



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**DEĞİŞKEN KOMŞULUK ARAMA SEZGİSEL YAKLAŞIMI
VE TEDARİK ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA**

**Hacer YUMURTACI AYDOĞMUŞ
Endüstri Mühendisliği Anabilim Dalı
Endüstri Mühendisliği Programı**

**Danışman
Doç.Dr. Ş.Alp BARAY**

Aralık, 2011

İSTANBUL



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**DEĞİŞKEN KOMŞULUK ARAMA SEZGİSEL YAKLAŞIMI
VE TEDARİK ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA**

Hacer YUMURTACI AYDOĞMUŞ
Endüstri Mühendisliği Anabilim Dalı
Endüstri Mühendisliği Programı

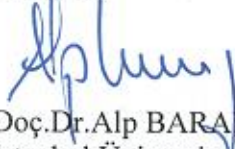
Danışman
Doç.Dr. Ş.Alp BARAY

Aralık, 2011

İSTANBUL

Bu çalışma 28/ 12/ 2011 tarihinde aşağıdaki jüri tarafından Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği programında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi



Doç.Dr. Alp BARAY
İstanbul Üniversitesi
Mühendislik Fakültesi



Prof.Dr.Şakir ESNAF
İstanbul Üniversitesi
Mühendislik Fakültesi



Prof.Dr. Güneş GENÇYILMAZ
İstanbul Aydın Üniversitesi
Mühendislik-Mimarlık Fakültesi



Prof. Dr. Selim ZAIM
Marmara Üniversitesi
Teknoloji Fakültesi



Doç.Dr. Mehpere TİMOR
İstanbul Üniversitesi
İşletme Fakültesi

ÖNSÖZ

Öncelikle tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Ş. Alp BARAY'a en içten dileklerle teşekkür ederim. Akademisyenlik hayatımda kendisinden çok şey öğrendim. Bölüm başkanımız değerli hocam Prof. Dr. Şakir ESNAF'a akademisyenlikteki yol göstericiliği ile çalışmalarımda verdiği destek için çok teşekkür ederim. Tez izleme komitemde bulunan Prof. Dr. Güneş GENÇYILMAZ'a da yardımlarından dolayı teşekkürlerimi sunarım. Ayrıca akademisyenliğe başladığım zamandan itibaren desteğini hiçbir zaman esirgemeyen Prof. Dr. Ekrem MANİSALI hocama teşekkürlerimi sunarım.

Bu çalışma boyunca yardımlarını esirgemeyen çalışma arkadaşlarımdan hepsine desteklerinden dolayı teşekkür ederim.

Bana en büyük desteği veren sevgili aileme, her birine, anneme, babama, ablama, kardeşime ve sevgili eşim Ufuk AYDOĞMUŞ'a yoğun çalışma sürecinde gösterdikleri maddi ve manevi destek için müteşekkirim.

Aralık, 2011

Hacer YUMURTACI AYDOĞMUŞ

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ.....	vii
TABLO LİSTESİ	xii
KISALTMALAR.....	xv
ÖZET.....	xvi
SUMMARY	xvii
1. GİRİŞ	1
2. GENEL KISIMLAR	3
2.1. OPTİMİZASYON MODELLERİ.....	3
2.2. YEREL ARAMA.....	7
2.2.1. Yerel Aramanın Tanımı	7
2.2.2. Yerel Arama Algoritması.....	8
2.2.3. Yerel Aramayı İyileştirme Stratejileri	9
2.2.3.1. Açgözlü Rassallaştırılmış Uyarlamalı Arama Yordamı	10
2.2.3.2. Düzgünleştirme Yöntemi.....	10
2.2.3.3. Tabu Arama Metasezgiseli	10
2.2.3.4. Tavlama Benzetimi Metasezgiseli	11
2.2.4. Geleneksel Yerel Arama Sezgiselleri	12
2.2.4.1. Şehir-takas sezgiseli (City-swap).....	12
2.2.4.2. 2-opt Sezgiseli.....	13
2.2.4.3. Or-Opt Sezgiseli.....	14
2.3. KOMŞULUK YAPISI	14
2.3.1. Komşuluk Yapısının Tanımı.....	15
2.3.2. Başlıca Komşuluk Yapıları	16
2.3.2.1. Değişirme (exchange)/ Takas-iç (swap-intra).....	16
2.3.2.2. Girinti (insert).....	16
2.3.2.3. 1-1 Değişimi (1-1 interchange) / Takas Prosedürü (swap procedure).....	16

2.3.2.4. 2-1 Değişimi (interchange(2-1)) / 2-1 Takas.....	17
2.3.2.5. Kaydırma (Shift).....	18
2.3.2.6. 2-0 Kaydırma (shift(2-0)).....	18
2.3.2.7. 3-0 Kaydırma (shift(3-0)).....	18
2.3.2.8. 2-2 Kaydırma (shift(2-2)).....	19
2.3.2.9. k-Or-Opt.....	19
2.3.2.10. 2009 Yılında Sunulan Yeni Bir Düzensizlik Mekanizması.....	19
2.3.2.11. 2-opt Operatörü.....	20
2.3.2.12. Tersine Operatör.....	20
2.4. TEDARİK ZİNCİRİ YÖNETİMİ.....	21
2.4.1. Tedarik Zinciri Yönetiminin Tanımı.....	21
2.4.2. Tedarik Zinciri Yönetimi Kavramının Ortaya Çıkışı.....	23
2.4.3. Tedarik Zinciri Ağının Aşamaları.....	25
2.4.3.1. Tedarikçi Aşaması.....	25
2.4.3.2. Fabrika Aşaması.....	26
2.4.3.3. Dağıtım Aşaması.....	26
2.4.4. Tedarik Zinciri Yönetiminde Karşılaşılan Problemler.....	27
2.5. ARAÇ ROTALAMA PROBLEMİ.....	28
2.5.1. Graf Yapılar.....	29
2.5.1.1. Hamilton Grafi.....	31
2.5.1.2. Euler Grafi.....	31
2.5.2. Gezgin Satıcı Problemi.....	31
2.5.2.1. Gezgin Satıcı Probleminin Doğrusal Gösterimi.....	32
2.5.2.2. Gezgin Satıcı Probleminin Tamsayılı Gösterimi.....	32
2.5.3. Araç Rotalama Problemi (ARP)'nin Önemi.....	33
2.5.4. Araç Rotalama Probleminin Tarihsel Gelişimi.....	34
2.5.5. Araç Rotalama Problemlerinin Çeşitleri.....	35
2.5.5.1. Kapasiteli Araç Rotalama Problemi.....	36
2.5.5.2. Zaman Pencereleli Araç Rotalama Problemi.....	38
2.5.5.3. Farklı Tipte (Heterojen) Araçlı Araç Rotalama Problemi.....	38
2.5.5.4. Açık Uçlu Araç Rotalama Problemi.....	38
2.5.5.5. Periyodik Araç Rotalama Problemi.....	39
2.5.5.6. Topla-Dağıt Araç Rotalama Problemi.....	39
2.5.6. Araç Rotalamanın Temel Problemleri ve Aralarındaki İlişki.....	39
2.5.7. Araç Rotalama Problemleri İçin Çözüm Yöntemleri.....	40
2.5.8. Kesin Çözüm Yöntemleri.....	40
2.5.8.1. Tam Sayılı Doğrusal Programlama.....	40
2.5.8.2. Dal-Sınır Yöntemi.....	41
2.5.8.3. Kesme Düzlemi Algoritması.....	41

2.5.8.4. Minimum K-ağaç.....	41
2.5.8.5. Dinamik Programlama.....	42
2.5.9. Sezgisel Yöntemler	42
2.5.9.1. En Kısa Yol Problemi	43
2.5.9.2. Tasarruf Algoritması - Clarke ve Wright Algoritması	43
2.5.9.3. Fisher ve Jaikumar (FJ) Algoritması	43
2.5.9.4. Süpürme Algoritması.....	43
2.5.10. Metasezgisel Yöntemler.....	45
2.5.10.1. Tavlama Benzetimi.....	45
2.5.10.2. Tabu Arama	45
2.5.10.3. Yapay Sinir Ağları.....	45
2.5.10.4. Karınca Kolonisi Optimizasyonu.....	47
2.5.10.5. Genetik Algoritma	48
2.6. ÇOK AMAÇLI PROGRAMLAMA	49
2.6.1. Çok Amaçlı Karar Verme Teknikleri	52
2.6.1.1. Doğrudan yöntemler	53
2.6.1.2. İki aşamalı yöntemler	53
2.6.1.3. Sezgisel yöntemler.....	54
2.6.2. Amaç Uzayı.....	55
2.6.3. Fayda Fonksiyonu Yaklaşımı.....	57
2.6.4. Hedef Programlama	59
2.6.5. Baskınlık	59
2.6.6. Baskın Olmayan Kriter Vektörü.....	60
2.6.7. Etkililik	60
2.6.8. Pareto Optimal.....	63
2.7. ÇOK AMAÇLI ARAÇ ROTALAMA PROBLEMLERİ.....	63
2.7.1. Araç Rotalama Problemlerinin Çok Amaçlı Kullanımı.....	64
2.7.1.1. Klasik Akademik Problemlerin Uzantısı Olarak Çok Amaçlı Kullanımı	64
2.7.1.2. Klasik Problemleri Genelleştirmek İçin Çok Amaçlı Kullanımı.....	65
2.7.1.3. Gerçek Hayat ile İlgili Vakalar İçin Çok Amaçlı Kullanımı	66
2.7.2. Çok Amaçlı Araç Rotalama Problemlerinde Kullanılan Amaçlar	67
2.7.2.1. Tur ile İlgili Amaçlar.....	67
2.7.2.2. Düğüm/Bağlantı Faaliyetleri ile İlgili Amaçlar	67
2.7.2.3. Kaynaklar ile İlgili Amaçlar.....	67
2.7.3. Literatürdeki Çok Amaçlı Araç Rotalama Uygulamalarından Örnekler	67
3. MALZEME VE YÖNTEM	69
3.1. DEĞİŞKEN KOMŞULUK ARAMA YAKLAŞIMI	69
3.1.1. Değişken Komşuluk Arama Algoritması.....	71

3.2. DEĞİŞKEN KOMŞULUK ARAMA ÇEŞİTLERİ.....	72
3.2.1. Değişken Komşuluk Azalma (DKAzalma).....	72
3.2.2. İndirgenmiş Değişken Komşuluk Arama (İDKA)	73
3.2.3. Genel Değişken Komşuluk Arama (GDKA)	74
3.2.4. Değişken Komşuluk Ayırıştırma Arama (DKAA).....	74
3.2.5. Eğik Komşuluk Ayırıştırma Arama (EDKA)	76
3.2.6. Paralel Komşuluk Ayırıştırma Arama (PDKA)	76
3.3. ARAÇ ROTALAMA VE DEĞİŞKEN KOMŞULUK ARAMA	77
3.3.1. Teorik Çalışmalardan.....	77
3.4. ÇOK AMAÇLI DEĞİŞKEN KOMŞULUK ARAMA.....	83
3.4.1. Çok Amaçlı Değişken Komşuluk Arama Algoritması	84
3.4.2. Çizelgeleme Problemlerinde Çok Amaçlı Değişken Komşuluk Arama Uygulamaları	85
3.4.3. Çok Amaçlı Çizelgeleme Problemlerinde Hibrid Değişken Komşuluk Arama Uygulamaları.....	86
3.4.4. Atama Problemlerinde Çok Amaçlı Değişken Komşuluk Arama Uygulamaları	89
3.5. GELİŞTİRİLEN YÖNTEM.....	91
3.6. BAŞLANGIÇ ÇÖZÜMÜ AŞAMASI.....	96
3.6.1. Süpürme Algoritması İle Başlangıç Çözümünün Elde Edilmesi	96
3.6.2. İleri İtmeli Ekleme Sezgisel Yöntemi ile Başlangıç Çözümünün Elde Edilmesi	110
3.7. KOMŞULUK YAPILARI	118
3.7.1. Rota İçi Değişim 1-1.....	118
3.7.2. Rota İçi Değişim 2-1.....	119
3.7.3. Rotalar Arası Değişim 1-1	119
3.7.4. Rotalar Arası Değişim 2-1	120
3.7.5. Rota İçi Kaydırma 2-0	120
3.7.6. Rotalar Arası Kaydırma 1-0.....	121
3.7.7. Rotalar Arası Kaydırma 2-0.....	121
3.7.8. Komşuluk Yapılarının Uygulama Sırasının Belirlenmesi	122

4. BULGULAR..... 123

4.1. PROGRAMIN ÇALIŞTIRILMASI	123
4.1.1. Başlangıç Çözümü Oluşturmak İçin Süpürme Algoritmasının Kullanıldığı Durum	123
4.1.1.1. Geliştirilen Yaklaşımın C101 Problem Kümesi İçin Uygulanması.....	124
4.1.1.2. Geliştirilen Yaklaşımın C201 Problem Kümesi İçin Uygulanması.....	129
4.1.1.3. Geliştirilen Yaklaşımın R101 Problem Kümesi İçin Uygulanması.....	133
4.1.1.4. Geliştirilen Yaklaşımın R201 Problem Kümesi İçin Uygulanması.....	137
4.1.1.5. Geliştirilen Yaklaşımın RC101 Problem Kümesi İçin Uygulanması	141
4.1.1.6. Geliştirilen Yaklaşımın RC201 Problem Kümesi İçin Uygulanması	145
4.1.2. Başlangıç Çözümü Oluşturmak İçin İleri İtmeli Ekleme Sezgisel Yönteminin Kullanıldığı Durum.....	153

4.1.3. Geliştirilen Yaklaşımın C101 Problem Kümesi İçin İncelenmesi	164
5. TARTIŞMA VE SONUÇ.....	167
KAYNAKLAR.....	177
EKLER.....	187
ÖZGEÇMİŞ.....	205

ŞEKİL LİSTESİ

Şekil 2.1	: Klasik optimizasyon modelleri.	4
Şekil 2.2	: Optimizasyon metodları.....	5
Şekil 2.3	: Metasezgisel yöntemlerin soyağacı	6
Şekil 2.4	: Yerel arama çözümünün ikili sunumunu kullanarak, en iyi komşuluğu seçme stratejisi.	7
Şekil 2.5	: Klasik yerel arama algoritması.	8
Şekil 2.6	: Global ve yerel optimum gösterimi	8
Şekil 2.7	: Yerel aramayı geliştirmek ve yerel optimumdan kaçınmaya yönelik tek-nokta aramalı metasezgisel ailesi.	9
Şekil 2.8	: Temel tabu arama algoritması	11
Şekil 2.9	: Temel tavlama benzetimi algoritması	12
Şekil 2.10	: Gezgin satıcı problemi için şehir-takas hareketi.	13
Şekil 2.11	: r-opt adımları.....	13
Şekil 2.12	: Gezgin satıcı problemi için 2-opt hareketi.	14
Şekil 2.13	: 3 elemanlı bir permütasyon probleminin komşulukları.	15
Şekil 2.14	: İki komşuluk kullanan değişken komşuluk arama	15
Şekil 2.15	: Takas-iç komşuluk yapısı için örnek hareket.....	16
Şekil 2.16	: Rota 2'den (1) ve rota 3'ten (10) numaralı müşteriler için takas prosedürü.	17
Şekil 2.17	: Takas (2,1) komşuluk yapısı için örnek hareket.	17
Şekil 2.18	: Rota 2'den rota 3'e (6) numaralı müşterinin kaydırılması prosedürü.....	18
Şekil 2.19	: 2-2 kaydırma komşuluk yapısı için örnek hareket.	19
Şekil 2.20	: k-Or-Opt komşuluk yapısı için örnek hareket.....	19
Şekil 2.21	: Gezgin satıcı problemi için 2-opt operatörü.	20
Şekil 2.22	: Inversion (tersine) operatörü.	20
Şekil 2.23	: Rekabet avantajı ve 3C.	21
Şekil 2.24	: Lojistik ve Rekabet Avantajı.....	22
Şekil 2.25	: Bir tedarik zinciri ağının genel yapısı	25
Şekil 2.26	: Tedarik zinciri bilgi fonksiyonu.....	27
Şekil 2.27	: Basit bir graf.....	30
Şekil 2.28	: 4 düğümlü yönlü bir graf.	30
Şekil 2.29	: Euler Grafi.....	31
Şekil 2.30	: Euler ve Hamilton grafları.	31
Şekil 2.31	: Araç rotalaması genel görünümü	33
Şekil 2.32	: Genel zaman pencereli araç rotaları.....	38
Şekil 2.33	: Araç rotalamasının temel problemleri aralarındaki ilişki	39
Şekil 2.34	: Basit bir yapay sinir ağı diyagramı.	47
Şekil 2.35	: Çok kriterli karar verme	50
Şekil 2.36	: $SE(P) = S \cup S'$ grafiği.....	54
Şekil 2.37	: Uygun çözüm alanı.	56
Şekil 2.38	: Uygun kriter alanı.	56
Şekil 2.39	: Karar uzayı	58

Şekil 2.40	: Kriter uzayı	58
Şekil 2.41	: Etkili çözüm için örnek şekil	61
Şekil 2.42	: Çok amaçlı metotta iki amaç	61
Şekil 2.43	: İdeal çözüm	62
Şekil 2.44	: Kuvvetlendirme için iyi çözüm	62
Şekil 2.45	: Çeşitlendirme için iyi çözüm	62
Şekil 3.1	: Temel değişken komşuluk arama algoritması.....	71
Şekil 3.2	: Değişken Komşuluk Azalma algoritması	72
Şekil 3.3	: Değişken komşuluk azalmanın prensibi.	73
Şekil 3.4	: İndirgenmiş değişken komşuluk arama algoritması.....	73
Şekil 3.5	: Genel değişken komşuluk arama algoritması.	74
Şekil 3.6	: Değişken komşuluk ayrıştırma arama algoritması.....	75
Şekil 3.7	: Eğik komşuluk ayrıştırma arama algoritması.	76
Şekil 3.8	: Taşıma operatörü.....	79
Şekil 3.9	: Çapraz değiştirme operatörü	79
Şekil 3.10	: ÇADKA algoritması	84
Şekil 3.11	: GDKA algoritması için geliştirilen programın örnek ekran görüntüsü ..	94
Şekil 3.12	: ÇAGDKA algoritması için işlem adımları.....	95
Şekil 3.13	: Pareto kümesinin güncellenmesi için işlem adımları.....	96
Şekil 3.14	: Süpürme algoritmasının adımları.....	97
Şekil 3.15	: Süpürme algoritması ile 25 müşterilik C101 problem kümesi için başlangıç rotası grafiği (3 rota, 264 br mesafe).....	98
Şekil 3.16	: 50 müşterilik C101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (5 rota, 525 br mesafe).....	98
Şekil 3.17	: 100 müşterilik C101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (10 rota, 1193 br mesafe).....	99
Şekil 3.18	: 25 müşterilik C201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 276 br mesafe).....	100
Şekil 3.19	: 50 müşterilik C201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 564 br mesafe).....	100
Şekil 3.20	: 100 müşterilik C201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (3 rota, 1035 br mesafe).....	101
Şekil 3.21	: 25 müşterilik R101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 407 br mesafe).....	102
Şekil 3.22	: 50 müşterilik R101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (4 rota, 815 br mesafe).....	102
Şekil 3.23	: 100 müşterilik R101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (8 rota, 1466 br mesafe).....	103
Şekil 3.24	: 25 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 361 br mesafe).....	104
Şekil 3.25	: 50 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 687 br mesafe).....	104
Şekil 3.26	: 100 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 1280 br mesafe).....	105
Şekil 3.27	: 25 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (3 rota, 463 br mesafe).....	106
Şekil 3.28	: 50 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (5 rota, 833 br mesafe).....	106

Şekil 3.29	: 100 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (9 rota, 1681 br mesafe).....	107
Şekil 3.30	: 25 müşterilik RC201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 342 br mesafe).....	108
Şekil 3.31	: 50 müşterilik RC201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 517 br mesafe).....	108
Şekil 3.32	: 100 müşterilik RC201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 1355 br mesafe).....	109
Şekil 3.33	: Rota içi değişim 1-1 komşuluk yapısının örnek uygulaması.	118
Şekil 3.34	: Rota içi değişim 2-1 komşuluk yapısının örnek uygulaması.	119
Şekil 3.35	: Rotalar arası değişim 1-1 komşuluk yapısının örnek uygulaması.	119
Şekil 3.36	: Rotalar arası değişim 2-1 komşuluk yapısının örnek uygulaması.	120
Şekil 3.37	: Rota içi kaydırma 2-0 komşuluk yapısının örnek uygulaması.....	120
Şekil 3.38	: Rotalar arası kaydırma 1-0 komşuluk yapısının örnek uygulaması.	121
Şekil 3.39	: Rotalar arası kaydırma 2-0 komşuluk yapısının örnek uygulaması.	121
Şekil 4.1	: 25 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	126
Şekil 4.2	: 50 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	126
Şekil 4.3	: 25 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	128
Şekil 4.4	: 100 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	128
Şekil 4.5	: 25 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	130
Şekil 4.6	: 50 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	130
Şekil 4.7	: 50 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	132
Şekil 4.8	: 100 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	132
Şekil 4.9	: 50 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	134
Şekil 4.10	: 100 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	134
Şekil 4.11	: 25 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	136
Şekil 4.12	: 50 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	136
Şekil 4.13	: 25 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	138
Şekil 4.14	: 100 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	138
Şekil 4.15	: 25 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	140
Şekil 4.16	: 50 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)	140
Şekil 4.17	: 25 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	142

Şekil 4.18	: 50 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	142
Şekil 4.19	: 25 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	144
Şekil 4.20	: 50 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	144
Şekil 4.21	: 25 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	146
Şekil 4.22	: 50 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli).....	146
Şekil 4.23	: 50 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	148
Şekil 4.24	: 100 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli).....	148
Şekil 4.25	: 25 müşterilik C101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözüm grafiği	159
Şekil 4.26	: 25 müşterilik C101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik.....	159
Şekil 4.27	: 100 müşterilik C101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözüm grafiği	160
Şekil 4.28	: 100 müşterilik C101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik.....	160
Şekil 4.29	: 25 müşterilik R101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözüm grafiği	161
Şekil 4.30	: 25 müşterilik R101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik.....	161
Şekil 4.31	: 50 müşterilik RC101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözüm grafiği	162
Şekil 4.32	: 50 müşterilik RC101 problem kümesi için elde edilen alternatiflerden 5 rotalı için en iyi değerli grafik.....	162
Şekil 4.33	: 100 müşterilik RC201 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözüm grafiği	163
Şekil 4.34	: 100 müşterilik RC201 problem kümesi için elde edilen alternatiflerden 5 rotalı için en iyi değerli grafik.....	163
Şekil 4.35	: 25 müşterilik C101 problem kümesi için elde edilen en iyi sonucun grafiği.	165
Şekil B.1	: Çok amaçlı genel değişken komşuluk arama genel kodu.	196
Şekil B.2	: Talep hesaplama fonksiyonunun kodu.....	197
Şekil B.3	: Toplam mesafe hesaplama fonksiyonunun kodu.	197
Şekil B.4	: Uzaklık hesaplama fonksiyonunun kodu.	197
Şekil B.5	: Pareto çözümün karşılaştırma fonksiyonunun kodu.	198
Şekil B.6	: Rota içi değişim 1-1 silkeleme komşuluk uygulaması fonksiyonunun kodu.....	198
Şekil B.7	: Rotalar arası değişim 1-1 ve değişim 2-1(silkeleme) komşuluk uygulaması fonksiyonunun kodu.	199
Şekil B.8	: Rotalar arası kaydırma 2-0 (silkeleme) komşuluk uygulaması fonksiyonunun kodu.....	200
Şekil B.9	: Rota içi değişim 1-1 ve değişim 2-1(yerel arama) komşuluk uygulaması fonksiyonunun kodu.....	201

Şekil B.10	: Rotalar arası deęişim 1-1 (yerel arama) komşuluk uygulaması fonksiyonunun kodu.....	202
Şekil B.11	: Rota ii kaydırma 1-0 ve deęişim 2-0 (yerel arama) komşuluk uygulaması fonksiyonunun kodu.	203
Şekil B.12	: Rotalar arası kaydırma 1-0 ve kaydırma 2-0 (yerel arama) komşuluk uygulaması fonksiyonunun kodu.	204

TABLO LİSTESİ

Tablo 2.1	: Araç rotalama problemlerinin sınıflandırması	36
Tablo 2.2	: GSP için tur sayılarının hesaplanması	40
Tablo 2.3	: Sezgisel yöntemlerin kıyaslanması tablosu.....	44
Tablo 2.4	: YSA hesaplama yaklaşımlarının kıyaslama tablosu	46
Tablo 2.5	: Metasezgisel yöntemlerin kıyaslama tablosu.....	48
Tablo 2.6	: Baskınlık ifadesi için örnek	60
Tablo 2.7	: Etkili çözüm için örnek	60
Tablo 3.1	: Solomon test verilerinin özet tablosu.....	93
Tablo 3.2	: Süpürme algoritması ile 25 müşterili C101 problem kümesi rota tablo ..	98
Tablo 3.3	: Süpürme algoritması ile 50 müşterili C101 problem kümesi rota tablo ...	99
Tablo 3.4	: Süpürme algoritması ile 100 müşterili C101 problem kümesi rota tablo .	99
Tablo 3.5	: Süpürme algoritması ile 25 müşterili C201 problem kümesi rota tablo .	100
Tablo 3.6	: Süpürme algoritması ile 50 müşterili C201 problem kümesi rota tablo .	101
Tablo 3.7	: Süpürme algoritması ile 100 müşterili C201 problem kümesi rota tablo	101
Tablo 3.8	: Süpürme algoritması ile 25 müşterili R101 problem kümesi rota tablo .	102
Tablo 3.9	: Süpürme algoritması ile 50 müşterili R101 problem kümesi rota tablo ...	103
Tablo 3.10	: Süpürme algoritması ile 100 müşterili R101 problem kümesi rota tablo	103
Tablo 3.11	: Süpürme algoritması ile 25 müşterili R201 problem kümesi rota tablo .	104
Tablo 3.12	: Süpürme algoritması ile 50 müşterilik R201 problem kümesi rota tablo	105
Tablo 3.13	: Süpürme algoritması ile 100 müşterili R201 problem kümesi rota tablo	105
Tablo 3.14	: Süpürme algoritması ile 25 müşterilik RC101 problem kümesi rota tablosu	106
Tablo 3.15	: Süpürme algoritması ile 50 müşterilik RC101 problem kümesi rota tablosu	107
Tablo 3.16	: Süpürme algoritması ile 100 müşterilik RC101 problem kümesi rota tablosu	107
Tablo 3.17	: Süpürme algoritması ile 25 müşterilik RC201 problem kümesi rota tablosu	108
Tablo 3.18	: Süpürme algoritması ile 50 müşterilik RC201 problem kümesi rota tablosu	109
Tablo 3.19	: Süpürme algoritması ile 100 müşterilik RC201 problem kümesi rota tablosu	109
Tablo 3.20	: İİES algoritması ile 25 müşterilik C101 problem kümesi rota tablosu (17 rota)	110
Tablo 3.21	: İİES algoritması ile 50 müşterilik C101 problem kümesi rota tablosu (34 rota)	111
Tablo 3.22	: İİES algoritması ile 100 müşterilik C101 problem kümesi rota tablosu (63 rota)	111
Tablo 3.23	: İİES algoritması ile 25 müşterilik C201 problem kümesi rota tablosu (15 rota)	112

Tablo 3.24	: İİES algoritması ile 50 müşterilik C201 problem kümesi rota tablosu (22 rota)	112
Tablo 3.25	: İİES algoritması ile 100 müşterilik C201 problem kümesi rota tablosu (30 rota)	113
Tablo 3.26	: İİES algoritması ile 25 müşterilik R101 problem kümesi rota tablosu (15 rota)	113
Tablo 3.27	: İİES algoritması ile 50 müşterilik R101 problem kümesi rota tablosu (25 rota)	114
Tablo 3.28	: İİES algoritması ile 100 müşterilik R101 problem kümesi rota tablosu (53 rota)	114
Tablo 3.29	: İİES algoritması ile 25 müşterilik R201 problem kümesi rota tablosu (6 rota)	115
Tablo 3.30	: İİES algoritması ile 50 müşterilik R201 problem kümesi rota tablosu (8 rota)	115
Tablo 3.31	: İİES algoritması ile 100 müşterilik R201 problem kümesi rota tablosu (11 rota)	115
Tablo 3.32	: İİES algoritması ile 25 müşterilik RC101 problem kümesi rota tablosu (8 rota)	115
Tablo 3.33	: İİES algoritması ile 50 müşterilik RC101 problem kümesi rota tablosu (19 rota)	116
Tablo 3.34	: İİES algoritması ile 100 müşterilik RC101 problem kümesi rota tablosu (39 rota)	116
Tablo 3.35	: İİES algoritması ile 25 müşterilik RC201 problem kümesi rota tablosu (7 rota)	117
Tablo 3.36	: İİES algoritması ile 50 müşterilik RC201 problem kümesi rota tablosu (7 rota)	117
Tablo 3.37	: İİES algoritması ile 100 müşterilik RC201 problem kümesi rota tablosu (12 rota)	117
Tablo 4.1	: 25, 50 ve 100 müşterilik C101 problem grupları için sonuç tablosu (yüksek dış döngü değerli)	125
Tablo 4.2	: C101 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)	127
Tablo 4.3	: C201 problem kümesi için sonuç tablosu (yüksek dış döngü değerli) ...	129
Tablo 4.4	: C201 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)	131
Tablo 4.5	: R101 problem kümesi için sonuç tablosu (yüksek dış döngü değerli) ...	133
Tablo 4.6	: R101 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)	135
Tablo 4.7	: R201 problem kümesi için sonuç tablosu (yüksek dış döngü değerli) ...	137
Tablo 4.8	: R201 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)	139
Tablo 4.9	: RC101 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)	141
Tablo 4.10	: RC101 problem kümesi için sonuç tablosu (yüksek iç döngü değerli) ...	143
Tablo 4.11	: RC201 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)	145
Tablo 4.12	: RC201 problem kümesi için sonuç tablosu (yüksek iç döngü değerli) ...	147
Tablo 4.13	: ÇAGDKA yaklaşımında dış döngü sayısı yüksek (200) tutularak elde edilen sonuçlar	149
Tablo 4.14	: ÇAGDKA yaklaşımında iç döngü sayısı yüksek (20) tutularak elde edilen sonuçlar	150
Tablo 4.15	: ÇAGDKA yaklaşımının uygulanmasında yüksek iç döngü veya dış döngünün tercih edilmesinde elde edilen sonuçların kıyaslama tablosu	151
Tablo 4.16	: ÇAGDKA yaklaşımında yüksek iç ve yüksek dış döngü kullanımı ile bulunan en iyi değerlerin işlem sürelerinin (sn) kıyaslama tablosu	152

Tablo 4.17	: Dış döngü ile iç döngü sayılarının farklılığına göre sonuçtaki başarı durumu	153
Tablo 4.18	: ÇAGDKA uygulaması için programın bir kere çalıştırılması sonucu pareto çözüm kümesi.....	154
Tablo 4.19	: C101 problem verisinin 100 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu	154
Tablo 4.20	: C101 problem verisinin 50 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu	155
Tablo 4.21	: C101 problem verisinin 25 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu	155
Tablo 4.22	: R201 problem verisinin 100 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu	156
Tablo 4.23	: ÇAGDKA yaklaşımı ile elde edilen sonuçların tablosu (İİES başlangıç çözüm yöntemi ve yüksek iç döngü değeri ile).....	157
Tablo 4.24	: ÇAGDKA yaklaşımının C101/25 problem kümesine s=100, l=80 değerleri ile uygulanmasının sonuçları.....	164
Tablo 4.25	: ÇAGDKA yaklaşımının C101/25 problem kümesine uygulanmasında s ve l değerlerinin değiştirilmesiyle değişen çözüme ulaşma süreleri tablosu	165
Tablo 5.1	: ÇAGDKA yaklaşımı ile ÇAGA yaklaşımının verdiği en iyi değerlerin kıyaslama tablosu(yüksek dışdöngü)	169
Tablo 5.2	: ÇAGDKA yaklaşımı ile ÇAGA yaklaşımının verdiği en iyi değerlerin kıyaslama tablosu(yüksek iç döngü)	170
Tablo 5.3	: ÇAGDKA yaklaşımı ile elde edilne sonuçların tablosu (İİES başlangıç çözüm yöntemi ve yüksek iç döngü değeri ile).....	171
Tablo A.1	: Solomon C-101 Problem Kümesine ait 100 noktanın koordinat ve talep verisi.....	189
Tablo A.2	: Solomon C-201 Problem Kümesine ait 100 noktanın koordinat ve talep verisi	190
Tablo A.3	: Solomon R-101 ve R201 Problem Kümelerine ait 100 noktanın koordinat ve talep verisi	191
Tablo A.4	: Solomon RC-101 ve RC201 Problem Kümelerine ait 100 noktanın koordinat ve talep verisi	193

KISALTMALAR

A*, IDA*	: Tekrarlı derinleşen algoritmalar (iterative deepening algorithms)
ARP	: Araç rotalama problemi
ÇARP	: Çok amaçlı araç rotalama problemi
ÇADKA	: Çok amaçlı değişken komşuluk arama
ÇAGA	: Çok amaçlı genetik algoritma
ÇAGDKA	: Çok amaçlı genel değişken komşuluk arama
ÇAKV	: Çok amaçlı karar verme
ÇAKO	: Çok amaçlı kombinatoriyal optimizasyon
ÇKKV	: Çok kriterli karar verme
ÇÖKV	: Çok ölçütlü karar verme
DKA	: Değişken komşuluk arama
DKAA	: Değişken komşuluk ayrıştırma araması (variable neighborhood decomposition search)
DKAz	: Değişken komşuluk azalma (variable neighborhood descent)
EDKA	: Eğik değişken komşuluk arama (skewed variable neighborhood search)
FJ	: Fisher ve Jaikumar
GA	: Genetik algoritma
GDKA	: Genel değişken komşuluk arama (general variable neighborhood search)
GSP	: Gezgin satıcı problemi
GRASP	: Açgözlü rassallaştırılmış uyarlamalı arama yordamı (greedy adaptive search procedure)
İDKA	: İndirgenmiş değişken komşuluk arama (reduced variable neighborhood search)
İİES	: İleri itmeli ekleme sezgisel (push forward insertion heuristic)
KARP	: Klasik araç rotalama problemi
KKO	: Karınca kolonisi optimizasyonu (Ant colony optimization)
PDKA	: Paralel değişken komşuluk arama (parallel variable neighborhood search)
SA	: Süpürme algoritması
SZO	: Sürü zekası optimizasyonu (Particle swarm optimization)
sn	: Saniye
TA	: Tabu arama
TB	: Tavlama benzetimi
TDKA	: Temel değişken komşuluk arama (basic variable neighborhood search)

ÖZET

DEĞİŞKEN KOMŞULUK ARAMA SEZGİSEL YAKLAŞIMI VE TEDARİK ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA

Bu çalışmada tedarik zinciri yönetiminde karşılaşılan problemlerden biri olan çok amaçlı araç rotalama problemine uygulamak üzere, değişken komşuluk arama yaklaşımı temelli bir yöntem geliştirmek amaçlanmıştır. Değişken komşuluk arama yaklaşımı kombinatoryal problemlerin çözümü için geliştirilmiş olan son metasezgisel yöntemlerden biridir. Bu yaklaşım ilk olarak N. Mladenović ve E. Hansen tarafından 1997 yılında geliştirilmiştir.

Genel kısımlarda ilk olarak sezgisel ve metasezgisel yöntemler hakkında bilgi verilmiştir. Değişken komşuluk arama yaklaşımı yerel arama tabanlı bir yaklaşım olduğu için bu bölümde “yerel arama” konusu ele alınmıştır. Ayrıca, sistematik olarak komşuluk değişimi fikrine dayanan değişken komşuluk aramada kullanılan hareket mekanizmalarını ifade eden “komşuluk yapısı” kavramı ve çeşitleri yer almaktadır.

Genel kısımlarda tedarik zinciri yönetiminin tanımı, önemi ve zincirde karşılaşılan problemlerden bahsedildikten sonra, söz konusu problemler içinde yer alan araç rotalamanın önemi incelenmiş, araç rotalama problemlerinin özellikleri, çeşitleri ve kullanılan çözüm yöntemleri ele alınmıştır. Çözüm yöntemleri klasik, sezgisel ve meta-sezgisel yöntemler olmak üzere ayrı başlıklar altında anlatılmıştır. Ardından çok amaçlı programlama konusu da ele alınarak, çok amaçlı problemlerin çözümünde kullanılan yaklaşımlar farklı özelliklerine göre gruplandırılmış ve bilgi verilmiştir. Farklı meta-sezgisel yöntemler kullanılarak çözülen çok amaçlı araç rotalama problemlerinden örnekler incelenmiştir.

Çalışmanın malzeme ve yöntem bölümünde, değişken komşuluk arama yaklaşımı hakkında detaylı bilgiler verilmiş, bu yaklaşımın çeşitleri ve algoritma adımları anlatılmıştır. Bu yaklaşımın uygulamalarından bahsedilmiş, ardından da özellikle araç rotalama problemleri üzerinde gerçekleştirilen uygulamaları hakkında bilgi verilmiştir.

Çok amaçlı araç rotalama probleminde değişken komşuluk arama yaklaşımının ilk defa genel değişken komşuluk arama çeşidinin uygulanmasıyla elde edilen sonuçlar, genetik algoritma metasezgiselinin daha önce uygulanmasıyla elde edilen literatürdeki sonuçlarla karşılaştırılmıştır. Sonuçlar grafik ve tablo şeklinde tezde sunulmuştur ve sonuç kısmında, yapılan çalışmanın katkıları ile ilgili bir değerlendirme bulunmaktadır.

SUMMARY

VARIABLE NEIGHBOURHOOD SEARCH HEURISTIC METHOD AND AN APPLICATION IN A SUPPLY CHAIN MANAGEMENT

The purpose of this study is to present a new approach based on variable neighborhood search for solving multi objective vehicle routing problem one of the problems encountered in supply chain management. Variable neighbourhood search (VNS) is one of the recent metaheuristics which has widely used to solve combinatorial optimization problems. This approach is proposed by Mladenovic and Hansen in 1997.

In first part of thesis, heuristics and meta-heuristics are discussed generally then local search is discussed since VNS is a local search based approach. The basic idea of VNS is systematically change of neighbourhoods. For this reason “neighborhood structure” concept and types are also detailed.

After presenting definition and importance of supply chain management and problems encountered in supply chain management, vehicle routing problem as an important subject in supply chain management is described and its variations, characteristics are considered too. Solving methods for vehicle routing problem are described under separate headings including classical solution methods, heuristic and meta-heuristic methods. Then, by considering the issue of multi objective programming, multi objective problem solving methods are grouped and detailed according to different features. Some examples from literature for multi objective vehicle routing problem which are solved by metaheuristics are presented.

The third part of this study contains basic rules and variations of VNS. The applications of VNS from the literature, especially for vehicle routing problem, are presented.

The fourth part of the study contains analysis of the multi objective general variable neighborhood search method (MOGVNS), which is prepared firstly in this study. The proposed method is compared with genetic algorithm metaheuristic. The results are given with diagrams and tables. In the conclusion, contributions of the developed method are explained.

1. GİRİŞ

Tedarik zinciri yönetimi felsefesi son yıllarda firmaların rekabet üstünlüğü sağlamak, varlıklarını devam ettirebilmek gibi sebeplerle uyguladıkları bir yaklaşım olmuştur. Malzemenin satın alınması, bu malzemenin ara ürün ve nihai ürüne dönüştürülmesi ve bu nihai ürünün müşterilere dağıtım işlevlerini gerçekleştiren araçlar ve dağıtım seçenekleri ağı olarak tanımlanan tedarik zinciri yönetimi içerisinde dağıtım aşaması en önemli aşamalardan birisi olmuştur (Ganeshan ve Harrison, 1995). Bunun bir sebebi geçmiş yıllarda firmalar üzerinde yapılan maliyet analizlerinde taşıma maliyetlerinin büyük paylara sahip olduğunun farkedilmiş olmasıdır. Taşıma faaliyeti firmaların üzerinde çalışarak maliyetini düşürebilecekleri, asıl ürünlerine gerçek değer katmayan faaliyetlerdendir. Dağıtım ağı denilince, dağıtım yapılacak müşterilere veya tedarikçilerden fabrikaya malzeme taşınması gibi taşıma işlemleri akla gelmektedir. Dağıtım ağına araçların rotalarının belirlenmesi için problem araç rotalama problemi olarak ele alınmaktadır. Araç rotalama problemi 1954 yılından beri incelenen bir optimizasyon problemi olmuştur ve çalışmalar artan bir şekilde devam etmektedir. İlk yıllarda araç rotalama probleminin statik çeşitleri üzerinde çalışmalar gerçekleştirilmiştir. Araç rotalama probleminin bu yıllarda olasılıklı, dinamik, çok amaçlı ve bulanık çeşitlerinin pek fazla çalışılmamasının sebebini Eksioğlu ve diğ. (2009), hesaplamadaki karmaşıklık ile mikrobilgisayarların yetersizliği olarak belirtmişlerdir. Araç rotalama problemleri genellikle maliyetin veya uzaklığın minimizasyonu gibi tek bir amaç doğrultusunda ele alınmıştır. Fakat, karşılaşılan çoğu endüstri probleminde birden fazla ve birbiriyle çelişen amaçlar bulunmaktadır. Bu yüzden çok amaçlı rotalama problemleri ilgi duyulan bir araştırma konusu olmuştur. 1990'lı yıllarda araç rotalama problemi araştırmaları, teknolojinin gelişmesi ve mikrobilgisayarlara daha kolay ulaşılabilmesiyle hızlanmış ve araştırmacılar daha karışık arama algoritmaları geliştirebilmişlerdir. Yine bu dönemde araç rotalama problemi gibi kombinatoriyel problemlerin çözümüne yönelik olarak metasezgisel terimi ortaya çıkmıştır. Osman ve Laporte (1996) tarafından metasezgiseller, çözüm uzayını

araştırmak için, farklı kavramları akıllıca bir araya getirerek sezgisel yöntemleri yönlendirmeye çabalayan tekrarlı oluşturma süreçleri olarak tanımlanmıştır.

Değişken komşuluk arama metasezgisel yaklaşımı ilk olarak Mladenovic ve Hansen (1997) tarafından geliştirilmiştir. Bu yaklaşımda sistematik olarak komşuluk değişimleri fikri kullanılmaktadır. Bu değişimin sağlanmasında mevcut bir çözüme “komşuluk yapısı” olarak adlandırılan hareket mekanizmaları uygulanarak çözümün farklı komşuluk serileri araştırılmaktadır. Değişken komşuluk arama yaklaşımı, yerel arama tabanlı bir metasezgisel yöntemdir ve diğer metasezgisel yöntemlere kıyasla daha basit bir yaklaşım olup, çok iyi çözümler sağlamaktadır.

Birbiriyle çelişen birden fazla amaca sahip problemlerde, tüm amaç fonksiyonlarını en iyi şekilde sağlayan bir çözümün bulunması oldukça zordur. Bu çalışmada tedarik zinciri yönetiminde çok amaçlı araç rotalama problemine uygulamak üzere bir yöntem geliştirmek amaçlanmıştır. Tek amaçlı araç rotalama problemlerinde çok sayıda başarılı uygulaması bulunan, basit ve iyi sonuçlar veren yapıya sahip değişken komşuluk arama yaklaşımı ile çeşitleri incelenerek geliştirilen yöntem, bu tezde çok amaçlı araç rotalama problemleri üzerinde uygulanmıştır. Genel değişken komşuluk arama yaklaşımı temel alınarak çok amaçlı araç rotalama problemi için çok amaçlı genel değişken komşuluk arama yaklaşımı (ÇAGDKA) geliştirilmiştir. DKA yaklaşımında başlangıç çözüm yönteminin seçimi ile yaklaşımın içinde kullanılan komşuluk yapılarının belirlenmesi büyük önem taşımaktadır. Geliştirilen ÇAGDKA yaklaşımında farklı başlangıç çözümleri kullanılarak yaklaşımın başarısı gösterilmiştir. Yaklaşım geliştirilirken problem için uygun komşuluk yapıları ve bu yapıların en uygun sıralamaları belirlenmiştir. Geliştirilen yaklaşım literatürde daha önce genetik algoritmanın uygulanmasıyla elde edilmiş olan sonuçlarla kıyaslanmış ve belirlenen amaçlar için daha iyi sonuç elde edilmiştir.

2. GENEL KISIMLAR

2.1. OPTİMİZASYON MODELLERİ

Bir problemin sonlu sayıda aşamada çözümlenip çözülemeyeceği probleme yaklaşımda önemli bir konu olmuştur. Nabiyev (2010), evrensel algoritmik modellerin 3 türünü aşağıdaki gibi sıralamıştır:

- i. Birinci türde, klasik hesaplama ve sayısal fonksiyonlar gibi matematiksel kavramlarla algoritma kavramı ilişkilendirilmektedir.
- ii. İkinci türde, her ayırık zamanda çok basit işlemleri yapan bir deterministik makinayla algoritma bağdaştırılmaktadır.
- iii. Üçüncü tür, herhangi bir alfabede sözcüklerin değiştirilmesine dayalı kelime işlemcilerdir.

