

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**HETEROJEN FİLOLU MİİK-RUN VE KONSOLİDE ÇAPRAZ
SEVKİYATLI TEDARİK ZİNCİRİ DAĞITIM AĞININ
OPTİMİZASYONU VE ÖZELLEŞTİRİLMİŞ GENETİK ALGORİTMA
İLE MODELLENMESİ**

Yasemin KOCAOĞLU

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Danışman

Prof. Dr. Alev TAŞKIN GÜMÜŞ

Şubat, 2020

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**HETEROJEN FİLOLU MİLK-RUN VE KONSOLİDE ÇAPRAZ SEVKİYATLI
TEDARİK ZİNCİRİ DAĞITIM AĞININ OPTİMİZASYONU VE
ÖZELLEŞTİRİLMİŞ GENETİK ALGORİTMA İLE MODELLENMESİ**

Yasemin KOCAOĞLU tarafından hazırlanan tez çalışması 07.02.2020 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. Alev TAŞKIN GÜMÜŞ
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

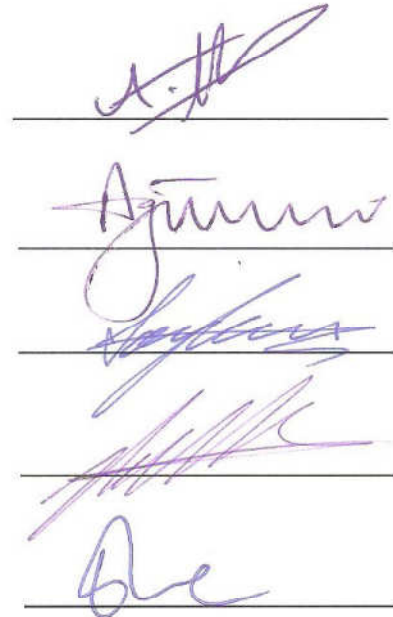
Prof. Dr. Alev TAŞKIN GÜMÜŞ, Danışman
Yıldız Teknik Üniversitesi

Prof. Dr. Ali Fuat GÜNERİ, Üye
Yıldız Teknik Üniversitesi

Prof. Dr. Bahar SENNAROĞLU, Üye
Marmara Üniversitesi

Doç. Dr. Bahadır GÜLSÜN, Üye
Yıldız Teknik Üniversitesi

Dr. Öğretim Üyesi Emre ÇAKMAK, Üye
Piri Reis Üniversitesi



Danışmanım Prof. Dr. Alev TAŞKIN GÜMÜŞ sorumluluğunda tarafımca hazırlanan Heterojen Filolu Milk-Run ve Konsolide Çapraz Sevkiyatlı Tedarik Zinciri Dağıtım Ağının Optimizasyonu ve Özelleştirilmiş Genetik Algoritma ile Modellenmesi başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Yasemin KOCAOĞLU

İmza



Aileme

ve

biricik eşime

Tedarik zinciri ağlarının planlanması, stratejik, taktik ve operasyonel seviyede kararların alınmasını gerektirmektedir. İyi bir tedarik zinciri dağıtım ağı, toplam dağıtım maliyetini mümkün olduğunca azaltarak, doğru ürünleri, doğru miktarda, doğru yerde, doğru zamanda müşteriye ulaştırmalıdır.

Bu tez çalışmasında, tedarik zinciri ağları planlamasının amaçlarından biri olan dağıtım problemi ele alınmıştır. Dağıtım yöntemi seçimi ve araç rotalama kararlarını içeren taktik ve operasyonel seviyede planlama yapılmıştır. Problemin çözümü için matematiksel model kurulmuş ve meta sezgisel algoritma geliştirilmiştir. İki çözüm yöntemi için sonuçlar karşılaştırılmıştır ve geliştirilen meta sezgisel yöntemle elde edilen sonuçların optimale yakınlığı değerlendirilmiştir.

Tez çalışması sırasında, desteğini ve yardımlarını esirgemeyen, değerli danışman hocam Prof. Dr. Alev Taşkın Gümüş'e sonsuz teşekkürlerimi sunarım. Tez izleme komitesinde bulunarak tezimin ilerlemesine büyük katkıda bulunan değerli hocalarım Prof. Dr. Bahar Sennaroğlu'na ve Prof. Dr. Ali Fuat Güneri'ye teşekkürlerimi sunarım. Yardımlarını ve fikirlerini esirgemeyen değerli hocam Yrd. Doç. Dr. Emre Çakmak'a teşekkür ederim. Bilgisi ve değerli fikirleriyle her zaman yanımda olan sevgili eşim Doç Dr. Batuhan Kocaoğlu'na ve doktora yapmamı destekleyen sevgili aileme çok teşekkür ederim.

Yasemin KOCAOĞLU

İÇİNDEKİLER

SİMGE LİSTESİ	ix
KISALTMA LİSTESİ	x
ŞEKİL LİSTESİ	xi
TABLO LİSTESİ	xiii
ÖZET	xvi
ABSTRACT	xviii
1 Giriş	20
1.1 Literatür Özeti	21
1.1.1 Araç Kısıtları, Taşıma Kısıtları ve Dağıtım Stratejisi Parametrelerine Göre Çalışmaların Sınıflandırılması	24
1.1.2 Amaç ve Çözüm Yöntemleri Parametrelerine Göre Çalışmaların Sınıflandırılması	32
1.2 Tezin Amacı	44
1.3 Hipotez	44
2 Tedarik Zinciri Yönetimi	45
2.1 Tedarik Zinciri ve Tedarik Zinciri Yönetimi Kavramı	45
2.2 Tedarik Zinciri Yönetiminin Amaçları	46
2.3 Tedarik Zinciri Yönetiminin Temel Fonksiyonları	47
2.4 Tedarik Zinciri Yönetiminde Dağıtım Ağlarının Planlanması	48
2.4.1 Doğrudan Sevkiyat Ağı (Direkt Taşıma)	49
2.4.2 Milk-run (Döngüsel Sefer)	50
2.4.3 Merkezi Dağıtım Merkezi ile Sevkiyat	51
2.4.4 Dağıtım Merkezinden Yapılan Stoklu Taşıma	52
2.4.5 Çapraz Sevkiyat (Cross-docking) ile Taşıma	52

2.4.6	Milk-Run Kullanarak Dağıtım Merkezi ile Sevkiyat.....	54
2.4.7	Uyarlanmış Ağlar (Tailored Networks).....	54
2.5	Tedarik Zincirinde Dağıtımda Dış Kaynak Kullanımı	55
3	Araç Rotalama Problemleri	57
3.1	Araç Rotalama Problem Çeşitleri.....	58
3.1.1	Klasik Araç Rotalama Problemleri.....	58
3.1.2	Heterojen Filolu Araç Rotalama Problemi (HFARP)	59
3.1.3	Açık Uçlu Araç Rotalama Problemi (AUARP).....	59
3.1.4	Heterojen Filolu Açık Uçlu Araç Rotalama Problemi (HFAUARP).....	59
3.1.5	Zaman Pencere Araç Rotalama Problemi (ZARP).....	59
3.1.6	Kapasite Kısıtlı Araç Rotalama Problemi (KKARP)	60
3.1.7	Mesafe Kısıtlı Araç Rotalama Problemi (MKARP)	60
3.1.8	Önce Dağıt Sonra Topla Araç Rotalama Problemi (ÖDSTARP).....	60
3.1.9	Eş Zamanlı Topla Dağıt Araç Rotalama Problemi (EZTDARP)	60
3.1.10	Dinamik Araç Rotalama Problemi (DARP).....	60
3.1.11	Çok Depolu Önce Dağıt Sonra Topla Araç Rotalama Problemi (ÇDÖDSTARP).....	61
3.1.12	Stokastik Araç Rotalama Problemi	61
3.1.13	Bölünebilir Talepli Araç Rotalama Problemi (BTARP)	61
3.1.14	Geri Toplamalı ARP (GTARP).....	61
3.1.15	Periyodik ARP (PARP).....	61
3.2	Milk-run ve Araç Rotalama.....	62
3.3	ARP'nin Çözümünde Kullanılan Yöntemler.....	63
3.3.1	Kesin Çözüm Yöntemleri.....	63
3.3.2	Klasik Sezgisel Yöntemler.....	64
3.3.3	Meta-Sezgisel Yöntemler.....	66

4	Tez Çalışmasında Kullanılan Yöntemler	68
4.1	Clarke ve Wright'ın Tasarruf Algoritması.....	68
4.2	Genetik Algoritma (GA).....	72
4.2.1	Çözümlerin Kodlanması.....	75
4.2.2	Başlangıç Popülasyonun Oluşturulması.....	77
4.2.3	Uygunluk Fonksiyonun Belirlenmesi	77
4.2.4	Seçim (Çoğalma)	78
4.2.5	Çaprazlama.....	79
4.2.6	Mutasyon.....	83
4.2.7	Yeni Kuşağın Oluşması ve Döngünün Durdurulması.....	84
4.2.8	Genetik Algoritmalarda Parametre Seçimi	84
5	Tedarik Zinciri Dağıtım Ağı Probleminin Özelleştirilmiş GA ile Çözümlemesi	86
5.1	Firma Tanıtımı.....	86
5.2	Problemin Tanımı.....	87
5.2.1	Direkt Taşıma.....	89
5.2.2	Milk-run ile taşıma.....	89
5.2.3	Çapraz Sevkiyat İle Taşıma.....	89
5.3	Problemin Matematiksel Modeli	89
5.4	Geliştirilen Hibrit Algoritma ile Problemin Çözümü.....	93
5.4.1	Adım 1: Başlangıç Popülasyonun Oluşturulması.....	93
5.4.2	Adım 2: Strateji= Milk-run için Modifiye edilmiş Tasarruf Algoritma Uygulanması.....	94
5.4.3	Adım 3: Eğer metot= Direkt Sevkiyat ise, Direkt Prosedürü Uygula.....	97
5.4.4	Adım 4: Eğer metot=Çapraz Sevkiyat ise,Çapraz Sevkiyat Prosedürünü Uygula.....	97

5.4.5	Adım 5: Genetik Algoritma- Uygunluk Değerinin Değerlendirilmesi	97
5.4.6	Adım 6: Genetik Algoritma- Yeni Popülasyonu Oluşturma.....	97
5.4.7	Adım 7: Genetik Algoritma: Elitizm Uygula.....	98
5.4.8	Adım 8: Genetik Algoritma: Algoritmanın Sonlandırılması.....	98
5.5	Tedarik Zinciri Dağıtım Ağı Probleminin Çözülmesi	101
5.5.1	Tedarik Zinciri Dağıtım Probleminin Küçük Büyüklükteki Veri Seti için Lingo ile Çözülmesi	102
5.5.2	Tedarik Zinciri Dağıtım Probleminin Özelleştirilmiş Genetik Algoritma (GA) ile Çözüm Sonuçları	126
5.5.3	Çözümlerin Karşılaştırılması.	175
6	Sonuç ve Öneriler	179
	Kaynakça	181
	A Özelleştirilmiş Genetik Algoritma Kaynak Kodu	196
	B Çözümde Kullanılan Müşteri Verileri	222
	Tezden Üretilmiş Yayınlar	228

SİMGE LİSTESİ

C_{ij}	i. ve j. müşteri arasındaki km cinsinden uzaklık
d_j	J müşterisinin talep miktarı
D	Milk-run turu uzunluğu
k	Çapraz sevkiyat merkezi
K_v	V aracının maksimum araç yükleme kapasitesi
L_v	Araç kiralama maliyeti
N_0	Tedarikçi/Üretici dâhil olmak üzere müşteri düğüm kümesi
M	Üretici/tedarikçiden çapraz sevkiyat merkezine nakliye için araç sayısı
N	Müşteri düğüm kümesi
O	Üretici/tedarikçiden çapraz sevkiyat merkezine sabit araç taşıma maliyeti
P	Tüketilen birim benzin maliyeti
S	Çapraz sevkiyat merkezine atanan aracın kapasitesi
T	Çapraz sevkiyat merkezinden müşterilere direkt dağıtım maliyeti
V	Araç kümesi
W	Milk-run durak maliyeti
X_{ijv}	V aracı ile müşteri i den j ye dağıtım karar değişkeni
Y_j	Müşteri j talebinin çapraz sevkiyat ile dağıtım karar değişkeni
Z_{jv}	V aracı ile müşteri j ye direkt dağıtım yapma karar değişkeni

KISALTMA LİSTESİ

ARP	Araç rotalama problemi
AUARP	Açık uçlu araç rotalama problemi
BTARP	Bölünebilir talepli araç rotalama problemi
CPU	Merkezi işlem birimi
CX	Dairesel çaprazlama
ÇDÖDSTARP	Çok depolu önce dağıt sonra topla araç rotalama problemi
DARP	Dinamik araç rotalama problemi
4PL	Dördüncü parti lojistik
EZTDARP	Eş zamanlı topla dağıt araç rotalama problemi
GA	Genetik algoritma
GTARP	Geri toplamalı araç rotalama problemi
GSP	Gezgin satıcı problemi
HFARP	Heterojen filolu araç rotalama problemi
HFAUARP	Heterojen filolu açık uçlu araç rotalama problemi
KARP	Klasik araç rotalama problemi
KKARP	Kapasite kısıtlı araç rotalama problem
KM	Kilometre
LLP	Lider lojistik hizmetleri tedarikçisi
Milk-Run	Döngüsel sefer uygulaması
MKARP	Mesafe kısıtlı araç rotalama problemi
NP	Polinomal olmayan zor
OX1	Sıralı çaprazlama
OX2	Sıraya dayalı çaprazlama
ÖDSTARP	Önce dağıt sonra topla araç rotalama problemi
PARP	Periyodik araç rotalama problemi
PMX	Kısmi planlı çaprazlama
POS	Pozisyona dayalı çaprazlama
Sn.	Saniye
TZY	Tedarik zinciri yönetimi
3PL	Üçüncü parti lojistik
ZARP	Zaman pencereci araç rotalama problemi

ŞEKİL LİSTESİ

Şekil 2.1	Tedarik Zinciri Aktörleri	46
Şekil 2.2	Tedarik Zinciri Yönetimi Fonksiyonları.....	47
Şekil 2.3	Doğrudan Sevkiyat Ağı.....	49
Şekil 2.4	Milk-run Çeşitleri.....	50
Şekil 2.5	Dağıtım Merkezi ile Sevkiyat.....	52
Şekil 2.6	Milk-run Kullanarak Dağıtım Merkezi Sevkiyatı	54
Şekil 4.1	Tasarruf Algoritması Gösterimi	69
Şekil 4.2	Genetik Algoritma Adımları.....	74
Şekil 4.3	İkili Kodlama Gösterimi.....	75
Şekil 4.4	Permütasyon Kodlama Gösterimi.....	75
Şekil 4.5	Değer Kodlama Gösterimi.....	76
Şekil 4.6	Ağaç Kodlama Gösterimi	76
Şekil 4.7	Sekizli Kodlama Gösterimi	76
Şekil 4.8	On Altılı Kodlama Gösterimi.....	77
Şekil 4.9	Tek Noktalı Çaprazlama.....	80
Şekil 4.10	Uniform Çaprazlama.....	80
Şekil 4.11	Sıralı çaprazlama.....	81
Şekil 4.12	Kısmi Planlı Çaprazlama.....	81
Şekil 4.13	Döngüsel Çaprazlama.....	82
Şekil 4.14	Pozisyona Dayalı Çaprazlama	82
Şekil 4.15	Ters Çevirme Mutasyonu	83
Şekil 4.16	Değiş -Tokuş Mutasyonu.....	83
Şekil 4.17	Ekleme Mutasyonu.....	84
Şekil 4.18	Yer Değiştirme Mutasyonu.....	84
Şekil 5.1	Tedarik Zinciri Ağı Yapısı.....	88
Şekil 5.2	Çözüm İçin Kullanılan Kromozom Örneği	94
Şekil 5.3	Modifiye Tasarruf Algoritması Akışı.....	95
Şekil 5.4	Önerilen Hibrit Algoritma Akışı.....	99
Şekil 5.5	4 Müşterili Problem için Lingo Sonuç Ekranı.....	104
Şekil 5.6	5 Müşterili Problem için Lingo Sonuç Ekranı.....	106

Şekil 5.7	6 Müşterili Problem için Lingo Sonuç Ekranı.....	108
Şekil 5.8	7 Müşterili Problem İçin Lingo Sonuç Ekranı.....	110
Şekil 5.9	8 Müşterili Problem İçin Lingo Sonuç Ekranı.....	113
Şekil 5.10	9 Müşterili Problem İçin Lingo Sonuç Ekranı.....	116
Şekil 5.11	10 Müşterili Problem İçin Lingo Sonuç Ekranı.....	119
Şekil 5.12	11 Müşterili Problem İçin Lingo Sonuç Ekranı.....	122
Şekil 5.13	12 Müşterili Problem İçin Lingo Sonuç Ekranı.....	125
Şekil 5.14	Taguchi Orthogonal Dizi Seçim Tablosu.....	127
Şekil 5.15	Taguchi Deneyleri İçin Ortalama Sonuç Grafiği	133
Şekil 5.16	Taguchi Deneyleri İçin Standart Sapma Sonuç Grafiği.....	133
Şekil 5.17	Taguchi Deneyleri İçin S/N Sonuç Grafiği.....	134
Şekil 5.18	Çözüm Maliyetlerinin Karşılaştırılması.....	176
Şekil 5.19	Çözüm Sürelerinin Karşılaştırılması.....	176
Şekil 5.20	Kolmogorov Smirnov Testi Sonucu.....	177
Şekil 5.21	Eşleştirilmiş t Testi Sonuçları	178

TABLO LİSTESİ

Tablo 1.1	İncelenen Çalışmaların Yıllara Göre Dağılımı.....	22
Tablo 1.2	Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma.....	25
Tablo 1.3	Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma.....	33
Tablo 1.4	Dağıtım Stratejileri Kapsamında Çalışmaların Özetlenmesi.....	42
Tablo 2.1	Dağıtım Ağı Seçeneklerinin Avantaj ve Dezavantajları.....	55
Tablo 4.1	Düğüm Arası Uzaklıklar (km) ve Talep Miktarı (m^3).....	70
Tablo 4.2	Hesaplanan Tasarruflar	70
Tablo 4.3	Tasarrufların Büyükten Küçüğe Sıralanması	71
Tablo 5.1	Araç Kapasiteleri ve Maliyetleri.....	102
Tablo 5.2	Diğer Parametreler	102
Tablo 5.3	Müşteriler Arası Uzaklık (km)	102
Tablo 5.4	Müşteri Tedarikçi/Üretici Arası Uzaklık (km).....	102
Tablo 5.5	Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	103
Tablo 5.6	Müşteri Talep Miktarları (m^3)	103
Tablo 5.7	Çapraz Sevkiyat Merkezi Direkt Taşıma Maliyeti (TL)	103
Tablo 5.8	Müşteriler Arası Uzaklık (km)	104
Tablo 5.9	Müşteri Tedarikçi/Üretici Arası Uzaklık (km).....	105
Tablo 5.10	Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	105
Tablo 5.11	Müşteri Talep Miktarları (m^3).....	105
Tablo 5.12	Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)	105
Tablo 5.13	Müşteriler Arası Uzaklık (km)	107
Tablo 5.14	Müşteri Tedarikçi/Üretici Arası Uzaklık (km)	107
Tablo 5.15	Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	107
Tablo 5.16	Müşteri Talep Miktarları (m^3).....	107

Tablo 5.17 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)	107
Tablo 5.18 Müşteriler Arası Uzaklık (km)	109
Tablo 5.19 Müşterilerin Tedarikçi/Üretici Arası Uzaklık (km)	109
Tablo 5.20 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)	109
Tablo 5.21 Müşteri Talep Miktarları(m ³).....	110
Tablo 5.22 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)	110
Tablo 5.23 Müşteriler Arası Uzaklık (km)	111
Tablo 5.24 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)	112
Tablo 5.25 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)	112
Tablo 5.26 Müşteri Talep Miktarları (m ³).....	112
Tablo 5.27 Çapraz sevkiyat merkezi taşıma maliyeti(TL)	112
Tablo 5.28 Müşteriler Arası Uzaklık (km)	114
Tablo 5.29 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)	114
Tablo 5.30 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	114
Tablo 5.31 Müşteri Talep Miktarları (m ³).....	115
Tablo 5.32 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti(TL).....	115
Tablo 5.33 Müşteriler Arası Uzaklık (km)	117
Tablo 5.34 Müşteri Tedarikçi/Üretici Arası Uzaklık(km)	117
Tablo 5.35 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	117
Tablo 5.36 Müşteri Talep Miktarları (m ³).....	117
Tablo 5.37 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti(TL).....	118
Tablo 5.38 Müşteriler Arası Uzaklık (km)	120
Tablo 5.39 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)	120
Tablo 5.40 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km).....	120
Tablo 5.41 Müşteri Talep Miktarları (m ³).....	121
Tablo 5.42 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti(TL).....	121

Tablo 5.43 Müşteriler Arası Uzaklık (km)	123
Tablo 5.44 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)	123
Tablo 5.45 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)	123
Tablo 5.46 Müşteri Talep Miktarları (m ³).....	124
Tablo 5.47 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti(TL)	124
Tablo 5.48 Değişik Seviyelerde Belirlenen GA Parametreleri	127
Tablo 5.49 Taguchi Metodu ile Belirlenen Deneyler	128
Tablo 5.50 Taguchi Deneyleri Özelleştirilmiş GA Çözümleri.....	129
Tablo 5.51 Taguchi Deney Hesaplamaları	132
Tablo 5.52 Taguchi Sonucu Bulunan Optimum Özelleştirilmiş GA Değerleri.....	134
Tablo 5.53 4 Müşterili Veri İçin Modifiye Özelleştirilmiş GA Çözümleri.....	135
Tablo 5.54 5 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri	137
Tablo 5.55 6 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri	138
Tablo 5.56 7 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri	140
Tablo 5.57 8 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri	142
Tablo 5.58 9 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri	144
Tablo 5.59 10 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	146
Tablo 5.60 11 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	148
Tablo 5.61 12 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	151
Tablo 5.62 20 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	153
Tablo 5.63 30 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	156
Tablo 5.64 50 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	160
Tablo 5.65 100 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri.....	165
Tablo 5.66 Çözümlerin Karşılaştırılması.....	175

Heterojen Filolu Milk-Run ve Konsolide Çapraz Sevkiyatlı Tedarik Zinciri Dağıtım Ağının Optimizasyonu ve Özelleştirilmiş Genetik Algoritma ile Modellenmesi

Yasemin KOCAOĞLU

Endüstri Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Prof. Dr. Alev TAŞKIN GÜMÜŞ

Tedarik zinciri dağıtım ağını optimal şekilde yönetmek, firmalar için hayati önem taşımaktadır. Müşterilerin talepleri ve yerleri sipariştten siparişe göre değişebilmekte ve bu durumda firmalar teslimat maliyetlerini azaltmak için farklı dağıtım stratejileri aramaya yönelmektedir. Uygun dağıtım stratejisinin ve dağıtım rotasının belirlenmesi, dağıtım maliyetlerini düşürür, müşteri hizmet düzeyini arttırır ve firmaların kârlılığını olumlu yönde etkiler. Müşteri yerlerinin değişkenlik gösterdiği durumda, “karma dağıtım ağı” farklı stratejilerin birlikte kullanılmasına olanak ve esneklik tanıyan iyi bir çözümdür. “Dağıtım stratejisi seçimi” şirketler için önemli bir konu olsa da araç rotalama problemine ve dağıtım stratejisi seçimine birlikte odaklanan sadece birkaç çalışma vardır. Buna ek olarak, “karma dağıtım ağı” sorununu inceleyen az çalışma yer almakta olup, bu problemi çözmek için tasarruf ve genetik algoritmayı birlikte kullanarak geliştirilmiş bir çözüm yoktur.

Bu çalışmada, milk-run (döngüsel sefer), direkt dağıtım ve çapraz sevkiyatlı teslimat yöntemlerini içeren “karma dağıtım” ağı problemi incelenmiştir. “Karma dağıtım” ağı stratejileri kullanılarak müşterilere ürünün en optimal maliyette ve zamanda

ulařtırılabilmesi amacıyla, en uygun dađıtım yönteminin ve en uygun rotanın belirlenmesi ele alınmıřtır. İncelenen problemin polinom zamanda çözülemeyen zor (NP-hard) yapısı nedeniyle, “karma dađıtım” problemine çözümler sağlamak amacıyla hibrit bir algoritma geliştirilmiřtir. Hibrit algoritmanın geliştirilmesinde, tasarruf algoritmasından ve genetik algoritmadan yararlanılmıřtır. Hibrit algoritma çözümleri ile, optimal çözümler karşılaştırılmıřtır. Geliřtirilen hibrit algoritmanın sağladığı çözümlerin doğruluđunu ölçümlemek için, istatistiksel testler yürütölmüřtür. Yürütölen testler sonucunda, geliştirilen yöntemin optimale yakın sonuçlar verdiđi ve dađıtım maliyetleri ile hesaplama zamanını azaltmak için iyi bir çözümler sağladığı gözlemlenmiřtir.

Geliřtirilen, yeni algoritma, literatüre “karma dađıtım ađı” sorununu verimli bir şekilde çözmek için yeni bir hibrit çözümler sunmaktadır. Ayrıca literatürde az incelenmiř dađıtım stratejisi seçimi ve karma dađıtım ađı problemini incelemesi yönleriyle de literatüre katkı sağlamaktadır.

Anahtar Kelimeler: Dađıtım stratejisi seçimi, araç rotalama, karma tedarik zinciri dađıtım ađı problemi, tasarruf algoritması, genetik algoritma.

Optimization and Modelling of Heterogeneous Fleeted Milk-Run and Consolidated Cross-Docking Supply Chain Distribution Network via Modified Genetic Algorithm

Yasemin KOCAOĞLU

Department of Industrial Engineering

Doctor of Philosophy Thesis

Advisor: Prof. Dr. Alev TAŞKIN GÜMÜŞ

Optimally managing the supply chain distribution network is vital for companies. Müşteris' demands and locations can vary from order to order, and companies are looking for different delivery strategies to reduce delivery costs. Determining the appropriate distribution strategy and distribution route reduces distribution costs, increases Müşteri service level, and positively affects firms' profitability. Where Müşteri locations change, a mixed distribution network is a good solution that provides the flexibility to use different strategies together. Although “distribution strategy selection is an important issue for companies, there are only a few studies focus together on vehicle routing problem and distribution strategy selection. In addition to this, there are few studies about a mixed delivery network problem. And there is no improved solution to solve this problem by using the savings and genetic algorithm. In this study, a mixed delivery network problem, which includes direct shipment, milk run shipment, and cross docking shipment, is studied. Using the mixed distribution network strategies, the most appropriate distribution method, and the most suitable route provided the optimal cost and time are determined. Due

to the non-polynomial-hard (NP-hard) nature of the problem, a hybrid algorithm is developed to solve the distribution problem for a large number of Müşteri clusters. The saving algorithm and genetic algorithm is used in the development of the hybrid algorithm. Hybrid algorithm solutions and optimal solutions is compared. Statistical tests are conducted to measure the accuracy of the solutions provided by the hybrid algorithm developed. As a result of the tests carried out, it has been observed that the developed method gives close results to optimal results and provides an efficient solution in reducing the distribution costs and calculation time. At the same time, it contributes to the literature in the examination of the less studied a mixed distribution network and also in the less studied selection problem of distribution strategy.

The new algorithm offers a new hybrid solution to the literature in order to solve the mixed distribution network problem efficiently. It also contributes to the literature in examining a less studied a mixed distribution network, as well as the less studied distribution strategy selection problem.

Keywords: Distribution strategy selection, vehicle routing, a mixed supply chain delivery network problem, savings algorithm, genetic algorithm.

Günümüzün rekabetçi ortamı, organizasyonları müşterilerin taleplerini minimum zaman ve maliyetle karşılamak için daha iyi dağıtım stratejileri bulmaya zorlamaktadır. Bu nedenle, dağıtım stratejisi seçimi, her dağıtım ağı planlaması için hayati bir sorundur. Dağıtım problemleri için optimal çözümlerin elde edilmesi son derece zordur, bu nedenle dağıtım ağı planlamasında dağıtım stratejisini seçmek ve dağıtım rotasını oluşturmak için iyi sezgiseller geliştirmek çok önemlidir.

Dağıtım ağı planlamasında “doğrudan sevkiyat”, “milk-run (döngüsel sefer)”, “çapraz sevkiyat” ve “karma sevkiyat” gibi farklı dağıtım stratejileri kullanılabilmektedir. “Doğrudan sevkiyat” taşımacılığın tedarikçilerden müşterilere veya üreticilere veya perakendecilere doğrudan yapıldığı bir dağıtım işlemidir. “Milk-run” aracın tedarik noktasından ayrıldığı ve tüm teslimatlar tamamlandıktan sonra tedarik noktasına geri döndüğü bir yalın lojistik uygulamasıdır. “Çapraz sevkiyat”, ürünlerin çapraz sevkiyat merkezinde toplandığı ve sonrasında çapraz sevkiyat merkezinden müşteriye gönderildiği bir dağıtım işlemidir. “Karma sevkiyat” sistemi, iki veya üç dağıtım modundan oluşan hibrit bir dağıtım sistemidir. Dağıtım yerleri ve dağıtım miktarları sipariştten siparişe göre değişebilmektedir. Bu durumda, dağıtım ağı planlamalarında karma dağıtım stratejisi kullanmak tek bir dağıtım stratejisinden kullanmaktan daha avantajlı olabilmektedir.

Dağıtım stratejisi seçimi işletmeler için dağıtım ağı planlamasında kritik bir konu olmasına rağmen, literatürde “dağıtım stratejisi seçimi” ve “araç rotalama problemi” üzerine aynı anda odaklanan birkaç dağıtım ağı planlama çalışması bulunmaktadır. Bu tez çalışmasında, “doğrudan sevkiyat”, “milk-run” ve “çapraz sevkiyat” dağıtım stratejilerinden oluşan karma bir dağıtım ağı problemini çözmek için özelleştirilmiş Clarke ve Wright’ın tasarruf algoritması ve genetik algoritmanın kullanılması ile hibrit sezgisel algoritma geliştirilmiştir. Geliştirilen algoritma dağıtım için uygun dağıtım stratejisine ve rotalara karar vermektedir.

Bu çalışma, literatürde az çalışılmış karma dağıtım sistemi problemine yeni bir sezgisel çözüm sunarak literatüre katkı sağlamaktadır.

Tez çalışması, toplam altı bölümden oluşmaktadır. Birinci bölümde, literatür araştırmasına ilişkin yapılan araştırmalardan, tezin amacı ve hipotezden bahsedilmektedir. İkinci bölümde tedarik zinciri yönetimi ve kavramlarından, tedarik zinciri yönetiminin temel fonksiyonlarından, tedarik zinciri yönetiminin amaçlarından, tedarik zinciri yönetiminde dağıtım ağlarının planlanmasından ve tedarik zincirinde dağıtımda dış kaynak kullanımından bahsedilmektedir. Üçüncü bölümde, araç rotalama problemi çeşitlerinden, milk-run ve araç rotalamadan, ve araç rotalama probleminin (ARP) çözümünde kullanılan yöntemlerden bahsedilmektedir. Dördüncü bölümde, tez çalışmasında kullanılan Clarke ve Wright'ın tasarruf algoritması ve genetik algoritma anlatılmaktadır. Beşinci bölümde, çalışmaya ilişkin geliştirilen algoritma ve optimizasyon modelinden, geliştirilen modelin ve algoritmanın gerçek hayat problemi üzerinde yapılan uygulamasından, uygulamanın sonuçlarından, sonuçların karşılaştırılarak sonuçlara ilişkin yürütülen istatistiksel testlerden ve uygulamanın yapıldığı firmaya ilişkin genel bilgilerden bahsedilmektedir. Altıncı bölümde, çalışmanın sonuçlarına ilişkin elden edilen bilgiler özetlenmektedir, ayrıca sonuçların yanı sıra çalışmanın literatüre katkıları, kısıtları ve gelecek çalışmalara ilişkin öneriler de yer almaktadır.

1.1 Literatür Özeti

Tez çalışmasında yürütülen literatür araştırması için yök tezleri, Science Direct, Springer ve Kluwer, Elsevier, Taylor ve Francis, Wiley, Informs, Academic Journals, IEEE Xplore, Irjes makale veritabanlarından ve internet kaynaklarından yararlanılmıştır. Arama yapılırken anahtar kelimeler olarak şunlar kullanılmıştır: Tedarik zincirinde dağıtım, dağıtım stratejisi seçimi, araç rotalama, milk-run ile dağıtım, çapraz sevkiyat ile dağıtım, kesin yöntemler ile dağıtım ağı planlaması, sezgisel yöntemler ile dağıtım ağı planlaması, çapraz sevkiyat ile dağıtımda genetik algoritma ve Clarke ve Wright tasarruf algoritmaları kullanımı, milk-run ile dağıtımda genetik algoritma, ve Clarke ve Wright tasarruf algoritmaları kullanımı. 1989-2019 yılları arasındaki 97 çalışma incelenmiştir. Literatür araştırmasında incelenen çalışmaların, yıllara göre dağılımı Tablo 1.1'de gösterilmiştir.

Tablo 1.1 İncelenen Çalışmaların Yıllara Göre Dağılımı

Çalışma	Yayın Yılı	Sayı
[1]	1989	1
[2]	1993	1
[3,4]	1994	2
[5-7]	2003	3
[8,9]	2004	2
[10]	2005	1
[11-14]	2006	4
[15-17]	2007	3
[18,19]	2008	2
[20-23]	2009	4
[24-29]	2010	6
[30-33]	2011	4
[34,35]	2012	2
[36-46]	2013	10
[47-57]	2014	11
[58-66]	2015	9
[67-79]	2016	13
[80-89]	2017	10
[90-94]	2018	5
[95-97]	2019	3
Toplam		97

Literatür araştırmasında incelenen çalışmaları sınıflandırmak için kullanılan parametreler şunlardır:

Amaç: Optimizasyon çalışmalarında amaçlar, maliyet minimizasyonu, fayda maksimizasyonu, müşteri memnuniyetinin maksimizasyonu gibi konuları içermektedir. Bu amaçlar optimizasyon modelleri içinde amaç fonksiyonunda tanımlanırlar.

Çözüm Yöntemleri (Kesin/Sezgisel/Meta-sezgisel): Tedarik zinciri dağıtım ağı planlamasında, problemlerin çözümünde kullanılacak yöntemlere veri büyüklüğüne göre karar verilmektedir. Küçük veriler için kabul edilebilir zamanlarda sonuç sağlayan doğrusal programlama, hedef programlama, dal sınır algoritması gibi kesin çözüm yöntemleri optimal sonuçlar vermektedir. Veri büyüklüğü arttıkça, problemlerin polinom zamanda çözilemeyen zor (NP-hard) yapısından ötürü, kesin yöntemlerle kısa zaman içerisinde sonuç elde etmek mümkün olmamaktadır. Bu nedenle sezgisel ve meta-sezgisel yöntemlerle, çözüm elde edilmeye çalışılmaktadır.

Araç Kısıtları: Araç rotalama problemlerinde kullanılan araçlara ilişkin kısıtları içermekte olup, aşağıda bahsedilmiştir:

- Araç Tipi (Homojen/Heterojen): Araç rotalama problemlerinde kullanılan araç tipleri kapasite, hız vs. gibi özellikler nedeniyle homojen veya heterojen yapıda olabilmektedir.
- Araç Sayısı (Limitli/Limitsiz): Araç rotalama problemlerinde kullanılan araç sayısı, sınırlı veya sınırsız olabilmektedir.
- Kapasite Kısıtı: Araç rotalama problemlerinde kullanılan araçların yüklenmesi için kapasite sınırlaması olabilmektedir.

Taşıma Kısıtları: Zaman ve mesafe ile ilgili kısıtları içermekte olup, aşağıda bahsedilmiştir:

- Mesafe Kısıtı: Rotaları belirlenmiş her aracın kat edebileceği belirli bir toplam mesafenin olmasıdır.
- Zaman Penceresi: Zaman pencereleri, müşteriler için servis başlama ve bitiş zamanının bulunduğu, zaman aralıklarıdır. Araç rotalamasında yer alan her araç belirli bir zaman penceresinde müşterilerine servis yapmak durumundadır.
- Zaman Kısıtı: Araç rotalama problemlerinde, bir aracın seyahat edilebileceği toplam seyahat zamanı kısıtı bulunabilmektedir.
- Bölünebilir Talep: Araçlar müşteri taleplerinin karşılanmasında aynı müşteriye bir kez veya çok sayıda ziyaret gerçekleştirebilmektedir. Çok sayıda ziyaret gerçekleştirilen problemlerde, müşterinin talebi bölünerek karşılanmaktadır.

Dağıtım Stratejisi: Müşterilerin taleplerinin karşılanmasında direkt dağıtım, milk-run ile dağıtım, çapraz sevkiyat ile dağıtım şeklinde farklı dağıtım yöntemleri uygulanabilmektedir.

1.1.1 Araç Kısıtları, Taşıma Kısıtları ve Dağıtım Stratejisi Parametrelerine Göre Çalışmaların Sınıflandırılması

[1, 2, 5, 10, 14, 15, 17, 18, 20, 21, 25, 27, 32, 36, 39, 40, 47, 48, 54-56, 58, 59, 64, 67, 68, 71, 75, 77, 80-86, 91-94] çalışmalarında homojen tipli, kapasite kısıtlı, limitli sayıda araç kullanılmıştır, araçlar aynı müşteriye bir kez uğramaktadır, talep miktarları bölünmemektedir ve sabittir.

[6, 31, 34, 41, 61, 62, 69, 90, 95-97] çalışmalarında heterojen tipli, kapasite kısıtlı, limitli sayıda araç kullanılmıştır, talepler miktarları sabittir.

[3, 7, 30, 37, 74] çalışmalarında homojen tipli, limitsiz sayıda, kapasite kısıtlı araç kullanılmıştır, talep miktarları sabittir. Bu çalışmalardan [37,74] çalışmalarında talepler bölünmemektedir.

[3, 4, 7-9, 11, 16, 19, 24, 26, 49, 51, 53, 63, 70, 73, 76] çalışmalarında müşteri talepleri bölünmektedir.

[73] çalışmasında heterojen tipli, limitsiz sayıda, kapasite kısıtlı araç kullanılmıştır, mesafe kısıtı mevcuttur, müşteri talepleri sabittir.

[12, 23, 30, 39, 52, 62, 63, 92, 94] çalışmalarında dağıtım yöntemi olarak çapraz sevkiyat yer almaktadır. [10, 13, 15, 16, 19, 21, 22, 24, 26, 41, 42, 51, 55, 65, 84, 85] çalışmalarında milk-run dağıtım yöntemi incelenmektedir. [9, 11, 25, 67, 71, 93] çalışmalarında dağıtım yöntemi olarak direkt ve çapraz sevkiyat yöntemi incelenmiştir. [32, 36, 47, 53, 54, 56, 69, 70, 77, 83, 86, 91, 95, 96] çalışmalarında dağıtım yöntemi olarak çapraz sevkiyat merkezinden milk-run yöntemini incelemişlerdir. [8, 97] çalışmasında direkt taşıma ve milk-run uygulamasını birlikte ele almıştır. [7, 49, 64] çalışmalarında direkt taşıma, milk-run ve çapraz sevkiyat yöntemlerinin üçünü birlikte ele almışlardır.

Çalışmaların araç, taşıma ve dağıtım yöntemleri parametreleri açısından sınıflandırılması Tablo 1.2’de gösterilmiştir.

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[1]	x	x	x					
[2]	x	x	x				x	
[3]	x		x	x				
[4]	x	x	x	x				
[5]	x	x	x				x	
[6]		x	x		x			
[7]	x		x	x				Direkt sevkiyat, çapraz sevkiyat, milkrun
[8]		x	x	x	x			
[9]	x	x	x	x				Direkt sevkiyat, çapraz sevkiyat
[10]	x	x	x		x		x	Milk-run
[11]	x	x	x	x	x	x		Direkt sevkiyat, çapraz sevkiyat
[12]	x	x	x			x	x	Çapraz sevkiyat
[13]	x	x	x				x	Milk-run
[14]	x	x	x			x	x	Direkt sevkiyat, hub sevkiyatı
[15]	x	x	x					Milk-run
[16]		x	x	x		x	x	Milk-run

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[17]	x	x	x		x	x		
[18]	x	x	x			x		
[19]	x	x	x	x			x	Milk-run
[20]	x	x	x		x	x		
[21]	x	x	x		x			Milk-run
[22]	x	x	x					Milk Run
[23]	x	x	x				x	Çapraz sevkiyat
[24]	x	x	x	x			x	Milk Run
[25]	x	x	x				x	Direkt sevkiyat, Çapraz sevkiyat
[26]	x	x	x	x			x	Milk-run
[27]	x	x	x					
[28]	x	x	x				x	Çapraz sevkiyat, milk-run
[29]	x	x	x	x				Çapraz sevkiyat, Direkt Sevkiyat
[30]	x				x	x		Çapraz Sevkiyat
[31]		x	x		x	x		
[32]	x	x	x					Çapraz sevkiyat, milk-run
[33]	x	x	x	x		x	x	Çapraz sevkiyat, milk-run

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[34]		x	x			x	x	
[35]	x	x	x	x			x	Çapraz sevkiyat, milk-run
[36]	x	x	x			x	x	Çapraz sevkiyat, milk-run-
[37]	x		x		x		x	
[38]			x			x		
[39]	x	x	x		x	x		Çapraz Sevkiyat
[40]		x	x					
[41]	x	x	x		x			Milk-run
[42]		x	x				x	Milk-run
[43]		x	x					Çapraz sevkiyat, milk-run
[44]		x	x				x	Çapraz sevkiyat, milk-run
[45]		x	x				x	Çapraz sevkiyat, milk-run
[46]		x	x				x	Çapraz sevkiyat, milk-run
[47]		x	x		x	x		Çapraz sevkiyat, milk-run

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[48]		x	x					
[49]		x	x	x			x	Direkt sevkiyat, çapraz sevkiyat, milk-run
[50]		x	x				x	
[51]	x	x		x			x	Milk-run
[52]	x	x	x					Çapraz sevkiyat
[53]	x	x	x	x			x	Çapraz sevkiyat, milk-run
[54]	x	x	x				x	Çapraz sevkiyat, milk-run
[55]	x	x	x					Milk-run
[56]	x	x	x		x		x	Çapraz sevkiyat
[57]	x	x	x				x	Çapraz sevkiyat, milk-run
[58]	x	x	x					
[59]	x	x	x		x	x	x	
[60]	x	x	x		x		x	Direkt sevkiyat, çapraz sevkiyat, milk-run

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[61]		x	x			x		
[62]		x	x		x	x		Çapraz sevkiyat
[63]		x	x	x				Çapraz sevkiyat
[64]	x	x	x		x			Direkt sevkiyat, çapraz sevkiyat, milk-run
[65]	x	x	x				x	Milk-run
[66]	x	x	x		x			Direkt sevkiyat, çapraz sevkiyat, milk-run
[67]	x	x	x		x			Direkt sevkiyat, çapraz sevkiyat
[68]	x	x	x		x	x	x	
[69]		x	x					Çapraz sevkiyat
[70]		x	x	x			x	Çapraz sevkiyat
[71]	x	x	x		x	x		Direkt sevkiyat, çapraz sevkiyat

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[72]		x	x		x	x	x	
[73]		x	x	x				
[74]	x		x			x		
[75]	x	x	x		x	x		
[76]		x	x	x		x		
[77]	x	x	x		x			Çapraz sevkiyat, milk-run
[78]	x	x	x		x			Çapraz sevkiyat, milk-run
[79]	x	x	x				x	Çapraz sevkiyat, milk-run
[80]	x	x	x					
[81]	x	x	x			x	x	
[82]	x	x	x					
[83]	x	x	x		x		x	Çapraz sevkiyat, milk-run
[84]	x	x	x			X	x	Milk-run
[85]	x	x	x					Milk-run
[86]	x	x	x				x	Çapraz sevkiyat, milk-run
[87]		x	x				x	Çapraz sevkiyat, milk-run

Tablo 1.2 Araç, Taşıma Kısıtları ve Dağıtım Yöntemi Açısından Sınıflandırma
(devamı)

Yayın	Araç Kısıtları			Taşıma Kısıtları				Dağıtım Stratejisi
	Homojen	Sayı Limiti	Kapasite Limiti	Bölünebilir	Zaman	Mesafe	Zaman Penceresi	
[88]	x	x	x				x	Çapraz sevkiyat, milk-run
[89]	x	x	x				x	Çapraz sevkiyat, milk-run
[90]		x	x		x	x	x	Direkt sevkiyat, milk-run
[91]	x	x	x				x	Çapraz sevkiyat, milk-run
[92]	x	x	x				x	Çapraz sevkiyat
[93]	x	x	x				x	Direkt sevkiyat, çapraz sevkiyat
[94]	x	x	x		x		x	Çapraz sevkiyat
[95]		x	x				x	Çapraz sevkiyat, milk-run
[96]		x	x				x	Çapraz sevkiyat, milk-run
[97]		x	x				x	Direkt sevkiyat, Milk-run

1.1.2 Amaç ve Çözüm Yöntemleri Parametrelerine Göre Çalışmaların Sınıflandırılması

[1,4,26,41,47,85,90,93] çalışmalarında kesin çözüm yöntemlerinden yararlanmışlardır. [6,9,11,30,32,48,58,60,73] çalışmalarında kesin ve sezgisel yöntemlerden yararlanmışlardır. [2,3] çalışmalarında ayrışma yaklaşımını (Decomposition Approach) kullanarak sezgisel yöntemlerden yararlanmışlardır. [7,42] çalışmalarında yerel arama sezgiselini kullanmışlardır. [17,23,70] çalışmalarında tabu arama sezgiselini kullanmışlardır. [31,50,65,67,69,92] çalışmalarında tavlama benzetimini kullanmışlardır. [53], tavlama benzetimi (simulated annealing) ve karınca algoritmasını birlikte uygulamıştır. [55,84] geliştirilmiş tasarruf algoritmasını kullanmışlardır. [5,18,20,22,27,34,38,40,61,64,68,74,76,82,86,95] çalışmalarında genetik algorithmadan faydalanmışlardır. [18,25,51,76] çalışmalarında karınca algoritmasından faydalanmışlardır. [77], çalışmasında yapay arı algoritması kullanmıştır. [91], çalışmasında parçacık sürü optimizasyonundan faydalanmışlardır. [94], çalışmasında özelleştirilmiş algoritma (demon algoritma) kullanmışlardır. [96], çalışmasında ayarlanabilir sağlam optimizasyon (adjustable robust optimization) algoritmasını geliştirmişlerdir. [97], ilk/en uygun strateji (first/best-fit strategy), ve evrim stratejisinin gruplandırılması (grouping evolution strategy) algoritmalarını geliştirmişlerdir.

[1-4,6,9-11,13,15,17,19,21,24,30,32,34,41,42,47,51,53-55,58,61,62,70,72,74,77,80,82] çalışmalarında amaç fonksiyonu olarak maliyetin minimizasyonunu ele almışlardır. [8,12,14,18,22,25,31,36,37,48,49,67,68,86,90-93,97] çalışmalarında amaç fonksiyonu olarak dağıtım maliyetinin minimizasyonunu kullanmışlardır. [5,16,23,26,50,59,81,83,84,94] ise çalışmalarında amaç fonksiyonu olarak toplam mesafenin minimizasyonunu ele almıştır. [20], çalışmasında toplam kullanılan araçların ve toplam rotalama maliyetinin minimizasyonunu ele almıştır. [60], çalışmasında toplam maliyet ve zamanın minimizasyonunu ele almıştır. [64], çalışmasında zaman, dağıtım maliyeti ve dağıtım sayısının minimizasyonunu amaç fonksiyonu olarak kullanmışlardır.

[76], çalışmasında dağıtım maliyetinin minimizasyonunu ve müşteri memnuniyetinin maksimizasyonunu ele almıştır. [75,95] çalışmasında toplam karın maksimizasyonunu ele almışlardır. [85], çalışmasında toplam taşınan miktarın ve toplam mesafenin minimizasyonunu amaç fonksiyonu olarak kullanmıştır.

Çalışmaların amaç ve çözüm yöntemleri parametreleri açısından sınıflandırılması Tablo 1.3'te gösterilmiştir.

