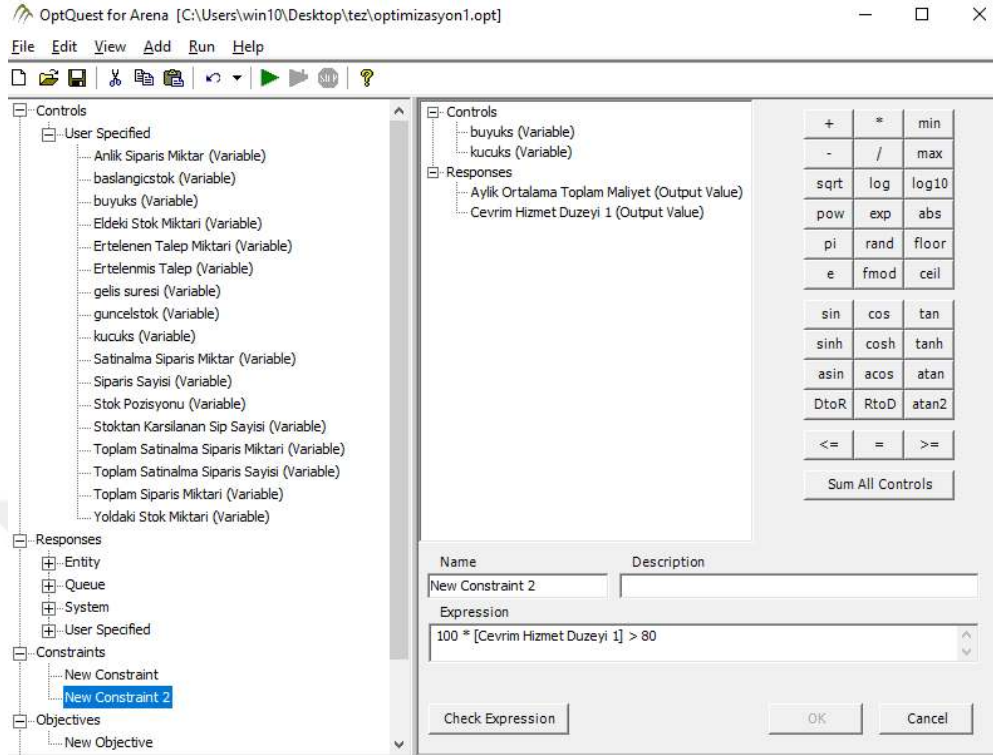
Name: New Constraint

Description:

Expression: [kucuks] < [buyuks]

Check Expression OK Cancel

EK 6.5 OptQuest'teki Kısıtlar Penceresi (2)



EK 6.6 OptQuest'teki Amaç ve Gereksinim Seçim Penceresi