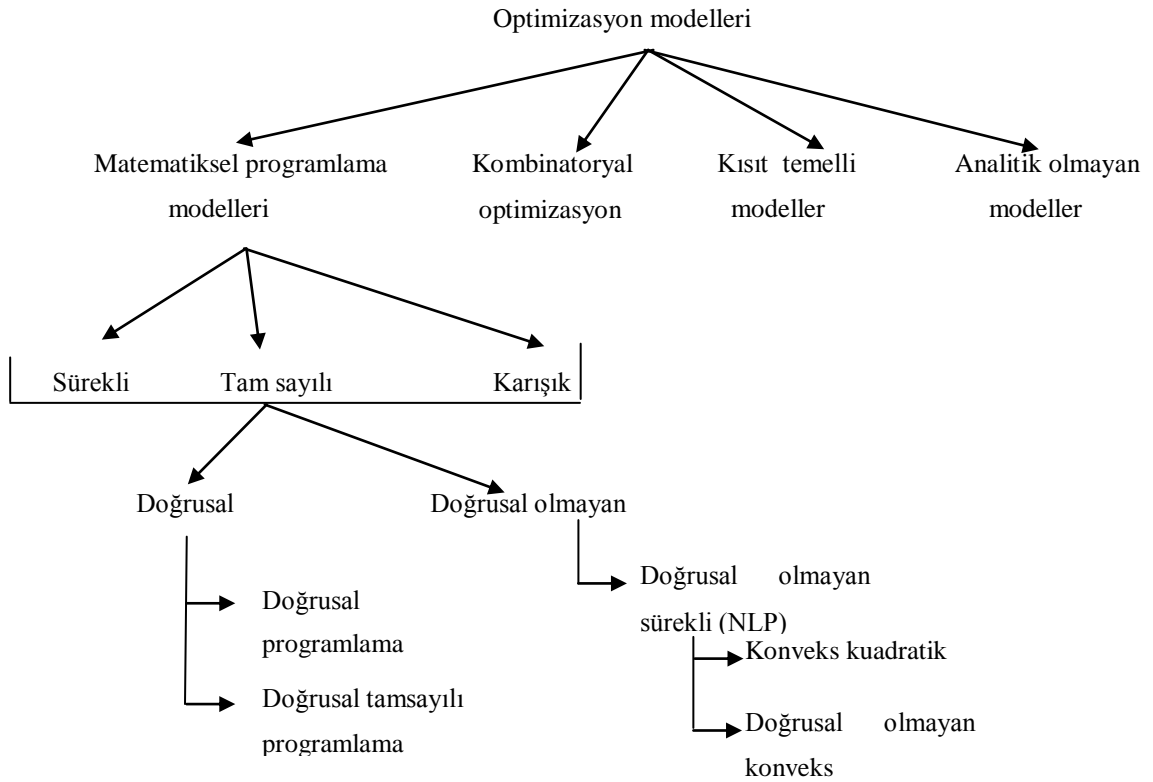
Bir problemin algoritmik çözümlerinin sınıflandırılmasında, yürütülmeleri için gerekli işlemlerin sayısı temel alınmaktadır. Bu ölçü “algoritmik karmaşıklık” olarak adlandırılmaktadır (Nabiyev, 2010).

Bir algoritma bir problemi çözebilmek için “zaman” ve “alan” olmak üzere iki önemli kaynağa ihtiyaç duymaktadır (Shor,1996). Talbi (2009) de, bir algoritmanın zaman karmaşıklığını; ‘ n ’ boyutlu bir problemi çözebilmek için ihtiyaç duyulan adım sayısı şeklinde tanımlamıştır. Karmaşıklık genellikle “en kötü durum analizi” terimleri ile ifade edilmektedir.

Optimizasyon problemleri genelde karmaşık (kompleks) yapıdadır veya geniş (sonsuz) çözüm alanına sahiptir. Lineer programlama ve Lagrange Çarpanı gibi popüler matematik teknikler kesin çözüm metotlarıdır. Kesin çözüm metotları (klasik yöntemler) NP-zor (non-deterministic polynomial time) problemlerin çözümünde zorluk taşımaktadır (O’Brien, 2008).

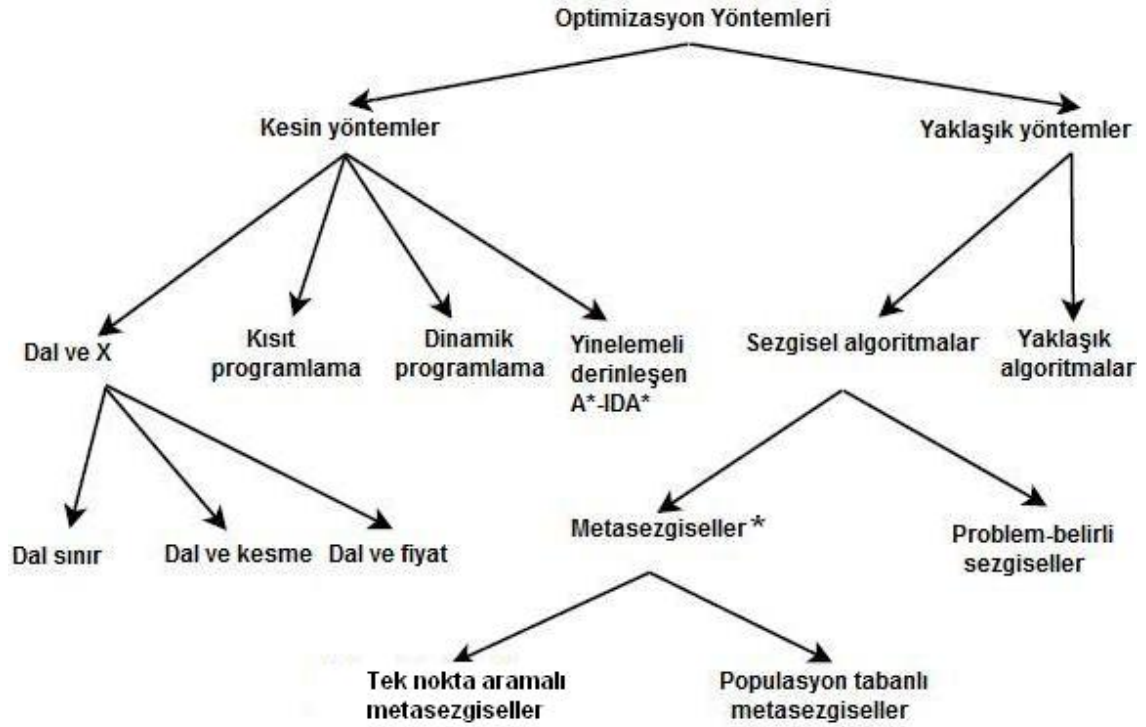
NP-zor problemler; klasik yöntemlerle çözülmesi çok zor olan veya en iyi sonucun çok uzun sürede elde edilebildiği problemlerdir. Sezgisel yöntemler son yıllarda bu problemlerin çözümünde geniş bir uygulama alanına sahip olmuştur. Eğer bir problemin çözülmesi için polinom zaman algoritması mevcut ise o problem “kolay” olarak adlandırılır. Aksi durumda “zor” olarak adlandırılmaktadır. Sezgisel yöntemler ile kestirim (approximation) algoritmaları arasındaki fark: genellikle makul bir süre içinde oldukça “iyi” çözümler bulan sezgisellerin aksine kestirim algoritmaları ispatlanabilir çözüm kalitesi ve ispatlanabilir çalışma süresi sınırları sağlamaktadır. Sezgiseller, büyük boyutlu problemlerde “iyi” çözümler bulmaktadır. Sezgiseller iki başlık altında ele alınmaktadır; belirli sezgiseller ve metasezgiseller. Belirli sezgiseller belirli problem ve/veya olay için uygulanmaktadır. Metasezgiseller ise, hemen hemen her optimizasyon problemini çözmek için uygulanabilecek genel amaçlı algoritmalarıdır (Talbi, 2009).

Talbi'nin (2009) optimizasyon modellerine ve bu modellerin çözümü için sunulan çözüm yöntemlerine dair sınıflandırması sırasıyla Şekil 2.1 ve 2.2'de gösterilmiştir.



Şekil 2.1: Klasik optimizasyon modelleri.

Matematiksel programlama modelleri, değişkenlerin sürekli, tamsayı veya karışık olmasına göre üçe ayrılmaktadır. Ayrıca kurulan model, doğrusal veya doğrusal olmayan denklemlerle ifade edilmektedir. Perez ve diğ. (2006) kombinatoriyal optimizasyon modellerini, probleme ait çözüm uzayının ayırık veya kısmen ayırık olması şeklinde ifade etmiştir.



Şekil 2.2: Optimizasyon metodları (Talbi, 2009)

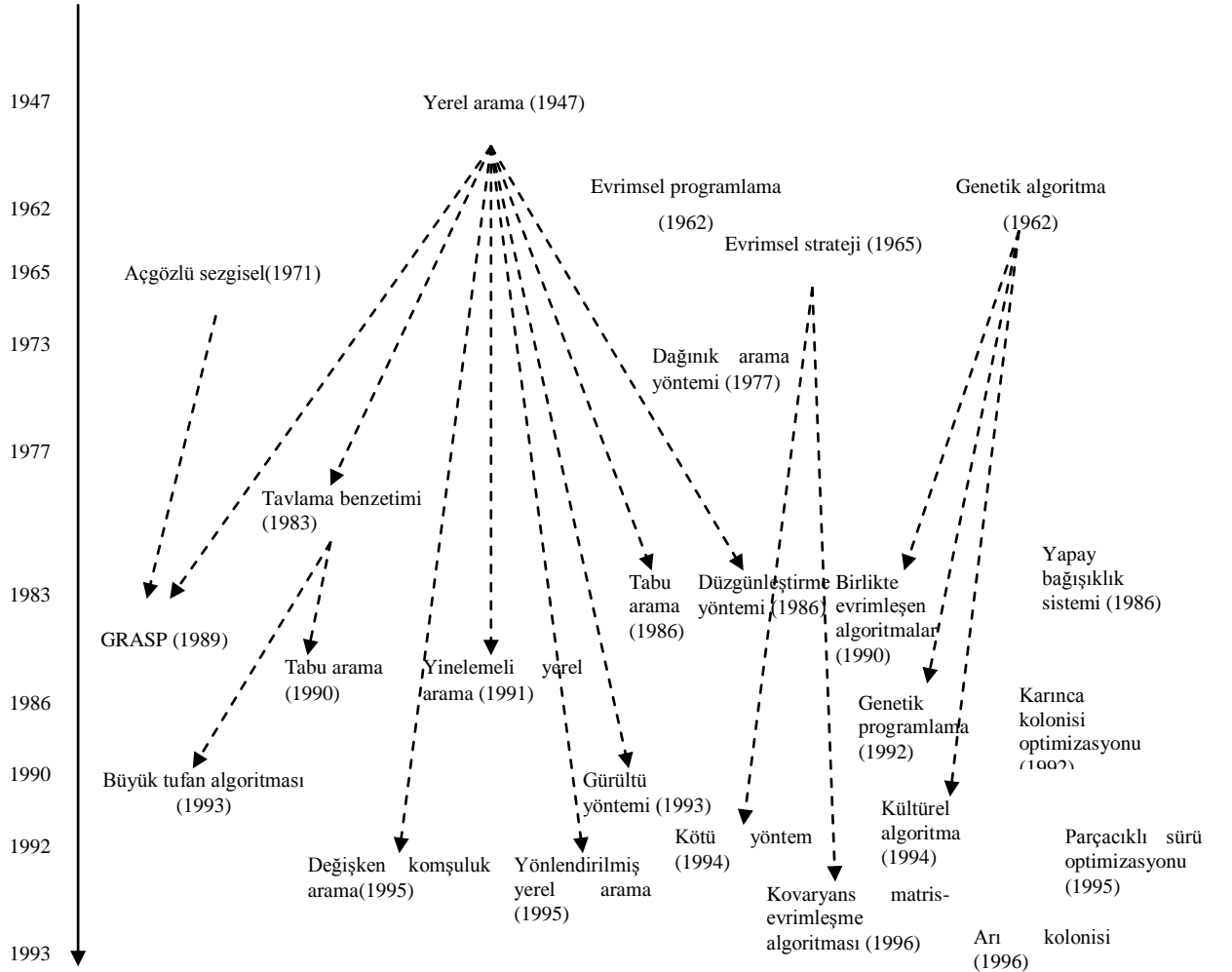
Optimizasyon yöntemleri kesin ve yaklaşık yöntemler olmak üzere ikiye ayrılmaktadır. Bu yöntemler ileriki bölümlerde detaylı olarak ele alınmıştır. Metasezgisel kelimesinde bulunan sezgisel ifadesinin İngilizcedeki karşılığı “heuristic”dir. “Heuristic”, bulma, keşfetme anlamına gelen “heuriskein” kelimesinden türetilmiş ve “Meta” kelimesi de üst seviye anlamına gelmektedir. Temel sezgisel yöntemleri birleştirmeye çabalayan yeni yaklaşık yöntemler “metasezgiseller (metaheuristic)” olarak adlandırılmaktadır (Ayyüce, erişim tarihi 12 Temmuz 2011).

Metasezgisel yöntemler deterministik olmayan yöntemlerdir. Optimum çözümü garanti etmemekle birlikte, denenmesi gereken çok sayıda olasılığın söz konusu olduğu durumlar için daha az deneme ile kabul edilebilir seviyede iyi çözüm bulmaktadırlar (Gözüdeli ve Akcayol, 2007).

Şekil 2.2’de görüldüğü gibi metasezgisel yöntemler tek nokta aramalı (single solution based metaheuristics) ve populasyon tabanlı metasezgiseller olarak ikiye ayrılmaktadır. Değişken komşuluk arama yaklaşımı tek nokta aramalı metasezgisel grubunda bulunan bir yaklaşımdır. Metasezgisel algoritmaların bazıları şu şekilde listelenmiştir (Dreo ve diğ., 2006; Gonzalez, 2007; Talbi, 2009);

- Tavlama Benzetimi
- Tabu Arama
- Karınca Kolonisi Optimizasyonu
- Değişken Komşuluk Arama
- Genetik Algoritmalar
- Kuş Sürüsü Optimizasyonu

Şekil 2.3’te metasezgisel yöntemler ilk kullanıldıkları yıllar itibari ile gösterilmektedir.



Şekil 2.3: Metasezgisel yöntemlerin soyağacı (Talbi, 2009)

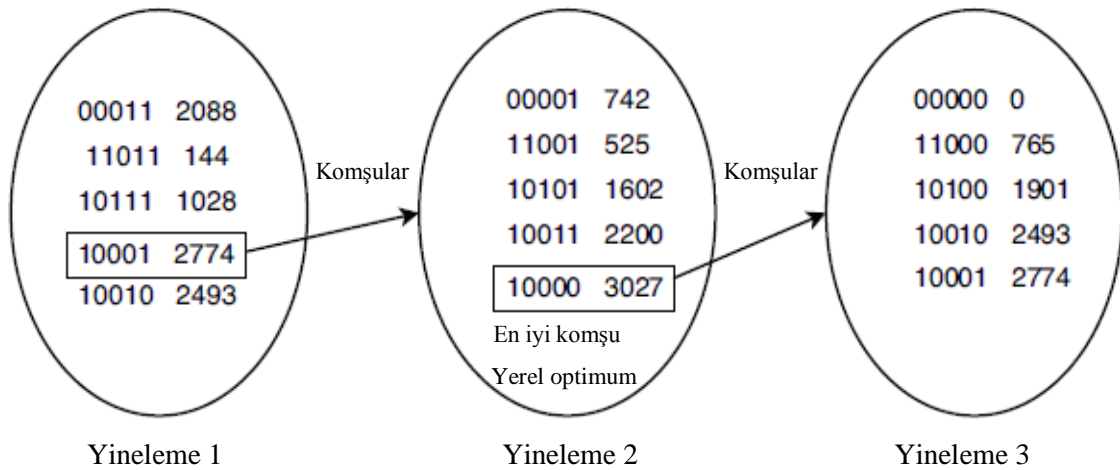
Detaylı olarak metasezgisel yöntemlerin bazıları yerel arama ve araç rotalama problemi konuları incelenirken ele alınmıştır.

2.2. YEREL ARAMA

Tez çalışmasında uygulama metodu olarak seçilen değişken komşuluk arama yaklaşımı, yerel arama tabanlı bir yaklaşım olduğu için ve temel değişken komşuluk arama algoritmasında “yerel arama” aşaması bulunduğu için bu başlık altında yerel arama yaklaşımı incelenmiştir. İlk olarak yerel arama hakkında bilgi verildikten sonra, diğer yerel arama tabanlı yaklaşımlardan kısaca bahsedilmiştir.

2.2.1. Yerel Aramanın Tanımı

Yerel arama yöntemleri en eski yöntemlerdir. Bir başlangıç çözümü ile başlamaktadır. Her s çözümünün komşuluğundaki çözümler setine ($N(s)$) dayanmaktadır. Yerel arama, her tekrarda tüm $N(s)$ çözümlerini değerlendirmesi gerekmemektedir, sadece bir alt setini incelemektedir (Dreo ve diğ., 2006). Bu tanımdan da anlaşılacağı gibi yerel aramanın temelinde “komşuluk” araştırarak çözüme ulaşma yer almaktadır. Bu doğrultuda, belirli bir hareket mekanizması ile komşu çözüm kümesinin oluşturulması gereklidir. Bu hareket mekanizması “komşuluk yapısı” olarak adlandırılmaktadır. Yerel aramada kullanılan komşuluk yapıları Bölüm 2.3 altında detaylı olarak ele alınmıştır.



Şekil 2.4: Yerel arama çözümünün ikili sunumunu kullanarak, en iyi komşuluğu seçme stratejisi.

Şekil 2.4'te Talbi'nin (2009) bir maksimizasyon problemine ait çözüm örneği gösterilmektedir. $s_0=(10001)$ için amaç fonksiyonu $f_0=2774$ değerli başlangıç çözümüyle başlayarak, son yerel optimum $s_y=(10000)$, $f_y=3027$ değeri ile gösterilmiştir. Global optimum çözümün ise $s_g=(01010)$, $f_g=4000$ olduğu ifade edilmiştir.

2.2.2. Yerel Arama Algoritması

Osman ve Kelly'in (1996) çalışmalarında yer alan yerel arama azalmaya ait algoritma kalıbı Şekil 2.5'te gösterilmiştir:

Adım 1: s ; Başlangıç çözümü olarak s oluşturulur ve amaç fonksiyonu $C(s)$ hesaplanır.

Tekrarlanacak Adımlar: İncelenmemiş s 'in komşu çözümleri s' var oldukça ($s' \in N(s)$)

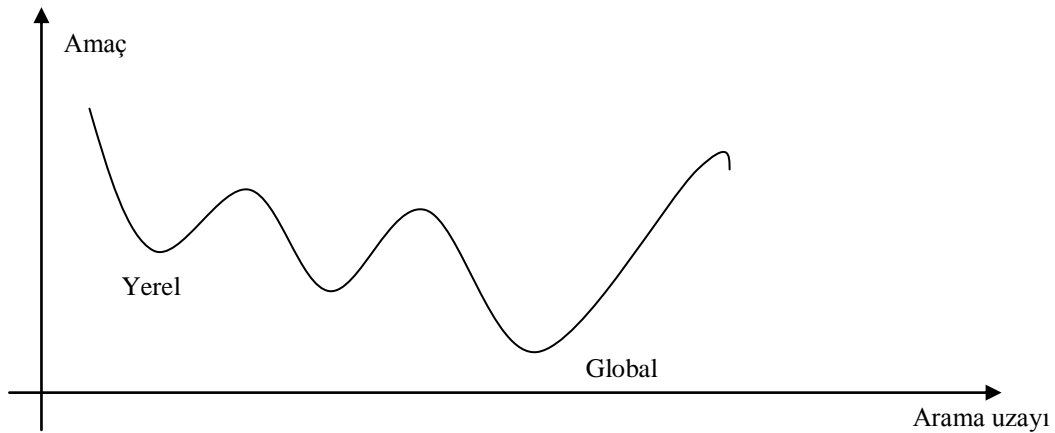
Adım 2: $s' \in N(s)$ aday komşuları üretilir. $C(s')$ amaç fonksiyonu değerleri hesaplanır.

Adım 3: Eğer, daha iyi komşu elde edilmiş ise ($C(s') \leq C(s)$); $s=s'$ gösterimi ile $s' \in N(s)$ olan daha iyi komşu seçilir. Aksi halde s tekrar alınır ve Adım2-3 tekrarlanır.

Sonlandırma: Son çözüm (yerel optimum) bulunmuş olur.

Şekil 2.5: Klasik yerel arama algoritması.

Elde edilen yeni çözüm değerinin önceki çözümle kıyaslanmasında amaç fonksiyonu minimizasyon ise azalma, maksimizasyon problemi için ise artış olduğunda yeni çözümün daha iyi olduğu söylenebilmektedir. Şekil 2.6'da azalma için yerel ve global optimum değerler gösterilmiştir.

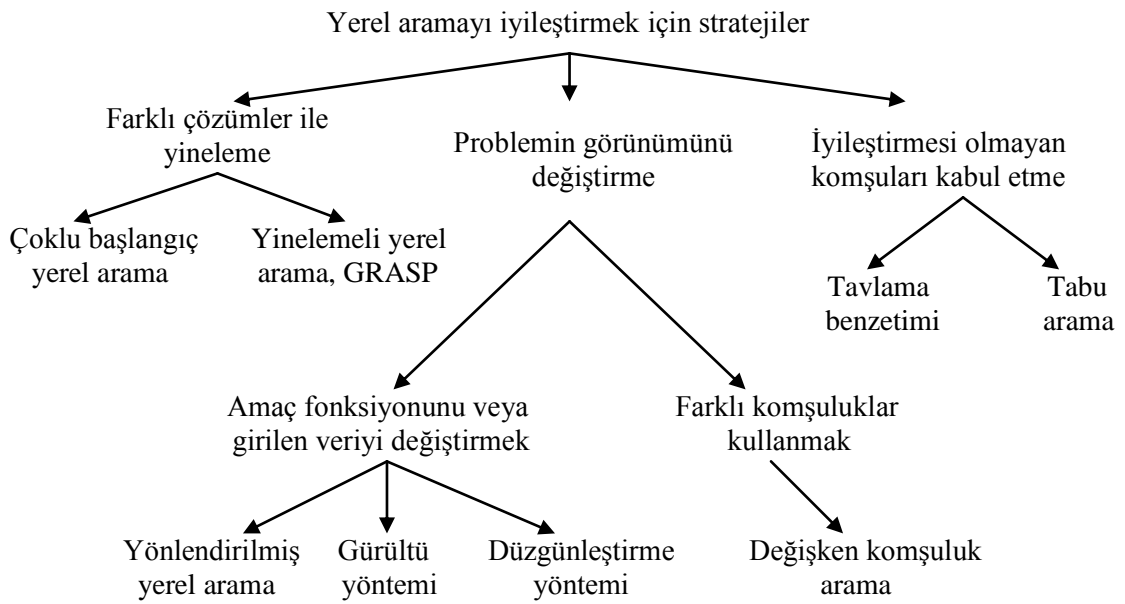


Şekil 2.6: Global ve yerel optimum gösterimi (Osman ve Kelly, 1996).

Yerel arama, çoğu sezgisel arama metodunun temel kaynağı olmuştur. Aday çözümlerin bulunduğu alanda arama yapmaktadır. Çözüm sunumu konusu önemlidir. Aday bir çözümden (muhtemelen rassal olarak üretilmiş) başlayarak, yerel arama amaç fonksiyonuna göre geçerli aday çözümden daha “iyi” bir “komşu”ya hareketi sağlamaktadır. Geçerli çözüme göre tüm komşular kalitesiz olduğu zaman yerel arama durmaktadır (Glover ve Kochenberger, 2003).

2.2.3. Yerel Aramayı İyileştirme Stratejileri

Yerel arama sezgiseli uygulanırken, her çözüm noktası için bir komşuluk tanımlanmaktadır (Hwang, 1996). Yerel arama çok hızlı şekilde iyi çözümler bulabilmektedir. Fakat yerel optimuma yakalanma gibi bir problemle karşılaşılabilir. Tüm komşularından daha iyi çözüm değerine sahip arama uzayında yer alan pozisyonlar, en iyi (global) olası çözümü de sağlamak zorunda değildir. Yerel aramanın etkinliğini arttırmak için yıllar boyunca çeşitli teknikler sunulmuştur. Tavlama benzetimi (TB), tabu arama (TA) ve yönlendirilmiş yerel arama gibi tekniklerin tümü, yerel aramayı yerel optimumdan kurtarmaya yönelik olarak geliştirilmiştir (Glover ve Kochenberger, 2003). Talbi (2009) kitabında yerel aramada yerel optimumdan kaçınmak için geliştirilen tek-nokta aramalı metasezgiselleri toplu olarak bir ağaç altında göstermiştir (Şekil 2.7).



Şekil 2.7: Yerel aramayı geliştirmek ve yerel optimumdan kaçınmaya yönelik tek-nokta aramalı metasezgisel ailesi.

Tez konusu olarak ele alınan deęişken komşuluk arama yaklaşımı da şekilde yer almaktadır. Diğer metasezgisellerden literatürde sıklıkla karşılaşılan bir kaç hakkında kısa bilgi verilmiştir. Deęişken komşuluk arama yaklaşımı malzeme ve yöntem bölümü altında detaylı olarak incelenmiştir.

2.2.3.1. Açgözlü Rassallaştırılmış Uyarlamalı Arama Yordamı

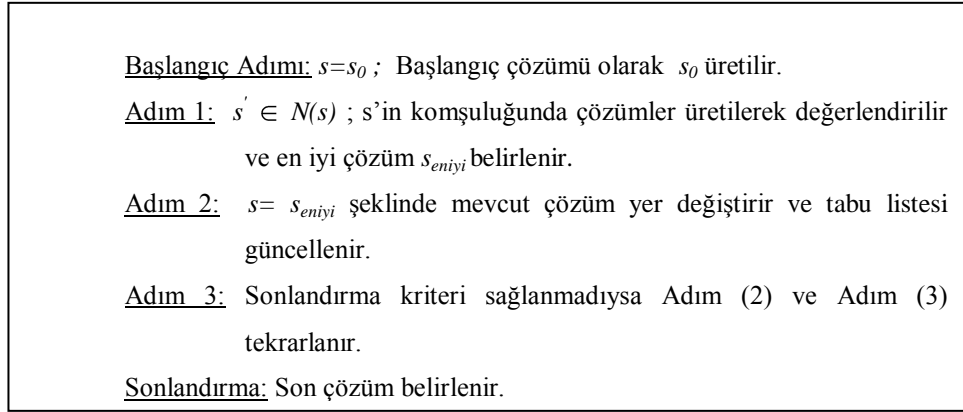
Açgözlü rassallaştırılmış uyarlamalı arama yordamı (greedy randomized adaptive search procedure) arama uzayının tekrarlı rassal örneklenmesine dayanmaktadır ve her bir tekrarda probleme ilişkin bir uygun çözüm hesaplanmaktadır. Rassallaştırılmış bir sezgisel dayalı “oluşturma aşaması” ve bu aşamada oluşturulan başlangıç çözümü iyileştiren “yerel arama aşaması” olmak üzere iki ana aşamadan oluşmaktadır. Tekrarlama süreci belirlenen bir tekrar sayısına ulaşıncaya kadar sürdürülür ve o ana kadar elde edilen en iyi çözüm son çözüm olarak değerlendirilmektedir (Alpay, 2008).

2.2.3.2. Düzgünleştirme Yöntemi

Arama uzayının düzleştirilmesi çözülecek problemi basitleştirmektedir. Düzleştirilmiş alanda orjinal alana göre daha az yerel optimum bulunmaktadır. Tüm tek nokta aramalı metasezgisel yöntemler düzgünleştirme yöntemi ile birlikte kullanılabilirler (Talbi, 2009).

2.2.3.3. Tabu Arama Metasezgiseli

F. Glover tarafından geliştirilen tabu arama (TA) algoritması, zor optimizasyon problemlerinin çözümü için ortaya konulmuş olan tekrarlamalı bir optimizasyon yöntemidir. Yöntemin esası bir çeşit komşuluk araştırma mekanizmasına dayalıdır. TA algoritması, bölgesel en iyi çözümün daha ilerisindeki çözümleri araştırabilmesi için bölgesel-sezgisel araştırmaya kılavuzluk yapmaktadır. TA'nın bölgesel en iyiyi aşmak için kullandığı temel prensip, değerlendirme fonksiyonu adı verilen fonksiyonun her tekrarlama en yüksek değerlendirme değerine sahip hareketi bir sonraki çözümü oluşturmak amacıyla seçmesine dayanmaktadır (Karaboęa, 2011). Temel TA algoritmasının kalıbı Şekil 2.8'de gösterilmiştir.



Şekil 2.8: Temel tabu arama algoritması (Karaboğa, 2011).

TA algoritmasının en önemli özelliklerinden biri, bir tabu listesine sahip olmasıdır. Belirli kriterleri sağlamayan ve diğer çözümlere göre kalitesi düşük olan çözümler bu listeye dahil edilerek araştırmanın yeniden bu çözümler üzerine yönelmesi engellenmiş olmaktadır. Tabu listesine dahil edilecek olan çözümlerin belirlenmesi bazı tabu şartlarına (tabu conditions) göre yapılmaktadır ve bu şartlar yakınlık hafızası ve sıklık hafızası olmak üzere iki önemli faktöre dayalıdır. Bir çözümün en son ne zaman denendiğini yakınlık hafızası, hangi sıklıkla denendiği bilgilerini ise sıklık hafızası saklamaktadırlar. Bu bilgiler değerlendirilerek, ilgili çözümün tabu listesine alınıp alınmayacağına karar verilir. Hafıza kriterleri sayesinde, araştırma sırasında çözüm vektörüne ait elemanların belirli bir disipline bağlı olarak değişim göstermeleri sağlanmış olur (Bağış, 2002).

2.2.3.4. Tavlama Benzetimi Metasezgiseli

Ted Carson'un (2001) standart yerel arama algoritmalarından bir tanesi olarak belirttiği tavlama benzetimi (TB) de komşu arama hareketine dayalı yöntemlerdendir. Fizik biliminde bir katının minimum enerji durumu elde edilene kadar yavaş yavaş soğutulma işlemine tavlama denilmektedir. Alp ve diğ. (2001) TB için "tavlama sürecini taklit eden olasılıklı bir arama yöntemi" ifadesini kullanmışlar ve bu yöntem ile üretilen çözümlerde amaç fonksiyon değerinin genel bir azalma eğiliminde olduğunu, bazı durumlarda amaç fonksiyon değeri yüksek olan çözümlerin de kabul edildiğini belirtmişlerdir. Bu yolla, yerel bir minimum çözüme takılmaktan kurtularak, daha iyi bir yerel veya global çözüm için arama yapmaya devam etmek amaçlanmaktadır. TB'nin kombinatoral optimizasyon problemleri için global eniyeye yakın çözümler veren

kullanışlı bir meta sezgisel yaklaşım olduğu belirtilmiştir (Alp ve diğ., 2001). TB algoritmasının kalıbı şekil 2.9'da gösterilmektedir:

<p><u>Başlangıç Adımı:</u> $s=s_0$; Sezgisel olarak veya rasgele bir başlangıç çözümü olarak s_0 oluşturulur. Sıcaklık T için başlangıç değeri T_s ve durdurma kriteri gibi diğer parametreler belirlenir.</p> <p><u>Adım 1:</u> $s' \in N(s)$; s''in komşuluğunda olan s' üretilir. $s \in N(s)$.</p> <p><u>Adım 2:</u> $\Delta = C(s) - C(s')$; Üretilen çözüm s' ile s çözümünün amaç değerleri ($C(s)$ ve $C(s')$) arasında ki fark hesaplanır.</p> <p><u>Adım 3:</u> <i>Eğer;</i></p> <p style="padding-left: 40px;">(i) $\Delta < 0$, yani s', s 'ten daha iyi ise</p> <p style="padding-left: 40px;">veya</p> <p style="padding-left: 40px;">(ii) s', s 'ten daha kötü ama mevcut sıcaklık T'de $-\Delta/(e^T) > \theta$ ile tanımlı $p(\delta E) = \exp(-\delta E/kT)$ işlemi ile kabul edilmiş ise s çözümü s' ile yer değiştirir. Aksi halde, s mevcut çözüm olarak korunur.</p> <p><u>Adım 4:</u> belirli kurallar setine göre T değiştirilir (soğutma tarifesine göre, üretilen çözümde gelişme olup olmamasına göre v.b. gibi)</p> <p><u>Sonlandırma:</u> durdurma kriteri sağlanıyorsa araştırma durdurulur, aksi halde Adım (1)'e gidilir.</p> <p>θ: 0 ile 1 arasında rasgele üretilen bir sayıdır.</p> <p>$p(\delta E)$: termodinamik kanunlarına göre T sıcaklığında enerjide δE genlikli bir artış olma olasılığıdır.</p>
--

Şekil 2.9: Temel tavlama benzetimi algoritması (Karaboğa, 2011).

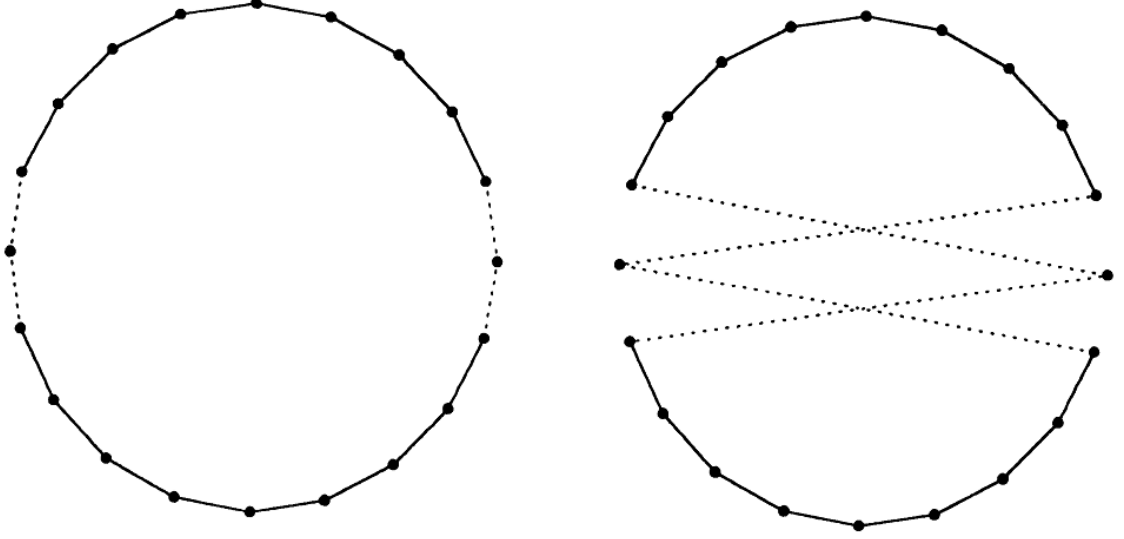
2.2.4. Geleneksel Yerel Arama Sezgiselleri

Hwang (1996), gezgin satıcı problemleri için tur geliştirici sezgisellerden biri olarak yerel aramayı ele almıştır. Yerel aramalar, sadece geçerli çözüm üzerinde gelişme sağlayan değişimlere odaklanmaktadır. Yerel arama algoritmalarının bir kusuru kolaylıkla yerel optimum çözümde takılı kalmasıdır. Hwang (1996) geleneksel yerel arama sezgisellerini şöyle sıralamıştır:

2.2.4.1. Şehir-takas sezgiseli (City-swap)

Tur geliştirmede basit bir metoddur. Bir rota üzerinde iki şehrin değiştirilmesidir. Eğer toplam mesafede azalma meydana gelirse çözüm basitçe tutulur, aksi halde başka bir değişim denenmektedir. Şekil 2.10'da Hwang'ın (1996) örneğine yer verilmiş ve solda

orijinal tur, sağda ise şehir-takas sezgiseli uygulandıktan sonra elde edilen tur gösterilmiştir.



Şekil 2.10: Gezgin satıcı problemi için şehir-takas hareketi.

Şekil 2.12'den de görüldüğü gibi 2-opt hareketinde turda bulunan bir bölüm ters çevrilmiş olmaktadır.

2.2.4.2. 2-opt Sezgiseli

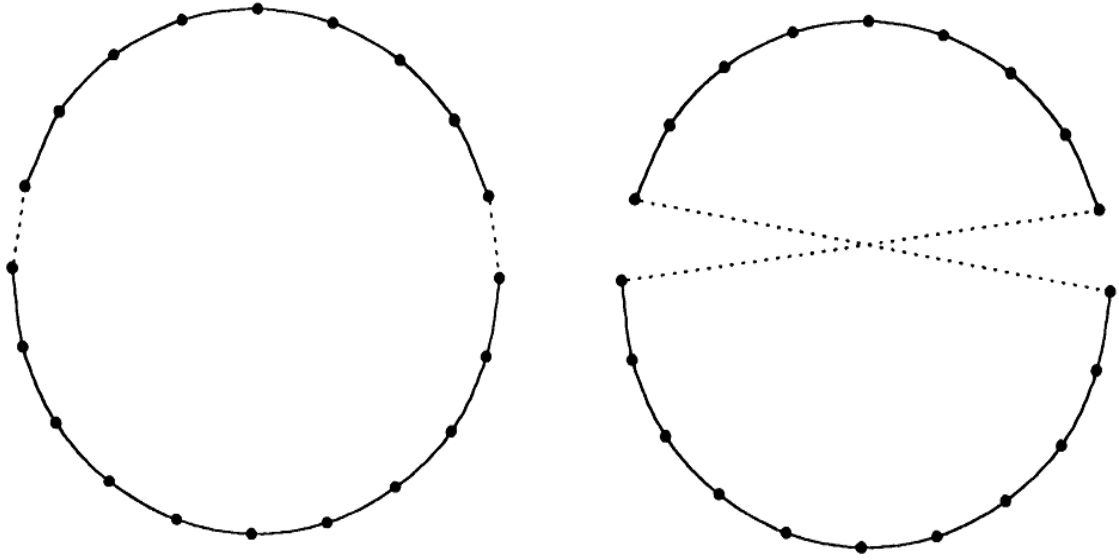
Şehir-takas sezgiseline ek olarak, gezgin satıcı problemleri için yerel geliştirme sezgisellerinde en iyi bilinen bağlantı değişimi prosedürüdür. Genel olarak, bağlantı değişimi prosedürleri 'r-opt' prosedürü olarak adlandırılırlar, 'r' değişecek bağlantı sayısıdır. Bir bağlantı iki şehir arasındaki ilişki olarak adlandırılmaktadır. r-opt süreci Şekil 2.11'de gösterilmiştir:

Adım 1. r kenarı bir rotadan silinir ve yeni bir tam rota oluşturmak için yeni bir r kenarı eklenir. Eğer sonuçta oluşan tur daha kısa ise değişim elde tutulur, aksi halde başka olası değişimler farklı kenar silme/ekleme ile denenir.

Adım 2. tüm uygun değişimler tarafından hiç bir gelişme olmayana kadar adım 1 tekrar edilir.

Şekil 2.11: r-opt adımları.

Genellikle daha büyük r , global optimum çözüme daha yakın çözümler sağlayabilen daha güçlü sezgisellere yol göstermektedir. Ancak, daha büyük r ayrıca hesaplama zamanını artıran olası değişimlerin çok sayıda testini de gerektirmektedir. Yani r sayısı arttıkça zaman karmaşıklığı da artmaktadır. $r=2$ ve $r=3$ değerleri en sık kullanılan değerlerdir (Hwang, 1996). 2-opt yerel arama sezgiselini ilk olarak 1958 yılında G. A. Croes kullanmış, 3-opt yerel arama sezgiselini ise ilk olarak 1958 yılında Bock sunmuştur (Laporte, 2006). Şekil 2.12’de solda orijinal tur, sağda ise şehir-takas sezgiseli uygulandıktan sonra elde edilen tur gösterilmiştir.



Şekil 2.12: Gezgin satıcı problemi için 2-opt hareketi. (Hwang, 1996).

2.2.4.3. Or-Opt Sezgiseli

3-Opt için olası değişimlerinin çok sayıda olmasından dolayı, değiştirilmiş 3-Opt prosedürü Or-Opt olarak adlandırılmıştır. Or-Opt 1976 yılında İlhan Or tarafından sunulmuştur. Or-Opt mevcut bitişik düğümlerin sadece bir, iki veya üç bağlantısını keserek elde edilen değişimleri göz önüne almaktadır (Hwang, 1996).

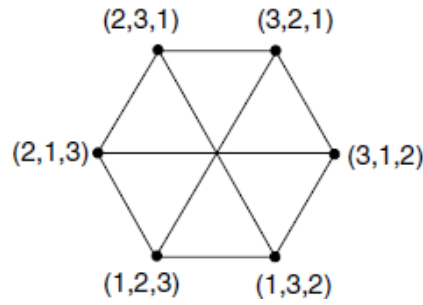
2.3. KOMŞULUK YAPISI

Yerel arama sezgiseli uygulanırken, her çözüm noktası için bir komşuluk tanımlanmaktadır (Hwang, 1996).

2.3.1. Komşuluk Yapısının Tanımı

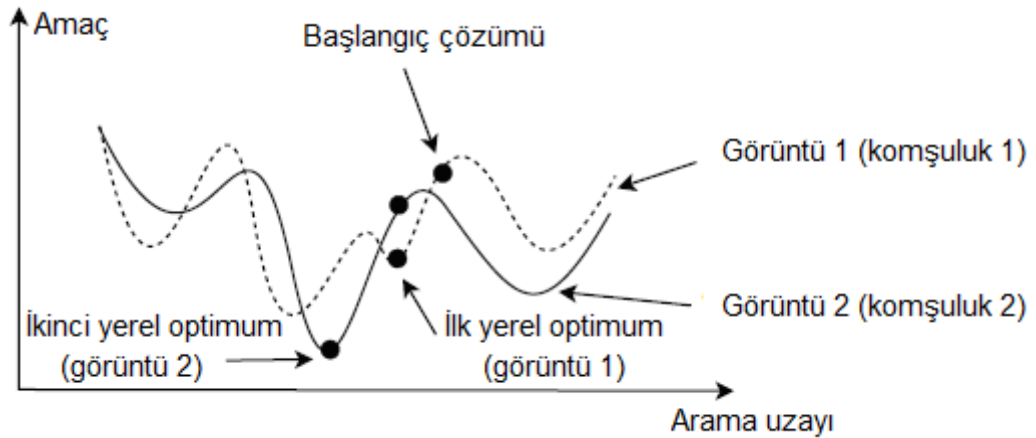
Komşuluk yapısı geçerli çözümü belirlenmiş bir düzenek yardımıyla değiştirerek yeni çözümler üretme şeklidir. Komşuluk yapısı, komşuluk araştırması temeline dayanan arama yöntemlerinin çözümünde çok önemli bir unsurdur (Geyik ve Cedimoğlu, 2010). Tanımdan da anlaşılacağı üzere problemin yapısına uygun komşuluk yapısı belirlenmediğinde metasezgisel yöntem problemin çözümde başarısız olacaktır.

Şekil 2.13'te bir permütasyon problemin için komşuluklar gösterilmiştir. Bu şekilde, örneğin, (2,3,1) çözümü için komşuluklar (3,2,1), (2,1,3) ve (1,2,3)'dir.



Şekil 2.13: 3 elemanlı bir permütasyon probleminin komşulukları. (Talbi, 2009)

Şekil 2.14'te Talbi'nin (2009) sunduğu iki komşuluk kullanan değişken komşuluk arama yaklaşımı için yerel optimum noktaları gösterilmiştir. Bu şekilde ilk yerel optimum komşuluk 1'e göre bulunmuş, komşuluk 2'ye göre ise, ilk yerel optimumdan 2.yerel optimum bulunmuştur.



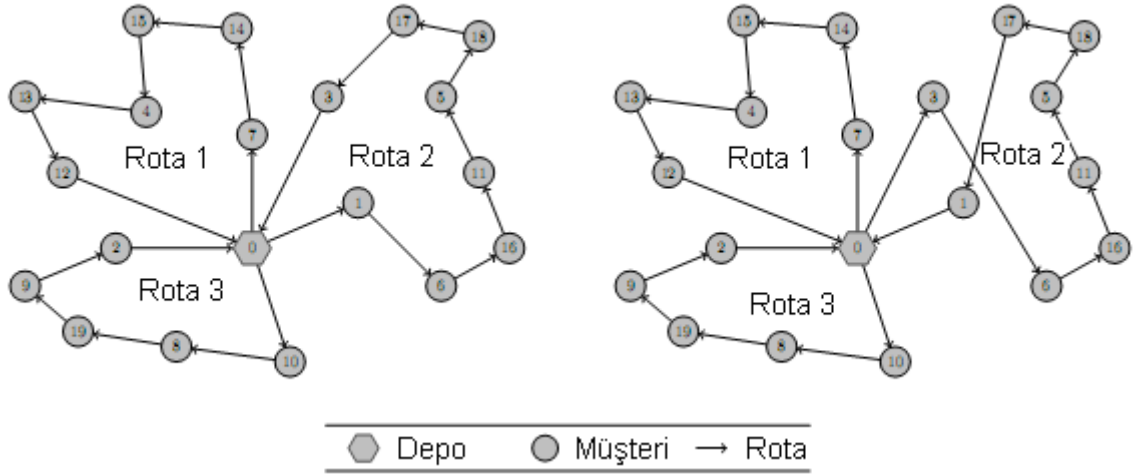
Şekil 2.14: İki komşuluk kullanan değişken komşuluk arama.

2.3.2. Başlıca Komşuluk Yapıları

Literatür taramasında karşılaşılan başlıca komşuluk yapıları açıklanarak ele alınmıştır.