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[1]	Min. Toplam maliyet	x		Dal ve Sınır Algoritması
[2]	Min. Toplam maliyet		x	Ayrışma Yaklaşımı (Decomposition Approach)
[3]	Min. Toplam maliyet		x	Ayrışma Yaklaşımı (Decomposition Approach)
[4]	Min. Toplam maliyet	x		Tamsayılı ve Karışık Tamsayılı Doğrusal Programlama
[5]	Min. Toplam mesafe		x	Hibrit Genetik Algoritma
[6]	Min Toplam Maliyet	x	x	Karışık Tamsayılı Programlama, Rotalama Sezgiseli
[7]	Min. Toplam mesafe		x	Yerel Arama Sezgiseli
[8]	Min. Dağıtım maliyeti		x	En Kısa Yol ve Rassal Sıralı Eşleştirme sezgisel Algoritması (Shortest Path Heuristics and Randomized Sequential Matching Algorithm (RSMA))
[9]	Min. Toplam maliyet	x	x	Doğrusal Olmayan Tamsayılı Programlama ve Geliştirilmiş Sezgisel Algoritma
[10]	Min. Toplam maliyet		x	Tabu Arama Tabanlı Sezgisel Algoritma
[11]	Min. Toplam maliyet	x	x	Açgözlü Sezgisel, Lagrange Gevşetme Sezgiseli ve Dal-Sınır Algoritması
[12]	Min. Dağıtım maliyeti		x	Tabu Arama Tabanlı Sezgisel

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[13]	Min. Toplam maliyet		x	Yerel Arama ile Ekleme Tabanlı Yapısal Sezgisel ve Tabu Arama Sezgiseli
[14]	Min. Dağıtım maliyeti		x	2-opt Yerel Arama ile Paralel Tasarruf Sezgiseli ve Evrim Tabanlı Sezgisel
[15]	Min. Toplam maliyet		x	Dağılım Araması (Scatter Search)
[16]	Min. Toplam mesafe		x	En Uygun Algoritma (Best Fit Algorithm), Değiş Tokuş Algortitması (2-Exchange Algorithm), İlk Uyan Algoritma (First Fit algorithm) ve Ekleme Algoritması (Insertion Algorithm)
[17]	Min Toplam Maliyet		x	Tabu Arama Sezgiseli
[18]	Min. Dağıtım maliyeti		x	Karınca ve Genetik Algoritma
[19]	Min. Toplam maliyet		x	İki Aşamalı Sezgisel: Tabu Arama ve Karışık Tamsayılı Programlama
[20]	(1) Min. Toplam kullanılan araçlar (2) Min toplam rotalama maliyet		x	Genetik Algoritma
[21]	Min. Toplam maliyet		x	Sütun Yaratma (Column Generation) Tabanlı Sezgisel
[22]	Min. Dağıtım maliyeti		x	Genetik Algoritma
[23]	Min. Toplam mesafe		x	Tabu Arama Sezgiseli
[24]	Min. Toplam maliyet		x	Sağlam Doğrusal Optimizasyon (Robust Linear optimization) ve Parçacık Sürü Optimizasyonu

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[25]	Min. Dağıtım maliyeti		x	Karınca Algoritması
[26]	Min. Toplam mesafe	x		Sütun Yaratma (Column Generation)
[27]	Min. Dağıtım maliyeti		x	Genetik Algoritma
[28]	Min. Toplam maliyet		x	Tabu Arama (Tabu search)
[29]	Min. Dağıtım maliyeti	x	x	Karışık tam sayılı programlama, İlk basit sezgisel (Simple Primal Heuristic)
[30]	Min. Toplam maliyet	x	x	Karma Tamsayılı Doğrusal Olmayan Programlama ve Tavlama Benzetimi (Simulated Annealing)
[31]	Min. Dağıtım maliyeti		x	Tavlama Benzetimi (Simulated Annealing)
[32]	Min. Toplam maliyet	x	x	Dal ve Fiyat, Dal ve Sınır Algoritması
[33]	Min. Dağıtım maliyeti		x	Süpürme Algoritması
[34]	Min. Toplam maliyet		x	Adaptif Çeşitlilik Kontrolü ile Hibrit Genetik Arama (Hybrid Genetic Search with Adaptive Diversity Control)
[35]	Min. Toplam maliyet	x		Karışık Tamsayılı Doğrusal Programlama
[36]	Min. Dağıtım maliyeti		x	Pareto Tabanlı Hibrit Algoritma
[37]	Min. Dağıtım maliyeti		x	Memetik Algoritma
[38]	Min. Dağıtım maliyeti		x	Genetik Algoritma Tabanlı Populasyon Sezgiseli
[39]	Min. Dağıtım maliyeti		x	İki Aşamalı Hibrit Tavlama Benzetimi Sezgiseli
[40]	Min. Toplam maliyet		x	Genetik Algoritma ve Yerel Yaklaşım Tabanlı Sezgisel
[41]	Min. Toplam maliyet	x		Karma Tamsayılı Doğrusal Programlama
[42]	Min. Toplam maliyet		x	Yerel Arama Sezgiseli

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[43]	Min. Toplam maliyet	x	x	
[44]	Min. Dağıtım maliyeti		x	Dal ve fiyat algoritması (Branch and Price) ve Doğrusal programlama tabanlı dal ve sınır algoritması (Linear Programming based branch and bound)
[45]	Min. Seyahat mesafesi			Adaptif çoklu yeniden başlatma tabu arama (adaptive multi restart tabu search)
[46]	Min. Dağıtım maliyeti	x	x	Karışık Tamsayılı Doğrusal Programlama, Süpürme Algoritması
[47]	Min. Toplam maliyet	x		Karma Tamsayılı Doğrusal Programlama
[48]	Min. Dağıtım maliyeti	x	x	Dal ve Kesim Algoritması ve Hibrit Metasezgiseli
[49]	Min. Dağıtım maliyeti		x	Harmoni Arama ve Tavlama Benzetimli Hibrit Sezgiseli
[50]	Min. Toplam mesafe		x	Tavlama Benzetimi
[51]	Min. Toplam maliyet		x	Karınca Algoritması
[52]	Min. Dağıtım maliyeti		x	Süpürme Sezgisel Algoritması
[53]	Min. Toplam maliyet		x	Tavlama Benzetimi ve Karınca Algoritması
[54]	Min. Toplam maliyet		x	İteratif Yerel Arama Metasezgiseli
[55]	Min. Toplam maliyet		x	Geliştirilmiş Tasarruf Algoritması
[56]	Min. Toplam maliyet	x		Bulanık Olasılıklı Stokastik Karma Tamsayılı Programlama

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[57]	Min. Dağıtım maliyeti	x		Bulanık karışık tamsayılı doğrusal programlama
[58]	Min Toplam Maliyet	x	x	Karma Tamsayılı Programlama ve GRASP, İteratif Yerel Arama ve Tavlama Benzetimli Hibrit Metasezgisel
[59]	Min. Toplam mesafe		x	Tabu Arama ve Tavlama Benzetimli Hibrit Meta-Sezgisel
[60]	Min. Toplam maliyet ve min. Zaman	x	x	Kümeleme (Bi-clustering) ve Genetik Algoritma
[61]	Min Toplam Maliyet		x	Hibrit Genetik Algoritma
[62]	Min. Toplam maliyet		x	Hibrit Genetik Algoritma
[63]	Min. Toplam maliyet		x	Süpürme Tabanlı Algoritma
[64]	Min. Zaman, Min. Dağıtım maliyeti, Min. Dağıtım sayısı		x	Baskın Olmayan Sıralamalı Genetik Algoritma (Non-Dominated Sorting Genetic Algorithm) ve Çok Amaçlı Parçacık Sürü Optimizasyonu (The Multi-objective Particle Swarm Optimization)
[65]	Min. Toplam maliyet		x	Tavlama Benzetimi
[66]	Min. Dağıtım maliyeti,	x	x	Karışık tam sayılı doğrusal programlama, Sütun üretimi (Column Generation), Dal fiyat algoritması (+Branch and Pprice
[67]	Min. Dağıtım maliyeti		x	Tavlama Benzetimi

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[68]	Min. Dağıtım maliyeti		x	Hibrit Genetik Algoritma ve Sezgisel Algoritma
[69]	Min. Toplam maliyet		x	Tavlama Benzetimi
[70]	Min. Toplam maliyet		x	Tabu Arama
[71]	Min. Toplam maliyet		x	Biyocoğrafya Tabanlı Optimizasyon (Biogeography-Based Optimization) (BBO) Algoritması
[72]	Min. Toplam maliyet		x	Hibrit Evrimsel Arama Algoritması (Hybridevolutionary Search Algorithm (HESA))
[73]	Mak. Toplam Kar	x	x	Geliştirilmiş Dal-sınır Algoritması ve Aç Gözlü Sezgisel
[74]	Min. Toplam maliyet		x	Hibrit Genetik Algoritma
[75]	Mak. Tedarik zinciri karı		x	Genetik Algoritma
[76]	Min. Dağıtım maliyeti ve mak. Müşteri memnuniyeti		x	Genetik Algoritma Tabanlı Karınca Algoritması
[77]	Min. Toplam maliyet		x	Adaptif Hafızalı Yapay Arı Algoritması (An Adaptive Memory Artificial Bee Colony (AMABC) Algorithm)
[78]	Min. Seyahat mesafesi	x	x	Karışık tamsayı programlama, Adaptif hafıza programlama (Adaptive Memory Programming) ve Tabu Arama

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[79]	Min. Dağıtım maliyeti	x	x	Tamsayılı olmayan doğrusal programlama, Adaptif büyük komşuluk arama (Adaptive large neighborhood search)
[80]	Min. Toplam maliyet		x	Değişken Komşu Arama (Variable Neighborhood Search) ve Tavlama Benzetimli (Simulated Annealing) Hibrit Sezgisel Algoritma
[81]	Min. Toplam mesafe		x	Yapay Arı Algoritması ve Tabu Arama Algoritmasını Birleştiren Hibrit Yaklaşım
[82]	Min. Toplam Maliyet		x	Genetik Algoritma
[83]	Min. Toplam mesafe		x	Adaptif Büyük Komşuluk Arama Algoritması (Adaptive Large Neighborhood Search(ALNS))
[84]	Min. Toplam mesafe		x	Geliştirilmiş Tasarruf Algoritması
[85]	Min. Toplam taşınan miktar ve toplam mesafe	x		Çok Amaçlı Programlama
[86]	Min. Dağıtım maliyeti	x	x	Karışık Tamsayılı Doğrusal Programlama ve Genetik Algoritma
[87]	Min. Dağıtım maliyeti	x	x	Karışık Tamsayılı Doğrusal Programlama, Tavlama Benzetimi (Simulated Annealing) ve Tabu arama (Tabu search)
[88]	Min. Dağıtım maliyeti	x	x	Karışık tamsayılı programlama, Büyük komşuluk arama (Large neighborhood search)

Tablo 1.3 Amaç ve Çözüm Yöntemleri Açısından Sınıflandırma (devamı)

Yayın	Amaç	Çözüm Yöntemi		
		Kesin	Sezgisel	Yöntem Adı
[89]	Min. Toplam Maliyet	x	x	Karışık Tamsayılı Doğrusal Programlama, Kendini uyarlayan emperyalist rekabet algoritması (Self adaptive imperialist competitive algorithm)
[90]	Min. Dağıtım maliyeti	x		Karışık Tamsayılı Doğrusal Programlama
[91]	Min. Dağıtım maliyeti	x	x	Karışık Tamsayılı Doğrusal Programlama ve Parçacık sürü Optimizasyonu
[92]	Min. Dağıtım maliyeti		x	Tavlama Benzetimi
[93]	Min. Dağıtım maliyeti (dağıtım + envanter)	x		Karışık tam sayılı doğrusal olmayan programlama
[94]	Min. Toplam Mesafe		x	Demon Algoritması
[95]	Max. Toplam Kar	x	x	Karışık tam sayılı programlama, Değişken komşu arama ve genetik algoritma
[96]	Min. Toplam maliyet, Max. Toplam dağıtılan ürün ağırlığı	x	x	Karışık tam sayılı programlama, Ayarlanabilir sağlam optimizasyon (adjustable robust optimization)
[97]	Min. Dağıtım maliyeti	x	x	Karışık tam sayılı programlama, ilk/en uygun strateji (first/best-fit strategy), evrim stratejisinin gruplandırılması (grouping evolution strategy)

Tablo 1.4'te literatür araştırması kapsamında incelenen çalışmalar bu tez çalışmasının konusu olan çapraz sevkiyat, direk taşıma, milk-run gibi dağıtım stratejileri kullanan çalışmalar için özetlenmiştir. Literatür taramasından elde edilen bilgilere göre, aşağıdaki sonuçlar çıkarılabilir:

- “Dağıtım stratejisi seçimi” dağıtım için kritik bir konu olmasına rağmen, bununla ilgili sadece birkaç çalışma vardır. Tablo 1.4'ten görüldüğü gibi, sadece birkaç çalışma bu konu üzerinde odaklanmıştır.
- “Doğrudan Sevkiyat”, “Milk-run” ve “Çapraz Sevkiyat” olmak üzere üç dağıtım stratejisinden oluşan karma dağıtım sistemi literatürde az çalışılmıştır. Çalışmaların çoğu, “Çapraz Sevkiyat” ve “Milk-run” olmak üzere iki dağıtım stratejisini inceleyen araç rotalama problemlerini ele almıştır.
- “Çapraz Sevkiyat”, “Doğrudan sevkiyat” ve “Milk-run” dağıtım stratejilerinden oluşan çalışmalar için “dağıtım seçimi” ve “araç rotalama” için geliştirilen hibrit sezgisel çözümler, genetik algoritma, tavlama benzetimi ve parçacık sürü optimizasyonu gibi algoritmalara odaklanmıştır. Bu üç teslimat modundan oluşan karma dağıtım sisteminde eşzamanlı olarak “araç rotalama” ve “dağıtım stratejisi seçimi” sorunlarını çözmek için “Clarke ve Wright’ın tasarruf algoritması” (CWS) ve “Genetik algoritma” (GA) kullanan herhangi bir sezgisel algoritma yoktur.
- Diğer bir bulgu ise, “Çapraz Sevkiyat”, “Milk-run” ve “Doğrudan Sevkiyat” olmak üzere üç teslimat modundan oluşan “karma dağıtım sistemi problemine” odaklanan çalışmalarda “Homojen ve sınırlı sayıda araç” kullanılmaktadır. Bugün birçok işletme üçüncü parti lojistik (3PL) sağlayıcısının kullanımını maliyetleri düşürme ve müşteri hizmetlerini iyileştirme seçeneği olarak görmektedir. Bu çalışmada üçüncü parti lojistik (3PL) sağlayıcısı farklı bir yaklaşım olarak dağıtımda kullanılmıştır ve araç filosu “Heterojen” ve “Sınırsız sayıda” belirlenmiştir. Bu nedenle, gerekli sayıda araç gerektiğinde kullanıma hazırdır.

Tablo 1.4 Dağıtım Stratejileri Kapsamında Çalışmaların Özetlenmesi

Yayın	Dağıtım Strateji Seçimi	Genetik algoritma (GA) ve Tasarruf algoritma (CWS)			Dağıtım Karakteristikleri									Dağıtım Stratejileri			
		GA	CWS	GA+CWS	Homojen Araç	Heterojen Araç	Sınırsız Araç	Sınırlı Araç	Kapasiteli Araçlar	Bölünmüş Teslimatlar	Seyahat Mesafesi	Seyahat Zamanı	Zaman Penceresi	Çapraz Sevkiyat	Direkt Sevkiyat	Milk-run	Grupaj Servisi
[7]			x		x		x		x					x	x	x	
[9]					x			x	x	x				x	x		
[11]	x				x			x	x	x				x	x		
[12]					x			x	x		x	x		x		x	
[14]	x				x			x	x					x	x	x	x
[22]		x			x			x	x							x	
[23]					x			x	x				x	x		x	
[25]					x			x	x					x	x		
[28]					x			x	x				x	x		x	
[29]	x				x			x	x	x				x	x		
[30]					x		x		x	x				x	x		
[32]					x			x	x					x		x	
[33]					x			x	x	x	x		x	x		x	
[35]					x			x	x	x			x	x		x	
[39]					x			x	x		x	x		x		x	
[43]					x			x	x					x		x	
[44]					x			x	x				x	x		x	
[45]					x			x	x				x	x		x	
[46]					x			x	x				x	x		x	
[47]					x			x	x				x	x		x	
[49]					x			x	x					x	x	x	
[52]					x			x	x			x		x		x	
[53]					x			x	x	x			x	x		x	
[54]					x			x	x				x	x		x	
[55]					x			x	x							x	
[57]					x			x	x				x	x		x	

Tablo 1.4 Dağıtım Stratejileri Kapsamında Çalışmaların Özetlenmesi (devamı)

Yayın	Dağıtım Strateji Seçimi	Genetik algoritma (GA) ve Tasarruf algoritma (CWS)			Dağıtım Karakteristikleri										Dağıtım Stratejileri			
		GA	CWS	GA+CWS	Homojen Araç	Heterojen Araç	Sınırsız Araç	Sınırlı Araç	Kapasiteli Araçlar	Bölünmüş Teslimatlar	Seyahat Mesafesi	Seyahat Zamanı	Zaman Penceresi	Çapraz Sevkiyat	Direkt Sevkiyat	Milk-run	Grupaj Servisi	
[60]	x	x			x			x	x					x	x	x		
[62]		x				x		x	x				x	x		x		
[63]						x		x	x			x		x		x		
[64]	x	x			x			x	x					x	x	x		
[66]					x			x	x			x		x	x	x		
[67]					x			x	x				x	x		x		
[71]					x			x	x		x	x		x	x	x		
[77]					x			x	x			x		x		x		
[79]					x			x	x				x	x		x		
[78]					x			x	x	x		x		x		x		
[84]			x		x			x	x		x		x			x		
[86]		x			x			x	x				x	x		x		
[87]						x		x	x	x			x	x		x		
[88]					x			x	x				x	x		x		
[89]					x			x	x				x	x		x		
[90]	x					x		x	x				x	x	x	x		
[91]		x			x			x	x				x	x		x		
[93]	x				x			x	x				x	x	x			
[95]		x				x		x	x				x	x		x		
[96]						x		x	x				x	x		x		
[97]	x					x		x	x				x		x	x		
Bu çalışma	x	x	x	x		x	x		x		x	x		x	x	x		

1.2 Tezin Amacı

Tedarik zinciri dağıtım ağı planlaması, tedarik zincirinin kritik bir sürecidir. Operasyonel ve taktik seviyede kararlar almayı gerektirmektedir. Yapılan planlama, işletmelerin kârlılığını, pazarda rekabet edebilirliğini ve müşteri memnuniyetini önemli derecede etkilemektedir.

Bu çalışma, tedarik zinciri dağıtım ağı planlamasında direkt dağıtım, çapraz sevkiyat ve milk-run olmak üzere farklı dağıtım stratejilerini birlikte ele alan gerçek hayat tedarik zinciri dağıtım problemlerine çözüm sunmayı amaçlamaktadır. Benzer problemlere çözüm oluşturmak için matematiksel model oluşturulmuş ve meta-sezgisel model geliştirilmiştir. Sunulan çözümle, en az maliyet ile müşterilere hizmet vermek amaçlanarak, en uygun dağıtım yöntemi ve araç tipi seçimi kararı alınabilmekte ve araç rotalama yapılabilmektedir.

1.3 Hipotez

Bu tez çalışmasında, geliştirilen algoritmanın optimale yakın sonuçlar vereceği ve karma dağıtım sistemi problemi için dağıtım maliyetlerini azaltmada uygulayıcılara yol göstereceği savunulmuştur. Çalışmanın hazırlanmasında literatür özeti bölümünde yer alan, literatür araştırmasından faydalanılmıştır.

Yürütülen çalışma şu yönüyle literatüre katkı sağlamaktadır: Literatürde “Çapraz Sevkiyat”, “Doğrudan Sevkiyat” ve “Milk-run” olmak üzere üç dağıtım modunu içeren karma dağıtım sistemine odaklanan az çalışma yer almaktadır. Bu çalışmada, karma bir dağıtım sistemi sorununu çözmek için dağıtım stratejisi seçimi ve araç rotalama hibrit algoritması olarak adlandırılmış “modifiye edilmiş” tasarruf algoritma tabanlı özelleştirilmiş genetik algoritma geliştirilmiştir. Bu algoritma, karma dağıtım sistemi problemini etkin bir şekilde çözmek için literatüre yeni bir hibrit çözüm sunmaktadır.

Tedarik zinciri yönetimi, işletmenin tedarikçileri ve müşterileri arasındaki karşılıklı malzeme ve bilgi akışının doğru zamanda, doğru yerde ve en düşük maliyetle sağlanmasıdır.

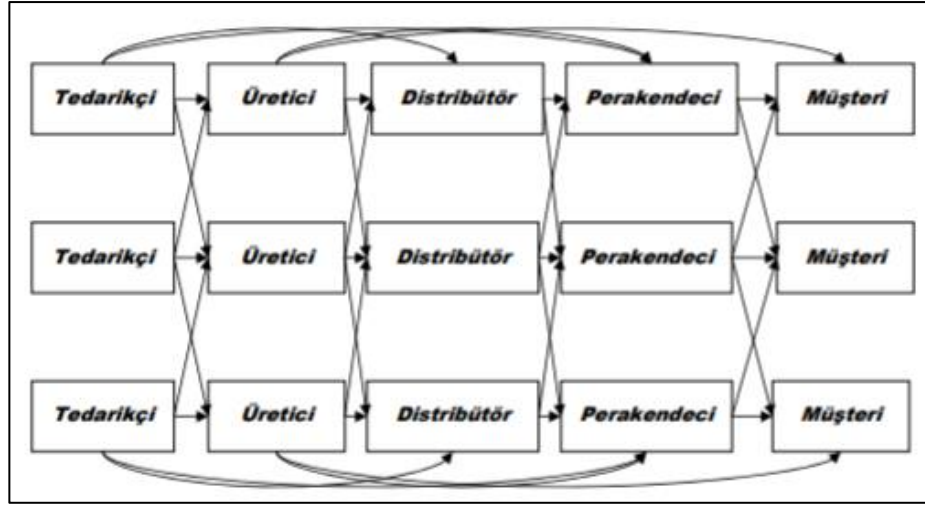
Tedarik zinciri yönetimi, organizasyonel üretkenliği ve kârlılığı arttıran ana rekabet stratejilerinden birisidir [98].

Tedarik yönetimi süreçlerini firmaların ihtiyaçlarına göre özelleştirip süreçleri verimli ve kârlı bir şekilde yönetmek ile azaltılmış maliyetler, artan verimlilik, kaynakların etkin kullanımı sağlanmakta, bunun sonucunda da kaliteli, zamanında üretim ve zamanında teslim gerçekleştirilebilmektedir. Bu sayede işletmeler yüksek müşteri memnuniyeti ve pazarda farklılık yaratarak; rakiplerine karşı güçlü rekabet imkânı sağlayabilmektedir.

2.1 Tedarik Zinciri ve Tedarik Zinciri Yönetimi Kavramı

Tedarik zinciri, hammaddenin temininden ürünün üretimine, tüketiciye ulaştırılmasına ve geri dönüşümüne kadar tüm süreçleri içeren tedarikçi, üretici, dağıtıcı, perakendeci ve lojistikçilerden oluşan bir organizasyondur.

[99]'e göre, “Tedarik zinciri, müşteri taleplerini karşılamak üzere, sadece üretici ve tedarikçiden oluşmayan, aynı zamanda lojistik hizmeti verenler, depolama hizmeti verenler, perakendeciler, müşteriler ve diğer tüm benzer aktörleri kapsayan bir zincirdir”. Şekil 2.1'de tedarik zincirinde yer alan aktörler görülmektedir.



Şekil 2.1 Tedarik Zinciri Aktörleri [99]

Tedarik zinciri yönetimi, zincir boyunca karşılıklı hareket eden bilgi, ürün ve para akışının entegre şekilde yönetilmesidir.

Tedarik zinciri yönetimi kısaca, son ürünlerin müşteriye ulaştırılmasını sağlayan, tüm faaliyetler ağıdır [100].

2.2 Tedarik Zinciri Yönetiminin Amaçları

Etkin bir tedarik zinciri yönetimi ile, işletmenin ulaşmayı hedeflediği amaçlardan bazıları aşağıda belirtilmiştir [100]:

- Kesintisiz malzeme, servis ve bilgi akışını gerçekleştirmek,
- Stok maliyetlerini ve kayıplarını en düşük seviyede tutmak,
- Ürün kalitesini korumak,
- Güvenilir tedarikçiler bulmak,
- Elde edilen hammadde, yarı mamul, parça ve servisi standart hale getirmek,
- Hammadde, yarı mamul ve hizmetleri en düşük maliyetle sağlamak,
- İşletmenin, pazarlık ve rekabet gücünü yükseltmek,
- En düşük yönetim gideri ile çalışmak.

Stratejik bir fonksiyon olarak tedarik zinciri yönetiminin amacı, işletmelerin uzun dönemli amaçlarının başarılmasında örgütsel faaliyetlerin desteklenmesi olarak belirtilebilir [101].

Tedarik Zinciri Yönetiminin temel faydası, zincirdeki işletmelerin tüm faaliyetlerini içerecek şekilde planlama yapması ve bu sayede hepsini ortak bir faydada buluşturmasıdır. Bu ortak fayda, günümüzün rekabet öncelikleri olarak kabul edilen kalite, hız, maliyet ve güvenilirlik konularında zincirin bütünü için yarar sağlanmasıdır [102].

2.3 Tedarik Zinciri Yönetiminin Temel Fonksiyonları

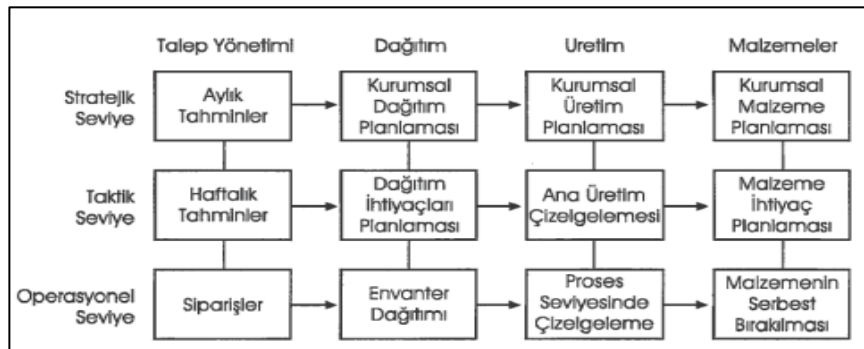
Tedarik zinciri yönetim fonksiyonları aşağıdaki üç seviyede planlamayı içermektedir:

- Stratejik seviye,
- Taktik seviye,
- Operasyonel seviye

Stratejik düzey, firmaya uzun dönemde etkisi bulunacak kararlarla ilgilidir. Bu kararlara örnek olarak fabrika yeri seçimi, depo yeri seçimi, depoların sayı ve kapasitelerinin belirlenmesi verilebilir.

Taktik düzey, her çeyrekte veya her yılda bir güncellenen kararları içerir. Bu kararlara örnek olarak üretim kararları, envanter politikası ve dağıtım stratejileri seçim kararları örnek olarak verilebilir.

Operasyonel düzey, günlük veya saatlik olarak yapılan kısa süreli planlamaları içerir. Bu kararlara örnek ulaşım zamanı tahmini, rota ve yükleme gibi günlük kararlar verilebilir. Şekil 2.2’de tedarik zinciri yönetim fonksiyonları ve seviyeleri gösterilmektedir.



Şekil 2.2 Tedarik Zinciri Yönetimi Fonksiyonları [103,104]

2.4 Tedarik Zinciri Yönetiminde Dağıtım Ağlarının Planlanması

Dağıtım ağlarının planlanması, stratejik planlamanın uygulandığı alanlardan biridir. Dağıtımda amaç, müşteri taleplerini verilen bir planlama ufku boyunca doğru miktarda ve doğru zamanda en az maliyetle karşılamaktır. Firmalar dağıtım kendileri gerçekleştirebilecekleri gibi dış kaynak kullanarak da sağlayabilirler. Dağıtım kararları, taşıma tipine, büyüklüğüne karar verme, yük birleştirme kararları, taşıyıcı rota tayini ve araç çizelgelemeyi içermektedir.

Dağıtım ağı planı, kârı ve müşteriye sunulan hizmeti en iyilerken açılacak dağıtım merkezlerinin sayısı, yerleri ve hangi müşterilere hizmet vereceklerinin belirlenmesinin yanı sıra, kullanılacak taşıma yöntemlerinin seçimi gibi teknik detayları da içermelidir [105].

Tedarik zincirinde bir ürünün tedarikçiden müşteriye ulaşması için, ürünün taşınması ve depolanmasına dağıtım denir. Dağıtım tedarik zincirinin aşamaları olan tedarikçi, imalatçı, dağıtımcı, perakendeci ve müşterilerden herhangi ikisi arasında gerçekleşebilmektedir. Dağıtım, bir firmanın toplam kârlılığı üzerinde önemli bir etkidir, çünkü hem tedarik zinciri maliyetini hem de müşteri deneyimini doğrudan etkiler [99].

Tedarik zincirinde dağıtım, tedarik zincirinin başlangıcından müşteriye ürünün ulaştırılmasındaki tüm ürün hareketlerini içermektedir. Dağıtım stratejileri olarak direkt dağıtım, çapraz sevkiyat ile dağıtım, milk-run ile dağıtım gibi farklı stratejiler bulunmaktadır. Tedarik zinciri dağıtım kararları hangi dağıtım türlerinin kullanılabileceğini, dağıtım türleri arasında kombinasyonun söz konusu olup olmayacağını belirlemektedir. Firmalar bu farklı dağıtım seçeneklerini karşılaştırırken, müşteri hizmetleri ve maliyetler üzerindeki etkisini değerlendirmelidir. Sevkiyat ve dağıtım noktalarının birbirinden uzak olmaları, doğrudan dağıtımı bazı durumlarda zorunlu hale getirebilmektedir.

Müşterileri yeterli bir seviyede memnun edecek mamul/hizmet teslimatı tedarik zincirinin en temel görevidir. Hammadde temininden nihai kullanıcıya kadarki lojistik faaliyetler ağı boyunca tüm faaliyetleri içerirken, bu faaliyetler sırasında hatalı planlamadan kaynaklı israflarda ortaya çıkmaktadır [106].

Yalınlık kavramı temel alınarak ileride verilecek yalın lojistik faaliyetleri ile israflar önlenirken, aynı zamanda etkili ve verimli bir mamul/hizmet akışı da sağlanmış olur [106].

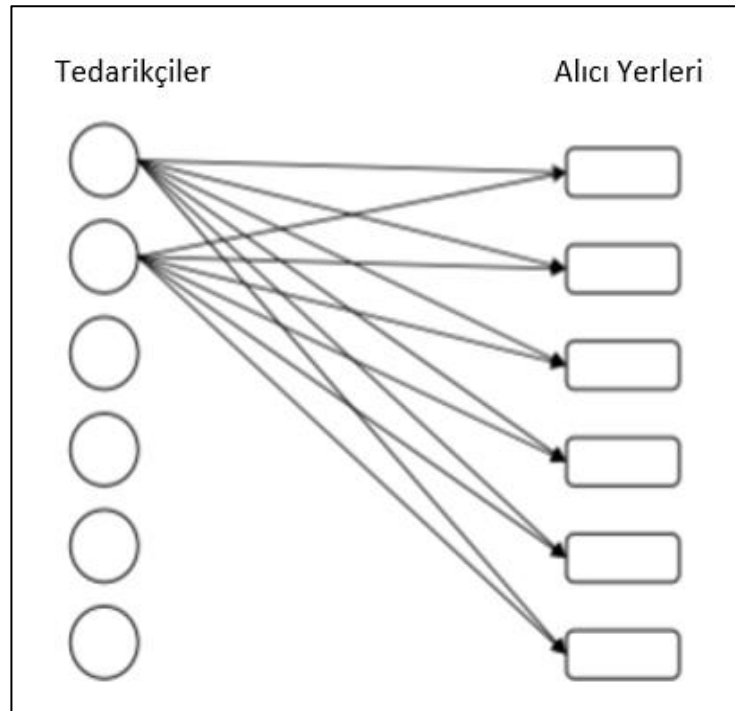
Bu bölümde, tedarik zinciri dağıtım ağlarının planlanmasında kullanılan dağıtım yöntemlerinden ve dış kaynak kullanımından bahsedilecektir.

2.4.1 Doğrudan Sevkiyat Ağı (Direkt Taşıma)

Doğrudan taşıma seçeneğinde, tüm taşımalar tedarikçilerden üreticilere veya üreticilerden perakendecilere doğrudan ve herhangi bir merkeze uğramadan yapılmaktadır (Şekil 2.3).

Doğrudan sevkiyat sisteminin avantajları, depoların elimine edilerek, operasyon ve koordinasyon kolaylığı sağlamasıdır.

Doğrudan sevkiyat ağı, alıcı bölgelerindeki talep yeterince büyükse; yani her bölgeye gönderilen sipariş miktarları kamyon yüküne yakınsa uygundur. Ancak alıcı bölgelerindeki talep küçük ise; doğrudan sevkiyat ağı yüksek maliyetlere neden olmaktadır [99] .



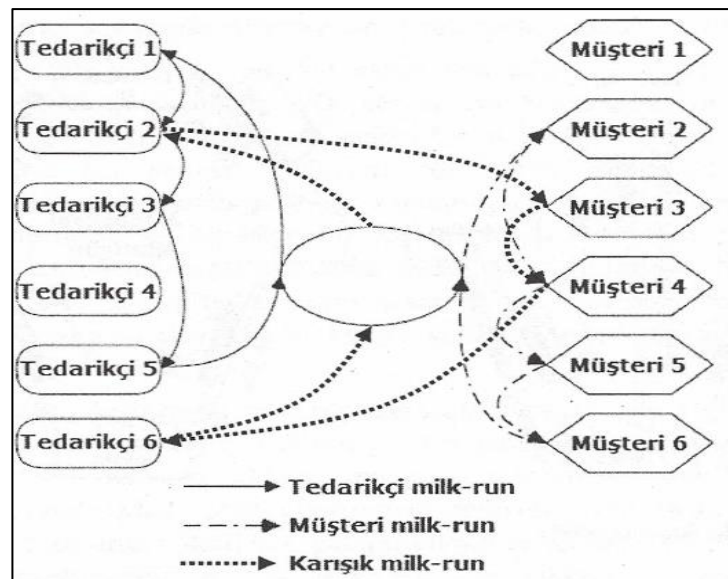
Şekil 2.3 Doğrudan Sevkiyat Ağı [99]

2.4.2 Milk-run (Döngüsel Sefer)

Milk-run, İngilizce bir terim olup; adını süt dağıtımından almaktadır. Bu uygulamada sütçüler bir önceki gün her kapıya kaç şişe süt bıraktıklarını bilmektedirler. Bu bilgi ile araçlarına sadece bir turda dağıtacakları kadar dolu süt almakta ve her kapıda buldukları boş şişeleri dolu süt şişesi ile değiştirmektedirler. Böylece her seferde sadece değişecek adette, satışı yapılacak kadar süt dağıtılmakta, tüketicilerin teker teker markete gitmelerinden doğan emek ve zaman harcaması önlenmektedir [107].

Milk-run, bir yalın lojistik uygulaması olup, tedarik zincirinde malzeme akışının sürekliliğini sağlamanın en iyi yöntemlerinden biridir.

Milk-run, tedarik noktasından yola çıkan aracın tüm teslimatlar tamamlandıktan sonra ters güzergâh izlenerek tekrar tedarik noktasına geri döndüğü kapalı döngü sistemlerdir. Döngüsel sefer olarak da bilinen milk-runda tedarikçi milk-run'ı, müşteri milk-run'ı ve karışık milk-run uygulamaları mevcuttur. Şekil 2.4'te milk-run çeşitleri gösterilmektedir. Tedarikçiler, dağıtım merkezi ve müşterilerden oluşan iki aşamalı bir tedarik zinciri yapısında, 3 çeşit milk-run sistemi kurulabilir [102]: Tedarikçiler arasında gerçekleşen tedarikçi milk-run'ı, Müşteriler arasında gerçekleşen müşteri milk-run'ı, Tedarikçi- dağıtım merkezi- müşteriler arasında geçen karışık milk-run.



Şekil 2.4 Milk-run Çeşitleri [16, 102]

Döngüsel Sefer, iş modeline ve müşterilerin coğrafi konumuna bağlı olarak, günlük veya haftalık olabilir [108].

Milk-runlar birden çok bölgeye yapılan sevkiyatları tek bir kamyon ile birleştirerek ulaştırma maliyetlerini düşürür [99].

Milk-run uygulamasının avantajlarından bazıları şunlardır [24].

- Stokları azaltır,
- Tahmin edilebilir ikmal miktarları ve zamanı sağlar,
- Stokların daha iyi görülebilmesini sağlar,
- Tedarikçi ile olan iletişimi geliştirir,
- Teslimat sürelerine olan güvensizlik azalır,
- Ürünlerin iç ve dış lojistik süreçlerini düzgünleştirir ve sürekliliğini sağlar,
- Fabrikalardaki depo boşlukları azalır,
- Malzemelerin ve parçaların üretime dâhil olması kolaylaşır,

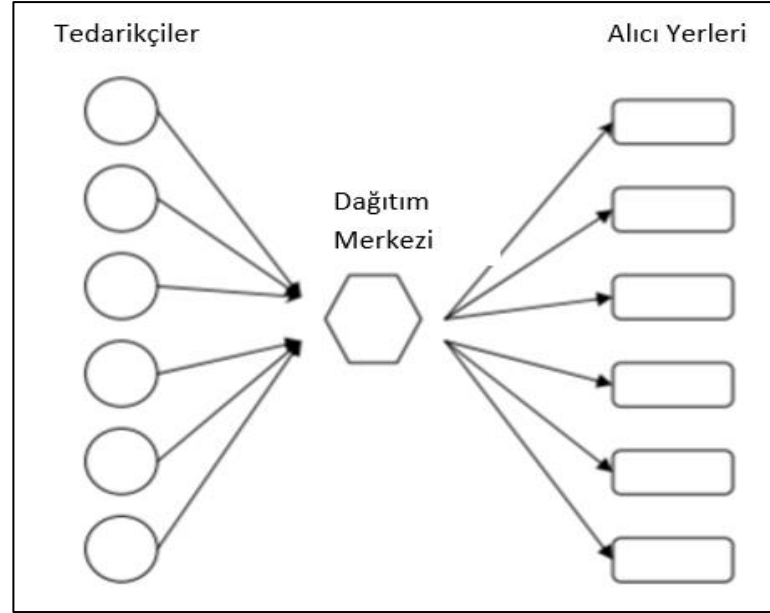
Taşıma ücretlerinin kontrolü, stok seviyelerinin azaltılması ve araç bakım maliyetlerinin düşürülmesi, sevkiyat araç kapasitelerinin verimli kullanılması, tedarik zinciri ve lojistik performansını arttırır.

2.4.3 Merkezi Dağıtım Merkezi ile Sevkiyat

Bu dağıtım şeklinde, tedarikçiler sevkiyatlarını alıcı bölgelerine doğrudan göndermezler. Teslimat noktaları coğrafik bölgelere göre ayrılır ve her bir bölge için dağıtım merkezi (DC: Distribution Center) kurulur. Tedarikçiler sevkiyatlarını dağıtım merkezine gönderirler ve dağıtım merkezi de her bir teslimat bölgesine uygun sevkiyatları iletir.

Tedarikçiler teslimat bölgelerinden uzakta konumlandığında ve bu durumda direkt dağıtım ile ulaştırma maliyetleri yüksek olduğundan, dağıtım merkezinin oluşu tedarik zinciri maliyetinin düşmesine yardım eder.

Dağıtım merkezi üzerinden sevkiyatta bir veya daha fazla çıkış noktasından, bir veya daha fazla varış noktasına hareket edecek ürünler birleştirilir (Şekil 2.5).



Şekil 2.5 Dağıtım Merkezi ile Sevkiyat [99]

Bu dağıtım şeklinde, dağıtım merkezinden stoklu çalışma ve çapraz sevkiyat ile taşıma olmak üzere iki tür çalışma yöntemi mevcuttur.

2.4.4 Dağıtım Merkezinden Yapılan Stoklu Taşıma

Bu yöntem, dağıtım merkezlerinde yüksek miktarda stok tutulmasını gerektirmektedir. Envanter açısından bakıldığında, dağıtım merkezinde stok tutulması yüksek talebi olan ürünler için müşteriye hızlı ürün ulaştırılmasını sağlayabilmektedir.

Tek bir müşteri siparişi için birden fazla gönderinin olması durumunda, üreticinin deposundan farklı olarak, dağıtım merkezi deposu müşteriye gönderilen siparişlerin tek bir gönderide paketlenmesini ve böylece nakliye ücretinin düşürülmesini sağlar [99].

Çapraz sevkiyatla karşılaştırıldığında, stok tutulması ve dağıtım merkezindeki operasyonel masrafların fazla olması dezavantaj olarak sayılabilir.

2.4.5 Çapraz Sevkiyat (Cross-docking) ile Taşıma

Lojistik uygulamalarında çapraz sevkiyat yapılması yeni bir uygulama olmakla beraber, yalın üretim, yalın düşünce çalışmalarının neticesinde ortaya çıkmıştır [109].

Çapraz sevkiyat, ürünün tedarikçiden alıcıya varıncaya kadar stoklanmadan hareket etmesini sağlar [110]. Dağıtım merkezi ürünü çapraz sevk ettiğinde, her bir tedarik kamyonu birçok tedarikçiden alıcı lokasyonuna ürün taşır. Çapraz sevkiyatın en büyük yararı, elde tutulacak çok az bir stoğa ihtiyaç duyulması ve tedarik zincirindeki ürün akışının daha hızlı olmasıdır. Çapraz sevkiyat ayrıca elleçleme masraflarını düşürür, çünkü ürünü deponun içinde ya da depo dışına hareket ettirmesi gerekmez [99].

Ürünü alıp, depoda tutup, etiketleyip göndermek yerine, çapraz sevkiyat, bir kamyonun bir palet veya konteynırı çeker doğrudan doktaki başka bir kamyonu taşır. Bu zamanı kısaltarak birçok değer katmayan adımı ortadan kaldırır. Dağıtım merkezlerine nelerin geldiği ile ilgili bilgiye sahip olunması ve kartonların veya paletlerin tedarikçiler tarafından önceden etiketlenmiş olması, çapraz sevkiyat yapmayı olanaklı kılan etkili yollardır [108].

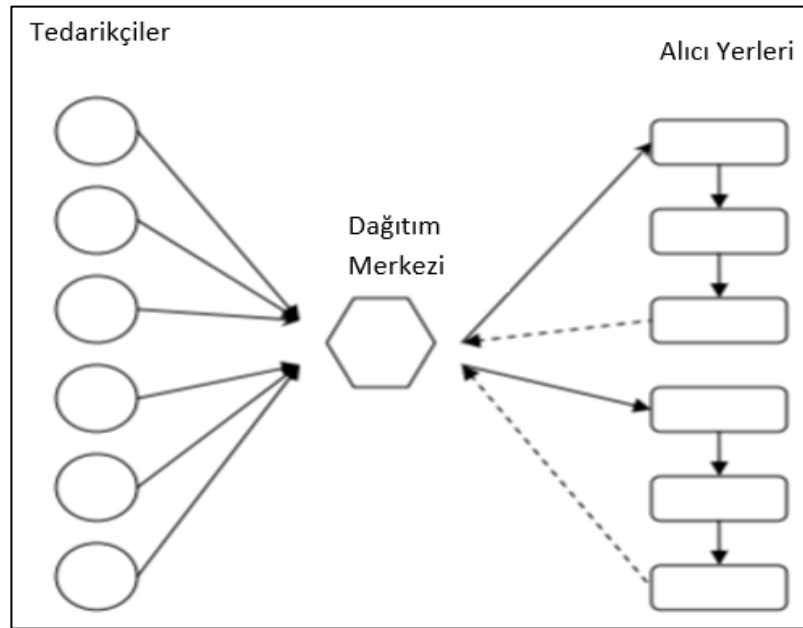
Çapraz sevkiyat (Cross-docking) sisteminin işletmeye sağlayacağı avantajlar şunlardır [111]:

- Maliyet avantajı sağlar: Çapraz sevkiyat ile elleçleme, işlem, stoklama, fire, hasar ve depo maliyetlerinden avantaj sağlanmaktadır. Ayrıca azaltılmış depolama miktarları kullanılan araçlar ve iş gücünden de tasarruf edilmesini sağlar.
- Çevrim süresini azaltır: Depolara ürünlerin girmesi ile kısa sürede işlenerek dağıtım merkezlerine sevk edilmesi, depolardaki bekleme süresini kısaltmıştır.
- Stok seviyesini azaltır: Klasik depo sistemine göre çapraz sevkiyatta stokların uzun süreler tutulmaması, stok seviyelerini kısa süreli tutulmasını gerektirir. Bu sayede de stoklarda ürünlerin kalma risklerini de ortadan kaldırmış olur.
- Daha az sayıda depolarla işlemler yürütülür: Klasik depolara göre çapraz sevkiyatta hızlı şekilde işlemlerin görülmesi, depo ihtiyacını ortadan kaldırmaktadır. Böylelikle daha az sayıda depo tesisi ile, aynı işlemler yapılabilir hâle gelmektedir.

- Müşteri hizmetlerini geliştirir: Sevkiyatların hızlı olarak gerçekleşmesi, tedarik bekleyen müşterilerin memnuniyetini sağlar. Ayrıca aynı sistemin tersten çalıştırılması, müşteri iadelerinin hızlıca ilgili birimlere ulaşımını sağlar.

2.4.6 Milk-Run Kullanarak Dağıtım Merkezi ile Sevkiyat

Milk-runlar küçük sevkiyatları birleştirerek, dağıtım maliyetlerini düşürürler. Çapraz sevkiyatın milk runlarla birlikte kullanımı, önemli derecede koordinasyon ve milk-runlara uygun rotalama ve çizelgeleme gerektirir [99]. Şekil 2.6'da milk-run kullanarak dağıtım merkezi ile sevkiyat gösterilmektedir.



Şekil 2.6 Milk-run Kullanarak Dağıtım Merkezi Sevkiyatı [99]

2.4.7 Uyarlanmış Ağlar (Tailored Networks)

Uyarlanmış dağıtım ağları, maliyetleri düşüren ve tedarik zincirinden alınan yanıtı iyileştiren seçeneklerin uygun bir kombinasyonudur. Bu taşıma çapraz sevkiyat, milk run ve doğrudan dağıtımın kombinasyonunu kullanır. Amaç, her durumda uygun seçenekleri kullanabilmektir. Bu dağıtım ağı yönetiminin karmaşıklığı yüksektir, çünkü her bir ürün ve perakende satış yeri için farklı sevkiyat prosedürleri kullanılır. Uyarlanmış bir ağın işlemesi, koordinasyona yardımcı olan bilgi altyapısına önemli derecede yatırımı gerektirmektedir [99].

Tablo 2.1’de, yukarıda anlatılan çeşitli dağıtım ağı stratejilerinin avantajları ve dezavantajları gösterilmektedir.

Tablo 2.1 Dağıtım Ağı Seçeneklerinin Avantaj ve Dezavantajları [99]

Ağ Yapısı	Avantajları	Dezavantajları
Doğrudan taşıma	1- Ara depolar yoktur. 2 Koordinasyon kolaydır.	Yüksek stok seviyeleri
Milk-run ile taşıma	1- Düşük dağıtım maliyetleri 2- Daha düşük stok seviyeleri	Koordinasyon karmaşıklığının artışı
Dağıtım merkezinden yapılan stoklu taşıma	Yük birleştirme sayesinde daha düşük taşıma maliyeti	1- Stok maliyetinin artışı 2- Dağıtım merkezindeki masrafların artışı
Dağıtım merkezlerinden çapraz sevkiyat ile taşıma	1-Az stok gereksinimi 2-Yük birleştirmeye sağlanan düşük taşıma maliyetleri	Koordinasyon karmaşıklığının artışı
Milk run kullanarak dağıtım merkezi ile taşıma	Küçük partiler için daha düşük taşıma maliyetleri	Daha fazla koordinasyon zorluğu
Uyarlanmış Ağlar	İhtiyaçları en iyi karşılayan taşıma seçeneği	En fazla koordinasyon zorluğu

2.5 Tedarik Zincirinde Dağıtımda Dış Kaynak Kullanımı

Tedarik zinciri yönetiminde karşılaşılabilecek en önemli risk, ürün yaşam döngülerinin kısılmasına bağlı olarak zincir üyelerinin yeterli esnekliği gösterememesidir [112]. Pazardaki rekabet ortamı, işletmeleri, uzmanlık alanları dışındaki faaliyetleri, konusunda uzman olan şirketlere devretmeye zorlamaktadır. Dağıtım faaliyetlerinin aksamadan hızlı ve ekonomik bir şekilde yapılandırılmasında, işletmelerin konusunda deneyimli lojistik firmalarından hizmet alması giderek artmaktadır.

İşletmelerin lojistik faaliyetlerinde bu şekilde dış kaynak kullanmak suretiyle gerçekleştirdikleri faaliyetler, üçüncü parti lojistik (3PL) olarak adlandırılmaktadır. Bu anlamda dış kaynak kullanımı, işletmelerin rekabet gücünü ve esnekliği artırmaya yönelik bir stratejidir.

Dış kaynak kullanımında ortaya çıkan bir başka kavram ise, dördüncü Parti Lojistik (4PL) yaklaşımıdır. Lojistik faaliyetlerini gerçekleştirmede optimizasyonu sağlayacak en başarılı 3PL şirketleri seçilmektedir ve 4PL şirketi bunlar arasındaki koordinasyonu sağlamaktadır. 4PL ile uzman firmanın bilgi, deneyim ve teknolojisi de alınarak, işletme süreçleri yeniden tasarlanarak geliştirilmektedir.

3PL'lerin en büyük hataları, sadece maliyet düşürme amacıyla hareket etmeleri, müşteri için değer yaratmaya çalışmamaları, sürekli gelişmenin ve yeniden yapılanmanın üzerinde durmamalarıdır. 4PL ve TZY faaliyetleri bir bütün olarak düşünüldüğünde hem maliyet hem de kalite alanında gelişmeler kaydedilecektir [113].

Dördüncü parti lojistik işletmelerinin sağladığı hizmetlerden bazıları şunlardır:

- Taşıma hizmetlerinin yanında dağıtım ve depolama gibi diğer lojistik faaliyetlerini de entegre bir biçimde sağlamak,
- Lojistik alanındaki değişimlerle birlikte organizasyonel konulardaki gelişmeleri de birleştirerek, işletme yönetimine sunmak,
- Çalıştıkları firmaların işlerini kısa bir sürede öğrenerek firma müşterileri için daha iyi çözümler üretmek,
- Güçlü teknolojik altyapı ile, başarılı bir tedarik zinciri uygulaması meydana getirmektir [113].

Araç Rotalama Problemi (ARP), ilk olarak [114] tarafından yapılan bir çalışma ile literatürde yerini almıştır. Problem, kamyon dağıtım problemi olarak ele alınmıştır. [114] tarafından önerilen metot [115] tarafından geliştirilmiş ve klasik tasarruf metodunu önermişlerdir.

Klasik araç rotalama probleminde, verilen n tane şehrin belirli talep miktarları vardır. Depo noktasının talebi 0 olarak kabul edilir. Amaç fonksiyonu temel olarak, rota maliyeti en küçüklenmesi şeklinde tanımlanır.

Klasik ARP'nin temel kısıtları şunlardır:

- Her şehre mutlaka ve sadece bir kere uğranmalıdır.
- Her müşteri noktası, k araç rotasından sadece bir tanesinde mutlaka yer almalıdır.
- Müşterilerin talepleri mutlaka karşılanmak zorundadır.
- Her rota, mutlaka depodan başlar ve depoda sona erer.
- Aracın rotasında bulunan şehirlerin talepleri toplamı, o aracın kapasitesini aşamaz.

ARP literatürde sıkça yer alan Gezgin Satıcı Problemi (GSP) ile benzerlik göstermektedir. GSP ile ARP arasındaki fark, GSP'de tek satıcı olmasına rağmen ARP'de birden fazla gezgin (araç) bulunmaktadır [116].

ARP'de, belirli bir müşteri kümesine hizmet edecek olan bir araç filosunun izleyeceği rotaların en iyisinin belirlenmesine çalışılır [117]. ARP'de bir veya birden fazla amaç dikkate alınabilmektedir. En bilinen amaçlardan bazıları şunlardır:

- Toplam taşıma maliyetlerini en küçüklemek,
- Tüm müşterilere hizmet etmek için gerekli olan toplam araç sayısını en küçüklemek,
- Toplam seyahat edilen mesafeyi en küçüklemek,

- Toplam seyahat süresini en küçükleme,
- Müşteri siparişlerinin geç karşılanmasından kaynaklı ceza maliyetlerini en küçükleme.

3.1 Araç Rotalama Problem Çeşitleri

3.1.1 Klasik Araç Rotalama Problemleri

Klasik araç rotalama problemlerine ait kısıtlar şu şekildedir:

- Rotalar depoda başlar ve depoda sona erer.
- Aynı kapasiteye sahip araçlar (homojen filo) bulunmaktadır.
- Müşteri talepleri deterministiktir.
- Her şehre mutlaka ve sadece bir araç uğramalıdır ve şehrin talebini tamamiyle karşılamalıdır.

[118] tarafından yapılan formülasyon aşağıdaki şekilde verilebilir:

$V = \{1, \dots, n\}$: düğüm kümesi, 1 noktası depoyu göstermektedir. Kalan düğümler müşteri noktalarıdır.

İndisler

c_{ij} : i şehrinden j şehrine gitme maliyeti veya zamanı

S: Alt tur sayısı

$v(S)$, optimal çözümde tüm S alt turları ziyaret etmek için gerekli araç sayısı alt sınırıdır.

Karar Değişkeni

$x_{ij} = \begin{cases} 1, & i \text{ şehrinden } j \text{ şehrine gidiliyorsa} \\ 0, & \text{diğer durumda} \end{cases}$

Model

$$\text{Min. } Z = \sum_{i \neq j} c_{ij} x_{ij} \quad (3.1)$$

$$\sum_j x_{ij} = 1, \forall i \in V \quad (3.2)$$

$$\sum_i x_{ij} = 1, \forall j \in V \quad (3.3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - v(S), (S: S \in V \setminus \{1\}, |S| \geq 2) \quad (3.4)$$

$$x_{ij} \in \{0,1\} \forall i, j \in V; i \neq j \quad (3.5)$$

Amaç fonksiyonu (3.1), toplam mesafeyi en küçükmektir. Kısıt (3.2)'ye göre her şehirden mutlaka bir başka şehre gidilmelidir. Kısıt (3.3)'e göre her şehre mutlaka bir yerden gelinmelidir. Kısıt (3.2) ve (3.3) kısıtları araç akış kısıtlarıdır. Kısıt (3.4) alt tur engelleme kısıtıdır. Kısıt (3.5) değerlerin 0 ve 1 değerlerinden birini alabileceğine ilişkin kısıttır.

3.1.2 Heterojen Filolu Araç Rotalama Problemi (HFARP)

Heterojen filolu araç rotalama probleminde, farklı türde (farklı kapasitede veya hızda) araçlar bulunmaktadır. Literatürde, heterojen filolu araç rotalama problemine ilk olarak, [119] tarafından tek sayfalık bir çalışma ile değinilmiştir.

Her türe ait araç sayısı, sınırsız ya da sınırlı sayıda olabilir. Filonun sınırsız olduğu durumda heterojen filoyu temel alarak, literatürde bu konu üzerine yapılan detaylı ilk çalışma [120] tarafından yapılmıştır.

3.1.3 Açık Uçlu Araç Rotalama Problemi (AUARP)

Literatürde bu konuya ilk defa [121] tarafından değinilmiştir. Bu problem türünde rotalar depoda başlar, bir müşteri noktasında sona erer. Tur depoda tamamlanmadığından açık uçlu olarak tanımlanmıştır. Araçların depoya dönmemesi, genelde firmaların kendi araç filosundaki araçlarının hepsinin firmaya ait olmaması, ya da araçların müşteri taleplerini karşılamada yetersiz olduğu durumlarda olur. Eğer araçlar başka bir firmadan kiralandıysa, bu kiralanan araçların firmanın kendi deposuna dönme zorunluluğu bulunmamaktadır [122].

3.1.4 Heterojen Filolu Açık Uçlu Araç Rotalama Problemi (HFAUARP)

Problem, literatürde [123] tarafından yapılan çalışma ile tanınmıştır. Bu problem tipinde araçlar ve bu araçların belirli sayıları bulunmaktadır. Araçlar depoya dönmemektedir. Rotalarını herhangi bir müşteri noktasında tamamlamaktadır.

3.1.5 Zaman Pencereci Araç Rotalama Problemi (ZARP)

Bu problemde müşteri talepleri belirli bir zaman penceresi $[a_i, b_i]$ içerisinde karşılanmalıdır. Burada a_i : i. müşteriye servisin başlaması gereken zamanı, b_i : i. müşteriden en geç çıkma zamanı olarak belirtilmektedir.

Araçlar depodan ayrıldıktan sonra, t_{ij} (i şehrinden j şehrine gitmek için harcanan zaman) kadarlık bir seyahat süresinden sonra müşteri noktasına uğrayarak, s_i kadarlık bir sürede müşteriye hizmet verirler.

3.1.6 Kapasite Kısıtlı Araç Rotalama Problemi (KKARP)

Kapasite kısıtlı araç rotalama probleminde, her aracın belirli bir kapasitesi vardır ve bu araçların kapasiteleri birbirine eşittir. Müşterilerin talepleri önceden bellidir. Araçlar rotalarına depodan başlar ve tekrar depoya dönerek rotalarını sonlandırırlar. Müşterilerin talepleri tek seferde karşılanır [124].

3.1.7 Mesafe Kısıtlı Araç Rotalama Problemi (MKARP)

Bu problem tipinde, belirlenmiş olan rotalara atanan her aracın gidebileceği maksimum mesafe kısıtı bulunmaktadır.

3.1.8 Önce Dağıt Sonra Topla Araç Rotalama Problemi (ÖDSTARP)

Bu problem çeşidinde, iki farklı müşteri kümesi vardır. Birinci kümeyi ürün toplanılacak, ikinci kümeyi ise ürün teslim alınacak müşteriler oluşturmaktadır.

Rotalar belirlenirken, depodan hareket etmeden önce araca yüklenecek ürünlerin miktarının ve müşterilerden teslim alınacak ürünlerin toplam miktarının, araç kapasitesini aşmamasına dikkat edilmelidir. Eğer belirlenen rotada her iki müşteri tipinden de bulunuyorsa, toplama yapılacak müşteriler dağıtım yapılacak müşterilere uğradıktan sonra ziyaret edilmelidir [112].

3.1.9 Eş Zamanlı Topla Dağıt Araç Rotalama Problemi (EZTDARP)

Eş zamanlı problem tipinin adından da anlaşıldığı gibi, toplama ve dağıtım işlemleri aynı zamanda gerçekleştirilir. Bir müşteriye ürün teslimatı yapılırken, aynı müşteriden ürün toplanılmasına da izin verilmektedir [124].

3.1.10 Dinamik Araç Rotalama Problemi (DARP)

Dinamik ARP, literatüre 1988 yılında kazandırılmıştır. DARP’de yer alan “dinamik”lik kavramı karar vericiler için, araç rotaları bulunması ve güncellenmesinde ihtiyaç duyulan bilgilerin zamana bağlı olarak açığa çıkmasını göstermektedir [125].

Yolların durumu, müşterilerin özellikleri, dağıtım ve toplama işlemlerinin aynı anda bulunup bulunmaması gibi unsurlar dikkate alınarak DARP türleri çeşitlendirilmektedir [125].

3.1.11 Çok Depolu Önce Dağıt Sonra Topla Araç Rotalama Problemi (ÇDÖDSTARP)

Önce dağıt sonra topla araç rotalama probleminin geliştirilmiş halidir. Tek depo yerine, birden fazla depo bulunmaktadır. Her depoda sınırlı sayıda araç vardır ve her araç hareketine başladığı depoya geri dönmelidir. Amaç, tüm araç sayısı ve kat edilen mesafeye ilişkin maliyetlerin toplamını minimize edecek şekilde, optimal rotalara karar vermektir [124].

3.1.12 Stokastik Araç Rotalama Problemi

Deterministik ARP'nin farklı bir türüdür, bu problemdeki bazı parametreler rassaldır. Problem, örneğin stokastik talepli ya da stokastik zamanlı olabilir.

3.1.13 Bölünebilir Talepli Araç Rotalama Problemi (BTARP)

BTARP, ilk olarak [126] tarafından 1989 yılında çalışılmıştır. Bu problem tipinde müşteri talepleri bölünerek, birden fazla araçla karşılanmaktadır.

3.1.14 Geri Toplamalı ARP (GTARP)

İade durumu gerektiren ürünlere uygulanan araç rotalama problemleridir. Ambalaj, depozito, otomotiv sektöründe yedek parçaların geri dönüşümü için fabrikalara geri gönderilmesi gibi işlemler bu araç rotalama problemine örnek verilebilmektedir [115].

3.1.15 Periyodik ARP (PARP)

Periyodik araç rotalama problemi, rekabetçi ortam koşullarında müşterilerine diğer dağıtıcılardan önce ulaşarak, satış miktarlarını etkilemeyi ele alan ARP çeşididir. Bu problemde asıl amaç, toplam mesafeyi minimize etmekle birlikte, müşterilere yapılacak olan dağıtım miktarını (satışı) maksimize etmektir [124].

3.2 Milk-run ve Araç Rotalama

Milk-run lojistiğine ilişkin problemler, araç rotalama problemlerinin (ARP) özel bir durumudur [49].

Milk-run rotalarının tespiti, teslim alma ve/veya sevkiyat taleplerini içeren klasik araç rotalama problemi ile yapılabilir. Milk run ağındaki araç sevk etme problemini çözmek, araç rotalama problemini çözmek ile benzerdir [16]. ARP, karmaşık deterministik olmayan polinom zamanda çözülemeyen problemlerin bir sınıfına aittir [127,128]. Bu tür, çözümü zor problemlere NP-zor ismi verilmektedir. Yani, problem büyüklüğü arttıkça hesaplama zamanı katlanarak artar. Bu nedenle, milk-run ile araç rotalamaları problemleri de NP-zor yapıdadır.

[129] tarafından milk-run için aşağıdaki model önerilmiştir. Bu problemde homojen tipli ve kapasiteli araç filosu bulunmaktadır, araçlar depodan milk-run turuna başlayıp tedarikçilerin talebini karşıladıktan sonra tekrar depoya dönmektedir.

İndisler

N_0 : Depoyu içeren düğüm kümesi ($i,j=0,1,2,...,N$)

N : Tedarikçi düğüm kümesi ($i,j=1,2,...,N$)

V : Araç kümesi ($v=1,2... V$)

Parametreler

K : Araç Kapasitesi (m^3)

d_i : Tedariçi i nin talep miktarı

C_{ij} : Tedarikçi i ile j arasındaki toplam mesafe

n : Tedarikçilerin sayısı

u_j : Alt turları elimine etmek için değişken

Karar Değişkeni

$X_{ijv} = \begin{cases} 1 & i \text{ tedarikçisinden } j \text{ tedarikçisine } v \text{ aracı ile gidiliyorsa} \\ 0 & \text{diğer durumda} \end{cases}$

Model

$$\text{Min } Z = \sum_{i,j \in N_0} \sum_{v \in V} C_{ij} X_{ijv} \quad (3.6)$$

$$\sum_{i \in N_0} \sum_{v \in V} X_{ijv} = 1, \forall j \in N \quad (3.7)$$

$$\sum_{j \in N_0} \sum_{v \in V} X_{ijv} = 1, \forall i \in N \quad (3.8)$$

$$\sum_{i \in N_0} \sum_{j \in N} d_j X_{ijv} \leq K, \forall v \in V \quad (3.9)$$

$$\sum_{j \in N} X_{0jv} = 1, \forall v \in V \quad (3.10)$$

$$\sum_{i \in N} X_{i0v} = 1, \forall v \in V \quad (3.11)$$

$$\sum_{i \in N_0} X_{ikv} - \sum_{j \in N_0} X_{kjh} = 0, \forall k \in N \text{ ve } \forall v \in V \quad (3.12)$$

$$X_{ijv} \in \{0,1\}, \forall (i,j) \in N, \text{ ve } \forall v \in V \quad (3.13)$$

$$u_i + 1 - n(1 - \sum_{v \in V} X_{ijv}) \leq u_j, i \neq j \text{ ve } i,j > 0 \quad (3.14)$$

3.3 ARP'nin Çözümünde Kullanılan Yöntemler

ARP'nin çözümünde kesin, sezgisel ve meta-sezgisel yöntemlerden faydalanılabilmektedir. Kesin yöntemler, ARP problemleri için optimal sonuçlar vermektedir, fakat rotalama problemleri NP-zor sınıfında olmaları nedeniyle, problem boyutu büyüdükçe optimum sonuca ulaşmak zorlaşmakta ve uzun çözüm sürelerinden sonra sonuç elde edilebilmektedir. Çözümü daha kısa sürede elde edebilmek için yaklaşık çözümler üretebilen sezgisel ve meta-sezgisel yöntemler, problemin çözümünde sıklıkla kullanılan yöntemler olmuşlardır.

3.3.1 Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri, en iyi çözümü bulmayı garanti etmektedir. Fakat çözüm süresi, problemin boyutuna bağlı olarak, üstel olarak artmaktadır. Genelde küçük ve orta boyutlu problemler, bu yöntemlerle çözülebilmektedir [130].

3.3.1.1 Dal Sınır Algoritması (Branch and Bound Algorithm)

Satır ve sütun eleme yöntemiyle, rotalar belirlenmeye çalışılır. Alt tur oluşursa; en kısa döngüyü engelleyecek kısıtlar ile dallandırılır. İstenilmeyen rotalara atama yapılmaması için, maliyet matrisinde ceza katsayısı olarak büyük M sayısı atandıktan sonra matris tekrar baştan çözülür [124].

Tüm dallar için aday çözümler belirlenir. Tüm dallar için aynı iterasyonlar tekrarlandıktan sonra, en iyi çözüme karar verilir [124].

3.3.1.2 Dal-Kesme Algoritması (Branch and Cut Algorithm)

Düzlem kesme tekniği ve dal-sınır algoritmalarının birlikte kullanıldığı yöntemdir. ARP'nin çözümü için ilk önce problemin doğrusal programlama ile çözümü yapılır. Modelin çözümünde oluşabilecek olan alt turlar sıfıra eşitlenerek, problem dallara ayrılır. Daha sonraki dallarda ise, modele alt tur engelleme kısıtlayıcısı eklenerek optimum sonuca ulaşılmaya çalışılır [124].

3.3.1.3 Dal-Fiyat Algoritması (Branch and Price Algorithm)

Dal fiyat algoritmasında, dal sınır ve dal kesme algoritmaları birlikte kullanılır. Dal fiyat algoritmasındaki kolon oluşturma yaklaşımı, fiyatlandırma alt problemini çözerek, fiyatlandırma için kolonda seçilecek en uygun değişkeni bulmaktır. Fiyatlandırılacak en uygun değişken minimizasyon probleminde pozitif, maksimizasyon probleminde negatif değerlidir. Eğer bu kolonlar bulunursa, doğrusal programlama optimize edilir. Aksi takdirde dallanma süreci başlatılır [131].

3.3.2 Klasik Sezgisel Yöntemler

Klasik sezgiseller tur kurucu, tur geliştirici ve iki aşamalı yöntemler olarak 3 gruba ayrılabilir [124]. Aşağıda ARP'de kullanılan sezgisel yöntemlerden bazıları kısaca anlatılmıştır.

3.3.2.1 Clarke ve Wright'ın Tasarruf Algoritması (Clarke and Wright Savings Algorithm)

Tasarruf algoritması, [115] tarafından 1964 yılında geliştirilen ve belki de bilinen en iyi tur oluşturma sezgiseli olan tasarruf yöntemidir. Bu yöntem, her bir müşteri ikilisi arasındaki maliyet tasarrufunu hesaplayarak başlar [132]. Bu algoritmanın adımları 4. Bölümde detaylı olarak açıklanacaktır.

3.3.2.2 En Yakın Komşu Sezgiseli (Nearest Neighbor Algorithm)

En yakın komşu sezgiselinde, her zaman en yakın şehir ziyaret edilir. En yakın komşu sezgiselinin adımları aşağıdaki gibidir:

Adım1: Rasgele bir şehir seçilir.

Adım2: Ziyaret edilmemiş en yakın şehre gidilir.

Adım 3: Ziyaret edilmemiş şehir kalmış mı kontrol edilir. Kaldı ise adım 2 tekrar edilir.

Adım 4: Başlangıç şehrine geri dönülür.

3.3.2.3 Süpürme (Sweep) Yöntemi

Bu yöntem, orta ve büyük boyutta KARP problemlerini çözmek için geliştirilmiştir.

Adımları aşağıdaki gibidir [132] :

Adım1: Kullanılmamış araç (k) seçilir.

Adım2: En düşük açığa sahip nokta ile başlanarak, noktalar k araç kapasitesi doluncaya kadar k aracına eklenir. Rota üzerindeki tüm noktalar bitinceye kadar, bu işlem devam eder.

Adım3: Her bir araç rotası, GSP yöntemlerinden biri ile optimize edilir.

3.3.2.4 Açgözlü (Greedy) Sezgiseli

Her adımda daha önce kullanılmamış olan en kısa değerli kenarları ekleyerek tur oluşturan ve bunu yaparken, alt tur oluşumunu engellemeye çalışan bir algoritmadır. Algoritmanın adımları aşağıdaki gibidir:

Adım 1. Oluşturulan tüm kenarlar için, büyükten küçüğe sıralama yapılır.

Adım 2. En kısa kenar, alt tur oluşturmayacak şekilde ilgili tura eklenir.

Adım 3. Bütün düğümler tura katılana kadar, 2. adım yenilenir.

3.3.2.5 2-Opt Algoritması

Muhtemel çözüm olarak, herhangi bir uygun çözüm alınabilir veya basit sezgisellerle başlangıç çözümü bulunabilir. Geliştirilecek algoritma ile, ilk önce muhtemel çözüm üzerinden iki nokta çifti seçilir. Sonraki aşamada, turu bozmayacak şekilde iki nokta yer değiştirir [133].

3.3.2.6 3-Opt Algoritması

3-opt algoritması, 2-opt ile aynı mantıkla çalışan ve iki kenar yerine, üç kenar değişimi gerçekleştiren bir yöntemdir. Bu durumda artık üç yolu bağlamak için, iki farklı alternatif oluşur ve hangi alternatife daha iyi olduğuna karar verilir. Modelde 3-opt değişimi kalmayana kadar algoritma devam eder ve sonlandırılır [133].

3.3.3 Meta-Sezgisel Yöntemler

Meta-sezgisel yöntemler, çözüm uzayında etkili ve verimli olarak arama yapılabilmesi için, farklı yapılarda olan sezgisel algoritmaların birleştirilmesi ile meydana gelen, tekrarlayan problem çözme yöntemleridir [134]. Meta-sezgisel yöntemler ile olası çözümler içinde en iyi çözüme yakın aramalar yapılarak, yerel çözümlere takılmadan optimum çözüme yaklaşılr. Büyük boyutlu ve karmaşık problemlerin kesin yöntemler ile çözümü çok uzun zaman sürmesine rağmen, meta-sezgisel çözüm yöntemler ile daha makul bir sürede çözülebilmektedir.

Klasik sezgisel algoritmaların tabiattan esinlenerek geliştirilmiş hali meta-sezgisel algoritmaları oluşturmaktadır [135].

Son yıllarda ARP için geliştirilen meta-sezgisel yöntemlerden bazıları; Genetik Algoritma, Karınca Algoritması, Tabu Arama ve Tavlama Benzetimidir. Bu algoritmalarından aşağıda kısaca bahsedilmiştir.

3.3.3.1 Genetik Algoritma

Genetik algoritma, ilk olarak 1975' te [136] tarafından ortaya çıkarılmıştır. Genetik algoritmaların temeli, genetik biliminden gelmektedir.

Genetik algoritmanın başlangıç çözümü, basit sezgisellerden elde edilen çözümle başlar. Buna başlangıç popülasyonu denir. Bu çözümün her bir parametresi bir gen olarak, çözümün tamamı ise kromozom olarak düşünülür [124].

İlk kromozomlar bu şekilde oluşturulduktan sonra, mevcut iki kromozomun çaprazlanmasıyla ya da kromozomun mutasyona uğramasıyla, yeni nesil kromozomlar üretilir. Tüm kromozomlar uygunluk fonksiyonu ile değerlendirilir, amaç fonksiyonuna en uzak kromozomlar elenir ve bu şekilde en iyi kromozom yani optimal çözüm elde edilinceye kadar süreç tekrarlanır [124].

Genetik algoritma, araç rotalama problemlerinin çözümünde de sıklıkla kullanılmaktadır. Bu amaçla son yıllarda özellikle klasik sezgisel yöntemlerle beraber kullanılarak, melez yöntemler geliştirilmiştir [137].

3.3.3.2 Tavlama Benzetim

[138] tarafından geliştirilmiştir. Katıların ısıtılması ve sonra kristalleşmeye kadar yavaş yavaş soğutulması esasına dayanır. Tavlama benzetimi, yüksek bir sıcaklık değeriyle başlar. Her bir hesaplama adımında, mevcut çözümün komşuları arasından çok sayıda çözüm üretilir. Yeni çözümler, belirlenen kriterlere göre kabul edilir veya reddedilir. Her bir hesaplama adımından sonra sıcaklık, belirlenen bir fonksiyona göre azaltılır. Algoritma istenen iterasyona ulaşıldığında ya da sıcaklık minimum değerine ulaştığında veya istenen çözüme ulaşıldığında sonlandırılır [139].

3.3.3.3 Tabu Arama

Yasaklı liste, tabu aramanın temelidir. Bu liste düzenli olarak güncellenir ve algoritmanın tekrar etmeyecek şekilde çalışmasını sağlar. Fakat üretilen çözüm o ana kadar üretilen en iyi çözümden iyiye, yasaklı liste içinde olsa bile kabul edilip, yeni çözüm üzerinden devam eder [124].

3.3.3.4 Karınca Algoritması

Doğal hayattaki karıncaların hareketlerinin taklit edilerek, matematiksel modele aktarılmasına dayanır. Karıncalar geçtikleri yollara, feromon denilen bir tür kimyasal madde bırakırlar. Diğer karıncalar da bu feromon kokusunu takip edip, muhtemel yiyecek kaynaklarına ulaşırlar. Feromonun yoğun olduğu yolların, karıncalar tarafından tercih edilmesi olasılığı daha yüksektir.

Bu şekilde belli bir zaman sonra uzun yolların kullanımı engellenir, en kısa yol karıncalar tarafından işaretlenmiş olur [124,137].

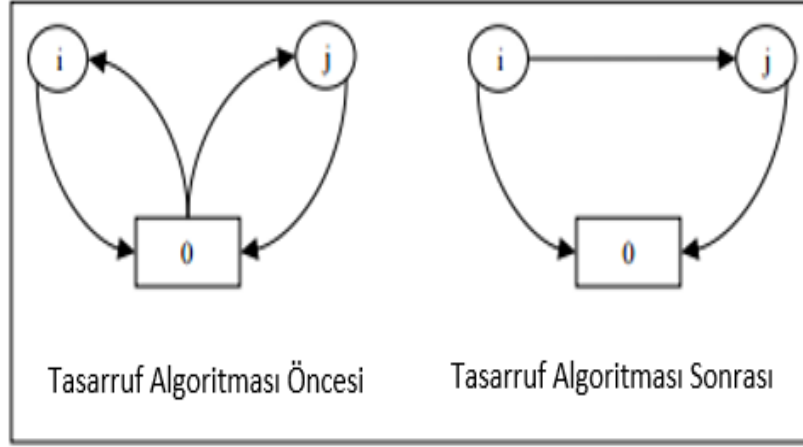
ARP, polinom zamanda çözülemeyen zor (NP-zor) bir problem olarak sınıflandırılır. Dolayısıyla, problem boyutu büyük olan gerçek hayat veri setlerini içerdiğinde, kesin optimizasyon yöntemlerini kullanmak, kabul edilebilir CPU zamanlarında bu problemleri çözmek için zor olabilir [130].

Bu çalışmada tur kurucu sezgisel yöntemlerden olan tasarruf algoritması ve meta-sezgisel yöntem olan genetik algoritma kullanılarak, tedarik zinciri dağıtım planlama problemi için çözüm geliştirilmiştir.

Bu bölümde, çalışmada kullanılan Clarke ve Wright'ın tasarruf algoritması ve genetik algoritma yöntemleri anlatılacaktır. Sonrasında bu iki yöntem kullanılarak geliştirilen algoritmadan bahsedilecektir.

4.1 Clarke ve Wright'ın Tasarruf Algoritması

Tur kurucu sezgiseller arasında en çok tercih edilendir. Herhangi iki nokta birbirine bağlanmadığında, depodan çıkan araç, noktaların her birine gidip gelecektir. Noktalar birbirine bağlandığında ise araç, sırasıyla birinci ve ikinci noktaya gidecek ve depoya geri dönecektir. Böylece iki noktanın bağlanması ve bağlanmaması arasındaki farka göre bir tasarruf oluşur. Bu tasarruflar tüm nokta çiftleri için hesaplanarak büyükten küçüğe doğru sıralanır ve tasarrufu yüksek olanlar kapasiteyi aşmadan birbirine ve depoya bağlanmaya çalışılır [124]. Tasarruf (savings) algoritması aşağıdaki şekilde gösterilen ayrı iki rotaya farklı bir rotanın eklenmesi sonucu elde edilen tasarrufu açıklar. Şekil 4.1 tasarruf algoritmasının şekilsel gösterimidir.



Şekil 4.1 Tasarruf Algoritması Gösterimi [140]

Algoritmanın adımları aşağıda verilmiştir [140]:

1. Her biri serimdeki yalnız bir düğüme uğrayan $|N|$ tane tur oluşturun.
2. Serimdeki tüm (i,j) çiftleri için $S_{ij} = d_{i0} + d_{0j} - d_{ij}$ tasarruflarını hesapla. (S_{ij} : i ve j düğümleri arasında hesaplanan tasarruf, d_{i0} : i düğümünün depoya uzaklığı, d_{0j} : Deponun j düğümüne uzaklığı, d_{ij} : i düğümüyle j düğümü arasındaki uzaklık.)
3. Elde edilen tasarruflar, büyükten küçüğe sırala.
4. Tasarruflar listesinin en başından (i,j) çiftini ele al. Eğer,
5. (i,j) ayrıtındaki düğümler ayrı turlarda ise,
6. araç kapasitesi aşılmamış ise,
7. i ve j düğümleri bulundukları turda ilk veya son düğüm ise,
8. Ayrı iki turu, (i,j) ayrıtı ile birleştir.
9. Tasarruf listesi tamamen gözden geçirilinceye kadar veya araç kapasitesi aşıncaya kadar 4. adımı tekrarla.

Algoritmanın işleyişi aşağıda bir örnekle açıklanmıştır [141]:

7 düğüm noktası bulunan Tablo 4.1'deki uzaklıklara sahip bir problem için, 0'ıncı düğüm kümesi depoyu göstermektedir. Rotalamada kullanılan araçların kapasiteleri 30 metreküptür.

Tablo 4.1 D ğ mler Arası Uzaklıklar (km) ve Talep Miktarı (m³)

Uzaklık (km)	0	1	2	3	4	5	6	7	Talep (m ³)
0									0
1	4								12
2	4	5,66							12
3	2,83	6,32	2,83						6
4	4	8	5,66	2,83					16
5	5	8,54	8,06	5,39	3				15
6	2	4,47	6	4,47	4,47	4,12			10
7	4,24	3,16	7,62	7,07	7,62	7	3,17		8

D ğ mler arasındaki tasarruflar $S_{ij} = d_{i0} + d_{0j} - d_{ij}$ form l ne g re hesaplanır. Hesaplanan tasarruflar Tablo 4.2’de g sterilmiřtir.

Tablo 4.2 Hesaplanan Tasarruflar

Node i	Node j	Tasarruf	Hesaplanan Tasarruf
1	2	$d_{10} + d_{02} - d_{12}$	$4 + 4 - 5,66 = 2,34$
1	3	$d_{10} + d_{03} - d_{13}$	$4 + 2,83 - 6,32 = 0,51$
1	4	$d_{10} + d_{04} - d_{14}$	$4 + 4 - 8 = 0$
1	5	$d_{10} + d_{05} - d_{15}$	$4 + 5 - 8,54 = 0,46$
1	6	$d_{10} + d_{06} - d_{16}$	$4 + 2 - 4,47 = 1,53$
1	7	$d_{10} + d_{07} - d_{17}$	$4 + 4,24 - 3,16 = 5,08$
2	3	$d_{20} + d_{03} - d_{23}$	$4 + 2,83 - 2,83 = 4$
2	4	$d_{20} + d_{04} - d_{24}$	$4 + 4 - 5,66 = 2,34$
2	5	$d_{20} + d_{05} - d_{25}$	$4 + 5 - 8,06 = 0,94$
2	6	$d_{20} + d_{06} - d_{26}$	$4 + 2 - 6 = 0$
2	7	$d_{20} + d_{07} - d_{27}$	$4 + 4,24 - 3,16 = 5,08$
3	4	$d_{30} + d_{04} - d_{34}$	$2,83 + 4 - 2,83 = 4$
3	5	$d_{30} + d_{05} - d_{35}$	$2,83 + 5 - 5,39 = 2,44$

Tablo 4.2 Hesaplanan Tasarruflar (devamı)

Node i	Node j	Tasarruf	Hesaplanan Tasarruf
3	6	$d_{30} + d_{06} - d_{36}$	$2,83 + 2 - 4,47 = 0,36$
3	7	$d_{30} + d_{07} - d_{37}$	$2,83 + 4,24 - 7,07 = 0$
4	5	$d_{40} + d_{05} - d_{45}$	$4 + 5 - 3 = 6$
4	6	$d_{40} + d_{06} - d_{46}$	$4 + 2 - 4,47 = 1,53$
4	7	$d_{40} + d_{07} - d_{47}$	$4 + 4,24 - 7,62 = 0,62$
5	6	$d_{50} + d_{06} - d_{56}$	$5 + 2 - 4,12 = 2,88$
5	7	$d_{50} + d_{07} - d_{57}$	$5 + 4,24 - 7 = 2,24$
6	7	$d_{60} + d_{07} - d_{67}$	$2 + 4,24 - 3,16 = 3,08$

Tasarruflar büyükten küçüğe sıralanır ve kapasite kısıtı olan 30 m³ dikkate alınarak rotalar oluşturulur. Tablo 4.3'te tasarrufların büyükten küçüğe sıralanışı görülmektedir.

Tablo 4.3 Tasarrufların Büyükten Küçüğe Sıralanması

Düğüm	Hesaplanan Tasarruf	Birleşmiş Talep
4-5	6	16+15=31
1-7	5,08	12+8=20
2-7	5,08	12+8=20
2-3	4	12+6=18
3-4	4	6+16=22
6-7	3,08	10+8=18
5-6	2,88	15+10=25
3-5	2,44	6+15=21
1-2	2,34	12+12=24
2-4	2,34	12+16=28
5-7	2,24	15+8=23
1-6	1,53	12+10=22
4-6	1,53	16+10=26
2-5	0,94	12+15=27

Tablo 4.3 Tasarrufların Büyükten Küçüğe Sıralanması (devamı)

Düğüm	Hesaplanan Tasarruf	Birleşmiş Talep
4-7	0,62	16+8=24
1-3	0,51	12+6=18
1-5	0,46	12+15=27
3-6	0,36	6+10=16
1-4	0	12+16=28
2-6	0	12+10=22
3-7	0	6+8=14

Tablo 4.3'te ilk sırada bulunan 4 ile 5 düğümünü birleştirmek talep toplamı 31 ettiği için mümkün değildir, bu nedenle 1 ve 7 düğümünü birleştirerek çözüme başlanır. Kapasite kısıtları ve tasarruflar dikkate alınarak düğümler birleştirildiğinde, problemin çözümüne ilişkin rotalar şu şekildedir: {0-1-7-6-0}, {0-3-4-0}, {0-2-5-0} sırasıyla rota uzunlukları da şöyledir: 12,32; 9,66; 17,06.

4.2 Genetik Algoritma (GA)

GA, [136] tarafından, Darwin'in evrim teorisine göre çevre şartlarına adapte olabilenlerin yaşamını sürdürdüğü prensibinden yola çıkarak modellenmiştir [142].

Genetik Algoritma (GA), 1970'te Michigan Üniversitesi'nde geliştirilmiştir ve orijinal olarak kısıtsız optimizasyon için uygulanmıştır. Yaklaşık olarak 10 yıl sonra, GA sıralama tabanlı problemleri içeren kısıtlı optimizasyon için uygulanmıştır [143].

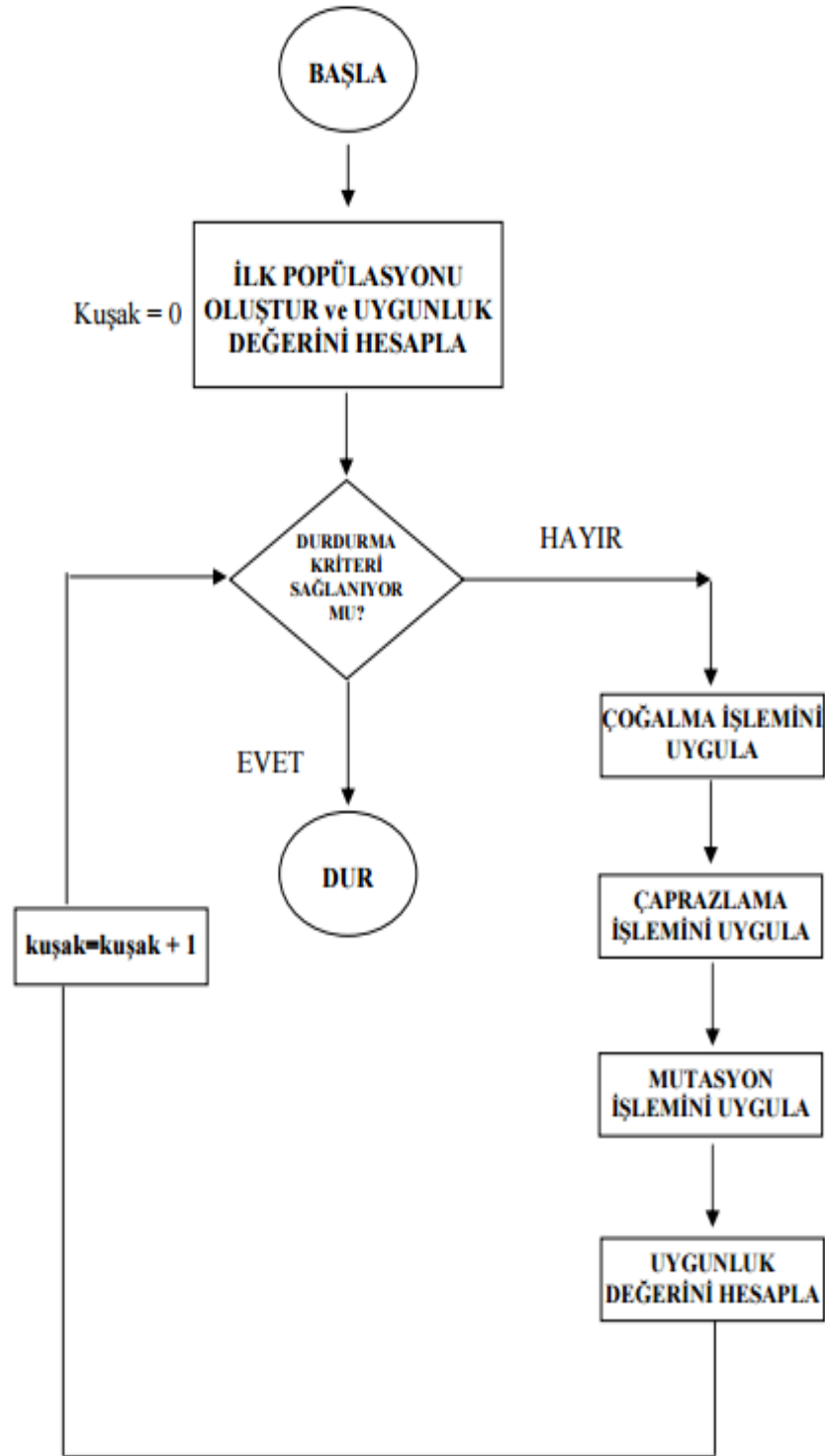
Genetik algorithmada kullanılan genel kavramlar aşağıda açıklanmıştır:

- Gen: Genler, çözümün bir özelliğini temsil eden en küçük anlamlı birimlerdir. Rotalama problemlerinde genler, çözüm kümesinin parametreleri olan konum değerlerini gösterirler.
- Kromozom: Birden fazla genin bir araya gelerek oluşturduğu diziye denilmektedir. Her bir kromozom, ilgili problemin olası çözümlerinden birini ifade eder.

- Popölasyon: Kromozomların bir araya gelerek oluşturduđu, alternatif çözüm kümesini temsil eden topluluktur. Popölasyon sayısı ile ilgili genel bir kural bulunmamaktadır.
- Kodlama: Probleme özgü bilgilerin, genetik algoritmanın kullanacağı şekle çevrilmesi işlemidir.
- Seçim: Oluşturulan popölasyondan bir sonraki nesle geçecek olan bireylerin belirlenmesi işlemidir.
- Uygunluk Fonksiyonu Değeri: Uygunluk fonksiyonu, oluşturulan her bir bireyin uyumunun hesaplanması için kullanılan fonksiyondur. Problem amacı ile orantılı olarak oluşturulan uygunluk fonksiyonu değeri en iyi olan kromozomlar, bir sonraki nesle aktarılır.
- Çaprazlama: Popölasyon içerisinde seçilmiş olan bireyler arasından, iki bireyin kromozomlarının belirli bir bölümünün karşılıklı olarak yer değiştirmesi operatörüdür. Belirli bir oran ile yapılan işlem sonucunda, farklı özelliklere sahip yeni bireyler oluşur.
- Mutasyon: Belirli bir oran doğrultusunda bir veya daha fazla genin değiştirilerek, farklı bir birey oluşturma işlemidir.

Genetik algoritmaların işlem adımları şöyle açıklanabilir [144]:

- Arama uzayındaki tüm mümkün çözümler, dizi olarak kodlanır.
- Genellikle rassal bir çözüm kümesi seçilir ve başlangıç popölasyonu olarak kabul edilir.
- Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
- Bir grup dizi, belirli bir olasılık değerine göre rassal olarak seçilip çoğalma işlemi gerçekleştirilir.
- Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Önceden belirlenen kuşak sayısı boyunca, yukarıdaki işlemler devam ettirilir.
- İterasyon, belirlenen kuşak sayısına ulaşınca işlem sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir.



Şekil 4.2 Genetik Algoritma Adımları [145]

Şekil 4.2’de algoritmanın işleyiş adımları görülmektedir. Genetik algoritma adımlarında yürütülen işlemler aşağıda anlatılmıştır.

4.2.1 Çözümlerin Kodlanması

Mümkün çözümlerin gösterimi için, kromozom yapılarının uygun şekilde kodlanması gerekmektedir. Kodlamanın şekli, problemin yapısına göre farklılık göstermektedir. Tüm problemler için uygun tek bir gösterim yoktur [146]. Kodlama çeşitleri aşağıdaki şekilde özetlenebilir.

İkili Kodlama

Çözüm uzayındaki değerlerin ikili sayma sisteminde ifade edilmesi ile gerçekleştirilir.

İkili kodlamada her bir kromozom, kullanıcı tarafından belirlenen uzunluğa sahip ikili (binary) bir bit dizisi olarak kodlanır. GA uygulamalarının çoğunda bu kodlama şekli kullanılmaktadır [147]. 1991’de, GSP için bir ikili kodlama yaklaşımını uygulanmıştır. Ancak küçük boyuttaki GSP’ler için yüksek kaliteli sonuçlar elde edilmiştir (en yüksek test, 100 şehirden oluşmaktadır) [148]. Şekil 4.3’te ikili kodlama örneği gösterilmektedir.

Kromozom1	110101110010
Kromozom2	011010011101

Şekil 4.3 İkili Kodlama Gösterimi

Permütasyonlu Kodlama

Sıralama problemlerinde permütasyonlu kodlama kullanılmaktadır. Kodlamada, her kromozom, dizideki bir sırayı temsilen sayılardan oluşmaktadır. Şekil 4.4’te permütasyonlu kodlama örneği gösterilmektedir.

Kromozom1	1 5 2 3 5 2 6 4 6 9 8
Kromozom2	8 6 3 6 3 9 6 3 1 5 8

Şekil 4.4 Permütasyon Kodlama Gösterimi

Değer Kodlama

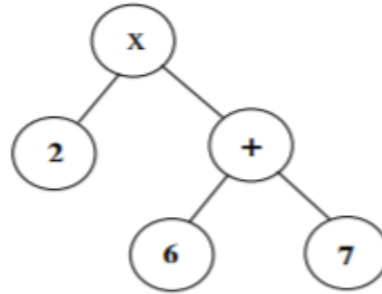
Değer kodlama, problemin gerçek değerlerinin direkt olarak kullanıldığı kodlama yaklaşımıdır. Kodlanan karakterler tamsayılar, ondalıklı sayılar, harfler ya da ilgili problem için anlamlı sözcükler olabilir. Şekil 4.5'te değer kodlama örneği gösterilmektedir.

Kromozom1	1.23, 2.12, 3.14, 0.34, 4.62
Kromozom2	ABDJEIFJDHDDLDFLFEGT

Şekil 4.5 Değer Kodlama Gösterimi

Ağaç Kodlama

Ağaç kodlama, genellikle programların geliştirilmesinde kullanılan bir kodlama biçimidir. Burada, çözüm ağaç yapısı şeklinde gösterilmektedir. Şekil 4.6'da ağaç kodlama örneği gösterilmektedir.



Şekil 4.6 Ağaç Kodlama Gösterimi

Sekizli Kodlama

Bu kodlamada kromozom 0-7 sayıları arasındaki sayılarla temsil edilirler [149]. Şekil 4.7'de sekizli kodlama örneği gösterilmektedir.

Kromozom1	06254524
Kromozom2	63726425

Şekil 4.7 Sekizli Kodlama Gösterimi [149]

Onaltılı Kodlama

Bu kodlamada kromozom onaltılık sayıları (0-9, A-F) göstermektedir [149]. Şekil 4.8’de, on altılık kodlamayı göstermektedir.

Kromozom1	97AE
Kromozom2	A2C6

Şekil 4.8 On Altılık Kodlama Gösterimi [149]

4.2.2 Başlangıç Popülasyonun Oluşturulması

Olası çözüm verilerini içeren kromozomlar bir araya gelerek popülasyonu oluştururlar. Popülasyon büyüklüğünü belirlemede genel bir kural bulunmamaktadır.

Genetik algoritmanın popülasyon büyüklüğü algoritmanın arama etkinliğini etkiler. Popülasyon büyüklüğü çok küçükse, genetik algoritma, iyi çözüm bulabilmek için çözüm alanında yeterince araştırma yapamaz. Öte yandan, büyük bir popülasyon büyüklüğü, işlem yükünü arttırarak, algoritmanın çözüm alanında kaybolmasına neden olabilir. Bu nedenle, popülasyon büyüklüğü belirlenmesinde her problem tipi ayrı olarak ele alınmalıdır [150].

Başlangıç popülasyonu, rassal ya da sezgisel yöntemlerden faydalanılarak oluşturulabilmektedir. Literatürdeki çalışmaların çoğu rasgele seçilen bir çözümle başlar. Bazı yazarlar daha iyi bir başlangıç oluşturmak için ağgözlü algoritmayı önermişlerdir [150].

4.2.3 Uygunluk Fonksiyonun Belirlenmesi

Uygunluk fonksiyonu, çözümün istenen optimum çözüme yakınsamasını test etmek için tanımlanan performans ölçütleridir. Kromozomlar ile tanımlanan istenen özelliklerin doğru şekilde ölçümlenebilmesi açısından uygunluk fonksiyonunun belirlenmesi önemlidir. Bütün sonuçları değerlendirilmek için uygunluk fonksiyonu kullanılacağından, etkili şekilde hesaplanabilir olmalıdır [150].

Kromozomların kalitesi, uygunluk fonksiyonu ile hesaplanan değerlere göre ölçülmekte olup; yüksek değerler kromozomun bir sonraki nesillere aktarılma olasılığının daha yüksek olduğunu belirtir. Optimizasyon problemlerinde genellikle uygunluk fonksiyonu oluşturmada amaç fonksiyonundan yararlanılmaktadır.

4.2.4 Seçim (Çoğalma)

Çoğalma, bireyleri seçme işleminden, seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur [145].

Seçim işlemi, bir sonraki kuşak için yavru üretmek amacıyla hangi ebeveynlerin yer alması gerektiğine karar vermektedir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımaktır [151,145].

Seçim operatörü GA'da yeniden üreme işleminin en önemli kısmını oluşturur.

Literatürde kullanılan en popüler seçim tekniklerinden bazıları şunlardır: turnuva, rassal, elitizm ve rulet tekeridir [150]. Bunlar aşağıda açıklanmıştır.

Rulet Teker: Rulet tekeri yöntemi ilk kez en sık kullanılan yöntemlerden biridir [133,152]. Bireyler uygunluk değerlerine göre belirli olasılık değerlerine sahiptirler. Bu olasılıklar, her bir bireyin uygunluk değerinin toplam uygunluk değerine oranlanması ile elde edilir. Hesaplanan uygunluk değerleri büyüklüklerine oranla bir çemberin üzerine yerleştirilir. Yöntemin adımları aşağıda açıklanmıştır [153]:

Adım 1. Her bir i bireyinin uygunluk değeri f_i hesaplanır.

Adım 2. Tüm bireylerin uygunluk değerleri toplanır ve toplam uygunluk değeri $F = \sum_{i=1}^m f_i$ hesaplanır.

Adım 3. Her bir bireyin seçilme olasılığı $p_i = f_i / \sum_{i=1}^m f_i$ hesaplanır.

Adım 4. Bu oranlar sırasıyla toplanarak her bir birey için (0,1) aralığında seçilme üst sınırını veren değerler elde edilir.

Adım 5. Popülasyondaki birey sayısı kadar (0,1) aralığında rassal sayı üretilir ve her rassal sayıdan bir sonra gelen birikimli olasılık değerine karşılık gelen birey seçilir.

Rassal Seçim: Başlangıç popülasyonu içerisinde, hiçbir kısıt veya kural olmadan, popülasyon sayısı kadar rasgele kromozom seçimi ile gerçekleştirilen yöntemdir. Kromozomları seçmek için belirli aralıklarda verilen düzgün (uniform) bir sayı üreticisi kullanılmaktadır [133].

Turnuva Seçimi: Turnuva seçimi yönteminde kromozomlar bir bekleme havuzuna atılırlar. Bu havuzdan rasgele olarak seçilen kromozomlar ile bir alt küme oluşturularak, bu alt küme elemanları turnuvaya sokulur. Bu kümeler genelde ikili şekilde oluşturulurken, çok büyük popülasyonlarda n elemanlı olarak oluşturulabilir. İkili kümeler için, havuzdan rassal olarak iki kromozom seçilir ve uygunluk değerleri karşılaştırılır. Uygunluk değeri yüksek olanın bir kopyası çiftleşme havuzuna koyulur. Bu işlemler popülasyon sayısına erişinceye kadar devam eder [133].

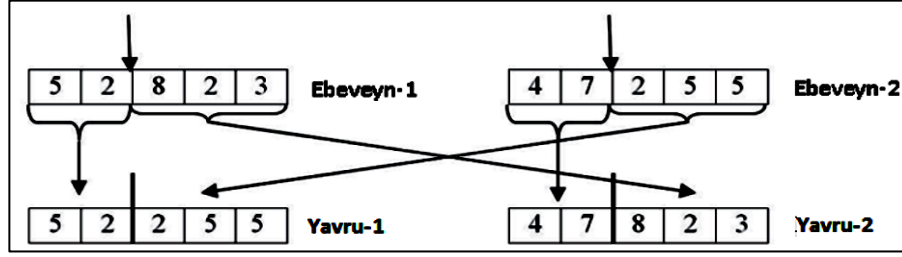
Elitizm: Elitizm yönteminde, her iterasyonda, bir önceki neslin en güçlü olarak belirlenen kromozomunun bir sonraki nesle doğrudan aktarılması sağlanmaktadır. Popülasyonun en iyi bireyi korunur. Diğer bireyler uygunluk değerine bağlı olarak diğer seçim yöntemleriyle değiştirilir. Kısaca elitizm, popülasyonun maksimum uygunluk değerinin azalmasını engelleyen bir yöntemdir [147]. Ebeveynlerden en iyi kromozomları seçme ve elitizm ile yavrular oluşturma iyi sonuçlar vermektedir. Literatürde yapılan bir çalışmada yerel optimum problemini aşmada elitizm tekniği popülasyon kontrol tekniği olarak kullanılmıştır [150].

4.2.5 Çaprazlama

Çaprazlama, yeni çözüm alanı yaratarak yeni yerlerde arama yapmak için birleştirme mekanizmasıdır. İki ebeveyn kromozomu arasında genetik özelliklerin, ebeveynlerin özelliklerini taşıyacak çocuklar oluşturmak amacıyla, değiş tokuşunu anlatmaktadır [154]. Bu tekniklerin farklı çeşitleri literatürde kullanılmaktadır [155-158]. Bu yöntemlerden çalışmalarda sıklıkla kullanılanlarından bazıları aşağıda bahsedilecektir.

Tek Noktalı Çaprazlama: Kromozom üzerinde bir nokta belirlenmektedir. Daha sonra, ebeveyn kromozom çiftleri, belirlenen bu noktadan ayrılmaktadır. Ayrılan parçalar karşılıklı olarak yer değiştirilip sonucunda yeni bireyler elde edilmektedir.

Şekil 4.9'da permütasyonlu kodlama için tek noktali çaprazlama örneği görülmektedir.



Şekil 4.9 Tek Noktalı Çaprazlama [150]

Çok Noktalı Çaprazlama: Çok noktali çaprazlamada, bir yerine n sayıda çaprazlama noktası belirlenmektedir. Burada, kromozom üzerinde belirlenen bu noktalar arasından çıkartılan gen parçaları karşılıklı yer değiştirilerek çaprazlama işlemi yapılmaktadır.

Düzenli (Uniform) Çaprazlama: Uniform çaprazlamada, 0 ve 1'lerden oluşan kromozomlarla aynı uzunlukta rasgele bir maske dizisi tanımlanır. Maske, ebeveyn kromozomları arasında değiş tokuş yapılacak genetik özelliklerin pozisyonunu belirlemektedir. Maskede, "0" ilk ebeveynin özelliğinin aktarılacağı anlamına gelmektedir, "1" ise ikinci ebeveynden aktarım yapılacağını göstermektedir [150]. İkinci yavru için bu işlemin tersi uygulanır. Şekil 4.10'da permütasyonlu kodlama için uniform çaprazlama örneği görülmektedir.

1	2	1	2	3	1	2	1	2	3	1	2	1	2	3	4	5	1	2	3	4																								
5	10	1	6	9	8	2	7	3	4	7	6	5	8	5	7	11	2	4	4	2																								
1	2	1	2	3	1	2	1	2	3	1	2	1	2	3	4	5	1	2	3	4																								
3	4	2	6	1	8	4	6	2	4	4	5	8	3	2	7	1	1	3	2	2																								
[					0					]					[					1					]					[					0					]				
1	2	1	2	3	1	2	1	2	3	1	2	1	2	3	4	5	1	2	3	4																								
5	10	1	6	9	8	2	7	3	4	4	5	8	3	2	7	1	2	4	4	2																								

Ebeveyn-1

Ebeveyn-2

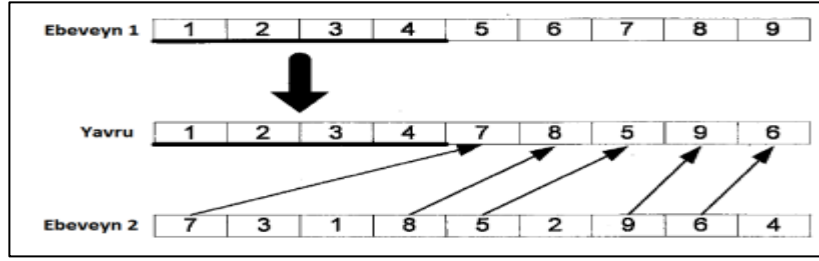
Maske

Yavru

Şekil 4.10 Uniform Çaprazlama [150]

Sıralı Çaprazlama (OX1): Bu çaprazlama operatörü, 1985'te önerilmiştir [148]. Tek noktali çaprazlamada olduğu gibi bir nokta belirlenmektedir ve bu noktadan kesilen birinci ebeveyn kromozomun ilk parçası, yavrulara aktarılmaktadır. Bu aktarımdan sonra, yavruda eksik kalan gen değerleri tespit edilerek, ikinci ebeveyndeki sıraları ile bu yavruya aktarılmaktadır. Aynı işlemin ters uygulaması ise ikinci yavru oluşturmak için yapılmaktadır [133].

Şekil 4.11’de sıralı çaprazlama örneği görülmektedir.

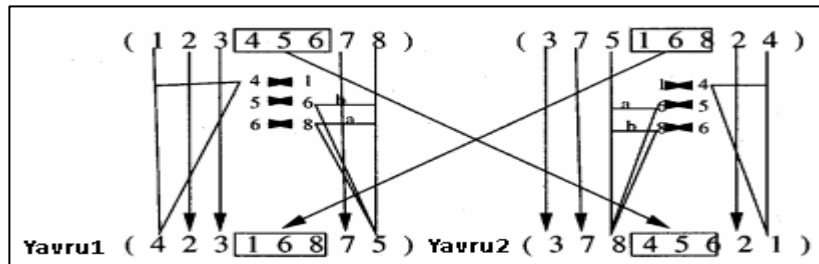


Şekil 4.11 Sıralı çaprazlama [133]

Sıraya Dayalı Çaprazlama (OX2): Bu yöntem, 1991’de geliştirilmiştir [148]. Bu yöntem ile çaprazlamada başlangıçta çaprazlama noktaları rassal olarak seçilmektedir ve birinci ebeveynde seçilen noktalarda bulunan genler yerlerini korumaktadır. İkinci ebeveynde bu noktalara karşılık gelen gen değerleri ise birinci ebeveynin aynı noktalarındaki genlerin ardına atanmaktadır. Boş kalan yerlere kalan genler sırası ile yerleştirilmektedir [133].