2.3.2.1. Değişirme (exchange)/ Takas-iç (swap-intra)

Şevkli ve Aydın (2007) değişirme komşuluk yapısını; rassal olarak seçilen iki elemanın değiştirilmesi olarak tanımlamıştır. Bir iş sıralama probleminde [2 1 2 2 1 3 1 3 3], başlangıç sıralaması iken rassal olarak seçilen 2. ve 8. elemanlara “değişirme” uygulanması ile yeni sıralama [2 3 2 2 1 3 1 1 3] şeklinde olmaktadır. Şevkli ve Aydın’ın (2007) “değişirme” olarak gösterdiği ve çizelgeleme için örnek verdiği bu komşuluk yapısının, Ribas’ın çalışmasında (2011) rotalama için “takas-iç” adı ile belirtildiği görülmüş ve Şekil 2.15’deki örnek gösterilmiştir.



Şekil 2.15: Takas-iç komşuluk yapısı için örnek hareket. Rota 2 içinde (1) ve (3) müşterilerinin yer değiştirmesi.

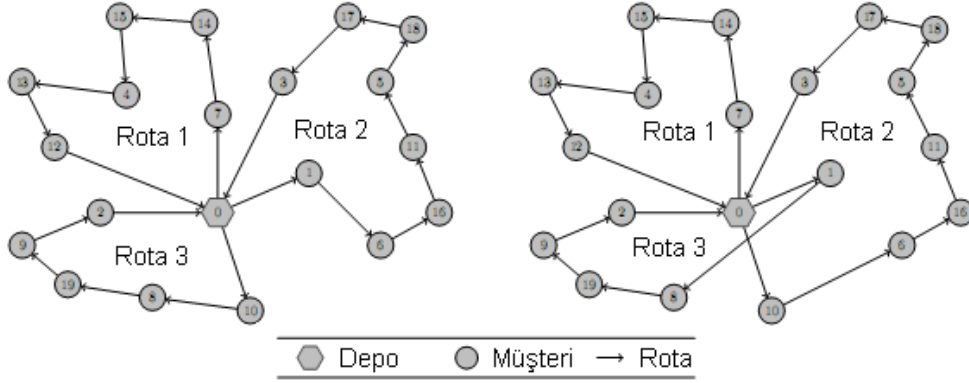
2.3.2.2. Girinti (insert)

Bu komşuluk yapısı için Şevkli ve Aydın (2007), rassal olarak seçilen bir elemanın başka bir rassalın ön veya arkasına yerleştirilmesi şeklinde tanım yapmış ve “değişirme” hareketi için verilen örnekteki başlangıç çözümünde rassal olarak 3. ve 6. elemanlar seçildiğinde (3., 6.’nın önüne gelerek) yeni sıralamanın [2 1 2 1 2 3 1 3 3] şeklinde olacağı örneğini vermiştir.

2.3.2.3. 1-1 Değişimi (1-1 interchange) / Takas Prosedürü (swap procedure)

Imran ve diğ.’nin (2009) rotalama problemlerinde belirttiği bu komşulukla, iki rotadan bir çift müşteriyi takas ederek optimum çözümü üretmek amaçlanmaktadır. Bu prosedür rassal olarak seçilen bir rotadan rassal bir müşteri olarak başlar ve bu müşteriyi diğer

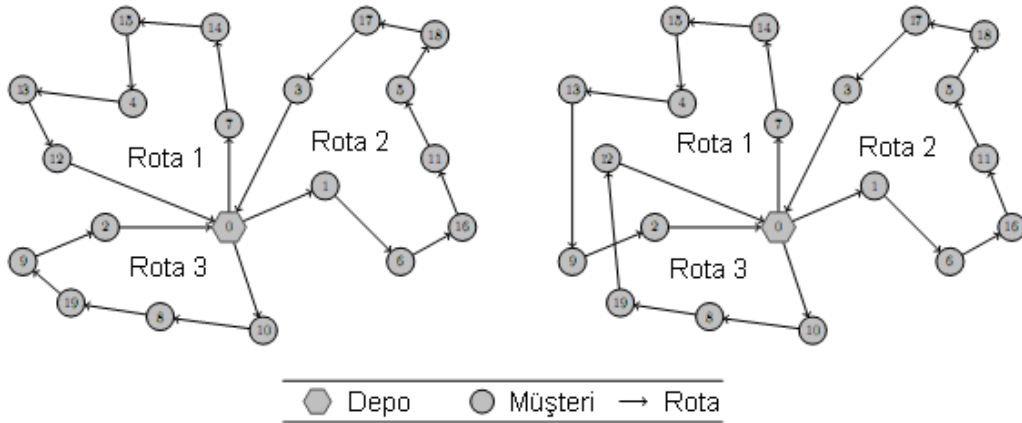
bütün rotalar dikkate alarak seçilen diğer müşteriler ile sistematik bir şekilde değiştirmeyi dener. Bu prosedür uygulanabilir bir hareket bulunana kadar devam eder. Şekil 2.16’da takas prosedürü için bir örnek gösterilmiştir.



Şekil 2.16: Rota 2’den (1) ve rota 3’ten (10) numaralı müşteriler için takas prosedürü. (Ribas, 2011)

2.3.2.4. 2-1 Değişimi (*interchange(2-1)*) / 2-1 Takas

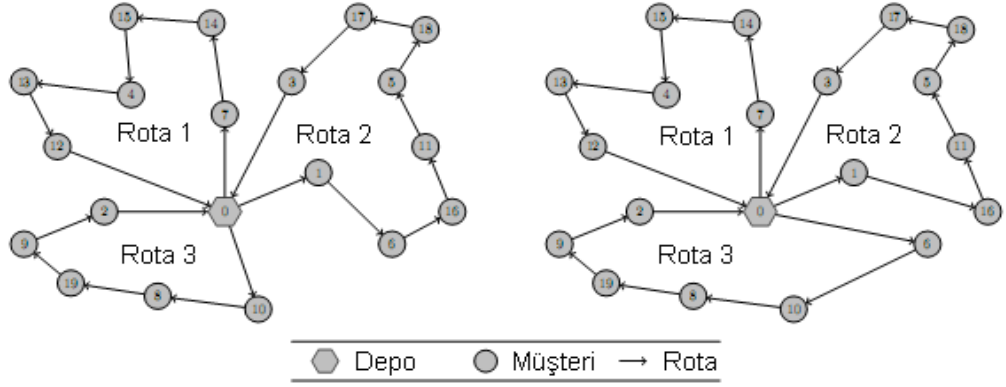
Bu ilave türü uygun bir hareket elde edilene kadar, rassal olarak seçilen bir rotadan ardışık iki rassal müşteriyi, sistematik olarak seçilen bir başka rotaya bu rotadan bir müşteriyi alırken kaydırmayı hedeflemektedir. Imran ve diğ.’nin (2009) 2-1 değişimi olarak gösterdiği bu komşuluk yapısı, Ribas’ın (2011) çalışmasında “2-1 takas” adı ile belirtilmiş ve Şekil 2.17’deki örnek gösterilmiştir.



Şekil 2.17: Takas (2,1) komşuluk yapısı için örnek hareket. Rota 3’ten (9) ve (2) numaralı müşterilerin, rota 1’den (12) numaralı müşterinin takası. (Ribas, 2011)

2.3.2.5. Kaydırma (Shift)

Bir rotadan seçilen bir müşterinin başka bir rotaya kaydırılmasıdır. Şekil 2.18’de kaydırma yapısı için Ribas’ın (2011) sunduğu örneğe yer verilmiştir.



Şekil 2.18: Rota 2’den rota 3’e (6) numaralı müşterinin kaydırılması prosedürü.

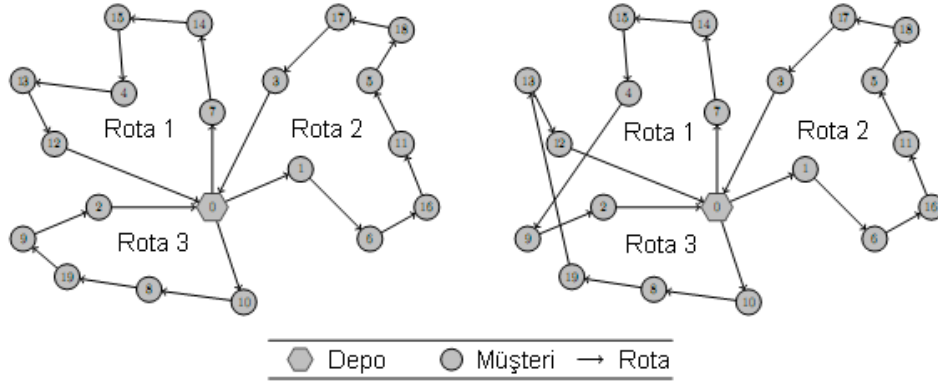
2.3.2.6. 2-0 Kaydırma (shift(2-0))

Imran ve diğ. (2009) bu komşuluk yapısını şu şekilde tanımlamıştır: 2-0 değişiminde rassal olarak seçilen bir rotadan iki ardışık müşteri seçilir. Bu iki müşteri sistematik bir şekilde diğer rotalara ilave edilmek konusunda beraber düşünülür. Bu prosedür uygulanabilir bir hareket bulunana kadar devam eder. Bu prosedür 1. tip 2-0 değişimi olarak adlandırılmaktadır. 2. tip 2-0 değişimi olarak adlandırılan diğer 2-0 kaydırması ise iki müşterinin iki farklı rotaya dahil edilmesine imkan tanınması hariç ilk kaydırma ile benzerdir.

2.3.2.7. 3-0 Kaydırma (shift(3-0))

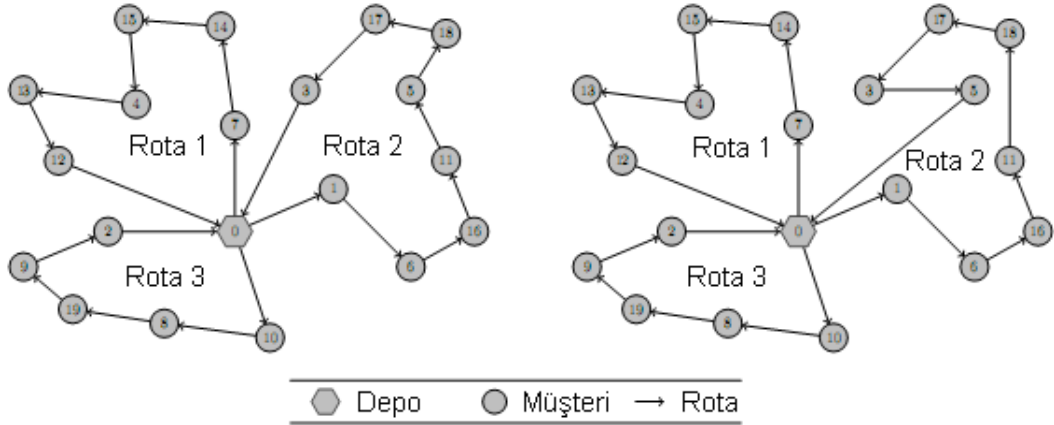
Bu komşuluk yapısında kaydırma yapısının temel mantığı kullanılmakta olup kaydırılan müşteri sayısı 3 olarak belirlenmektedir.

2.3.2.8. 2-2 Kaydırma (shift(2-2))



Şekil 2.19: 2-2 kaydırma komşuluk yapısı için örnek hareket. Rota 1'den (13) ve (12), rota 3'ten (9) ve (2). (Ribas, 2011)

2.3.2.9. k-Or-Opt



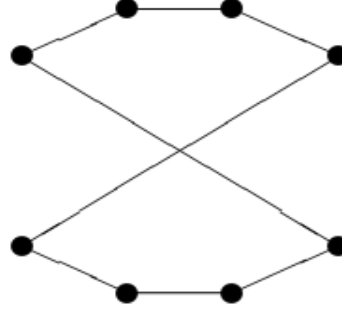
Şekil 2.20: k-Or-Opt komşuluk yapısı için örnek hareket. (k=3) için 18, 17 ve 3 müşterilerinin rota 2'de (11,5) bağlantısının arasına. (Ribas, 2011)

2.3.2.10. 2009 Yılında Sunulan Yeni Bir Düzensizlik Mekanizması

Imran ve diğ.'nin (2009) çalışmalarında geliştirdiği yeni bir mekanizmadır. Bu şema başlangıçta 1987 yılında Salhi ve Rand tarafından ARP için eş zamanlı üç rota düşünülerek geliştirilmiştir. Burada, rassal olarak seçilmiş bir rotadan rassal olarak seçilmiş bir müşteri almakla başlanmakta ve bu müşterinin alıcı rotadaki kapasiteyle zaman kısıtlarını düşünmeden bir başka rotaya yeniden yerleştirilmesi amaçlanmaktadır. Alıcı rotadan bir müşteri daha sonra eğer ikinci ve üçüncü rotalardaki kapasite ve zaman kısıtları aşılmıyorsa üçüncü bir rotaya aktarılmaktadır. Buna 1. tip düzensizlik adı

verilmiştir. Bu tür bir düzensizliğin bir açılımı bir rotadan iki ardışık müşteriye kaydıran bir türdür. Bu prosedürde başlangıçta bir müşterinin hareket ettirilmesi yerine iki müşteri birden hareket ettirilmektedir. Bu prosedür de 2. tip düzensizlik olarak adlandırılmıştır.

2.3.2.11. 2-opt Operatörü



Şekil 2.21: Gezgin satıcı problemi için 2-opt operatörü. (Attila,2005)

Gezgin satıcı problemi gibi permütasyon problemlerini çözmek için literatürde sıklıkla kullanılan iki operatör 2-opt ve şehir takasıdır. n adet şehir için komşuluk büyüklüğü; 2-opt operatöründe $(n(n-3)/2)$ formülü ile hesaplanmaktadır. Şehir takası (değişim) hareketinde ise $n(n-1)/2$ formülü ile hesaplanmaktadır (Fonlupt ve diğ., 1997). Şekil 2.21'de 2-opt operatörü gösterilmiştir. 2-opt operatörü, çizelgeleme problemleri için çok büyük bir değişim (yani zayıf bölge (locality)) oluştururken, GSP gibi rotalama problemleri için ise daha küçük değişim sağlaması sebebiyle (yani güçlü bölge) çok etkili bir operatördür (Talbi, 2009)

2.3.2.12. Tersine Operatör

Tersine operatörde, rassal olarak iki eleman seçilmekte ve elemanlar arasında kalan bölüm tersine çevrilmektedir. Özellikle genetik algoritmada da sıkça kullanılan bu operatör için Şekil 2.22'de bir örnek gösterilmiştir (Ferreira, 2002). (a) durumunda seçilen bölüme tersine operatörü uygulandıktan sonra (b) halini almıştır.

(a)	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
	b	h	f	g	k	l	m	C	L	E	B	F	K	G	J	H	I	A
(b)	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
	b	h	f	g	k	l	m	K	F	B	E	L	C	G	J	H	I	A

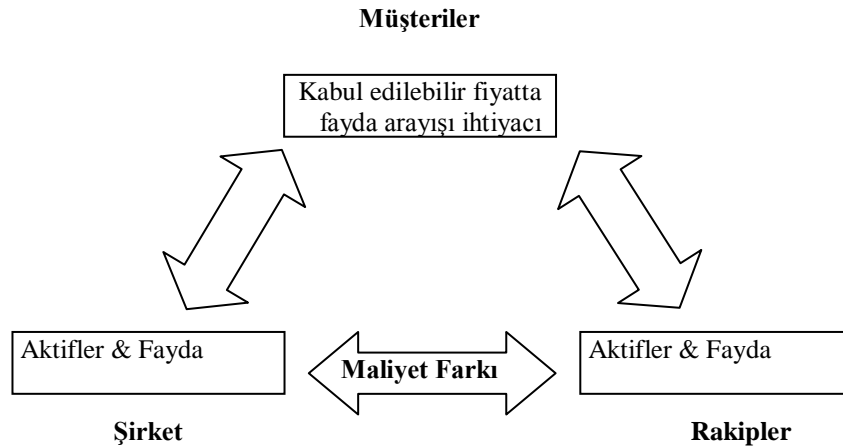
Şekil 2.22: Inversion (tersine) operatörü.

Literatür taramasında genel olarak en basit komşuluk yapıları olan kaydırma, değiştirme ve girinti yapıları gözlenmiştir. Bu temel yapıların hareketlerinin rota-içi, rotalar-arası olacak şekilde ve hareketin yönüne göre, değişim gören nokta sayısına göre çeşitlendirildiği gözlenmiştir.

2.4. TEDARİK ZİNCİRİ YÖNETİMİ

2.4.1. Tedarik Zinciri Yönetiminin Tanımı

Günümüzün rekabet koşulları, firmaların müşteri isteklerine rakiplerinden daha hızlı cevap vermesini, hatta müşterinin ne isteyeceğinin analizinin yapılmasını gerekli kılmaktadır. Rekabet avantajı daha çok firmanın rekabet ettiği iş alanında oluşturduğu rekabet stratejisine ilişkin bir kavram olarak algılanmaktadır (Porter, 1987). Firmaların rakiplerine karşı rekabet üstünlüğü elde edebilmelerinin kaynağı ilk olarak organizasyonların kendilerini farklı kılabilmeye yeteneklerinde bulunmaktadır. Bu farklılık müşteri açısından ele alınmalıdır. Rekabet avantajının ikinci kaynağı ise; daha düşük maliyetle işleme ve bu sebeple daha fazla kazanç elde etmesindedir (Christopher, 1998). Şekil 2.23'te Christopher'in (1998) kitabında yer verdiği rekabet avantajı ve "müşteri", "rakip", "firma" kelimelerinin İngilizce karşılıklarının başharflerinden oluşan 3C (customer, competitor, company) ilişkisi gösterilmiştir.



Şekil 2.23: Rekabet avantajı ve 3C.

Christopher (1998) herhangi bir rekabetçi ilişkide başarının temellerini oluşturan ticari başarının maliyet avantajı veya değer avantajından çıktığını (idealist olanı her ikisinin

de olmasıdır) belirtmiştir. Herhangi bir endüstri sektöründeki en kazançlı rakip, en düşük maliyeti sağlayan ve kaynaklarını en etkin şekilde kullanabilen işletmelerdir. Basit ifadeyle, başarılı firmalar ya verimlilik avantajına ya da değer avantajına veya her ikisinin de kombinasyonuna sahiptir. Verimlilik avantajı daha düşük maliyet profili sunmakta; değer avantajı ise ürünün veya rakip ürünlerin üzerinde farklı ilaveler sunmaktadır (Christopher, 1998).



Şekil 2.24: Lojistik ve Rekabet Avantajı (Christopher, 1998)

Maliyet avantajını elde etmek lojistik yönetiminde de gereklidir ve çoğu endüstride lojistik maliyeti toplam maliyet içinde önemli bir oran kaplamaktadır (Christopher, 1998). Lojistik Yönetim Konseyi*, lojistik kavramını şu şekilde tanımlamaktadır:

“Lojistik; müşteri ihtiyaçlarını karşılamak amacıyla, mal, hizmet ve ilgili bilgilerin üretim ve tüketim noktaları arasındaki ileri ve geri yöndeki akışlarıyla depolanmalarının etkin ve verimli bir şekilde plânlanması, uygulanması ve kontrolünü kapsayan faaliyetlerdir.”

(<http://cscmp.org/aboutcscmp/definitions.asp>, [Erişim tarihi:5 Ağustos 2011])

Rakiplerine karşı rekabet avantajı sağlamak için artık çok sayıda işletme, tedarik zincirlerinde yer alan diğer firmalarla, tüm tedarikçileriyle ve müşterileriyle olan ilişkilerine önem vermekte ve bu doğrultuda çalışmalara yoğunlaşmaktadırlar.

Müşteriye daha düşük maliyette ve daha yüksek servis seviyesinde hizmet ve de mal sunma amacı ile; fabrikalar birimler bazında farklı kararlar almak yerine daha bütünleşik ve düzenli tasarım ile bütün parçaların kontrolüne yönelmişlerdir. Tedarik

* The Council of Logistics Management

zinciri yönetimi de ticaret ortamında son zamanlarda çok fazla kabul gören bir alan olmuştur (Thomas ve Griffin, 1996).

Tedarik zinciri kavramı ile ilgili olarak literatürde çok sayıda tanım mevcuttur. Örneğin, Christine Harland'ın (1996) tedarik zinciri yönetimi tanımı şu şekildedir: *“Firmanın aktivitelerini ve ilişkilerini; organizasyon içinde, yakın tedarikçiler ile, tedarik zinciri boyunca birinci-ikinci tedarikçiler ve müşterilerle tedarik zincirinin tamamı boyunca yönetme işidir.”*(Tan, 2001a)

Ganeshan ve Harrison'un (1995) tariflerinde ise tedarik zinciri; *“Malzemenin satın alınması, bu malzemenin ara ürün ve nihai ürüne dönüştürülmesi ile bu nihai ürünün müşterilere dağıtım işlevlerini gerçekleştiren araçlar ve dağıtım seçenekleri ağı”* olarak anılmaktadır.

APICS Sözlüğünde tedarik zinciri tanımı, *“Tedarik zinciri, ürün meydana getirmek ve müşteriye hizmet sağlamak için zincire değer katan, firma içi ve firma dışı fonksiyonlar bütünü”* şeklinde yapılmaktadır.

Thomas ve Griffin (1996) tedarik zinciri yönetimi tanımını ise *“Satıcılar, imalat ve montaj fabrikaları, dağıtım merkezleri gibi tesislerin hem içinde, hem de arasındaki malzeme ve bilgi akışının yönetimidir.”* şeklinde yapmıştır.

Global tedarik zinciri forumuna (global supply chain forum) göre tedarik zinciri yönetimi, son kullanıcıdan temeldeki tedarikçilere kadar ürün, hizmet, bilgi sağlayan ve müşterilerle birlikte diğer hissedarlara da değer kazandıran anahtar işletme süreçlerinin bir entegrasyonudur (Lambert, 1998).

2.4.2. Tedarik Zinciri Yönetimi Kavramının Ortaya Çıkışı

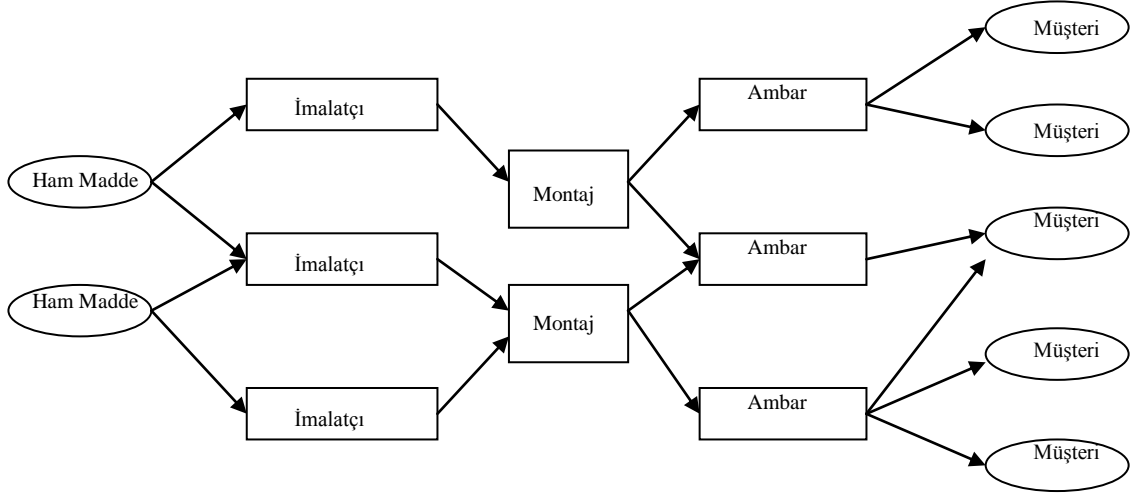
1970'lerde üretim kaynak planlaması (Manufacturing Resource Planning - MRP II) yöntemleri kullanımı ile yüksek stokun üretim maliyetleri, kalite, yeni ürün geliştirme ve tedarik zamanı üzerindeki etkisi fark edilmiş ve üreticiler, performansı artırmak için yeni malzeme yönetimi kavramları üzerinde durmaya başlamışlardır. 1980'lerde, global rekabetin şiddetlenmesi, dünya çapındaki firmaları daha düşük maliyetle, yüksek

kalitede, tasarım esnekliğine sahip, güvenilir ürünler sunmaya mecbur kılmıştır. Üreticiler, üretimin etkinliğini ve çevrim süresini geliştirmek için tam zamanında üretim (Just in Time - JIT) ve diğer yönetim felsefelerini kullanmışlardır. Az envanter bulunduran hızlı tam zamanında üretim ortamında yönteminin kullanılmaya başlanması ile, üreticiler, stratejik ve işbirlikçi alıcı-tedarikçi ilişkisinin önemini ve sağlayacağı faydaları fark etmeye başlamışlardır. Üreticilerin mevcut tedarikçileri ile stratejik ortaklıklar kurmaya başlamaları ile tedarik zinciri yönetimi kavramı ortaya çıkmıştır (Tan, 2001a).

Tarih boyunca, tedarik zincirinin üç temel aşaması; alım, üretim ve dağıtımdır ve bunlar bağımsız olarak yönetilmekte ve büyük envanterlerle tamponlanmaktadırlar. Bu politika şimdi artık değişmiştir. Özellikle üreticiler giderek artan şekilde lojistik konuların yönetimine önem vermeye başlamışlar ve onlara rekabet avantajı sağlayacak strateji arayışına girmişlerdir. Zincirdeki bir bölümün performansının artırılması yerine lojistik zincirinin geliştirilmesiyle daha fazla kazanç elde etmenin mümkün olduğu bilinen bir gerçektir (Korpela ve diğ., 2001).

Depo ağı tasarımı üzerine çalışma yapan Korpela ve Lehmusvaara (1999) müşteri odaklı yaklaşımı hedef almıştır. Geleneksel tedarik zinciri ağ tasarım problemi, verilen kısıtlar içinde minimum maliyet ya da maksimum kazanç elde etmeye odaklanmaktadır. Bu; mümkün olabilecek en iyi kazancı elde etmek isteyen bir dağıtım firması için ağ yapısının başlıca stratejik sonucudur.

Tedarik zinciri, hammaddelerin ve alt montaj parçalarının temin edilmesiyle başlamaktadır. Thomas ve Griffin (1996) tedarik zinciri yönetiminde operasyonel planlamayı; alıcı-satıcı, üretim-dağıtım ve stok-dağıtım koordinasyonları şeklinde üç başlık altında ele almışlardır. Tedarik zincirinin geleneksel üç aşaması olarak belirtilen elde etmek, üretim ve dağıtım aşamalarının her biri dünyanın farklı yerlerinde birkaç tesisten ibaret olabilmektedir. Şekil 2.25'te bu yapı gösterilmiştir.



Şekil 2.25: Bir tedarik zinciri ağının genel yapısı (Thomas ve Griffin, 1996)

2.4.3. Tedarik Zinciri Ağının Aşamaları

Erengüç ve diğ. (1999) tedarik zinciri ağı için üç ayrı aşamada odaklanmışlardır. Bu aşamalar şu şekildedir;

1.Aşama; Tedarik ağı m adet malzeme tedarikçisinden ibarettir ve birkaç fabrikaya hizmet sunmaktadır. Bu ağ; tüm ham malzeme, yarı mamul ve yardımcı malzeme tedarikçilerinden ve fabrika için herhangi bir başka tedarikçilerden oluşmaktadır.

2.Aşama; Asıl taşıma işleminin ortaya çıktığı ve üretim/hizmet' in olduğu n adet fabrikadan ibarettir.

3.Aşama; k adet dağıtımçıdan oluşan bir dağıtım ağıdır.

Bu üç aşama için alınan kararları ise şu şekilde sıralamışlardır:

2.4.3.1. Tedarikçi Aşaması

Bu aşamadaki tedarikçi, Tedarik Dağıtım Merkezi ve fabrika dağıtım aşamasında sırasıyla fabrika, dağıtım merkezi ve müşteri üzerinde rol oynamaktadır. Tipik bir imalatçı toplam satışının %60'ını hammaddeleri, alt montaj ve esas parçalar gibi maddeleri satın almada harcamaktadır. Bu aşamada ağın ana kararları şunlardır:

- Tedarikçi nasıl seçilir? Tedarikçi seçiminde ne gibi kriterler söz konusudur?
- Her bir kategori veya malzeme grubu için kaç adet tedarikçi bulunmalıdır?

- Her bir tedarikçi ile ne çeşit bir ilişki kurulmalıdır?
- Her bir tedarikçiden yüklemenin hacmi ve sıklığı nedir? Tedarikçiden yükleme direkt olarak veya ara stok noktalarına yapılabilmektedir. Bilgi tedarikçi ve Tedarik Dağıtım Merkezi fabrikaları arasında nasıl paylaşılmalıdır? Her Tedarik Dağıtım Merkezi'nde muhafaza edilmesi gereken her malzemenin stokunun toplamı nedir? Her Tedarik Dağıtım Merkezi'nde stok kontrolü için ne gibi karar kuralları kullanılmalıdır?
- Dağıtım Ağı: Tedarikçiden fabrikaya dağıtım ağının şekli nedir? Tedarikçiden yüklenen eşya fabrikaya direkt mi gönderilmeli yoksa tedarikçi ve fabrika arasında ana stok noktaları olmalı mıdır? Sonrakinin avantajı; bir merkezi yer olabilmesidir. Burada her bir fabrikaya girdilerin kalitesi izlenebilmektedir.
- Yerleştirme/dağıtma Kararı: Kaç adet Tedarik Dağıtım Merkezi işletilmelidir? Nereye yerleştirilmelidirler? Hangi tedarikçi hangi Tedarik Dağıtım Merkezi'nin hangi fabrikasına ne miktarda yükleme yapılmalıdır?

2.4.3.2. Fabrika Aşaması

Bu aşamanın önemli kararları:

- Taşıma Ağı: Taşıma süreçlerinin ağ konfügrasyonu nedir? Bu ağda fabrikaları bağlanmaya zorlayan kapasitenin yapısı nedir?
- Yerleştirme/Dağıtma Kararı: Taşıma süreçleri nerede yerleştirilmelidir? Besleyici fabrikalardan stok akışı organizasyonel sınırları aşar mı?
- Stok: Taşıma ağındaki diğer fabrikalara sahip bir fabrikada stok uygulamalarının etkisi nedir? Stoğa ait olan, merkezileşmemeye karşı merkezleşmenin etkileri nelerdir?

2.4.3.3. Dağıtım Aşaması

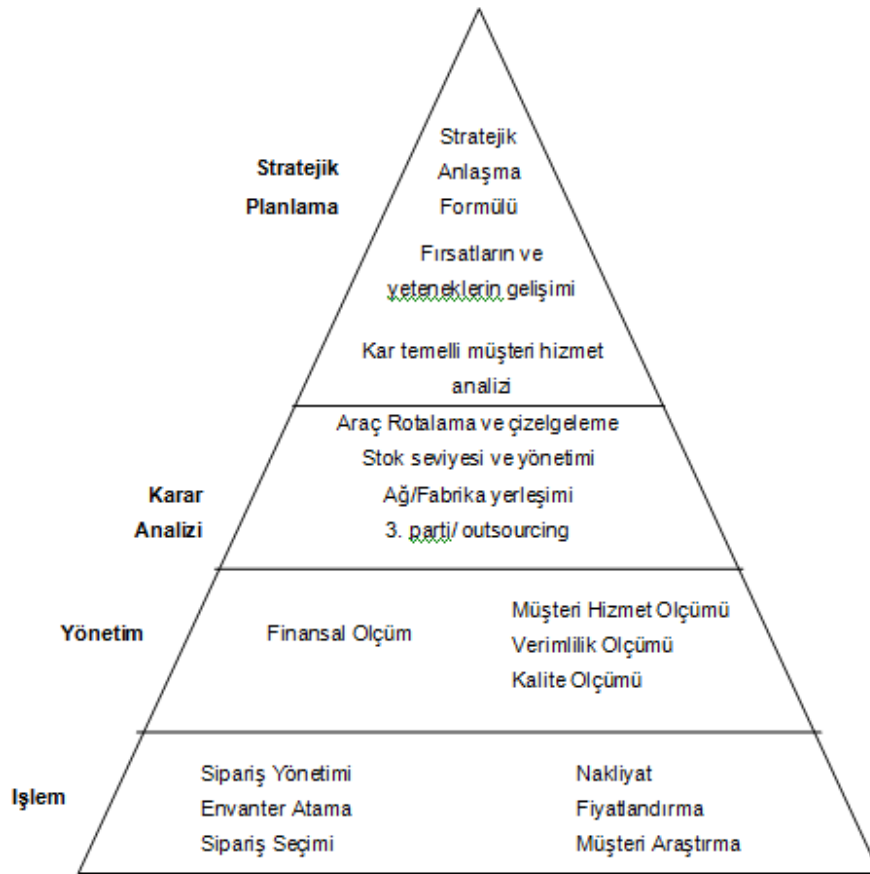
Bu aşamanın önemli kararları:

- Dağıtım Ağı: Dağıtım kanalının ağ konfügrasyonu nedir? Tedarikçi ve dağıtım merkezleri arasında ara stok noktaları var mıdır? Son müşteri memnuniyeti talebi nasıldır? Dağıtım merkezleri müşteriye doğrudan mı tedarikte bulunuyor, yoksa talepler perakende pazarından mı karşılanıyor?
- Yerleştirme/Dağıtma Kararı: Kaç adet dağıtım merkezi işletilmeli? Nereye yerleştirilmeliler? Her merkez için tam kapasite nedir? Hangi ürün talebi, her bir dağıtım merkezi tarafından kullanılmalıdır?

- Stok: Her dağıtım merkezinde sürdürülen her ürünün stok toplamı nedir? Her merkezde stok kontrolü için hangi kararlar yerine getirilmelidir? Dağıtım ağında bilginin rolü nedir?

Chandra ve Tumayan (2003) da bu 3 aşamayı almış, fakat ilave olarak 4. aşama olarak “tüketim aşaması” nı da ilave etmişlerdir.

Bowersox (2002) tedarik zincirindeki bilgi fonksiyonu Şekil 2.26’daki gibi göstermiştir. Karar çeşitleri ve alındıkları aşamalar verilmiştir.



Şekil 2.26: Tedarik zinciri bilgi fonksiyonu (Bowersox, 2002)

2.4.4. Tedarik Zinciri Yönetiminde Karşılaşılan Problemler

Tedarik zinciri yönetimi (TZY) problemi üç ana uygulama alanında ele alınabilmektedir. Bunlar; müşteri yönetimi, üretim yönetimi ve ürün geliştirmedir. Müşteri yönetimi hedef pazarın tanımlanması ve hedef müşteri kitlesine yönelik

programların uygulanması ile ilgili faaliyetleri ele almaktadır. Üretim yönetimi satın alma, talep tahmin, siparişlerin karşılanması ve lojistik gibi farklı süreçleri içermekte, ürün geliştirme ise, tedarik zinciri yönetimi bağlamında, tedarik zinciri için tasarım ve yerleştirme için tasarım gibi stratejileri kapsamaktadır (Küçükdeniz, 2004).

Uzun zamandır taşıma faaliyetlerinin firmalar için ne kadar yüksek maliyetlere sahip olduğu bilinen bir gerçektir. Lojistik operasyonların bir parçasını oluşturan taşıma; organizasyonun, müşteri hizmetine rekabetçi açıdan bakmak, ücret ve kalite açısından rekabetçilik ve pazar taleplerine cevap vermede esnekliğe sahip olmak gibi asıl hedeflerini doğrudan destekleyen ve aynı zamanda değer ekleyen bir süreçtir. Lojistiğin odağında, toplam taşımayı veya lojistik maliyetlerini minimize etmek ya da tedarikçinin kazancını maksimize etmek kadar, müşteriye daha iyi hizmet sağlamak da yer almaktadır. Ayrıca, lojistik yöneticileri bütün lojistik zincirinin kontrol edilmesine, lojistik prosesindeki bir kademenin problemlerine konsantre olmaktan daha çok önem vermektedirler. Bunun bir sonucu olarak; ortaklık ve müşteri memnuniyeti konularına olan ilgi gittikçe artmaktadır. Firmalar daha düşük maliyet ve yüksek hizmet kademeleriyle müşteriye daha iyi mal ve hizmet sağlamak için; ayrıştırılmış bir karar verme sürecinden, daha koordine olmuş ve birleştirilmiş tasarımlara yönelmekte ve tedarik zincirlerini kontrol etmektedirler. Firmalar ayrıca değişimlere etkili olarak cevap verebilmeli, rekabet edebilmek ve pazarda görülen fırsatların avantajlarından yararlanarak kazanç oluşturabilmek için kaynaklarını yeniden şekillendirebilmelidirler (Korpela ve diğ., 2001). Tüm bu bilgiler ışığında tedarik zinciri yönetimi problemlerinden dağıtım aşamasının tez çalışmasında uygulama olarak seçilmesi kararlaştırılmıştır. Müşteriye hızlı yanıt vermek, istenen zamanlarda, istenen kalitede ve miktarlarda ürün teslimatı gerçekleştirmek büyük önem taşımaktadır. Dağıtım aşamasının altında yer alan “araç rotalama” konusu literatürde ve gerçek hayatta sıklıkla ele alınan bir konu olmuştur. Taşıdığı önem sebebiyle belirlenen araç rotalama problemi çeşidinde değişken komşuluk arama yaklaşımı uygulanmıştır.

2.5. ARAÇ ROTALAMA PROBLEMİ

Araç rotalama probleminin (ARP), (vehicle routing problem (VRP)) tanımı yapıldıktan ve tercih edilme sebebine değinildikten sonra literatürde karşılaşılan araç rotalama

problemi çeşitleri ile uygulanmış çözüm yöntemleri üzerinde durulmuştur. Araç rotalama problemlerine özgü komşuluk yapılarına değinilmiştir.

Tedarik zinciri yönetimindeki dağıtım aşaması taşıma maliyetleri, stok maliyetleri gibi sebeplerle önemli bir maliyet kalemidir. Taşımanın, lojistik işlemlerin bir parçasını oluşturduğu bir önceki bölümde ifade edilmişti. Tedarik zinciri yönetiminde müşteriye hızlı yanıt vermek, istenen zamanlarda, istenen kalitede ve miktarlarda ürün teslimatı gerçekleştirmek büyük önem taşımaktadır. Dağıtım aşamasının altında yer alan araç rotalama konusu literatürde ve gerçek hayatta sıklıkla ele alınan bir konu olmuştur. Gencer ve Yaşa (2007) araç rotalama problemlerinin gezgin satıcı problemlerinin genel hali olduğunu ifade etmişlerdir. Araç rotalama problemleri kombinatoriyal optimizasyon problemleri sınıfına dahil edilmektedirler. Araç rotalama problemleri detaylı şekilde ele alınmadan önce graf yapısının ve gezgin satıcı problemlerinin kısaca tanımakta fayda vardır.

2.5.1. Graf Yapılar

Rifat Çölkesen'in 2002 yılında yazdığı kitabında graf, bir olay veya ifadenin düğüm ve çizgiler kullanılarak gösterilmesi şekli olarak tanımlanmıştır. Geçmiş 18. yüzyıla kadar dayanmasına rağmen graf günümüzdeki birçok bilgisayar uygulama alanında ve modellemede sıklıkla kullanılmaktadır (Çölkesen, 2002).

Graf teorisi, 1736 yılında İsviçreli matematikçi Leonhard Euler'in, Königsberg'in "Yedi Köprü" problemini baz alarak kaleme aldığı makalesinin yayımlanması ile ortaya çıkmıştır (Çolakoğlu, 2010). Çok sayıda probleme ait durum uzayının veya çözüm ağacının gösterilmesinde graf yapıları kullanılmaktadır. Graf kelimesi ilk olarak 1822 yılında J. J. Sylvester tarafından kullanılmıştır. Graf, bir düğümler kümesi ile bu düğümler arasındaki ilişkileri ifade eden kenarlar yardımıyla tanımlanan bir yapıdır. Her kenar iki düğümü birleştirmektedir ve birbiriyle kenarla ilişkisi olan düğümler "komşu düğümler" olarak adlandırılmaktadır (Nabiyev, 2010).

Graflar, çok sayıda gerçek yaşam problemlerinin analizi için matematiksel modeller kurulmasını sağlamaktadır. Fizik, kimya, iletişim bilimi, bilgisayar teknolojisi, genetik,

psikoloji, sosyoloji ve dilsel yapılar graf teorisiyle problem olarak modellenebilmektedirler (Balakrishnan ve Ranganathan, 1999).

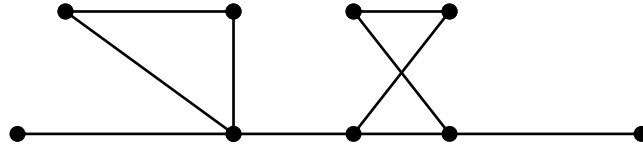
Balakrishnan ve Ranganathan (1999) graf yapısının anlaşılması için temel bilgilerde ilk olarak sokakları gösteren bir yol ağı örneğini incelemiştir. Buradan da anlaşılacağı gibi graf yapılar rotalama problemlerinin modellenmesi için sıklıkla kullanılmaktadırlar.

Matematiksel anlamda, graf, düğümler ve bu düğümler arasındaki ilişkiyi gösteren kenarlardan oluşan bir kümedir. Bir grafta n düğüm ve m kenar bulunmakta ise, graf kümesi (G), düğüm (D) ve kenar (K) kümelerinin elamanlarının ilişkilendirilmesiyle yapılmaktadır (Çölkesen, 2002).

$$D = \{ d_0, d_1, d_2, \dots, d_{n-1}, d_n \}$$

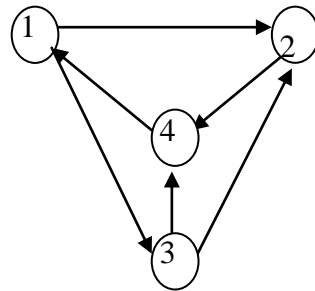
$$K = \{ k_0, k_1, k_2, \dots, k_{n-1}, k_n \}$$

$$G = \{ D, K \}$$



Şekil 2.27: Basit bir graf. (Balakrishnan ve Ranganathan, 1999)

Şekil 2.27’de basit bir graf yapısı örneği verilmiştir. Eğer graf üzerinde bulunan kenarlar bir düğümden diğer düğüme giden yönleri göstermekteyse bu graflara yönlü graf denilmektedir (Çolakoğlu, 2010). Şekil 2.28’de yönlü graf örneği gösterilmiştir.



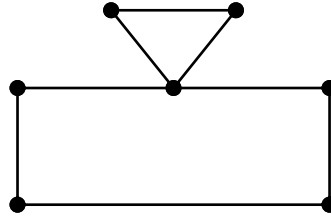
Şekil 2.28: 4 düğümlü yönlü bir graf. (Balakrishnan, 1997)

2.5.1.1. Hamilton Grafi

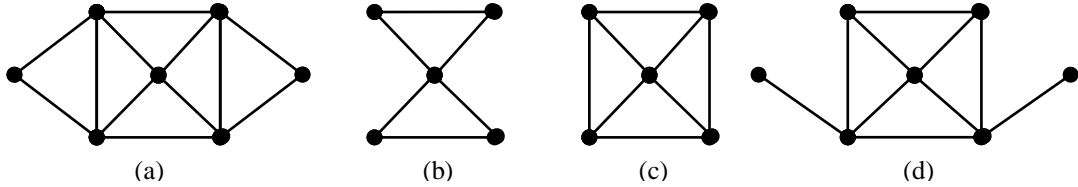
Bir graf yapıda iki düğüm arasındaki yol, eğer graftaki her düğümden geçiyorsa bu yol Hamilton yolu olarak adlandırılmaktadır. Kapalı bir yol olması halinde Hamilton çevrimi oluşmaktadır (Balakrishnan, 1997).

2.5.1.2. Euler Grafi

Bir kenar üzerinden bir kereden fazla geçmeden tüm kenarlardan geçilerek bir tur oluşturulabiliyorsa bu graf yapısı Euler graftır. Euler grafi üzerine çalışmalar 18. yüzyılda, Hamilton grafi üzerine çalışmalar 19. yüzyılda başlatılmıştır (Balakrishnan ve Ranganathan, 1999).



Şekil 2.29: Euler Grafi. (Balakrishnan ve Ranganathan, 1999)



Şekil 2.30: Euler ve Hamilton grafları.

Şekil 2.30'da (a) hem Euler hem de Hamilton grafi, (b) Euler grafi, (c) Hamilton grafi ve (d) ise ne Euler ne de Hamilton grafidir (Aldous ve Wilson, 2003).

2.5.2. Gezgin Satıcı Problemi

Gezgin satıcı problemi (GSP), bir başlangıç noktasından yola çıkan bir satıcının önceden tanımlanmış şehirleri ziyaret edip başlangıç noktasına tekrar geri döndüğü rotayı, her şehri sadece bir kere ziyaret etmesi halinde katettiği toplam mesafenin minimum kılınması problemidir (Gutin ve Punnen, 2002). Gezgin satıcı problemi, Hamilton çevrimini en kısa mesafeyi katederek oluşturmaya çalışmaktadır.

Gezgin satıcı probleminde, n adet şehir $\{x_1, x_2, \dots, x_n\}$, ikili $\{x_i, x_j\}$ ilişkileri ve bunlar arasındaki $d\{x_i, x_j\}$ şeklinde tanımlanmış uzaklıklar mevcuttur. Buradaki amaç

şehirlerin taleplerini karşılayarak en kısa yolun elde edilmesidir. Bu miktar rotanın uzunluğu olarak tanımlanır. Satıcı başlangıç şehrine geri dönecek şekilde bu rota uzunluğunu veren özel bir permütasyon sırasına göre tüm şehirleri ziyaret etmektedir (Johnson ve diğ., 1995).