Kısmi Planlı Çaprazlama (PMX): Şekil 4.12’de kısmi planlı çaprazlama örneği görülmektedir. Sıralama ve değer bilgisi ebeveynlerden yavrulara aktarılmaktadır [148]. PMX, çaprazlama yöntemi adımları şunlardır [133]:

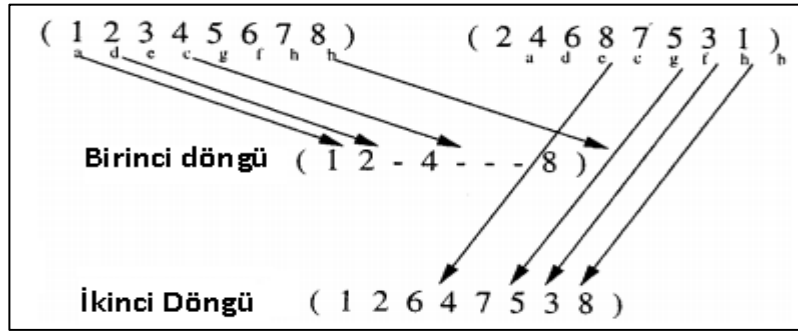
- Kromozomlar üzerinde rassal olarak iki çaprazlama noktası belirlenmektedir.
- Ortada kalan eşleştirme parçası karşılıklı olarak değiştirilmektedir.
- Değiştirilen parçalar arasındaki ilişki (eşleştirme ilişkisi) belirlenmektedir.
- Değiştirilmeyen parçalara da eşleştirme ilişkisi kapsamında değişimler uygulanmaktadır.



Şekil 4.12 Kısmi Planlı Çaprazlama [148]

Dairesel Çaprazlama (CX): Bu çaprazlama yöntemi, CX operatörünün PMX operatöründen gezgin satıcı problemleri için daha iyi sonuçlar verdiğini çıkarmışlardır [148]. Şekil 4.13'te döngüsel çaprazlama örneği görülmektedir. Adımları şunlardır [133]:

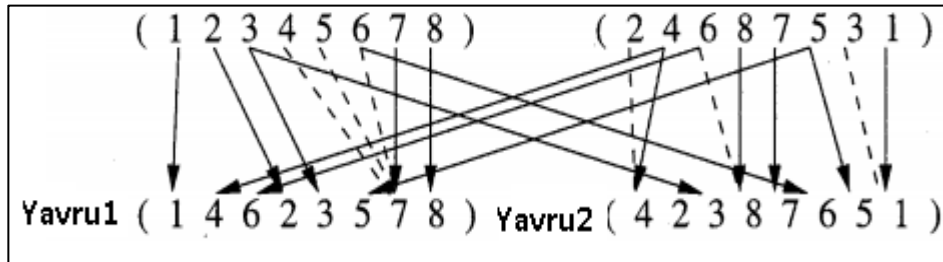
- Ebeveynlerin genleri arasında ilişki kurulup, döngü oluşturulmaktadır.
- Döngü içindeki genler birinci ebeveynden alınarak yavrudaki aynı noktaya aktarılmaktadır.
- İkinci ebeveynden halen döngüde olan ilgili genler silinmektedir.
- Kalan genler yavruya yerleştirilmektedir.



Şekil 4.13 Döngüsel Çaprazlama [148]

Pozisyona Dayalı Çaprazlama (POS): Pozisyona dayalı operatör, ebeveynlerden pozisyonların rasgele seçilmesi ile başlar [148]. Burada rassal olarak seçilmiş genler ilk ebeveynden çocuğa, yerleri aynı olacak şekilde aktarılır.

Diğer genler ise; ikinci ebeveynde bulundukları sıra ile boş kalan gen noktalarına yerleştirilir [133]. Şekil 4.14'te pozisyona dayalı çaprazlama örneği gösterilmektedir.

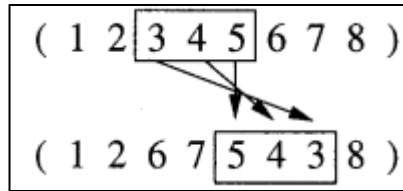


Şekil 4.14 Pozisyona Dayalı Çaprazlama [148]

4.2.6 Mutasyon

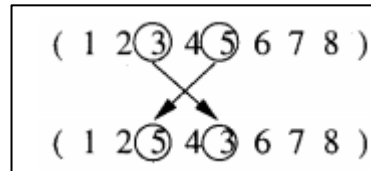
Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat popülasyon gerekli tüm bilgiyi vermez ise, çaprazlama tatmin edici bir çözüm sağlayamaz. Bundan dolayı, mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatör gerekir. Bu görevi mutasyon gerçekleştirir [145]. Mutasyon, kromozomdaki genlerin modifiye edilmesi ile gerçekleştirilmektedir. Yerel optimumlara takılmayı önleyerek çeşitliliğin devamını sağlamak amacıyla uygulanırlar [150]. Literatürde kullanılan mutasyon yöntemlerinden bazıları aşağıda anlatılmıştır.

Ters Çevirme Mutasyonu (Inversion Mutation): Ters çevirme mutasyonu, yer değiştirme mutasyonu ile benzerdir. Ters mutasyonda, kromozom içerisindeki genler arasından sıralı bir alt dizi seçilmekte ve seçilen bu alt dizi ters çevrilerek, seçilen rassal bir pozisyona yerleştirilmektedir [148]. Şekil 4.15’de bir örnek gösterilmektedir.



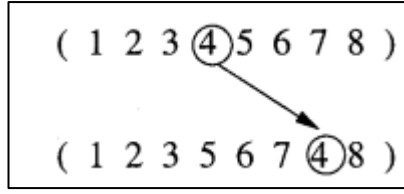
Şekil 4.15 Ters Çevirme Mutasyonu [148]

Değiş-Tokuş Mutasyonu (Exchange mutation): Bu operatör, rassal olarak kromozomdaki iki geni seçer ve bunları kendi aralarında yer değiştirir. Şekil 4.16’da bir örnek gösterilmektedir.



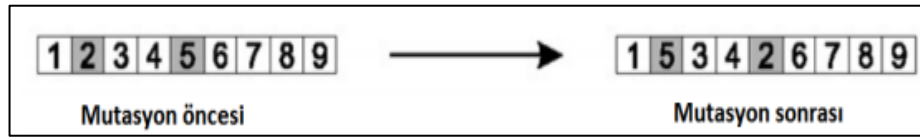
Şekil 4.16 Değiş -Tokuş Mutasyonu [148]

Ekleme Mutasyonu (Insertion mutation): Bu operatör ile, rassal olarak kromozomdaki bir gen seçilir ve kromozom üzerinde rasgele seçilen bir yere eklenir. Şekil 4.17’de bir örnek gösterilmektedir.



Şekil 4.17 Ekleme Mutasyonu [148]

Yer Değiştirme Mutasyonu (Displacement Mutation): Yer değiştirme mutasyonunda bir kromozom altkümesi rassal olarak seçilir ve rassal olarak seçilen bir pozisyona yerleştirilir. Yer değiştirme sağ veya sol tarafa rassal olarak seçilen pozisyon sayısınca gerçekleşebilir. Şekil 4.18’de bir örnek gösterilmektedir.



Şekil 4.18 Yer Değiştirme Mutasyonu [133]

4.2.7 Yeni Kuşağın Oluşması ve Döngünün Durdurulması

Yeni kuşak, üreme, çaprazlama ve mutasyon işlemlerinden sonra oluşup, bir sonraki kuşağın ebeveynleri olmaktadır. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder [145].

4.2.8 Genetik Algoritmalarla Parametre Seçimi

Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır. Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır [145,159].

Popülasyon büyüklüğünün belirlenmesi problemin tipine göre değişmekte olup, önemli olan araştırma alanının iyi bir şekilde taranmasını sağlayacak uygun büyüklüğü belirlemektir [150]. Popülasyon büyüklüğü 10’dan 100’e kadar değiştirilip, yapılan çalışmalarda en iyi sonucun en büyük popülasyon büyüklüğünün verdiğini göstermektedir [143]. Literatürdeki çalışmaların çoğunda popülasyon büyüklüğü 20 ile 100 arasında değişmektedir [150]. [160,161] tarafından, popülasyon büyüklüğü 100, çaprazlama oranı 0,6 olarak önerilmiştir.

Sezgisellerle birlikte GA kullanılan çalışmalarda, çatışan fikirler olduğu görülmektedir [143].

Çaprazlama oranı olarak, en sık kullanılan aralık [0.45, 0.95]'dir [161]. Mutasyon oranları için literatürde ortak düzenlemeler bulunmuştur. Bu değerler literatürde genellikle çaprazlama için %50-%95, mutasyon için %5-%70 aralığında değişmektedir. Literatürdeki bulgular, operatör olasılıkları ile çözümün etkinliği arasındaki ilişkinin kesin olmadığını göstermektedir [155,162]. Popülasyonun performans aralığı yerel optimuma takılmaya başladığında, çaprazlama ve mutasyon oranları, olasılıkları artmaktadır [163].

Her algoritmanın kendine özel bir sonlandırma kriteri bulunmaktadır [150]. Bazı yazarlar sonlandırma kriterini iterasyon sayısı olarak belirlemişlerdir [150,155]. Başka bir strateji ise, en iyi çözüm değeri, t nesil sayısı kadar yapılan işlemler sonunda değişmeyip aynı kalıyorsa algoritmanın durdurulmasıdır. Literatürde, t değeri 50 ve 200 arasında değişmektedir [164]. Sonlandırma kriteri, maksimum nesil parametresi olarak da düzenlenebilmektedir. Başka bir teknik ise, CPU zamanının sonlandırma kriteri olarak kullanılmasıdır [150].

Tedarik Zinciri Dağıtım Ağı Probleminin Özelleştirilmiş Genetik Algoritma ile Çözümlemesi

Bu bölümde, problem verilerinin sağlandığı firma hakkında tanıtıcı bilgiler, problemin çözümüne ilişkin oluşturulan optimizasyon model ve geliştirilen algoritma, her iki yöntemle elde edilen sonuçlar ve elde edilen sonuçlar için yapılan istatistiksel testlerin sonuçları anlatılacaktır.

5.1 Firma Tanıtımı

XYZ firması, uluslararası bir şirket olup dünya çapında 220 ülke ve bölgede faaliyet göstermektedir. 350.000'den fazla çalışanı ile lojistik ihtiyaçları için firmalara çözümler sunmaktadır.

Lojistik alanında verdiği hizmetlerden bazıları şunlardır:

Navlun Taşımacılığı: Hava, deniz, karayolu veya demiryolu ile taşımacılık hizmetlerini kapsamaktadır.

Lider Lojistik Hizmetleri Tedarikçisi: (Lead Logistics Provider-LLP): Hizmet sağladığı müşterilerin ihtiyaçlarına göre lojistik hizmetinin tümünü veya bir kısmını üstlenerek, lojistik hizmet sağlayıcıları (3PL) seçmektedir. Bu kapsamda müşterilerinin operasyonel ve performans yönetimi, maliyet yönetimi ve geliştirilmesine ilişkin sorumluluklarını üstlenmektedir. LLP hizmeti altında verdikleri hizmetler şunlardır:

- Tedarik zinciri analizi ve tasarımı
- Danışmanlık
- İş planlama
- Değişim yönetimi
- Proje yönetimi

- Tedarik zinciri yönetimi hizmetleri, birçok taşımacılık modunu ve geniş bir coğrafyada tedarikçileri koordine etmek
- Envanter planlama ve yönetimi
- Mal kabul, mal teslim ve tersine lojistik yönetimi

Depolama ve Dağıtım: Büyük bir şirketin depolama, yurtiçi dağıtım, üretim lojistiği, uluslararası hava, deniz ve karayolu taşımacılığı gibi tedarik zinciri boyunca ihtiyaç duyulabileceği tüm hizmetlerden, küçük bir şirketin ihtiyacı olabilecek taşımacılık hizmetine kadar tüm hizmetleri kapsamaktadır.

Yeşil Lojistik Çözümleri: Sera gazı emisyonlarının optimize edilmesi ve dengelenmesi için ihtiyacı olan müşterilere etkin önlemler geliştirme için süreçler vasıtasıyla yön göstermektedir. Bu konudaki bazı hizmetleri şunlardır: Yakıt ve araç yönetimi, Filo optimizasyonu, Model değişikliği, Şebeke tasarımı.

Tedarik Zinciri Çözümleri: Danışmanlık ve tasarım adımlarından son teslimat anına ve hatta tersine lojistiğe kadar, tüm sektörler için özel tedarik zinciri çözümleri üretmektedir.

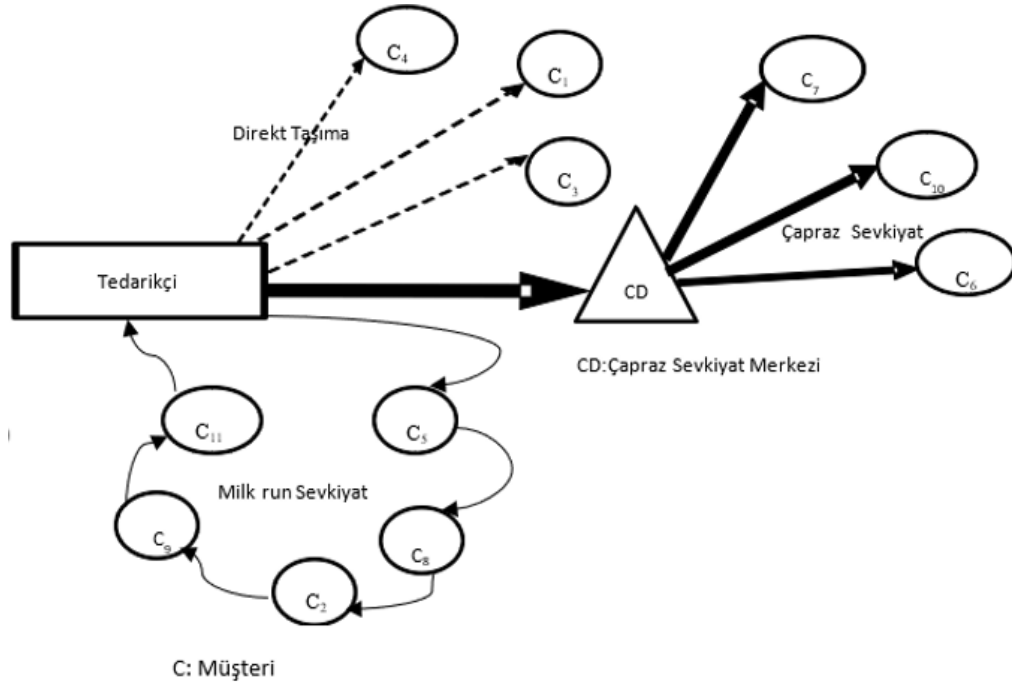
5.2 Problemin Tanımı

XYZ firmasının taşımacılık alanında hizmet verdiği uluslararası bir elektronik ürünler üreten fabrikanın dağıtım problemi için çözüm geliştirilmiştir. Amaç, elektronik ürünler üreten fabrikanın müşterilerine en uygun taşıma yöntemi ve en uygun araç seçimi ile maliyet minimizasyonu sağlanarak en optimal şekilde hizmet verebilmesidir.

XYZ firması, elektronik ürünler üreten fabrikanın taşımacılık hizmetlerini yönetmek için lojistik hizmet sağlayıcıları (3PL) kullanmaktadır. Bu nedenle, bu çalışmada kullanılan araç sayısının sınırsız olduğu kabul edilmektedir.

Taşımada, 4 farklı araç tipi kullanılmaktadır. Araçlar heterojen tipli olup, kiralama maliyetleri ve kapasiteleri birbirinden farklıdır. Araç sayısı sınırlaması olmadığından, müşterilere ürünlerin gönderimi için araçlar her an hazır bulunmaktadır. Müşterilere ürünün geç gönderilmesinden kaynaklanan ceza maliyeti bu nedenle oluşmamaktadır.

Elektronik ürünler üreten fabrikanın, Balkanlar, Polanya, Avusturya, Baltık, Macaristan ve Slovakya olmak üzere 6 bölgede müşterileri bulunmaktadır. Her bölgeye hizmet veren bir dağıtım merkezi bulunmaktadır. Elektronik ürünler üreten fabrika için tedarik zinciri ağı yapısı 2 aşamalıdır. Ağ yapısı, her bölge için 1 fabrika, 1 çapraz sevkiyat merkezi ve müşterilerden oluşmaktadır. Şekil 5.1’de ağ yapısı görülmektedir.



Şekil 5.1 Tedarik Zinciri Ağı Yapısı

Müşterilere ürünler direkt taşıma, milk-run ile taşıma, çapraz sevkiyat merkezi üzerinden taşıma olmak üzere 3 farklı şekilde teslim edilebilmektedir. Direkt taşıma ve milk-run taşıma için 4 farklı araç tipi de kullanılabilmektedir. Çapraz sevkiyat merkezi üzerinden taşımada ise, müşterilerin ürünleri en büyük araç kapasitesine olan araç tipi ile konsolide olarak fabrikadan çapraz sevkiyat merkezine taşınmaktadır.

Çapraz sevkiyat merkezinden müşterilere ise; her müşteri için küçük araç yola çıkarılarak direkt taşıma yöntemi ile teslimat yapılmaktadır. Aşağıda taşıma yöntemlerinden detaylı olarak bahsedilmiştir.

5.2.1 Direkt Taşıma

Bu yöntemle, fabrikadan müşterilere ürünler direkt olarak farklı kapasitedeki 4 farklı araç tipi ile taşınabilmektedir. Her müşteriye yalnızca bir araç hizmet vermektedir. Araçların tekrar depoya dönme koşulu aranmamaktadır. Araç sayısı kısıtlaması yoktur.

5.2.2 Milk-run ile taşıma

Bu yöntemle, fabrikadan müşterilere ürünler döngüsel sefer uygulaması ile taşınmaktadır. Farklı kapasitedeki 4 farklı araç tipi ile taşıma yapılabilmektedir. Fabrikadan müşterilere taşıma yapıldıktan sonra araç yine fabrikaya geri dönmektedir. Her müşteriye uğradığında araç için durak maliyeti ortaya çıkmaktadır.

5.2.3 Çapraz Sevkiyat İle Taşıma

Bu yöntemde, en büyük araç kapasitesine sahip olan araç tipi ile taşıma yapılmaktadır. Fabrikadan çapraz sevkiyat merkezine kadar tüm müşterilerin ürünleri konsolide olarak taşınmaktadır. Çapraz sevkiyat merkezinde müşterilerin talepleri ayrıştırılarak, çapraz sevkiyat merkezinden müşterilere özel küçük araçlarla teslimat sağlanmaktadır. Küçük araçlarla teslimat planlamasında araç rotalaması yapılmayacaktır. Küçük araçlarla dağıtım maliyeti hesaplanmasında, araçların, m^3 başına taşıma maliyetlerinden yararlanılacaktır. Çapraz sevkiyat merkezinde stoklama yapılmamaktadır.

5.3 Problemin Matematiksel Modeli

GSP, maliyetleri minimize etme amaçlı yürütülen klasik rotalama problemidir.

GSP'nin değişik çeşitleri de NP-hard problemlerindendir [143].

[49, 129, 118] tarafından geliştirilen modeller referans alınarak, bu çalışmada kullanılan model geliştirilmiştir. Matematiksel model uygun dağıtım stratejisini ve optimal rotaları belirlemektedir.

Mevcut problemin çözümüne ilişkin varsayımlar ve kısıtlar şunlardır:

- Müşteri siparişleri bölünememektedir, her müşteriye yalnızca bir araç atanabilmektedir.

- Araç filosu heterojendir ve tüm araçlar farklı kapasiteye sahiptir.
- Araç sayısı sınırlaması yoktur, her araç tipinin ihtiyaç duyulduğunda kullanıma hazır olduğu kabul edilmektedir. Bu ancak 3PL lojistik sağlayıcı kullanımı ile mümkündür [30].
- Milk-run ile dağıtımda durak maliyetleri ortaya çıkmaktadır.
- Müşteri siparişleri ertelenmemektedir ve günlük olarak karşılanmaktadır. Bu nedenle ceza masrafı oluşmamaktadır.
- Milk-run rota uzunluğu, sürücünün yasal sürüş süreleri nedeniyle belirli bir mesafe ile sınırlandırılmıştır.
- Müşteri talep miktarları deterministiktir ve bilinmektedir.
- Araçla taşınan toplam yük, araç kapasitesini aşmamaktadır.
- Tüm müşterilerin talebini karşılamak için üretici /tedarikçi kapasitesi yeterlidir.
- Tek bir çapraz sevkiyat merkezi bulunmaktadır ve çapraz sevkiyat merkezinde stok bulunmamaktadır.
- Çapraz sevkiyatta, üreticiden/tedarikçiden çapraz sevkiyat merkezine nakliye en büyük araç kapasitesine sahip araçlarla yapılmaktadır. Gönderiler çapraz sevkiyat merkezinden müşterilere doğrudan dağıtılmaktadır ve bu doğrudan gönderim bedeli müşterinin talep hacmine göre fiyatlandırılmaktadır.

Modelde kullanılan kümeler, indisler ve değişkenler aşağıdaki gibidir:

Kümeler ve İndisler

N_0 = Tedarikçi/Üretici dahil olmak üzere düğüm kümesi	$\{i,j= 0,1,2,3,...,n\}$
N = Müşteri düğüm kümesi	$\{i,j= 1,2,3,...,n\}$
V =Araç Kümesi	$\{V=1,2,3, 4,\dots v\}.$
K =Çapraz Sevkiyat Merkezi	$\{k=1\}.$
$K_v = m^3$ cinsinden maksimum araç yükleme kapasitesi	$\{v=1,2,3, 4,...v\}.$
$d_j = m^3$ cinsinden talep miktarı	$j \in N$
C_{ij} : i ve j müşterisi arasındaki km cinsinden uzaklık	

S: Çapraz Sevkiyat merkezine atanan aracın kapasitesi

D: Milk-run turunun km cinsinden uzunluğu

L_v : Araç kiralama maliyeti $\{v=1,2,3,4,..v\}$

O: Üretici/tedarikçiden çapraz sevkiyat merkezine sabit araç taşıma maliyeti

W: Milk-run durak maliyeti

T: Çapraz sevkiyat merkezinden müşterilere direkt dağıtım maliyeti

P: Km başına tüketilen birim benzin maliyeti

Karar Değişkenleri

M= Üretici/tedarikçiden çapraz sevkiyat merkezine kadar nakliye için kullanılan kamyon sayısı

$X_{ijv} = \begin{cases} 1 & v \text{ aracı müşteri } i \text{ den müşteri } j \text{ ye dağıtım yapıyorsa} \\ 0 & \text{yok ise} \end{cases}$ $i \in N_0, j \in N_0$ ve

$i \neq j, v \in V$

$Z_{jv} = \begin{cases} 1 & v \text{ aracı direkt müşteri } j \text{ ye dağıtım yapıyor ise} \\ 0 & \text{yok ise} \end{cases}$ $j \in N$ ve $v \in V$

$Y_j = \begin{cases} 1 & j. \text{ müşteri talebi çapraz sevkiyat ile dağıtım yapılıyorsa} \\ 0 & \text{yok ise} \end{cases}$ $j \in N$

Amaç Fonksiyonu

Min Z= Direkt sevkiyat maliyeti + Milk-run sevkiyat maliyeti + Çapraz sevkiyat maliyeti

- Direkt Sevkiyat Maliyeti = Araç kiralama fiyatı + toplam tüketilen benzin maliyeti $= \sum_{i=0, j \in N} \sum_{v \in V} Z_{jv} \cdot (L_v + C_{ij} \cdot P_v)$
- Milk-run sevkiyat Maliyeti = Araç kiralama maliyeti + toplam tüketilen benzin maliyeti + durak maliyeti = $\sum_{i=0, j \in N} \sum_{v \in V} X_{ijv} \cdot (L_v + C_{ij} \cdot P_v) + \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} X_{ijv} \cdot C_{ij} \cdot P_v + \sum_{i \in N_0} \sum_{j \in N} \sum_{v \in V} W \cdot X_{ijv}$
- Çapraz Sevkiyat Maliyeti = Tedarikçi/Üreticiden çapraz sevkiyat merkezine taşıma maliyeti+ çapraz sevkiyat merkezinden müşterilere direkt taşıma maliyeti $= M.O + \sum_{j \in N} T \cdot Y_j$

Optimizasyon Modeli:

$$\text{Min}Z = \sum_{i=0, j \in N} \sum_{v \in V} Z_{jv} \cdot (L_v + C_{ij} \cdot P_v) + \sum_{i=0, j \in N} \sum_{v \in V} X_{ijv} \cdot (L_v + C_{ij} \cdot P_v) + \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} X_{ijv} \cdot C_{ij} \cdot P_v + \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} W \cdot X_{ijv} + M \cdot 0 + \sum_{j \in N} T \cdot Y_j \quad (5.1)$$

$$\sum_{i \in N} \sum_{v \in V} X_{ijv} + \sum_{v \in V} Z_{jv} + Y_j = 1, \quad \forall j \in N, i \neq j \quad (5.2)$$

$$\sum_{i \in N} \sum_{j \in N} d_j \cdot X_{ijv} \leq K_v, \quad \forall v \in V, v=1,2,3,4,..v, i \neq j \quad (5.3)$$

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} X_{ijv} \leq |S| - 1 \quad \forall S \subset N, |S| \geq 2, v = 1,2,3,4,..v \quad (5.4)$$

$$\sum_{v \in V} X_{ijv} + \sum_{v \in V} Z_{iv} + Y_i \leq 1, \quad \forall i \in N, j = 0 \quad (5.5)$$

$$\sum_{v \in V} X_{jiv} + \sum_{v \in V} Z_{iv} + Y_i \leq 1, \quad \forall i \in N, j=0 \quad (5.6)$$

$$\sum_{j \in N} X_{jiv} + \sum_{j \in N} Z_{jv} \leq 1, \quad \forall v \in V, i=0, j \neq 0, i \neq j \quad (5.7)$$

$$\sum_{i \in N} X_{ijv} - \sum_{l \in N} X_{jlv} = 0 \quad \forall j \in N, \forall v \in V, i \neq j, l \neq j \quad (5.8)$$

$$d_i \cdot \sum_{i \in N} Z_{iv} \leq K_v, \quad \forall i \in N, \forall v \in V, v=1,2,3,4,..v \quad (5.9)$$

$$\sum_{j \in N} d_j \cdot Y_j \leq M \cdot S, \quad s \in S \quad (5.10)$$

$$\sum_{i \in N} \sum_{j \in N} C_{ij} \cdot X_{ijv} \leq D, \quad \forall v \in V, i \neq j \quad (5.11)$$

$$\sum_{j \in N} Y_j \leq N \quad (5.12)$$

$$X_{ijv} \in \{0,1\} \quad (5.13)$$

$$Z_{iv} \in \{0,1\} \quad (5.14)$$

$$Y_{iv} \in \{0,1\} \quad (5.15)$$

(5.1), amaç fonksiyonudur ve toplam dağıtım maliyetlerini en aza indirmeyi amaçlar. Kısıt (5.2), müşteri taleplerinin mevcut üç dağıtım stratejisinden biriyle karşılanmasını sağlar. Kısıt (5.3), milk-run turunda taşınan miktarların toplamının araç kapasitesini aşmamasını sağlar. Kısıt (5.4) alt turları ortadan kaldırır. Kısıtlamalar (5.5) ve (5.6), doğrudan sevkiyat, çapraz sevkiyat veya milk-run dağıtım stratejilerinden sadece birinin mevcut olacağını garanti eder.

Kısıt (5.7), aynı aracın hem doğrudan sevkiyat hem de milk-run turu yapmasını kısıtlar. Kısıt (5.8), aynı aracın milk-run turunu tamamladığını garanti eder.

Kısıt (5.9), müşterinin talebini doğrudan sevkiyatla karşılamak için uygun aracın tahsis edilmesini sağlar. Kısıt (5.10), çapraz sevkiyat merkezine gönderilen toplam miktarın müşteri taleplerini karşılayacağını garanti eder. Kısıt (5.11) milk-run rota uzunluğunu sınırlar. Kısıt (5.12), çapraz sevkiyat merkezine yapılan atamanın azami müşteri sayısı kadar olacağını garanti eder. Kısıtlamalar (5.13), (5.14) ve (15), değişkenlerin 0 veya 1 ikili değerlerinden birini almasını sağlar.

5.4 Geliştirilen Hibrit Algoritma ile Problemin Çözümü

Problem için geliştirilen hibrit algoritma, C sharp programlama dili ile kodlanmıştır. Algoritmaya ilişkin kodlar, ekler bölümünde yer almaktadır. Literatürden “genetik algoritmanın” araç rotalama ve çapraz sevkiyat dağıtım stratejisini içeren dağıtım ağı problemlerini çözmek için başarıyla kullanıldığını görüyoruz [56,60,95]. Öte yandan, “Clarke ve Wright”, araç rotalama problemini, [7, 84] başarıyla kullanmıştır. Bu yüzden, Clarke ve Wright’ın Tasarruf algoritması ve genetik algoritma kullanılarak "Dağıtım Stratejisi Seçimi ve Araç Yönlendirme Hibrit Algoritması" (DSSAYHA) adlı yeni bir hibrit algoritması geliştirilmiştir. Bu algoritma, karma bir dağıtım ağı problemini verimli bir şekilde çözmek için literatüre yeni bir hibrit çözüm sunmaktadır.

Hibrit algoritma, minimum maliyeti sağlamak için uygun dağıtım stratejisini seçmekte ve optimal rotaları belirlemektedir. İki alt algoritmadan oluşmaktadır: Genetik ve Modifiye Edilmiş Tasarruf Algoritma. İlk olarak, genetik algoritma ile müşterilere rastgele dağıtım stratejileri atanmaktadır. Tasarruf algoritması ile milk-run rotaları oluşturulmaktadır.

Aşağıdaki bölümlerde, hibrit algoritmanın akışı ve adımları anlatılacaktır. Problem için çözümler hibrit algoritma ve optimizasyon modeliyle oluşturulacaktır.

5.4.1 Adım 1: Başlangıç Popülasyonun Oluşturulması

Hibrit algoritma ile oluşturulan çözüm modeli için amaç minimum toplam dağıtım maliyetini sağlayan, müşterilere en uygun dağıtım tipi ile araç tipi atamalarını yapmak ve milk-run rotalarını bulmaktır.

Genin kodlanmasında değer kodlama yaklaşımı kullanılmıştır. 1: Çapraz sevkiyat ile taşıma; 2: Direkt Taşıma; 3: Milk-run ile taşıma değerlerini göstermektedir.

Kromozom uzunluğu müşteri sayısının uzunluğu kadardır. [160,161] tarafından, popülasyon büyüklüğü 100 olarak önerilmiştir. Buradan hareketle, popülasyon büyüklüğü $N=100$ olarak belirlenmiştir. Popülasyondaki her kromozoma rassal olarak 1,2,3 değerlerinden biri rassal olarak atanmaktadır. Böylelikle problemin çözümü için başlangıç çözümler oluşturulmaktadır. 10 müşterili bir kromozom örneği Şekil 5.2’de gösterilmiştir.

1	2	3	4	5	6	7	8	9	10
1	3	2	3	2	3	1	1	3	3

Şekil 5.2 Çözüm İçin Kullanılan Kromozom Örneği

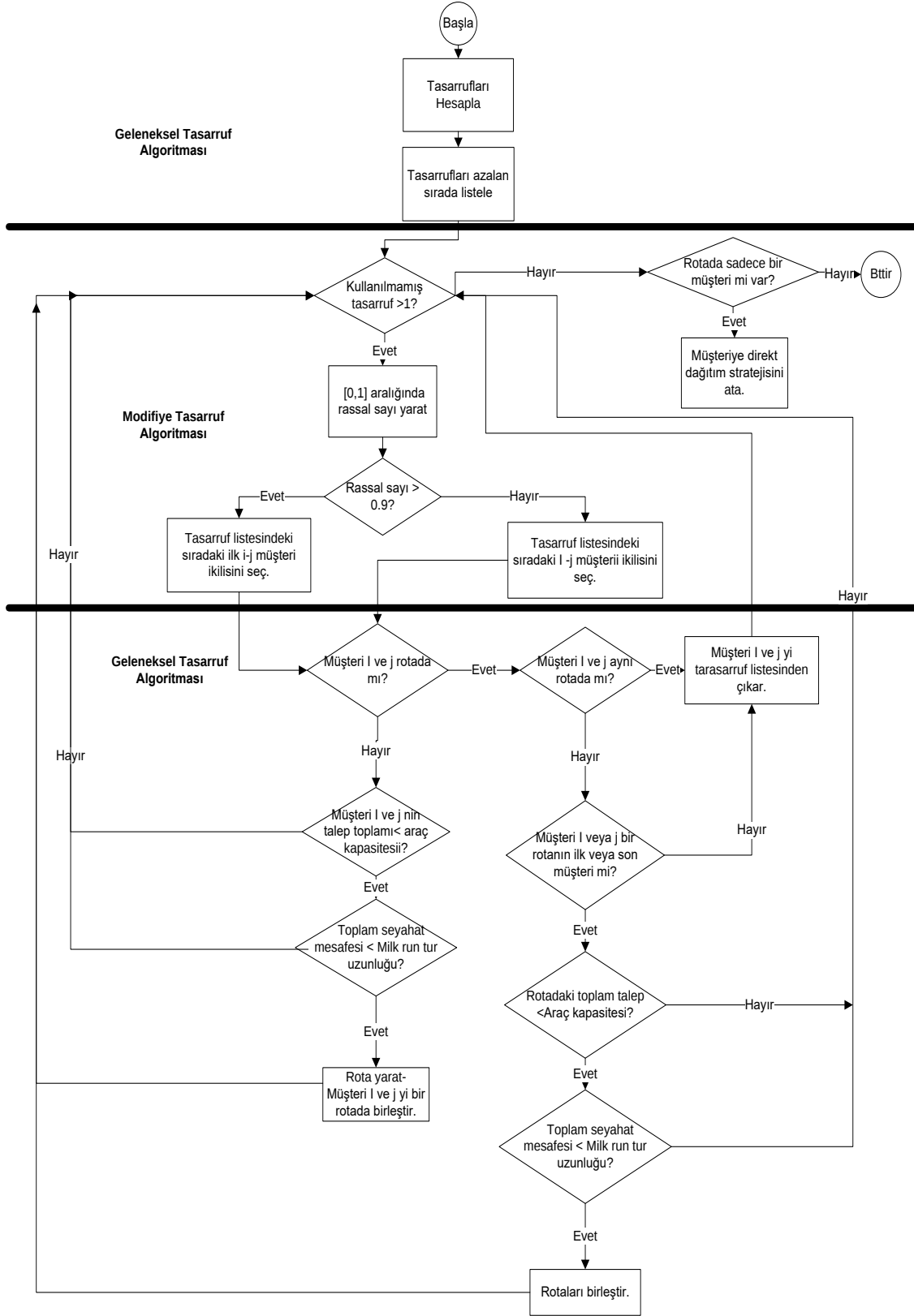
5.4.2 Adım 2: Strateji= Milk-run için Modifiye edilmiş Tasarruf Algoritma Uygulanması

[115], bir depodan teslimat noktalarına teslimatta çeşitli kapasitelere sahip bir araç filosunun en uygun şekilde rotdalanması için basit bir yöntem önermiştir. Geleneksel tasarruf yönteminin rota oluşturmaya başlamadan önce üç temel adımı vardır:

- i ve j müşterileri arasındaki tasarruflar [126]’ün $S_{ij} = d_{i0} + d_{0j} - d_{ij}$ formülü ile hesaplanır. (d_{ij} , i ve j arasındaki seyahat mesafesini ifade etmektedir. 0 depoyu temsil etmektedir.)
- Tasarruflar azalan sırada sıralanır.
- Müşterilerden en büyük tasarruf değerine sahip i ve j ikilisi seçilir.

Geliştirilen modifiye prosedür ise, geleneksel i) ve ii) adımlarıyla başlamaktadır ve iii) adımında farklılık göstermektedir. Maksimum tasarruf sağlayan müşteri ikililerini seçmek yerine, rotaları oluştururken rastgele seçim prosedürü uygulanmaktadır. Böylece, yerel çözümlerden kaçınılarak farklı milk-run rotaları oluşturmak mümkün olmaktadır. Geleneksel tasarruf yönteminde rotada bir müşteriye izin verilmektedir, fakat değiştirilmiş prosedürümüzde rotada bir müşteriye izin verilmemektedir.

Rotada bir müşteri olduğunda, müşteriye doğrudan sevkiyat stratejisi ile hizmet verilmektedir. Önerilen modifiye tasarruf algoritmasının akış şeması Şekil 5.3’te verilmiştir.



Şekil 5.3 Modifiye Tasarruf Algoritması Akışı

Modifiye tasarruf algoritmasının sözde kodu (pseudo code) aşağıda verilmiştir:

Adım 1: [Geleneksel tasarruf algoritması] Kromozomdaki milk-run dağıtım stratejisine atanan her i-j müşteri çifti için tasarrufları hesapla.

Adım 2: [Geleneksel tasarruf algoritması] S_{ij} tasarruf değerlerini azalan şekilde sırala.

Adım 3: [Modifiye tasarruf algoritması]

[0,1] aralığında rasgele sayı oluşturun ve 4. adıma git.

Rotada yalnızca bir müşteri varsa,

- Müşterilere doğrudan sevkiyat yöntemini ata ve milk-run maliyetini hesapla ve prosedürü sonlandır.
- Aksi halde milk-run maliyetini hesapla ve prosedürü sonlandır.

Adım 4: [Modifiye tasarruf algoritması] Üretilen rastgele sayı > 0.9 ise (0.9 eşik değer olarak tanımlanmıştır.),

- Tasarruf listesinde birinci sırada olan müşteri i ve j ikilisini seçin.
- Aksi takdirde, tasarruf listesinde bir sonraki sırada olan müşteri i ve j ikilisini seçin.

Adım 5: [Geleneksel tasarruf algoritması] i ve j müşterileri bir rotanın üyesi ise,

- Müşteri i ve j'nin aynı rotada olup olmadığını kontrol et,
 - i ve j müşterileri tasarruf listesinden çıkar ve 3. adıma git.
 - Aksi halde müşteri i veya j'nin rotadaki ilk veya son müşteri olup olmadığını kontrol et,
 - Eğer toplam araç talebi $<$ araç kapasitesi ise iv. adıma git; aksi takdirde 3. adıma git.
 - Eğer Toplam yol mesafesi $<$ milk-run tur uzunluğu ise, milk-run rotalarını birleştir; aksi takdirde 3. adıma git.
- Müşteri i ve j'nin toplam talebi $<$ araç kapasitesi ise, c adımına git; aksi takdirde 3. adıma git.
- Toplam seyahat mesafesi $<$ milk-run tur uzunluğu ise, yeni rota oluşturun; aksi takdirde 3. adıma git.

5.4.3 Adım 3: Eğer metot= Direkt Sevkiyat ise, Direkt Prosedürü Uygula

Uygun Araç Tipini Ata: Kromozom çözümünde doğrudan sevkiyat stratejisine atanan müşterilere, müşteri talebi miktarına göre uygun araç tipi atanır.

Direkt Sevkiyat Maliyetini Hesapla: Doğrudan nakliye maliyeti hesaplanır. Doğrudan maliyet, araç tipinin sabit maliyetini ve seyahat mesafesi maliyetini içerir.

5.4.4 Adım 4: Eğer metot=Çapraz Sevkiyat ise,Çapraz Sevkiyat Prosedürünü Uygula

Çapraz Sevkiyat Merkezine Aktarılabacak Müşteri Taleplerini Birleştir: Kromozom çözümünde çapraz sevkiyat stratejisine atanan müşteriler için toplam talep hacmi hesaplanır. Çapraz sevkiyat merkezine aktarım için yeterli sayıda araç kullanılır.

Çapraz Sevkiyat Maliyetini Hesapla: Bu maliyet, çapraz sevkiyat merkezine nakliye maliyetini ve çapraz sevkiyat merkezinden müşteriye doğrudan nakliye maliyetini içerir. Çapraz sevkiyat merkezinden müşteriye doğrudan nakliye maliyeti, müşteri talebinin hacmine göre değişen sabit bir maliyettir.

5.4.5 Adım 5: Genetik Algoritma- Uygunluk Değerinin Değerlendirilmesi

Uygunluk fonksiyonu olarak optimizasyon modeli için geliştirilen modelin amaç fonksiyonu kullanılmıştır. Amaç fonksiyonu, 5.3 bölümünde belirtilen optimizasyon modelinin 5.1 formülasyonuna göre hesaplanmıştır. Hibrit algoritma ile oluşturulan çözümler için uygunluk değerleri hesaplanmaktadır, amaç toplam dağıtım maliyetinin minimizasyonu olduğu için maliyetler en küçükten en büyüğe olmak üzere sıralanmaktadır. En düşük maliyet en iyi maliyet olarak saklanmaktadır.

5.4.6 Adım 6: Genetik Algoritma- Yeni Popülasyonu Oluşturma

Seçim: Yeni nesilleri oluşturmak için, kromozom seçim prosedürü olarak [142] tarafından önerilen Rulet Teker seçimi yöntemi kullanılmaktadır. Seçim olasılığı, $p_i = f_i / \sum_{i=1}^N f_i$ formülüne göre hesaplanmaktadır. f_i kromozomun uygunluk değerini, N ise; popülasyon büyüklüğünü ifade eder.

Çaprazlama: En çok kullanılan çaprazlama oran aralığı [0.45, 0.95] olup, mutasyon oranları için de literatürde ortak düzenlemeler bulunmaktadır [161].

Literatürde bu değerler çaprazlama için [0.50, 0.95] arasında mutasyon için ise; [0.05-0.70] arasında değişmektedir [165,166]. Tek nokta çaprazlama kullanılmaktadır. Rassal sayıların üretimi ile kesme noktaları belirlenmektedir. Başlangıç noktalarından kesme noktalarına kadar kromozomlarda yer alan kısımlar karşılıklı olarak yer değiştirilmektedir.

Mutasyon: Çaprazlamadan sonra tek nokta mutasyon yürütülmektedir. Gen rassal olarak seçilmektedir. Rassal sayılar [0,1] aralığında üretilmektedir. Mutasyon için 0.5 değeri eşik değer olarak kabul edilmiştir. Aşağıdaki prosedür uygulanmaktadır:

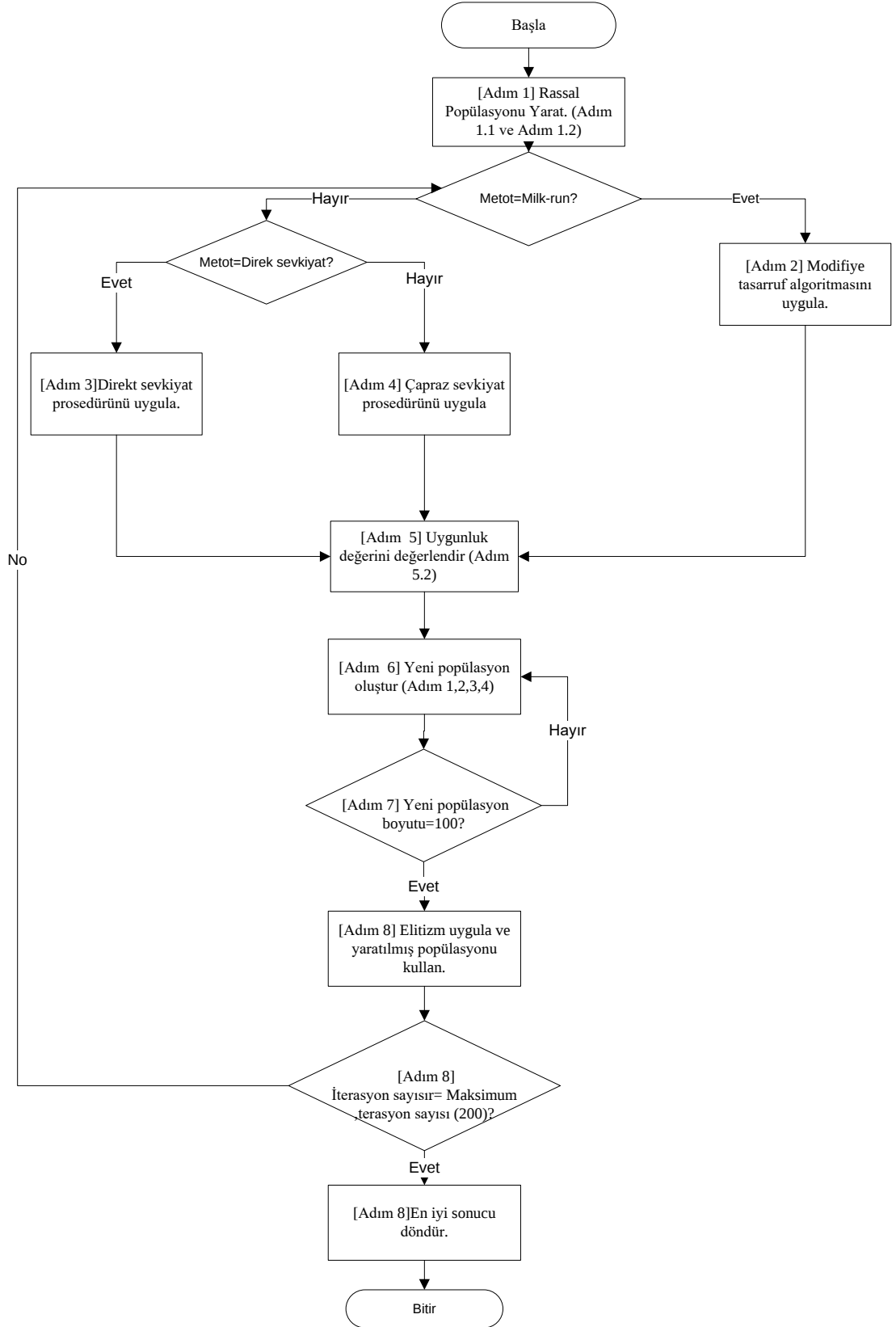
- Eğer genin dağıtım metodu değeri çapraz sevkiyat ve oluşturulan rassal sayı <0.5 ise; dağıtım metodu direkt dağıtım olarak değiştirilir, değilse; milk-run olarak set edilir.
- Eğer genin dağıtım metodu değeri direkt taşıma ve oluşturulan rassal sayı <0.5 ise; dağıtım metodu çapraz sevkiyat olarak değiştirilir, değilse; milk-run olarak set edilir.
- Eğer genin dağıtım metodu değeri milk-run ve oluşturulan rassal sayı <0.5 ise; dağıtım metodu çapraz sevkiyat olarak değiştirilir, değilse; direkt dağıtım olarak set edilir.

5.4.7 Adım 7: Genetik Algoritma: Elitizm Uygula

Bütün iterasyonlarda her bir seçim sırasında en iyi uygunluk değerine sahip kromozom yeni nesillere direkt olarak taşınır. Popülasyonun diğer bireyleri için seçim prosedürü yürütülür.

5.4.8 Adım 8: Genetik Algoritma: Algoritmanın Sonlandırılması

Algoritmanın devam etmesi ya da durması kararını sağlayan kriter, geliştirilen algoritma için, belirlenmiş maksimum iterasyon sayısıdır. Parametrelerin tespiti bölümünde denemeler ve belirlenen iterasyon sayısı verilmektedir. Algoritma için iterasyon sayısı 200 olarak belirlenmiştir. Algoritmanın akışı Şekil 5.4'te gösterilmiştir.



Şekil 5.4 Önerilen Hibrit Algoritma Akışı

Önerilen hibrit algoritmanın sözde kodu (pseduo code) aşağıda verilmiştir:

1. [Adım 1] Rastgele popülasyon Oluşturma

- Popülasyon oluşturmaya popülasyon büyüklüğü 100 olana kadar devam et.
- Popülasyondaki müşterilere rastgele dağıtım stratejisi ata.

2. [Adım 2] Eğer metot = Milk-run ise, modifiye tasarruf algoritması adımlarını uygula.

3. [Adım 3] Eğer metot = doğrudan sevkiyat ise, doğrudan sevkiyat prosedürünü uygula.

4. [Adım 4] Eğer metot = çapraz sevkiyat ise, çapraz sevkiyat prosedürünü uygula.

5. [Adım 5] Uygunluk değerini değerlendir.

- Toplam dağıtım maliyetini hesapla.
- Maliyetleri azalan şekilde sırala.

6. [Adım 6] Yeni Popülasyon Oluşturma

- [Seçim] Uygunluk değerlerine göre iki ana kromozom seç.
- [Çaprazlama] Yavru oluşturmak için tek nokta çaprazlama uygulayın.
- [Mutasyon] Yeni bir yavru oluşturmak için tek noktalı mutasyon uygulayın.
- [Yerleştirme] Yeni yavruları yeni bir popülasyona yerleştir.

7. [Adım 7] Yeni popülasyon oluşturmaya bırak. Eğer popülasyon büyüklüğü = 100 ise, yeni bir yavru oluşturma işlemini tamamla, aksi takdirde 6. adıma geç.

8. [Adım 8] Değiştir- Elitizm uygula ve yeni oluşturulan popülasyonu kullan.

9. [Adım 9] Algoritmayı sonlandır- Eğer yineleme sayısı = maksimum yineleme sayısı ise, algoritmayı tamamla ve en iyi çözümü döndür; aksi takdirde 2. adıma geç.

5.5 Tedarik Zinciri Dağıtım Ağı Probleminin Çözülmesi

Elektronik ürün üreten fabrikanın taşımacılık hizmetleri XYZ firmasının yönetimindedir. Elektronik ürün üreten fabrikaya verilen siparişlerin dağıtımı için XYZ firması günlük olarak planlama yapmaktadır. Problem verileri için elektronik ürün üreten fabrikanın günlük siparişleri kullanılmıştır. Problem setleri, 4-12 müşteri arası küçük büyüklükte veri seti, 20-50 arası orta büyüklükte veri seti, 100 büyüklüğünde büyük veri seti şeklinde ayarlanmıştır.

Küçük büyüklükteki veri seti için 4 müşteriden 12 müşteriye kadar olan planlamada optimizasyon modeli ve genetik algoritma ile çözümler oluşturulmuştur. Genetik algoritmanın çalıştırılmasında literatürden yararlanılarak parametreler düzenlenmiştir. Genetik algoritma her veri seti için 30 kere çalıştırılmıştır. Çeşitli parametrelerin her müşteri veri seti için denenmesi çok uzun zaman gerektirdiğinden, optimum değerleri veren parametrelerin belirlenmesi için deneysel tasarım tekniği olan taguchi yönteminden yararlanılmıştır. 11 müşterili veri seti için taguchi yöntemi ile, optimum parametreler belirlenmiş diğer müşteri veri setleri için de taguchi yöntemi ile elde edilen optimum değerlerle çözümler elde edilmiştir. 12 müşteriden büyük veri setleri için çözümler, çözümlerin elde edilmesi lingo paket program ile çok uzun zaman ve hafıza gerektirdiğinden, genetik algoritma ile oluşturulmuştur.

İki yöntemle bulunan çözümler eşleştirilmiş t testi yöntemiyle değerlendirilmiştir. Optimizasyon modeli ile çözümde lingo paket programı kullanılmıştır, genetik algoritmanın kodlamasında C sharp yazılım dili, istatistiksel testlerin yürütülmesinde minitab ve spss paket programları kullanılmıştır.

Problemden probleme değişmeyen maliyetler ve kapasiteler aşağıda verilmiştir. Milk-run ve direkt taşıma için kullanılan araç kapasitesi ve kiralama maliyetleri Tablo 5.1'de verilmiştir. Çapraz sevkiyat merkezine taşımada yalnızca araç tipi 4 kullanılmaktadır. Tablo 5.2'de tedarikçi/üretici çapraz sevkiyat merkezi arası uzaklık ve milk-run turunda kullanılan durak maliyeti verilmiştir.

Tablo 5. 1 Araç Kapasiteleri ve Maliyetleri

Araç Tipi	Araç kapasitesi (m ³)	Kiralama Maliyetleri (TL)
1	9 m ³	279,6
2	25 m ³	364,3
3	30 m ³	437,3
4	50 m ³	521,7

Tablo 5. 2 Diğer Parametreler

Tedarikçi/Üretici Çapraz Sevkiyat Merkezi Arası Uzaklık (km)	Milk-run Durak Maliyeti (TL)
169	15,60

5.5.1 Tedarik Zinciri Dağıtım Probleminin Küçük Büyüklükteki Veri Seti için Lingo ile Çözülmesi

Küçük büyüklükteki veri seti için 4 müşterili, 5 müşterili, 6 müşterili, 7 müşterili, 8 müşterili, 9 müşterili, 10 müşterili, 11 müşterili ve 12 müşterili dağıtım problemleri çözülmüştür.

Dağıtım Yapılacak 4 Müşteri için Problem Verileri

Dağıtım yapılacak 4 müşterili probleme ilişkin müşteriler arası uzaklıklar Tablo 5.3'te verilmiştir.

Tablo 5. 3 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4
1	0,00	42,50	53,90	39,60
2	42,50	0,00	83,90	20,70
3	53,90	83,90	0,00	63,80
4	39,60	20,70	63,80	0,00

Tablo 5.4'te müşteri tedarikçi/üretici arası uzaklık verilmiştir.

Tablo 5. 4 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4
181,90	202,20	203,50	203,10

Tablo 5.5'te müşteri çapraz sevkiyat arası uzaklık verilmiştir.

Tablo 5. 5 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4
714,80	669,00	725,50	669,60

Tablo 5.6'de müşteri talep miktarları verilmiştir.

Tablo 5. 6 Müşteri Talep Miktarları (m³)

1	2	3	4
15,00	2,24	5,00	7,12

Tablo 5.7'de müşteriler için çapraz sevkiyat merkezinden direkt taşıma maliyetleri verilmiştir.

Tablo 5. 7 Çapraz Sevkiyat Merkezi Direkt Taşıma Maliyeti (TL)

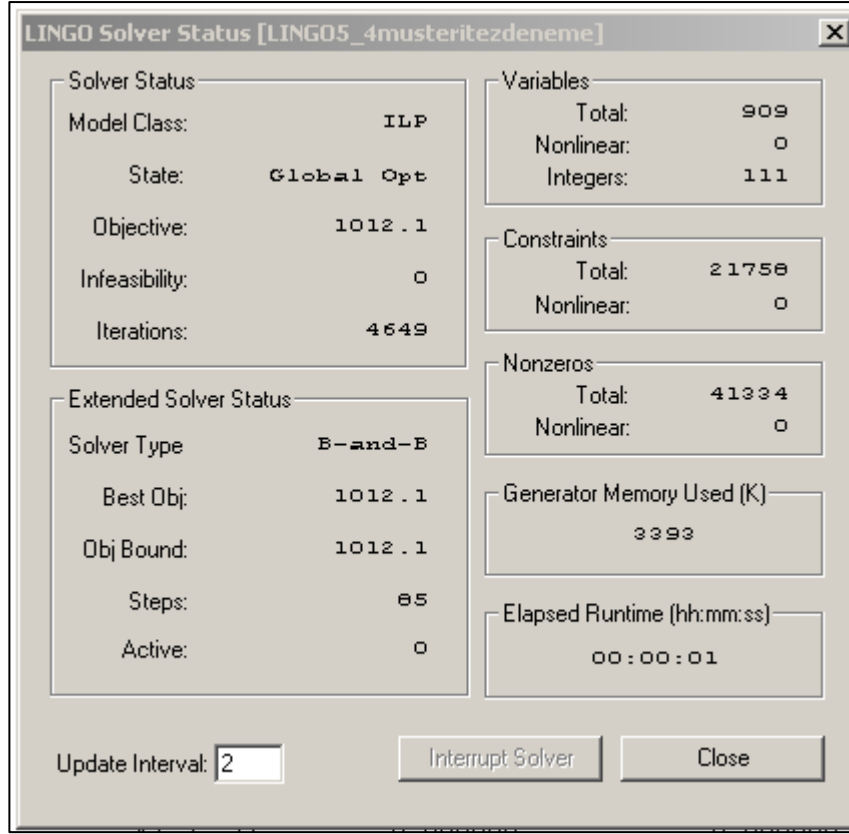
1	2	3	4
45,00	6,720	15,00	21,35

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.5'te gösterilmiştir.

Amaç Değeri: 1012.100

Toplam Çözüm İterasyon Sayısı: 3692

Değişken	Değer	Azaltılmış Maliyet
X(3, 1, 2)	1.000000	0.000000
X(3, 2, 3)	1.000000	7.889503
X(3, 3, 5)	1.000000	0.000000
X(3, 5, 4)	1.000000	23.81050
X(3, 4, 1)	1.000000	768.8000



Şekil 5.5 4 Müşterili Problem için Lingo Sonuç Ekranı

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma Araç Tipi Rota/Müşteri

Milk-run 3 0-1-2-4-3-0

Dağıtım Yapılacak 5 Müşteri için Problem Verileri

Dağıtım yapılacak 5 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.8'de verilmiştir.

Tablo 5. 8 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5
1	0,00	25,50	230,40	230,00	201,00
2	25,50	0,00	249,70	285,00	240,00
3	230,40	249,7	0,00	215,40	127,10
4	230,00	285,00	215,40	0,00	94,30
5	201,00	240,00	127,10	94,30	0,00

5 müşteri problemi ile ilgili müşteri tedarikçi/üretici arası uzaklık Tablo 5.9’da verilmiştir.

Tablo 5. 9 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5
204,10	249,70	341,30	200,00	250,00

Probleme ilişkin, Tablo 5.10’da müşteri çapraz sevkiyat merkezi arası uzaklık verilmiştir.

Tablo 5. 10 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5
785,70	805,00	660,40	494,00	526,00

Tablo 5.11’de probleme ilişkin talepler miktarları metreküp (m³) cinsinden verilmiştir.

Tablo 5. 11 Müşteri Talep Miktarları (m³)

1	2	3	4	5
10,50	18,20	23,00	8,00	6,45

Tablo 5.12’de probleme ilişkin çapraz sevkiyat merkezinden müşterilere direkt taşıma maliyetleri verilmiştir.

Tablo 5. 12 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5
31,50	54,60	69,00	24,00	19,40

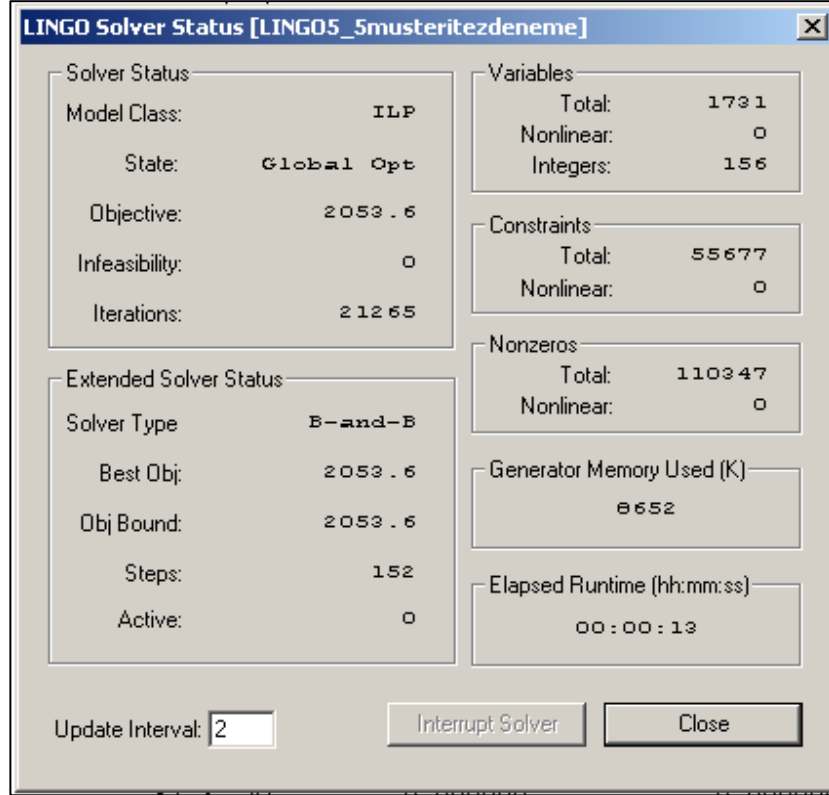
Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.6’da gösterilmiştir.

Amaç Değeri: 2053.600

Toplam Çözüm İterasyon Sayısı: 21265

Değişken	Değer	Azaltılmış Maliyet
C(2, 1, 4)	1.000000	553.6000
X(4, 1, 5)	1.000000	0.000000

X(4, 5, 6)	1.000000	0.000000
X(4, 6, 3)	1.000000	0.000000
X(4, 3, 2)	1.000000	10.90000
X(4, 2, 1)	1.000000	728.2000



Şekil 5.6 5 Müşterili Problem için Lingo Sonuç Ekranı

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma Araç Tipi Rota/Müşteri

Milk-run 4 0-4-5-2-1-0

Direkt 2 3

Dağıtım Yapılacak 6 Müşteri için Problem Verileri

Dağıtım yapılacak 6 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.13'te verilmiştir.

Tablo 5. 13 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5	6
1	0	300	190	207,80	265,60	230
2	300	0	60,80	250	208,70	60
3	190	60,80	0	200	148,80	207,50
4	207,80	250	200	0	195,30	252,80
5	265,60	208,70	148,80	195,30	0	269,60
6	230	60	207,50	252,80	269,60	0

Müşteri tedarikçi/üretici arası uzaklık Tablo 5.14'te verilmiştir.

Tablo 5. 14 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6
775,30	207,80	216,70	578,60	354,50	419,40

Müşteri çapraz sevkiyat merkezi arası uzaklıklar kilometre cinsinden Tablo 5.15'te verilmiştir.

Tablo 5. 15 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6
50	50	30	55	135	90

Müşteri talep miktarları Tablo 5.16'da metreküp (m³) cinsinden verilmiştir.

Tablo 5. 16 Müşteri Talep Miktarları (m³)

1	2	3	4	5	6
10	7	12	25	24,40	15

Probleme ilişkin çapraz sevkiyat merkezinden müşterilere taşıma maliyetleri Tablo 5.17'de verilmiştir.

Tablo 5. 17 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

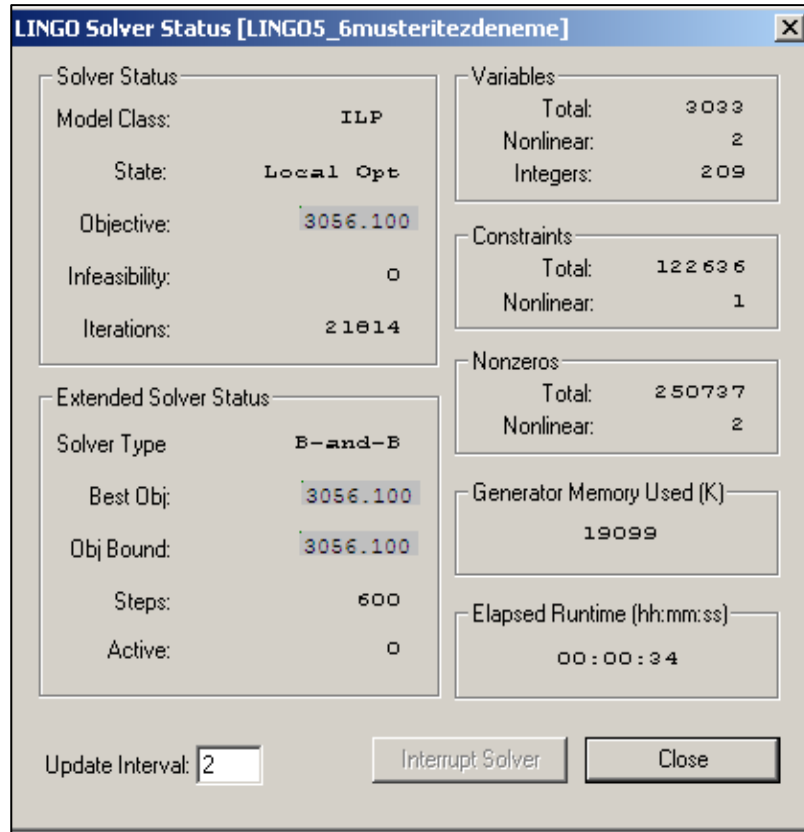
1	2	3	4	5	6
30	21	36	75	73,20	45

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.7'de gösterilmiştir.

Amaç Değeri: 3056.100

Toplam Çözüm İterasyon Sayısı: 2229

Değişken	Değer	Azaltılmış Maliyet
X(4, 1, 3)	1.000000	0.000000
X(4, 3, 4)	1.000000	0.000000
X(4, 4, 6)	1.000000	0.000000
X(4, 6, 1)	1.000000	206.2000
Y(2)	1.000000	-0.3600000
Y(5)	1.000000	128.7500
Y(7)	1.000000	70.41000
N(1)	1.000000	0.000000



Şekil 5.7 6 Müşterili Problem için Lingo Sonuç Ekranı

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-2-3-5-0
Çapraz Sevkiyat	4	1
Çapraz Sevkiyat	4	4
Çapraz Sevkiyat	4	6

Dağıtım Yapılacak 7 Müşteri için Problem Verileri

Dağıtım yapılacak 7 müşterili probleme ilişkin, müşteriler arası uzaklık Tablo 5.18’de verilmiştir.

Tablo 5. 18 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5	6	7
1	0	100	200	209	250	120	30
2	100	0	10	100	120	220	40
3	200	10	0	130	40	240	70
4	209	100	130	0	45	250	100
5	250	120	40	45	0	290	120
6	120	220	240	250	290	0	225
7	30	40	70	100	120	225	0

Müşteri tedarikçi/üretici arası uzaklıkları Tablo 5.19’da verilmiştir.

Tablo 5. 19 Müşterilerin Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6	7
200	300	100	150	250	100	150

Müşteri çapraz sevkiyat merkezi arası uzaklıkları Tablo 5.20’da verilmiştir.

Tablo 5. 20 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7
715,10	673,30	712,60	746,40	828,10	734,40	745,50

Müşteri talep miktarları Tablo 5.21’de verilmiştir.

Tablo 5. 21 Müşteri Talep Miktarları(m³)

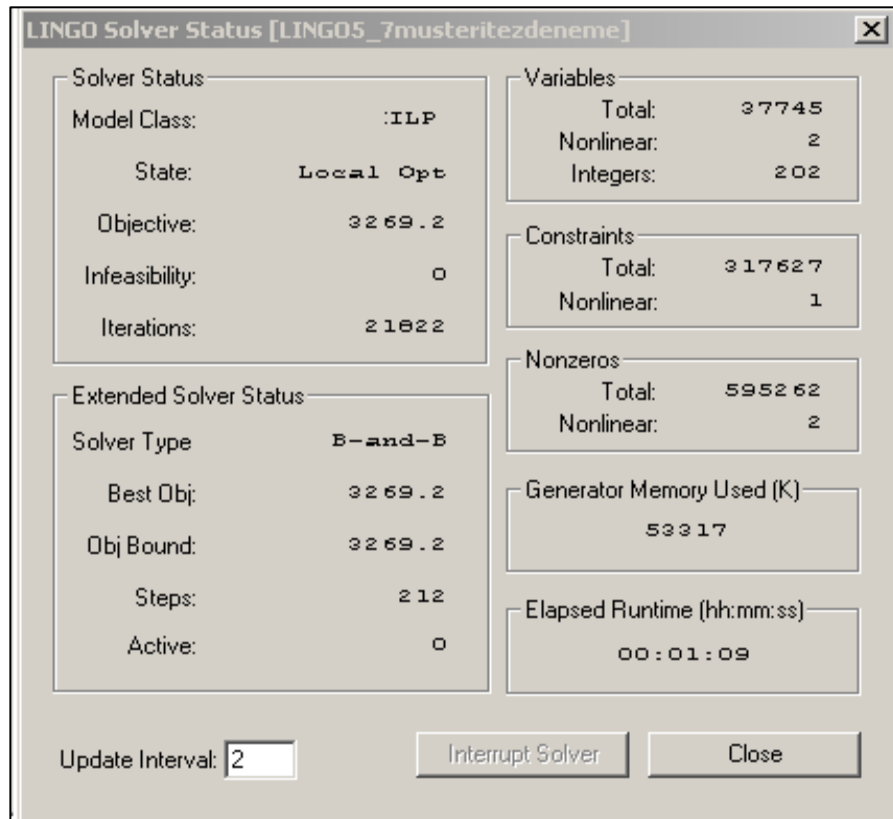
1	2	3	4	5	6	7
4,00	43,00	5,00	50,00	49,90	10,00	30,00

Müşterilerin çapraz sevkiyat merkezinden müşterilere taşıma maliyetleri Tablo 5.22’de verilmiştir.

Tablo 5. 22 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7
12	129	15	150	149,70	30	90

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.8’de gösterilmiştir.



Şekil 5.8 7 Müşterili Problem İçin Lingo Sonuç Ekranı

Amaç Değeri: 3269.200

Toplam Çözüm İterasyon Sayısı: 21822

Amaç	Değer
C(4, 1, 3)	1.000000
C(4, 1, 5)	1.000000
C(4, 1, 6)	1.000000
X(4, 1, 7)	1.000000
X(4, 7, 2)	1.000000
X(4, 2, 8)	1.000000
X(4, 8, 4)	1.000000
X(4, 4, 1)	1.000000

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-6-1-7-3-0
Direkt	4	5

Dağıtım Yapılacak 8 Müşteri için Problem Verileri

Dağıtım Yapılacak 8 müşterili probleme ilişkin, müşteriler arası uzaklık Tablo 5.23'te verilmiştir.

Tablo 5. 23 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5	6	7	8
1	0,00	109	216,10	120	222,40	221,30	123,10	116,50
2	109	0,00	191,50	179,40	225,90	198,10	207,20	113,60
3	216,10	191,50	0	135	66	18	133,60	100,10
4	120	179,40	135	0	230	210	240	214,90
5	222,40	225,90	66	230	0	55,20	122,10	133,30
6	221,30	198,10	18	210	55,20	0	129,20	105,30
7	123,10	207,20	133,60	240	122,10	129,20	0	113,40
8	116,50	113,60	100,10	214,90	133,30	105,30	113,40	0

Müşterilerin tedarikçi/üretici arası uzaklıkları Tablo 5.24'te verilmiştir.

Tablo 5. 24 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6	7	8
200	210	130	140	175	210	170	202

Müşterilerin çapraz sevkiyat merkezine uzaklıkları Tablo 5.25'te verilmiştir.

Tablo 5. 25 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7	8
563,80	526,30	713,40	617,10	746,40	718,60	689,60	635,60

Müşteri talep miktarları Tablo 5.26'da verilmiştir.

Tablo 5. 26 Müşteri Talep Miktarları (m³)

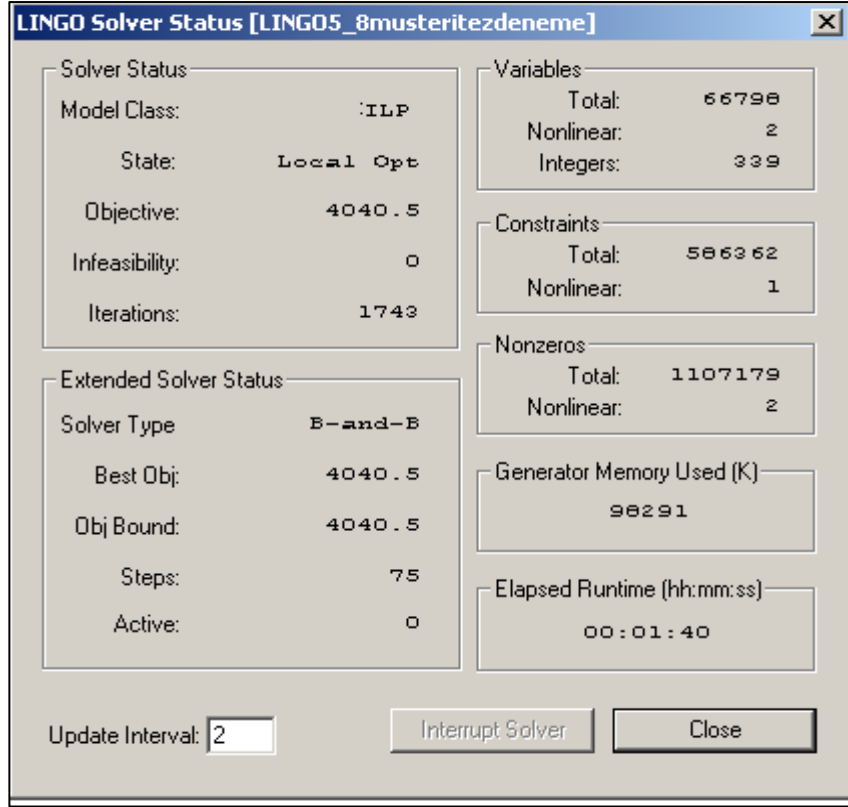
1	2	3	4	5	6	7	8
32	2	50	7	49,90	50	50	5

Çapraz sevkiyat merkezinden müşterilere direkt taşıma maliyeti Tablo 5.27'de verilmiştir.

Tablo 5. 27 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8
96	6	150	21	149,70	150	150	15

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.9'da gösterilmiştir.



Şekil 5.9 8 Müşterili Problem İçin Lingo Sonuç Ekranı

Amaç Değeri: 4040.500

Toplam Çözüm İterasyon Sayısı:1743

Amaç	Değer
C(4, 1, 4)	1.000000
C(4, 1, 6)	1.000000
C(4, 1, 7)	1.000000
C(4, 1, 8)	1.000000
X(4, 1, 5)	1.000000
X(4, 5, 2)	1.000000
X(4, 2, 3)	1.000000
X(4, 3, 9)	1.000000
X(4, 9, 1)	1.000000

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma Araç Tipi Rota/Müşteri

Milk-run 4 0-4-1-2-8-0

Direkt 4 3

Direkt 4 5

Direkt 4 6

Direkt 4 7

Dağıtım Yapılacak 9 Müşteri için Problem Verileri

Dağıtım yapılacak 9 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.28'de verilmiştir.

Tablo 5. 28 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5	6	7	8	9
1	0	109	216,10	187	222,40	221,30	123,10	116,50	116,10
2	109	0	191,50	179,40	225,90	198,10	207,20	113,60	228,30
3	216,10	191,50	0,00	210	66	18	133,60	100,10	215,80
4	187	179,40	210	0,00	140	135	124	214,90	134,00
5	222,40	225,90	66	140	0,00	55,20	122,10	133,30	176,60
6	221,30	198,10	18	135	55,20	0	129,20	105,30	221,40
7	123,10	207,20	133,60	124,00	122,10	129,20	0,00	113,40	85,40
8	116,50	113,60	100,10	214,9	133,30	105,30	113,40	0,00	196,80
9	116,10	228,30	215,80	134,00	176,60	221,40	85,40	196,80	0,00

Müşteri tedarikçi/üretici arası uzaklıkları Tablo 5.29'da verilmiştir.

Tablo 5. 29 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9
200	250	230	240	210	185	270	201	202,2

Müşterilerin çapraz sevkiyat merkezine uzaklıkları Tablo 5.30'da verilmiştir.

Tablo 5. 30 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9
563,80	560	713,40	617,10	746,40	718,60	105	635,60	580

Müşterilerin talep miktarları Tablo 5.31’de verilmiştir.

Tablo 5. 31 Müşteri Talep Miktarları (m³)

1	2	3	4	5	6	7	8	9
25	50	35	11	21	12	50	45	50

Çapraz sevkiyat merkezinden müşterilere taşıma maliyeti Tablo 5.32’de verilmiştir.

Tablo 5. 32 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9
75	150	105	31,20	63	36	150	135	150

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.10’da gösterilmiştir.

Amaç Değeri: 5190.600

Toplam Çözüm İterasyon Sayısı: 7899

Değişken Veri

C(4, 1, 5) 1.000000

C(4, 1, 7) 1.000000

C(4, 1, 9) 1.000000

C(4, 1, 10) 1.000000

X(3, 1, 2) 1.000000

X(3, 2, 3) 1.000000

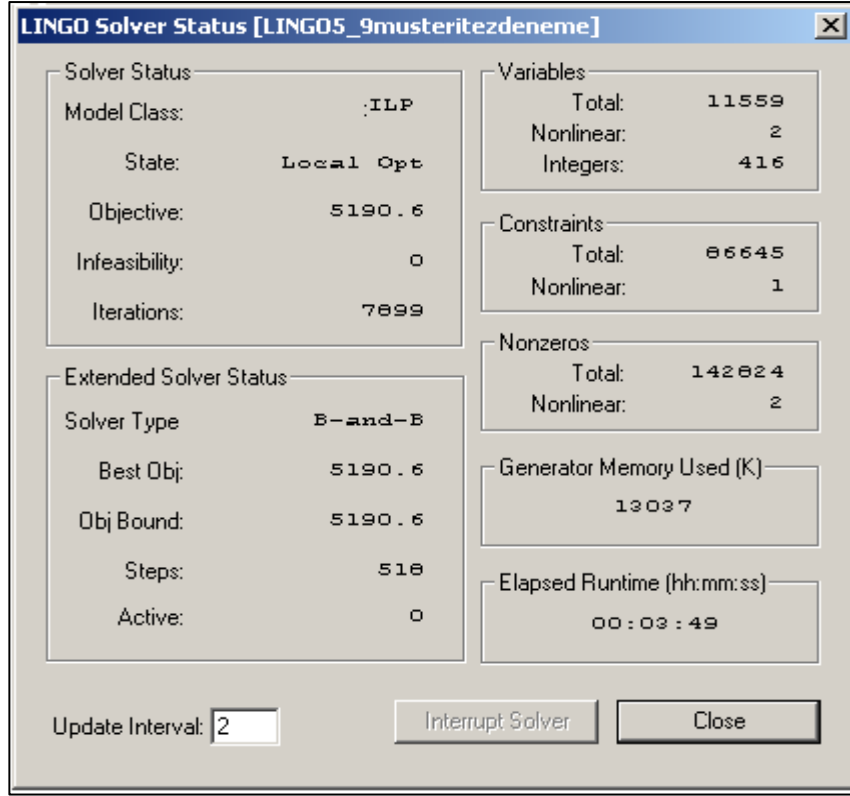
X(3, 3, 1) 1.000000

X(4, 1, 6) 1.000000

X(4, 4, 8) 1.000000

X(4, 6, 4) 1.000000

X(4, 8, 1) 1.000000



Şekil 5.10 9 Müşterili Problem İçin Lingo Sonuç Ekranı

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma Araç Tipi Rota/Müşteri

Milk-run	3	0-1-2-0
Milk-run	4	0-5-3-7-0
Direkt	4	4
Direkt	4	6
Direkt	4	8
Direkt	4	9

Dağıtım Yapılacak 10 Müşteri için Problem Verileri

Dağıtım yapılacak 10 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.33'te verilmiştir.

Tablo 5. 33 Müşteriler Arası Uzaklık (km)

Müşteri	1	2	3	4	5	6	7	8	9	10
1	0	150	150	200	250	210	170	210	190	180
2	150	0	210	210	280	220	230	200	280	285
3	150	210	0	210	210	140	170	190	220	220
4	200	210	210	0	110	100	130	140	145	150
5	250	280	210	110	0	200	240	225	210	260
6	210	220	140	100	200	0	267	185	195	100
7	170	230	170	130	240	267	0	125	175	225
8	210	200	190	140	225	185	125	0	230	60
9	190	280	220	145	210	195	175	230	0	180
10	180	285	220	150	260	100	225	260	180	0

Müşteri tedarikçi/üretici arası uzaklıkları Tablo 5.34'te verilmiştir.

Tablo 5. 34 Müşteri Tedarikçi/Üretici Arası Uzaklık(km)

1	2	3	4	5	6	7	8	9	10
200	210	210	150	170	200	240	210	200	200

Müşteri çapraz sevkiyat merkezi arası uzaklıkları Tablo 5.35'te verilmiştir.

Tablo 5. 35 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9	10
563,80	640	713,40	617,10	746,40	718,60	650	635,60	500	460

Müşteri talep miktarları Tablo 5.36'da verilmiştir.

Tablo 5. 36 Müşteri Talep Miktarları (m³)

1	2	3	4	5	6	7	8	9	10
50	45	25	50	12	50	13	50	10	20

Müşterilerin çapraz sevkiyat merkezinden direkt taşıma maliyetleri Tablo 5.37’de verilmiştir.

Tablo 5. 37 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9	10
150	135	75	150	36	150	39	150	30	60

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.11’de gösterilmiştir.

Amaç Değeri:5985.500

Toplam Çözüm İterasyon Sayısı: 4615

Değişken Değer Azaltılmış Maliyet

C(4, 1, 2) 1.000000 601.1000

C(4, 1, 3) 1.000000 521.1000

C(4, 1, 5) 1.000000 596.1000

C(4, 1, 7) 1.000000 601.1000

C(4, 1, 9) 1.000000 591.1000

X(3, 1, 11) 1.000000 727.3000

X(3, 10, 1) 1.000000 0.000000

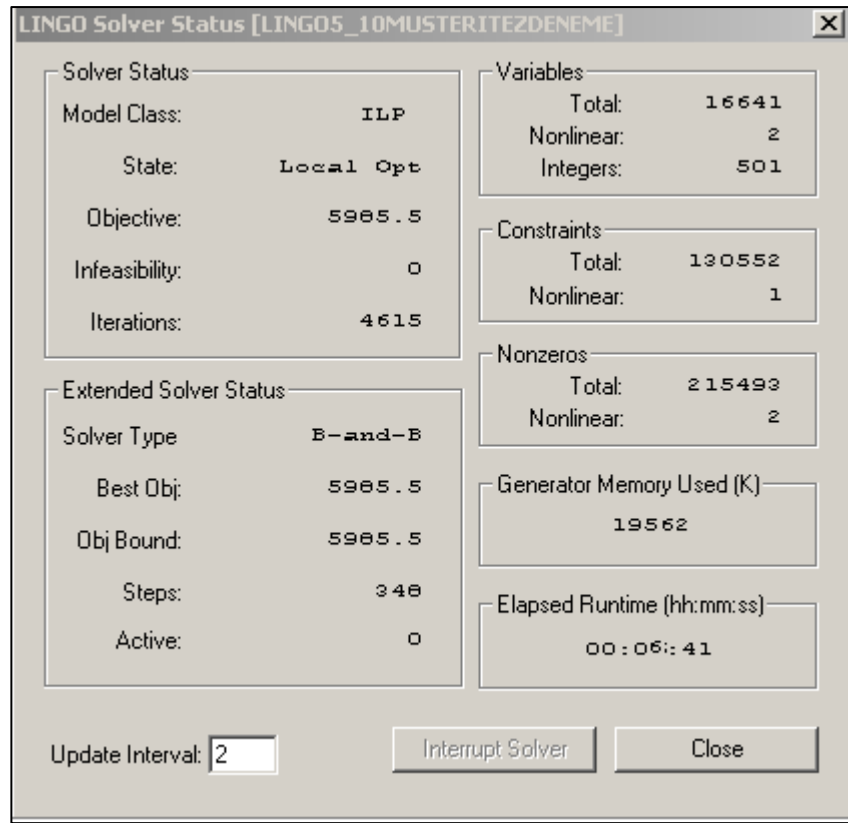
X(3, 11, 10) 1.000000 0.000000

X(4, 1, 8) 1.000000 786.7000

X(4, 4, 6) 1.000000 25.00000

X(4, 6, 1) 1.000000 0.000000

X(4, 8, 4) 1.000000 40.00000



Şekil 5.11 10 Müşterili Problem İçin Lingo Sonuç Ekranı

Çıkan sonuca göre rotalama aşağıdaki gibidir:

Taşıma Araç Tipi Rota/Müşteri

Milk-run	3	0-10-9-0
Milk-run	4	0-7-3-5-0
Direkt	4	1
Direkt	4	2
Direkt	4	4
Direkt	4	6
Direkt	4	8

Dağıtım Yapılacak 11 Müşteri için Problem Verileri

Dağıtım yapılacak 11 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.38’de verilmiştir.

Tablo 5. 38 Müşteriler Arası Uzaklık (km)

	1	2	3	4	5	6	7	8	9	10	11
1	0	298	149,60	117,60	270	250	260	265	270,40	205	277
2	298	0	190,70	190,90	64	267,40	84,40	133,70	84,40	205,90	186,30
3	149,60	190,70	0	90,90	253,60	142,30	135,30	209,60	140,10	175,20	290
4	117,60	190,90	90,90	0	253,80	213	143	209,50	145,80	267,20	290
5	270	64	253,60	253,80	0	229	143,80	210,40	144,80	268,10	184,10
6	250	267,40	142,30	213	229	0	188,40	170	188,40	103,20	213
7	260	84,40	135,30	143,00	143,80	188,40	0	78,70	5,30	126,90	202,50
8	265	133,70	209,60	209,50	210,40	170	78,70	0	78,70	56,90	192,50
9	270,40	84,40	140,10	145,80	144,80	188,40	5,30	78,70	0	126,90	202,50
10	205	205,90	175,20	267,20	268,10	103,20	126,90	56,90	126,90	0	217,30
11	277	186,30	290	290	184,10	213	202,50	192,50	202,50	217,30	0

Müşteri tedarikçi/üretici arası uzaklıklar Tablo 5.39’da verilmiştir.

Tablo 5. 39 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9	10	11
180	195	125	150	230	160	200	201	204	203	130

Müşterilerin çapraz sevkiyat merkezine uzaklıkları Tablo 5.40’da verilmiştir.

Tablo 5. 40 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9	10	11
540	560	650	720	705	410	580	590	615	735	750

Müşteri talep miktarları Tablo 5.41’de verilmiştir.

Tablo 5. 41 Müşteri Talep Miktarları (m³)

1	2	3	4	5	6	7	8	9	10	11
16	50	14	50	50	50	50	12	20	50	13

Müşterilerin çapraz sevkiyat merkezinden direkt taşıma maliyetleri Tablo 5.42’de verilmiştir.

Tablo 5. 42 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9	10	11
72	39	60	105	36	90	39	12	45	90	78

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.12’de gösterilmiştir.

Amaç Değeri: 6365.000

Toplam Çözüm İterasyon Sayısı:5579

Amaç Değer

C(4,1,3) 1.000000

C(4,1,5) 1.000000

C(4,1,6) 1.000000

C(4,1,7) 1.000000

C(4,1,8) 1.000000

C(4,1,11) 1.000000

X(3,1,2) 1.000000

X(3,2,4) 1.000000

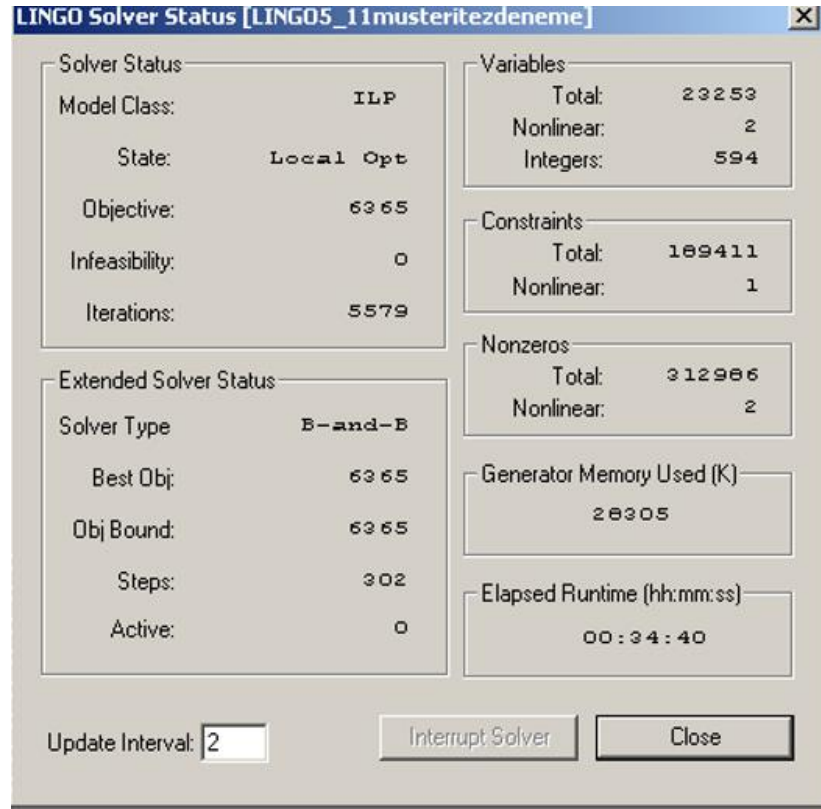
X(3,4,1) 1.000000

X(4,1,10) 1.000000

X(4,9,12) 1.000000

X(4,10,9) 1.000000

X(4,12,1) 1.000000



Şekil 5.12 11 Müşterili Problem İçin Lingo Sonuç Ekranı

Lingo sonucuna göre rotalama aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	3	0-1-3-0
Milk-run	4	0-9-8-11-0
Direkt	4	2
Direkt	4	4
Direkt	4	5
Direkt	4	6
Direkt	4	7
Direkt	4	10

Dağıtım Yapılacak 12 Müşteri için Problem Verileri

Dağıtım yapılacak 12 müşterili probleme ilişkin, müşteriler arası uzaklıklar Tablo 5.43'te verilmiştir.

Tablo 5. 43 Müşteriler Arası Uzaklık (km)

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	161	178	4,90	180,30	191,80	112,50	139	226,30	138,70	198,50	178
2	161,10	0	135,80	160	227,90	254	215,03	287	248	287	290	135,80
3	178	136	0	176	124,70	280	275	213	240	212,80	286,90	200
4	4,90	160	175,50	0	62,70	208,10	265	129	181,50	128,80	214,90	176,40
5	180,30	228	124,70	62,7	0	214,70	205	136	222,40	135,90	221,30	124,7
6	191,80	254	280	208	214,70	0	263,40	77,40	63,60	77,4	16,30	279,50
7	112,50	215	275	265	205	263,40	0	211	298,90	211,20	271	204,40
8	138,70	287	212,80	129	135,90	77,40	211,2	0	114	221	86,30	213,10
9	226,30	248	240	182	222,40	63,60	298,90	114	0	114	55,20	204
10	138,70	287	212,80	129	135,90	77,40	211,20	221	114	0	171,60	282,60
11	198,50	290	286,90	215	221,30	16,3	271	86,30	55,20	171,60	0	287,10
12	178	136	200	176	124,70	279,50	204,40	213	204	282,60	287,10	0

Müşteri tedarikçi/üretici arası uzaklıklar Tablo 5.44'te verilmiştir.

Tablo 5. 44 Müşteri Tedarikçi/Üretici Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9	10	11	12
900	920	800	750	775	730	702	790	710	780	100	710

Müşterilerin çapraz sevkiyat merkezine uzaklıkları Tablo 5.45'te verilmiştir.

Tablo 5. 45 Müşteri Çapraz Sevkiyat Merkezi Arası Uzaklık (km)

1	2	3	4	5	6	7	8	9	10	11	12
100	200	350	150	124	105	200	250	130	145	125	250

Müşterilerin talep miktarları Tablo 5.46'da verilmiştir.

Tablo 5. 46 Müşteri Talep Miktarları (m³)

1	2	3	4	5	6	7	8	9	10	11	12
20	30	30	40	50	50	40	20	50	40	20	10

Müşterilere çapraz sevkiyat merkezinden direkt taşıma maliyetleri Tablo 5.47'de verilmiştir.

Tablo 5. 47 Çapraz Sevkiyat Merkezinden Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9	10	11	12
60	90	90	120	150	150	120	60	90	20	60	30

Problemin lingo paket program ile çözüm sonuçları aşağıdaki gibi olup sonuç ekranı Şekil 5.13'te gösterilmiştir.

Amaç Değeri: 8694.600

Toplam Çözüm İterasyon Sayısı:1721

Değişken Değer

Y(2) 1.000000

Y(3) 1.000000

Y(4) 1.000000

Y(5) 1.000000

Y(6) 1.000000

Y(7) 1.000000

Y(8) 1.000000

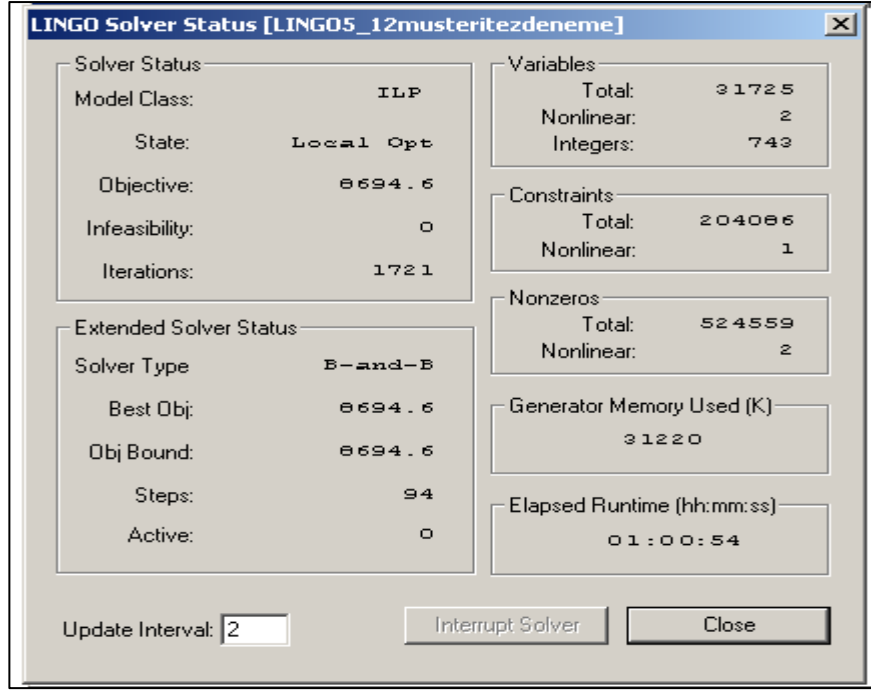
Y(9) 1.000000

Y(10) 1.000000

Y(11) 1.000000

Y(12) 1.000000

Y(13) 1.000000



Şekil 5.13 12 Müşterili Problem İçin Lingo Sonuç Ekranı

Lingo sonucu çıkan rotalama aşağıda gösterilmiştir:

Taşıma	Araç Tipi	Rota/Müşteri
Çapraz sevkiyat	4	1
Çapraz sevkiyat	4	2
Çapraz sevkiyat	4	3
Çapraz sevkiyat	4	4
Çapraz sevkiyat	4	5
Çapraz sevkiyat	4	6
Çapraz sevkiyat	4	7
Çapraz sevkiyat	4	8
Çapraz sevkiyat	4	9
Çapraz sevkiyat	4	10
Çapraz sevkiyat	4	11
Çapraz sevkiyat	4	12

5.5.2 Tedarik Zinciri Dağıtım Probleminin Özelleştirilmiş Genetik Algoritma (GA) ile Çözüm Sonuçları

Bu bölümde 5.5.1'deki veri setleri kullanılarak çözümler oluşturulmuştur. Genetik algoritma parametreleri olan popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı parametreleri için farklı değerler taguchi yöntemi ile değerlendirilmiştir, çözümün kalitesini arttıran en uygun parametre değerleri saptanmıştır. Bulunan en uygun parametre değerleri için veri setleri özelleştirilmiş GA ile 30 kez çalıştırılarak çözümler oluşturulmuştur.

5.5.2.1 Özelleştirilmiş Genetik Algoritma (GA) Parametrelerinin Taguchi Yöntemi ile Belirlenmesi

Genetik algoritma parametrelerinin belirlenmesi için taguchi yönteminden yararlanılmıştır. Veriler MINITAB programı ile analiz edilmiştir.

Taguchi Metodu, ortogonal dizileri kullanarak kontrol edilemeyen faktörlerin etkilerini minimize etmeye çalışan bir deneysel tasarım tekniğidir [167]. Taguchi metodu, istatistiksel bir metottur [168].

Taguchi yöntemi, farklı parametrelerin, farklı seviyeleri arasında en iyi kombinasyonu saptamak için oldukça kullanışlı bir yöntemdir. Her bir parametrenin, her bir seviyesini içeren kombinasyonlar için oldukça fazla deneysel çalışma yapılması gerektiği durumlarda, taguchi yöntemindeki ortogonal dizi tablosu kullanılarak daha az sayıda denemelerle sonuca ulaşmak mümkündür. Taguchi, performans karakteristiklerinin hedef değerden sapmalarını ölçümlemek için kayıp fonksiyon değerinin hesaplanmasını önermektedir. Kayıp fonksiyon değeri, sinyal-gürültü oranı (S/N) değerine dönüştürülmektedir. S/N oranının analizinde performans karakteristikleri için 3 kategori bulunmaktadır: nominal-en iyi, en büyük-en iyi ve en küçük-en iyi [169]. Taguchi yönteminin uygulanmasında izlenen adımlar aşağıda anlatılmıştır.

Kontrol Faktörlerinin ve Seviyelerinin Belirlenmesi: Bu aşamada, performans karakteristiğini etkileyeceği düşünülen faktörler ve seviyeleri belirlenmiştir.

Çalışma için popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranı kontrol faktörleridir. Her faktör için 5 farklı seviye belirlenmiştir. Tablo 5.48’de faktörler için belirlenen seviyeler görülmektedir.

Tablo 5. 48 Değişik Seviyelerde Belirlenen GA Parametreleri

Faktörler	Seviye 1	Seviye 2	Seviye 3	Seviye 4	Seviye 5
Popülasyon Büyüklüğü	20	30	40	50	100
Çaprazlama Oranı	0,5	0,6	0,7	0,8	0,9
Mutasyon Oranı	0,05	0,25	0,45	0,5	0,7

Ortogonal Dizinin Seçilmesi: Ortogonal düzenin en önemli özelliği, birçok faktörün minimum sayıda test ile değerlendirilmesini sağlaması ve klasik metottan farklı olarak faktör kademelerini teker teker değiştirmek yerine eş zamanlı olarak değiştirmeyi önermesidir.

Taguchi metodunda seviye sayılarına göre ortogonal dizi seçim tablosu Şekil 5.14’te gösterilmektedir.

SEVIYE SAYISI												
PARAMETRE SAYISI	2		L4	3		L9	4		L' 16	5		L25
	P=2	S=2		P=2	S=3		P=2	S=4		P=2	S=5	
	P=3	S=2		P=3	S=3		P=3	S=4		P=3	S=5	
	P=4	S=2		P=4	S=3		P=4	S=4		P=4	S=5	
	P=5	S=2	L8	P=5	S=3	L18	P=5	S=4	L' 32	P=5	S=5	L50
	P=6	S=2		P=6	S=3		P=6	S=4		P=6	S=5	
	P=7	S=2		P=7	S=3		P=7	S=4		P=7	S=5	
	P=8	S=2		P=8	S=3		P=8	S=4		P=8	S=5	
	P=9	S=2	L11	P=9	S=3	L27	P=9	S=4	L' 32	P=9	S=5	L50
	P=10	S=2		P=10	S=3		P=10	S=4		P=10	S=5	
	P=11	S=2		P=11	S=3					P=11	S=5	
	P=12	S=2		P=12	S=3					P=12	S=5	
	P=13	S=2	L16	P=13	S=3	L36			L' 32			L50
	P=14	S=2		P=14	S=3							
	P=15	S=2		P=15	S=3							
	P=16	S=2		P=16	S=3							
	P=17	S=2	L32	P=17	S=3	L36			L' 32			L50
	P=18	S=2		P=18	S=3							
	P=19	S=2		P=19	S=3							
	P=20	S=2		P=20	S=3							
P=21	S=2	P=21		S=3								
P=22	S=2	P=22		S=3								
P=23	S=2	P=23		S=3								
P=24	S=2											
P=25	S=2											
P=26	S=2											
P=27	S=2											
P=28	S=2											
P=29	S=2											
P=30	S=2											
P=31	S=2											

Şekil 5.14 Taguchi Ortogonal Dizi Seçim Tablosu [170,171]

Bu çalışmada, Şekil 5.14 taguchi ortogonal dizi seçim tablosundan yararlanılarak, 3 faktör 5 seviye için L25 ortogonal dizi kullanılmıştır.

Kayıp Fonksiyon ve S/N Oranının Hesaplanması: Özelleştirilmiş GA, amaç fonksiyonu değeri olan Z değerinin optimize edilmesini sağlamaya çalışmaktadır.

Bu nedenle amaç fonksiyonu değeri, kayıp fonksiyon değerinin hesaplanmasında kullanılmıştır. Amaç fonksiyonu bu çalışma için en küçük-en iyidir. S/N oranı da en küçük en iyi olduğu durumda denklem (5.17) kullanılarak hesaplanmaktadır.

$$S/N = -10 \log \left(\frac{1}{n} \sum_{i=1}^n y_i^2 \right) \quad (5.17)$$

Deneylerin Yapılması ve Analizi: Taguchi metodu deneyleri L25 ortogonal dizi ile belirlenmiştir. Tablo 5.49'da Taguchi metodu ile belirlenen deneyler gösterilmiştir.

Tablo 5. 49 Taguchi Metodu ile Belirlenen Deneyler

Deney No	Popülasyon Büyüklüğü	Çaprazlama Oranı	Mutasyon Oranı
1	20	0,5	0,05
2	20	0,6	0,25
3	20	0,7	0,45
4	20	0,8	0,5
5	20	0,9	0,7
6	30	0,5	0,25
7	30	0,6	0,45
8	30	0,7	0,5
9	30	0,8	0,7
10	30	0,9	0,05
11	40	0,5	0,45
12	40	0,6	0,5
13	40	0,7	0,7
14	40	0,8	0,05
15	40	0,9	0,25
16	50	0,5	0,5
17	50	0,6	0,7
18	50	0,7	0,05
19	50	0,8	0,25
20	50	0,9	0,45
21	100	0,5	0,7
22	100	0,6	0,05
23	100	0,7	0,25
24	100	0,8	0,45
25	100	0,9	0,5

Bölüm 5.5.1'deki 11 müşterili veri seti için her deney özelleştirilmiş algoritma ile 30 kez tekrar edilerek sonuçlar oluşturulmuştur. Tablo 5.50'de her deneyin 30 kez tekrarlanması ile elde edilen çözümler görülmektedir.