Gezgin satıcı probleminde n adet şehir için, $(n-1)!$ olası çözüm vardır (Taha, 2000). Büyük boyutlu problemlerde tüm olası durumların incelenmesi mümkün olmamaktadır. Bunun için doğrusal programlama ile problem modellenmiş ve matematiksel modellerin çözüm sunmasının çok zaman aldığı veya çözüm sunmadığı durumlar için son yıllarda sezgisel yöntemlerle problemin çözümü elde edilmeye çalışılmıştır (Yumurtacı ve diğ., 2008).

2.5.2.1. Gezgin Satıcı Probleminin Doğrusal Gösterimi

n adet şehir için noktalar kümesi $X=\{x_1, x_2, \dots, x_n\}$ ile gösterilirken;

$$\text{Minimum} \quad \sum_{j=1}^n d_j x_j$$

$$\text{Kısıtlar altında} \quad X \in P(F)$$

şeklinde gösterilmektedir. $P(F)$ ile, GSP'i çözmek için doğrusal programlama tekniklerinin kısıt uygulamalarının doğrusal eşitsizlik tanımlamalarıdır (Gutin ve Punnen, 2002).

2.5.2.2. Gezgin Satıcı Probleminin Tamsayı Gösterimi

$$\text{Minimum} \quad \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ij} \quad (2.1)$$

Kısıtlar altında:

$$\sum_{i=1}^n x_{ij} = 1, \quad j=1, \dots, n, \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i=1, \dots, n, \quad (2.3)$$

$$x_{ij} = 0 \text{ veya } 1$$

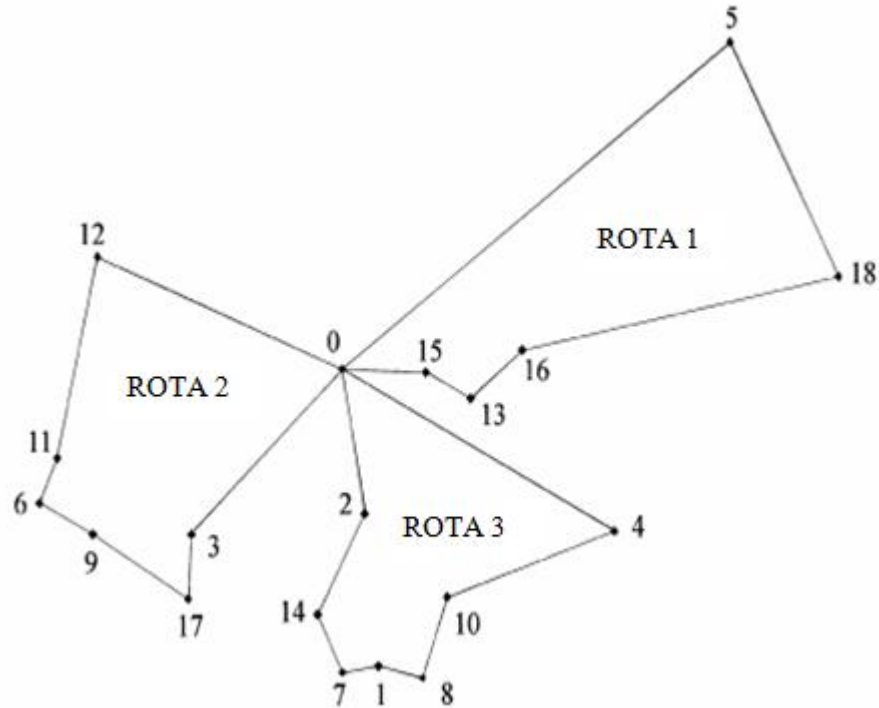
Herhangi bir $X = (x_{ij})_{n \times n}$ için, $S = \{(ij) : x_{ij} = 1\}$ seti göz önüne alınır. Burada, S alt turların bir toplamını veya bir turu göstermektedir (Gutin ve Punnen, 2002).

2.5.3. Araç Rotalama Problemi (ARP)'nin Önemi

Araç rotalama problemi; ilk olarak Dantzig ve Ramser tarafından 1959 yılında ele alınmasından bu yana geçen 53 yıllık sürede hep ele alınan bir konu olmuştur. Toth ve Vigo (2002) kitaplarında, klasik kombinatoryel optimizasyon problemlerinden olan araç rotalama problemlerinin taşıdığı ekonomik öneme aşağıdaki ifadeyle yer vermişlerdir:

“Hem Kuzey Amerika hem de Avrupa'daki çok sayıda gerçek-dünya uygulamaları göstermiştir ki; dağıtım süreci planlaması için bilgisayarlı prosedürlerin kullanılması küresel taşıma maliyetlerinde önemli tasarruflar (genellikle % 5'den % 20'ye kadar) sağlamaktadır. Bu da küresel ekonomik sistem üzerindeki söz konusu tasarrufların etkisinin önemli olduğunu görmeyi kolaylaştırmaktadır. Aslında, ulaştırma süreci, üretim ve dağıtım sistemlerinin tüm aşamalarını içermektedir ve malların nihai maliyetiyle (genellikle %10 'dan % 20'ye) ilgili bir bileşeni temsil eder.”

Şekil 2.31'de araç rotalama probleminin genel bir görünümü sunulmuştur.



Şekil 2.31: Araç rotalaması genel görünümü (Salvendy, 2001)

Jesper Larsen (1999) tez çalışmasında, Lawrence Bodin ve diğ.'nin 1980 yılında raporladıkları dağıtım sistemleri üzerine yapılan harcamaya ait ifadelere yer vermiştir. Buna göre ABD'de yıllık dağıtım maliyeti 400 milyar dolar ve İngiltere'nin yıllık dağıtım harcamasının 15 milyar sterlin olarak tahmin edildiği belirtilmiştir. Ayrıca, Marshall Fisher'in 1997 yılında yazdığı Fiziksel Dağıtım Ulusal Konseyi'nden bir çalışmada da ABD gayri safi milli hasılasının %15'ini taşıma maliyetlerinin oluşturduğu belirtilmiştir. Danimarka Ulaştırma Bakanlığı'nın 1998 yılında yayınladığı bir raporda da söz konusu maliyetlerin 1981 yılında %13 ve 1994 yılında %15 olduğu ifade edilmiştir (Larsen, 1999).

2.5.4. Araç Rotalama Probleminin Tarihsel Gelişimi

Bu başlık altında Eksioglu ve diğ.'nin (2009) ARP üzerine incelemeleri göz önüne alınmıştır. ARP literatüründe ilk kayıt olarak 1954 yılında Dantzig, Fulkerson ve Johnson'un yayınları geçmektedir. Bu çalışmada büyük ölçekli GSP çalışılmış ve bir çözüm metodu sunulmuştur. Bu çalışmayı geniş bir ölçüde diğer GSP yayınları takip etmiştir. 1964 yılında Clarke ve Wright, ilk olarak problem formülasyonunda birden fazla aracı ifade etmişlerdir. Bu yüzden, bu yayın aslında ARP literatüründe "ilk" olarak kabul edilebilir. 1972 yılında "araç rotalama" ifadesini başlığında kullanan ilk çalışma Golden ve diğ.'nin çalışmaları olmuştur. 1970'li yılların başında diğer ARP çeşitleri ortaya çıkmaya başlamıştır. Bunlara örnek olarak filo rotalama, talebe bağlı servis sistemi (dial-a-bus systems), ulaştırma ağ tasarımı, kamu servis araçları rotalama, dağıtım yönetimi ve katı atık toplama verilebilir. ARP'de olasılık kavramını ilk olarak 1978 yılında Golden ve Stewart tarafından sunulmuştur. ARP'nin en bilindik çeşitlerinden olan zaman pencereci ARP, Solomon tarafından 1983 yılında klasik ARP'ne zaman-penceresi kısıtı ilave edilmesiyle gündeme gelmiş ve en iyi bilinen karşılaştırma problem seti olarak "Solomon örnekleri" sunulmuştur. 1980'ler boyunca, orijinal problemin farklı statik biçimlerinde ARP çalışmaları yürütülmüştür. Hesaplama karmaşıklığı ve mikrobilgisayarların az bulunması sebepleriyle bu yıllarda ARP'nin olasılıklı, dinamik ve bulanık çeşitleri pek fazla çalışılmamıştır. 1990'lı yıllarda ARP araştırmaları hızlanmıştır. Mikrobilgisayarlara daha kolay ulaşılabilmesi ve yetenekleriyle araştırmacılar daha karışık arama algoritmaları geliştirebilmişlerdir. Bu dönemde metasezgisel terimi ARP ve diğer kombinatoriyel problemlerin çözümü için ortaya çıkmıştır.

Bu tezde ARP'nin kısa olarak belirtilen tarihsel gelişiminde bahsedilen yöntemler ve özellikle metasezgisel yaklaşımlar, ARP çözüm yöntemleri başlığı altında ele alınmıştır.

2.5.5. Araç Rotalama Problemlerinin Çeşitleri

Araç rotalama üzerine yapılan çoğu literatür çalışması aşağıdaki özellikleri taşıyan problemlerle ilişkilidir (Caccetta ve Hill, 2001):

- i) Tek bir ürün, tek bir depodan, müşterilere bilinen taleplerle dağıtılmaktadır.
- ii) Her müşterinin talebi bir araç tarafından sunulmaktadır.
- iii) Her araç aynı kapasiteye sahiptir ve tek bir tur yapmaktadır.
- iv) Her aracın katettiği toplam mesafede aracın belirli kapasitesi aşılamamaktadır.
- v) Her müşteriye belirli bir zaman aralığında (penceresinde) hizmet sunulmalıdır.
- vi) Amaç, tüm araçlar tarafından kat edilen toplam mesafeyi minimum kılmaktır.

Gevşeklik kısıtlamaları (iv) ve (v), kapasiteli araç rotalama problemi olarak adlandırılan en fazla çalışılan araç rotalama problemi çeşidine yol açmaktadır. Caccetta ve Hill'in belirttiği maddeler doğrultusunda, bir araç rotalama probleminin ele alınabilmesi için talep, uzaklık (maliyet), araç çeşit ve kapasitesi, zaman penceresi gibi bazı bilgilerin bilinmesi gerektiği ifade edilebilir.

ARP'nin ana amacı malzeme dağıtım maliyetinin minimize edilmesidir. ARP literatürde genellikle zaman pencereli araç rotalama problemleri, kapasiteli araç rotalama problemleri, depo sayısına göre çok depolu veya tek depolu araç rotalama problemleri, talep yapısına göre stokastik veya deterministik talepli araç rotalama problemleri başlıkları altında ele alınmıştır (Pisinger ve Ropke, 2007). Araç rotalama problemi karmaşık yapıda ve çözümü uzun süren bir problem olduğundan çözümü için çoğunlukla sezgisel yöntemler kullanılmıştır (Jozefowicz ve diğ., 2008a). Araç rotalama problemlerinin çeşitlerinin ifade edilmesinde gözönüne alınan durumlar Jozefowicz ve diğ.'nin (2008b) çalışmasındaki ifadeler esas alınarak tablo haline getirilmiş ve Tablo 2.1'de gösterilmiştir.

$$\text{Amaç fonksiyonu: } \text{Min.} \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot X_{ij} \quad (2.4)$$

Kısıtlar:

$$\sum_{i,i \neq j} X_{ij} = 1 \quad (2.5)$$

$$\sum_{j,j \neq i} X_{ij} = 1 \quad (2.6)$$

$$\sum_i \sum_j D_i \cdot X_{ij} \leq Q \quad (2.7)$$

$$\sum_{j=1}^n X_{0j} = K \quad (2.8)$$

$$\sum_{i=1}^n X_{i0} = K \quad (2.9)$$

$$X_{ij} \in \{0,1\} \quad (2.10)$$

d_{ij} : i ve j noktaları arası mesafe.

X_{ij} : i den hemen sonra j noktasına gidilecekse 1; değilse 0'dır.

Q : Araç kapasitesi.

D_i : i . müşterinin toplam ürün talebidir.

K : Araç sayısıdır.

i : 0,1.....n

j : 0,1.....n

(2.4) numaralı denklemde toplam katedilen yolun minimum olması amacı ifade edilmiştir. i noktasından j noktasına gidişin olması durumunda 1 değerini alan X_{ij} 'ler ile i - j arası mesafeyi belirten d_{ij} değerlerinin çarpımları toplamı toplam mesafeyi verecektir. (2.5) numaralı kısıtta bir noktaya gelen yolların sadece bir tane olması gerektiği ve (2.6) numaralı kısıtta ise bir noktadan sadece tek bir noktaya gidiş olması gerektiği ifade edilmektedir. (2.7) numaralı denklem bir turdaki ziyaret edilen noktaların toplam talebinin araç kapasitesini aşmaması gerektiğini göstermektedir. (2.8) ve (2.9) numaralı kısıtlarda da depodan çıkan ve depoya gelen noktaların araç sayısına eşit olması gerektiği ifade edilmiştir.

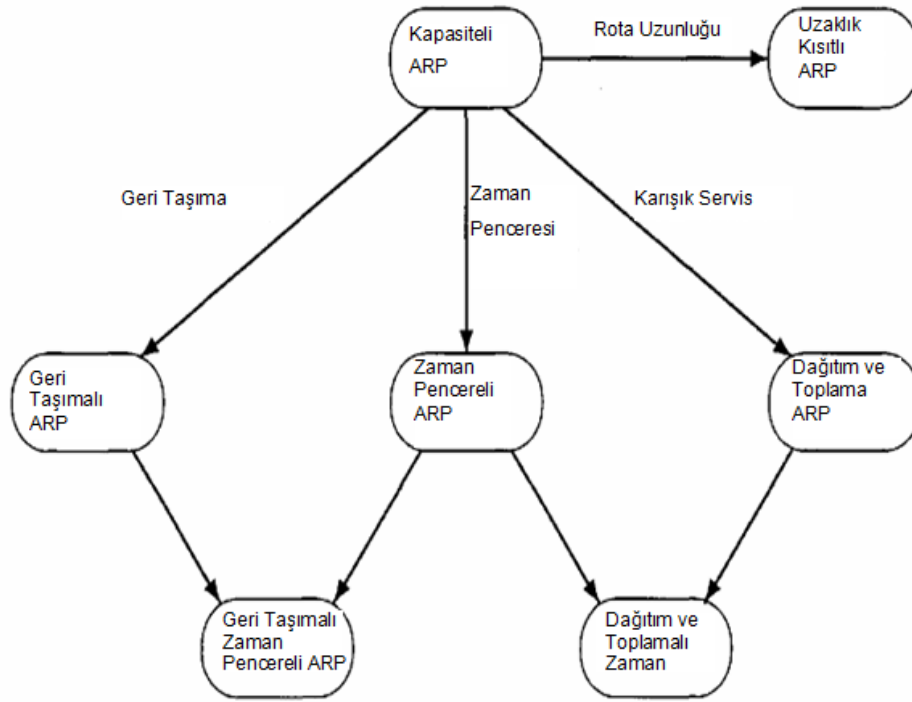
2.5.5.5. Periyodik Araç Rotalama Problemi

Bu problem türünde müşteriler belirli aralıklarla, yani frekans değerleri ile teslimatın gerçekleşmesini talep etmektedirler. Örneğin haftada iki kere birer gün arayla, ... gibi.

2.5.5.6. Topla-Dağıt Araç Rotalama Problemi

Kapasiteli araç rotalama problemlerinde söz konusu araçlar rota içerisinde sadece müşterilere teslimat gerçekleştirmektedir. Bazı rotalarda dağıtım yaparken aynı zamanda toplama da yapma gereği oluşabilmektedir. Aracın dağıtırken artan boş alanı, toplama yaparken dolmaktadır.

2.5.6. Araç Rotalamanın Temel Problemleri ve Aralarındaki İlişki



Şekil 2.33: Araç rotalamanın temel problemleri aralarındaki ilişki (Toth ve Vigo, 2002)

Şekil 2.33'te zaman penceresi, rota uzunluğu, geri taşıma ve karışık servis durumlarına göre ARP çeşitleri, aralarındaki ilişkiyi gösteren ok yönleriyle gösterilmiştir.

2.5.7. Araç Rotalama Problemleri İçin Çözüm Yöntemleri

GSP ve ARP çözülmesi zor problemlerdendir. Hwang (1996) simetrik GSP için alternatif çözümleri gösteren tur sayılarına ait bir tablo sunmuştur (Tablo 2.2). Burada da görüldüğü gibi 70 şehirlik bir problemde 96 basamaklı çözüm alternafi bulunmaktadır. Tüm çözümleri bulması istenen bir program çalıştırılrsa da, programın çalışmasının aylar, nokta sayısı arttıkça da yıllar bile sürebileceği tahmin edilmektedir.

Tablo 2.2: GSP için tur sayılarının hesaplanması (Hwang, 1996)

Şehir sayısı	Tur sayısı
n	$(n-1)!/2$
5	1.2×10^1
10	1.8×10^5
15	4.4×10^{10}
20	6.1×10^{16}
25	3.1×10^{23}
30	4.4×10^{30}
35	1.5×10^{38}
40	1.0×10^{46}
45	1.3×10^{54}
50	3.0×10^{62}
55	1.2×10^{71}
60	6.9×10^{79}
65	6.3×10^{88}
70	8.6×10^{97}

2.5.8. Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri en iyi çözümü tüm çözüm alternatiflerinin listesini içererek elde etmektedir (Marinakis ve diğ., 2007).

2.5.8.1. Tam Sayılı Doğrusal Programlama

Tam sayılı doğrusal programlama (TDP), değişkenlerden bazılarının veya hepsinin tam sayılı (ya da kesikli) değerler aldığı bir doğrusal programlama problemidir (Taha, 2000). Araç rotalama problemi tam sayılı olarak modellenebilmektedir. Tam sayılı

doğrusal programlamada kullanılan başlıca çözüm yöntemleri dal-sınır ve kesme düzlemi algoritmalarıdır. Dal-sınır yöntemi GSP ve ARP'de de kullanılmaktadır.

2.5.8.2. Dal-Sınır Yöntemi

Eastman 1958 yılında ilk dal-sınır algoritmasını geliştirmiştir (Laporte, 2006). Dal-sınır yöntemi kapasiteli araç rotalama problemleri ve versiyonları için uygulanabilmektedir. Toth ve Vigo (2004) 1980'lerin sonuna kadar kapasiteli araç rotalama problemleri için en etkili kesin çözüm yönteminin dal-sınır algoritmalarının olduğunu belirtmişlerdir. 1981 yılında ARP'de dal-sınır yöntemini Christofides ve diğ. uygulamıştır (Laporte, 2009). Dal-sınır yönteminin uygulanması için başlangıç optimum çözümünün bulunmuş olması gerekmektedir. Bu optimum çözüm tamsayılı olmadığından dal-sınır algoritması aracılığı ile tamsayılı optimum çözüm belirleninceye kadar çözüm uzayının düzenlenmesine devam edilmektedir. Dal-Sınır'da ilk olarak çözümdeki tamsayı değeri almamış tam sayı olması istenen bir değişken (x_i), keyfi olarak seçilir ve bu değer *küçük tam sayılı kısım* $\langle x_i \rangle$ ve *büyük tam sayılı kısım* olarak iki alana ayrılır, bunun sebebi *küçük tam sayılı kısım* $\langle x_i \rangle < x_i < \langle x_i \rangle + 1$ bölgesinde x_i 'in tamsayılı değerleri olmamalıdır. (Örneğin $x_i=3,21$ için *küçük tam sayılı kısım*=3 ve *büyük tam sayılı kısım*=4 olmaktadır) Uygun kısıtlar yazarak tamsayılı çözümler içermeyen bölgeleri çıkarmak suretiyle devam edildiğinde, tamsayı kısıtlarını sağlayan optimum bir uç noktaya ulaşılabilecektir. Aslında burada yapılan şey, arka arkaya doğrusal programlama problemi çözerek tam sayılı doğrusal programlama problemini çözmektir (Taha, 2000).

2.5.8.3. Kesme Düzlemi Algoritması

Kesme düzlemi algoritmasında da işlemlere dal-sınır algoritmasındaki gibi sürekli bir doğrusal programlama probleminin optimum çözümü ile başlamaktadır. Fakat bu yöntemde dallanma ve sınırlamadan çok, kesme adı verilen özel kısıtlar ardı ardına oluşturularak çözüm uzayının düzenlenmesine gidilmektedir (Taha, 2000).

2.5.8.4. Minimum K-ağaç

K -ağaç yöntemi; verilmiş olan bir $n+1$ düğümlü G grafını kapsayan $n+K$ kenar seti olarak tanımlanmaktadır (Fisher, 1994). Fisher (1994) K-ağaç yapısında; c_{ij} ile $i-j$ noktaları arasındaki maliyeti, $c(e)$ ile de e bağlantısının maliyetini göstermiştir.

$S \subseteq E$ olmak üzere;

$$c(S) = \sum_{e \in S} c(e) \quad (2.11)$$

$$\sum_{i=1}^n X_{0j} = 2K \text{ depoya göre } 2K \text{ derecesi ile } K\text{-ağaç maliyetini veren düğüm seti}$$

olarak gösterilmiştir.

2.5.8.5. Dinamik Programlama

1981 yılında dinamik programlamayı Christofides ve diğ. uygulamışlardır (Laporte, 2009). Dinamik programlama, ‘ n ’ değişkenli bir problemin optimum çözümünü, problemi ‘ n ’ aşamaya ayrıştırarak ve her aşamada tek değişkenli bir altproblemi çözerek belirlemektedir. Böylece n değişkenli altproblemler yerine tek değişkenli alt problemleri optimum kılarak hesaplama avantajı sağlamaktadır. Dinamik programlamanın asıl katkısını, problemleri aşamalara ayrıştırmasının çerçevesini oluşturan optimumluk ilkesi sağlamaktadır. Optimizasyon problemlerine bağlı olarak aşamaların yapısı da farklılıklar göstermektedir. Bu yüzden dinamik programlamada her bir aşamayı optimum kılmak için gereken hesaplamaların ayrıntısı verilmemektedir. Bu ayrıntılar problem çözücü tarafından doğaçlama olarak gerçekleştirilip tasarlanmaktadır. Dinamik programlamada hesaplamalar yinelenerek yapılmaktadır ve böylece bir alt problemin “optimum çözümü” bir sonraki alt problem için “girdi”yi oluşturmaktadır. Son altproblem çözüldüğünde ise, problemin tamamı için bir optimum çözüm elde edilmiş olur. Alt problemler bazı ortak kısıtlarla birbirleriyle ilişkilendirilmişler ve çözüm aşamalarında bir alt problemden bir sonrakine ilerledikçe bu kısıtların uygunluğuna dikkat etmek zorunludur (Taha, 2000).

2.5.9. Sezgisel Yöntemler

Araç rotalama için literatürde birkaç sezgisel yöntem rastlanmıştır. Bu sezgiseller iki sınıfa ayrılmaktadır. İlki 1960 ve 1990 yılları arasında geliştirilen “klasik sezgisellerdir”. Diğeri ise son 20 yılda geliştirilen metasezgisellerdir (Toth ve Vigo, 2002). Bu çalışmada metasezgiseller ayrı bir başlık altında ele alınmıştır.

1960 ve 1970’li yıllarda araç rotalama problemlerini çözmek için klasik sezgiseller; yapıcı sezgiseller, iki-aşamalı sezgiseller ve geliştirici sezgiseller olarak üç kategoriye bölünmüşlerdir (Toth ve Vigo, 2002; Marinakis ve diğ., 2007). Marinakis ve diğ., (2007) Clarke ve Wright’s Saving Algoritmasını, Fisher and Jaikumar Algoritmasını ve başka birkaç algoritmayı 1980’lerde problemin çözümü için sunulan matematik programlama prosedürleri olarak saymıştır. Toth ve Vigo’nun (2002) ele aldığı klasik

sezgisel çözüm yöntemlerinden bazıları; yapısal sezgiseller grubu altında Clarke ve Wright Saving Algoritması ile geliştirilmiş türler, iki aşamalı yöntemler grubu altında tek rota ve çok rota iyileştirmeli sezgisel algoritmalarıdır. Literatürde ayrıca en kısa yol yöntemi yapıcı sezgiseller grubunda, Fisher ve Jaikumar algoritması ile süpürme algoritması da iki aşamalı sezgiseller grubunda bulunmaktadır. Clarke & Wright, Fisher & Jaikumar ve süpürme algoritmaları popülerliklerinden dolayı en iyi klasik sezgisel yöntemler olarak bilinmektedirler (Cordeu ve diğ., 2002)

2.5.9.1. En Kısa Yol Problemi

En kısa yol probleminde, ulaştırma şebekesindeki kaynak ile varış noktası arasındaki en kısa yol belirlenmektedir (Taha, 2000).

2.5.9.2. Tasarruf Algoritması - Clarke ve Wright Algoritması

Clarke ve Wright (1964) çalışmalarında, iyi veya en iyiye yakın rotanın rassal seçimini sağlayan tekrarlı bir süreç geliştirmişlerdir.

Clarke ve Wright Saving Algoritması'na, P_i dağıtım noktalarının (P_0 depo olmak üzere) birbirleri ile olan uzaklık matrisinin hazırlanması ile başlanmaktadır. $d_{y,z}$ ile y ve z noktaları arasındaki uzaklık gösterilmek üzere, tasarruflar $(d_{0,y} + d_{0,z} - d_{y,z})$ formülü ile hesaplanmakta ve uzaklık-tasarruf matrisi oluşturulmaktadır. Daha sonra en yüksek tasarruftan başlanarak, araç kapasite kısıtı aşılmadıkça, noktalar birleştirilmektedir (Sule, 2001).

2.5.9.3. Fisher ve Jaikumar (FJ) Algoritması

Fisher ve Jaikumar tarafından 1981 yılında geliştirilen iki aşamalı algoritmanın ilk aşamasında müşteri grupları Genel Atama Problemi çözümü ile oluşturulmaktadır. İkinci aşamada ise oluşturulan gruplara gezgin satıcı problemi çözüm teknikleri uygulanarak rotalar belirlenmektedir (Cordeau ve diğ., 2002).

2.5.9.4. Süpürme Algoritması

İki aşamalı çözüm yöntemlerinden olan süpürme algoritması (SA), 1974 yılında Gillet ve Miller tarafından geliştirilmiştir (Toth ve Vigo, 2002). Bu yöntemin ilk olarak Wren ve Holliday tarafından ortaya atılmış olmasına rağmen, yaygınlaşması Gillet ve Miller tarafından sağlanmıştır. Bu yöntem iki aşamalı olarak ele alınmaktadır. İlk aşamada yaklaşık rotalar belirlenmekte ve ikinci aşamada rotaları geliştirmek için adaptasyon yapılmaktadır. Müşterilerin tümünden gelmiş olan taleplerin toplamının araç

kapasitesine oranlanması ile minimum gereken araç sayısı bulunmaktadır (Sule, 2001). Süpürme algoritmasında, ARP'nin düzlemsel mesafesi uygulanmaktadır. Uygun rotalar, deponun merkez alındığı bir doğrunun döndürülmesiyle ve bu döndürme esnasında, araç kapasite veya mesafe kısıtı aşılanaya kadar derece derece müşterilerin eklenmesiyle oluşturulmaktadır. Kısıt aşıldığı zaman yeni rotaya geçiş yapılmaktadır (Cordeu ve diğ., 2002). Standart halinde 'slack' (gevşeklik) diye adlandırılan değişken kullanılmayan SA'nın gevşeklik 'slack' kullanımının söz konusu olduğu uygulamaları da mevcuttur. SA'nda önce minimum araç sayısı belirlenmekte ve daha sonra depodan başlayarak bir "dönme" işlemi sırasında kısıtlar doğrultusunda rotalar oluşturulmaktadır (Yumurtacı ve diğ., 2008). Ardından, müşteriler ve depodan oluşan noktaların x-y koordinatları belirlenir. Depo koordinatı (0,0) alınarak diğer noktaların koordinatları buna göre yeniden hesaplanırlar. Koni ucu depo olacak şekilde yol üzerindeki müşterileri kapsayan koni çizilir ve depoya daha yakın olan yerin katılmasıyla elde edilmektedir. Kapasite kısıtı aşılmadıkça çizginin geçtiği noktalar rotaya dahil edilmektedir (Sule, 2001)

Tablo 2.3'te Cordeu ve diğ.'nin (2002) sezgisel yöntemleri kıyaslaması sunulmuştur.

Tablo 2.3: Sezgisel yöntemlerin kıyaslanması tablosu.

Klasik sezgiseller	Doğruluk	Hız	Basitlik	Esneklik
Clarke ve Wright (CW)	Düşük	Çok yüksek	Çok yüksek	Düşük
İki kıyaslama tabanlı yöntem	Yüksek	Çok düşük	Düşük	Düşük
Süpürme	Düşük	Orta-yüksek	Yüksek	Düşük
1-Petal	Düşük	Yüksek	Orta	Orta
2-Petal	Orta	Orta	Orta	Orta
Fisher ve Jaikumar (FJ)	Değerlendirmek zor	Orta	Düşük	Düşük
Yer seçimi tabanlı FJ, Bramel ve Simchi-Levi tarafından	Orta	Düşük	Düşük	Düşük

2.5.10. Metasezgisel Yöntemler

Toth ve Vigo (2004) araç rotalama problemlerinde uygulaması olan altı metasezgisel yöntemden bahsetmiştir. Bunlar; tavlama benzetimi, deterministik tavlama, tabu arama, genetik algoritma, karınca sistemleri, genetik algoritmalar ve yapay sinir ağlarıdır.

Tavlama benzetimi ve tabu arama metasezgisel yöntemleri yerel aramanın anlatıldığı bölümde ele alınmıştır. Bu yüzden bu başlık altında sadece ARP uygulamaları hakkında bilgi verilmiştir.

2.5.10.1. Tavlama Benzetimi

Tavlama Benzetimi'ni kapasiteli araç rotalama problemlerinde 1993 yılında Dueck, 1993 yılında Osman, 2001 yılında Breedam ve 2002 yılında Tarantilis ve diğ. uygulamış, zaman pencereli araç rotalama problemlerinde 1996 yılında Chiang ve Russell, 2001 yılında Arbelaitz ve diğ., 2002 yılında Czech ile Czarnas uygulamıştır (Marinakis ve diğ., 2007).

2.5.10.2. Tabu Arama

Tabu Arama'yı kapasiteli ARP'nde 1993 yılında Osman, yine 1993 yılında Taillard, 1994 yılında Gendreau ve diğ., 1996 yılında Rego ve Roucairol, 1999 yılında Barbarosoğlu ve Özgür, 2001 yılında Rego, 2003 yılında Toth ve Vigo uygulamış, zaman pencereli araç rotalama problemlerinde 1997 yılında Badeau ve diğ., 1997 yılında Taillard ve diğ., 1998 yılında Brandao, 1999 yılında Homberger ve Gehring, 2001 yılında Braysy ve Gendreau, yine 2001 yılında Cordeau ve diğ., 2004 yılında Li ve diğ., yine 2004 yılında Lim ve Wang uygulamıştır (Marinakis ve diğ., 2007).

2.5.10.3. Yapay Sinir Ağları

İnsan beyni üzerine çalışmalar binlerce yıllıktır. Elektroniğin gelişmesiyle birlikte, bu düşünce sürecini kullanmayı denemek de mümkün olmuştur. Yapay sinir ağlarına doğru ilk adım 1943 yılında bir nörofizyolog olan Warren McCulloch'in ve ve genç bir matematikçi olan Walter Pitts'in nöronların nasıl çalıştığına dair bir yazı yazmaları ile atılmıştır. McCulloch ve Pitts, elektrik devreleri ile basit bir sinir ağı modellemiştir (Anderson ve McNeill, 1992).

Yapay sinir ağları, verideki ilginç modelleri veya yapıları tanımlamak için yardımcı modellerinin geliştirilmesi ve veri incelenmesi için basit hesaplama araçlarıdır. Bu

modelleri geliřtirmek için kullanılan veriler, eğitim verileri olarak bilinmektedir. Bir kere sinir ağı eğitim verilerine maruz kaldıktan ve bu verilerdeki mevcut modelleri öğrendikten sonra, çeřitli sonuçlar elde ederek yeni veriler için uygulanabilmektedir (Smith ve Gupta, 2002). Smith ve Gupta (2002) yapay sinir ağlarının kullanımını řu şekilde belirtmiřtir:

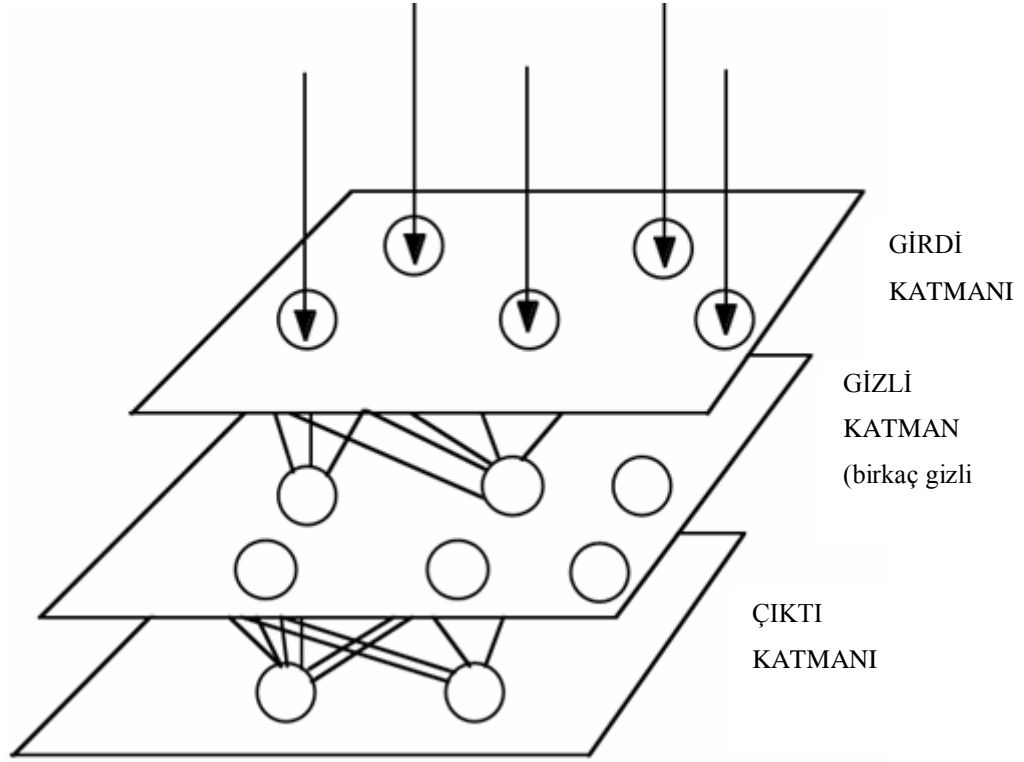
- gemiř eğitim verilerinde gözlenen modellere dayalı gelecekteki olayları tahmin etmeyi öğrenmek;
- eğitim verilerinde gözlenen özelliklere dayanan önceden tanımlanmış gruplara görünmeyen verileri sınıflandırmayı öğrenmek;
- eğitim verilerindeki özelliklere benzerliğe dayalı doğal grupta eğitim verilerini kümelemeyi öğrenmek.

Buradan da görüldüğü gibi Yapay Sinir Ağları genel olarak tahminlemede, sınıflandırmada veya kümelemede kullanılmaktadır. Anderson ve McNeill (1992) Yapay Sinir Ağları hesaplama yaklaşımlarının kıyaslamasını Tablo 2.4'te sunmuşlardır.

Tablo 2.4: YSA hesaplama yaklaşımlarının kıyaslama tablosu

Karakteristikleri	Geleneksel hesaplama (Uzman sistemler dahil)	Yapay sinir ağları
İřlem stili	Sıralı	Paralel
Fonksiyonlar	Mantıksal (sol beyin) -kurallar -kavramlar -hesaplama yoluyla	Biim (sağ beyin) -görüntüler -resimler -kontroller yoluyla
Öğrenme yöntemi	Kurallar ile (didaktik)	Örnekler ile (Sokratik)
Uygulamalar	Muhasebe, kelime işleme, matematik, stok, dijital iletiřim	Sensör işleme, ses tanıma, desen tanıma, metin tanıma

Şekil 2.34'te Anderson ve McNeill'in (1992) yapay sinir ağı gösterimine yer verilmiştir.



Şekil 2.34: Basit bir yapay sinir ağı diyagramı.

2.5.10.4. Karınca Kolonisi Optimizasyonu

Popülasyon tabanlı bir metasezgisel olan karınca kolonisi optimizasyonu, ilk olarak Marco Dorigo tarafından 1991 yılında önerilmiştir. Gerçek karınca kolonilerinin yiyecek bulmak için kullandıkları yöntem ve davranışlar temel alınarak algoritma oluşturulmuştur. Literatürde üzerinde oldukça fazla çalışılmış olan gezgin satıcı problemi, bütün karınca kolonisi optimizasyonu algoritmalarının gelişiminde önemli bir role sahip olmuş ve birçok karınca kolonisi optimizasyonu algoritması gezgin satıcı probleminin çözümü için geliştirilmiştir (Nabiyev, 2010). Karıncaların yiyecek bulmasındaki temel eleman feromon maddesidir. Karıncalar ilerledikleri yola belirli miktarda feromon bırakmakta ve gidilecek yönün seçiminde feromon miktarı önem taşımaktadır. Karıncalar daha yüksek bir olasılıkla feromonun daha yoğun olduğu yolu, daha az olduğu yola tercih etmektedirler. Bu şekilde hareket eden karıncalar yiyeceğe olan en kısa yolu bulmaktadırlar. Dorigo ve arkadaşları çalışmalarında karınca kolonilerinin davranışlarını tam olarak modellemek yerine, gerçek karıncaların davranışlarından bazı farklılıklara sahip yapay karınca kolonileri sunmuşlardır.

(Karaboğa, 2011). Karınca kolonisi optimizasyonunu kapasiteli araç rotalama problemlerinde 1997 ve 1999 yıllarında Bullnheimer ve diğ., 2002 yılında Doemer ve diğ., 2004 yılında Mazzeo ve Loiseau 2002, 2003 ve 2004 yıllarında Reimann ve diğ. uygulamış, zaman pencereli araç rotalama problemlerinde ise 1999 yılında Gambardella ve diğ. uygulamıştır (Marinakis ve diğ., 2007).

2.5.10.5. Genetik Algoritma

Evrimsel programlamanın bir parçası olan genetik algoritma, Darwin'in evrim teorisinden esinlenerek ortaya çıkmıştır. 1960'lı yıllarda I. Rechenberg'in "Evrimsel Stratejileri" adlı çalışmasıyla gündeme getirilen evrimsel programlama, genetik algoritmaları, evrimleşme stratejisini ve genetik programlamayı kapsamaktadır (Nabiyev, 2010). Karaboğa (2011) ise genetik algoritmaları, evrimsel programlamayı, evrimsel stratejiyi, genetik programlamayı ve diferansiyel gelişim algoritmalarını kapsayan genel terim olarak "evrimsel hesaplama" ifadesini kullanmıştır. Özellikle son 20-30 yılda evrim prensibine dayalı algoritmalara olan ilginin oldukça arttığı da belirtilmiştir. Genetik algoritmayı kapasiteli araç rotalama problemlerinde 2003 yılında Jaskiewicz ve Kominek, Berger ve Mohamed, Baker ve Ayechev, 2004 yılında Alba ve Dorronsoro, 2005 yılında Alba ve diğ., 2006 yılında Alba ve Dorronsoro, Marinakis ve diğ., 2007 yılında Mester ve diğ. uygulamış, zaman pencereli araç rotalama problemlerinde 1991 ve 1993 yıllarında Thangiah ve diğ., 1996 yılında Potvin ve Bengoi, 2001 yılında Tan ve diğ., 2002 yılında Hwang, 2007 yılında da Alvarenga ve diğ. uygulamıştır (Marinakis ve diğ., 2007).

Cordeu ve diğ.'nin (2002) metasezgisel yöntemleri kıyaslaması Tablo 2.5'te sunulmuştur.

Tablo 2.5: Metasezgisel yöntemlerin kıyaslama tablosu.

Metasezgiseller	Doğruluk	Hız	Basitlik	Esneklik
Taburota	Yüksek	Orta	Orta	Yüksek
Taillard	Çok yüksek	Düşük	Orta-düşük	Yüksek
Uyarlanabilir hafıza	Çok yüksek	Düşük	Orta-düşük	Yüksek
Taneli tabu arama	Yüksek	Orta	Orta	Yüksek
Birleşik tabu arama algoritması	Yüksek	Orta	Orta	Yüksek

2.6. ÇOK AMAÇLI PROGRAMLAMA

Karar verme insanları diğer canlılardan ayıran temel özelliklerden birisidir. Karar verme ve analiz, yönetim biliminin önemli bir parçasıdır ve insanlık kadar da eskidir (Farahani ve diğ., 2010). Tek amaçlı kısıtlı optimizasyon probleminin gösterimi şu şekildedir:

$$\text{En iyi} \quad \{f(x) = z\}$$

$$\text{Şu kısıtlar altında} \quad x \in S$$

$f(x)$ amaç fonksiyonu, S ise uygun alandır. Maksimizasyon probleminde S içersinde en büyük amaç fonksiyonunu veren noktayı bulmak istenmektedir. Eğer $f(x)$ doğrusal ve S de lineer kısıtlar ile tanımlanmışsa, tek amaçlı doğrusal programlama probleminden bahsedilmektedir. Eğer $f(x)$ veya S 'i tanımlayan herhangi bir kısıt doğrusal değilse, tek amaçlı doğrusal olmayan programlama probleminden bahsedilmektedir. Tek amaçlı matematiksel programlama problemlerini çözmeye yönelik yöntemler üzerine 1940'lerden beri çalışmalar yapılmaktadır. Tek amaçlı karar verme yöntemleri basit ve erken bir dönemi yansıtmıştır. Fakat dünya daha karmaşık hale gelmiş ve neredeyse tüm önemli gerçek-dünya problemleri birden fazla amaç barındırır olmuştur (Steuer, 1986).

Genel olarak ele alınan problemlerde maksimizasyon veya minimizasyon şeklinde tek amaç kullanılmasına rağmen bir sistemin çok sayıda çelişen amaca sahip olabileceği durumlar da söz konusudur. Özellikle gerçek hayatta karşılaşılan problemlerde birden fazla amacın olduğu durumlar söz konusudur. Bu amaçlar ekonomik, sosyal veya çevresel tabanlı olabilmektedir.

$$\text{maks} \quad \{f_1(x) = z_1\}$$

$$\text{maks} \quad \{f_2(x) = z_2\}$$

$$\text{maks} \quad \{f_3(x) = z_3\}$$

.

.

$$\text{maks} \quad \{f_k(x) = z_k\}$$

$$\text{Şu kısıtlar altında:} \quad x \in S$$

Çok amaçlı programlamanın gösterimi şu şekilde genelleştirilebilir (Jozefowicz ve diğ., 2008a);

$$\begin{array}{l} \text{Çok Amaçlı} \\ \text{Programlama} \end{array} \left\{ \begin{array}{l} F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{şu kısıtlar altında: } x \in D \end{array} \right.$$

n amaç fonksiyonu sayısı olmak üzere, $n \geq 2$

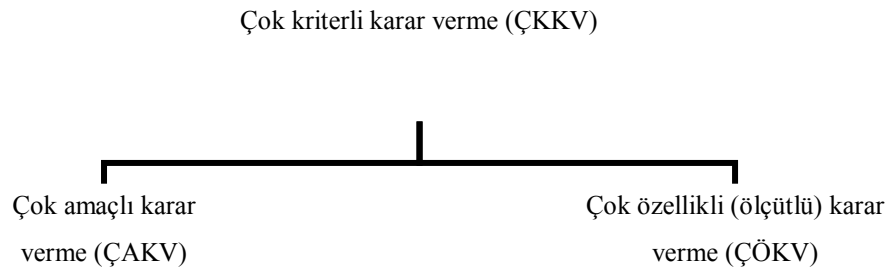
$x = (x_1; x_2; \dots; x_r)$: Karar vektörü (veya çözüm)

D : uygun çözüm uzayı

$F(x)$: amaç vektörü'nü ifade etmektedir.

Çok amaçlı programlama, sadece tek bir amaç yerine çelişen birden fazla amaca sahip olması dışında tıpkı tek amaçlı programlama gibidir ve doğrusal, tamsayı veya doğrusal olmayan olabilmektedir (Steuer, 1986). Birbiriyle çelişen birden fazla amaca sahip problemlerde, tüm amaç fonksiyonlarını en iyi şekilde sağlayan bir çözümün bulunması oldukça zordur.

Farahani ve diğ. (2010), çok kriterli karar verme (ÇKKV) problemlerinin çok ölçütlü karar verme (ÇÖKV) ve çok amaçlı karar verme (ÇAKV) tekniklerinin kombinasyonu olarak alındığını belirtmişlerdir. Steuer'de (1986), çok kriterli optimizasyonu çok kriterli karar verme başlığı altında ele almıştır. Çok kriterli karar vermenin ikiye ayrıldığını ifade ederek, bunları çok amaçlı matematiksel programlama olarak da kabul edilen çok kriterli optimizasyon ile çok özellikli karar analizleri olarak belirtmiştir. Şekil 2.35'te ÇKKV'ye ait şema gösterilmiştir.