Tablo 5. 50 Taguchi Deneyleri Özelleştirilmiş GA Çözümleri

No	1	2	3	4	5	6	7	8	9	10
1	6065,60	6179,50	6150,60	5985,50	6179,50	6179,50	6179,50	6065,60	6115,60	5985,50
2	6179,50	6179,50	5985,50	6065,60	6179,50	6179,50	5985,50	5985,50	6179,50	6050,60
3	6179,50	5985,50	6179,50	6179,50	6179,50	6000,60	6050,60	5985,50	6000,60	6179,50
4	5985,50	6179,50	6179,50	5985,50	6179,50	6179,50	5985,50	5985,50	6179,50	6179,50
5	5985,50	6179,50	6050,60	6179,50	5985,50	6179,50	6179,50	6179,50	6000,60	5985,50
6	6179,50	5985,50	5985,50	6085,60	6085,60	6000,60	6179,50	5985,50	6000,60	6050,60
7	6179,50	6179,50	6000,60	5985,50	5985,50	6179,50	5985,50	5985,50	6179,50	6000,60
8	5985,50	6000,60	5985,50	6179,50	5985,50	5985,50	5985,50	5985,50	5985,50	6000,60
9	5985,50	5985,50	6179,50	5985,50	6000,60	6085,60	5985,50	5985,50	6065,60	5985,50
10	6179,50	6179,50	6179,50	6179,50	6000,60	6791,60	5985,50	6070,60	6179,50	6000,60
11	6000,60	6070,60	6179,50	6179,50	6000,60	5985,50	5985,50	6000,60	6179,50	5985,50
12	5985,50	5985,50	5985,50	6179,50	6179,50	6179,50	6000,60	5985,50	5985,50	6000,60
13	6050,60	5985,50	5985,50	6179,50	5985,50	5985,50	5985,50	5985,50	5985,50	6000,60
14	6065,60	6179,50	6179,50	6179,50	6179,50	6085,60	6179,50	6179,50	6179,50	6179,50
15	6179,50	6179,50	5985,50	5985,50	6150,60	6115,60	6065,60	6179,50	5985,50	6070,60
16	5985,50	5985,50	6000,60	6000,60	5985,50	6000,60	6000,60	6050,60	5985,50	6000,60
17	6179,50	6179,50	6000,60	6065,60	6000,60	5985,50	5985,50	6000,60	5985,50	6070,60
18	6000,60	6179,50	6179,50	6000,60	6115,60	6050,60	6150,60	6179,50	6179,50	5985,50
19	5985,50	6115,60	6065,60	6179,50	6065,60	5985,50	6050,60	6179,50	6179,50	6070,60
20	6120,60	6065,60	6065,60	6179,50	5985,50	6179,50	6000,60	6179,50	6065,60	5985,50
21	6000,60	6050,60	6050,60	6000,60	6000,60	5985,50	6000,60	6000,60	5985,50	6000,60
22	5985,50	6179,50	6085,60	6179,50	6179,50	6050,60	6000,60	6065,60	6179,50	6000,60
23	6179,50	6179,50	6000,60	5985,50	5985,50	6179,50	6150,60	6050,60	6000,60	6070,60
24	6065,60	6000,60	6050,60	5985,50	6050,60	6179,50	6000,60	6000,60	6000,60	6000,60
25	6000,60	5985,50	6050,60	6000,60	6050,60	6065,60	5985,50	6000,60	6000,60	6000,60

Tablo 5. 50 Taguchi Deneyleri Özelleştirilmiş GA Çözümleri(devamı)

No	11	12	13	14	15	16	17	18	19	20
1	6179,50	6179,50	6179,50	6115,60	5985,50	6179,50	6179,50	6179,50	5985,50	6050,60
2	6179,50	5985,50	5985,50	6050,60	6179,50	6179,50	6179,50	6179,50	6179,50	6065,60
3	5985,50	6179,50	6179,50	6179,50	6179,50	6179,50	5985,50	6179,50	6179,50	6179,50
4	5985,50	6000,60	6179,50	5985,50	6179,50	5985,50	6179,50	6179,50	6179,50	6179,50
5	6179,50	6000,60	5985,50	6179,50	6179,50	6000,60	6179,50	5985,50	6179,50	6179,50
6	6179,50	6179,50	6070,60	6179,50	6115,60	6179,50	6070,60	6065,60	6115,60	6179,50
7	5985,50	5985,50	5985,50	6179,50	5985,50	5985,50	5985,50	6000,60	5985,50	6179,50
8	5985,50	6179,50	5985,50	5985,50	5985,50	5985,50	6000,60	5985,50	5985,50	5985,50
9	6000,60	6179,50	6000,60	6120,60	6000,60	6000,60	5985,50	5985,50	5985,50	6179,50
10	6179,50	5985,50	5985,50	5985,50	6179,50	5985,50	6179,50	6000,60	6179,50	5985,50
11	6179,50	6000,60	5985,50	5985,50	6000,60	5985,50	6179,50	5985,50	6000,60	5985,50
12	5985,50	5985,50	5985,50	6179,50	5985,50	5985,50	5985,50	5985,50	5985,50	6000,60
13	5985,50	6150,60	6179,50	5985,50	5985,50	5985,50	6179,50	6000,60	5985,50	6179,50
14	6179,50	6179,50	5985,50	6179,50	5985,50	6070,60	6115,60	6179,50	5985,50	5985,50
15	6179,50	6000,60	5985,50	6070,60	6050,60	6000,60	6179,50	5985,50	6050,60	6179,50
16	6050,60	5985,50	6000,60	6179,50	6000,60	6000,60	5985,50	6085,60	5985,50	5985,50
17	5985,50	5985,50	5985,50	6000,60	6179,50	6000,60	5985,50	6000,60	5985,50	6000,60
18	6179,50	6179,50	6065,60	6065,60	6179,50	6179,50	6179,50	6150,60	6179,50	6179,50
19	6179,50	6070,60	6179,50	6050,60	6179,50	6179,50	6115,60	6179,50	6179,50	6065,60
20	6065,60	5985,50	6050,60	6179,50	6179,50	6179,50	6050,60	5985,50	6000,60	6000,60
21	6000,60	6000,60	6000,60	5985,50	6000,60	6150,60	5985,50	6065,60	5985,50	6050,60
22	6179,50	6120,60	5985,50	6065,60	6179,50	6150,60	6065,60	5985,50	5985,50	6179,50
23	6179,50	6000,60	6050,60	6065,60	5985,50	6065,60	5985,50	6050,60	6179,50	6050,60
24	6000,60	6000,60	6065,60	5985,50	5985,50	6179,50	6050,60	6065,60	6065,60	6050,60
25	6050,60	6050,60	6000,60	5985,50	6179,50	6000,60	5985,50	6179,50	6050,60	6065,60

Tablo 5. 50 Taguchi Deneyleri Özelleştirilmiş GA Çözümleri(devamı)

No	21	22	23	24	25	26	27	28	29	30
1	6179,50	6179,50	6070,60	6179,50	6179,50	6070,60	6179,50	5985,50	6179,50	6179,50
2	6000,60	6000,60	6179,50	6115,60	6000,60	6179,50	5985,50	6179,50	6179,50	5985,50
3	5985,50	5985,50	6179,50	6179,50	6000,60	6179,50	5985,50	6179,50	6000,60	6179,50
4	6179,50	6179,50	6179,50	5985,50	6179,50	6179,50	5985,50	5985,50	5985,50	6179,50
5	5985,50	6000,60	5985,50	5985,50	6179,50	5985,50	6179,50	6000,60	6000,60	6000,60
6	6179,50	6000,60	6179,50	6179,50	5985,50	5985,50	5985,50	6070,60	6070,60	6050,60
7	6179,50	6000,60	5985,50	6179,50	6000,60	6179,50	6179,50	6000,60	5985,50	6000,60
8	5985,50	5985,50	6179,50	6179,50	6179,50	6179,50	6150,60	6000,60	5985,50	5985,50
9	6050,60	5985,50	6179,50	5985,50	6000,60	5985,50	5985,50	6000,60	5985,50	6070,60
10	5985,50	5985,50	6179,50	6179,50	5985,50	6179,50	6179,50	6000,60	5985,50	6179,50
11	6000,60	5985,50	6000,60	6050,60	5985,50	5985,50	5985,50	5985,50	6050,60	5985,50
12	5985,50	5985,50	6179,50	6050,60	6179,50	5985,50	5985,50	6000,60	6000,60	6000,60
13	5985,50	5985,50	6000,60	5985,50	6000,60	6150,60	6070,60	6065,60	6065,60	5985,50
14	6179,50	5985,50	6179,50	5985,50	6000,60	6085,60	6179,50	6065,60	6065,60	6179,50
15	6150,60	6179,50	6179,50	6179,50	6179,50	6085,60	6065,60	6179,50	6179,50	6050,60
16	5985,50	5985,50	6000,60	5985,50	6000,60	6000,60	5985,50	6179,50	5985,50	5985,50
17	6000,60	5985,50	5985,50	6000,60	6000,60	6050,60	5985,50	6000,60	5985,50	5985,50
18	6179,50	6179,50	6179,50	6179,50	6179,50	6000,60	6179,50	6179,50	6179,50	6115,60
19	6050,60	6000,60	5985,50	6050,60	6179,50	6070,60	6179,50	6179,50	6179,50	6070,60
20	6179,50	6000,60	6050,60	6179,50	6000,60	6115,60	5985,50	6179,50	6000,60	6070,60
21	6000,60	6000,60	5985,50	5985,50	6050,60	5985,50	5985,50	5985,50	5985,50	5985,50
22	6070,60	6179,50	6179,50	6000,60	6179,50	6179,50	6115,60	6000,60	5985,50	5985,50
23	6050,60	6070,60	6000,60	6179,50	6179,50	6150,60	6085,60	6085,60	6179,50	5985,50
24	6000,60	6050,60	6085,60	5985,50	6179,50	6050,60	6050,60	6000,60	6050,60	6065,60
25	6000,60	6000,60	6000,60	6179,50	6000,60	6000,60	6070,60	6050,60	6000,60	6000,60

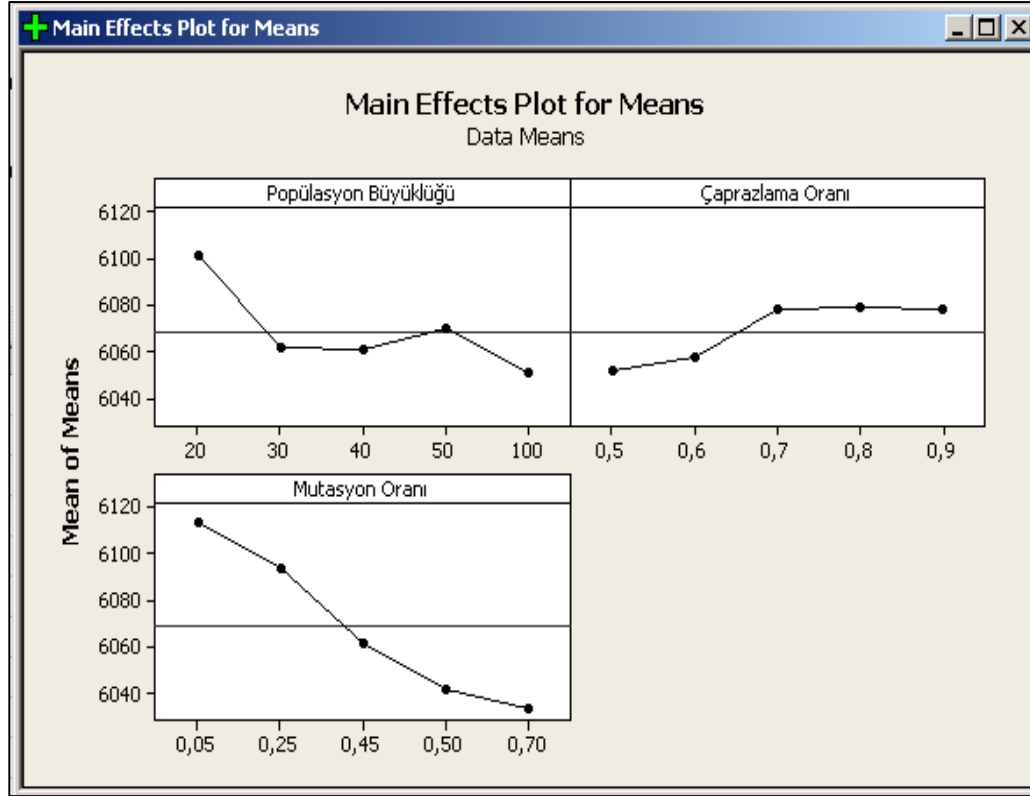
Deneylerin S/N oranları, standart sapma ve ortalama hesaplamaları yapılmıştır. Tablo 5.51’de hesaplanan değerler gösterilmiştir.

Tablo 5. 51 Taguchi Deney Hesaplamaları

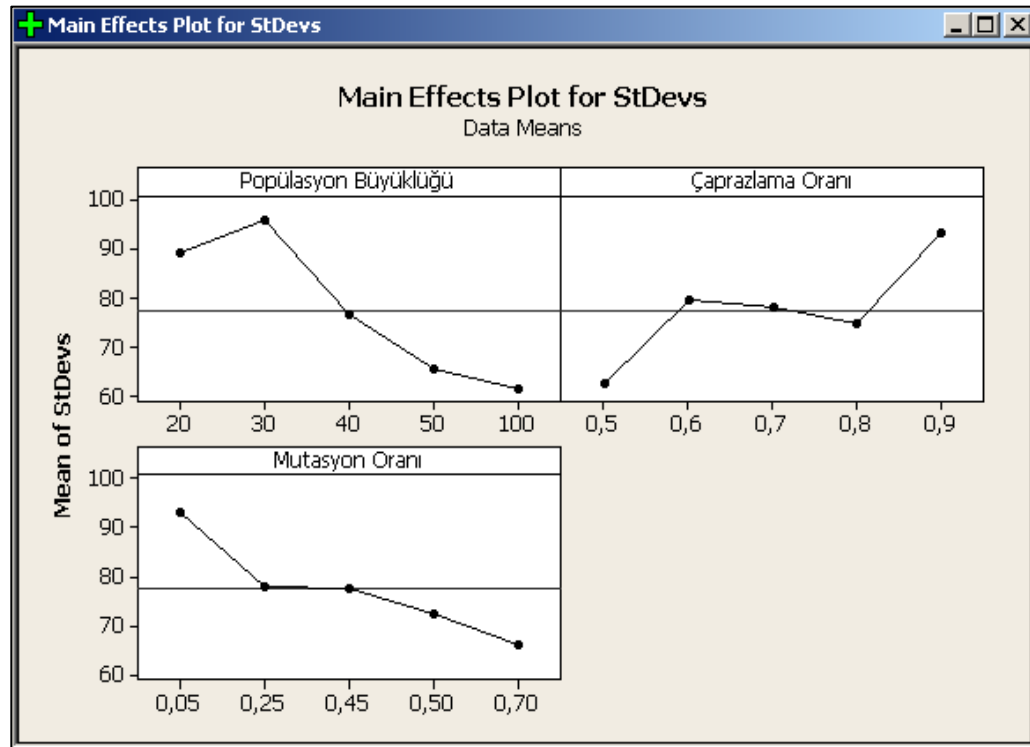
Deney No	S/N	Standart Sapma	Ortalama
1	-75,740	75,884	6122,793
2	-75,705	87,807	6098,027
3	-75,716	92,174	6106,083
4	-75,711	96,075	6102,403
5	-75,672	93,476	6075,260
6	-75,686	77,796	6085,380
7	-75,641	90,679	6053,690
8	-75,610	80,864	6031,817
9	-75,607	68,935	6030,407
10	-75,720	159,361	6107,753
11	-75,606	71,645	6029,037
12	-75,606	77,319	6029,490
13	-75,614	73,711	6034,733
14	-75,723	81,132	6111,327
15	-75,708	78,376	6100,310
16	-75,580	51,114	6011,647
17	-75,590	59,133	6018,113
18	-75,760	67,475	6137,067
19	-75,717	71,257	6106,780
20	-75,672	78,218	6075,587
21	-75,575	36,026	6008,393
22	-75,692	81,914	6089,343
23	-75,677	75,194	6078,790
24	-75,626	55,747	6043,477
25	-75,611	56,783	6033,143

Taguchi deneyleri sonrası grafikler incelendiğinde, problem minimizasyon problemi olduğu için Şekil 5.15’te ortalama değerleri, Şekil 5.16’da standart sapma değerleri en küçük olan değerler en iyi değerlerdir.

En iyi değer popülasyon büyüklüğü 100, çaprazlama oranı 0,5 ve mutasyon oranının 0,7 olduğu görülmektedir.

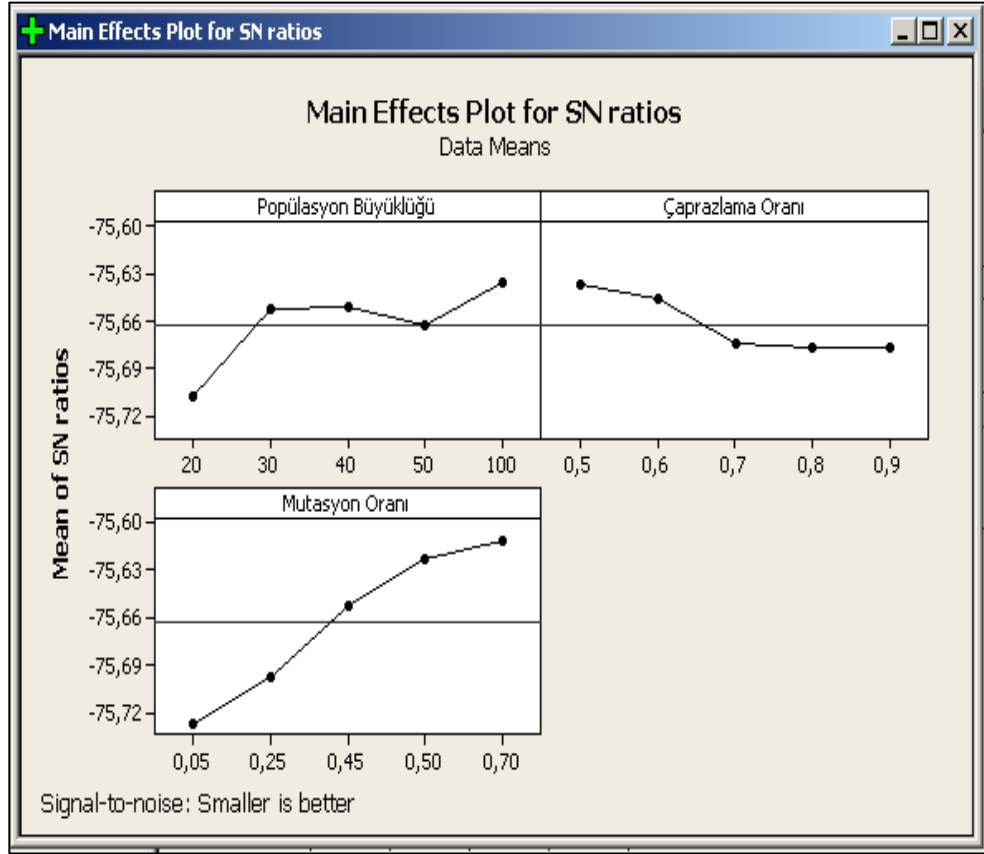


Şekil 5.15 Taguchi Deneyleri İçin Ortalama Sonuç Grafiği



Şekil 5.16 Taguchi Deneyleri İçin Standart Sapma Sonuç Grafiği

Şekil 5.17’de Taguchi deneyleri için S/N sonuç grafiğini vermektedir. S/N sonuç grafiğinin çıkan en iyi değerleri desteklediği görülmektedir.



Şekil 5.17 Taguchi Deneyleri İçin S/N Sonuç Grafiği

Faktör seviyeleri, S/N oranı kalite ölçüsü olup en yüksek olan optimum koşulu sağlamaktadır. Optimum değerleri Tablo 5.52’de özetlenmiştir.

Tablo 5. 52 Taguchi Sonucu Bulunan Optimum Özelleştirilmiş GA Değerleri

Parametre	Optimum Değer
Popülasyon Büyüklüğü	100
Çaprazlama Oranı	0,5
Mutasyon Oranı	0,7

5.5.2.2 Küçük Büyüklükteki Veri Seti için Özelleştirilmiş Genetik Algoritma ile Elde Edilen Sonuçlar

Taguchi metodu ile bulunan Tablo 5.52'deki optimum değerler, diğer veri setlerinin özelleştirilmiş GA ile çözümünde kullanılmıştır. Her veri seti için 30 kez algoritma çalıştırılmıştır.

Dağıtım Yapılan 4 Müşterili Veri Seti için Sonuçlar: 4 müşterili veri seti için elde edilen sonuçlar Tablo 5.53'de gösterilmiştir.

Tablo 5. 53 4 Müşterili Veri için Modifiye Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	1012,10	0,07
2	1012,10	0,09
3	1012,10	0,07
4	1012,10	0,07
5	1012,10	0,07
6	1012,10	0,07
7	1012,10	0,07
8	1012,10	0,07
9	1012,10	0,07
10	1012,10	0,07
11	1012,10	0,07
12	1012,10	0,07
13	1012,10	0,07
14	1012,10	0,07
15	1012,10	0,07
16	1012,10	0,07
17	1012,10	0,07
18	1012,10	0,07
19	1012,10	0,07
20	1012,10	0,07
21	1012,10	0,07
22	1012,10	0,07
23	1012,10	0,07
24	1012,10	0,07

Tablo 5. 53 4 Müşterili Veri İçin Modifiye Özelleştirilmiş GA Çözümleri (devamı)

25	1012,10	0,07
26	1012,10	0,07
27	1012,10	0,07
28	1012,10	0,07
29	1012,10	0,07
30	1012,10	0,07

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 3, 3, 3,

Müşteri 1: Milkrun

Müşteri 2: Milkrun

Müşteri 3: Milkrun

Müşteri 4: Milkrun

İterasyon: 4

Toplam Maliyet: 1012,10

Rota 0: 2, 3, 1, 0; Araç Tipi: 2; Tonaj: 29,35556

Süre: 69,2564

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma Araç Tipi	Rota/Müşteri
Milk-run 3	0-3-4-2-1-0

Toplam Maliyet: 1012,10

Süre (Milisaniye): 69,2564

Dağıtım Yapılan 5 Müşterili Veri Seti için Sonuçlar: 5 müşterili veri seti için elde edilen sonuçları Tablo 5.54’de gösterilmiştir.

Tablo 5. 54 5 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	2053,65	0,08
2	2053,90	0,11
3	2053,65	0,08
4	2053,65	0,08
5	2053,90	0,08
6	2053,90	0,08
7	2053,90	0,08
8	2053,90	0,08
9	2053,65	0,09
10	2053,90	0,08
11	2053,90	0,08
12	2053,65	0,08
13	2053,90	0,08
14	2053,90	0,08
15	2053,65	0,08
16	2053,90	0,08
17	2053,90	0,08
18	2053,90	0,08
19	2053,90	0,08
20	2053,65	0,08
21	2053,90	0,09
22	2053,90	0,08
23	2053,90	0,08
24	2053,90	0,08
25	2053,90	0,08
26	2053,90	0,08
27	2053,65	0,08
28	2053,90	0,08
29	2053,90	0,08
30	2053,90	0,08

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 3, 2, 3, 3,

Müşteri 1: Milkrun

Müşteri 2: Milkrun

Müşteri 3: Direkt Sevkiyat

Müşteri 4: Milkrun

Müşteri 5: Milkrun

İterasyon: 1

Toplam Maliyet: 2053,65

Rota 0: 0, 1, 4, 3; Araç Tipi: 3; Tonaj: 43,15

Süre: 77,612

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma Araç Tipi	Rota/Müşteri
Milk-run 4	0-1-2-5-4-0
Direkt 2	3

Toplam Maliyet: 2053,65

Süre (Milisaniye): 77,612

Dağıtım Yapılan 6 Müşterili Veri Seti için Sonuçlar: 6 müşterili veri seti için elde edilen sonuçları Tablo 5.55'te gösterilmiştir.

Tablo 5. 55 6 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	3056,30	0,12
2	3056,30	0,16
3	3056,30	0,12
4	3056,30	0,11
5	3056,30	0,12
6	3056,30	0,12

Tablo 5. 55 6 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
7	3056,30	0,13
8	3056,30	0,11
9	3056,30	0,07
10	3056,30	0,07
11	3056,30	0,07
12	3056,30	0,07
13	3056,30	0,07
14	3056,30	0,07
15	3056,30	0,07
16	3056,30	0,11
17	3056,30	0,11
18	3056,30	0,10
19	3056,30	0,11
20	3056,30	0,11
21	3056,30	0,11
22	3056,30	0,11
23	3056,30	0,11
24	3056,30	0,12
25	3056,30	0,12
26	3056,30	0,12
27	3056,30	0,12
28	3056,30	0,14
29	3056,30	0,14
30	3056,80	0,21

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 1,3,3,1,3,1,

Müşteri 1: Çapraz Sevkiyat

Müşteri 2: Milkrun

Müşteri 3: Milkrun

Müşteri 4: Çapraz Sevkiyat

Müşteri 5: Milkrun

Müşteri 6: Çapraz Sevkiyat

İterasyon: 198

Toplam Maliyet: 3056,30

Rota 0: 4, 2, 1; Araç Tipi: 3; Tonaj: 43,4

Süre: 69,82

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-5-3-2-0
Çapraz Sevkiyat	4	1,4,6

Toplam Maliyet: 3056,30

Süre (Milisaniye): 69,82

Dağıtım Yapılan 7 Müşterili Veri Seti için Sonuçlar: 7 müşterili veri seti için elde edilen sonuçları Tablo 5.56'da gösterilmiştir.

Tablo 5. 56 7 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet	Süre (Saniye)
1	3269,30	0,10
2	3269,30	0,13
3	3270,90	0,11
4	3270,90	0,11
5	3270,90	0,11
6	3270,90	0,11
7	3270,90	0,11
8	3270,90	0,11
9	3270,90	0,11
10	3270,90	0,11
11	3270,00	0,10
12	3270,90	0,11
13	3270,90	0,11
14	3270,90	0,11
15	3270,90	0,11

Tablo 5. 56 7 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet	Süre (Saniye)
16	3270,90	0,11
17	3270,90	0,11
18	3270,90	0,10
19	3270,90	0,11
20	3270,90	0,10
21	3270,85	0,10
22	3270,85	0,11
23	3270,85	0,10
24	3270,85	0,10
25	3270,85	0,11
26	3270,85	0,11
27	3270,85	0,11
28	3270,85	0,10
29	3270,85	0,10
30	3270,85	0,11

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 2, 3, 2, 2, 3, 3,

Müşteri 1: Milkrun

Müşteri 2: Direkt Sevkiyat

Müşteri 3: Milkrun

Müşteri 4: Direkt Sevkiyat

Müşteri 5: Direkt Sevkiyat

Müşteri 6: Milkrun

Müşteri 7: Milkrun

İterasyon: 1

Toplam Maliyet: 3269,30

Rota 0: 5, 0, 6, 2; Araç Tipi: 3; Tonaj: 49

Süre: 100,8237

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-6-1-7-3-0
Direkt	4	2
Direkt	4	4
Direkt	4	5

Toplam Maliyet: 3269,30

Süre (Milisaniye): 100,82

Dağıtım Yapılan 8 Müşterili Veri Seti için Sonuçlar: 8 müşterili veri seti için elde edilen sonuçlar Tablo 5.57’de gösterilmiştir.

Tablo 5. 57 8 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet	Süre (Saniye)
1	4040,60	0,12
2	4050,80	0,16
3	4050,80	0,12
4	4050,80	0,11
5	4050,80	0,12
6	4050,80	0,12
7	4050,80	0,13
8	4050,80	0,11
9	4050,80	0,13
10	4050,80	0,12
11	4050,80	0,12
12	4040,60	0,12
13	4040,60	0,12
14	4040,60	0,12
15	4040,60	0,12
16	4040,60	0,12
17	4040,60	0,13
18	4040,60	0,12
19	4040,60	0,12
20	4040,60	0,13

Tablo 5. 57 8 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet	Süre (Saniye)
21	4040,60	0,13
22	4040,60	0,12
23	4040,60	0,12
24	4040,60	0,12
25	4040,60	0,12
26	4040,60	0,12
27	4040,60	0,11
28	4040,60	0,13
29	4040,60	0,12
30	4040,60	0,13

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 3, 2, 3, 2, 2, 2, 3,

Müşteri 1: Milkrun

Müşteri 2: Milkrun

Müşteri 3: Direkt Sevkiyat

Müşteri 4: Milkrun

Müşteri 5: Direkt Sevkiyat

Müşteri 6: Direkt Sevkiyat

Müşteri 7: Direkt Sevkiyat

Müşteri 8: Milkrun

İterasyon: 10

Toplam Maliyet: 4040,60

Rota 0: 7, 1, 0, 3; Araç Tipi: 3 ; Tonaj: 46

Süre: 114,80

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-8-2-1-4-0
Direkt	4	3
Direkt	4	5
Direkt	4	6
Direkt	4	7

Toplam Maliyet: 4040,60

Süre (Milisaniye): 114,80

Dağıtım Yapılan 9 Müşterili Veri Seti için Sonuçlar: 9 müşterili veri seti için elde edilen sonuçları Tablo 5.58’de gösterilmiştir.

Tablo 5. 58 9 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	5191,00	0,14
2	5191,00	0,12
3	5191,00	0,12
4	5191,00	0,12
5	5191,00	0,12
6	5191,00	0,12
7	5191,00	0,12
8	5191,00	0,12
9	5191,00	0,12
10	5191,00	0,12
11	5191,00	0,11
12	5191,00	0,12
13	5191,00	0,12
14	5191,00	0,12
15	5191,00	0,12
16	5191,00	0,12
17	5191,00	0,11
18	5191,00	0,12

Tablo 5. 58 9 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
19	5191,00	0,12
20	5191,00	0,12
21	5191,00	0,12
22	5191,00	0,12
23	5191,00	0,12
24	5191,00	0,12
25	5191,00	0,12
26	5191,00	0,12
27	5191,00	0,12
28	5191,00	0,11
29	5191,00	0,12
30	5191,00	0,12

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 2, 2, 2, 3, 3, 3, 2, 2, 2,

Müşteri 1: Direkt Sevkiyat

Müşteri 2: Direkt Sevkiyat

Müşteri 3: Direkt Sevkiyat

Müşteri 4: Milkrun

Müşteri 5: Milkrun

Müşteri 6: Milkrun

Müşteri 7: Direkt Sevkiyat

Müşteri 8: Direkt Sevkiyat

Müşteri 9: Direkt Sevkiyat

İterasyon: 12

Toplam Maliyet: 5191,00

Rota 0: 5, 4, 3; Araç Tipi: 3 ;Tonaj: 44

Süre: 110,091

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma Araç Tipi		Rota/Müşteri
Milk-run	4	0-6-5-4-0
Direkt	4	1
Direkt	4	2
Direkt	4	3
Direkt	4	7
Direkt	4	8
Direkt	4	9

Toplam Maliyet: 5191,00

Süre (Milisaniye): 110,91

Dağıtım Yapılan 10 Müşterili Veri Seti için Sonuçlar: 10 müşterili veri seti için elde edilen sonuçlar Tablo 5.59'da gösterilmiştir.

Tablo 5. 59 10 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	6050,00	0,15
2	5985,50	0,16
3	5985,50	0,14
4	5985,50	0,13
5	6050,00	0,15
6	6050,00	0,13
7	5985,50	0,14
8	5985,50	0,14
9	5985,50	0,13
10	6050,00	0,14
11	6050,00	0,14
12	5985,50	0,13
13	6050,00	0,14
14	5985,50	0,14
15	6050,00	0,14

Tablo 5. 59 10 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
16	6050,00	0,14
17	5985,50	0,14
18	6050,00	0,14
19	5985,50	0,13
20	6050,00	0,14
21	6050,00	0,14
22	6050,00	0,14
23	6065,60	0,15
24	6050,00	0,14
25	6050,00	0,14
26	6050,00	0,15
27	6050,00	0,13
28	6050,00	0,14
29	6050,00	0,14
30	6050,00	0,14

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 2, 2, 3, 2, 2, 2, 3, 2, 3, 2,

Müşteri 1: Direkt Sevkiyat

Müşteri 2: Direkt Sevkiyat

Müşteri 3: Milkrun

Müşteri 4: Direkt Sevkiyat

Müşteri 5: Direkt Sevkiyat

Müşteri 6: Direkt Sevkiyat

Müşteri 7: Milkrun

Müşteri 8: Direkt Sevkiyat

Müşteri 9: Milkrun

Müşteri 10: Direkt Sevkiyat

İterasyon: 142

Toplam Maliyet: 5985,50

Rota 0: 2, 6, 8; Araç Tipi: 3; Tonaj: 48

Süre: 132,26

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma Araç Tipi	Rota/Müşteri
Milk-run	4
	0-3-7-9-0
Direkt	4
	1
Direkt	4
	2
Direkt	4
	4
Direkt	4
	5
Direkt	4
	6
Direkt	4
	8
Direkt	4
	10

Toplam Maliyet: 5985,50

Süre (Milisaniye) : 132,26

Dağıtım Yapılan 11 Müşterili Veri Seti için Sonuçlar: 11 müşterili veri seti için elde edilen sonuçlar Tablo 5.60'ta gösterilmiştir.

Tablo 5. 60 11 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	6365,10	0,18
2	6365,10	0,23
3	6365,10	0,19
4	6365,10	0,17
5	6365,10	0,18
6	6365,10	0,18
7	6365,10	0,19
8	6365,10	0,16
9	6365,10	0,19

Tablo 5. 60 11 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
10	6365,10	0,20
11	6365,10	0,19
12	6365,10	0,18
13	6365,10	0,19
14	6365,10	0,19
15	6365,10	0,18
16	6365,10	0,19
17	6365,10	0,20
18	6365,10	0,18
19	6365,10	0,19
20	6365,10	0,18
21	6365,10	0,21
22	6365,10	0,17
23	6365,10	0,20
10	6365,10	0,20
11	6365,10	0,19
12	6365,10	0,18
13	6365,10	0,19
14	6365,10	0,19
15	6365,10	0,18
16	6365,10	0,19
17	6365,10	0,20
18	6365,10	0,18
19	6365,10	0,19
20	6365,10	0,18
21	6365,10	0,21
22	6365,10	0,17
23	6365,10	0,20
24	6365,10	0,18
25	6365,10	0,17
26	6365,10	0,19
27	6365,10	0,20
28	6365,10	0,18
29	6365,10	0,18
30	6365,10	0,17

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 2, 3, 2, 2, 2, 2, 3, 3, 2, 3,

Müşteri 1: Milkrun

Müşteri 2: Direkt Sevkiyat

Müşteri 3: Milkrun

Müşteri 4: Direkt Sevkiyat

Müşteri 5: Direkt Sevkiyat

Müşteri 6: Direkt Sevkiyat

Müşteri 7: Direkt Sevkiyat

Müşteri 8: Milkrun

Müşteri 9: Milkrun

Müşteri 10: Direkt Sevkiyat

Müşteri 11: Milkrun

İterasyon: 125

Toplam Maliyet: 6365,10

Rota 0: 8, 7, 10; Araç Tipi: 3; Tonaj: 45

Rota 1: 0, 2, Araç Tipi: 2, Tonaj: 30

Süre: 161,65

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma Araç Tipi	Rota/Müşteri	
Milk-run	4	0-9-8-11-0
Milk-run	3	0-1-3-0
Direkt	4	2
Direkt	4	4
Direkt	4	5

Direkt 4 6
Direkt 4 7
Direkt 4 10

Toplam Maliyet: 6365,10

Süre (Milisaniye): 161,65

Dağıtım Yapılan 12 Müşterili Veri Seti için Sonuçlar: 12 müşterili veri seti için elde edilen sonuçlar Tablo 5.61’de gösterilmiştir.

Tablo 5. 61 12 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	8694,70	0,09
2	8694,70	0,11
3	8695,80	0,10
4	8694,70	0,09
5	8695,80	0,09
6	8694,70	0,09
7	8694,70	0,09
8	8694,70	0,09
9	8694,70	0,09
10	8694,70	0,09
11	8694,70	0,09
12	8694,70	0,09
13	8695,80	0,09
14	8695,80	0,09
15	8695,80	0,09
16	8695,80	0,09
17	8694,70	0,09
18	8695,80	0,09
19	8695,80	0,09
20	8695,80	0,09
21	8694,70	0,09
22	8695,80	0,09
23	8695,80	0,09
24	8695,80	0,09

Tablo 5. 61 12 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
25	8695,80	0,09
26	8695,80	0,09
27	8695,80	0,09
28	8695,80	0,09
29	8695,80	0,09
30	8695,80	0,09

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

Müşteri 1: Çapraz Sevkiyat

Müşteri 2: Çapraz Sevkiyat

Müşteri 3: Çapraz Sevkiyat

Müşteri 4: Çapraz Sevkiyat

Müşteri 5: Çapraz Sevkiyat

Müşteri 6: Çapraz Sevkiyat

Müşteri 7: Çapraz Sevkiyat

Müşteri 8: Çapraz Sevkiyat

Müşteri 9: Çapraz Sevkiyat

Müşteri 10: Çapraz Sevkiyat

Müşteri 11: Çapraz Sevkiyat

Müşteri 12: Çapraz Sevkiyat

İterasyon: 11

Toplam Maliyet: 8694,70

Süre: 91,06

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Çapraz Sevkiyat	4	1
Çapraz Sevkiyat	4	2
Çapraz Sevkiyat	4	3
Çapraz Sevkiyat	4	4
Çapraz Sevkiyat	4	5
Çapraz Sevkiyat	4	6
Çapraz Sevkiyat	4	7
Çapraz Sevkiyat	4	8
Çapraz Sevkiyat	4	9
Çapraz Sevkiyat	4	10
Çapraz Sevkiyat	4	11
Çapraz Sevkiyat	4	12

Toplam Maliyet: 8694,70

Süre (Milisaniye): 91,06

5.5.2.3 Orta Büyüklükte Veri Seti için Özelleştirilmiş Genetik Algoritma (GA) ile Edilen Sonuçlar

Dağıtım Yapılan 20 Müşterili Veri Seti için Sonuçlar: 20 müşterili veri seti için özelleştirilmiş GA ile edilen sonuçlar Tablo 5.62’de verilmiştir.

Tablo 5. 62 20 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	9842,8	0,48
2	9898,8	0,53
3	9986	0,50
4	10008,4	0,53
5	9856,8	0,61

Tablo 5. 62 20 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
6	9898,8	0,45
7	9981,8	0,51
8	9898,8	0,46
9	9898,8	0,64
10	9842,8	0,49
11	9761,4	0,51
12	9991,4	0,49
13	9898,8	0,64
14	9898,8	0,55
15	9898,8	0,50
16	9898,8	0,51
17	9842,8	0,60
18	9898,8	0,52
19	9786,8	0,53
20	10106,2	0,53
21	9786,8	0,38
22	9898,8	0,52
23	9955,4	0,59
24	9806,8	0,50
25	9898,8	0,50
26	10082,4	0,44
27	9898,8	0,52
28	9840,8	0,44
29	9749,8	0,44
30	9761,4	0,50

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 3, 2, 3, 2, 2, 2, 3, 3, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1,

Müşteri 1: Milkrun

Müşteri 2: Milkrun

Müşteri 3: Direkt Sevkiyat

Müşteri 4: Milkrun

Müşteri 5: Direkt Sevkiyat

Müşteri 6: Direkt Sevkiyat

Müşteri 7: Direkt Sevkiyat

Müşteri 8: Milkrun

Müşteri 9: Milkrun

Müşteri 10: Çapraz Sevkiyat

Müşteri 11: Çapraz Sevkiyat

Müşteri 12: Milkrun

Müşteri 13: Çapraz Sevkiyat

Müşteri 14: Çapraz Sevkiyat

Müşteri 15: Çapraz Sevkiyat

Müşteri 16: Çapraz Sevkiyat

Müşteri 17: Çapraz Sevkiyat

Müşteri 18: Çapraz Sevkiyat

Müşteri 19: Milkrun

Müşteri 20: Çapraz Sevkiyat

İterasyon: 59

Toplam Maliyet: 9749,80

Rota 0: 18, 11, 1, 0, Araç Tipi: 3, Tonaj: 50

Rota 1: 7, 3, 8; Araç Tipi: 3; Tonaj: 50

Süre: 436,33

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-19-12-2-1-0
Milk-run	4	0-8-4-9-0
Direkt	2	3

Direkt	4	5
Direkt	4	6
Direkt	4	7
Çapraz Sevkiyat	4	10
Çapraz Sevkiyat	4	11
Çapraz Sevkiyat	4	13
Çapraz Sevkiyat	4	14
Çapraz Sevkiyat	4	15
Çapraz Sevkiyat	4	16
Çapraz Sevkiyat	4	17
Çapraz Sevkiyat	4	18
Çapraz Sevkiyat	4	20

Toplam Maliyet: 9749,80

Süre (Milisaniye): 436,33

Dağıtım Yapılan 30 Müşterili Veri Seti için Sonuçlar: 30 müşterili veri seti için bulunan sonuçlar Tablo 5.63'te gösterilmiştir.

Tablo 5. 63 30 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	16991,28	6,58
2	16814,94	6,61
3	17101,48	7,32
4	16813,19	7,32
5	17085,94	6,38
6	17216,91	6,55
7	16945,59	6,45
8	16952,04	5,44
9	16882,64	6,17
10	16985,94	6,07
11	16738,64	7,46

Tablo 5. 63 30 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri **(devamı)**

Deneme No	Maliyet (TL)	Süre (Saniye)
12	16773,04	6,14
13	16898,89	5,97
14	16908,51	5,05
15	17013,34	5,95
16	16980,75	6,71
17	17135,42	7,38
18	16855,54	4,86
19	16944,79	6,67
20	16990,20	5,90
21	17124,91	7,81
22	16882,75	7,26
23	17163,59	6,69
24	17176,89	5,90
25	17055,79	6,86
26	16840,64	5,64
27	16950,89	6,79
28	17025,44	6,63
29	17219,50	6,77
30	17183,50	6,78

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 1, 3, 3, 3, 3, 3, 3, 2, 3, 2, 2, 3, 3, 1, 1, 1, 3, 1, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3, 1,

Müşteri 1: Çapraz Sevkiyat

Müşteri 2: Milkrun

Müşteri 3: Milkrun

Müşteri 4: Milkrun

Müşteri 5: Milkrun

Müşteri 6: Milkrun

Müşteri 7: Milkrun

Müşteri 8: Direkt Sevkiyat

Müşteri 9: Milkrun

Müşteri 10: Direkt Sevkiyat

Müşteri 11: Direkt Sevkiyat

Müşteri 12: Milkrun

Müşteri 13: Milkrun

Müşteri 14: Çapraz Sevkiyat

Müşteri 15: Çapraz Sevkiyat

Müşteri 16: Çapraz Sevkiyat

Müşteri 17: Milkrun

Müşteri 18: Çapraz Sevkiyat

Müşteri 19: Direkt Sevkiyat

Müşteri 20: Direkt Sevkiyat

Müşteri 21: Direkt Sevkiyat

Müşteri 22: Milkrun

Müşteri 23: Milkrun

Müşteri 24: Çapraz Sevkiyat

Müşteri 25: Direkt Sevkiyat

Müşteri 26: Milkrun

Müşteri 27: Milkrun

Müşteri 28: Direkt Sevkiyat

Müşteri 29: Milkrun

Müşteri 30: Çapraz Sevkiyat

İterasyon: 126

Toplam Maliyet: 16738,64

Rota 0: 16, 12, 28, 22, 1, Araç Tipi: 3 Tonaj: 44,22

Rota 1: 25, 26, 4, Araç Tipi: 3 Tonaj: 46,51

Rota 2: 11, 21, Araç Tipi: 3 Tonaj: 36

Rota 3: 5, 2, 6, Araç Tipi: 3 Tonaj: 47

Rota 4: 3, 8, Araç Tipi: 3 Tonaj: 50

Süre: 7461,67

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-17-13-29-23-2-0
Milk-run	4	0-26-27-5-0
Milk-run	4	0-12-22-0
Milk-run	4	0-6-3-7-0
Milk-run	4	0-4-9-0
Direkt	2	8
Direkt	4	10
Direkt	4	11
Direkt	3	19
Direkt	4	20
Direkt	2	21
Direkt	2	25
Direkt	2	28
Çapraz Sevkiyat	4	1
Çapraz Sevkiyat	4	14
Çapraz Sevkiyat	4	15
Çapraz Sevkiyat	4	16

Çapraz Sevkiyat 4 18

Çapraz Sevkiyat 4 24

Çapraz Sevkiyat 4 30

Toplam Maliyet: 16738,64

Süre (Milisaniye): 7461,67

Dağıtım Yapılan 50 Müşterili Veri Seti için Sonuçlar: 50 müşterili veri seti için bulunan sonuçlar Tablo 5.64'te gösterilmiştir.

Tablo 5. 64 50 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	29202,7	69,22
2	29643,5	55,30
3	29302,42	57,13
4	29642,34	59,16
5	29409,9	65,76
6	29081,92	52,69
7	29322	74,23
8	29581,44	45,13
9	29124,2	50,75
10	29298,1	57,79
11	29405,1	68,39
12	29338,32	66,88
13	29472,72	71,10
14	29539,62	69,35
15	29341,6	66,58
16	29633,62	57,02
17	29744,42	55,01
18	29440,72	71,23
19	29554,32	57,02
20	29368,92	55,90
21	29253,82	63,17
22	29437,44	53,34
23	29717,8	61,51
24	29443,8	59,29

Tablo 5. 64 50 Müşterili Veri İçin Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
25	29626,42	50,66
26	29690,22	51,60
27	29684	59,31
28	29324,8	67,20
29	29443,72	51,46
30	29223,62	64,64

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 3, 3, 2, 1, 1, 1, 1, 3, 1, 3, 1, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 3, 2, 2, 2, 3, 2, 2, 3, 1, 3, 2, 2, 3, 1, 3, 1, 3, 3, 2, 3, 3, 2, 3, 1, 3, 3, 3, 3,

Müşteri 1: Milkrun

Müşteri 2: Milkrun

Müşteri 3: Milkrun

Müşteri 4: Direkt Sevkiyat

Müşteri 5: Çapraz Sevkiyat

Müşteri 6: Çapraz Sevkiyat

Müşteri 7: Çapraz Sevkiyat

Müşteri 8: Çapraz Sevkiyat

Müşteri 9: Milkrun

Müşteri 10: Çapraz Sevkiyat

Müşteri 11: Milkrun

Müşteri 12: Çapraz Sevkiyat

Müşteri 13: Milkrun

Müşteri 14: Milkrun

Müşteri 15: Milkrun

Müşteri 16: Milkrun

Müşteri 17: Çapraz Sevkiyat
Müşteri 18: Direkt Sevkiyat
Müşteri 19: Milkrun
Müşteri 20: Milkrun
Müşteri 21: Milkrun
Müşteri 22: Milkrun
Müşteri 23: Milkrun
Müşteri 24: Direkt Sevkiyat
Müşteri 25: Direkt Sevkiyat
Müşteri 26: Direkt Sevkiyat
Müşteri 27: Milkrun
Müşteri 28: Direkt Sevkiyat
Müşteri 29: Direkt Sevkiyat
Müşteri 30: Milkrun
Müşteri 31: Çapraz Sevkiyat
Müşteri 32: Milkrun
Müşteri 33: Direkt Sevkiyat
Müşteri 34: Direkt Sevkiyat
Müşteri 35: Milkrun
Müşteri 36: Çapraz Sevkiyat
Müşteri 37: Milkrun
Müşteri 38: Çapraz Sevkiyat
Müşteri 39: Milkrun
Müşteri 40: Milkrun
Müşteri 41: Direkt Sevkiyat

Müşteri 42: Milkrun

Müşteri 43: Milkrun

Müşteri 44: Direkt Sevkiyat

Müşteri 45: Milkrun

Müşteri 46: Çapraz Sevkiyat

Müşteri 47: Milkrun

Müşteri 48: Milkrun

Müşteri 49: Milkrun

Müşteri 50: Milkrun

İterasyon: 134

Toplam Maliyet: 29081,92

Rota 0: 18, 38, 36; Araç Tipi: 3; Tonaj: 50

Rota 1: 31, 48; Araç Tipi: 3; Tonaj: 47

Rota 2: 26, 39, 20; Araç Tipi: 3; Tonaj: 47

Rota 3: 46, 29, 41; Araç Tipi: 3; Tonaj: 50

Rota 4: 42, 47, 12; Araç Tipi: 3; Tonaj: 49

Rota 5: 8, 22, 34; Araç Tipi: 3; Tonaj: 49,5

Rota 6: 13, 44; Araç Tipi: 3; Tonaj: 50

Rota 7: 19, 21; Araç Tipi: 3; Tonaj: 40

Rota 8: 2, 15; Araç Tipi: 3; Tonaj: 49,5

Rota 9: 0, 49; Araç Tipi: 3; Tonaj: 35

Rota 10: 1, 10; Araç Tipi: 3; Tonaj: 50

Rota 11: 14; Araç Tipi: 3; Tonaj: 35

Süre: 52694,93

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-19-39-37-0
Milk-run	4	0-32-49-0
Milk-run	4	0-27-40-21-0
Milk-run	4	0-47-30-42-0
Milk-run	4	0-43-48-13-0
Milk-run	4	0-9-23-35-0
Milk-run	4	0-14-45-0
Milk-run	4	0-20-22-0
Milk-run	4	0-3-16-0
Milk-run	4	0-1-50-0
Milk-run	4	0-2-11-0
Milk-run	4	0-15-0
Çapraz Sevkiyat	4	5
Çapraz Sevkiyat	4	6
Çapraz Sevkiyat	4	7
Çapraz Sevkiyat	4	8
Çapraz Sevkiyat	4	10
Çapraz Sevkiyat	4	12
Çapraz Sevkiyat	4	31
Çapraz Sevkiyat	4	36
Çapraz Sevkiyat	4	17
Çapraz Sevkiyat	4	38
Çapraz Sevkiyat	4	46

Direkt	4	4
Direkt	2	18
Direkt	4	24
Direkt	4	25
Direkt	4	26
Direkt	3	28
Direkt	3	33
Direkt	4	34
Direkt	2	41
Direkt	4	44

Toplam Maliyet: 29081,92

Süre (Milisaniye): 52694,93

5.5.2.4 Büyük Veri Seti için Özelleştirilmiş Genetik Algoritma (GA) ile Edilen Sonuçlar

Dağıtım yapılan 100 müşterili veri seti için elde edilen sonuçlar Tablo 5.65'te gösterilmiştir.