Şekil 2.35: Çok kriterli karar verme (Jozefowicz ve diğ., 2008a)

Çok özellikli karar analizlerinde daha önceden belirlenmiş az sayıda alternatifler bulunurken, çok amaçlı optimizasyonda alternatif sayısı oldukça fazladır (Farahani ve diğ., 2010; Steuer, 1986). ÇÖKV'deki alternatifler her amacı özel bir seviyede karşılamakta ve karar verici de alternatifler arasında -her amacın önceliğine ve aralarındaki bağı göre- en iyi/iyileri seçmektedir. ÇAKV teknikleri ise en iyi alternatifi tasarlamaya çalışmaktadır. ÇAKV problemleri çeşitli parçalara sahiptir, fakat temel karakteristiklerinin Farahani ve diğ.'ne (2010) göre iyi tanımlanmış:

- a) Amaç seti,
- b) Kısıt seti olması gerekir.

Farahani ve diğ. (2010), ÇAKV için başlıca çözüm teknikleri olarak; küresel kriter yöntemi (global criterion method), yetenek fonksiyonu, metrik doğrusal programlama yöntemleri, sınırlandırılmış amaç fonksiyonu, hedef programlama, hedef ulaşma yöntemlerini saymışlardır.

Son yıllarda gelişen çözüm metodları ile artık çok amaçlı problemlere yönelik çalışmalar da artmıştır. 1984 yılında Gerald Evans çalışmasında “*son 15 yılda yöneylem araştırması ve yönetim bilimi alanında çok amaçlı matematiksel programlama hızla yükselmiştir*” ifadesini kullanmıştır (Evans, 1984). Bu ifadeden de çok amaçlı programlama çalışmalarının 1970'lerde yoğunluk kazandığı anlaşılmaktadır.

Uygulama alanı açısından farklılıkları ele alındığında; ÇÖKV analizleri genel olarak çözümü zor kamu politikası problemlerinde (havaalanı yeri seçimi, ilaç rehabilitasyon program tipi belirleme gibi) uygulanırken, ÇAKV daha az çelişkiye sahip iş ve idari problemlerde kullanışlıdır (Steuer, 1986). Kullanılan yöntem bakılmaksızın, ÇAKV problemlerinin çözümünde çakışan amaçlar ile etkili çözüm ikilemini gözden kaçırmamak gerekmektedir (Farahani ve diğ., 2010). Steuer'in (1986), çok amaçlı problemlere verdiği çeşitli örnekler şunlardır:

Petrol rafinerisi çizelgeleme

Minimum {maliyet}

Minimum {ithal edilen ham petrol}

Minimum {yüksek kükürlü ham petrol}

Minimum {talepte sapmalar}

Minimum {gaz genişlemesi}

Üretim planlama

Maksimum {toplam net gelir}

Maksimum {herhangi bir periyottaki en düşük net gelir}

Minimum {karşılanmamış sipariş}

Minimum {fazla mesai}

Minimum {bitmiş mal stoğu}

Taşıma

Minimum {maliyet}

Minimum {öncelikli müşterilere ortalama teslimat süresi}

Minimum {verilen bir işlemi kullanarak üretim}

Minimum {yakıt tüketimi}

Farahani ve diğ. de (2010) “kuruluş yeri kararları”ndan bahsetmişlerdir. Bu tür problemlerde de en az bir amaç fonksiyonu; maliyet, fayda, taşıma uzaklığı, servis bekleme zamanı bulunduğu belirtilerek kullanılabilen amaçlar için toplam kurulum maliyetini minimum kılmak, mevcut fabrikalardan en uzun mesafeyi minimum kılmak, duyarlılığı (talebe hızlı cevap verme) maksimum kılmak gibi örnekler verilmiştir. Son yıllarda çeşitli problemlerde kullanılan, enerji maliyeti, yer kullanımı ve inşaa maliyeti, gürültü, yaşam kalitesi, kirlilik, fosil yakıt (petrol,kömür vs) krizi ve turizm temelli çevresel ve sosyal amaçlar klasikleşmişlerdir.

2.6.1. Çok Amaçlı Karar Verme Teknikleri

Çok-amaçlı problemlerin çözümünde farklı yaklaşımlar bulunmaktadır. Bu yaklaşımlar farklı özelliklere göre literatürde gruplandırılmıştır. Örneğin, Farahani ve diğ. (2010), bu yaklaşımları 3 kategoriye ayırmışlardır;

- i. *Klasik yaklaşımlar*: çok amaçlıyı tek amaçlıya dönüştürmeye çalışırlar ve yeni tek amaçlıyı optimize ederler.

- ii. *Pareto optimal yaklaşımlar*: problem çözüldüğünde bir çözümler kümesiyle sonuçlandırılmaktadır. Bu yaklaşımlardan bazıları çok amaçlı genetik algoritma (ÇAGA), baskın olmayan derecelemede GA'dır.
- iii. *Karmaşık ÇAKV problemleri çözmek için daha başka özel amaç yaklaşımları*: Vektör evrimsel GA (VEGA), sözlük sıralaması, ağırlıklı en büyüklerin en küçüğü (minimaks) metodu ve uzaklık metodu olarak sayılabilmektedir. Son on yılda, çok amaçlı kombinatoriyal optimizasyon (ÇAKO) farklı çok kriterli problemleri çözmek için yeterli olabilmektedir.

Floudas ve Pardalos (2009), etkin çözüm kümesini belirlemede uygulanan üç yaklaşımı şu şekilde sıralamışlardır:

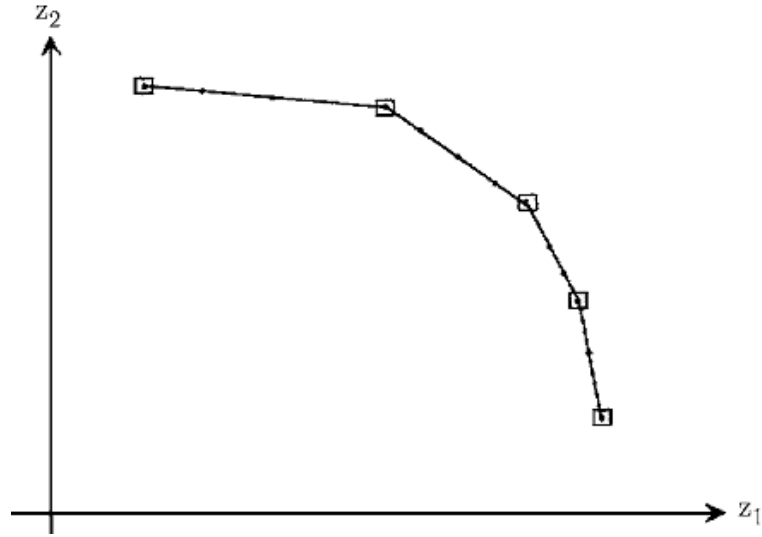
- i) Doğrudan yöntemler
- ii) İki aşamalı yöntemler
- iii) Sezgisel yöntemler

2.6.1.1. Doğrudan yöntemler

Tek amaçlı problemler için yoğun olarak kullanılan klasik yöntemlerin kullanılması şeklindedir. Dal-sınır yaklaşımı bunlardan birisidir.

2.6.1.2. İki aşamalı yöntemler

İki aşamalı yöntemlerde ilk aşamada desteklenen etkili çözümler kümesi olan $SE(P)$ belirlenmektedir. $S \cup S'$ bu listeyi temsil etmektedir. S , sırasıyla z_1 ve z_2 amaçları için etkili iki optimum çözüm ile başlatılmaktadır. İlk aşama S' 'teki tüm çözüm çiftleri denenene kadar devam ettirilir. Son olarak, $SE(P) = S \cup S'$ elde edilmektedir, Şekil 2.36'da gösterilmiştir.



Şekil 2.36: $SE(P) = S \cup S'$ grafiği (Floudas ve Pardalos, 2009)

2.6.1.3. Sezgisel yöntemler

Optimizasyon problemlerinin çözümünde ilk olarak doğrusal modellemeler kullanılırken, bu yöntemlerin her problemin çözümü için yeterli olmadığı anlaşılmıştır. Çok amaçlı optimizasyon problemlerinin çözümünde son yıllarda genetik ve evrimsel algoritmalar yaygın olarak kullanılmaya başlanmıştır (Sağ ve Çunkaş, 2009). Bunun yanında tavlama benzetimi, tabu arama ve genetik algoritma gibi metasezgiseller de çok amaçlı programlama için kullanılmışlardır. Karar vericinin katılımına ise göre çok amaçlı problemleri çözmek için üç yaklaşım bulunmaktadır (Jozefowicz ve diğ., 2008b);

- i) *Deneyden önce (A priori) yaklaşımlar*; karar vericinin farklı amaçlar için tercihler sunduğu yaklaşımdır.
- ii) *İnteraktif (etkileşimli) yaklaşımlar*; karar vericinin seçimlerinin problem çözüme esnasında gerçekleştiği yaklaşımlardır.
- iii) *Deneyden sonra (A posteriori) yaklaşımlar*; olası baskın olmayan çözümlerin oluşturulması ve karar vericinin bunlar arasından seçim yaptığı yaklaşımdır.

Atlas (2008) ise çok amaçlı programlama çözüm tekniklerini literatürde genel olarak karşılaşılan şu iki sınıflandırmada ele almıştır:

- a) Tercih bilgisinin gereğine ve zamanına bağlı çözüm teknikleri: Çok amaçlı optimizasyon için tercih bilgisi zamanına bağlı dört genel yaklaşım; karar

vericiden tercih bilgisinin çözümden önce, çözüm esnasında, çözümden sonra alınması ve tercih bilgisi alınmadan olması şeklindedir.

- b) Karar değişkenlerinin sürekli ya da en az birinin kesikli olmasına göre yapılan sınıflandırmadır.

2.6.2.Amaç Uzayı

Tek amaçlı problem genellikle karar uzayında çalışılırken, çok-amaçlı optimizasyon çoğunlukla amaç uzayında çalışmaktadır. Karar uzayında uygun bölge S ile, kriter bölgesinde uygun bölge ise Z ile gösterilmiştir. Steuer (1986) notasyonu şu şekilde göstermiştir:

$$Z = \{ z \in R^k / z = Cx, x \in S \}$$

Başka bir ifadeyle, S 'teki tüm noktaların görüntü seti Z 'dir. S negatif olmayan bir alanı tanımlamak zorunda iken, Z bu şartı taşımak zorunda değildir. Steuer'in (1986) verdiği örneklerle konu daha açıklayıcı olacaktır. Ancak önce örneklerdeki gösterimleri şu şekilde açıklayabiliriz: Amaç uzayındaki çözümün görüntüsü ($z = [z^1, z^2, \dots, z^k]$ şeklinde) noktalardan oluşmaktadır. Amaç fonksiyonlarının katsayıları sırası ile z_1 amaç fonksiyonu için c^1 ile, z_2 amaç fonksiyonu için c^2 ile, z_3 amaç fonksiyonu için c^3 ile... gösterilmiştir. S uygun alanındaki çözüm noktaları x^1, x^2, \dots, x^k şeklinde gösterilmiştir. ($x^1 = (x_1^1, x_2^1), x^2 = (x_1^2, x_2^2), \dots, x^k = (x_1^k, x_2^k)$)

Örneğin; iki değişkenli z_1 ve z_2 fonksiyonlarının katsayıları kümesi c^1 ve c^2 ile gösterilmiş olsun.

$$c^1 = (1, -1) \text{ gösterimi } z_1 = 1x_1 - 1x_2 \text{ ifadesindeki katsayılar,}$$

$$c^2 = (-1, 2) \text{ gösterimi de } z_2 = -1x_1 + 2x_2 \text{ ifadesindeki katsayılardır.}$$

Şekil 2.32'deki gibi uygun çözüm alanı S olsun. Çözüm noktaları olan x^1, x^2, x^3, x^4 için z^1, z^2, z^3, z^4 değerleri şu şekilde bulunur:

$$x^1 = (0,0) \quad z^1 = (0,0)$$

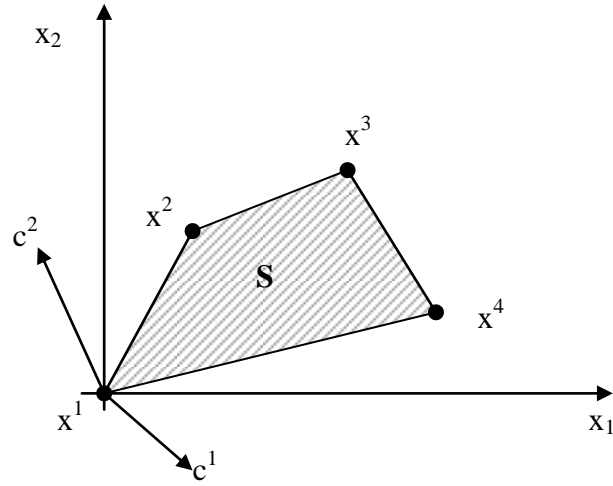
(x^1 noktası $x_1=0$ ve $x_2=0$ karşılık gelen noktadır. z_1 'de bu değerler yerine yazılınca $z_1=1*0-1*0=0$ ve z_2 'de bu değerler yerine yazılınca $z_2=-1*0+2*0=0$ bulunmakta ve $z^1=(z_1, z_2)$ noktası $(0,0)$ ile gösterilmektedir.)

$$x^2 = (1,2) \quad z^2 = (-1,3)$$

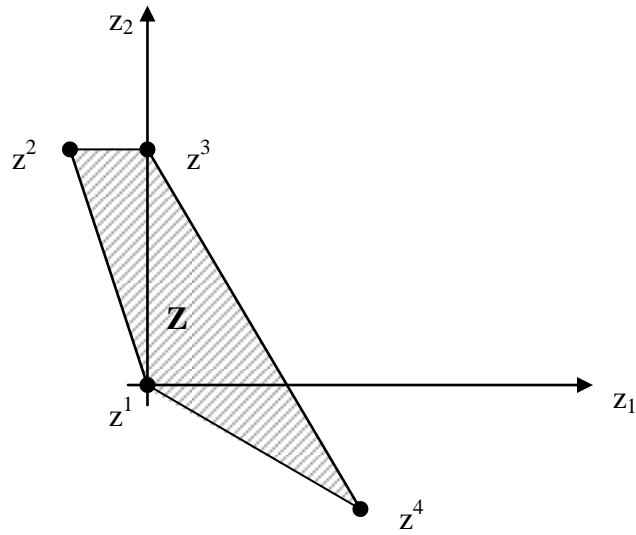
$$x^3 = (3,3) \quad z^3 = (0,3)$$

$$x^4 = (4,1) \quad z^4 = (3,-2)$$

Bu örneğin grafik gösterimi Şekil 2.37 ve Şekil 2.38'de gösterilmiştir.



Şekil 2.37: Uygun çözüm alanı. (Steuer, 1986)



Şekil 2.38: Uygun kriter alanı. (Steuer, 1986)

Şekil 2.37 ve Şekil 2.38'den görüldüğü gibi S 'in uç noktaları Z 'de de uç noktaları göstermektedir ve Z 'de negatif olmama şartı aranmamaktadır.

2.6.3. Fayda Fonksiyonu Yaklaşımı

U , karar verici için fayda fonksiyonunu göstermek üzere, çok kriterli optimizasyon şu şekilde gösterilebilmektedir;

$$\begin{aligned} \text{optimum} \quad & \{U(z_1, z_2, \dots, z_k)\} \\ \text{şu kısıtlar altında: } & f_i(x) = z_i \quad 1 \leq i \leq k \\ & x \in S \end{aligned} \quad (2.12)$$

(2.12) ifadesi genellikle doğrusal olmayan şekildedir. Fakat bu, yaklaşımdaki asıl zorluk değildir. Asıl zorluk, çoğu problemde karar vericinin fayda fonksiyonunu matematiksel olarak ifade edememesidir. Gerçek uygulamalarda, karar vericinin fayda fonksiyonunun matematiksel bir gösterime sahip olması beklenmemektedir. Fakat bu şekilde olduğu varsayımı ile notasyonu Steuer (1986) şu şekilde göstermiştir:

$$\begin{aligned} \text{enbüyük} \\ \text{enküçük} \end{aligned} = \{ U(z) / z = Cx, x \in S \}$$

U fayda fonksiyonunun elde edilmesindeki zorluk ve bu fonksiyonun neredeyse tamamıyla doğrusal olmayan şekilde olması sebepleriyle çok amaçlı optimizasyon probleminde fayda fonksiyonu göz önüne alınmamaktadır. Buna rağmen, fayda fonksiyonu yaklaşımı kullanışlıdır. Eğer fayda fonksiyonu ile ifade edebilme durumu söz konusu ise grafik gösterimi Steuer'in (1986) örnekleri üzerinden şu şekilde olacaktır:

Örneğin, iki maksimizasyon amacına sahip bir probleme ait denklemler şu şekilde verilmiş olsun;

$$\text{maksimum } \{x_1 + x_2 = z_1\}$$

$$\text{maksimum } \{x_1 = z_2\}$$

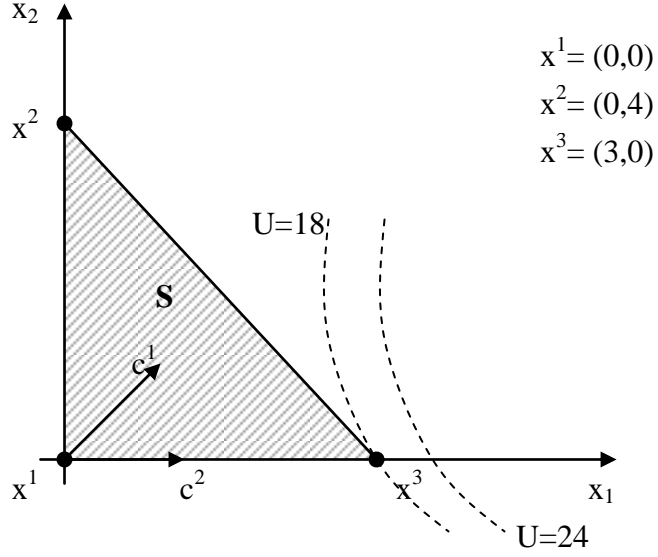
şu kısıtlar altında:

$$4x_1 + 3x_2 \leq 12$$

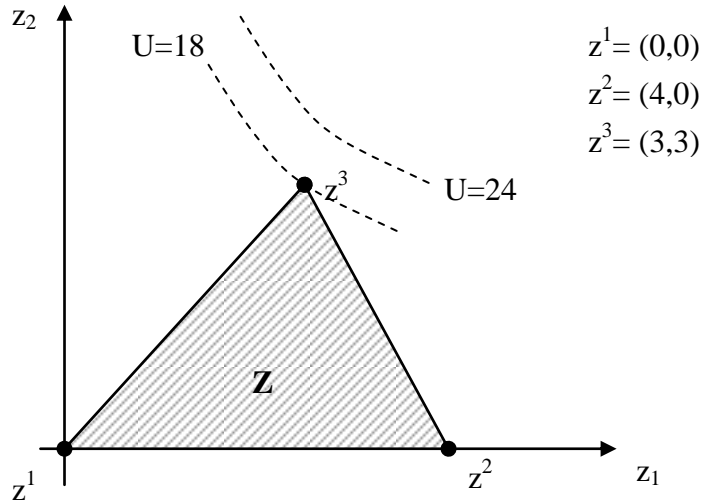
$$x_1, x_2 \geq 0$$

$$U = 2z_1z_2$$

Bu örneğin karar uzayı ve kriter uzayı Şekil 2.39 ve Şekil 2.40'deki gibi olmaktadır.



Şekil 2.39: Karar uzayı (Steuer, 1986)



Şekil 2.40: Kriter uzayı (Steuer, 1986)

Şekil 2.39'da en iyi çözüm değeri $x^3 = (3,0)$ noktası ve Şekil 2.40'da $(3,3)$ vektörü en iyi kriter vektörüdür. En iyi değer ise $U=18$ 'dir.

2.6.4. Hedef Programlama

Çok amaçlı programlama modelleri arasında en çok kullanılan tekniklerden biri olan hedef programlamayı 1955 yılında Charnes ve diğ. geliştirmiş, 1961 yılında Charnes ve Cooper tarafından daha açık bir şekilde, doğrusal programlamanın genişletilmiş bir versiyonu olarak ortaya koymuşlardır (Özdemir ve Gümüšoğlu, 2006; Keçek, 2005). Hedef programlamada doğrusal olarak ifade edilen modeldeki eşitsizlikler gerekirse kısıtların ihlal edilebileceği esnek hedeflere dönüştürülmektedir. Bunun için sapma değişkenleri ilave edilir ve modeldeki iki sapma değişkeninden biri pozitif değer olarak model çözülür (Taha, 2000).

Steuer (1986) hedef programlamanın, doğrusal programlamadan ayrılan dört noktasını şu şekilde belirtmiştir:

- i. Amaçların hedef olarak alınması,
- ii. Hedeflere ağırlık veya öncelik atanması,
- iii. Hedeflenen t_i seviyesinden olan fazlalık veya eksikliği ifade eden sapma değişkenlerinin kullanılması,
- iv. Hedefleri en iyi şekilde sağlayan çözümleri bulmak için sapma değişkenlerinin ağırlıklı toplamlarının minimizasyonu.

2.6.5. Baskınlık

$z^1, z^2 \in R^k$ olmak üzere iki kriter vektörü ele alındığında;

Eğer maksimizasyon için $z^1 \geq z^2$ (minimizasyon için $z^1 \leq z^2$) ve $z^1 \neq z^2$ ise z^1, z^2 'ye baskındır denir. Bu durumda z^1 'in hiçbir bileşeni z^2 'nin aynı bileşenlerinden daha düşük olamaz ve en azından z^1 'in bir bileşeni z^2 'nin bileşenine göre daha iyi olmalıdır. Eğer maksimizasyon için $z^1 > z^2$ (minimizasyon için $z^1 < z^2$) ve $z^1 \neq z^2$ ise z^1, z^2 'ye kuvvetli baskındır denir. Bu durumda z^1 'in her bileşeni z^2 'nin aynı bileşenlerinden daha iyi olmalıdır.

Tablo 2.6'daki maksimizasyon kriter vektörleri için z^1, z^3 ve z^5 tabloda belirtilen "Hakim Vektör" sütununda ifade edilen vektörler tarafından bastırılmaktadırlar. Örneğin, z^2 vektörünün $z_1=2$ değeri z^1 vektörünün $z_1=-1$ değerinden, $z_2=4$ değeri z^1 vektörünün $z_1=3$ değerinden ve $z_3=6$ değeri z^1 vektörünün $z_1=4$ değerinden daha büyüktür ve maksimizasyon doğrultusunda, z^2 vektörü üç kriter değeriyle de z^1 vektörüne ait

değerlerden daha iyi olduğu için, z^2 , z^1 'e kuvvetli baskındır. z^2 , z^4 ve z^6 arasında herhangi bir baskınlık ifadesi mevcut değildir.

Tablo 2.6: Baskınlık ifadesi için örnek (Steuer, 1986)

Kriter Vektörü	Kriter Değerleri			Hakim Vektör
	z_1	z_2	z_3	
z^1	-1	3	4	z^2 (kuvvetli baskın)
z^2	2	4	6	
z^3	2	2	5	z^2 ve z^4 (zayıf baskın)
z^4	3	2	5	
z^5	3	3	-1	
z^6	8	3	0	z^6 (zayıf baskın)

2.6.6. Baskın Olmayan Kriter Vektörü

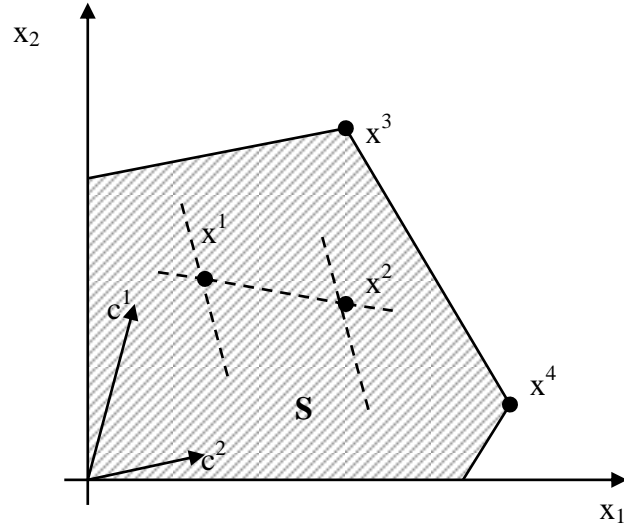
Bir kriter (ölçüt) vektörünü eğer bir başka uygun kriter vektörü baskılamamışsa bu vektöre “bastırılmamış” denilmektedir. Bir kriter vektörü bastırılmamış olmadıkça, en iyi olamamaktadır.

2.6.7. Etkililik

Optimumluk (en iyilik) için çözüm noktası etkili olmalıdır. S 'teki bir noktanın, sadece ve sadece kriter vektörü bastırılmamışsa (nondominated) etkili olduğu söylenebilmektedir. Yani, bir noktanın diğer amaçlarından en az bir tanesi azalmadan, bir amacını artırma şeklinde uygun şekilde hareket ettirmek mümkün değilse o nokta etkilidir. Etkisiz çözümler optimallik için aday değildirler. Uygulamada, optimuma yakın çözümler kabul edilmektedir. Optimum veya optimuma yakın olsun karar sürecini tatmin edici bir şekilde sonlandıran her çözüme “son çözüm” adı verilmektedir (Steuer, 1986). Eğer $z = f(x)$ gibi bir $x \in X$ çözümü mevcut ise, z noktası ulaşılabilirdir. Ulaşılabilir tüm noktalar kümesi Z ile gösterilmiştir.

Tablo 2.7: Etkili çözüm için örnek (Steuer, 1986)

Uygun Nokta	Kriter Değerleri			Kriter Vektörünün Hakim Noktası
	z_1	z_2	z_3	
x^1	7	5	-1	
x^2	2	6	-2	
x^3	1	1	-2	x^1, x^2, x^4
x^4	5	2	-1	x^1
x^5	9	2	-6	

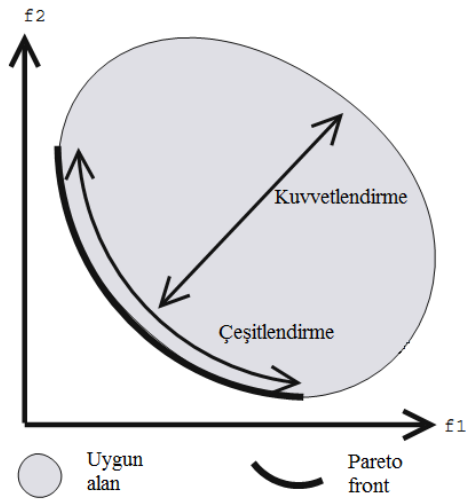


Şekil 2.41: Etkili çözüm için örnek şekil (x^3, x^4 etkili çözüm kümesidir)

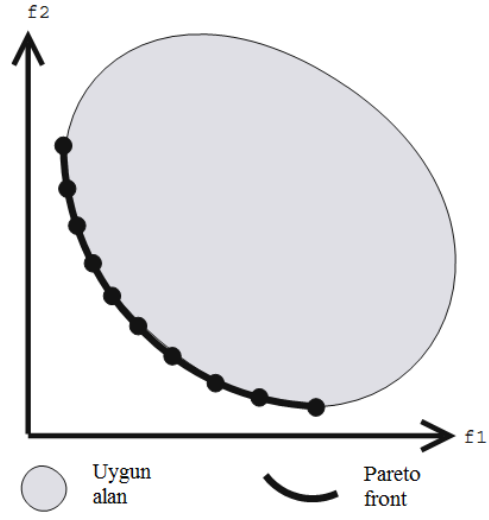
İdeal olan amaç vektörü z^* şu şekilde tanımlanmıştır:

$$z^* = [opt f_1(x), opt f_2(x), \dots, opt f_k(x)]$$

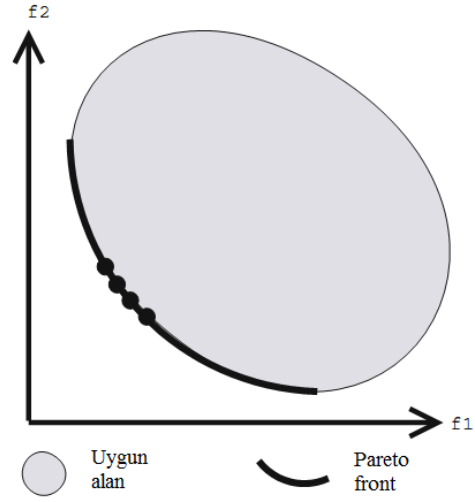
Metasezgisellerin tasarımında çelişen iki kriter olan kuvvetlendirme ve çeşitlendirme için Jozefowicz ve diğ. (2008a) amaç uzayında görsel örnekler vermiştir. Bunlar, Şekil 2.42 ile Şekil 2.45 arasında gösterilmişlerdir.



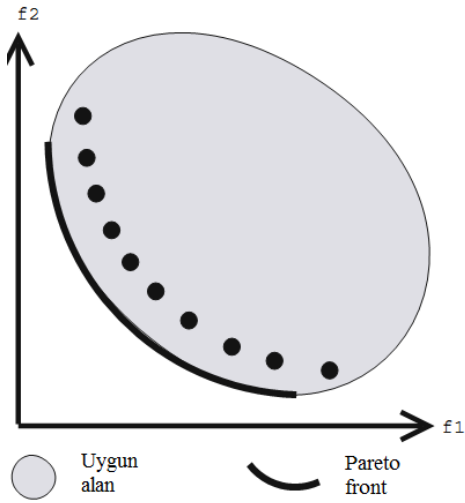
Şekil 2.42: Çok amaçlı metodda iki amaç.



Şekil 2.43: İdeal çözüm.



Şekil 2.44: Kuvvetlendirme için iyi çözüm.



Şekil 2.45: Çeşitlendirme için iyi çözüm.

2.6.8. Pareto Optimal

Problemdeki amaçların herhangi biri için en kötü olmayan ve en azından bir amaç için diğerlerinden daha iyi olan çözüm Pareto-Optimal çözümdür. Başka bir ifadeyle, çözüm kümesinde yer alan herhangi bir diğer çözüm tarafından bastırılmamış olan çözümdür (Sağ ve Çunkaş, 2009). Pareto optimum (bastırılmayan çözüm) kavramı, çok amaçlı optimizasyon problemlerindeki en iyi çözümünü ifade etmede kullanılmaktadır. Örneğin bir minimizasyon problemi için, (2.13) numaralı eşitliği sağlayan başka bir x çözümü bulunmuyorsa, x^* etkin çözüm (pareto en iyi çözüm) olarak ifade edilmektedir. Etkin çözümlerin hepsinin aynı derecede önemli olduğu belirtilmiş olup herhangi bir etkin çözüm için diğerlerinden daha iyidir denilememektedir. Etkin çözümlerin bulunması için literatürde önerilen çok sayıda yöntem bulunmaktadır (Şahin ve Türkbey, 2010).

$$f_i(x) \leq f_i(x^*) \quad \forall i \quad i=1, 2, \dots, n \text{ için} \quad (2.13)$$

$$f_i(x) < f_i(x^*) \quad \text{en az bir } i \text{ için}$$

2.7. ÇOK AMAÇLI ARAÇ ROTALAMA PROBLEMLERİ

Bu başlık altında çok amaçlı araç rotalama problemlerinin literatür taraması yapılmış ve kullanılan amaç fonksiyonları hakkında bilgi verilmiştir. Araç rotalama problemlerinin sınıflandırılmasında esas olan konular ağ, talep, araç, maliyet ve amaç başlıkları altında toplanmıştır. Araç rotalama problemlerinde amaçlar, çeşitli ve çok sayıda olabilmektedir.

Araç rotalama problemleri genellikle maliyet (uzaklık veya finansal) minimizasyonu gibi tek amaca sahip olmaktadır. Çoğu gerçek endüstri problemleri ise sadece maliyet ile sınırlandırılmayıp, birden çok amaca sahiptir. Örneğin, sadece maliyet bile göz önüne alınmış olsa, aynı problem için farklı maliyetlerin (zaman ve para gibi) tanımlanması mümkündür. Sadece maliyet ile sınırlandırılmış da değil, ayrıca gecikme ve eşitlik gibi kavramlarla da uğraşan amaçlar olabilmektedir. Bu sebeplerden çok amaçlı araç rotalama problemlerine ilgi gösterilmektedir. Çok amaçlı araç rotalama problemi çalışmaları, birkaç (genellikle çelişen) amaç barındıran problemlere yönelik çözüm metodları sunan “çok amaçlı optimizasyon” alanına girmektedir. Bu ilgi alanının kökleri de Edgeworth ve Pareto’nun 19. yüzyılda yapmış oldukları ekonomik araştırma

çalışmalarında yatmaktadır. 1980'li yılların ortalarından beri bu alan artan bir ilgi görmektedir. 1990'lı yılların ortalarından itibaren, özellikle çok amaçlı evrimsel algoritmalar gibi metodlara hızlı bir ilgi artışı söz konusudur (Jozefowicz ve diğ., 2008a).

2.7.1. Araç Rotalama Problemlerinin Çok Amaçlı Kullanımı

Jozefowicz ve diğ. (2008b) araç rotalama problemlerinde çok amaçlı kullanımının genellikle şu üç şekilde yöneldiğini belirtmişlerdir; (uygulama alanlarına göre sınıflandırma)

- i) Klasik akademik problemlerin bir uzantısı (pratik uygulamalarını geliştirmek için)
- ii) Klasik problemleri genelleştirmek için
- iii) Gerçek hayat vakalarını çözmek için (karar verici vardır)

2.7.1.1. Klasik Akademik Problemlerin Uzantısı Olarak Çok Amaçlı Kullanımı

Problem tanımı değişmeden kalmakta ve yeni amaçlar ilave edilmektedir. Bu uzantıların amacı genelde modelin pratik uygulamalarını, lojistik problemlerin sadece maliyet temelli olmadığını farkına vararak artırmaktır. Jozefowicz ve diğ., (2008a) şu şekilde uygulamaları sıralamıştır:

- *Sürücünün iş yükünün eşit dağıtılması*: Lee ve Ueng 1998 yılında tur uzunluğunun dengelenmesini göz önüne alan bir araç rotalama problemi uzantısı sunmuşlardır. Bu denge amacı oluşturulan çözümdeki eşitliği artırmak için eklenmiştir. Bu çalışmanın yazarları Tayvan'da sürücülerin atamasına karar veren kişinin kişisel tercihlerinin, eksik bilginin etkili olabileceği gerçeğinden yola çıkmışlardır. Sürücüler çizelgelerinden şikayet ettiklerinde Lee ve Ueng, sürücüler şirket için fark oluşturabildikçe yardımlarının önemli olduğunu düşünmüşlerdir.

- *Müşteri memnuniyeti*: 1998 yılında Sessomboon zaman pencereli araç rotalama problemlerine, dağıtım günlerine göre müşteri memnuniyetini geliştirmek için bu amacı eklemiştir.

- *Ticari dağıtım* ; 2001 yılında Riberio ve Lourenço periyodik araç rotalama problemlerinin bir uzantısı olarak sunmuştur. Bu dağıtım yiyecek veya içecek endüstrisi için ele alınmıştır. Yazarlar pratik bir uygulamasını gerçekleştirmemiştir.

- *Çok amaçlı Gezgin Satıcı Problemleri ve çeşitleri*: Bazı araştırmacılar çok amaçlı gezgin satıcı problem üzerine çalışmalar yürütmüşlerdir.

- *Diğer çok amaçlı araç rotalama problemleri*: Bazı çok amaçlı araç rotalama problemleri, klasik rotalama problemleri ile genel amaçları paylaşmamaktadır. Örnek olarak, toplam ortalama taşıma zamanı ve taşıma zamanındaki değişimin minimizasyonu verilebilir.

2.7.1.2. Klasik Problemleri Genelleştirmek İçin Çok Amaçlı Kullanımı

Bir başka çok amaçlı optimizasyon uygulaması da problemi genelleştirilebilmek için kısıt veya parametre yerine amaç eklenmesidir.

Literatürde rotalama problemlerinde, kullanılan amaç fonksiyonlarına göre özel oluşturulmuş bazı problem çeşitleri bulunmaktadır. Boeffley (1995) buna örnek olarak şu problemleri vermiştir:

a) Tehlikeli madde dağıtımında dört amaç kullanılabilir. Bu amaçlar:

- (1) kaza olasılığını minimum kılmak,
- (2) kaza gerçekleşmesi durumunda risk dağılımını minimum kılmak,
- (3) geçiş yapılacak bölgeler arasında “olabildiği kadar eşit risk dağılımı” eşitliğini maksimum kılmak,
- (4) rota uzunluğunu (veya maliyeti veya zamanını) minimum kılmak,

(1) ve (2) numaralı amaçların birleştirilmesi mantıklı olabilmektedir. Problem böylece eş zamanlı olarak riskin minimizasyonu, eşitliğin maksimizasyonu ve rota uzunluğunun minimizasyonu haline gelmiş olacaktır. Boeffley (1995), tehlikeli madde dağıtım amaçları üzerinden iki özel problem tanımından bahsetmiştir:

İki kriterli en kısa yol (Bicriterion Shortest Path): eğer sadece toplam risk ve rota uzunluğunun minimizasyonu istenirse, Vincke'nin 1974 yılında ve Hansen'in 1980 yılında ele aldığı bu problem elde edilecektir.

Maksimum eşitlikli en kısa yol (Maximal Equity Shortest Path): eğer sadece eşitliğin ve rota uzunluğunun optimizasyonu istenirse 1990 yılında Gopalan'ın ele aldığı bu problem elde edilecektir.

b) Sağlık hizmeti veya kütüphane servisi sağlamak gibi hareketli araçların söz konusu olduğu bir örnekte üç amaç kullanılabilir. Bu amaçlar:

- (1) servis istasyonunun belirlenmiş mesafe (veya zaman) içindeki sayısını maksimum kılmak, (maximum cover)
- (2) en yakın bölgelerindeki servis istasyonundan yararlanacak tüm insanların mesafesini minimum kılarak, servise ulaşımı maksimum yapmak,
- (3) rota uzunluğunu (veya maliyeti veya zamanını) minimum kılmak.

Maksimum kapsayan en kısa yol (Maximum Covering Shortest Path): sadece (1) ve (3) numaralı amaçlar göz önüne alındığında 1985 yılında Current ve diğ.'nin çalıştığı problemdir.

Eşit miktarda en kısa yol (median shortest path): sadece (2) ve (3) numaralı amaçlar göz önüne alındığında 1987 yılında Current ve diğ.'nin çalıştığı bu problem elde edilecektir.

2.7.1.3. Gerçek Hayat ile İlgili Vakalar İçin Çok Amaçlı Kullanımı

Çok amaçlı rotalama problemleri gerçek hayatta karşılaşılan problemlere de uygulanmıştır. Jozefowicz ve diğ.'nin (2008a) verdiği örnekler şu şekildedir:

- *Ulaştırma dağıtım rotalama:* 2001 yılında El-Sherbeny bir Belçika taşıma firması için çalışma yapmıştır. Firma tarafından sekiz amaç belirlenmiştir. Farklı çeşitlerdeki araçlar kullanılmış ve kapasite kısıtı kullanılmamıştır.

- *Şehir otobüsü rotası belirleme:* 1995 yılında Bowerman ve diğ. okul otobüsü rotalama üzerine çalışma yapmıştır. Farklı bölgelerdeki öğrencilerin yerel okul otobüsleri ile okullarına dağıtılması şeklinde, belirli şehirler özellikle de Wellington için, çalışma yapılmıştır. Yazarlar bu problemde dört amaçlı bir matematiksel model sunmuştur. Bu amaçlar; toplam rota uzunluğunun minimizasyonu, toplam öğrenci yürüme mesafesinin minimizasyonu, eşit dağıtım yüklemesi ve toplam katedilen mesafenin otobüslere eşit dağılımı olarak belirlenmiştir.

- *Köy otobüsü rotası belirleme:* 2002 yılında Corberan ve 2006 yılında Pacheco ile Marti öğrencilerin evden okula ve tekrar okul çıkışı evlerine dönüşlerine yönelik bir çalışma gerçekleştirmiştir. Ekonomiklik ve rahatlık sağlarken aynı zamanda mümkün olduğunca güvenlik de sağlanmak istenmiştir.

- *Şehirde çöp toplama:* 2006 yılında Lacomme, Fransa'nın Troyes şehrinde sokaklardaki çöplerin toplanmasına ve atık arıtma tesisine dağıtımına yönelik olarak çalışma yapmıştır. Problem, bağlantı-rotalama olarak modellenmiştir. Yazarlar toplam

rota uzunluğunun minimizasyonu ve en uzun turun minimizasyonu şeklinde iki amaç göz önüne alınmıştır.

- *Tehlikeli ürün dağıtımı*: 1998 yılında Giannikos tehlikeli ürün dağıtımı üzerine çalışma yapmıştır. Fabrikalardan toplanan ürünlerin gemilere taşınması esnasında dört tane amacı göz önüne almıştır. Bunlar toplam operasyon maliyetinin minimizasyonu, toplam hissedilen riskin minimizasyonu, nüfus merkezleri arasında riski adil şekilde dağıtabilmek ve arıtma tesislerindeki faaliyetlerden kaynaklanan olumsuzlukların dengeli ve eşit dağıtılmasını sağlamak şeklindedir.

2.7.2. Çok Amaçlı Araç Rotalama Problemlerinde Kullanılan Amaçlar

Jozefowicz ve diğ. (2008b), ARP'de en sık kullanılan amaç fonksiyonlarını üç genel başlık altında toplamıştır:

2.7.2.1. Tur ile İlgili Amaçlar

Bu amaçlara örnek olarak şunlar verilmiştir;

Maliyet (yol, zaman, maliyet)

Makespan (en uzun turu minimum yapmak gibi)

Balans (müşteri sayısı, dağıtılan malzeme, turda geçen süre gibi başlıklar altında denge)

Spesifik amaçlar (tehlikeli ürünlerin taşınmasında risk gibi)

2.7.2.2. Düğüm/Bağlantı Faaliyetleri ile İlgili Amaçlar

Zaman pencereci ARP, müşteri veya sürücünün bekleme zamanını minimum kılmak, nokta veya bağlantıya öncelik atanması gibi örnekler mevcuttur.

2.7.2.3. Kaynaklar ile İlgili Amaçlar

Araç sayısını minimum kılmak, araç maliyet-etkililiğini (veya kapasite) maksimum kılmak ve benzeri amaçlardır.

2.7.3. Literatürdeki Çok Amaçlı Araç Rotalama Uygulamalarından Örnekler

Pacheo ve Marti (2006) iki amaçlı rotalama problemi için tabu arama metasezgiseli ile uygulama yapmışlardır. Bu yayında okul otobüslerini rotalamak için iki amaç kullanılmıştır. Bu amaçlar:

(1) Otobüs sayısını minimum kılmak,

(2) Bir öğrencinin otobüste geçirdiği en uzun süreyi minimum kılmak.

(1) ve (2) numaralı amaç fonksiyonların ikisi de minimizasyon olmasına rağmen mantıksal çakışma vardır.

Tan ve diğ. (2006) minimum rota uzunluğu ve kamyon sayısı istenen iki amaçlı problemi incelemişlerdir. Boş veya yüklü konteynırların hareketlerini lojistik firması için, ulaştırma problemi olarak ele almış ve sınırlı sayıda araca (kamyon ve treyler) sahip olan firmanın dışarıdan destek aldığı kabul etmişlerdir. Burada zaman penceresi kısıtı kullanılmıştır. Ombuki ve diğ. de (2006) yine araç sayısını ve toplam maliyetin (uzaklığın) minimizasyonu amaçlarını kullanarak, zaman pencereli araç rotalama problemi için genetik algoritma metasezgiselinden yararlanmışlardır.

Martin Josef Geiger (2001) zaman pencereli araç rotalama problemi için genetik algoritma metasezgiselini kullanmış ve dört amaç belirlemiştir. Bu amaçlar:

- (1) Araçların katettiği toplam mesafeyi minimum kılmak,
- (2) Kullanılan araç sayısını minimum kılmak,
- (3) Zaman penceresindeki bozulmayı minimum kılmak,
- (4) Bozulan zaman penceresi sayısını minimum kılmak.

Dabia ve diğ. (2010) müşterilerin bekleme zamanlarını da içeren toplam süreyi minimum kılmayı ve dağıtım gerçekleştirilen talebi maksimum kılmayı amaçlayan çok amaçlı kapasiteli tek tip araçlı zaman pencereli araç rotalama probleminin çözümü için dinamik programlamadan faydalanmışlardır. Ghoseiri ve Ghannadpour (2010) hedef programlama yaklaşımı ile genetik algoritmanın uygulandığı minimum araç sayısı ve minimum mesafeyi elde etmek amaçları olmak üzere iki amaçlı zaman pencereli araç rotalama problemi üzerinde çalışmışlardır. Jozefowicz ve diğ. (2009) minimum ve maksimum rota uzunluklarının arasında denge kurmak gibi rotalar arası dengeyi ve toplam mesafeyi minimum kılmayı amaçlayan araç rotalama problemini çözmek için evrimsel algoritma tabanlı metasezgisel yaklaşım uygulamışlardır.

3. MALZEME VE YÖNTEM

Bu tezde çok amaçlı araç rotalama problemi için deęişken komşuluk arama (DKA) metasezgisel yaklaşımı uygulanmıştır. Deęişken komşuluk arama metodunun çok amaçlı problemler üzerinde fazla uygulaması olmadığı için araç rotalama probleminin çok amaçlı çeşidi uygulamada tercih edilmiştir.

3.1. DEĞİŞKEN KOMŞULUK ARAMA YAKLAŞIMI

DKA ilk olarak Mladenovic ve Hansen (1997) tarafından geliştirilmiş ve p-medyan problemi üzerinde uygulanmıştır. P-medyan probleminde fabrika veya depo kurulabilecek alternatif n adet aday yer bulunmakta ve bunlar içersinden kurulumun gerçekleşeceği m adet yerin seçimi gerçekleştirilmektedir. Literatür taraması sonucunda DKA'nın genellikle yer seçimi, rotalama, proje çizelgeleme, iş sıralama, sürekli optimizasyon ve graf yapılarla ilgili uygulamalara sahip olduğu görülmüştür. Örneğin Fleszar ile Hindi (2004) kısıtlı proje çizelgelemede ve kapasite kısıtlı p-medyan probleminde (2006) DKA yaklaşımını uygulamışlardır. P-medyan problemi üzerinde Hansen ve dię. (2001) DKA'nın "ayırıştırma" çeşidi ile uygulama gerçekleştirmişlerdir. Kytöjoki ve dię. (2007) büyük ölçekli araç rotalama problemi üzerinde, Bräysy (2003) ise zaman pencereli araç rotalama problemini incelemiştir. Ayrıca bu yöntem çizelgeleme problemlerinde de uygulanmıştır. Burke ve dię. (2008) hemşirelerin nöbet çizelgelemesinde, Roshanaei ve dię. (2009) ise yöntemi çizelge boyunu minimum kılmayı amaçlayan makina çizelgeleme problemlerinde uygulamışlardır.

Deęişken komşuluk arama yaklaşımı çözülmesi zor olan karmaşık problemlerin (NP-zor) çözümü için geliştirilmiş olan son metasezgisel yöntemlerden biridir. Deęişken komşuluk arama yaklaşımı, sistematik komşuluk deęişimleri fikrini kullanmaktadır (Liberti ve Maculan, 2006).

DKA yaklaşımı, başlangıç noktasından yerel minimuma ulaşmada azalan (descent) metod olarak çalışmaktadır. Daha sonra bu çözümün önceden tanımlanmış farklı komşuluklarının serisi araştırılarak, her tekrarda geçerli komşuluğun bir veya birkaç noktası, yerel azalma yöntemini çalıştırmak için başlangıç noktası olarak kullanılır. Arama eğer sadece ve sadece geçerli olan çözümden daha iyi ise yeni yerel minimuma atlamaktadır. Bu bağlamda, DKA; tavlama benzetimi veya tabu arama gibi yörünge izleme metodu (aynı komşuluk içerisinde gelişmemiş hareketlere izin veren) değildir ve diğer metasezgisellerden daha basit yollarla çok iyi çözümler sağlamaktadır (Perez ve diğ., 2006).

Perez ve diğ. (2006) bir optimizasyon probleminin, rasgele seçilmiş X kümesinde tanımlanmış olan gerçek değerli f fonksiyonunun minimum veya maksimumunu bulmaya bağlı olduğunu belirtmişlerdir. Minimizasyon için komşuluk ve çözüm değerlerini de matematiksel olarak (3.1) ve (3.2)'deki gibi göstermişlerdir.

$$\min \{ f(x) : x \in X \} \quad (3.1)$$

(3.1) ifadesinde X çözüm uzayını, x uygun çözümü, f ise amaç fonksiyonunu göstermektedir. Bu formülden elde edilen minimum değer, problemin optimal çözümü (ya da global minimumu) olan x^* uygun çözümdür. Bu yüzden;

$$x^* \in X, f(x^*) \leq f(x) \text{ i sağlar, } \forall x \in X \quad (3.2)$$

yazılır.

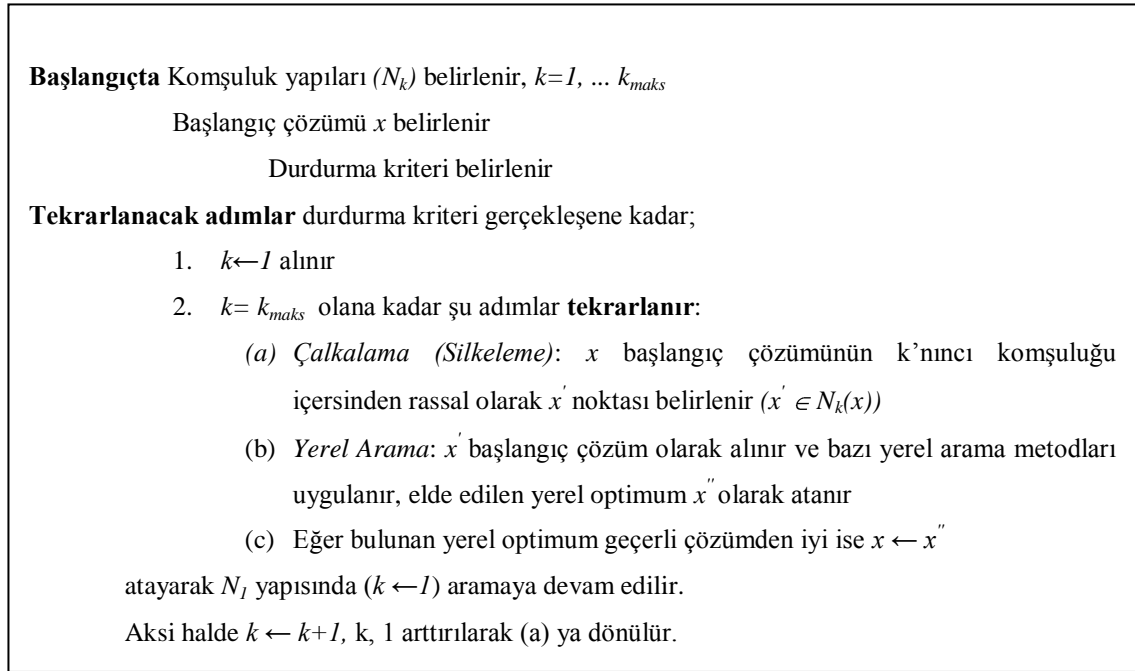
(3.2) ifadesinde $N(x) \subset X$ olmak üzere $\forall x \in X$, x 'in komşuluk kümesini ifade etmektedir. Komşuluk arama yöntemleri, geçerli çözümü daha iyi bir komşusuna taşıyan yöntemlerdir ve yerel bir minimuma ulaştığında bu komşular tutulmaktadırlar. Birkaç metasezgiselde yerel minimumda takılı kalmaktan kaçınmak için bu tablo genişletilmiştir. Bunlardan en iyi bilinenleri genetik arama, tavlama benzetimi ve tabu aramadır. Değişken komşuluk arama hem yerel (lokal) minimuma ulaşma hem de bunları bulduran alandan kaçma amacıyla komşuluk değişimleri fikrini sistematik olarak kullanan son metasezgiseldir.

Diğer metasezgisellerden farklı olarak, temel değişken komşuluk arama yaklaşımının düşüncesi basittir ve bu düşünce birkaç parametreye ihtiyaç duymakta, hatta bazen hiç duymamaktadır. Yöntem basitliğine rağmen etkilidir. Değişken komşuluk arama yaklaşımı sistematik olarak aşağıdaki düşüncelere dayalıdır:

- Bir komşuluk yapısına göre lokal minimum olan, başka bir komşuluk yapısına göre de yerel minimum olmak zorunda değildir
- Bir global optimum tüm komşuluk yapılarına göre yerel optimumdur.
- Bir çok problem için yerel minimum, bir veya birkaç komşuluğa göre birbirlerine yakındır (Hansen ve diğ., 2006).

3.1.1. Değişken Komşuluk Arama Algoritması

Yerel arama fonksiyonu ile, yerel komşuluk içerisinde gelişmiş bir çözüm araştırılır, silkeleme (shake) ise başka bir yerel komşuluğa geçerek çözümü çeşitlendirir. Silkeleme fonksiyonu arama uzayının başka bir bölgesine, yeni bir yerel aramayı orada yürütmek amacıyla değişimi sağlamaktadır (Şevkli ve Aydın, 2007). Temel değişken komşuluk arama algoritması Şekil 3.1’de gösterildiği gibidir.



Şekil 3.1: Temel değişken komşuluk arama algoritması (Hansen ve Mladenovic, 2001).

3.2. DEĞİŞKEN KOMŞULUK ARAMA ÇEŞİTLERİ

Literatür taramasında karşılaşılan başlıca değişken komşuluk arama çeşitleri şunlardır:

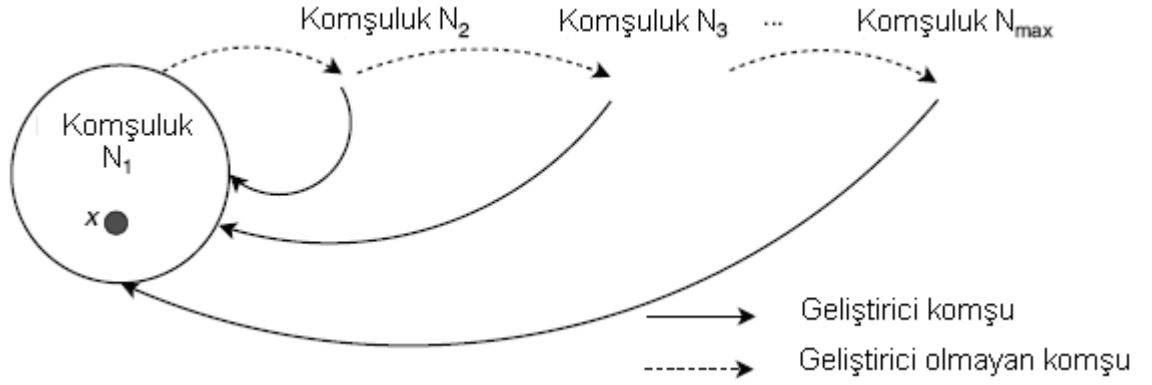
- Temel değişken komşuluk arama (Basic variable neighborhood search (TDKA))
- Değişken komşuluk azalma (Variable neighborhood descent (DKAz))
- İndirgenmiş değişken komşuluk arama (Reduced variable neighborhood search (İDKA))
- Genel değişken komşuluk arama (General variable neighborhood search (GDKA))
- Değişken komşuluk ayrıştırma araması (Variable neighborhood decomposition search (DKAA))
- Eğik değişken komşuluk arama (Skewed variable neighborhood search (EDKA))
- Paralel değişken komşuluk arama (Parallel variable neighborhood search (PDKA))

3.2.1. Değişken Komşuluk Azalma (DKAzalma)

Değişken komşuluk arama yaklaşımı “değişken komşuluk azalma”ya dayalıdır. Değişken komşuluk azalma, değişken komşuluk aramanın silkeleme aşamasının olmadığı deterministik versiyonudur (Hansen ve diğ., 2006). Aşamaları Şekil 3.2’de gösterildiği gibidir.

<p>Başlangıçta Komşuluk yapıları (N_k') belirlenir, $k=1, \dots, k_{maks}$</p> <p style="text-align: center;">Başlangıç çözümü x belirlenir</p> <p>Tekrarlanacak adımlar çözümde iyileşme elde edilmeyene kadar;</p> <ol style="list-style-type: none"> 1. $k \leftarrow 1$ alınır 2. $k = k_{maks}$ olana kadar şu adımlar tekrarlanır: <ol style="list-style-type: none"> (a) x' olarak başlangıç çözümünün en iyi komşuluk değerini belirlenmesi ($x' \in N_k'(x)$) (b) Elde edilen x' çözümü başlangıç çözümü x'ten iyi ise $x \leftarrow x'$ atayarak N_1 yapısında ($k \leftarrow 1$) aramaya devam edilir. <p>Aksi halde $k \leftarrow k+1$, k, 1 artırılarak (a) ya dönlür.</p>

Şekil 3.2: Değişken Komşuluk Azalma algoritması (Hansen ve Mladenovic, 2001).



Şekil 3.3: Değişken komşuluk azalmanın prensibi.

Şekil 3.3'te değişken komşuluk azalmanın komşuluk yapılarının değişimi gösterilmiştir (Talbi, 2009).

3.2.2. İndirgenmiş Değişken Komşuluk Arama (İDKA)

İndirgenmiş değişken komşuluk arama yaklaşımı, değişken komşuluk aramanın adımlarından sadece silkelemeyi bulunduran stokastik versiyonudur (Silkelemede rassal olarak seçimler bulunduğu için) (Perez ve diğ., 2006).

Başlangıçta Komşuluk yapıları (N_k) belirlenir, $k=1, \dots, k_{maks}$

Başlangıç çözümü x belirlenir.

Durdurma kriteri belirlenir.

Tekrarlanacak adımlar durdurma kriteri gerçekleşene kadar;

1. $k \leftarrow 1$ alınır,

2. $k = k_{maks}$ olana kadar şu adımlar **tekrarlanır**:

(a) *Çalkalama (Silkeleme)*: x başlangıç çözümünün k 'nci komşuluğu içersinden rassal olarak x' noktası belirlenir ($x' \in N_k(x)$)

(b) Elde edilen x' çözümü geçerli çözüm olan x 'ten iyi ise $x \leftarrow x'$

atayarak N_k yapısında ($k \leftarrow 1$) aramaya devam edilir

Aksi halde $k \leftarrow k+1$, k , 1 arttırılarak (a) ya dönülür.

Şekil 3.4: İndirgenmiş değişken komşuluk arama algoritması (Perez ve diğ., 2006).

3.2.3. Genel Değişken Komşuluk Arama (GDKA)

Genel değişken komşuluk aramada, basit yerel arama aşaması değişken komşuluk azalma ile yer değiştirmektedir (Talbi, 2009).

Başlangıçta Komşuluk yapıları; silkeleme için, (N_k) , $k=1, \dots, k_{maks}$ ve yerel arama için (N_l) , $l=1, \dots, l_{maks}$ olarak belirlenir.

Başlangıç çözümü x belirlenir.

Durdurma kriteri belirlenir.

Tekrarlanacak adımlar durdurma kriteri gerçekleşene kadar;

1. $k \leftarrow 1$ alınır,
2. $k = k_{maks}$ olana kadar şu adımlar **tekrarlanır**:
 - (a) *Çalkalama (Silkeleme)*: x başlangıç çözümünün k . komşuluğu içersinden rassal olarak x' noktası belirlenir ($x' \in N_k(x)$)
 - (b) *DK Azalma ile Yerel Arama*: $l=1$ alınır ve **Tekrarlanacak adımlar** $l=l_{maks}$ olana kadar;
 1. x' çözümünün en iyi komşuluk değeri x'' belirlenmesi ($x'' \in N_l(x')$)
 2. Elde edilen x'' , (x') 'ten iyi ise $x' \leftarrow x''$ ($l \leftarrow l$)
değilse $l \leftarrow l+1$, yapılarak (b) ye dönülür.

Eğer bulunan yerel optimum geçerli çözümden iyi ise $x \leftarrow x''$

atayarak N_l yapısında ($k \leftarrow l$) aramaya devam edilir

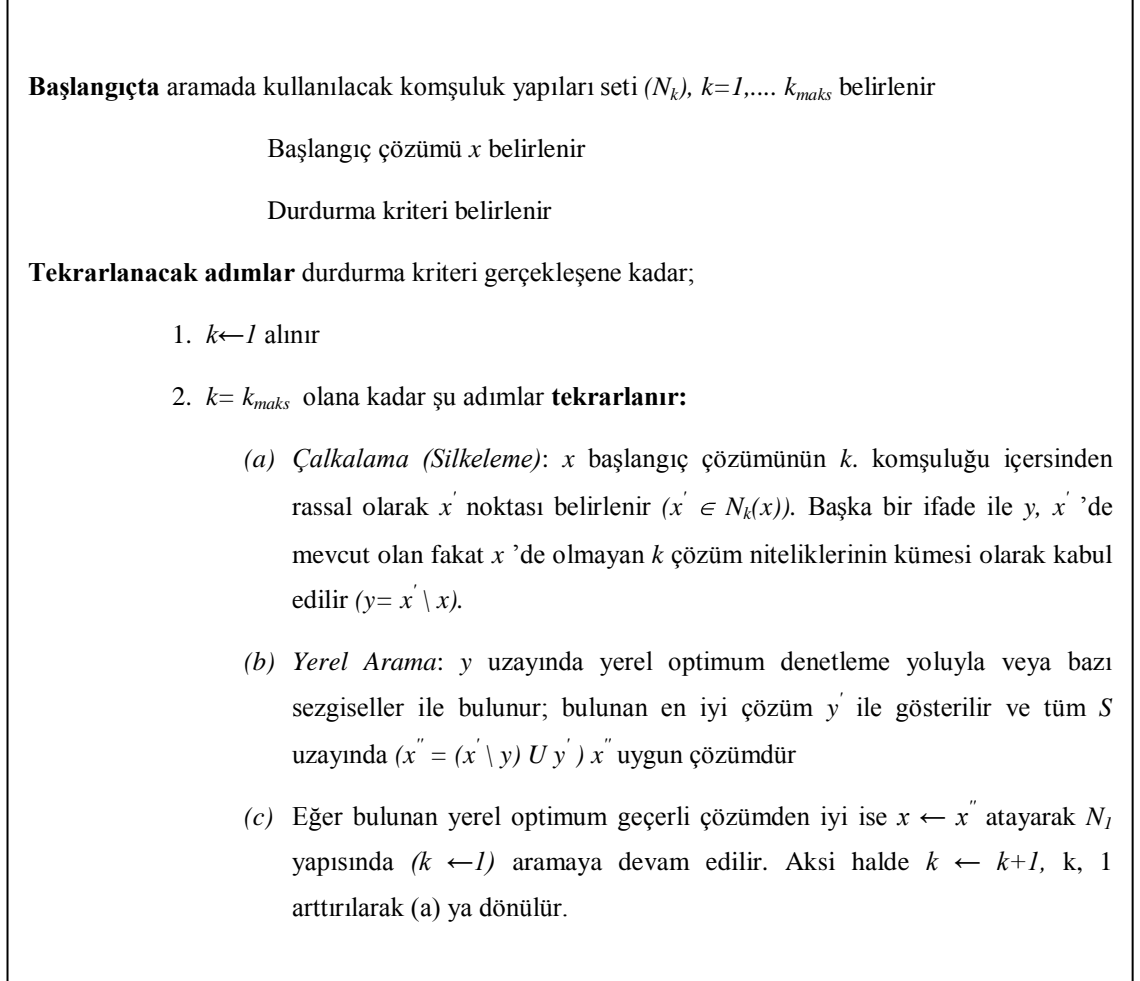
Aksi halde $k \leftarrow k+1$, k , 1 arttırılarak (a) ya dönülür.

Şekil 3.5: Genel değişken komşuluk arama algoritması. (Talbi, 2009)

3.2.4. Değişken Komşuluk Ayırıştırma Arama (DKAA)

Değişken komşuluk ayırıştırma aramada, temel değişken komşuluk arama, problemin ayırıştırılmasına dayanan iki aşamalı değişken komşuluk aramaya dönüştürülmektedir. Temel DKA ile DKAA arasındaki fark 2b aşamasındadır. x' başlangıç çözümü alınarak

tüm çözüm uzayını temsil eden S 'de yerel aramayı uygulamak yerine, DKAA'da her iterasyonda rassal seçilen sabit olmayan değişkenler uzayında tek boyutlu bir alt problem olarak çözülmektedir (Hansen ve diğ., 2001).



Şekil 3.6: Değişken komşuluk ayrıştırma arama algoritması. (Hansen ve diğ., 2001)

DKAA ilk olarak Hansen ve diğ. (2001) tarafından geliştirilmiş ve p-medyan probleminde uygulanmıştır. Şekil 3.6'da DKAA algoritma adımları verilmiştir.

3.2.5. Eğik Komşuluk Ayırıştırma Arama (EDKA)

EDKA yaklaşımına ait algoritma Şekil 3.7'de sunulmuştur.

Başlangıçta aramada kullanılacak komşuluk yapıları seti (N_k) , $k=1, \dots, k_{maks}$ belirlenir. Başlangıç çözümü x ve değeri $f(x)$ belirlenir, $x_{opt} \leftarrow x$ ve $f_{opt} \leftarrow f(x)$ olarak atanır. Durdurma kriteri ve α parametresi belirlenir

Tekrarlanacak adımlar durdurma kriteri gerçekleşene kadar;

1. $k \leftarrow 1$ alınır,
2. $k = k_{maks}$ olana kadar şu adımlar **tekrarlanır**:
 - (a) *Çalkalama (Silkeleme)*: x başlangıç çözümünün k . komşuluğu içersinden rassal olarak x' noktası belirlenir.
 - (b) *Yerel Arama*: x' başlangıç çözüm olarak alınır ve bazı yerel arama metodları uygulanır. Elde edilen ilk en iyi yerel optimum x'' olarak atanır. Eğer $f(x'') < f_{opt}$ ise; $f_{opt} \leftarrow f(x)$ ve $x_{opt} \leftarrow x''$ olarak atanır.
 - (c) *Komşuluk değişimi*: Eğer $f(x'') - \alpha \rho(x, x'') < f(x)$ ise; $x \leftarrow x''$ ve $k \leftarrow 1$ olarak atanır. Aksi halde $k \leftarrow k+1$, $k, 1$ yapılarak (a) ya dönülür.

Şekil 3.7: Eğik komşuluk ayırıştırma arama algoritması. (Hansen ve diğ., 2010)

3.2.6. Paralel Komşuluk Ayırıştırma Arama (PDKA)

Literatürde seri uygulamaların yanında hesaplama yükünü karşılamak amacıyla paralel algoritmalar da kullanılmaktadır. Paralel işlemci teknolojisini kullanarak algoritmanın veriminin artırılması ve kaynakların etkin kullanılabilmesi sağlanmaktadır.

İşlemci tabanlı paralelleştirme çözümü, düşük maliyetli bilgisayarların bir ağ yardımıyla birleştirilmesi sonucunda çok yüksek işlem güçlerine ulaşılabilmesini temel almaktadır. Foton haritalama yöntemi, ışın doğrulama algoritmasını temel aldığından dolayı paralelleştirmeye uygun bir yapıyı olanaklı kılmaktadır (Yavuz ve diğ., 2008). Şevkli ve Aydın'ın (2007) deneysel çalışmalarında dört farklı PDKA uygulaması incelenmiştir. Etkili bir uygulama geliştirmek için önce çeşitli DKA

algoritmalarının deneysel sonuçları elde edilmiştir. Ardından paralelizasyon uygulanmıştır. Komşuluk yapılarını kullanarak silkeleme ve yerel arama fonksiyonlarının konfigürasyonları ile birbirinden farklılık gösteren yedi adet DKA algoritması incelenmiştir. Amaç, uygulamanın etkinliğini artırmaktır. Sonuçlar üzerinde belirli bir fonksiyon ile başlamanın etkisi olup olmadığını anlamak için sıralamalar değiştirilerek uygulamalar gerçekleştirilmiştir.

3.3. ARAÇ ROTALAMA VE DEĞİŞKEN KOMŞULUK ARAMA

Bu başlık altında araç rotalama problemlerinde değişken komşuluk arama yaklaşımının uygulandığı çalışmalar hakkında bilgi verilmiştir. Bunun için yaklaşımın uygulanmasında hangi komşuluk yapılarının kullanıldığı ve problemi çözmek için hangi aşamaların uygulandığı üzerinde durulmuştur.

3.3.1. Teorik Çalışmalardan

Fleaser ve diğ. (2009) açık araç rotalama probleminde değişken komşuluk arama metasezgiselini kullanmıştır. Açık araç rotalama için ilk çalışma Saviklis ve Powell' in iki aşamalı uygulamalarıdır. Aşamalardan ilkinde önce müşteriler öbeklere ayrılmıştır. Kapasite kısıtı (en az öbek, örneğin en az araç) doğrultusunda, sonraki aşamada her öbek için en düşük kapsayan ağaç (minimum spanning tree) uygulanmıştır. Araç rotalama problemlerinin açık araç rotalama çeşidinde depoya dönme şartı yoktur. Lojistik firmalarının isteği ile bu problem geliştirilmiştir. Problemden talep ve koordinatları bilinen müşteriler bulunmaktadır. Araçlar aynı kapasitededir. Problemden belirlenen amaçlar araç sayısının ve katedilen yolun minimizasyonudur. Bu makalede önce en düşük araç sayısı bulunup, daha sonra en kısa yol amacıyla problem çözülmüştür. LB_1 ve LB_2 (Lower Bound) araç sayısı değeri olarak gösterilmiştir.

$$LB_1 = \sum_i^n L_i / L_{maks}$$

$$LB_2 = (MST + \sum_i^n u_i) / T^{enbüyük}$$

L_i : i müşterisine ait talep değeri. L_{maks} : araç kapasitesi.

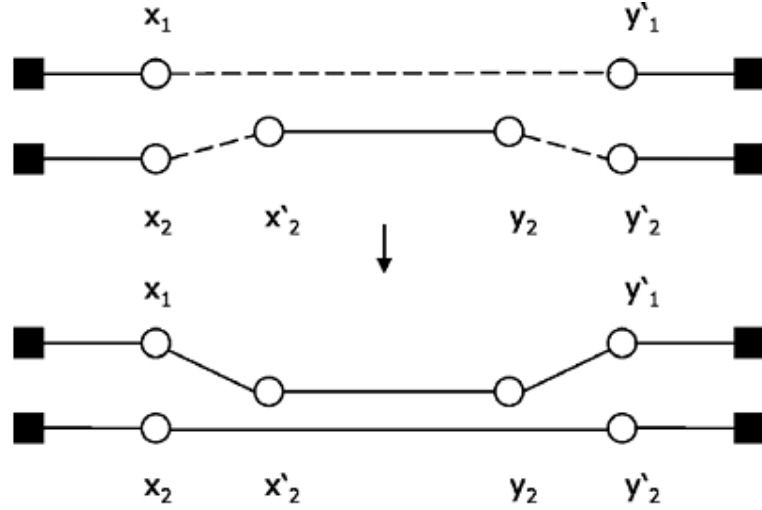
n : müşteri merkezi sayısı, m : araç sayısı.

MST: Minimum kapsayan ağaç (minimum spanning tree). u_i : i .müşteri için boşaltma zamanı (unloading time).

Minimum araç sayısı olarak “ m ”, LB_1 ve LB_2 'nin maksimum olanının seçimiyle belirlenmektedir. Belirlenen m adet araç için “en uygun talep azalışı (Best fit decreasing demand (BFDD))” ile a_0 başlangıç çözümü bulunmaktadır. Bu yöntemde önce her araç için boş rota alınmakta sonra müşteriler rotalara atanmaktadır (Fleaser ve diğ., 2009).

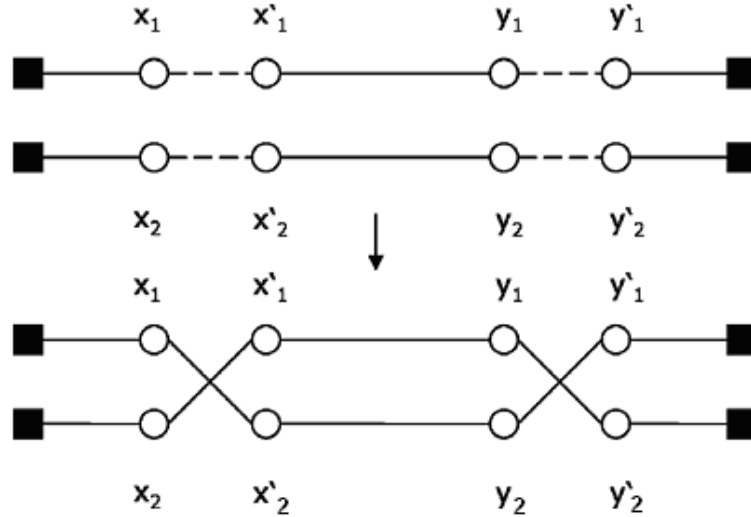
Fleaser ve diğ., (2009) DKA yaklaşımı uygulanmasının silkeleme aşamasında “ k ” rassal bölümünün seçimini gerçekleştirmişlerdir. Öncelikle rassal olarak iki rota seçilmekte ve rassal olarak ilk rotadan rassal eleman sayısındaki bölüm seçilmektedir. Daha sonra, prosedür rassal sırada ikinci rotada numaralandırılmaktadır. Bu işlem araç kapasitesi kısıtını aşmadan başarılı bir şekilde değiştirilene kadar devam ettirilmektedir. Eğer başarılı bir değişim bulunamaz ise prosedür başarılı olması için gerekli sayıda tekrar edilmektedir. Yerel optimizasyon aşamasında ise, bir rota içerisinde bir bölümü ters çevirme ve iki rota arasında bölümlerin değiştirilmesi hareketine dayanmaktadır.

Hemmelmayr ve diğ. (2009) periyodik araç rotalama probleminde DKA metasezgiselini kullanmışlardır. Periyodik araç rotalama probleminde, verilen bir zaman içerisinde talepleri belli müşterilere tekrarlı ziyaretler gerçekleştirilmektedir. Çalışmada diğer araç rotalama problemi çeşitlerine DKA'nın başarı ile uygulandığı belirtilmiştir. Başlangıç çözümünü elde etmek için her müşteri rassal olarak bir ziyaret günü kombinasyonuna atanmıştır. Rotalar her gün için Clarke ve Wright Saving Algoritması kullanılarak oluşturulmuştur. Clarke ve Wright Saving Algoritması iki rotanın uygun olarak birleştirilemediği yerde sonlandırılmaktadır. Başlangıç çözümünden, ilk komşuluktan bir çözümün rassal olarak seçilmesi ile silkeleme uygulanmıştır. Daha sonra tekrarlı iyileştirme algoritması bu işlemi takip etmiştir. Bu yineleme yeni geçerli çözüm bulunana kadar tekrar ettirilmiş, yeni çözüm bulunamayınca da sonraki komşuluğa geçilerek silkeleme aşaması ve ardından yine iyileştirme algoritması uygulanmıştır. Silkeleme aşamasında, araç rotalama için “taşımaya (move)” ve “çapraz değiştirme (cross exchange)” operatörlerine dayanan iki popüler ve etkili komşuluk kullanılmıştır. Her komşuluk geçerli çözümü bozmak ile geçerli çözümün iyi parçalarını korumak arasındaki dengeyi kurabilmelidir.



Şekil 3.8: Taşıma operatörü. (Hemmelmayr ve diğ., 2009)

Taşıma operatöründe, bir rotanın bir bölümü başka bir rotaya taşınmaktadır. Şekil 3.8'de x'_2 ve y_2 müşterileri bir rotadan diğer rotaya taşınmıştır.



Şekil 3.9: Çapraz değiştirme operatörü. (Hemmelmayr ve diğ., 2009)

Çapraz değiştirme operatöründe iki farklı rotadaki iki bölüm değiştirilmektedir. Şekil 3.9'da ilk rotadan x'_1 ve y_1 müşteri bölümü diğer rotadaki x'_2 ve y_2 müşteri bölümü ile değiştirilmiştir.

Polacek ve diğ. (2008) çok depolu zaman pencereci araç rotalama problemi için DKA metasezgiselini uygulamış ve yirmi problemten on bir tanesi en iyi çözümü sağlamıştır.

Polacek ve diğ.'nin (2004) yayınlarında ise çok depolu zaman pencereli araç rotalama problemi (ÇDZPARP) için DKA'nın paralelleştirilmesi incelenmiştir. 2004 yılındaki yayında ayrıca tırmanma hareketlerine izin verilmiştir.

Aşağıda Polacek ve diğ.'nin (2008) ÇDZPARP çözümü için DKA'nın her aşamasını uyarlamaları anlatılmıştır. Bu anlatımda başlangıç çözümünün yapım aşaması, silkeleme aşaması, yerel arama metodu ve kabul kriteri bulunmaktadır.

1) *Başlangıç çözümünde:* İlk olarak her müşteri en yakın depoya atanır. Sonra bir depo ile ilişkilendirilmiş tüm müşteriler zaman pencerelerinin artan ortalamalarına göre sıralanır. Son olarak, tüm rotalar için rotanın sonuna ön siparişli müşterilerin eklenmesiyle rotalar hazırlanmıştır. Başlangıç çözümünün uygun olması gerekli değildir, DKA'nın sonraki yineleme sürecinde bunun üstesinden gelinecek ve fizibil çözüm yolu bulunacaktır.

2) *Silkeleme aşamasında:* Silkeleme için kullanılan komşuluk yapısı kümesi DKA'nın özüdür. Yerel optimumdan kaçınma ve etkinlik arasında denge bulmak esas zorluktur. Mevcut çözüm için komşu tanımlamada, bir uygun fonksiyon veya operatör belirlenmiş olmalıdır. Komşuluk operatörünün mevcut çözümü yeterince silkelemeye izin vermesi, aynı zamanda yeni çözümün mevcut çözümün önemli parçalarını elde tuttuğundan emin olunması esas konudur. Burada kullanılan operatör 1997 yılında geliştirilmiş olan “çapraz değişim”dir. Temel fikir farklı rotalardaki iki segmentin seçilerek değiştirilmesidir. 2004 yılındaki yayına kıyasla, aynı rotayı iki kere seçmek bu yayında mümkündür. Bräysy (2003) tarafından çapraz değişim operatörünün bir uzantısı geliştirilmiştir. Bu uzantıda tersine çevirme bulunmaktadır ve adına “tersine çapraz değişim (inverted cross exchange)” denilmiştir. Bu makalede de hem çapraz değişim, hem de tersine çapraz değişim kullanılmıştır. Çalışmada komşuluk yapıları iki gruba ayrılmış, ilk yarısı ait olunan depodaki rotaları göz önüne alırken diğer yarısı farklı iki depodaki rotaları göz önüne almıştır. İlk altı komşuluk yapısında, rotadaki sıralı müşteri sayısınca değişim yapılmaktadır. Daha ayrıntılı açıklanırsa, ilk komşuluk yapısında bir müşteri değiştirilir (exchange), ikinci komşuluk yapısında sıralı olarak iki müşteriye kadar uzunlukta değiştirilir, komşuluk yapısı beşe kadar beş müşteri sırayla ele alınmaktadır. Altıncı komşuluk yapısında tüm rota müşterilerini başka rotadaki

müşterilerle değiřtirmek mümkündür. Kalan altı komřuluk yapısı da iki depo arasındaki rotaları göz önüne almaktadır.

3) *Yinelemeli iyileřtirme ařaması:* Polacek ve diğ. (2008) “yinelemeli iyileřtirme (iterative improvement)” olarak adlandırdıkları yerel arama ařamasında 3-opt komřuluk yapısının kullanımı ile sınırlandırma yapmışlardır (değışebilecek sıranın uzunluđu üst sınır değeri 3 ile sınırlanmıştır). Ayrıca, müşteriler zaman penceresine sahip olduđu için, sınırlı versiyon 3-opt’un, olası zaman penceresi ihlalleri ve çalışma zamanı bakımından faydalı olduđu belirtilmiştir. Her silkeleme ařamasından sonra, sadece değışen iki rota tekrar optimize edilmeye ihtiyaç duymaktadır. Yinelemeli iyileřtirme ařamasında, ilk iyileřtirme stratejisi uygulanmaktadır.

4) *Kabul kararı:* (2) ve (3) prosedürleri gerçekteřtikten sonra elde edilen sonuç mevcut çözümlerle kıyaslanır.

Polacek ve diğ.’nin 2004 yılında hazırladıkları yayında da ÇDZPARP ele alınmıştır. Bu makalede V_1 ’den V_m ’e kadar depolar ve V_{m+1} ’den V_{m+n} ’e kadar müşteriler belirtilmiştir. Durdurma kriteri; en fazla izin verilen CPU zamanı, en fazla yineleme sayısı veya iki gelişme arasındaki en fazla iterasyon sayısı olabilmektedir.

Heterojen araç filosu rotalama problemi, DKA’nın bazı uyarlamaları kullanılarak incelenmiştir. İmran ve diğ. (2009) bu problem üzerinde çalışma yapmışlardır. Bu çalışmada başlangıç çözümü maliyet tabanlı Dijkstra’nın algoritması ile elde edilmiş, ağ ise süpürme algoritması ve 2-opt ile yapılmıştır. Bu problem için uyarlanan birkaç komřuluk yapısı ve DKA’ya ilave olarak, çeřitlendirme prosedürü kullanılmıştır. Literatürdeki veri testleri üzerinde uygulandıđında rekabet edebilen, yeni en iyi sonuçlar sunmuştur. Ayrıca daha büyük veri setleri de ilerideki çalışmalarda kıyaslama için kullanılmak üzere oluşturulmuştur. İmran ve diğ. (2009) yerel arama setini “arıtma prosedürleri” olarak adlandırmıştır. Bu prosedürde x başlangıç çözümü alınır ve x_{best} , global en iyi olarak atanır. Başlangıç çözümü üç ařamada elde edilmiştir. Bu ařamalar şunlardır:

- a) Süpürme (sweep) algoritması kullanılarak büyük bir tur yapılması,
- b) Daha sonra bu turun 2-opt kullanılarak geliştirilmesi,

- c) Maliyet ağı oluşturulması ve daha sonra Dijkstra'nın algoritmasının uygun optimum filo büyüklüğünü bulmak için uygulanması.

Bräysy (2003) zaman pencereli araç rotalama problemi (ZPARP) için DKA yaklaşımını kullanmıştır. Son yıllarda yayımlanan çoğu ZPARP sezgiselleri iki aşamalı yaklaşımlardır. İlk aşamada, yapı sezgiseli uygun bir başlangıç çözümü geliştirmek için kullanılmakta, ikinci aşamada ise başlangıç çözümüne bir iyileştirici sezgisel uygulanmaktadır. Söz konusu rota geliştirici metotlar daha iyi komşu çözümler için yinelemeli olarak yerel aramaların yapılması suretiyle geçerli çözümü değiştirmektedir. Genellikle bir komşuluk, k bağlantı alt kümesinin değiştirilmesi ile, geçerli olandan ulaşılan çözümler kümesini içermektedir. Bu makalede dört aşamalı bir yaklaşım kullanılmıştır. İlk aşamada parametre değerlerinin farklı kombinasyonları ile yapısal sezgisel kullanılarak birkaç başlangıç çözümü oluşturulmuştur. İkinci aşamada zincir atma (ejection chain), üçüncü ve dördüncü aşamalarda ise değişken komşuluk azalma uygulanmıştır. Hansen ve Mladonavic tarafından 2000 yılında sunulan değişken komşuluk azalma ile bu makale arasında üç temel farklılık bulunmaktadır. İlki, en iyinin kabulü stratejisi yerine elde edilen ilk stratejinin kabulünün geliştirme prosedürlerinde kullanılmasıdır. Diğer farklılıklar ise değişen parametre değerleri ve geliştirme prosedürüne kullanılan amaç fonksiyonlarıdır. Sunulan şema deterministiktir prosedür ve hiçbir gelişme bulunamayana kadar uygulanmaktadır.

Kytöjoki ve diğ. (2007) kapasiteli araç rotalama probleminde DKA uygulamıştır. Amaç, en düşük maliyetli rotaları belirlemektir. Çok büyük boyutlu gerçek veriler ile uygulama (yirmi bin müşteriye kadar) yapılmıştır. Bu makalede DKA iki aşamalı uygulanmıştır. İlk aşamada başlangıç çözümü; “en ucuz ekleme sezgiseli” ile birleştirilmiş olan yedi geliştirici sezgisel küme ile oluşturulmuştur. İkinci aşamada, başlangıç çözümünü geliştirmek için yine aynı yedi sezgisel DKAzalma'ya göre hiçbir gelişme bulunamayana kadar uygulanmıştır. Yedi geliştirici sezgiselin üç tanesi rota içi (2 – opt, or-opt, 3- opt) ve dört tanesi rotalar arası (değiştirme, yer değiştirme (relocate), 2-opt, çapraz değişim (cross-exchange)) olarak belirlenmiştir.

Crispim ve Brandao (2001) geri toplamalı araç rotalama probleminde (GTARP) DKAzalma metodunu uygulamışlardır. GTARP'de sadece müşterilere malzeme

dağıtılması yapılmamakta, aynı zamanda müşterilerden malzemeler de toplanmaktadır. GTARP’de klasik araç rotalama probleminin kısıtlarının yanı sıra, geri toplama müşterileri de vardır. Tüm dağıtımlar bitirilmeden toplama yapılmaya başlanması araç kapasitesinde, ... vs. sorun oluşturmaktadır. Geri toplama müşterilerinin boş kalması durumunda, problem klasik araç rotalama problemi halini almaktadır. Crispim ve Brandao’nun (2001) makalesinde iki farklı metasezgisele dayanan iki algoritma sunulmuştur. Metasezgiseller; etkin tabu arama (ETA) ve DKAzalmazdır. DKAzalma’da da ETA’da da başlangıç çözümü süpürme algoritması ile elde edilmiştir. DKAzalma’da kullanılan beş komşuluk yapısı ise şunlardır:

- 1- Ekleme (insertion) prosedürü,
- 2- Değişirme (change) prosedürü,
- 3- Değişirme ve ekleme prosedürlerinin kombinasyonları,
- 4- Ekleme prosedürlerinde ağırlık merkezi metodunun eklenecek müşteri seçiminde kullanımı,
- 5- Sadece geçerli çözümün doğrudan komşularını değil, ayrıca en iyi doğrudan komşuların da komşularını inceleyen arama prosedürleri.

3.4. ÇOK AMAÇLI DEĞİŞKEN KOMŞULUK ARAMA

Bu bölüm altında çok amaçlı problemlerde değişken komşuluk arama yaklaşımının uygulamaları incelenmiştir. Bilimsel yayın veri tabanlarında yapılan taramalar sonucunda çok amaçlı problemlerde değişken komşuluk arama yaklaşımı uygulamalarının az sayıda olduğu görülmüştür. Liang ve diğ. de (2009a) değişken komşuluk arama yaklaşımının tek amaçlı kombinatoriyal problemlerin çözümünde çok sayıda başarılı uygulamaya sahip olduğunu, fakat çok amaçlı problemlerdeki uygulamaların sınırlı sayıda olduğunu ifade etmişlerdir. Gerçekte, iki veya daha fazla çakışan amacın aynı anda göz önüne alınması yöneticilerin veya tasarımcıların bu konudaki taleplerini karşılamak için gittikçe daha çok gerekli görülmektedir (Liang ve Lo, 2010). Çok amaçlı değişken komşuluk arama (ÇADKA) yapısı ilk olarak Geiger (2004) tarafından makina çizelgeleme problemi üzerinde uygulanarak sunulmuştur (Liang ve diğ., 2009a)

3.4.1. Çok Amaçlı Değişken Komşuluk Arama Algoritması

Geiger'in (2004) hazırladığı algoritma şöyledir:

<p>Adım 1: Başlangıç çözümü x belirlenir</p> <p>Adım 2: Pareto optimal kümesini ifade eden $P^{approx} = \{x\}$ çözümü alınır.</p> <p>Tekrarlanacak adımlar</p> <p style="padding-left: 40px;">Adım 3: Henüz incelenmemiş bir $N(x)$ komşuluğu için $x \in P^{approx}$ olan x seçilir.</p> <p style="padding-left: 40px;">Adım 4: x''in komşu $N(x)$ çözümü üretilir.</p> <p style="padding-left: 40px;">Adım 5: Tüm $x' \in N(x)$ için P^{approx} kümesi güncellenir.</p> <p style="padding-left: 40px;">Adım 6: Eğer, $x \in P^{approx}$ ise, $N(x)$ incelenmiş olarak işaretlenir.</p> <p>Sonlandırma: İncelenecek $N(x)$ ile $x \in P^{approx}$ buldukça adımlar (3)'ten itibaren tekrarlanır.</p>

Şekil 3.10: ÇADKA algoritması. (Geiger, 2004)

Yukarıdaki algoritmaya göre, komşu çözümler rassal olarak seçilen bir başlangıç çözümünden iyileşme elde edilmeyene kadar üretilmektedir. P tane alternatiften oluşan yaklaşık Pareto kümesinin hedefi, yaklaşık küme olan P^{approx} 'un arama esnasında korunmasını sağlamaktır. P^{approx} 'un güncellenmesi, Pareto baskınlık ilişkisine göre gerçekleştirilmektedir (Baskınlık ilişkisi 2.6.6. numaralı başlıkta ele alınmıştır). Bu yüzden P^{approx} sadece bastırılmamış alternatifleri içermektedir. $x' \in N(x)$ olan x' çözümleri, x çözümüne baskın x' 'i P^{approx} kümesinden çıkarmaktadır.

ÇADKA, tek amaçlı DKA algoritmasının bir uzantısı olarak çok amaçlı optimizasyon problemlerinin çözümü için kullanılmaktadır. ÇADKA'nın, tek amaçlı DKA'nın çeşitlerinden en temel farkı komşulukların rassal seçimini barındırması, temel çözüm için yeni bir seçim stratejisi ve komşuluk çözümlerini kullanmasıdır. ÇADKA algoritması uygun bir başlangıç çözümünün rassal olarak oluşturulmasıyla başlar. Bu başlangıç çözümü genelde başlangıç yaklaşık Pareto-önyüz (P^{approx}) içindeki ilk ve tek üyedir. Başlangıçta uygun bir komşuluk kümesinin de tanımlanmış olması gereklidir. Arama döngüsü, geçerli başlangıç yaklaşık Pareto-önyüz içinden temel çözümün seçimi ile başlamaktadır. Önceden tanımlanmış komşuluk yapıları havuzu içersinden, eşit olasılıkta, rassal olarak bir komşuluk seçilir. Ardından silkeleme (shaking) ve komşuluk arama (yerel arama) sırayla uygulanır. Tek amaçlı DKA algoritmasının aksine, temel çözümü güncellemek için sadece en iyi komşuluk çözümü göz önüne alınırken,

ÇADKA'da tüm komşuluk çözümleri yaklaşık Pareto-önyüz güncellemesi için göz önüne alınmaktadır. Arama prosedürü, önceden tanımlanmış durdurma kriterine kadar sürdürülür (Liang ve Lo, 2010).