Tablo 5. 65 100 Müşterili Veri için Özelleştirilmiş GA Çözümleri

Deneme No	Maliyet (TL)	Süre (Saniye)
1	73530,80	60,50
2	73509,30	81,36
3	72832,70	70,12
4	72711,50	96,49
5	74115,90	69,68
6	73295,10	76,01
7	73028,90	93,19
8	73983,50	56,29
9	72714,70	125,83
10	73867,60	83,12
11	72938,30	58,79
12	73355,80	128,43

Tablo 5. 65 100 Müşterili Veri için Özelleştirilmiş GA Çözümleri (devamı)

Deneme No	Maliyet (TL)	Süre (Saniye)
13	73813,50	67,94
14	73877,70	70,17
15	73320,40	83,89
16	74050,10	68,33
17	73343,90	85,19
18	73199,90	131,48
19	74114,10	49,45
20	73190,50	84,13
21	73604,60	73,29
22	73123,10	83,75
23	72955,80	72,34
24	73601,00	57,38
25	72941,50	69,91
26	73227,60	94,98
27	73618,20	60,18
28	73247,70	100,40
29	73536,30	64,64
30	73252,00	68,54

En iyi sonucun algoritma çıktısı aşağıda verilmiştir:

199. İterasyon Sonuçları: 3, 2, 1, 1, 3, 2, 1, 2, 3, 2, 1, 1, 1, 3, 1, 2, 1, 1, 1, 1, 1, 1, 3, 3, 1, 2, 2, 2, 2, 1, 2, 3, 2, 3, 3, 3, 1, 1, 3, 3, 1, 2, 3, 1, 2, 2, 2, 1, 2, 2, 1, 3, 2, 1, 3, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 3, 2, 2, 1, 1, 3, 2, 2, 3, 3, 2, 2, 1, 2, 3, 3, 1, 1, 1, 1, 3, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 3, 1,

Müşteri 1: Milkrun

Müşteri 2: Direkt Sevkiyat

Müşteri 3: Çapraz Sevkiyat

Müşteri 4: Çapraz Sevkiyat

Müşteri 5: Milkrun

Müşteri 6: Direkt Sevkiyat

Müşteri 7: Çapraz Sevkiyat

Müşteri 8: Direkt Sevkiyat

Müşteri 9: Milkrun

Müşteri 10: Direkt Sevkiyat

Müşteri 11: Çapraz Sevkiyat

Müşteri 12: Çapraz Sevkiyat

Müşteri 13: Çapraz Sevkiyat

Müşteri 14: Milkrun

Müşteri 15: Çapraz Sevkiyat

Müşteri 16: Direkt Sevkiyat

Müşteri 17: Çapraz Sevkiyat

Müşteri 18: Çapraz Sevkiyat

Müşteri 19: Çapraz Sevkiyat

Müşteri 20: Çapraz Sevkiyat

Müşteri 21: Çapraz Sevkiyat

Müşteri 22: Çapraz Sevkiyat

Müşteri 23: Milkrun

Müşteri 24: Milkrun

Müşteri 25: Çapraz Sevkiyat

Müşteri 26: Direkt Sevkiyat

Müşteri 27: Direkt Sevkiyat

Müşteri 28: Direkt Sevkiyat

Müşteri 29: Direkt Sevkiyat

Müşteri 30: Çapraz Sevkiyat

Müşteri 31: Direkt Sevkiyat

Müşteri 32: Milkrun

Müşteri 33: Direkt Sevkiyat

Müşteri 34: Milkrun
Müşteri 35: Milkrun
Müşteri 36: Milkrun
Müşteri 37: Çapraz Sevkiyat
Müşteri 38: Çapraz Sevkiyat
Müşteri 39: Milkrun
Müşteri 40: Milkrun
Müşteri 41: Çapraz Sevkiyat
Müşteri 42: Direkt Sevkiyat
Müşteri 43: Milkrun
Müşteri 44: Çapraz Sevkiyat
Müşteri 45: Direkt Sevkiyat
Müşteri 46: Direkt Sevkiyat
Müşteri 47: Direkt Sevkiyat
Müşteri 48: Çapraz Sevkiyat
Müşteri 49: Direkt Sevkiyat
Müşteri 50: Direkt Sevkiyat
Müşteri 51: Çapraz Sevkiyat
Müşteri 52: Milkrun
Müşteri 53: Direkt Sevkiyat
Müşteri 54: Çapraz Sevkiyat
Müşteri 55: Milkrun
Müşteri 56: Direkt Sevkiyat
Müşteri 57: Çapraz Sevkiyat
Müşteri 58: Çapraz Sevkiyat

Müşteri 59: Direkt Sevkiyat
Müşteri 60: Direkt Sevkiyat
Müşteri 61: Direkt Sevkiyat
Müşteri 62: Direkt Sevkiyat
Müşteri 63: Direkt Sevkiyat
Müşteri 64: Çapraz Sevkiyat
Müşteri 65: Direkt Sevkiyat
Müşteri 66: Milkrun
Müşteri 67: Direkt Sevkiyat
Müşteri 68: Direkt Sevkiyat
Müşteri 69: Çapraz Sevkiyat
Müşteri 70: Çapraz Sevkiyat
Müşteri 71: Milkrun
Müşteri 72: Direkt Sevkiyat
Müşteri 73: Direkt Sevkiyat
Müşteri 74: Milkrun
Müşteri 75: Milkrun
Müşteri 76: Direkt Sevkiyat
Müşteri 77: Direkt Sevkiyat
Müşteri 78: Çapraz Sevkiyat
Müşteri 79: Direkt Sevkiyat
Müşteri 80: Milkrun
Müşteri 81: Milkrun
Müşteri 82: Çapraz Sevkiyat
Müşteri 83: Çapraz Sevkiyat

Müşteri 84: Çapraz Sevkiyat

Müşteri 85: Çapraz Sevkiyat

Müşteri 86: Milkrun

Müşteri 87: Direkt Sevkiyat

Müşteri 88: Çapraz Sevkiyat

Müşteri 89: Direkt Sevkiyat

Müşteri 90: Çapraz Sevkiyat

Müşteri 91: Çapraz Sevkiyat

Müşteri 92: Direkt Sevkiyat

Müşteri 93: Direkt Sevkiyat

Müşteri 94: Çapraz Sevkiyat

Müşteri 95: Çapraz Sevkiyat

Müşteri 96: Çapraz Sevkiyat

Müşteri 97: Çapraz Sevkiyat

Müşteri 98: Çapraz Sevkiyat

Müşteri 99: Milkrun

Müşteri 100: Çapraz Sevkiyat

İterasyon: 198

Toplam Maliyet: 72711,5

Rota 0: 73, 79; Araç Tipi: 3; Tonaj: 47

Rota 1: 80, 31, 33; Araç Tipi: 3; Tonaj: 50

Rota 2: 0, 98; Araç Tipi: 3; Tonaj: 47

Rota 3: 65, 34, 8; Araç Tipi: 3; Tonaj: 48

Rota 4: 51, 54, 85; Araç Tipi: 3; Tonaj: 50

Rota 5: 23, 22, 70; Araç Tipi: 3; Tonaj: 49

Rota 6: 39, 13, 35; Araç Tipi: 3; Tonaj: 49

Rota 7: 4, 38; Araç Tipi: 3; Tonaj: 42

Rota 8: 42; Araç Tipi: 3; Tonaj: 34

Rota 9: 74; Araç Tipi: 3; Tonaj: 46

Süre (Milisaniye): 96487,79

En iyi sonucun rota, maliyeti ve süresi aşağıdaki gibidir:

Taşıma	Araç Tipi	Rota/Müşteri
Milk-run	4	0-74-79-0
Milk-run	4	0-81-32-34-0
Milk-run	4	0-1-99-0
Milk-run	4	0-66-35-9-0
Milk-run	4	0-52-55-86-0
Milk-run	4	0-24-23-71-0
Milk-run	4	0-40-14-36-0
Milk-run	4	0-5-39-0
Milk-run	4	0-43-0
Milk-run	4	0-75-0
Çapraz Sevkiyat	4	3
Çapraz Sevkiyat	4	4
Çapraz Sevkiyat	4	7
Çapraz Sevkiyat	4	11
Çapraz Sevkiyat	4	12
Çapraz Sevkiyat	4	13
Çapraz Sevkiyat	4	15
Çapraz Sevkiyat	4	17

Çapraz Sevkiyat	4	18
Çapraz Sevkiyat	4	19
Çapraz Sevkiyat	4	20
Çapraz Sevkiyat	4	21
Çapraz Sevkiyat	4	22
Çapraz Sevkiyat	4	25
Çapraz Sevkiyat	4	30
Çapraz Sevkiyat	4	37
Çapraz Sevkiyat	4	38
Çapraz Sevkiyat	4	41
Çapraz Sevkiyat	4	44
Çapraz Sevkiyat	4	48
Çapraz Sevkiyat	4	51
Çapraz Sevkiyat	4	54
Çapraz Sevkiyat	4	57
Çapraz Sevkiyat	4	58
Çapraz Sevkiyat	4	64
Çapraz Sevkiyat	4	69
Çapraz Sevkiyat	4	70
Çapraz Sevkiyat	4	78
Çapraz Sevkiyat	4	82
Çapraz Sevkiyat	4	83
Çapraz Sevkiyat	4	84
Çapraz Sevkiyat	4	85
Çapraz Sevkiyat	4	88

Çapraz Sevkiyat	4	90
Çapraz Sevkiyat	4	91
Çapraz Sevkiyat	4	94
Çapraz Sevkiyat	4	95
Çapraz Sevkiyat	4	96
Çapraz Sevkiyat	4	97
Çapraz Sevkiyat	4	98
Çapraz Sevkiyat	4	100
Direkt	4	2
Direkt	2	6
Direkt	4	8
Direkt	2	10
Direkt	4	16
Direkt	4	26
Direkt	4	27
Direkt	4	28
Direkt	4	29
Direkt	4	31
Direkt	4	33
Direkt	4	42
Direkt	4	45
Direkt	4	46
Direkt	4	47
Direkt	2	49
Direkt	4	50

Direkt	3	53
Direkt	2	56
Direkt	3	59
Direkt	4	60
Direkt	4	61
Direkt	4	62
Direkt	4	63
Direkt	4	65
Direkt	2	67
Direkt	4	68
Direkt	3	72
Direkt	3	73
Direkt	4	76
Direkt	4	77
Direkt	4	79
Direkt	3	87
Direkt	4	89
Direkt	4	92
Direkt	4	93

Toplam Maliyet: 72711,50

Süre (Milisaniye): 96487,79

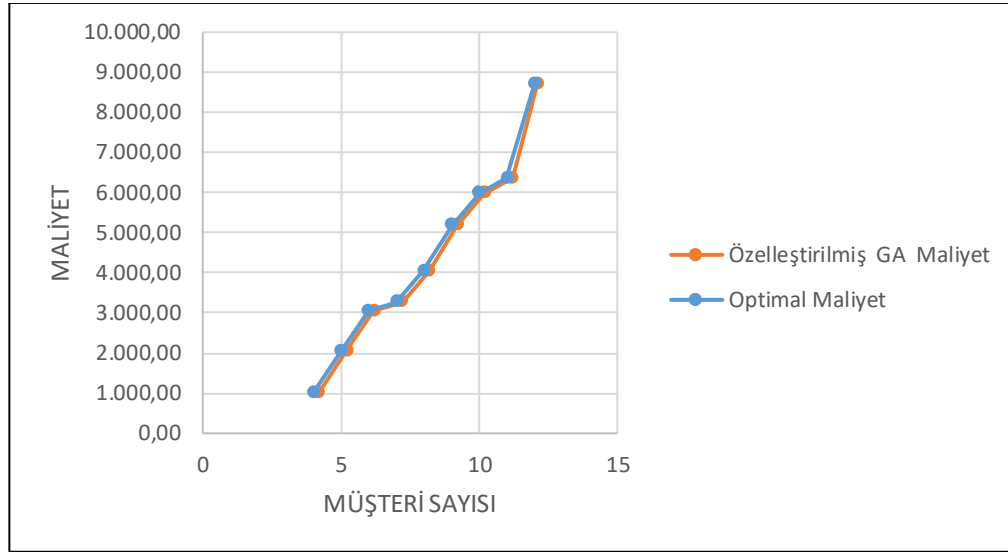
5.5.3 Çözümlerin Karşılaştırılması

Elde edilen çözümler Tablo 5.66'da özetlenmiştir. 4-12 müşteri arası veri seti için bulunan optimal çözümler ve özelleştirilmiş GA çözümleri SPSS paket programı yardımıyla kolmogorov smirnov testi ve eşleştirilmiş t testi yürütülerek karşılaştırılmıştır. Bulunan sonuçların benzer sonuçlar verip vermediği araştırılmıştır.

Tablo 5. 66 Çözümlerin Karşılaştırılması

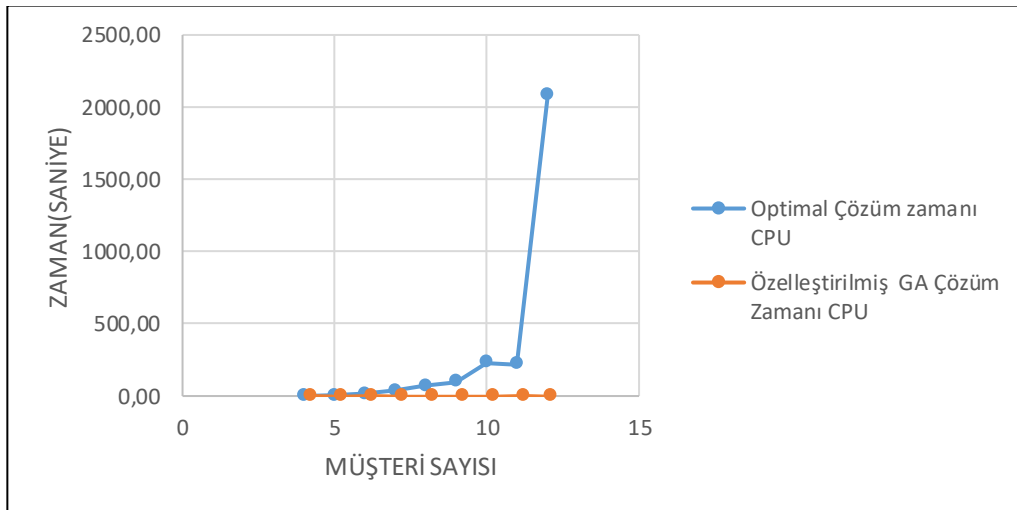
Veri Seti	Optimal Çözüm	Süre (Sn.)	En iyi Özelleştirilmiş GA Çözümü	Süre (Sn.)
4 müşteri	1012,1	1	1012,1	0,07
5 müşteri	2053,6	13	2053,65	0,08
6 müşteri	3056,1	34	3056,3	0,07
7 müşteri	3269,2	69	3269,3	0,1
8 müşteri	4040,5	100	4040,6	0,11
9 müşteri	5190,6	229	5191	0,11
10 müşteri	5985,5	222	5985,5	0,13
11 müşteri	6365	2080	6365,1	0,16
12 müşteri	8694,6	3654	8694,7	0,09
20 müşteri	-	-	9749,80	0,44
30 müşteri	-	-	16738,64	7,46
50 müşteri	-	-	29081,92	52,69
100 müşteri	-	-	72711,50	96,49

Şekil 5.18’de özelleştirilmiş GA maliyet çözümleri ile optimal maliyet çözümleri karşılaştırılmıştır. Küçük müşterili veri setleri için çözümlerin benzer sonuçlar verdiği görülmektedir.



Şekil 5.18 Çözüm Maliyetlerinin Karşılaştırılması

Şekil 5.19’da, her iki yöntem için de CPU çalışma zamanlarını göstermektedir. Küçük boyutlu örneklerde, iki yöntemin yaklaşık olarak aynı çalışma süresine sahip olduğu, ancak daha büyük boyutlu örneklerde optimizasyon yönteminin çalışma süresinin katlanarak arttığı görülmektedir. Bu yüzden büyük boyutlu örnekler için, sezgisel yöntem makul bir sürede çözümler elde etmek için gereklidir.



Şekil 5.19 Çözüm Sürelerinin Karşılaştırılması

5.5.3.1 Kolmogorov Smirnov Testi

Çözümlerin karşılaştırılmasında eşleştirilmiş t testinin uygulanıp uygulanmayacağını belirlemek için çözüm değerleri arasındaki farkların normal dağılıma uyup uymadığı test edilmiştir.

- Kolmogorov smirnov testi sonucu elde edilen p değeri, $p > 0,05$ ise; “ H_0 = Normal dağılıyor” hipotezi kabul edilir.
- Kolmogorov smirnov testi sonucu elde edilen p değeri, $p < 0,05$ ise; “ H_1 = Normal dağılmıyor” hipotezi kabul edilir.

Şekil 5.20’den de görüldüğü üzere çözüm değerleri arasındaki fark değerleri normal dağılım göstermektedir. Çünkü test sonuçlarında yer alan anlamlılık düzeyi (Sig.) $p=0,155$ ’tir ve bu değer 0,05 değerinden büyüktür. Bu nedenle çözümler arasındaki ilişkinin incelenmesinde parametrik testlerden eşleştirilmiş t testi uygulanabileceğine karar verilmiştir.

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Çözümler Arasındaki Fark Değeri	,237	9	,155	,843	9	,063

Şekil 5.20 Kolmogorov Smirnov Testi Sonucu

5.5.3.2 Eşleştirilmiş T Testi

- Eşleştirilmiş t testi sonucu elde edilen p değeri, $p > 0,01$ ise; “ H_0 =Çözümler arasında fark yoktur.” hipotezi kabul edilir.
- Eşleştirilmiş t testi sonucu elde edilen p değeri, $p < 0,01$ ise; “ H_1 =Çözümler arasında fark vardır.” hipotezi kabul edilir.

	Ortalama	Standart Sapma	Standart Hata Ortalaması	%95 Güven Aralığı		t	df	Sig.
				En Alt Sınır	En Üst Sınır			
Optimal Çözüm-GACözüm	-,11667	,12247	,04082	-,21081	-,02252	-2,858	8	,021

Şekil 5.21 Eşleştirilmiş t Testi Sonuçları

Şekil 5.21’de eşleştirilmiş t testi sonuç tablosunun Sig. (Anlamlılık düzeyi) sütunundaki değerin 0,021 olduğu görülmektedir. Söz konusu değer, $p > 0,01$ olduğu için optimal çözüm ve özelleştirilmiş GA çözüm değerleri arasında fark olmadığı görülmektedir. H_0 hipotezi kabul edilir. Buradan iki yöntemle üretilen çözümlerin benzer sonuçlar verdiği sonucuna varılmaktadır.

Bu çalışmada, heterojen filolu milk-run ve çapraz sevkiyatlı teslim yöntemlerini içeren gerçek hayat dağıtım problemi için tam sayılı doğrusal programlama modeli ve meta-sezgisel algoritma geliştirilmiştir. Algoritmanın geliştirilmesinde genetik algoritmadan ve tasarruf algoritmasından faydalanılmıştır.

Geliştirilen matematiksel model ve algoritma, müşterilere ürünün en optimal maliyette ve zamanda ulaştırılabilmesi için en uygun dağıtım yönteminin ve en uygun rotanın belirlenmesini sağlamaktadır.

Yürütülen çalışma, direkt dağıtım, çapraz sevkiyat ve milk-run dağıtım yöntemlerinin üçünü de birlikte elen çalışmalardan; heterojen tipli ve limitsiz sayıda araç filosu kullanılması, mesafe kısıtlı milk-run rotalaması kullanılması, maliyet hesabında milk-run maliyetinde ortaya çıkan durak maliyetinin dikkate alınması, literatürde iyi bilinen ve araç rotalama problemlerinde etkinliği tanımlanmış genetik algoritma ve tasarruf algoritmasının birlikte kullanılması yönleriyle farklıdır.

Geliştirilen algoritmanın testi için 4-12 müşteri arası küçük, 20-50 arası orta, 100 müşterili büyük müşteri veri seti kullanılmıştır. 4-12 müşteri arası olan küçük veri seti için çözümler, tam sayılı doğrusal programlama modeli ve geliştirilen algoritma kullanılarak oluşturulmuştur. Tam sayılı doğrusal programlama modeli ile çözümde, lingo paket programı kullanılmıştır. Lingo paket programı ile elde edilen optimal sonuçlar ile geliştirilen özelleştirilmiş GA algoritması ile edilen sonuçlar karşılaştırıldığında, GA algoritmasının kısa sürede çözüm verebildiği gözlenmiştir. İki yöntemle elde edilen sonuçların benzer sonuçlar verip vermediğini test etmek için eşleştirilmiş t testinden faydalanılmıştır. Eşleştirilmiş t testi sonucunda, sonuçlar arasında fark olmadığı ve iki yöntemin benzer sonuçlar ürettiği görülmüştür.

Orta ve büyük müşteri veri seti için, problemin NP- zor yapısından ötürü, lingo ile çözüm üretilmemiştir. Geliştirilen algoritma ile sonuçlar elde edilmiştir. Ve sonuçların küçük veri setleri için test edilip, optimale yakın sonuçlar verdiği görüldüğünden orta ve büyük veri setleri için de optimale yakın sonuçlar verdiği kabul edilmiştir.

Genetik algorithma da yer alan popülasyon büyüklüğü, mutasyon oranı ve çaprazlama oranı parametreleri için literatür incelemesi yapılarak değerler belirlenmiştir. Parametre değerlerinin doğru kombinasyonu için istatistiksel yöntem olan taguchi yönteminden faydalanılarak en optimum parametre değerleri saptanmıştır. Bulunan optimum parametre değerleri, tüm müşteri veri setleri için çözümün oluşturulmasında kullanılmıştır.

Geliştirilen algoritma ve matematiksel model ile ilgili yapılabilecek gelecek çalışmalar ve öneriler şunlardır:

- Bu çalışmada geliştirilen tam sayılı doğrusal programlama modeli ve meta-sezgisel algoritma, benzer ağ yapısındaki problemlerin çözümünde kullanılabilir.
- Araç sayısının limitli olması durumunun önerilen rotalar, maliyetler ve çözüm süreleri üzerindeki etkileri incelenebilir.
- Talebin bölünebilir olduğu, siparişlerin bölünerek farklı araçlarla gönderiminin mümkün olduğu durumlar incelenebilir.
- Talebin stokastik yapıda yani belirsiz olduğu durumlar incelenebilir ve geliştirilen yöntemlerde bu durumlar dikkate alınabilir.
- İhtiyaç duyulması durumunda ek dağıtım merkezlerinin açılması veya kapanması durumları incelenebilir.

- [1] G. Carpaneto, M. Dell'amico, M. Fischetti and P. Toth, "A branch and bound algorithm for the multiple depot vehicle scheduling problem," *Networks*, vol. 19, no. 5, pp. 531-548, 1989.
- [2] P. Chandra, "A Dynamic Distribution Model with Warehouse and Customer Replenishment Requirements," *The Journal of the Operational Research Society*, vol. 44, no. 7, pp. 681-692, 1993.
- [3] P. Chandra and M. L. Fisher, "Coordination of production and distribution planning," *European Journal of Operational Research*, vol. 72, no. 3, pp. 503-517, 1994.
- [4] M. G. Speranza and W. Ukovich, "Minimizing Transportation and Inventory Costs for Several Products on a Single Link," *Operations Research*, vol. 42, no. 5, pp. 879 - 894, 1994.
- [5] J. Berger and M. Barkaoui, "A Hybrid Genetic Algorithm for the Capacitated Vehicle Routing Problem," in *Genetic and Evolutionary Computation Conference*, Springer, 2003, pp. 646-656.
- [6] L. Lei, S. Liu, A. Ruszczyński and S. Park, "On the Integrated Production, Inventory and Distribution Routing Problem," *IIE Transactions*, vol. 38, no. 11, pp. 955-970, 2006.
- [7] J. Liu, C. Li and C. Chan, "Mixed Truck Delivery Systems with both Hub-and-Spoke and Direct Shipment," *Transportation Research Part E*, vol. 39, pp. 325-339, 2003.
- [8] V. Gaur and M. L. Fisher, "A Periodic Inventory Routing Problem at a Supermarket Chain," *Operations Research*, vol. 52, no. 6, pp. 813 - 822, 2004.
- [9] M. Gümüş and J. H. Bookbinder, "Cross-docking and Its Implications in Location-Distribution Systems," *Journal of Business Logistics*, vol. 25, no. 2, pp. 199-228, 2004.
- [10] A. Rusdiansyah and D.-b. Tsao, "An integrated model of the periodic delivery problems for vending-machine supply chains," *Journal of Food Engineering*, vol. 70, no. 3, pp. 421-434, 2005.
- [11] O. Berman and Q. Wang, "Inbound Logistic Planning: Minimizing Transportation and Inventory Cost," *Transportation Science*, vol. 40, no. 3, pp. 287 - 299, 2006.

- [12] Y. Lee, W. Jung and K. Lee, "Vehicle routing scheduling for cross-docking in the supply chain," *Computers & Industrial Engineering*, vol. 51, no. 2, pp. 247-256, 2006.
- [13] P. Parthanadee and R. Logendran, "Periodic product distribution from multi-depots under limited supplies," *IIE Transactions*, vol. 38, no. 11, pp. 1009-1026, 2006.
- [14] J. Branke, D. Haußler and C. Schmidt, "Transport channel selection," in *Operations Research Proceedings*, Springer, 2006, pp. 349-354.
- [15] J. Alegre, M. Laguna and J. Pacheco, "Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts," *European Journal of Operational Research*, vol. 179, no. 3, pp. 736-746, 2007.
- [16] T. Du, F. Wang and P. Lu, "A real-time vehicle-dispatching system for consolidating milk runs," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 5, pp. 565-577, 2007.
- [17] B. Crevier, J. Cordeau and G. Laporte, "The Multi-Depot Vehicle Routing Problem with Inter-Depot Routes," *European Journal of Operational Research*, vol. 176, no. 2, pp. 756-773, 2007.
- [18] W. Peng and C. Zhou, "Solving Vehicle Routing Problem Using Ant Colony and Genetic Algorithm," in *International Conference on Intelligent Computing*, Shanghai, Springer, 2008, pp. 23-30.
- [19] J. Ohlmann, M. Fry and T. Barrett, "Route Design for Lean Production Systems," *Transportation Science*, vol. 42, no. 3, pp. 352-370, 2008.
- [20] B. Ombuki-Berman and F. Hanshar, *Using Genetic Algorithms for Multi-depot Vehicle Routing*, Berlin:Springer, 2009.
- [21] B. Raa and E. Aghezzaf, "A practical solution approach for the cyclic inventory routing problem," *European Journal of Operational Research*, vol. 192, no. 2, pp. 429-441, 2009.
- [22] S. Sadjadi, M. Jafari and T. Amini, "A new mathematical modeling and a genetic algorithm search for milk run problem (an auto industry supply chain case study)," *The International Journal of Advanced Manufacturing Technology*, vol. 44, p. 194, 2009.
- [23] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau and G. Laporte, "Vehicle Routing with Cross-Docking," *Journal of the Operational Research Society*, vol. 60, no. 12, pp. 1708-1718, 2009.

- [24] M. Jafari-Eskandari, S. Jalali-Naiini, A. Aliahmadi and S. Sadjadi, "A Robust Optimization Approach for the Milk Run Problem with Time Windows under Inventory Uncertainty - An Auto Industry Supply Chain Case Study," in Proceedings of the 2010 International Conference on Industrial Engineering and Operations Management, IEEE, 2010, pp. 1-7.
- [25] R. Musa, J. Arnaout and H. Jung, "Ant colony optimization algorithm to solve for the transportation problem of cross-docking network," Computers & Industrial Engineering, vol. 59, no. 1, pp. 85-92, 2010.
- [26] B. Peterson, J. W. Van Hoeve, S. Kekre and L. Debo, Flexible Milk-Runs for Stochastic Vehicle Routing, Tepper School of Business, Oct. 2010. Accessed on: Jan. 19, 2018. [online]. Available: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=2492&context=tepper>.
- [27] P. Pop, O. Matei and C. Pop Sitar, "A Genetic Algorithm for Solving the Generalized Vehicle Routing Problem," Hybrid Artificial Intelligence Systems, vol. 6077, pp. 119-126, 2010.
- [28] C. J. Liao, Y. Lin and S. C. Shih, "Vehicle routing with cross-docking in the supply chain," Expert Systems with Applications, vol. 37, no. 10, pp. 6868-6873, 2010.
- [29] T. Schöneberg, A. Koberstein and L. Suhl, "An optimization model for automated selection of economic and ecologic delivery profiles in area forwarding," Flex. Serv. Manuf. J., vol. 22, pp. 214-235, 2010.
- [30] H. Charkhgard and A. Tabar Yahya, "Transportation problem of cross-docking network with three-dimensional trucks," African Journal of Business Management, vol. 5, no. 22, pp. 9297-9303, 2011.
- [31] V. Mahdavi, S. Sadeghi and M. O. Makhmalbaf, "A stochastic hybrid algorithm for multi-depot and multi-product routing problem with heterogeneous vehicles," Journal of American Science, vol. 7, no. 9, pp. 1006-1015, 2011.
- [32] F. Santos, G. Mateus and A. Cunha, "A Branch-and-price algorithm for a Vehicle Routing Problem with Cross-Docking," Electronic Notes in Discrete Mathematics, vol. 37, no. 1, pp. 249-254, 2011.
- [33] R. Dondo, C. A. Méndez and J. Cerdá, "The multi-echelon vehicle routing problem with cross docking in supply chain management," Computers & Chemical Engineering, vol. 35, no. 12, pp. 3002-3024, 2011.
- [34] T. Vidal, G. T. Crainic, M. Gendreau, N. Lahrichi and W. Rei, "A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems," Operations Research, vol. 60, no. 3, pp. 611 - 624, 2012.

- [35] A. Hasani-Goodarzi and R. Tavakkoli-Moghaddam, "Capacitated Vehicle Routing Problem for Multi-Product Cross- Docking with Split Deliveries and Pickups," *Procedia - Social and Behavioral Sciences*, vol. 62, pp. 1360-1365, 2012.
- [36] J. Baños, C. Ortega, L. Gil, A. Márquez and F. Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 286-296, 2013.
- [37] C. Çetinkaya, İ. Karaoglan and H. Gökçen, "Two-stage vehicle routing problem with arc time windows: A mixed integer programming formulation and a heuristic approach," *European Journal of Operational Research*, vol. 230, no. 3, pp. 539-550, 2013.
- [38] S. Liu, "A hybrid population heuristic for the heterogeneous vehicle routing problems," *Transportation Research Part E*, vol. 54, pp. 67-78, 2013.
- [39] S. Mousavi Meysam and R. Tavakkoli-Moghaddam, "A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain," *Journal of Manufacturing Systems*, vol. 32, no. 2, pp. 335-347, 2013.
- [40] P. Pop, O. Matei and C. Pop Sitar, "An improved hybrid algorithm for solving the generalized vehicle routing problem," *Neurocomputing*, vol. 109, pp. 76-83, 2013.
- [41] T. Sinphatsirikul, P. Nantaseri, S. Penpokai, K. Yusatharporn and A. Dumrongsiri, "Mathematical Model for Combined Outbound and Inbound Logistics Using Milk Run Deliveries," in *Proceedings of the 4th International Conference on Engineering, EPPM*, 2013, pp. 871-879.
- [42] D. Gyulai, A. Pfeiffer, T. Sobottka and J. Vancza, "Milk-run vehicle routing problem approach for shop-floor logistics," *Procedia CIRP*, vol. 7, pp. 127-132, 2013.
- [43] S. Mousavi, R. Tavakkoli-Moghaddam and F. Jolai, "A possibilistic programming approach for the location problem of multiple cross-docks and vehicle routing scheduling under uncertainty," *Engineering Optimization*, vol. 45, no. 10, pp. 1223-1249, 2013.
- [44] F. Santos, G. Mateus and A. Salles da Cunha, "The Pickup and Delivery Problem with Cross-Docking," *Computers & Operations Research*, vol. 40, no. 4, pp. 1085-1093, 2013.
- [45] C. Tarantilis, "Adaptive multi-restart Tabu Search algorithm for the vehicle routing problem with cross-docking," *Optimization Letters*, vol. 7, pp. 1583-1596, 2013.

- [46] R. Dondo and J. Cerdá, "A sweep-heuristic based formulation for the vehicle routing problem with cross-docking," *Computers & Chemical Engineering*, vol. 48, pp. 293-311, 2013.
- [47] D. Agustina, C. Lee and R. Piplani, "Vehicle scheduling and routing at a cross docking center for food supply chains," *International Journal of Production Economics*, vol. 152, pp. 29-41, 2014.
- [48] H. M. Hà, N. Bostel, A. Langevin and L. Rousseau, "An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size," *Computers & Operations Research*, vol. 43, pp. 9-19, 2014.
- [49] S. D. Hosseini, M. A. Shirazi and B. Karimi, "Cross-docking and milk run logistics in a consolidation network: A hybrid of harmony search and simulated annealing approach," *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 567-577, 2014.
- [50] M. Adelzadeh, V. M. Asl and M. Koosha, "A mathematical model and a solving procedure for multi-depot vehicle routing problem with fuzzy time window and heterogeneous vehicle," *The International Journal of Advanced Manufacturing Technology*, vol. 75, no. 5-8, pp. 793-802, 2014.
- [51] Z. Chen and R. B. Sarker, "An integrated optimal inventory lot-sizing and vehicle-routing model for a multisupplier single-assembler system with JIT delivery," *International Journal of Production Research*, vol. 52, no. 17, pp. 5086-5114, 2014.
- [52] R. Dondo and J. Cerdá, "A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors," *Computers & Chemical Engineering*, vol. 63, pp. 184-205, 2014.
- [53] S. Moghadam, S. Fatemi Ghomi and B. Karimi, "Vehicle routing scheduling problem with cross docking and split deliveries," *Computers & Chemical Engineering*, vol. 69, no. 3, pp. 98-107, 2014.
- [54] V. W. Morais, G. R. Mateus and T. F. Noronha, "Iterated local search heuristics for the Vehicle Routing Problem with Cross-Docking," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7495-7506, 2014.
- [55] Z. You and Y. Jiao, "Development and Application of Milk-Run Distribution Systems in the Express Industry Based on Saving Algorithm," *Mathematical Problems in Engineering*, pp. 1-6, 2014.
- [56] S. Mousavi Meysam, B. Vahdani, R. Tavakkoli-Moghaddam and H. Hashemi, "Location of cross-docking centers and vehicle routing scheduling under

uncertainty: A fuzzy possibilistic–stochastic programming model," *Applied Mathematical Modelling*, vol. 38, no. 7-8, pp. 2249-2264, 2014.

- [57] S. Mousavi Meysam, B. Vahdani, R. Tavakkoli-Moghaddam and H. Hashemi, "A fuzzy possibilistic–stochastic programming model," *European Journal of Operational Research*, vol. 75, no. 11, pp. 3500-3615, 2016.
- [58] S. Allahyari, M. Salari and D. Vigo, "A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem," *European Journal of Operational Research*, vol. 242, no. 3, pp. 756-768, 2015.
- [59] İ. Küçükoğlu and N. Öztürk, "An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows," *Computers & Industrial Engineering*, vol. 86, pp. 60-68, 2015.
- [60] M. Mokhtarinejad, A. Ahmadi, B. Karimi and S. Rahmati, "A novel learning based approach for a new integrated location-routing and scheduling problem within cross-docking considering direct shipment," *Applied Soft Computing*, vol. 34, pp. 274-285, 2015.
- [61] B. Yang, Z. Hu, C. Wei, S.-Q. Li, L. Zhao and S. Jia, "Routing with time-windows for multiple environmental vehicle types," *Computers & Industrial Engineering*, vol. 89, pp. 150-161, 2015.
- [62] F. Ahmadizar, M. Zeynivand and J. Arkat, "Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach," *Applied Mathematical Modelling*, vol. 39, no. 22, pp. 7065-7081, 2015.
- [63] R. Dondo and J. Cerdá, "The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system," *Computers & Chemical Engineering*, vol. 76, pp. 42-62, 2015.
- [64] A. Mohtashami, M. Tavana, F. J. Santos-Arteaga and A. Fallahian-Najafabadi, "A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems," *Applied Soft Computing*, vol. 31, pp. 30-47, 2015.
- [65] Y. Lin, Z. Bian, S. Shuijing and T. Xu, "A two-stage simulated annealing algorithm for the many-to-many milkrun routing problem with pipeline inventory cost," *Mathematical Problems in Engineering*, vol. 2015, pp. 1-22, 2015.
- [66] M. Cóccola, C. A. Méndez and R. G. Dondo, "A branch-and-price approach to evaluate the role of cross-docking operations in consolidated supply chains," *Computers and Chemical Engineering*, vol. 80, pp. 15-29, 2015.
- [67] F. V. Yu, P. Jewpanya and P. A. Redi, "Open vehicle routing problem with cross-docking," *Computers & Industrial Engineering*, vol. 94, pp. 6-17, 2016.

- [68] H. Bae and I. Moon, "Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles," *Applied Mathematical Modelling*, vol. 40, pp. 13-14, 2016.
- [69] Ş. Birim, "Vehicle routing problem with cross docking: A simulated annealing approach," *Procedia - Social and Behavioral Sciences*, vol. 235, pp. 149-158, 2016.
- [70] F. Farshchi, D. Jafari and S. Moghaddam, "Integrating Vehicle Routing With Cross Dock In Supply Chain," *The Turkish Online Journal of Design Art and Communication*, vol. 6, 2016.
- [71] A. Goodarzi and S. H. Zegordi, "A location-routing problem for cross-docking networks: A biogeography-based optimization algorithm," *Computers & Industrial Engineering*, vol. 102, pp. 132-146, 2016.
- [72] Ç. Koç, T. Bektaş, O. Jabali and G. Laporte, "The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm," *European Journal of Operational Research*, vol. 248, no. 1, pp. 33-51, 2016.
- [73] J. Liu, A. Smith and D. Qian, "The vehicle loading problem with a heterogeneous transport fleet," *Computers & Industrial Engineering*, vol. 97, pp. 137-145, 2016.
- [74] R. B. Lopes, C. Ferreira and B. Santos, "A simple and effective evolutionary algorithm for the capacitated location-routing problem," *Computers & Operations Research*, vol. 70, pp. 155-162, 2016.
- [75] Y. B. Park, J. Yoo and H. Park, "A genetic algorithm for the vendor-managed inventory routing problem with lost sales," *Expert Systems with Applications*, vol. 53, no. 1, pp. 149-159, 2016.
- [76] T. Yalian, "An Improved Ant Colony Optimization for Multi-Depot Vehicle Routing Problem," *International Journal of Engineering and Technology*, vol. 8, no. 5, pp. 385-388, 2016.
- [77] P. Yin and Y. Chuang, "Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking," *Applied Mathematical Modelling*, vol. 40, no. 21-22, pp. 9302-9315, 2016.
- [78] A. Nikolopoulou, P. Repoussis, C. Tarantilis and E. Zachariadis, "Adaptive memory programming for the many-to-many vehicle routing problem with cross-docking," *Operational Research*, vol. 19, no. 1, pp. 1-38, 2016.
- [79] Y. Maknoon and G. Laporte, "Vehicle routing with cross-dock selection," *Computers & Operations Research*, vol. 77, pp. 254-266, 2016.

- [80] A. N. Qamsari, S. Hosseini Motlagh and A. Jokar, "A Two-Phase Hybrid Heuristic Method for a Multi-Depot Inventory-Routing Problem," *International Journal of Transportation Engineering*, vol. 4, no. 16, pp. 287-304, 2017.
- [81] D. Zhang, S. Cai, F. Ye, Y. Si and T. T. Nguyen, "A hybrid algorithm for a vehicle routing problem with realistic constraints," *Information Sciences*, vol. 394–395, pp. 167-182, 2017.
- [82] A. Hiassat, A. Diabat and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," *Journal of Manufacturing Systems*, vol. 42, pp. 93-103, 2017.
- [83] Y. Maknoon and G. Laporte, "Vehicle routing with cross-dock selection," *Computers & Operations Research*, vol. 77, pp. 254-266, 2017.
- [84] H. Mei, Y. Jingshuai, M. Teng, L. Xiuli and W. Ting, "The Modeling of Milk-run Vehicle Routing Problem Based on Improved C-W Algorithm that Joined Time Window," *Transportation Research Procedia*, vol. 25, pp. 716-728, 2017.
- [85] E. Baran, "A Multi-Objective Approach for The In-Plant Milk-Run in A Furniture Factory," *International Journal of Engineering Technologies*, vol. 3, no. 4, pp. 186-189, 2017.
- [86] A. Baniamerian, M. Bashiri and F. Zabihi, "Two phase genetic algorithm for vehicle routing and scheduling problem with cross-docking and time windows considering Customer satisfaction," *Journal of Industrial Engineering International*, vol. 4, no. 1, pp. 15-30, 2017.
- [87] J. Wang, A. K. Ranganathan Jagannathan, X. Zuo and C. Murrayd, "Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries," *Computers & Industrial Engineering*, vol. 112, pp. 84-98, 2017.
- [88] P. Grangier, M. Gendreau, F. Lehuédé and L. Rousseau, "A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking," *Computers & Operations Research*, vol. 84, pp. 116-126, 2017.
- [89] S. Mousavi and B. Vahdani, "A robust approach to multiple vehicle location-routing problems with time windows for optimization of cross-docking under uncertainty," *Journal of Intelligent & Fuzzy Systems*, vol. 32, pp. 49-62, 2017.
- [90] A. Meyer and B. Amberg, "Transport concept selection considering supplier milk runs – An integrated model and a case study from the automotive industry," *Transportation Research Part E: Logistics and Transportation Review*, vol. 113, pp. 147-169, 2018.

- [91] A. Ahkamiraad and Y. Wang, "Capacitated and multiple cross-docked vehicle routing problem with pickup, delivery, and time windows," *Computers & Industrial Engineering*, vol. 119, pp. 76-84, 2018.
- [92] Y. Kaboudani, S. H. Ghodsypour, H. Kia and A. Shahmardan, "Vehicle routing and scheduling in cross docks with forward and reverse logistics," *Oper Res Int J*, pp. 1-34, 2018.
- [93] A. Goodarzi and S. H. Zegordi, "Vehicle routing problem in a kanban controlled supply chain system considering cross-docking strategy," *Operational Research*, pp. 1-29, 2018.
- [94] H. Yurtdaş, "Çapraz Sevkiyatta Araç Rotalama Problemi için Sezgisel Çözüm Yaklaşımı," *Doktora Tezi, Fen Bilimleri Enstitüsü, Gazi Üniversitesi, Ankara*, 2018.
- [95] A. Baniamerian, M. Bashiri and R. Tavakkoli-Moghaddam, "Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking," *Applied Soft Computing*, vol. 75, pp. 441-460, 2019.
- [96] A. Rahbari, M. Nasiri, F. Werner, M. M. Musavi and F. Jolai, "The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models," *Applied Mathematical Modelling*, vol. 70, pp. 605-625, 2019.
- [97] F. Ranjbaran, H. A. Kashan and A. Kazemi, "Mathematical formulation and heuristic algorithms for optimisation of auto-part milk-run logistics network considering forward and reverse flow of pallets," *International Journal of Production Research*, pp. 1-35, 2019.
- [98] A. Gunasekaran, C. Patel and E. R. McGaughey, "A framework for supply chain performance measurement," *Int. J. Production Economics*, vol. 87, pp. 333-347, 2004.
- [99] S. Chopra and P. Meindl, *Supply Chain Management*. New Jersey, Pearson Education Inc, 2007
- [100] N. Sevinç, "Tedarik Zinciri Yönetiminde Bilgi Teknolojilerinin Kullanılması ve Önemi," *Yüksek Lisans Tezi, Sosyal Bilimler Enstitüsü, Trakya Üniversitesi, Edirne*, 2008.
- [101] L. Ellram and A. Carr, "Strategic Purchasing: A History and Review of the Literature," *International Journal of Purchasing and Materials Management*, vol. 30, no. 2, pp. 10-18, 1994.

- [102] K. H. Güleş, T. Paksoy, H. Bülbül ve E. Özceylan, Stratejik Planlama, Modelleme ve Optimizasyon. Ankara, Gazi Kitabevi, 2012.
- [103] M. S. Fox, J. F. Chionglo and M. Barbuçounu, The Integrated Supply Chain Management System, Dec. 1993. Accessed on: Jun. 27, 2019, [Online]. Available: <https://www.researchgate.net/publication/239033756>
- [104] H. Yalçın, "Tedarik Zinciri Yönetiminde Bilişim Teknolojileri," Yüksek Lisans Tezi, Sosyal Bilimler Enstitüsü, Sakarya Üniversitesi, Sakarya, 2004.
- [105] T. Paksoy and F. Altıparmak, "Dağıtım Ağlarının Tasarımı ve Eniyilemesi Kapsamında Tedarik Zinciri ve Lojistik Yönetimine Bir Bakış: Son Gelişmeler ve Genel Durum," Sigma, pp. 149-166, 2003.
- [106] H. Jung, H. Chen and B. Jeong, Trends in Supply Chain Design and Management Technologies and Methodologies. Springer, 2007.
- [107] M. Hugos, Essentials of supply chain management. Newyork, John Wiley & Sons, 2003.
- [108] T. Wall, Lean Manufacturing Lessons From a Road Raged Operations Executive, The Northwest Lean Networks, 2003.
- [109] S. Yıldız, "Lojistik Yönetiminde Milkrun Planlaması ve Çapraz Sevkiyat Uygulaması," Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Kocaeli Üniversitesi, 2008.
- [110] C. Marchwinski, J. Shook, A. Schroeder, J. Ferro, D. Jones and J. Womack, Lean Lexicon A Graphical Glossary for Lean Thinkers. USA, The Lean Enterprise Institute Cambridge, 2008.
- [111] R. Ray, Supply chain management for retailing. New Delhi, Tata McGrawHill Education, 2010.
- [112] C. Duhamel, J.-Y. Potvin and J.-M. Rousseau, "A tabu search heuristic for the vehicle routing problem with backhauls and time windows," Transportation Science, vol. 31, no. 1, pp. 49-59, 1997.
- [113] M. Çancı ve M. Erdal, Lojistik Yönetimi. İstanbul, Erler Matbaacılık San. ve Tic. A.Ş., 2003.
- [114] G. Dantzig and J. Ramser, "The Truck Dispatching Problem," Management Science, vol. 6, pp. 80-91, 1959.
- [115] G. Clarke and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, vol. 12, pp. 568-581, 1964.

- [116] T. L. Magnanti, "Combinatorial Optimization and Vehicle Fleet Planning Perspectives and Prospects," *Networks*, vol. 11, no. 2, pp. 179-213, 1981.
- [117] M. Gendreau, G. Laporte and R. Séguin, "Stochastic vehicle routing," *European Journal of Operational Research*, vol. 88, no. 1, pp. 3-12, 1996.
- [118] G. Laporte, "The Vehicle Routing Problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp. 345-358, 1992.
- [119] D. Kirby, "Is your fleet the right size?," *Journal of the Operational Research Society*, vol. 10, no. 4, p. 252, 1959.
- [120] B. Golden, A. Assad, L. Levy and F. Gheysen, "The fleet size and mix vehicle routing problem," *Computers and Operations Research*, vol. 11, pp. 49-66, 1984.
- [121] L. Schrage, "Formulation and structure of more complex/realistic routing and scheduling problems," *Networks*, vol. 11, pp. 229-232, 1981.
- [122] D. Sariklis and S. Powell, "A heuristic method for the open vehicle routing problem," *Operational Research Society*, vol. 51, pp. 564-573, 2000.
- [123] X. Li, S. Leung and P. Tian, "Multi Start Adaptive Memory-based Tabu Search Algorithm for the Heterogeneous Fixed Fleet Open Vehicle Routing Problem," *Expert Systems with Applications*, vol. 39, pp. 365-374, 2012.
- [124] T. Keskindürk, N. Topuk and O. Özyeşil, "Araç Rotalama Problemleri ile Çözüm Yöntemlerinin Sınıflandırılması ve Bir Uygulama," *İşletme Bilimi Dergisi*, vol. 3, no. 2, pp. 77-107, 2015.
- [125] E. Y. Demirtaş and E. Özdemir, "Dinamik Araç Rotalama Problemleri için Yeni Bir Çözüm Önerisi," *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 22, no. 3, pp. 807-823, 2017.
- [126] M. Dror and P. Trudeau, "Savings by Split Delivery Routing, *Transportation*," vol. 23, no. 2, pp. 141-145, 1989.
- [127] M. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations Research*, vol. 4, pp. 285-305, 1985.
- [128] G. Laporte and I. Osman, "Routing problems: a bibliography," *Annals of Operations Research*, vol. 61, pp. 1227-262, 1995.
- [129] A. Aksoy and N. Öztürk, "A Two-Stage Method to Optimize The Milk-Run System," *European Journal of Engineering Research & Science*, vol. 1, no. 3, pp. 7-11, 2016.

- [130] S. Kumar and R. Panneerselvam, "A Survey on the Vehicle Routing Problem and Its Variants," *Intelligent Information Management*, vol. 4, pp. 66-74, 2012.
- [131] M. Lai, "A Hybrid Genetic/Benders' Algorithm with Application to Vehicle routing and scheduling problems," *Doktora Tezi*, Iowa Üniversitesi, Iowa, 2004.
- [132] M. Demircioğlu and E. Düzakın, "Araç Rotalama Problemleri ve Çözüm Yöntemleri," *Çukurova Üniversitesi İİBF Dergisi*, vol. 13, no. 1, pp. 68-87, 2009.
- [133] A. Yücel, "Mesafe Kısıtlı Çok Yönlü Kümelenmiş Açık Rotalama Probleminin Genetik Algoritma ile Çözümü ve Bir Uygulama," *Doktora Tezi*, Sosyal Bilimler Sayısal Yöntemler, İstanbul Üniversitesi, İstanbul, 2016.
- [134] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308, 2003.
- [135] G. Laporte, M. Gendreau, J. Y. Potvin and F. Semet, "Classical And Modern Heuristics For The Vehicle Routing Problem," *International Transactions in Operation Research*, vol. 7, pp. 285-300, 2000.
- [136] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. USA, The University of Michigan Press, 1975.
- [137] Y. Şahin and A. Eroğlu, "Kapasite Kısıtlı Araç Rotalama Problemi İçin Metasezgisel Yöntemler: Bilimsel Yazın Taraması," *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 19, no. 4, pp. 337-355, 2014.
- [138] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [139] M. A. Akçayol, *Zeki Optimizasyon Teknikleri: Ders Notları*, Accessed on: Jan. 27, 2018. [Online]. Available: <http://w3.gazi.edu.tr/~akcayol/files/ZOL3Tavlama.pdf>.
- [140] B. Keçeci, "Heterojen Eş Zamanlı Topla Dağıt Araç Rotalama Problemi için Matematiksel Modeller ve Sezgisel Yaklaşımlar," *Doktora Tezi*, Fen Bilimleri Enstitüsü, Gazi Üniversitesi Ankara, 2014.
- [141] "IEOR151 – Lecture 18:Savings Algorithm," Accessed on: Jan. 28,2018. [Online]. Available:http://ieor.berkeley.edu/~ieor151/lecture_notes/ieor151_lec18.pdf.

- [142] D. E. Goldenberg, Genetic Algorithms in Search Optimization and Machine Learning. Boston, Addison Wesley Longman, 1989.
- [143] L. Schmitt and M. M. Amini, "Performance characteristics of alternative genetic algorithmic approaches to the traveling salesman problem using path representation: An empirical study," vol. 108, pp. 551-570, 1998.
- [144] O. Engin, "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu," Doktora Tezi, Fen Bilimleri Enstitüsü, İTÜ, İstanbul, 2001.
- [145] Ç. Taşkın and G. G. Emel, "Genetik Algoritmalar ve Uygulama Alanları," Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, vol. 1, no. 21, pp. 129-152, 2002.
- [146] Ö. İşçi and S. Korukoğlu, "Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama," Yönetim ve Ekonomi: Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakülte Dergisi, vol. 10, no. 2, p. 193, 2003.
- [147] T. Back, D. B. Fogel and Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators. Philadelphia, Institute of Physics Publishing, 2000.
- [148] P. Larranaga, C. Kuijpers, R. Murga, I. Inza and S. Dizdarevic, Artificial Intelligence Review: Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators, Netherlands, Kluwer Academic Publishers, 1999.
- [149] A. Kumar, "Encoding Schemes in Genetic Algorithm," International Journal of Advanced Research in IT and Engineering, vol. 2, no. 3, pp. 1-7, 2013.
- [150] C. Iris and S. S. Asan, "A Review of Genetic Algorithm Applications in Supply Chain Network Design," Computational Intelligence Systems in Industrial Engineering, vol. 6, pp. 203-230, 2012.
- [151] J. Jang, C. Sun and E. Mizutani, A Computational Approach To Learning and Machine Intelligence. USA, Prentice-Hall, 1997.
- [152] M. Sakawa, "Genetic Algorithms and Fuzzy Multiobjective Optimization," Science & Business Media, vol. 14, pp. 1-287, 2012.
- [153] M. Gen and R. Cheng, Genetic Algorithms and Engineering Design. USA, John Wiley and Son, 1997.
- [154] A. Chen, J. Kim, S. Lee and Y. Kim, "Stochastic multi-objective models for network design problem," Expert Systems with Applications, vol. 37, no. 2, pp. 1608-1619, 2010.