3.4.2. Çizelgeleme Problemlerinde Çok Amaçlı Değişken Komşuluk Arama Uygulamaları

Geiger'in 2004 yılındaki uygulamasında makina çizelgeleme problemi ele alınmıştır. Burada iki amaç kullanılmıştır. Bunlar:

- Maksimum makina aylıklığı değerinin minimizasyonu, (C_{maks})
- Toplam makina aylıklığının minimizasyonu ($C_{toplamlam}$)'dur.

Geiger'in (2004) kullandığı komşuluk yapıları ise şu şekildedir:

- *Değiştirme (exchange) komşuluğu*: Π_j ve Π_k işlerinin pozisyonlarını deęiş tokuş etmek ($j \neq k$),
- *İleri taşıma komşuluğu*: Π_j işini kaldırarak Π_k pozisyonuna yerleştirmek ($k > j$),
- *Geri taşıma komşuluğu*: Π_j işini kaldırarak Π_k pozisyonuna yerleştirmek ($k < j$),
- *Tersine komşuluk*: seçilmiş bir Π alt setini tersine çevirmek.

Söz konusu çalışmada, C_{maks} , $C_{toplamlam}$ dışında maksimum gecikmenin (T_{maks}), toplam gecikmenin ($T_{toplamlam}$), maksimum makina aylıklığının (I_{maks}) ve toplam makina aylıklıklarının ($I_{toplamlam}$) minimizasyon optimallik kriterlerinin kümelendirilmesi ile on iki kriter oluşturularak yüz test problemi üzerinde uygulama yapılmıştır.

Geiger'in (2004) ÇADKA yaklaşımının çizelgeleme problemlerine uygulandığı başka bir çalışma Liang ve dię. (2009a) tarafından hazırlanmıştır. Bu çalışmada Tayvan'da baskılı devre kartı fabrikasından elde edilen gerçek verilere dayalı üç problem kümesine DKA ve başka sezgisel yöntemler uygulanmıştır. Problem kümelerinden ilki 35 iş-7 makina, ikincisi 50 iş 10 makina, üçüncüsü ise 65 iş ve 13 makinalıdır. Her problem kümesi her yöntem için onar kere çalıştırılmış, sonuçlar $D1_R$ ölçümüne göre minimum, ortalama ve maksimum işlem süreleri başlıkları altında gösterilmiştir. Liang ve dię.'nin (2009a) *minimum* C_{maks} (maksimum tamamlama süresi) ve *en minimum* toplam gecikme olmak üzere çelişen iki amacın kullanıldığı makalelerinde kullanılan DKA yönteminin

1. ve 3. problem kümelerinde (35 iş-7 makina ve 65 iş-13 makina) en düşük, ortalama ve en yüksek işlem süreleri kıyaslamasında tümünde diğer iki yöntemle göre daha iyi sonuç verdiği, 2. problem kümesinde (50 iş-10 makina) ise en düşük ve en yüksek işlem süreleri kıyaslamasında daha iyi sonuçlar verdiği görülmüştür. Yöntemin performansının ölçümü için $D1_R$ ölçeği kullanılmıştır. $D1_R$ 2003 yılında Ishibuchi tarafından geliştirilmiş ve en düşük Öklidyen mesafenin ortalamasının hesaplamasına dayalı bir ölçektir. DKA' da girinti (insert) ve takas (swap) olmak üzere iki komşuluk yapısı kullanılmıştır.

3.4.3. Çok Amaçlı Çizelgeleme Problemlerinde Hibrid Değişken Komşuluk Arama Uygulamaları

Çok amaçlı çizelgeleme problemlerine ilişkin literatürde başka sezgisel yöntemler ile melez yapılar oluşturmak suretiyle elde edilen değişken komşuluk arama (DKA) yaklaşımı uygulamaları da bulunmaktadır. Örneğin, Zhang ve diğ. (2010) esnek iş sıralama problemleri için değişken komşuluk genetik algoritma (DKGA) uygulamasını geliştirmişlerdir. Bu yöntem genetik algoritma (GA) ile DKA yaklaşımlarının avantajlarını barındıran melez bir yöntem olarak ifade edilmiştir. Üç problem kümesi kullanılarak yöntem test edilmiş ve diğer yöntemlerle kıyaslanmıştır. Zhang ve diğ. (2010) ile Li ve diğ. (2010a) esnek iş sıralama (atölye tipi çizelgeleme) problemlerine, DKGA uygulamasını melez bir yöntem olarak gerçekleştirmişlerdir. İki çalışmada da bulunan sonuçlar birbirine çok yakın değerlerdedir. İki yayında da üç amaç belirlenmiştir. Bu amaçlar:

- Minimum çizelge boyu (makespan) (C_m),
- Makinalardaki minimum toplam iş yükü (W_T),
- En fazla iş yüküne sahip makinanın minimum yüke (W_m) sahip olmasıdır.

Bu üç amaç fonksiyonu farklı ağırlıklar atanarak tek amaç fonksiyonu haline dönüştürülmüş ve problem iki aşamada çözülmeye çalışılmıştır. Bu aşamalar: makinalara iş atama ve işlerin sıralanmasıdır. Bu çalışmada ayrıca 2 komşuluk yapısı kullanılmıştır. Bu makalede bunlar belirtilmemiş; fakat yazarın 2008 yılı yayınına bu yapılar için atıfta bulunulmuştur.

Li ve diğ.'nin (2010b) yayını, bahsedilen Li ve diğ.'nin (2010a) yayını ile çok benzerdir. Aynı yazarların hazırladığı bu iki makalenin ilkinde (2010a) değişken komşuluk genetik algoritma (DKGA), yani DKA ile GA hibrid olarak kullanılırken, ikincisinde (2010b) DKA ile tabu arama yaklaşımları hibrid olarak kullanılmıştır. Uygulamalarda seçilen problem çeşidi ve örnek problemler ise aynıdır. Üç amaç fonksiyonu bu makalede de ağırlıklar kullanılarak tek amaç fonksiyonu haline dönüştürülmüştür. Problemden iki karar aşaması, dolayısıyla iki vektör bulunmaktadır. Makina atama, iki vektörle ifade edilmektedir. Bunlardan ilki, makinelerin kendi içinde sırasının bulunduğu bir vektör ve ikincisi operasyon çizelgeleme, yani işlerin kendi içinde sırasının bulunduğu bir vektördür. Örneğin (2,1,2) ve (1,2,2) vektörleri sırasıyla ikinci makinada birinci işin, birinci makinada ikinci işin ve ikinci makinada ikinci işin yapılacağını göstermektedir. Atama algoritması olarak tabu arama algoritması, çizelgeleme algoritması olarak ise DKA algoritması kullanılmıştır. Yazarların bahsedilen ilk makalesinde (2010a) ise atama algoritması olarak genetik algoritma kullanılmıştır.

Li ve diğ.'nin (2010b) çalışmasında üç adet girinti (insert) ve takas (swap) komşuluk yapıları geliştirilmiştir. Bu yapılar şu şekildedir:

- *Takas (swap):* x ile y 'nin değişimi,
- *Girinti I (insert I):* y 'den sonra x 'in araya alınması,
- *Girinti II (insert II):* y 'den önce x 'in araya alınması.

DKA yaklaşımının melez olarak çok amaçlı çizelgeleme probleminde bir başka uygulaması ise Adibi ve diğ.'ne (2010) ait çalışmadır. Bu çalışmada rassal iş varışları ve makina arızalarını göz önüne alan dinamik atölye çizelgeleme problemi üzerinde çalışılmıştır. Bu çalışmada herhangi bir yeniden çizelgeleme noktasında, Yapay Sinir Ağları'ndan (YSA) elde edilen ağırlıklar kullanılarak, DKA için uygun parametreler hesaplanmaktadır. Burada bahsi geçen uygun parametreler; çizelgeleme metodunun performansını önemli şekilde geliştiren parametrelerdir. Dinamik çizelgelemede, DKA'nın parametreleri, YSA tarafından güncellenmiştir. Zandieh ve Adibi'nin (2009) yaptıkları başka bir çalışmada da bu yaklaşım uygulanmıştır. Adibi ve diğ. (2010) DKA yönteminin seçilme sebebi olarak bir meta sezgisel için istenen çoğu özelliği (örneğin basitlik, verimlilik, etkenlik, genellik gibi) taşımasını belirtmişlerdir. Bu makalede

DKA'nın verimliliğini ve etkinliğini arttırmak için DKA'nın parametreleri YSA tarafından güncellenmiştir.

Dinamik çizelgeleme uygulamasında, Adibi ve diğ. (2010) iki amaç göz önüne almışlardır. Bu amaçlar şunlardır:

- *Minimum çizelge boyu (makespan)* : Bir grup işin tamamlanması için gerekli toplam süre.
- *Minimum gecikme (tardiness)* : Her işin gerçekleşmesi gereken zaman ile daha sonrasında gerçekleştiği zaman arasındaki fark.

Adibi ve diğ. (2010) işlem süresinin çok uzun sürmesinden kaçınmak amacıyla en iyi komşuluk yapısı sayısı olarak belirtilen “2”yi seçmişlerdir. Bu iki komşuluk yapısı “girinti” ve “takas” tır. Silkeleme ve yerel arama aşamalarında bu yapılar kullanılmıştır. Makalede, sunulan metodun performansını ispatlamak için on makinalı bir atölye çizelgelemesi simüle edilmiştir. Sonuçlar dört genel kural (son giren ilk çıkar, ilk giren ilk çıkar, en kısa işlem süresi, en erken teslim zamanı) ile kıyaslanmış ve DKA'nın daha iyi sonuç bulduğu görülmüştür.

Tavakkoli-Moghaddam ve diğ. (2011) ağırlıklı ortalama iş akış süresini ve toplam gecikme cezasını minimum yapmak amaçlarına sahip bir atölye tipi çizelgeleme problemini incelemişlerdir. Genetik operatörlerle DKA yaklaşımı ve sürü zekası optimizasyonu (SZO) yaklaşımları birlikte kullanılmıştır. Sunulan algortimanın performans testlerini gerçekleştirmek için test problemleri hazırlanmıştır. Çok bilinen iki evrimsel algoritma ile sonuçlar kıyaslanmış ve daha iyi sonuçlar bulunduğu görülmüştür.

Liu ve diğ.'nin (2007) makalelerinde de DKA ile SZO'nun kombinasyonunu içeren DKSZO melez metasezgiseli tanıtılmıştır. Bu makalede yöntem çok amaçlı esnek atölye çizelgeleme probleminin de çözümü için uygulanmış olup, amaç fonksiyonu olarak aşağıda yazılı iki hedef belirlenmiştir:

- *Minimum C_{toplam}* : akış süreleri toplamının minimizasyonu,
- *Minimum C_{max}* : çizelge boyunun minimizasyonu.

C_{max} 'in minimizasyonu $C_{toplaml}$ 'in maksimizasyonuna sebep olabilmektedir. Ağırlıklı kümeleme (agregasyon) en genel yaklaşımdır. Bu yaklaşıma göre bu iki amaç ($f_1 ; \min C_{toplaml}$ ve $f_2 ; \min C_{max}$) ağırlıklandırılmış olarak toplanmaktadır. Bu çalışmada üç tane SZO algoritması sunulmuştur. Bunlar;

- Standart SZO
- DKSZO
- Çok başlangıçlı SZO (ÇBSZO)'dur.

DKSZO ile ÇBSZO arasında sonuçlarda kıyaslama yapılmış. DKSZO algoritmasının özellikle büyük ölçekli problemlerde etkili olduğu sonucuna varılmıştır.

Liu ve diğ.'nin 2006 yılında hazırladıkları çalışmada, 2007 yılına ait çalışmadan farklı olarak işlem başına ortalama makina sayısını ifade eden λ ölçekleme katsayısı kullanılmamıştır. Her iki çalışmada da DKA yaklaşımı ile SZO yaklaşımından oluşan melez DKSZO uygulanmıştır.

3.4.4. Atama Problemlerinde Çok Amaçlı Değişken Komşuluk Arama Uygulamaları

Bir başka çok amaçlı değişken komşuluk arama (ÇADKA) uygulaması Liang ve diğerleri (2009b) tarafından gerçekleştirilmiştir. Bu çalışmada kaynak atama probleminde iki çakışan amaç kullanılmıştır. Bunlar; maliyet minimizasyonu ve etkinliğin maksimizasyonudur. Bu çalışmada Geiger'in (2004) çok amaçlı değişken komşuluk arama (ÇADKA) yaklaşımı kullanılmıştır. Liang ve diğ.'nin (2009b) uygulamasında da ÇADKA yine bir başlangıç çözümünün oluşturulmasıyla başlamaktadır. Daha sonra yerel arama için rassal olarak bir komşuluk yapısı seçilmekte, komşu çözüm oluştuğunda da baskın olmayan çözümler güncellenmektedir. Komşuluk aramayı tamamlayınca, yeni temel çözüm baskın olmayan bu kümeden rassal olarak seçilmekte ve süreç tekrarlanmaktadır. Durdurma kriteri gerçekleşene kadar bu süreç devam ettirilmektedir. Kaynak atama probleminde yer alan "k. komşuluk"; bir işteki işçi sayısını azaltıp başka işe atama şeklinde uygulanmıştır. Üç test kümesinde de ÇADKA yaklaşımı uygulanmıştır. $D1_R$ ölçeği yine kıyaslama için kullanılmıştır. Sunulan yöntemde, karınca kolonisi algoritması ve genetik algoritma ile rekabet eden sonuçlar elde edilmiştir.

Liang ve Lo 2010'daki çalışmalarında Geiger'in (2004) ÇADKA yaklaşımını uygulamışlardır. ÇADKA'nın aşamaları daha detaylı olarak şu şekilde incelenebilir:

1) Başlangıç çözümü oluşturma

Metasezgiselerde başlangıç çözümü oluşturulmasında en genel yaklaşım, başlangıç çözümünü rassal olarak oluşturmaktır. Kısıtlı optimizasyon problemleriyle uğraşırken, başlangıç çözümü yöntemin etkinliğinde önemli bir rol oynar. Liang ve Lo'nun bu çalışmasında sadece uygun başlangıç çözümlerine izin verilmiştir.

2) Yaklaşık Pareto önyüzün güncellenmesi

Tüm amaçlar eş zamanlı olarak göz önüne alındığında tek bir global optimum bulunamayacağı için, tek amaçlı optimizasyon problemlerindeki optimallik fikri doğrudan olarak çok-amaçlı optimizasyon problemlerine uygulanamamaktadır. Örneğin sistem güvenilirliğinin maksimizasyonu (R_s) ve sistem maliyetinin (C_s) minimizasyonunun amaç olarak belirlendiği bir bolluk atama optimizasyonu (redundancy allocation optimization) için; y^1 çözümü başka bir y^2 çözümüne göre daha fazla sistem güvenilirliği fakat daha yüksek sistem maliyetine sahipse, y^1 ve y^2 birbirine üstün değildir, ya da y^1 ve y^2 birbirine baskın değildir. Eğer y^1 herhangi bir amaç açısından y^2 'den daha iyi, ve diğer amaç için de eşit veya daha iyi ise çözüm y^1 çözüm y^2 'ye baskındır. Arama uzayında bulunan herhangi bir çözüm tarafından bastırılmamış çözümler baskın olmayan (non-dominated) çözümler olarak adlandırılır. Bundan dolayı, çok amaçlı bir optimizasyon probleminde optimal çözümler genelde baskın olmayan çözümlerin bir kümesinden oluşmaktadır, mesela buna örnek olarak yaklaşık Pareto-önyüz verilebilir (Liang ve Lo, 2010).

ÇADKA'da, yaklaşık Pareto-önyüz başlangıç çözümü gibi bir "tek" üye ile başlamaktadır. Daha sonra, komşuluk arama esnasında önyüz güncellenmektedir. Önyüz güncellenirken tüm komşuluk çözümleri kontrol edilmektedir.

Rezgui ve diğ. (2010) çoklu-radyo kablosuz örgü ağlarında kanal atama problemi üzerinde çalışma yapmışlardır. Çoklu-radyo kablosuz örgü ağlarında genellikle, çakışan yayınlar ve parazit aralıklar içinde aynı kanalı paylaşan çeşitli bağlantılar (link) bulunmaktadır. Dört amaç belirledikleri problemde DKA yaklaşımını uygulamışlardır.

Kullanıcının frekans deęiřtirmesini minimum, yükü dengelemek için trafik yükü sapmalarını minimum, genel verimi maksimum ve örgüdeki müşteriler arasında eşitlięi maksimum kılmaktır. Tek amaçlı fonksiyonlara kıyasla çok amaçlı fonksiyonların deęerlendirme maliyeti çok yüksektir, bu yüzden problem çözümü için uygulanacak sezgisel dikkatli seçilmelidir (Rezgui ve dię., 2010). Bu makalede basitlik için, dört amaç fonksiyonu bir amaç fonksiyonunda kümelenmiř şekle dönüřtürülmüřtür. Gözüpek ve Genç (2009), hücresele aęlardaki kanal planlaması probleminini, frekansların baz istasyonlarına giriřim kısıtlarını bozmayacak ve hücrelerin trafik gerekliliklerini karşılayacak şekilde tayin edilmesi ve tekrar kullanılması şeklinde tanımlamıřtır.

Son olarak hibrid bir çok amaçlı problemde uygulama için incelenen bařka bir yayın Xu ve Qu'ya (2010) aittir. Bu çalıřmada evrimsel çok amaçlı tavlama benzetimi algoritmasını deęiřen komřuluk yapısıyla birlikte kullanmıřtır. Telekomunikasyondaki çok amaçlı çoklu yayın rotalama problemi üzerinde uygulama gerçekleřtirilmiřtir. Özaslan ve Saka (2004) GSM'de frekans atama problemini tanımlarken "*iletiřime etki edebilecek miktarda giriřim, aynı frekansı ya da komřu frekansı kullanan verici çiftleri arasında meydana gelebilir. Ayrım gereksinimi, elektromanyetik spektrumda, vericilere atanan frekanslar arasında olması gereken minimum uzaklıęı belirtir.*" ifadelerini kullanmıřtır. Xu ve Qu (2010) baęlantı aktarma ve düęüm aktarma olmak üzere iki komřuluk operatörü kullanmıřtır. Bulunan sonuçlar, çok amaçlı tavlama benzetimi (EMOSA) metoduyla bulunan sonuçlarla kıyaslanmıř ve sunulan algoritmanın test edilen problemlerin tümünde daha yüksek sayıda baskın olmayan çözümler elde ettięi görülmüřtür.

3.5. GELİřTİRİLEN YÖNTEM

Bu tezde çok amaçlı araç rotalama probleminin (ÇARP) çözümü için, Geiger'in 2004 yılında ilk defa ortaya attıęı çok amaçlı deęiřken komřuluk arama (ÇADKA) yapısı temel alınarak uygulama gerçekleřtirilmiřtir. Bir önceki bölümde vurgulandıęı gibi, DKA yaklařımının çok amaçlı problemler üzerinde çok az sayıda uygulamasının olduęu ve ÇARP üzerinde ise uygulamasının olmadığı belirlenmiřtir. Bu sebeple ÇARP üzerinde DKA uygulaması seçilmiřtir. DKA çeřitleri incelendięinde ise genel deęiřken

komşuluk arama (GDKA) çeşidinin uygulanmasına karar verilmiştir. Bu kararın alınmasında stokastik ve deterministik süreçleri barındırması ile iyi çözümler sağlayabilmesi etkili olmuştur. Bu tezde, Geiger'in (2004) DKA yaklaşımı için hazırladığı algoritma GDKA yaklaşımına uyarlanmış ve ÇADKA'da komşuluklar içerisinden rassal olarak seçim yapma stratejisi yerine, belirli bir sıra takip edilmiştir. Ayrıca, GDKA'nın yapısı gereği silkeleme ve yerel arama aşamalarında komşuluk yapılarının uygun sıraları belirlenmiştir. Yerel arama aşaması için program içindeki "tekrarlama" sayıları, programın çalıştırılmasıyla denemeler yapılarak en uygun şekilde belirlenmeye çalışılmıştır. Bu değişiklikler yapıldıktan sonra elde edilen yaklaşım, bundan sonra çok amaçlı genel değişken komşuluk arama (ÇAGDKA) adıyla anılacaktır. Tekrarlama sayıları ise "döngü" şeklinde ifade edilecektir.

ÇAGDKA uygulamasında klasik araç rotalama problemi (KARP) çok amaçlı olarak ele alınmıştır. KARP'nde tek bir depo, aynı kapasitede araçlar, belirli talebe sahip müşteriler bulunmaktadır. Her aracın depodan çıkıp depoya tekrar geri dönme ve her müşterinin sadece bir kez ziyaret edilme zorunluluğu vardır. Uygulamanın başarısını ölçmek için, Chand ve diğ.'nin (2010) ÇARP üzerinde uygulamış olduğu genetik algoritmanın verdiği sonuçlar ile ÇAGDKA yönteminin verdiği sonuçlar arasında karşılaştırma yapılmıştır. Chand ve diğ. (2010) de ele aldıkları statik araç rotalama probleminin modelini "her müşteri d_i talebine sahiptir, her aracın izlediği rota depoda başlayıp depoda sonlanmaktadır, her araç aynı kapasiteye sahiptir, depo ile müşteriler ve müşterilerin birbirleri ile mesafesinin belirlenmesinde öklid uzaklık kullanılmıştır, kullanılan iki kısıttan birisi araca ait kapasite kısıtı, diğeri ise her müşterinin sadece bir kere ziyaret edilmesi kısıtıdır" şeklinde ifade etmişlerdir. Karşılaştırmanın yapılabilmesi için aynı amaç fonksiyonları seçilmiştir. Bu fonksiyonlar katedilen mesafenin ve araç sayısının minimizasyonudur.

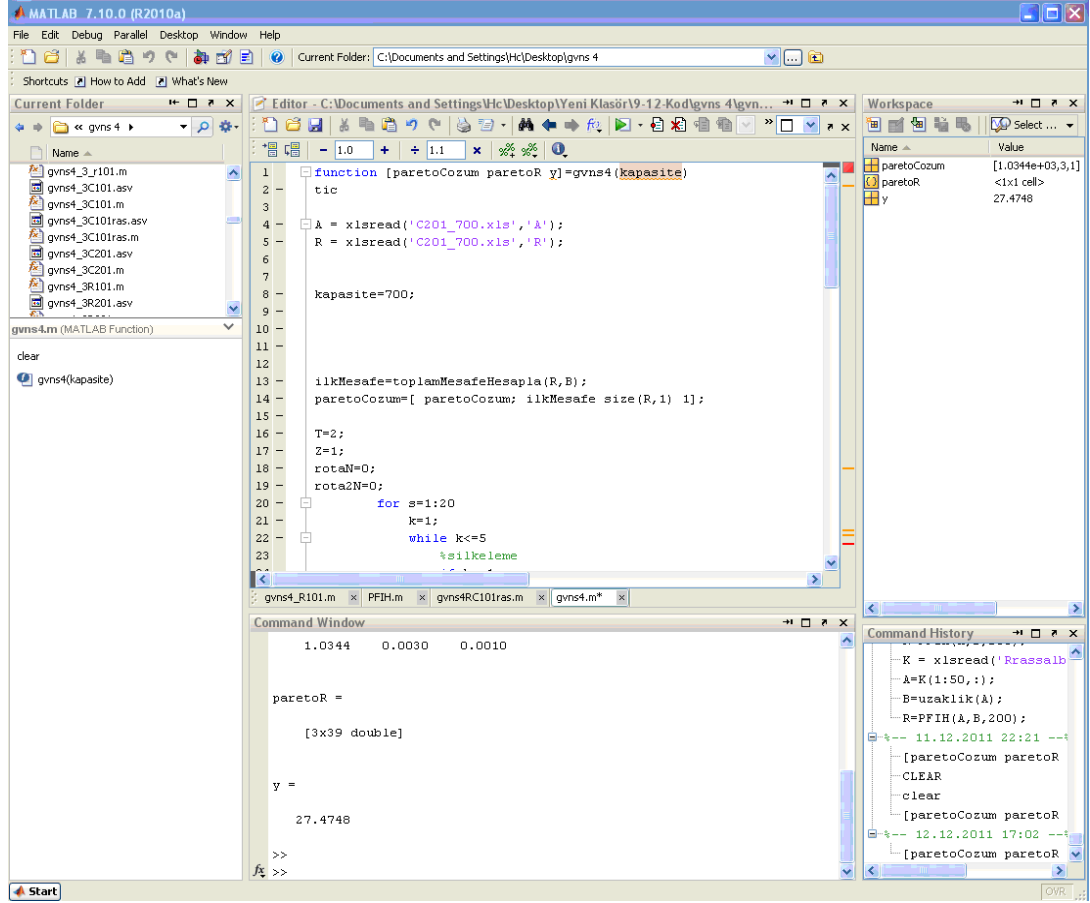
Uygulamadaki test verileri, Chand ve diğ.'nin de (2010) kullanmış olduğu Solomon'a ait verilerdir. Solomon tarafından 1983 yılında sunulan "Solomon örnekleri" en iyi bilinen karşılaştırma problem seti olarak literatürde yerini almıştır. Bu veriler Ek-A'da sunulmuştur. Tablo 3.1'de veri kümelerinin toplam talep ve kapasite değerleri verilmiştir. Toplam talep miktarlarının araç kapasitesi değerine oranı sonucu elde edilen

değer, tüm müşterilere istenen taleplerin dağıtılabilmesi için gerekli olan minimum araç (rota) sayısını vermektedir.

Tablo 3.1: Solomon test verilerinin özet tablosu

Problem kümesi	Müşteri sayısı	Toplam Talep	Araç kapasitesi	Talep / kapasite oranı	Gerekli en az araç sayısı
C101	25	460	200	2,3	3
C201	25	460	700	0,6	1
R101	25	332	200	1,6	2
R201	25	332	1000	0,3	1
RC101	25	540	200	2,7	3
RC201	25	540	1000	0,5	1
C101	50	860	200	4,3	5
C201	50	860	700	1,2	2
R101	50	721	200	3,6	4
R201	50	721	1000	0,7	1
RC101	50	970	200	4,8	5
RC201	50	970	1000	0,9	1
C101	100	1810	200	9,0	10
C201	100	1810	700	2,6	3
R101	100	1458	200	7,3	8
R201	100	1458	1000	1,5	2
RC101	100	1724	200	8,6	9
RC201	100	1724	1000	1,7	2

DKA yaklaşımı için başlangıç çözüm yönteminin seçimi ile yaklaşımın içinde kullanılan komşuluk yapılarının belirlenmesi büyük önem taşımaktadır. Bu yüzden tez çalışmasında farklı iki başlangıç çözümü yöntemi kullanılarak ayrı uygulamalar gerçekleştirilmiştir. Her iki yöntem sonucunda elde edilen değerler özet şeklinde hazırlanarak kıyaslanmış ve tablolarda sunulmuştur. Ele alınan başlangıç yöntemlerinden ilki süpürme algoritması, diğeri ise ileri itmeli ekleme sezgisel (İİES-push forward insertion heuristic) yöntemidir. Elde edilen başlangıç çözüm değerlerinin herbirine ÇAGDKA yaklaşımı uygulanmış ve iyileşmiş çözümler elde edilmiştir. ÇAGDKA algoritmasının program kodları MATLAB üzerinde hazırlanmıştır. Şekil 3.11'de program arayüzü sunulmuştur.



Şekil 3.11: GDKA algoritması için geliştirilen programın örnek ekran görüntüsü

Başlangıç çözüm yöntemleri ile elde edilen çözümler ve kullanılan komşuluk yapıları sıralanmadan önce hazırlanan kod hakkında bilgi verilmesi faydalıdır. Kod içinde sonuçlar üzerinde iyileştirme yapabilmek için GDKA komşuluk yapılarının değiştirilme durumlarını belirten döngüler kullanılmıştır. Hazırlanan kodun yapısında durdurma kriterini ifade eden en dış döngü için tekrar sayısı verilmiştir. Çeşitli tekrar sayıları denenerek problem verileri üzerinde hangi durdurma kriteri ile daha iyi sonuçlar elde edilebileceği belirlenmiştir. Başlangıç çözümüne silkeleme aşamasında uygulanan rassal özellikteki komşuluk yapısından sonra, elde edilen çözümü iyileştirmek amacıyla yerel arama aşamasına geçilmiştir. Silkeleme aşamasında beş, yerel arama aşamasında ise altı komşuluk yapısı kullanılmıştır. Komşuluk yapıları ve sıralarının belirlenmesi, Bölüm 3.7’de ele alınmıştır. Yerel aramada komşuluk yapıları belirlenen uygun sırada gerçekleştirildikten sonra, bir tekrar ile tüm bu yapıların elde edilmesinin çok iyi sonuçlar vermediği gözlenmiştir. Bu sebeple yerel arama için de bir iç döngü hazırlanmıştır. İç döngü olan yerel arama döngüsünün belirli sayıdaki tekrarı, tüm

silkeleme komşuluk yapılarına uygulandıktan sonra, durdurma kriteri sayısı bir birim arttırılmıştır. Burada iki farklı durum incelenmiştir: İlki, dış döngü sayısının yüksek tutulması (seçilen sayı 200) ve yerel arama iç döngüsünün bir kere tekrarlanması, ikincisi ise dış döngü sayısının çok fazla yüksek tutulmayarak (seçilen sayı 50) iç döngü tekrar sayısının yüksek (seçilen sayı 20) tutulmasıdır. Tezin ilerleyen bölümlerinde ilk durum için “yüksek dış döngü”, ikinci durum için ise “yüksek iç döngü” şeklinde bahsedilmiştir. Yüksek iç döngü ile, yüksek dış döngüye oranla daha iyi sonuçlar elde edilmiştir. Bu açıklamalar altında ÇAGDKA için işlem adımları Şekil 3.12’de gösterilmiştir. Bu şekilde k ile silkeleme aşamasına ait komşuluk yapıları, l ile de yerel arama aşamasına ait komşuluk yapıları ifade edilmiştir. “Adım 5” için Pareto kümesinin güncellenmesine yönelik işlem adımları Şekil 3.13’te gösterilmiştir.

Adım 1: Seçilen yöntemin çalıştırılması ile başlangıç çözümü oluşturulur.

Adım 2: Başlangıç çözümünün iki amaç fonksiyonu değeri “pareto” kümesine atılır.

Dış döngü tekrarı:
 $k=1$ olarak alınır.
 $k \leq 5$ için tekrar adımları:

Adım 3: Silkeleme aşamasının gerçekleştirilmesi için, amaç fonksiyonu değerlerine bakılmadan, geçerli çözüme k komşuluk yapısının uygulanması ile kısıtları sağlayan rassal bir çözüm üretilir.

İç döngü tekrarı (yerel arama):
 $l=1$ olarak alınır.
 $l \leq 6$ için tekrar adımları:

Adım 4: geçerli çözüme l komşuluk yapısının uygulanması ile kısıtları sağlayan çözümler üretilir.

Adım 5: l komşuluk yapısı için üretilen her çözüm, iki amaç fonksiyonu değerlerine bakılarak Pareto kümesine alınmasına veya alınmamasına karar verilir. Verilen kararlar doğrultusunda Pareto kümesi güncellenir.

Adım 6: Pareto kümesinin güncellenmesi sonucunda kümeye yeni giren çözüm değerleri varsa bunlar incelenmemiş olarak “0” indisi ile belirtilir. İncelenmemiş çözüm değerlerinden herhangi birisi rassal olarak seçilir ve yeni pareto kümesi üzerinden $l=l+1$ alınır.

Sonlandırma: Durdurma kriterine ulaşıldığında işlemler durdurulur.

Şekil 3.12: ÇAGDKA algoritması için işlem adımları

Adım 1: Yerel aramadan gelen tüm çözümleri barındıran geçerli Pareto çözüm kümesindeki eleman sayısı belirlenir ve m olarak atanır. İki amaç fonksiyonundan her ikisinin de değeri, kümedeki çözümlere ait değerlerden kötü olmayan çözümlerin atanacağı geçici bir küme belirlenir ve başlangıçta boş olarak belirlenir.

$i=1$ alınır:

$i \leq m$ için aşağıdaki adımlar tekrarlanır

Eğer: 1. ve 2. amaç fonksiyonlarının her ikisi de Pareto kümesindeki çözümlerden iyi ise baskın olan bu çözüm boş kümeye atanır.

Eğer: 1. ve 2. amaç fonksiyonlarının her ikisi de Pareto kümesindeki çözümlerden kötü ise bu çözüm incelenmeye devam edilmez, çözüme atanmaz.

Eğer: 1. ve 2. amaç fonksiyonlarının sadece herhangi birisi Pareto kümesindeki çözümlerden iyi ise bu çözüm boş kümeye atanır.

Eğer: 1. ve 2. amaç fonksiyonlarının herhangi birisi Pareto kümesindeki çözümlerden birine eşit ise, ayrıca eşit olmayan amaç fonksiyonu daha iyi ise bu çözüm boş kümeye atanır, değilse atanmaz.

Adım 2: Geçici boş kümeye alınan herhangi bir çözüm olması halinde Pareto kümesi bu değerler ile güncellenir.

Şekil 3.13: Pareto kümesinin güncellenmesi için işlem adımları

3.6. BAŞLANGIÇ ÇÖZÜMÜ AŞAMASI

3.6.1. Süpürme Algoritması İle Başlangıç Çözümünün Elde Edilmesi

İki aşamalı çözüm yöntemlerinden olan süpürme algoritmasında ilk olarak müşterilerin tümünden gelmiş olan talep toplamının araç kapasitesine oranlanması ile bulunan en düşük gerekli araç sayılarına ait veriler Tablo 3.1'den alınmıştır. Daha sonra depoyu merkez alan doğrusal bir çizgi ile “dönme” işlemi yapılarak araç kapasitesi kısıtı altında rotalar oluşturulmuştur. Süpürme algoritmasının uygulanmasında izlenen adımlar Şekil 3.14'de gösterilmiştir.

Adım 1: İlk olarak yeni bir araç dağıtım için atanır. Başlangıç değeri olarak “araç sayısı = 1” alınır.

Adım 2: Atama gerçekleşmeyen talep noktalarından saat yönü doğrultusunda olan ilk müşteri seçilir.

Adım 3:

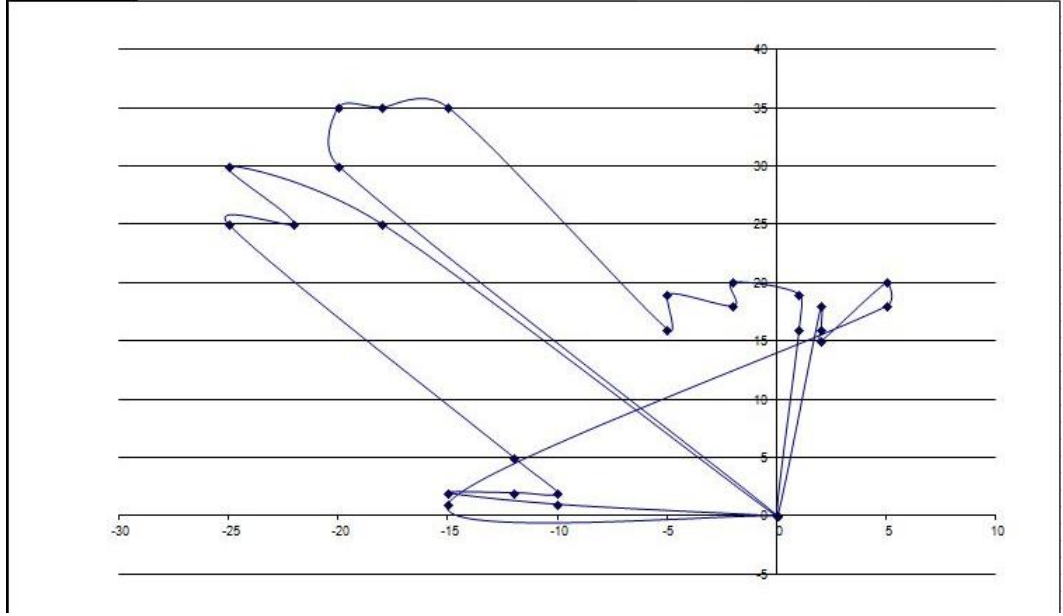
Eğer; seçilen talep merkezi “*i*” için; aracın kalan kapasitesi, bu merkezin (*i* noktasının) talebinden fazla ise *i* noktasına dağıtım gerçekleştirilir ve *Adım 4*'e geçilir.

Eğer; seçilen talep merkezi “*i*” için; aracın kalan kapasitesi, bu merkezin (*i* noktasının) talebinden fazla değilse önce dağıtım yapılacak diğer müşteriler aynı şekilde kontrol edilir ve talebi araç kapasitesinden düşük olan müşterilere atamalar gerçekleştirilir. Bu kontrol sonucunda mevcut müşteri yoksa araç sayısı 1 arttırılarak *Adım 1*'e geri dönülür.

Adım 4: Başlangıç noktası depodan çıkan doğrusal bir çizgi saat yönü doğrultusunda döndürülerek seçilen müşteriden sonra bulunan sıradaki talep merkezi seçilir ve yeni müşteri için *Adım 3*'e dönülür.

Şekil 3.14: Süpürme algoritmasının adımları.

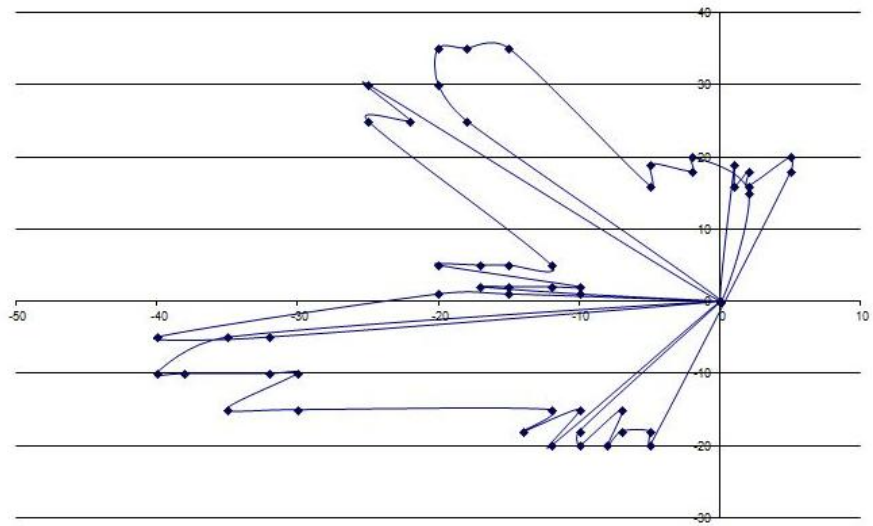
Solomon test verilerindeki talep noktaları ile depo arasındaki ve müşterilerin birbirleri arasındaki uzaklıklar hesaplanırken öklid uzaklık yöntemi kullanılmıştır. Bunun için depoya ve her talep noktasına ait “*X-Y*” koordinatları test verilerinden alınmıştır. Süpürme algoritması, Yumurtacı ve diğ.’nin (2008) Microsoft Office Excel’de makrolar yardımı ile hazırladıkları bir programda test problemlerine uygun değişiklikler yapılmasıyla çalıştırılmıştır. Hazırlanan programda koordinatlar ile talep miktarları modele girdi olarak verilerek çözüm başlatılmakta ve sonuç olarak rota elde edilmektedir. Süpürme algoritmasının test problemlerine uygulanması sonucunda elde edilen rotalar ile rota sıralamaları sırasıyla şekil ve tablolarda gösterilmiştir.