- [155] L. Lin, M. Gen and X. Wang, "Integrated multistage logistics network design by using hybrid evolutionary algorithm," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 854–873, 2009.
- [156] J. Xu, Q. Liu and R. Wang, "A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor," *Information Sciences*, vol. 178, no. 8, pp. 2022–2043, 2008.
- [157] A. Costa, G. Celano, S. Fichera and E. Trovato, "A new efficient encoding/decoding procedure for the design of a supply chain network with genetic algorithms," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 986–999, 2010.
- [158] M. Gen, R. Cheng and S. Oren, "Network design techniques using adapted genetic algorithms," *Advances in Engineering Software*, vol. 32, no. 9, pp. 731–744, 2001.
- [159] F. Altıparmak, B. Dengiz and A. Smith, "An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks," vol. 12, no. 1-2, pp. 57-75, 2000.
- [160] K. DeJong, "An analysis of the behaviour of a class of genetic adaptive systems," PhD thesis, University of Michigan, USA, 1975.
- [161] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems Man, and Cybernetics*, vol. 16, pp. 122-128, 1986.
- [162] L. Lin, M. Gen and X. Wang, "A Hybrid Genetic Algorithm for Logistics Network Design with Flexible Multistage Model," *International Journal of Information Systems for Logistics and Management*, vol. 3, no. 1, pp. 1-12, 2007.
- [163] H. Guner Goren, S. Tunali and R. Jans, "A review of applications of genetic algorithms in lot sizing," *Journal of Intelligent Manufacturing*, vol. 21, no. 4, p. 575–590, 2010.
- [164] K. Sourirajan, L. Ozsen and R. Uzsoy, "A genetic algorithm for a single product network design model with lead time and safety stock considerations," *European Journal of Operational Research*, vol. 197, no. 2, pp. 599–608, 2009.
- [165] L. Lin, M. Gen and X. Wang, "A Hybrid Genetic Algorithm for Logistics Network Design with Flexible Multistage Model," *International Journal of the Information Systems for Logistics and Management*, vol. 3, no. 1, pp. 1-12, 2007.
- [166] L. Lin, M. Gen and X. Wang, "Integrated multistage logistics network design by using hybrid evolutionary algorithm," *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 854-873, 2009.

- [167] E. Canıyılmaz and F. Kutay, "Taguchi Metodunda Varyans Analizine Alternatif Yaklaşım," Gazi Üniv. Müh. Mim. Fak. Der., vol. 18, no. 3, pp. 51-63, 2003.
- [168] G. Taguchi and S. Konishi, "Taguchi Methods, orthogonal arrays and linear graphs, tools for quality American supplier," American Supplier Institute, pp. 8-35, 1987.
- [169] S. Athreya and Y. Dr. Venkatesh, "Application Of Taguchi Method For Optimization Of Process Parameters In Improving The Surface Roughness Of Lathe Facing Operation," International Refereed Journal of Engineering and Science, vol. 1, no. 3, pp. 13-19, 2012.
- [170] G. Gökçe and S. Taşgetiren, "Kalite için Deney Tasarımı," Makine Teknolojileri Elektronik Dergisi, vol. 6, no. 1, pp. 71-83, 2009.
- [171] S. Yıldırım, "Ürün Tasarımı Geliştirilmesi: Taguchi Tasarımı," Yüksek Lisans Tezi, Endüstri Mühendisliği, Başkent Üniversitesi, Ankara, 2011.

Özelleştirilmiş Genetik Algoritma Kaynak Kodu

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
namespace DistributionGenetic
{
    class Genetic
    {
        int m_customer, m_population, m_distrestriction, m_iteration;
        double [,] m_customerArray, m_saving;
        int[,] m_genes, m_milkrunlist, milkruncustomer;
        int[] numberofmilkrun;
        double besttotalcost, m_tourconstriction ;
        double[] m_factoryArray, m_platformArray, m_weightArray, m_costplatArray,
m_trucktype, m_truckconsumption, totalcost, m_trucktypecost;
        double m_crossoverProb, m_mutationProb, m_crossover, m_mutation, m_selection;
        private static readonly Random RastgeleDeger = new Random();
        public string yazdirma, elapsedMS;

        public Genetic(int Customer, int Distrestriction, double[,] CustomerArray, double[]
FactoryArray, double[] PlatformArray, double[] WeightArray, double[] CostPlatArray,
double CrossoverProb, double MutationProb, int Population, int Iteration, int Crossover, int
Mutation, int Selection, double tourconstriction)
        {
            m_customer = Customer; m_customerArray = CustomerArray; m_factoryArray =
FactoryArray; m_platformArray = PlatformArray;
            m_weightArray = WeightArray; m_crossoverProb = CrossoverProb; m_mutationProb
= MutationProb;
            m_population = Population; m_iteration = Iteration; m_distrestriction =
Distrestriction;
            m_crossover = Crossover; m_mutation = Mutation; m_selection = Selection;
m_costplatArray = CostPlatArray;
            m_tourconstriction = tourconstriction;
            m_trucktype = new double [5];

```

```

    m_truckconsumption = new double[5];
    m_trucktypecost = new double[5];
    m_trucktype[0] = 9.0; m_trucktype[1] = 25.0; m_trucktype[2] = 30.0; m_trucktype[3]
= 50.0; m_trucktype[4] = 50.0;

    m_truckconsumption[0] = 1.0; m_truckconsumption[1] = 1.0;
    m_truckconsumption[2] = 1.0; m_truckconsumption[3] = 1.0; m_truckconsumption[4] =
1.0;

    m_trucktypecost[0] = 279.6; m_trucktypecost[1] = 364.3; m_trucktypecost[2] =
437.3; m_trucktypecost[3] = 521.7; m_trucktypecost[4] = 521.7;

    m_milkrunlist = new int[m_population, m_customer];
    milkruncustomer = new int[m_population, m_customer
    numberofmilkrun = new int[m_population];
    totalcost = new double[m_population];
    for (int k = 0; k < 1; k++)
    {
        besttotalcost = 0.0;
    }
}
{
    Stopwatch sw2 = new Stopwatch();
    sw2.Start();
    GenRecord[] bestgenrecord = new GenRecord[m_iteration]
    m_genes = new int[m_population, m_customer];
    m_saving = new double[m_customer-1, m_customer];

    m_saving = determiningsaving(m_customerArray);
    firstiteration(m_genes);
    for (int q = 0; q < m_iteration; q++)
    {
        bestgenrecord[q] = new GenRecord()
        milkrunprocessing(m_genes, milkruncustomer, numberofmilkrun);
        totalcostandroutess(m_genes, m_saving, m_milkrunlist, bestgenrecord, q);
        geneticsphase(m_genes, m_mutation, m_mutationProb, m_population,
m_crossover, m_crossoverProb, bestgenrecord, besttotalcost);
    }
}

```

```

    }
    displayresults(bestgenrecord, m_iteration-1);
    TimeSpan elapsedTime = sw2.Elapsed;
    elapsedMS = elapsedTime.TotalMilliseconds.ToString();
    sw2.Stop();
    yazdirma += "    Süre: " + elapsedMS + "\n";
}
private void displayresults(GenRecord[] bestgenrecord, int q)
{
    yazdirma += q + ". Iteration Results: ";
    for (int i = 0; i < m_customer; i++)
    {
        yazdirma += bestgenrecord[q].gen[i] + ", ";
    }
    yazdirma += "\n";
    for (int i = 0; i < m_customer; i++)
    {
        int l = i + 1;
        if (Convert.ToInt32(bestgenrecord[q].gen[i]) == 1)
            yazdirma += "        Customer " + l + ": " + "From Platform";
        else if (Convert.ToInt32(bestgenrecord[q].gen[i]) == 2)
            yazdirma += "        Customer " + l + ": " + "From Factory";
        else
            yazdirma += "        Customer " + l + ": " + "From Milkrun";
        yazdirma += "\n";
    }
    yazdirma += "Iteration: " + bestgenrecord[q].bestiteration + "\n";
    yazdirma += "Total Cost: " + besttotalcost
    yazdirma += "\n";

    int t1 = bestgenrecord[q].numroutes.Count();
    for (int k = 0; k < bestgenrecord[q].numroutes.Count(); k++)

```

```

{
    if (bestgenrecord[q].numroutes[k].Count > 0)
    {
        yazdirma += " Rota " + (k) + ": ";
        for (int t = 0; t < bestgenrecord[q].numroutes[k].Count; t++)
            yazdirma += bestgenrecord[q].numroutes[k][t] + ", ";

        yazdirma += " " + "Araç Tipi: " + bestgenrecord[q].trucktype[k];
        yazdirma += " " + "Tonaaj: " + bestgenrecord[q].weight[k];
        yazdirma += "\n";
    }
    else
        yazdirma += "\n";
}

private void geneticsphase(int[,] m_genes, double m_mutation, double
m_mutationProb, int m_population, double m_crossover, double m_crossoverProb,
GenRecord[] bestgenrecord, double besttotalcost)
{
    int nummutation = Convert.ToInt32(Math.Round(m_population * m_mutationProb));
    int numcrossover = Convert.ToInt32(Math.Round(m_population *
m_crossoverProb));
    double allcost=0;
    double[] selectionprob = new double[m_population];
    for (
        int i = 0; i < m_population; i++)
        allcost += 1 / totalcost[i];
    for (int k = 0; k < m_population; k++)
        selectionprob[k] = (1/totalcost[k]) / allcost;
    int[,] offspring = new int[(2 * numcrossover), m_customer];

    int[] pick = new int[2];

```

```

for (int t = 0; t < numcrossover; t++)
{
    int[] copygene1 = new int[m_customer];
    int[] copygene2 = new int[m_customer];
    for (int l = 0; l < 2; l++)
    {
        double offset = 0;
        double rdn = RastgeleDeger.NextDouble();
        for (int j = 0; j < m_population; j++)
        {
            offset += selectionprob[j];
            if (rdn < offset)
            {
                pick[l] = j;
                break;
            }
        }
    }
    for (int f1 = 0; f1 < m_customer; f1++)
    {
        copygene1[f1] = m_genes[pick[0], f1];
        copygene2[f1] = m_genes[pick[1], f1];
    }
    int randomlocation = 0;
    randomlocation = RastgeleDeger.Next(m_customer);

    for (int f2 = randomlocation; f2 < m_customer; f2++)
    {
        copygene2[f2] = m_genes[pick[0], f2];
        copygene1[f2] = m_genes[pick[1], f2];
    }
}

```

```

    for (int f3 = 0; f3 < m_customer; f3++)
    {
        offspring[2*t,f3] = copygene1[f3];
        offspring[2*t+1, f3] = copygene2[f3];
    }
}
for (int t1 = 0; t1 < numcrossover; t1++)
{
    for (int f4 = 0; f4 < m_customer; f4++)
    {
        m_genes[t1,f4] = offspring[t1, f4];
        m_genes[t1 + 1,f4] = offspring[t1 + 1, f4];
    }
}
for (int t2 = 0; t2 < nummutation; t2++)
{
    int randomcustomer = 0;
    randomcustomer = RastgeleDeger.Next(0, m_population - 2);
    int randomgene = 0;
    randomgene = RastgeleDeger.Next(m_customer);
    double randomnum = RastgeleDeger.NextDouble();
}
else
{
    if (randomnum < 0.5)
        m_genes[randomcustomer, randomgene] = 1;
    else
        m_genes[randomcustomer, randomgene] = 2;
}
}
}

```

```

private void totalcostandroutest(int[,] m_genes, double[,] m_saving, int[,] m_milkrunlist,
GenRecord[] bestgenrecored, int q)
{
    for (int i = 0; i < m_population; i++)
    {
        int subtotalsaving = (((numberofmilkrun[i]) * (numberofmilkrun[i] - 1))/2);
        double[,] submilkrungroup = new double[subtotalsaving, 3];
        int i1 = 0;
        for (int k = 0; k < numberofmilkrun[i]; k++)
        {
            for (int l = k + 1; l < numberofmilkrun[i]; l++)
            {
                submilkrungroup[i1, 0] = m_saving[milkruncustomer[i, k],
milkruncustomer[i, l]];
                submilkrungroup[i1, 1] = milkruncustomer[i, k];
                submilkrungroup[i1, 2] = milkruncustomer[i, l];
                i1 += 1;
            }
        }
        #region /---Tasarruflar Sıralanır---\
        for (int k = 0; k < subtotalsaving; k++)
        {
            for (int j = k + 1; j < subtotalsaving; j++)
            {
                if (submilkrungroup[k, 0] < submilkrungroup[j, 0])
                {
                    double[,] contemporary = new double[1, 3];
                    contemporary[0, 0] = submilkrungroup[k, 0];
                    contemporary[0, 1] = submilkrungroup[k, 1];
                    contemporary[0, 2] = submilkrungroup[k, 2];
                    submilkrungroup[k, 0] = submilkrungroup[j, 0];
                    submilkrungroup[k, 1] = submilkrungroup[j, 1];
                    submilkrungroup[k, 2] = submilkrungroup[j, 2];
                }
            }
        }
    }
}

```

```

        submilkrungroup[j, 0] = contemporary[0, 0];
        submilkrungroup[j, 1] = contemporary[0, 1];
        submilkrungroup[j, 2] = contemporary[0, 2];
    }
}
}
#endregion
ArrayList complementofmilk = new ArrayList ();

for (int k = 0; k < numberofmilkrun[i]; k++)
    complementofmilk.Add( milkruncustomer[i, k]);
Rota[] milkrunrotalar = new Rota[numberofmilkrun[i]];
for (int r = 0; r < numberofmilkrun[i]; r++)
{
    milkrunrotalar[r] = new Rota();
    milkrunrotalar[r].m_routes = new ArrayList();
}
#region /----Tasarruf ---\
for (int sb = 0; sb < subtotalsaving; sb++)
{
    if ((RastgeleDeger.NextDouble() < 0.5) && (i < m_population-2) )
        goto Outer;
    if (submilkrungroup[sb, 0] != -1000)
    {
        if ((milkrunrotalar[0].weight == 0) )//
        {
            if (((2 * m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 1])] + 2 *
m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 2])] - submilkrungroup[sb, 0]) <=
m_tourconstriction))
            {
                double weightcheck =
m_weightArray[Convert.ToInt32(submilkrungroup[sb, 1])] +
m_weightArray[Convert.ToInt32(submilkrungroup[sb, 2])];

```

```

        if (weightcheck <= m_trucktype[3])
        {
            ilkrotabelirleme(milkrunrotalar, complementofmilk,
milkrunrotalar[0].weight, milkrunrotalar[0].startingcity, milkrunrotalar[0].finishingcity,
truck, submilkrungroup, sb, m_weightArray, weightcheck);

        }

        milkrunrotalar[0].routedistance = 2 *
m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 1])];
        milkrunrotalar[0].routedistance += 2*
m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 2])];
        milkrunrotalar[0].routedistance -= submilkrungroup[sb, 0];
    }
    else
        goto Outer;
}
else
{
    int firstcity = Convert.ToInt32(submilkrungroup[sb, 1]);
    int secondcity = Convert.ToInt32(submilkrungroup[sb, 2]);
    int routecity1 = -1; int routecity2 = -1;
    Boolean city1, city2;
    city1 = false; city2 = false;
    for (int l = 0; l < numberofmilkrun[i]; l++)
    {
        for (int k = 0; k < milkrunrotalar[l].m_routes.Count; k++)
        {
            int deneme = Convert.ToInt32(milkrunrotalar[l].m_routes[k]);
            if (firstcity == Convert.ToInt32(milkrunrotalar[l].m_routes[k]))
            {
                city1 = true;
                routecity1 = l;
            }
        }
    }
}

```

```

        submilkrungroup[j, 0] = contemporary[0, 0];
        submilkrungroup[j, 1] = contemporary[0, 1];
        submilkrungroup[j, 2] = contemporary[0, 2];
    }
}
}
#endregion
ArrayList complementofmilk = new ArrayList ();

for (int k = 0; k < numberofmilkrun[i]; k++)
    complementofmilk.Add( milkruncustomer[i, k]);
Rota[] milkrunrotalar = new Rota[numberofmilkrun[i]];
for (int r = 0; r < numberofmilkrun[i]; r++)
{
    milkrunrotalar[r] = new Rota();
    milkrunrotalar[r].m_routes = new ArrayList();
}
#region /----Tasarruf ---\
for (int sb = 0; sb < subtotalsaving; sb++)
{
    if ((RastgeleDeger.NextDouble() < 0.5) && (i < m_population-2) )
        goto Outer;
    if (submilkrungroup[sb, 0] != -1000)
    {
        if ((milkrunrotalar[0].weight == 0) )//
        {
            if (((2 * m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 1])] + 2 *
m_factoryArray[Convert.ToInt32(submilkrungroup[sb, 2])] - submilkrungroup[sb, 0]) <=
m_tourconstriction))
            {
                double weightcheck =
m_weightArray[Convert.ToInt32(submilkrungroup[sb, 1])] +
m_weightArray[Convert.ToInt32(submilkrungroup[sb, 2])];

```

```

        if (secondcity == Convert.ToInt32(milkrunrotalar[l].m_routes[k]))
        {
            city2 = true;
            routecity2 = l;

        }
    }
    int tip = -1;
    if ((city1 == true) && (city2 == true) )
    {
        if ((milkrunrotalar[routecity1].weight +
milkrunrotalar[routecity2].weight <= m_trucktype[3]) && (routecity1 !=
routecity2)&&((milkrunrotalar[routecity1].routedistance +
milkrunrotalar[routecity2].routedistance - submilkrungroup[sb,0]) <=m_tourconstriction
))

            tip = 0;
        else
        {
            goto Outer;
        }
    }
    else if ((city1 == false) && (city2 == true))
    {
        if ((milkrunrotalar[routecity2].weight + m_weightArray[firstcity] <=
m_trucktype[3]) && ((milkrunrotalar[routecity2].routedistance + 2 *
m_factoryArray[firstcity] - submilkrungroup[sb, 0]) <= m_tourconstriction))
            tip = 1;
        else
        {
            goto Outer;
        }
    }
    else if ((city1 == true) && (city2 == false) )
    {

```

```

        if ((milkrunrotalar[routecity1].weight + m_weightArray[secondcity]
<= m_trucktype[3]) && ((milkrunrotalar[routecity1].routedistance + 2 *
m_factoryArray[secondcity] - submilkrungroup[sb, 0]) <= m_tourconstriction))
            tip = 2;
        else
        {
            goto Outer;
        }
    }
    else
        if ((m_weightArray[secondcity] + m_weightArray[firstcity] <=
m_trucktype[3]) && (((2 * m_factoryArray[firstcity] + 2 * m_factoryArray[secondcity] -
submilkrungroup[sb, 0]) <= m_tourconstriction)))
            tip = 3;
        else
        {
            goto Outer;
        }

    switch (tip)
    {
        case 0:
        {
            if (milkrunrotalar[routecity1].startingcity == firstcity)
            {
                if (milkrunrotalar[routecity2].startingcity == secondcity)
                {
                    for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
                    {
milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
                        milkrunrotalar[routecity1].finishingcity = Convert.ToInt32(
milkrunrotalar[routecity2].m_routes[l]);
                    }

```

```

        milkrunrotalar[routecity1].startingcity = Convert.ToInt32(
milkrunrotalar[routecity1].m_routes[0]);
        milkrunrotalar[routecity1].weight =
milkrunrotalar[routecity1].weight + milkrunrotalar[routecity2].weight;
        routedistancedetermination(milkrunrotalar,routecity1);
        milkrunrotalar[routecity2].m_routes.Clear();
        milkrunrotalar[routecity2].startingcity = -1;
        milkrunrotalar[routecity2].finishingcity = -1;

    }
    if (milkrunrotalar[routecity2].finishingcity == secondcity)
    {
        for (int l = 0; l < milkrunrotalar[routecity1].m_routes.Count; l++)
        {

milkrunrotalar[routecity2].m_routes.Add(milkrunrotalar[routecity1].m_routes[l]);
            milkrunrotalar[routecity2].finishingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[l]);
        }
        milkrunrotalar[routecity2].startingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[0]);
        milkrunrotalar[routecity2].weight =
milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
        routedistancedetermination(milkrunrotalar, routecity2);
        milkrunrotalar[routecity1].m_routes.Clear();
        milkrunrotalar[routecity1].startingcity = -1;
        milkrunrotalar[routecity1].finishingcity = -1;
    }

}
else if (milkrunrotalar[routecity1].finishingcity == firstcity)
{
    if (milkrunrotalar[routecity2].startingcity == secondcity)
    {

```

```

        for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
        {

milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
            milkrunrotalar[routecity1].finishingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[l]);
        }
            milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
            milkrunrotalar[routecity1].weight =
milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
            routedistancedetermination(milkrunrotalar, routecity1);
            milkrunrotalar[routecity2].m_routes.Clear();
            milkrunrotalar[routecity2].startingcity = -1;
            milkrunrotalar[routecity2].finishingcity = -1;
        }
        if (milkrunrotalar[routecity2].finishingcity == secondcity)
        {
            for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
            {

milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
                milkrunrotalar[routecity1].finishingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[l]);
            }
                milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
                milkrunrotalar[routecity1].weight =
milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
                routedistancedetermination(milkrunrotalar, routecity1);
                milkrunrotalar[routecity2].m_routes.Clear();
                milkrunrotalar[routecity2].startingcity = -1;
                milkrunrotalar[routecity2].finishingcity = -1;
            }
        }
    }
}

```

```

    }
    else if (milkrunrotalar[routecity1].startingcity == secondcity)
    {
        if (milkrunrotalar[routecity2].startingcity == firstcity)
        {
            for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
            {
                milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
                milkrunrotalar[routecity1].finishingcity =
                Convert.ToInt32(milkrunrotalar[routecity2].m_routes[l]);
            }
            milkrunrotalar[routecity1].startingcity =
            Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
            milkrunrotalar[routecity1].weight =
            milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
            routedistancedetermination(milkrunrotalar, routecity1)
            milkrunrotalar[routecity2].startingcity = -1;
            milkrunrotalar[routecity2].finishingcity = -1;
        }
        if (milkrunrotalar[routecity2].finishingcity == firstcity)
        {
            for (int l = 0; l < milkrunrotalar[routecity1].m_routes.Count; l++)
            {
                milkrunrotalar[routecity2].m_routes.Add(milkrunrotalar[routecity1].m_routes[l]);
                milkrunrotalar[routecity2].finishingcity =
                Convert.ToInt32(milkrunrotalar[routecity1].m_routes[l]);
            }
            milkrunrotalar[routecity2].startingcity =
            Convert.ToInt32(milkrunrotalar[routecity2].m_routes[0]);
            milkrunrotalar[routecity2].weight =
            milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
            routedistancedetermination(milkrunrotalar, routecity2);
            milkrunrotalar[routecity1].m_routes.Clear();
            milkrunrotalar[routecity1].startingcity = -1;

```

```

        milkrunrotalar[routecity1].finishingcity = -1;
    }
}
else if (milkrunrotalar[routecity1].finishingcity == secondcity)
{
    if (milkrunrotalar[routecity2].startingcity == firstcity)
    {
        for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
        {
            milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
            milkrunrotalar[routecity1].finishingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[l]);
        }
        milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
        milkrunrotalar[routecity1].weight =
milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;
        routedistancedetermination(milkrunrotalar, routecity1);
        milkrunrotalar[routecity2].m_routes.Clear();
        milkrunrotalar[routecity2].startingcity = -1;
        milkrunrotalar[routecity2].finishingcity = -1;
    }
    if (milkrunrotalar[routecity2].finishingcity == firstcity)
    {
        for (int l = 0; l < milkrunrotalar[routecity2].m_routes.Count; l++)
        {
            milkrunrotalar[routecity1].m_routes.Add(milkrunrotalar[routecity2].m_routes[l]);
            milkrunrotalar[routecity1].finishingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[l]);
        }
        milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
        milkrunrotalar[routecity1].weight =
milkrunrotalar[routecity2].weight + milkrunrotalar[routecity1].weight;

```

```

        routedistancedetermination(milkrunrotalar, routecity1);
        milkrunrotalar[routecity2].m_routes.Clear();
        milkrunrotalar[routecity2].startingcity = -1;
        milkrunrotalar[routecity2].finishingcity = -1;
    }
}
break;
}
case 1:
{
    if (milkrunrotalar[routecity2].startingcity == secondcity)
    {
        milkrunrotalar[routecity2].m_routes.Add(firstcity);
        milkrunrotalar[routecity2].finishingcity = firstcity;
        milkrunrotalar[routecity2].startingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[0]);
        milkrunrotalar[routecity2].weight += m_weightArray[firstcity];
        routedistancedetermination(milkrunrotalar, routecity2);
        complementofmilk.Remove(firstcity);
    }
    else
    {
        milkrunrotalar[routecity2].m_routes.Add(firstcity);
        milkrunrotalar[routecity2].finishingcity = firstcity;
        milkrunrotalar[routecity2].startingcity =
Convert.ToInt32(milkrunrotalar[routecity2].m_routes[0]);
        milkrunrotalar[routecity2].weight += m_weightArray[firstcity];
        routedistancedetermination(milkrunrotalar, routecity2);
        complementofmilk.Remove(firstcity);
    }
    break;
}
case 2:

```

```

{
    if (milkrunrotalar[routecity1].startingcity == firstcity)
    {
        milkrunrotalar[routecity1].m_routes.Add(secondcity);
        milkrunrotalar[routecity1].finishingcity = secondcity;
        milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
        routedistancedetermination(milkrunrotalar, routecity1);
        complementofmilk.Remove(secondcity);
    }
    else
    {
        milkrunrotalar[routecity1].m_routes.Add(secondcity);
        milkrunrotalar[routecity1].finishingcity = secondcity;
        milkrunrotalar[routecity1].startingcity =
Convert.ToInt32(milkrunrotalar[routecity1].m_routes[0]);
        routedistancedetermination(milkrunrotalar, routecity1);
        complementofmilk.Remove(secondcity);
    }
    break;
}
case 3:
{
    truck += 1;
    milkrunrotalar[truck].m_routes.Add ( firstcity);
    milkrunrotalar[truck].m_routes.Add ( secondcity);

    milkrunrotalar[truck].finishingcity =
Convert.ToInt32(secondcity);

    routedistancedetermination(milkrunrotalar, firstcity, secondcity,
truck);

    complementofmilk.Remove(firstcity);

```

```

        complementofmilk.Remove(secondcity);
        break;
    }
    default:
    {
        break;
    }
}

    }
}
Outer:
    continue;
}
for (int a1=0; a1 < complementofmilk.Count; a1++)
{
    if (milkrunrotalar[0].weight == 0)
    {
        milkrunrotalar[a1].m_routes.Add(complementofmilk[a1]);
        milkrunrotalar[a1].weight = Convert.ToInt32(complementofmilk[a1]);
    }
    else
    {
        truck += 1;
        milkrunrotalar[truck].m_routes.Add(complementofmilk[a1]);
        milkrunrotalar[truck].weight = m_weightArray[Convert.ToInt32(
complementofmilk[a1])];
    }
}
    #endregion
for (int tour = 0; tour < numberofmilkrun[i]; tour++)
{

```

```

if (milkrunrotalar[tour].m_routes.Count >= 1)
{
    if (milkrunrotalar[tour].m_routes.Count >= 2)
    {
        determiningtrucktypeformilkrun(milkrunrotalar, tour);
        milkrunrotalar[tour].costofroute(tour, m_truckconsumption,
milkrunrotalar[tour].m_routes.Count, milkrunrotalar[tour].trucktype, m_trucktypecost);
    }
    else
    {
        milkrunrotalar[tour].routedistance =
m_factoryArray[Convert.ToInt32(milkrunrotalar[tour].m_routes[0])];
        milkrunrotalar[tour].routedistance +=
m_factoryArray[Convert.ToInt32(milkrunrotalar[tour].m_routes[(milkrunrotalar[tour].m_r
outes.Count - 1)])];
        determiningtrucktypeformilkrun(milkrunrotalar, tour);
        milkrunrotalar[tour].costofroute(tour, m_truckconsumption,
milkrunrotalar[tour].m_routes.Count, milkrunrotalar[tour].trucktype, m_trucktypecost);
    }
}
else // |
{
    milkrunrotalar[tour].trucktype = -1;
    milkrunrotalar[tour].routedistance = 0;
}
}
direkcostdetermination (i, m_genes, milkrunrotalar);
if ((q == 0) & (i == 0))
{
    besttotalcost = totalcost[0];
    bestgenrecord[0].cost = besttotalcost;
    bestgenrecord[0].numroutes = new ArrayList(numberofmilkrun[0]);
    bestgenrecord[0].trucktype = new ArrayList();
    bestgenrecord[0].weight = new ArrayList();
}

```

```

        bestgenrecord[0].gen = new ArrayList();
        bestgenrecord[0].bestiteration = 0;
        for (int k = 0; k < numberofmilkrun[0]; k++)
        {
            for (int b = 0; b < m_customer; b++)
                bestgenrecord[0].gen.Add(m_genes[i, b]);
        }
        if (besttotalcost > totalcost[i])
        {
            besttotalcost = totalcost[i];
            bestgenrecord[q].cost = besttotalcost;
            bestgenrecord[q].numroutes = new ArrayList [numberofmilkrun[i]];
            bestgenrecord[q].trucktype = new ArrayList();
            bestgenrecord[q].weight = new ArrayList();
            bestgenrecord[q].gen = new ArrayList();
            bestgenrecord[q].bestiteration = q;
            for (int k = 0; k < numberofmilkrun[i]; k++)
            {
                bestgenrecord[q].numroutes[k] = new ArrayList(numberofmilkrun[0]);
                bestgenrecord[q].numroutes[k] =
                (ArrayList)milkrunrotalar[k].m_routes.Clone();
                bestgenrecord[q].weight.Add(milkrunrotalar[k].weight);
            }
            for (int b = 0; b < m_customer; b++)
                bestgenrecord[q].gen.Add(m_genes[i, b]);
        }
    }
}

private void routedistancedetermination(Rota[] milkrunrotalar, int routecity1, int
routecity2, int truck)
{
    milkrunrotalar[truck].routedistance = m_factoryArray[routecity1];
    milkrunrotalar[truck].routedistance += m_factoryArray[routecity2];
}

```

```

        milkrunrotalar[truck].routedistance += m_customerArray[routecity1, routecity2];
    }
    private void routedistancedetermination(Rota[] milkrunrotalar1, int routecity)
    {
        milkrunrotalar1[routecity].routedistance = milkrunrotalar1[routecity].routedistance
        = m_factoryArray[Convert.ToInt32(milkrunrotalar1[routecity].m_routes[0])];
        for (int k = 0; k < milkrunrotalar1[routecity].m_routes.Count - 1; k++)
        {
            milkrunrotalar1[routecity].routedistance +=
            m_customerArray[Convert.ToInt32(milkrunrotalar1[routecity].m_routes[k]),
            Convert.ToInt32(milkrunrotalar1[routecity].m_routes[k + 1])];
        }
    }
    private void direkcostdetermination(int pop, int[,] genes, Rota[] rota)
    {
        totalcost[pop] = 0;
        int direct = 0;
        for (int k = 0; k < m_customer; k++)
        {
            if (m_genes[pop, k] == 2)
                direct += 1;
        }
        DirectRoutes[] directarray = new DirectRoutes[direct];
        for (int l = 0; l < direct; l++)
            directarray[l] = new DirectRoutes();
        ConsolidationRoutes consolidation = new ConsolidationRoutes();
        double costconsolidation=0;
        int numstopofconsolidation = 0;
        int t = 0;
        for (int k = 0; k < m_customer; k++)
        {
            if ((m_genes[pop, k] == 2) && t < direct)
            {

```

```

        directarray[t].customer = k;
        directarray[t].routedistance = m_factoryArray[k];
        directarray[t].weight = m_weightArray[k];
        directarray[t].trucktype = determiningtrucktypefordirect(directarray, t);
        directarray[t].routecost = directarray[t].routedistance *
m_truckconsumption[directarray[t].trucktype] +
m_trucktypecost[directarray[t].trucktype];
        totalcost[pop] += directarray[t].routecost;
        t++;
    }
    if (m_genes[pop, k] == 1)
    {
        numstopofconsolidation += 1;
        consolidation.weight += m_weightArray[k];
        costconsolidation += m_platformArray[k]
        costconsolidation += m_costplatArray[k];
    }
}
totalcost[pop] += costconsolidation;
consolidation.numtruck = Convert.ToInt32(Math.Ceiling ( consolidation.weight /
m_trucktype [4]));
totalcost[pop] += consolidation.routecost ;
double totmlkrncost = 0;
for (int i = 0; i < numberofmilkrun[pop]; i++)
    totmlkrncost += rota[i].routecost;
totalcost[pop] += totmlkrncost;
}

private double determinationcostconsolidation(int[,] m_genes, double[]
m_platformArray, int k, double[] m_weightArray, double[] m_trucktype, double[]
m_truckconsumption)
{
    double cost=0.0;
    if (m_weightArray[k] <= m_trucktype[0])
        cost = (m_platformArray[k]) + m_trucktypecost[0];

```

```

else if (m_weightArray[k] <= m_trucktype[1])
else if (m_weightArray[k] <= m_trucktype[0])
    cost = ( m_platformArray[k]) + m_trucktypecost[2];
else
    cost = (m_platformArray[k]) + m_trucktypecost[3];
return cost;
}

private int determiningtrucktypefordirect(DirectRoutes[] rotalar1, int tour1)
{
    int t = -1;
    if (rotalar1[tour1].weight <= m_trucktype[0])
        t = 0;
    else if (rotalar1[tour1].weight <= m_trucktype[1])
        t = 1;
    else if (rotalar1[tour1].weight <= m_trucktype[2])
        t = 2;
    else
        t = 3;
    return t;
}

private void determiningtrucktypeformilkrun(Rota[] rotalar1, int tour1)
{
    if (rotalar1[tour1].weight <= m_trucktype[0])
        rotalar1[tour1].trucktype = 0;
    else if (rotalar1[tour1].weight <= m_trucktype[1])
        rotalar1[tour1].trucktype = 1;
    else if (rotalar1[tour1].weight <= m_trucktype[2])
        rotalar1[tour1].trucktype = 2;
    else
        rotalar1[tour1].trucktype = 3;
}

```

```

    private void ilkrotabelirleme(Rota[] rotalar, ArrayList complementofmilk, double wght,
int strngcty, int fnshng, int trck, double[,] submilkrungroup, int sb, double[] m_wghtArry,
double wghtchck)
    {
        rotalar[0].m_routes.Add(Convert.ToInt32(submilkrungroup[sb, 1]));
        rotalar[0].m_routes.Add(Convert.ToInt32(submilkrungroup[sb, 2]));
        rotalar[0].weight = wghtchck;
        rotalar[0].weight = m_weightArray[Convert.ToInt32(submilkrungroup[sb, 1])] +
m_weightArray[Convert.ToInt32(submilkrungroup[sb, 2])];
        rotalar[0].startingcity = Convert.ToInt32(submilkrungroup[sb, 1]);
        rotalar[0].finishingcity = Convert.ToInt32(submilkrungroup[sb, 2]);
    }
    private void milkrunprocessing(int[,] m_genes, int[,] milkruncustomer, int[]
numberofmilkrun)
    {
        for (int i = 0; i < m_population; i++)
        {
            numberofmilkrun[i] = 0;
            for (int k = 0; k < m_customer; k++)
            {
                if (m_genes[i, k] == 3)
                    numberofmilkrun[i] += 1;
            }
            int t = 0;
            for (int k = 0; k < m_customer; k++)
            {
                if ((m_genes[i, k] == 3) && t < numberofmilkrun[i])
                {
                    milkruncustomer[i, t] = k;
                    t++;
                }
            }
        }
    }
}

```

```

    }
    private double[,] determiningsaving(double[,] m_customerArray)
    {
        double[,] save = new double[m_customer-1, m_customer];
        for (int i = 0; i < m_customer-1; i++)
        {
            for (int j = 0; j < m_customer; j++)
                save[i, j] = -1000;
        }
        for (int i = 0; i < m_customer ; i++)
        {
            for (int j = i + 1; j < m_customer; j++)
            {
                if (m_customerArray[i, j] <= m_distrestriction)
            }
        }
        return save;
    }
    private void firstiteration(int[,] m_genes)
    {
        for (int i = 0; i < m_population; i++)
        {
            for (int k = 0; k < m_customer; k++)
            }
        }
    }
}

```

B

Çözümde Kullanılan Müşteri Verileri

20 Müşteri Uzaklık Matrisi (Km)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	161	178	5	180	192	113	139	226	139	199	178	188	198	208	218	228	238	248	258
2	161	0	136	160	228	254	215	287	248	287	290	136	240	245	241	246	242	247	243	248
3	178	136	0	176	125	280	275	213	240	213	287	200	150	154	151	155	152	156	153	157
4	5	160	176	0	63	208	265	129	182	129	215	176	155	168	156	169	157	170	158	171
5	180	228	125	63	0	215	205	136	222	136	221	125	151	158	152	159	153	160	154	161
6	192	254	280	208	215	0	263	77	64	77	16	280	156	165	157	166	158	167	159	168
7	113	215	275	265	205	263	0	211	299	211	271	204	152	160	153	161	154	162	155	163
8	139	287	213	129	136	77	211	0	114	221	86	213	157	114	221	86	213	129	136	77
9	226	248	240	182	222	64	299	114	0	114	55	204	153	139	287	213	129	136	77	114
10	139	287	213	129	136	77	211	221	114	0	172	283	158	287	248	287	290	158	161	55
11	199	290	287	215	221	16	271	86	55	172	0	287	154	129	136	77	211	160	178	204
12	178	136	200	176	125	280	204	213	204	283	287	0	159	287	248	287	290	145	5	6
13	188	240	150	155	151	156	152	157	153	158	154	159	0	226	248	240	182	222	64	299
14	198	245	154	168	158	165	160	114	139	287	129	287	226	0	139	226	139	199	192	193
15	208	241	151	156	152	157	153	221	287	248	136	248	248	139	0	129	215	176	113	114
16	218	246	155	169	159	166	161	86	213	287	77	287	240	226	129	0	86	213	139	140
17	228	242	152	157	153	158	154	213	129	290	211	290	182	139	215	86	0	283	226	240
18	238	247	156	170	160	167	162	129	136	158	160	145	222	199	176	213	283	0	226	227
19	248	243	153	158	154	159	155	136	77	161	178	5	64	192	113	139	226	226	0	210
20	258	248	157	171	161	168	163	77	114	55	204	6	299	193	114	140	240	227	210	0

20 Müşteri Talebi(M³)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
12	18	20	25	35	50	40	20	5	40	4	10	5	34	28	40	21	16	10	11

20 Müşteri Fabrikaya Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
200	240	180	150	340	230	200	60	100	400	500	700	800	850	810	900	1000	245	115	105

20 Müşteri Çapraz Sevkiyata Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
900	240	350	450	444	105	200	250	200	145	125	250	550	10	200	5	65	70	123	100

20 Müşteri Plarformdan Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
36	54	60	75	105	150	120	60	15	120	12	30	15	102	84	120	63	48	30	33

30 Müşteri Uzaklık Matrisi (Km)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	210	71	81	108	200	163	70	283	75	211	160	221	249	281	110	279	170	24	168	56	26	243	110	241	33	296	140	219	183
2	210	0	345	344	396	344	371	303	380	333	306	328	346	344	315	371	376	392	395	358	360	397	363	368	337	302	307	301	398	340
3	71	345	0	145	35	78	55	100	102	13	111	99	91	131	58	33	149	24	92	94	85	147	96	14	92	113	149	50	87	25
4	81	344	145	0	220	209	170	82	57	102	79	151	150	156	169	89	196	90	237	145	229	235	183	153	104	103	249	102	199	180
5	108	396	35	220	0	249	251	107	226	312	229	299	155	107	146	244	103	214	151	100	301	235	152	107	271	312	272	313	167	130
6	200	344	78	209	249	0	225	227	248	232	209	231	224	234	241	209	225	244	209	227	246	249	215	228	246	239	223	232	238	212
7	163	371	55	170	251	225	0	143	186	95	176	353	176	221	341	407	205	62	361	105	32	426	459	421	258	361	293	166	168	279
8	70	303	100	82	107	227	143	0	119	122	65	38	40	58	65	96	58	83	74	31	30	54	112	64	91	85	46	37	37	104
9	283	380	102	57	226	248	186	119	0	262	256	111	233	125	195	292	302	330	127	216	282	125	310	295	203	137	162	277	139	198
10	75	333	13	102	312	232	95	122	262	0	145	100	102	132	108	135	115	134	128	148	139	146	148	114	139	102	131	145	101	131
11	211	306	111	79	229	209	176	65	256	145	0	127	77	126	95	95	71	57	26	68	21	124	35	99	72	34	31	66	120	93
12	160	328	99	151	299	231	353	38	111	100	127	0	132	105	45	293	150	289	248	229	214	43	239	252	77	234	270	161	143	264
13	221	346	91	150	155	224	176	40	233	102	77	132	0	41	99	77	36	105	60	159	172	198	75	199	30	56	11	22	12	79
14	249	344	131	156	107	234	221	58	125	132	126	105	41	0	273	435	366	503	204	308	434	229	594	206	557	550	320	417	378	450
15	281	315	58	169	146	241	341	65	195	108	95	45	99	273	0	186	134	155	131	191	117	101	162	191	198	135	183	118	177	102
16	110	371	33	89	244	209	407	96	292	135	95	293	77	435	186	0	263	272	150	235	296	258	105	283	186	138	230	271	282	206
17	279	376	149	196	103	225	205	58	302	115	71	150	36	366	134	263	0	118	58	148	131	156	93	169	183	168	92	175	159	80
18	170	392	24	90	214	244	62	83	330	134	57	289	105	503	155	272	118	0	263	382	181	201	269	224	402	287	408	329	446	287
19	24	395	92	237	151	209	361	74	127	128	26	248	60	204	131	150	58	263	0	274	178	278	416	259	236	379	443	363	156	148
20	168	358	94	145	100	227	105	31	216	148	68	229	159	308	191	235	148	382	274	0	215	250	204	220	207	236	222	241	233	238
21	56	360	85	229	301	246	32	30	282	139	21	214	172	434	117	296	131	181	178	215	0	343	312	219	344	367	23	92	132	61
22	26	397	147	235	235	249	426	54	125	146	124	43	198	229	101	258	156	201	278	250	343	0	180	146	150	159	140	155	157	173
23	243	363	96	183	152	215	459	112	310	148	35	239	75	594	162	105	93	269	416	204	312	180	0	70	18	174	61	194	128	54
24	110	368	14	153	107	228	421	64	295	114	99	252	199	206	191	283	169	224	259	220	219	146	70	0	359	451	487	446	337	336
25	241	337	92	104	271	246	258	91	203	139	72	77	30	557	198	186	183	402	236	207	344	150	18	359	0	169	225	211	165	119
26	33	302	113	103	312	239	361	85	137	102	34	234	56	550	135	138	168	287	379	236	367	159	174	451	169	0	26	77	37	83
27	296	307	149	249	272	223	293	46	162	131	31	270	11	320	183	230	92	408	443	222	23	140	61	487	225	26	0	80	31	74
28	140	301	50	102	313	232	166	37	277	145	66	161	22	417	118	271	175	329	363	241	92	155	194	446	211	77	80	0	161	102
29	219	398	87	199	167	238	168	37	139	101	120	143	12	378	177	282	159	446	156	233	132	157	128	337	165	37	31	161	0	66
30	183	340	25	180	130	212	279	104	198	131	93	264	79	450	102	206	80	287	148	238	61	173	54	336	119	83	74	102	66	0

30 Müşteri Talebi (M³)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
10	8	9	15	12	18	20	25	35	50	40	20	5	40	4	10	5	34	28	40	21	16	10	11	18	17	17	17	16	16

30 Müşteri Fabrikaya Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
904	60	398	372	38	249	207	189	193	371	489	91	492	630	709	643	243	241	209	237	708	522	438	523	448	29	798	205	389	602

30 Müşteri Çapraz Sevkiyata Uzaklık(Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
578	12	934	953	570	401	915	805	655	198	682	550	935	124	181	411	634	65	815	596	872	949	924	33	952	565	951	119	794	668

30 Müşteri Plarformdan Taşıma Maliyeti(TL)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
30	24	27	45	36	54	60	75	105	150	120	60	15	120	12	30	15	102	84	120	63	48	30	33	54	52	51	50	49	47

50 Müşteri Talebi (M³)

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

50 Müşteri Fabrikaya Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
120	150	220	250	775	730	500	600	95	780	100	710	180	225	105	110	880	235	280	205	175	100	230	201	368	354	195	188	502	400	740	369	270	145	96	360	325	600	580	470	210	230	155	160	126	322	245	215	350	85

50 Müşteri Çapraz Sevkiyata Uzaklık(Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
900	700	750	600	540	425	200	250	130	145	125	250	780	930	1000	1125	890	930	900	630	670	885	670	800	960	100	130	900	940	700	560	640	760	863	420	140	456	500	605	704	230	400	350	963	178	288	380	462	489	250

50 Müşteri Plarformdan Taşıma Maliyeti (TL)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
60	90	90	120	24	45	120	60	44	120	60	30	21	66	105	59	15	69	78	63	27	57	75	105	129	150	78	90	96	48	102	72	78	135	30	90	30	96	42	36	132	39	75	111	84	48	63	51	69	45

50 Müşteri Uzaklık Matrisi (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
1	0	1	0	3164	1496	176	373	357	264	3	85	2656	300	33	683	2338	462	1002	3205	3935	2656	438	5547	2569	1534	4448	2943	143	1776	286	417	136	496	3225	972	2943	324	2569	2330	5053	154	4345	1002	1389	1418	116	424	270	477	301
2	3164	0	1895	1898	643	2674	816	1482	15	23	2656	9863	2171	188	3332	124	2067	153	1812	2179	2119	8891	15	797	1812	335	1482	2862	1940	3334	1585	1888	2684	1838	1496	144	3474	797	2319	2172	2255	1931	1842	3532	2866	1612	2635	1973	381	4025
3	1895	1898	0	909	2336	1423	153	1401	1753	336	1667	9448	153	1401	293	1118	05	153	1889	29	1933	97	919	49	1611	1174	25	2684	943	496	956	163	168	959	1852	1118	1088	3803	05	293	1697	103	153	2886	1281	1786	366			
4	1763	3895	909	0	2338	123	143	2095	141	2672	3333	1795	355	1611	95	268	1401	293	1803	1825	893	153	1688	29	1933	97	919	49	1611	1174	25	2684	943	496	956	163	168	959	1852	1118	1088	3803	05	293	1697	103	153	2886	1281	1786
5	3793	643	2536	2538	0	3341	153	2149	1454	2726	843	153	1739	4959	633	2734	1654	2656	2846	2786	2558	153	1482	2865	627	695	2949	2516	4919	3812	2565	2884	1265	1496	695	441	1432	633	2786	2939	2386	2558	2885	499	3533	2336	3282	294	1495	1176
6	3793	643	2536	2538	0	3341	153	2149	1454	2726	843	153	1739	4959	633	2734	1654	2656	2846	2786	2558	153	1482	2865	627	695	2949	2516	4919	3812	2565	2884	1265	1496	695	441	1432	633	2786	2939	2386	2558	2885	499	3533	2336	3282	294	1495	1176
7	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
8	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
9	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
10	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
11	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
12	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
13	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
14	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
15	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
16	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
17	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
18	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
19	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
20	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
21	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
22	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
23	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
24	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
25	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
26	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
27	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003	213	2119	1867	1441	387	
28	6287	2674	1423	213	3341	0	1884	170	97	3032	2025	1381	2168	2486	77	1743	97	2062	514	525	1422	1571	1949	2682	7728	381	2102	2868	3357	213	3884	213	1423	2778	2432	9449	2738	255	516	2943	1422	2682	2486	3003						

100 Müşteri Uzaklık Matrisi (Km)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100																																																																										
1	0	10	15	12	18	8	22	25	30	20	28	35	10	32	27	17	24	19	33	29	14	31	26	34	11	36	21	37	16	38	23	39	13	40	24	41	15	42	25	43	16	44	26	45	17	46	27	47	18	48	28	49	19	50	29	51	30	52	31	53	32	54	33	55	34	56	35	57	36	58	37	59	38	60	39	61	40	62	41	63	42	64	43	65	44	66	45	67	46	68	47	69	48	70	49	71	50	72	51	73	52	74	53	75	54	76	55	77	56	78	57	79	58	80	59	81	60	82	61	83	62	84	63	85	64	86	65	87	66	88	67	89	68	90	69	91	70	92	71	93	72	94	73	95	74	96	75	97	76	98	77	99	78	100																				
2	10	0	20	17	25	15	35	28	32	12	38	22	40	14	42	19	44	24	46	29	17	48	31	50	13	52	26	54	16	56	30	58	18	60	32	62	20	64	25	66	19	68	33	70	21	72	34	74	22	76	35	78	23	80	36	82	24	84	37	86	25	88	38	90	26	92	39	94	27	96	40	98	28	100	41	102	29	104	30	106	31	108	32	110	33	112	34	114	35	116	36	118	37	120	38	122	39	124	40	126	41	128	42	130	43	132	44	134	45	136	46	138	47	140	48	142	49	144	50	146	51	148	52	150	53	152	54	154	55	156	56	158	57	160	58	162	59	164	60	166	61	168	62	170	63	172	64	174	65	176	66	178	67	180	68	182	69	184	70	186	71	188	72	190	73	192	74	194	75	196	76	198	77	200
3	15	20	0	25	18	30	22	35	25	17	40	14	45	15	48	20	50	18	52	22	19	55	16	58	20	60	17	62	21	64	19	66	23	68	21	70	24	72	22	74	26	76	24	78	27	80	25	82	28	84	26	86	29	88	27	90	30	92	28	94	31	96	29	98	30	100	31	102	32	104	33	106	34	108	35	110	36	112	37	114	38	116	39	118	40	120	41	122	42	124	43	126	44	128	45	130	46	132	47	134	48	136	49	138	50	140	51	142	52	144	53	146	54	148	55	150	56	152	57	154	58	156	59	158	60	160	61	162	62	164	63	166	64	168	65	170	66	172	67	174	68	176	69	178	70	180	71	182	72	184	73	186	74	188	75	190	76	192	77	194	78	196	79	198	80	200								
4	12	17	25	0	30	22	38	30	28	20	42	15	48	16	50	20	52	22	54	24	19	56	17	58	21	60	18	62	22	64	20	66	24	68	22	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200								
5	8	25	18	30	0	35	25	38	30	22	45	16	50	20	52	22	54	24	56	26	19	58	21	60	18	62	22	64	20	66	24	68	22	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200										
6	22	35	25	38	35	0	40	32	45	28	50	17	52	24	56	26	58	28	60	30	19	62	22	64	20	66	24	68	22	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200														
7	30	32	38	40	45	40	0	48	40	32	52	18	54	26	58	30	60	32	62	32	19	64	22	66	24	68	22	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200																
8	25	45	40	48	48	48	48	0	50	40	52	19	56	28	60	32	62	32	64	34	19	66	24	68	22	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200																		
9	32	48	48	50	50	50	50	50	0	52	20	58	29	62	32	64	34	66	36	68	38	19	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178	71	180	72	182	73	184	74	186	75	188	76	190	77	192	78	194	79	196	80	198	81	200																					
10	28	38	40	52	52	52	52	52	52	0	54	20	56	28	62	32	64	34	66	36	68	38	70	25	72	23	74	27	76	25	78	28	80	26	82	29	84	27	86	30	88	28	90	31	92	29	94	32	96	30	98	31	100	32	102	33	104	34	106	35	108	36	110	37	112	38	114	39	116	40	118	41	120	42	122	43	124	44	126	45	128	46	130	47	132	48	134	49	136	50	138	51	140	52	142	53	144	54	146	55	148	56	150	57	152	58	154	59	156	60	158	61	160	62	162	63	164	64	166	65	168	66	170	67	172	68	174	69	176	70	178</																																											

100 Müşteri Talebi (M^3)[illegible]

100 Müşteri Fabrikaya Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

100 Müşteri Çapraz Sevkiyata Uzaklık (Km)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

100 Müşteri Plarformdan Taşıma Maliyeti (TL)

[illegible]

Tezden Üretilmiş Yayınlar

İletişim Bilgisi: kocaogluyasemin84@yahoo.com

Makale

1. Y. Kocaoğlu, A. Gümüş Taşkın ve B. Kocaoğlu, "Supply Chain Optimization Studies: A Literature Review and Classification", Doğu Üniversitesi Dergisi, vol. 19, no. 1, pp. 79 – 98, 2018.