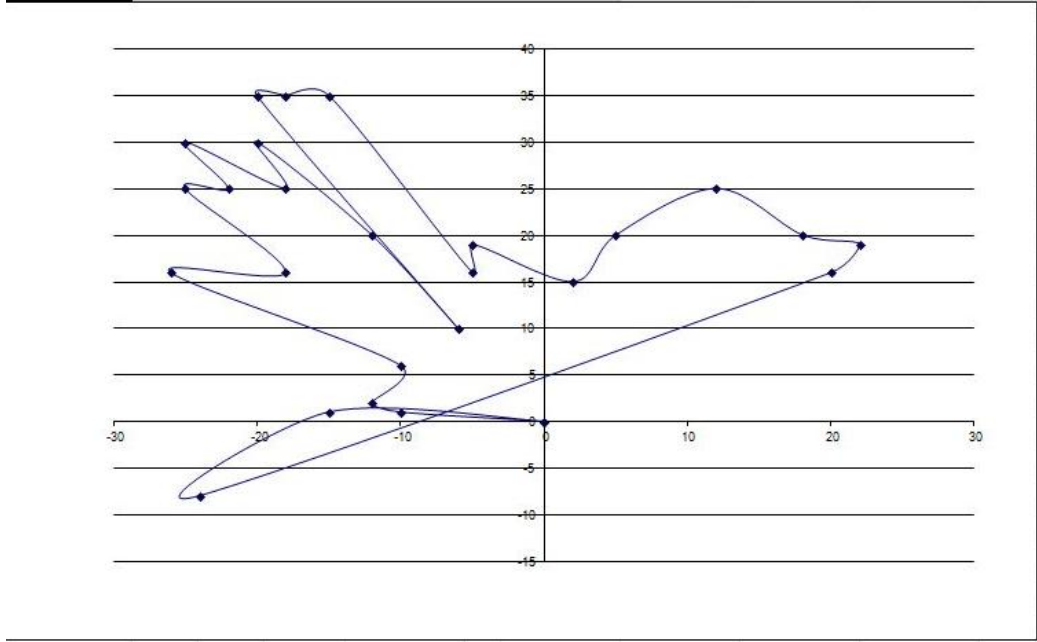
Şekil 3.15: Süpürme algoritması ile 25 müşterilik C101 problem kümesi için başlangıç rotası grafiği (3 rota, 264 br mesafe)

Tablo 3.2: Süpürme algoritması ile 25 müşterilik C101 problem kümesi rota tablosu

Rota1	20	25	22	21	23	18	17	19	13	0
Rota2	15	16	14	12	10	11	8	9	6	7
Rota3	4	3	5	2	1	24	0	0	0	0



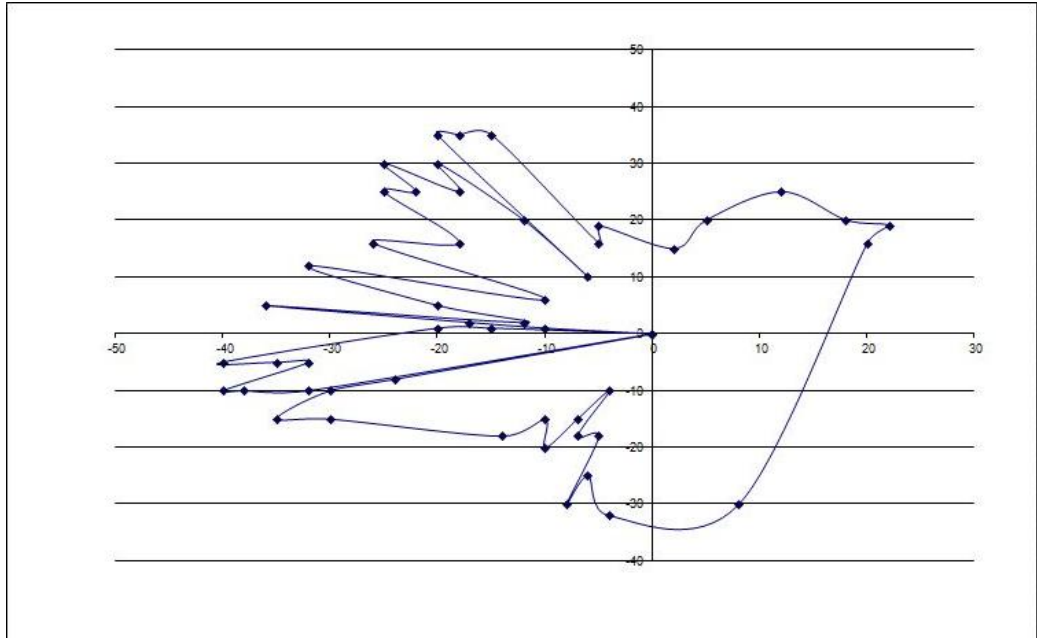
Şekil 3.16: 50 müşterilik C101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (5 rota, 525 br mesafe)



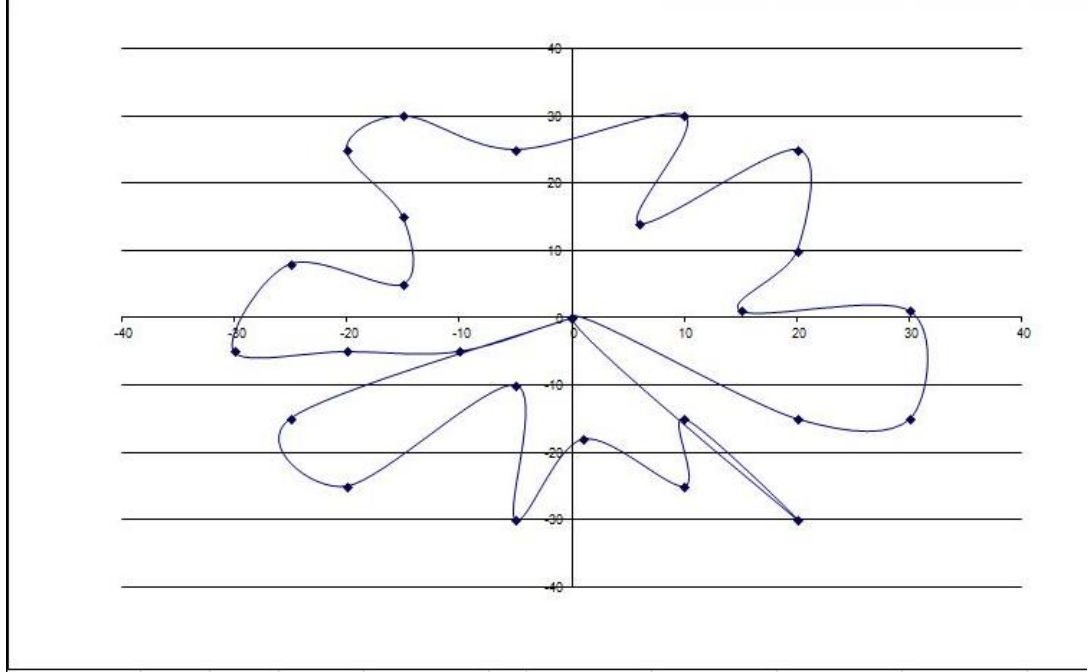
Şekil 3.18: 25 müşterilik C201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 276 br mesafe)

Tablo 3.5: Süpürme algoritması ile 25 müşterilik C201 problem kümesi rota tablosu

Rota1	20	22	21	23	25	18	17	19	13	15	9	8	16	14	12	10	11	5	2	1	7	3	4	6	24
-------	----	----	----	----	----	----	----	----	----	----	---	---	----	----	----	----	----	---	---	---	---	---	---	---	----



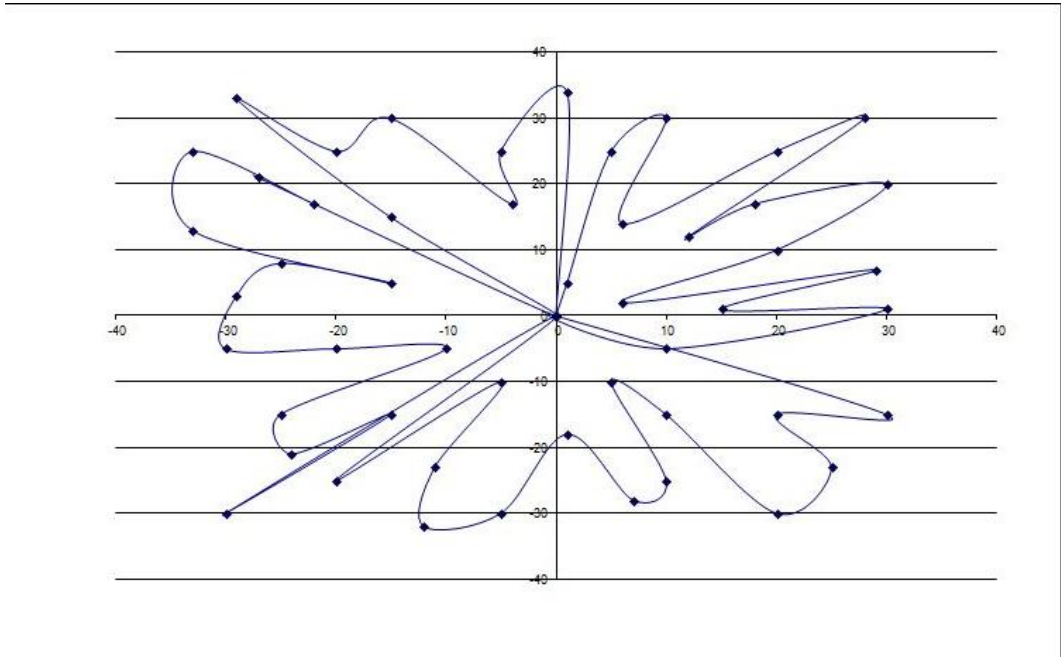
Şekil 3.19: 50 müşterilik C201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 564 br mesafe)



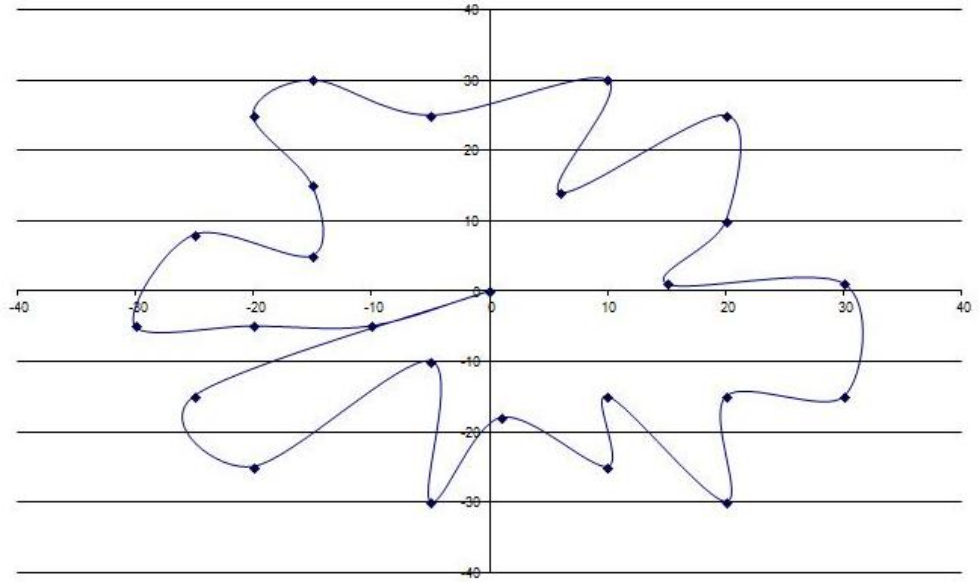
Şekil 3.21: 25 müşterilik R101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 407 br mesafe)

Tablo 3.8: Süpürme algoritması ile 25 müşterilik R101 problem kümesi rota tablosu

Rota 1	6	5	17	8	18	7	19	11	10	20	1	9	3	12	24	25	4
Rota 2	23	21	22	2	15	13	14	16	0	0	0	0	0	0	0	0	0



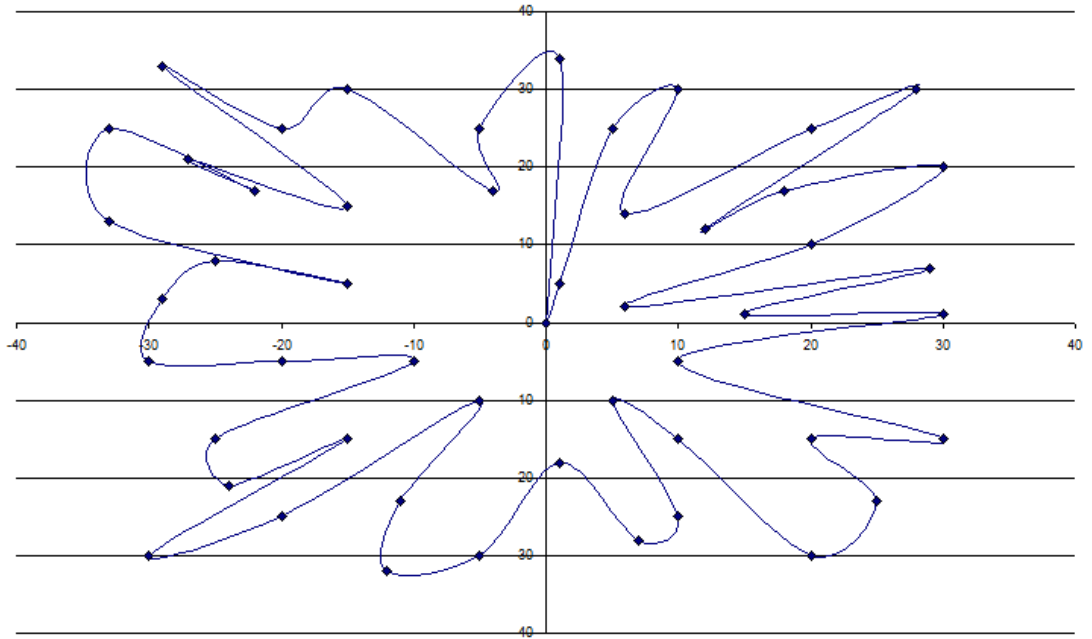
Şekil 3.22: 50 müşterilik R101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (4 rota, 815 br mesafe)



Şekil 3.24: 25 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 361 br mesafe)

Tablo 3.11: Süpürme algoritması ile 25 müşterilik R201 problem kümesi rota tablosu

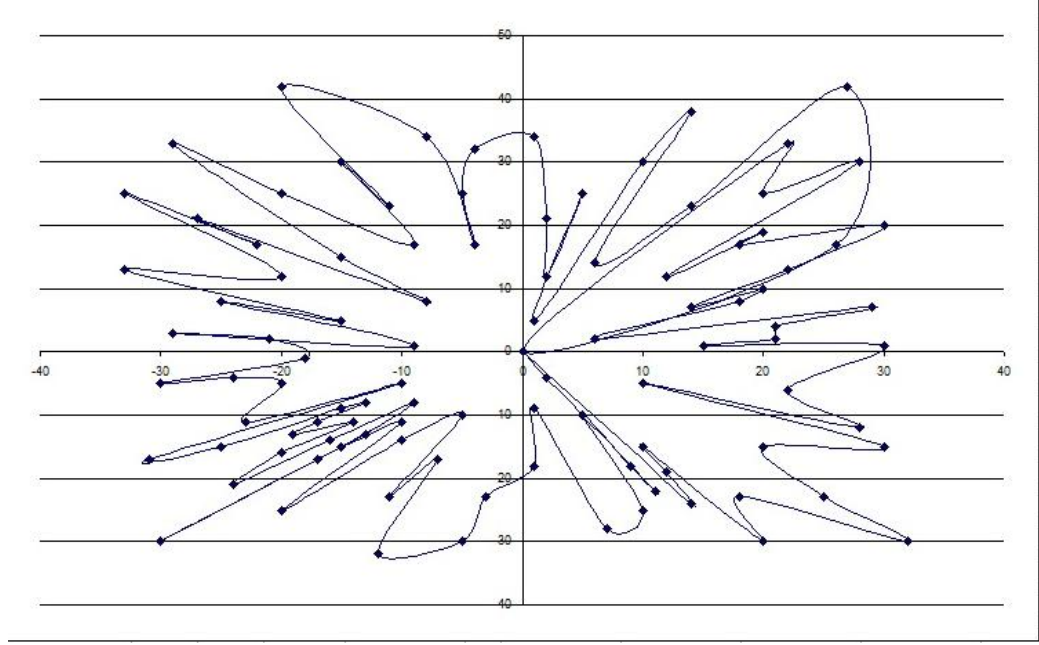
Rota 1	6	5	17	8	18	7	19	11	10	20	1	9	3	12	24	25	4	23	21	22	2	15	13	14	16
--------	---	---	----	---	----	---	----	----	----	----	---	---	---	----	----	----	---	----	----	----	---	----	----	----	----



Şekil 3.25: 50 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 687 br mesafe)

Tablo 3.12: Süpürme algoritması ile 50 müşterilik R201 problem kümesi rota tablosu

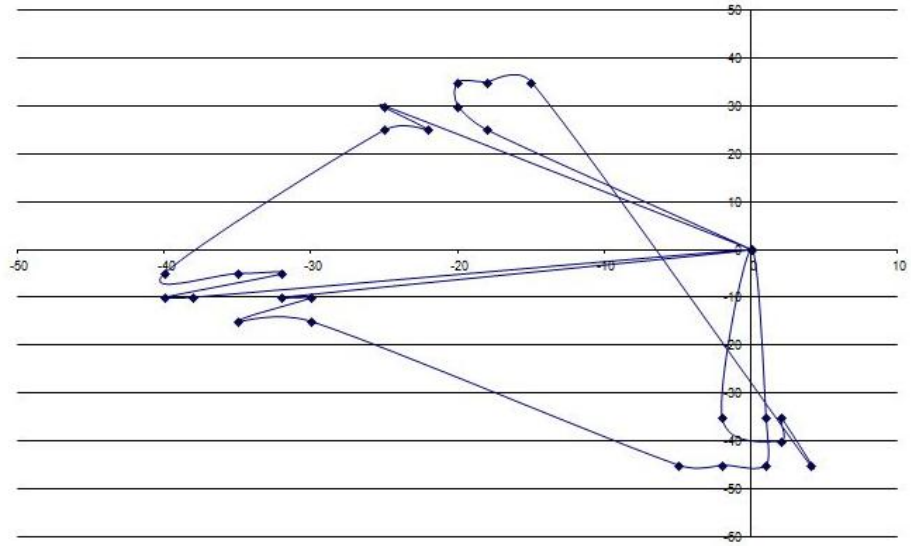
Rota 1	27	30	20	1	9	35	50	33	34	3	28	29	12	24	26	25	4
(rota1 devam satırı)	39	23	21	40	22	41	2	15	43	42	13	14	38	37	44	16	6
(rota1 devam satırı)	5	17	45	8	18	46	36	48	47	7	49	19	11	31	10	32	0



Şekil 3.26: 100 müşterilik R201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (2 rota, 1280 br mesafe)

Tablo 3.13: Süpürme algoritması ile 100 müşterilik R201 problem kümesi rota tablosu

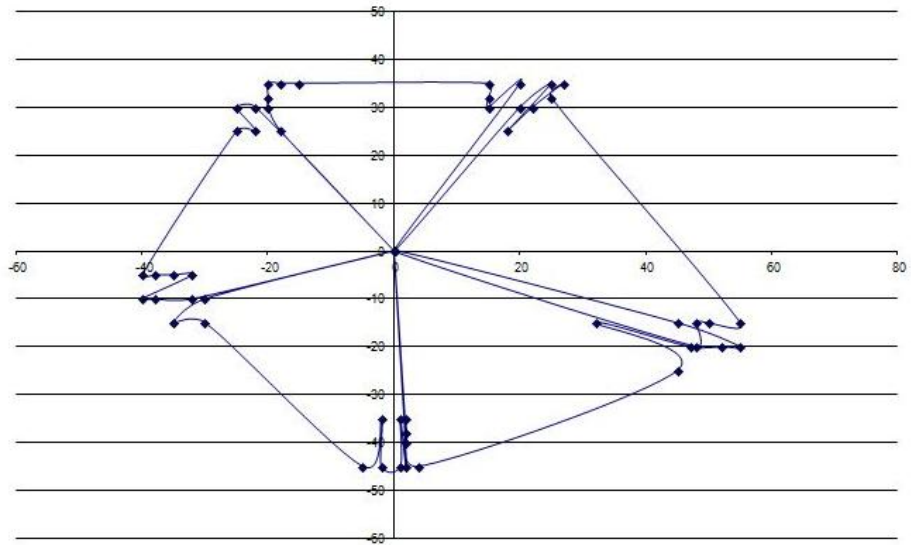
Rota1	53	74	73	40	22	41	58	2	57	15	43	87	42	13	97	14	95
	100	92	38	37	94	98	44	91	59	85	93	96	99	16	86	6	61
	5	84	17	60	83	45	89	8	18	46	82	36	48	47	52	7	49
	19	88	11	62	64	63	31	10	90	32	70	69	30	27	20	66	1
	51	65	78	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rota2	71	9	35	50	81	33	34	79	76	3	77	28	29	68	80	12	24
	54	55	26	25	4	39	67	56	23	21	72	75	0	0	0	0	0



Şekil 3.27: 25 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (3 rota, 463 br mesafe)

Tablo 3.14: Süpürme algoritması ile 25 müşterilik RC101 problem kümesi rota tablosu

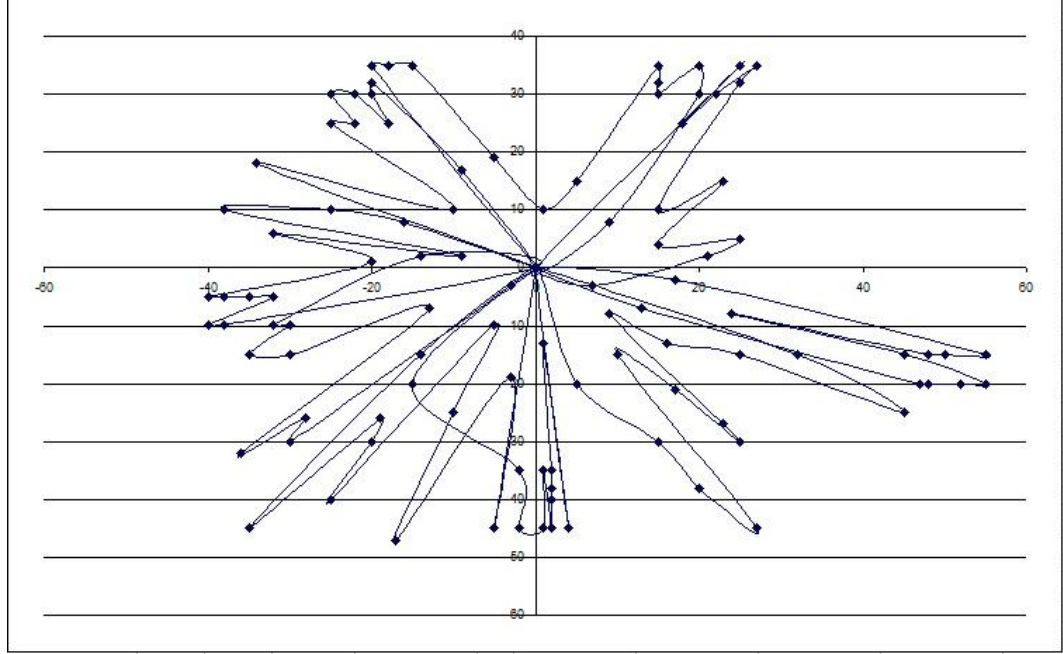
Rota 1	2	4	5	3	1	18	20	19	24
Rota 2	22	21	23	25	9	13	10	11	0
Rota 3	15	16	12	14	17	7	6	8	0



Şekil 3.28: 50 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (5 rota, 833 br mesafe)

Tablo 3.15: Süpürme algoritması ile 50 müşterilik RC101 problem kümesi rota tablosu

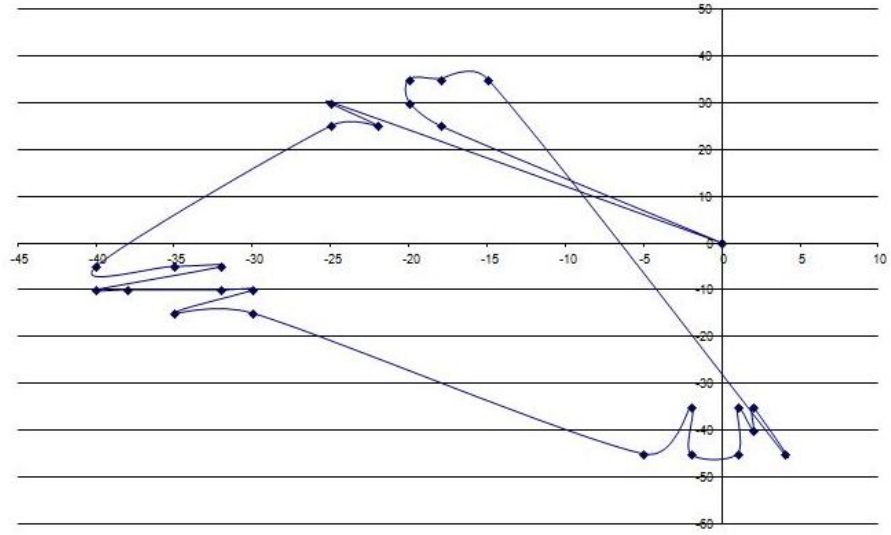
Rota 1	2	4	45	5	3	1	43	44	42	40	0
Rota 2	39	36	41	38	35	37	27	29	31	30	0
Rota 3	34	26	28	32	50	33	18	20	49	19	0
Rota 4	48	22	21	23	24	25	9	13	10	0	0
Rota 5	11	15	16	12	14	47	17	7	6	8	46



Şekil 3.29: 100 müşterilik RC101 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (9 rota, 1681 br mesafe)

Tablo 3.16: Süpürme algoritması ile 100 müşterilik RC101 problem kümesi rota tablosu

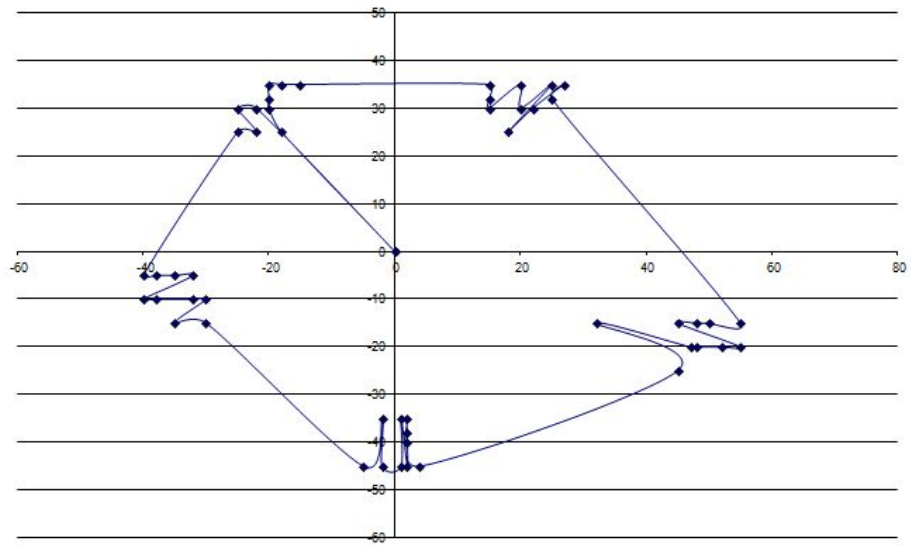
Rota 1	90	59	87	97	82	9	13	10	11	98	0	0	0
Rota 2	15	16	12	14	47	17	53	78	69	73	60	88	0
Rota 3	79	55	7	6	8	46	2	4	45	100	0	0	0
Rota 4	5	3	1	70	68	61	43	44	42	40	39	81	0
Rota 5	36	41	38	35	37	54	72	96	71	93	80	0	0
Rota 6	94	27	29	31	67	34	26	28	30	32	92	0	0
Rota 7	50	33	62	95	91	85	63	84	56	89	76	51	64
Rota 8	18	66	20	49	19	48	22	21	23	24	52	0	0
Rota 9	99	75	86	74	58	65	57	77	83	25	0	0	0



Şekil 3.30: 25 müşterilik RC201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 342 br mesafe)

Tablo 3.17: Süpürme algoritması ile 25 müşterilik RC201 problem kümesi rota tablosu

Rota1	2	4	5	3	1	18	20	19	22	21	23	24	25	9	13	10	11	15	16	12	14	17	7	6	8
-------	---	---	---	---	---	----	----	----	----	----	----	----	----	---	----	----	----	----	----	----	----	----	---	---	---



Şekil 3.31: 50 müşterilik RC201 problem kümesi için süpürme algoritması ile elde edilen başlangıç rotası grafiği (1 rota, 517 br mesafe)

3.6.2. İleri İtmeli Ekleme Sezgisel Yöntemi ile Başlangıç Çözümünün Elde Edilmesi

Süpürme algoritmasında en düşük araç sayısı bulunduğu için daha yüksek rota sayılı fakat daha düşük mesafeli rotalar elde edilip edilmediği incelenememiştir. Bu yüzden daha fazla sayıda pek iyi sonuçlar üretmeyen ileri itmeli ekleme sezgisel yöntemi (İİES) ile de her test problemi için alternatif çözümler hazırlanmıştır. Tan ve diğ.'nin (2001b) başlangıç çözümleri elde etmede yapıcı sezgisellerden olan basit bir yöntem olarak tanımladığı bu yöntemde, müşterilerin depoya yakınlığına bakılarak müşterilerin rotalara atanması gerçekleştirilmiş ve atanacağı rotada en düşük maliyetli bağlantı tercih edilmiştir. Bu yöntem için MATLAB programında kod hazırlanmış ve elde edilen başlangıç çözümleri tablolarda sunulmuştur. Bu yöntem, daha kötü sonuçlar vermesine rağmen ÇAGDKA uygulaması ile iyileştirme yapılmaya çalışıldığı için alternatif çözümleri görmek ve pareto çözüm sürecinin çalıştığını göstermek amacıyla seçilmiştir.

Tablo 3.20: İİES algoritması ile 25 müşterilik C101 problem kümesi rota tablosu (17 rota)

Rota1	20	0	0	0	0	0
Rota2	13	0	0	0	0	0
Rota3	17	0	0	0	0	0
Rota4	5	0	0	0	0	0
Rota5	3	0	0	0	0	0
Rota6	24	18	15	10	0	0
Rota7	7	0	0	0	0	0
Rota8	25	19	11	9	6	4
Rota9	8	0	0	0	0	0
Rota10	16	0	0	0	0	0
Rota11	14	0	0	0	0	0
Rota12	12	0	0	0	0	0
Rota13	23	0	0	0	0	0
Rota14	2	0	0	0	0	0
Rota15	22	0	0	0	0	0
Rota16	21	0	0	0	0	0
Rota17	1	0	0	0	0	0

Tablo 3.21: İİES algoritması ile 50 müşterilik C101 problem kümesi rota tablosu (34 rota)

Rota1	20	0	0	0	Rota13	25	35	44	0	Rota25	12	34	21	0
Rota2	13	0	0	0	Rota14	27	0	0	0	Rota26	26	0	0	0
Rota3	17	0	0	0	Rota15	8	0	0	0	Rota27	6	0	0	0
Rota4	32	0	0	0	Rota16	10	38	39	36	Rota28	4	0	0	0
Rota5	43	33	19	0	Rota17	29	16	9	0	Rota29	23	0	0	0
Rota6	5	0	0	0	Rota18	46	0	0	0	Rota30	2	0	0	0
Rota7	3	0	0	0	Rota19	30	0	0	0	Rota31	50	0	0	0
Rota8	42	0	0	0	Rota20	11	0	0	0	Rota32	22	0	0	0
Rota9	24	18	15	0	Rota21	14	0	0	0	Rota33	1	49	0	0
Rota10	41	40	37	0	Rota22	45	0	0	0	Rota34	47	0	0	0
Rota11	31	0	0	0	Rota23	28	0	0	0					
Rota12	7	0	0	0	Rota24	48	0	0	0					

Tablo 3.22: İİES algoritması ile 100 müşterilik C101 problem kümesi rota tablosu (63 rota)

Rota1	81	0	0	0	Rota22	35	0	0	0	Rota43	100	0	0	0
Rota2	57	0	0	0	Rota23	76	0	0	0	Rota44	88	0	0	0
Rota3	20	0	0	0	Rota24	62	0	0	0	Rota45	48	0	0	0
Rota4	98	0	0	0	Rota25	53	37	0	0	Rota46	12	34	0	0
Rota5	78	0	0	0	Rota26	27	56	77	0	Rota47	64	0	0	0
Rota6	13	0	0	0	Rota27	8	92	93	0	Rota48	26	0	0	0
Rota7	17	0	0	0	Rota28	10	38	39	0	Rota49	6	0	0	0
Rota8	32	0	0	0	Rota29	29	16	0	0	Rota50	99	0	0	0
Rota9	43	55	71	0	Rota30	44	58	79	0	Rota51	89	0	0	0
Rota10	90	96	95	0	Rota31	84	61	59	0	Rota52	51	0	0	0
Rota11	33	0	0	0	Rota32	70	0	0	0	Rota53	4	0	0	0
Rota12	5	42	54	74	Rota33	46	0	0	0	Rota54	68	0	0	0
Rota13	67	87	86	94	Rota34	30	0	0	0	Rota55	23	0	0	0
Rota14	65	0	0	0	Rota35	11	97	80	0	Rota56	2	0	0	0
Rota15	3	0	0	0	Rota36	73	0	0	0	Rota57	91	0	0	0
Rota16	24	18	82	72	Rota37	60	0	0	0	Rota58	50	0	0	0
Rota17	63	40	0	0	Rota38	85	0	0	0	Rota59	66	0	0	0
Rota18	41	83	0	0	Rota39	14	0	0	0	Rota60	22	52	49	47
Rota19	31	0	0	0	Rota40	45	36	0	0	Rota61	69	0	0	0
Rota20	7	0	0	0	Rota41	28	0	0	0	Rota62	21	0	0	0
Rota21	25	19	15	0	Rota42	9	0	0	0	Rota63	1	75	0	0

Tablo 3.25: İİES algoritması ile 100 müşterilik C201 problem kümesi rota tablosu (30 rota)

Rota1	20	93	2	74	72	61	64	66	69	0	0
Rota2	67	5	24	27	30	29	32	38	49	55	54
Rota3	22	63	62	65	94	98	0	0	0	0	0
Rota4	75	0	0	0	0	0	0	0	0	0	0
Rota5	1	99	6	36	59	57	0	0	0	0	0
Rota6	100	0	0	0	0	0	0	0	0	0	0
Rota7	97	0	0	0	0	0	0	0	0	0	0
Rota8	33	0	0	0	0	0	0	0	0	0	0
Rota9	92	0	0	0	0	0	0	0	0	0	0
Rota10	31	95	4	89	86	84	83	82	71	76	0
Rota11	68	35	37	39	34	26	18	23	13	9	0
Rota12	7	0	0	0	0	0	0	0	0	0	0
Rota13	3	0	0	0	0	0	0	0	0	0	0
Rota14	53	0	0	0	0	0	0	0	0	0	0
Rota15	56	0	0	0	0	0	0	0	0	0	0
Rota16	91	58	40	46	45	51	50	25	11	8	0
Rota17	28	60	85	77	96	87	90	0	0	0	0
Rota18	88	0	0	0	0	0	0	0	0	0	0
Rota19	19	0	0	0	0	0	0	0	0	0	0
Rota20	44	16	15	10	0	0	0	0	0	0	0
Rota21	14	0	0	0	0	0	0	0	0	0	0
Rota22	12	0	0	0	0	0	0	0	0	0	0
Rota23	17	0	0	0	0	0	0	0	0	0	0
Rota24	70	80	47	48	21	0	0	0	0	0	0
Rota25	52	0	0	0	0	0	0	0	0	0	0
Rota26	73	0	0	0	0	0	0	0	0	0	0
Rota27	43	79	0	0	0	0	0	0	0	0	0
Rota28	42	81	0	0	0	0	0	0	0	0	0
Rota29	41	0	0	0	0	0	0	0	0	0	0
Rota30	78	0	0	0	0	0	0	0	0	0	0

Tablo 3.26: İİES algoritması ile 25 müşterilik R101 problem kümesi rota tablosu (15 rota)

Rota1	12	0	0	Rota9	9	10	0
Rota2	14	15	0	Rota10	8	24	0
Rota3	2	21	25	Rota11	19	20	1
Rota4	5	0	0	Rota12	6	17	13
Rota5	16	0	0	Rota13	22	0	0
Rota6	18	0	0	Rota14	3	0	0
Rota7	23	0	0	Rota15	7	4	0
Rota8	11	0	0				

Tablo 3.27: İIES algoritması ile 50 müşterilik R101 problem kümesi rota tablosu (25 rota)

Rota1	39	0	0	0	0	Rota14	23	0	0	0	0
Rota2	36	0	0	0	0	Rota15	11	0	0	0	0
Rota3	33	0	0	0	0	Rota16	9	26	0	0	0
Rota4	47	0	0	0	0	Rota17	2	8	10	0	0
Rota5	45	0	0	0	0	Rota18	27	19	49	0	0
Rota6	12	50	0	0	0	Rota19	14	7	46	0	0
Rota7	42	0	0	0	0	Rota20	38	48	0	0	0
Rota8	5	44	16	24	0	Rota21	28	29	34	32	1
Rota9	15	0	0	0	0	Rota22	6	17	13	0	0
Rota10	21	0	0	0	0	Rota23	22	43	0	0	0
Rota11	40	37	0	0	0	Rota24	3	4	25	0	0
Rota12	31	30	35	20	0	Rota25	41	0	0	0	0
Rota13	18	0	0	0	0						

Tablo 3.28: İIES algoritması ile 100 müşterilik R101 problem kümesi rota tablosu (53 rota)

Rota1	65	0	0	0	0	Rota18	31	30	35	0	0	Rota35	28	84	0	0	0
Rota2	39	0	0	0	0	Rota19	62	18	56	0	0	Rota36	94	43	80	0	0
Rota3	36	0	0	0	0	Rota20	76	79	26	0	0	Rota37	29	78	32	89	0
Rota4	72	0	0	0	0	Rota21	23	0	0	0	0	Rota38	22	0	0	0	0
Rota5	33	66	37	0	0	Rota22	11	0	0	0	0	Rota39	51	0	0	0	0
Rota6	67	0	0	0	0	Rota23	75	0	0	0	0	Rota40	3	0	0	0	0
Rota7	47	0	0	0	0	Rota24	69	57	0	0	0	Rota41	7	4	58	0	0
Rota8	45	0	0	0	0	Rota25	71	9	10	48	0	Rota42	53	74	60	0	0
Rota9	64	50	68	0	0	Rota26	2	73	24	0	0	Rota43	41	0	0	0	0
Rota10	92	12	40	0	0	Rota27	98	20	0	0	0	Rota44	97	0	0	0	0
Rota11	42	0	0	0	0	Rota28	88	49	17	0	0	Rota45	90	0	0	0	0
Rota12	5	44	16	0	0	Rota29	99	46	0	0	0	Rota46	96	0	0	0	0
Rota13	15	0	0	0	0	Rota30	14	61	85	6	0	Rota47	13	0	0	0	0
Rota14	21	0	0	0	0	Rota31	95	86	0	0	0	Rota48	54	93	0	0	0
Rota15	63	87	0	0	0	Rota32	38	91	0	0	0	Rota49	100	0	0	0	0
Rota16	83	82	0	0	0	Rota33	59	81	34	55	25	Rota50	1	77	0	0	0
Rota17	52	0	0	0	0	Rota34	27	19	8	0	0	Rota51	70	0	0	0	0

Tablo 3.29: İİES algoritması ile 25 müşterilik R201 problem kümesi rota tablosu (6 rota)

Rota1	2	14	5	7	19	8	18	3	20	10
Rota2	12	21	15	16	6	0	0	0	0	0
Rota3	11	23	24	0	0	0	0	0	0	0
Rota4	22	9	0	0	0	0	0	0	0	0
Rota5	13	17	4	25	0	0	0	0	0	0
Rota6	1	0	0	0	0	0	0	0	0	0

Tablo 3.30: İİES algoritması ile 50 müşterilik R201 problem kümesi rota tablosu (8 rota)

Rota1	33	39	45	36	11	31	10	0	0	0	0	0
Rota2	5	47	14	42	16	44	41	22	40	3	34	26
Rota3	27	28	29	12	21	15	38	7	19	8	18	0
Rota4	2	23	9	50	48	17	0	0	0	0	0	0
Rota5	30	0	0	0	0	0	0	0	0	0	0	0
Rota6	6	49	20	35	4	24	0	0	0	0	0	0
Rota7	46	43	37	13	1	25	0	0	0	0	0	0
Rota8	32	0	0	0	0	0	0	0	0	0	0	0

Tablo 3.31: İİES algoritması ile 100 müşterilik R201 problem kümesi rota tablosu (11 rota)

Rota1	36	63	65	72	39	15	61	16	44	87	94	0	0	0	0	0	0
Rota2	42	92	45	47	83	33	69	82	31	88	0	0	0	0	0	0	0
Rota3	27	52	5	95	98	75	23	73	76	78	51	81	79	9	66	20	50
Rota4	28	29	12	21	19	64	8	18	6	53	0	0	0	0	0	0	0
Rota5	30	62	11	67	40	58	0	0	0	0	0	0	0	0	0	0	0
Rota6	59	14	2	71	10	48	60	89	0	0	0	0	0	0	0	0	0
Rota7	7	99	38	85	57	22	41	3	34	26	0	0	0	0	0	0	0
Rota8	84	86	37	43	56	35	68	0	0	0	0	0	0	0	0	0	0
Rota9	90	49	46	96	97	54	24	55	4	13	0	0	0	0	0	0	0
Rota10	32	74	93	100	91	17	0	0	0	0	0	0	0	0	0	0	0
Rota11	1	70	77	80	25	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 3.32: İİES algoritması ile 25 müşterilik RC101 problem kümesi rota tablosu (8 rota)

Rota1	14	19	22	0	0
Rota2	5	2	0	0	0
Rota3	15	16	6	0	0
Rota4	21	23	18	20	0
Rota5	12	11	9	24	0
Rota6	7	8	4	3	1
Rota7	10	13	17	0	0
Rota8	25	0	0	0	0

Tablo 3.33: İİES algoritması ile 50 müşterilik RC101 problem kümesi rota tablosu (19 rota)

Rota1	33	28	27	0	0	Rota11	21	23	18	20	0
Rota2	31	29	30	0	0	Rota12	12	11	9	0	0
Rota3	39	36	32	0	0	Rota13	40	37	0	0	0
Rota4	19	34	0	0	0	Rota14	7	8	46	4	3
Rota5	47	14	0	0	0	Rota15	50	0	0	0	0
Rota6	49	22	0	0	0	Rota16	10	13	17	0	0
Rota7	45	5	2	0	0	Rota17	25	48	0	0	0
Rota8	44	38	35	41	0	Rota18	43	0	0	0	0
Rota9	15	16	6	0	0	Rota19	1	0	0	0	0
Rota10	42	26	24	0	0						

Tablo 3.34: İİES algoritması ile 100 müşterilik RC101 problem kümesi rota tablosu (39 rota)

Rota1	33	28	27	0	0	0	Rota21	86	0	0	0	0	0
Rota2	59	76	0	0	0	0	Rota22	7	8	46	4	0	0
Rota3	31	29	30	0	0	0	Rota23	65	85	50	0	0	0
Rota4	14	98	0	0	0	0	Rota24	79	74	77	0	0	0
Rota5	36	39	90	0	0	0	Rota25	92	95	87	10	13	17
Rota6	64	78	0	0	0	0	Rota26	83	61	0	0	0	0
Rota7	19	34	0	0	0	0	Rota27	82	73	25	48	0	0
Rota8	47	22	49	0	0	0	Rota28	69	53	3	80	0	0
Rota9	45	2	32	0	0	0	Rota29	62	71	43	68	100	0
Rota10	5	0	0	0	0	0	Rota30	84	94	54	0	0	0
Rota11	44	38	35	41	0	0	Rota31	67	57	58	0	0	0
Rota12	15	16	6	0	0	0	Rota32	99	0	0	0	0	0
Rota13	52	26	0	0	0	0	Rota33	56	96	70	0	0	0
Rota14	42	88	0	0	0	0	Rota34	97	0	0	0	0	0
Rota15	21	23	18	0	0	0	Rota35	81	55	0	0	0	0
Rota16	75	9	0	0	0	0	Rota36	89	0	0	0	0	0
Rota17	11	12	37	0	0	0	Rota37	66	0	0	0	0	0
Rota18	72	40	0	0	0	0	Rota38	91	93	0	0	0	0
Rota19	63	0	0	0	0	0	Rota39	1	0	0	0	0	0
Rota20	51	20	24	60	0	0							

Tablo 3.35: İİES algoritması ile 25 müşterilik RC201 problem kümesi rota tablosu (7 rota)

Rota1	5	14	11	12	21	23	19	22
Rota2	15	2	3	6	0	0	0	0
Rota3	7	16	10	20	24	0	0	0
Rota4	9	18	0	0	0	0	0	0
Rota5	8	0	0	0	0	0	0	0
Rota6	13	17	1	4	0	0	0	0
Rota7	25	0	0	0	0	0	0	0

Tablo 3.36: İİES algoritması ile 50 müşterilik RC201 problem kümesi rota tablosu (7 rota)

Rota1	42	36	27	28	31	33	21	23	19	16	7	8	46	6	1
Rota2	45	5	47	14	11	12	22	49	50	26	34	35	37	43	0
Rota3	39	29	30	18	9	10	0	0	0	0	0	0	0	0	0
Rota4	15	2	44	40	38	41	32	24	25	48	0	0	0	0	0
Rota5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rota6	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rota7	13	17	4	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 3.37: İİES algoritması ile 100 müşterilik RC201 problem kümesi rota tablosu (12 rota)

Rota1	36	39	65	59	15	11	12	88	98	90	0	0	0	0
Rota2	45	42	72	27	28	33	76	19	64	18	22	87	86	53
Rota3	5	47	14	52	75	21	23	51	71	94	81	54	0	0
Rota4	95	63	29	30	31	38	44	6	7	9	78	100	0	0
Rota5	92	69	82	2	40	61	0	0	0	0	0	0	0	0
Rota6	62	83	99	16	73	79	8	46	3	43	37	35	55	96
Rota7	85	67	41	1	4	60	13	17	0	0	0	0	0	0
Rota8	84	34	57	49	20	10	74	70	0	0	0	0	0	0
Rota9	56	50	32	26	77	24	48	89	0	0	0	0	0	0
Rota10	97	66	68	0	0	0	0	0	0	0	0	0	0	0
Rota11	80	91	25	0	0	0	0	0	0	0	0	0	0	0
Rota12	93	58	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 3.2'den Tablo 3.20'ye kadar süpürme algoritması ve Tablo 3.21'den Tablo 3.37'ye kadar İİES algoritması ile bulunan başlangıç çözümleri incelendiğinde şu durumlar gözlenmiştir:

- i) Süpürme algoritmasının sonucunda tüm veri setleri için elde edilen sonuçlarda araç sayıları en düşük araç sayısına eşittir.
- ii) İİES algoritmasının sonucunda tüm veri setleri için elde edilen sonuçlarda ise araç sayıları çok yüksek değerlerde bulunmuştur.

3.7. KOMŞULUK YAPILARI

Genel olarak silkeleme aşamasında çeşitlilik sağlamak amaçlandığı için iyileştirme göz önüne alınmamıştır, Pareto çözüm değerlerinin karşılaştırılması ve kümesinin güncelleştirilmesi sadece yerel aramada komşuluk yapısı uygulanması esnasında gerçekleştirilmektedir. Ayrıca kapasite kısıtının aşılp aşılmadığı sadece ister silkeleme isterse yerel aramada uygulansın, geçerli komşuluk yapısı rotalar arası ise kontrol edilmektedir. Rotalar arası komşuluk yapılarında, farklı iki rotadaki farklı müşterilerin talepleri birbirinden farklı olduğundan, değişim hareketinden sonraki rotaların yeni hallerine ait talepler de farklı olacaktır. Rota içi uygulanan komşuluk yapılarında araç kapasitesinin aşılp aşılmadığına bakılmaz. Çünkü rotadaki elemanlar başka bir rotaya geçmediği için geçerli rotanın toplam talep değeri aynı kalmaktadır. Uygulamada kullanılan komşuluk yapıları alt başlıklar halinde aşağıda ele alınmıştır.

3.7.1. Rota İçi Değişim 1-1

Bu komşuluk yapısına silkeleme ve yerel aramada yer verilmiştir. Silkelemede ilk olarak geçerli çözümün rota sayısı belirlenmekte, bu rota içinden rassal bir rota ve rassal olarak belirlenen bu rotadan da yine rassal olarak bir müşteri seçilmektedir. Seçilen müşteri *müşteri1* olarak adlandırılmaktadır. Daha sonra aynı rota içinde, *müşteri2* olarak adlandırılan rassal başka bir müşteri seçilir. *müşteri1*'in rotada bulunduğu konuma *müşteri2*; *müşteri2*'nin rotada bulunduğu konuma ise *müşteri1* atanarak işlem tamamlanır ve yerel arama aşamasında kullanılacak yeni çözüm belirlenmiş olur. Yerel arama aşamasında ise bir önceki çözümde incelenen rota üzerinde işlem yapılmaktadır. Yine rassal bir müşteri seçilir, fakat bu müşterinin aynı rotadaki kendisi dışında kalan tüm diğer müşteriler ile birebir yer değişimleri incelenerek her değişimde amaç fonksiyonlarında iyileşme olup olmadığı kontrol edilir. Pareto çözümlerin karşılaştırılması için Şekil 4.3'te gösterilen "Pareto karşılaştır" fonksiyonunun kodu çalıştırılmaktadır. Örnek bir rota üzerinde rota içi değişim 1-1 Şekil 3.33'te gösterilmiştir.

Rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
--	----	----	----	----	----	----	----	----	----	----	----

Rotanın komşuluk yapısı uygulandıktan sonraki hali	36	44	65	72	39	15	61	16	63	87	94
--	----	----	----	----	----	----	----	----	----	----	----

Şekil 3.33: Rota içi değişim 1-1 komşuluk yapısının örnek uygulaması.

3.7.2. Rota İçi Değişim 2-1

Bu komşuluk yapısına da silkeleme ve yerel aramada yer verilmiştir. Rota içi değişim ile aynı esasta olan bu yapıda, *müşteri1* olarak seçilen sadece tek bir müşteri değil, ardışık iki müşteridir. Şekil 3.34’te örnek hareket gösterilmiştir.

Rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
Rotanın komşuluk yapısı uygulandıktan sonraki hali	36	44	16	39	15	61	65	72	44	87	94

Şekil 3.34: Rota içi değişim 2-1 komşuluk yapısının örnek uygulaması.

3.7.3. Rotalar Arası Değişim 1-1

Bu komşuluk yapısına da silkeleme ve yerel aramada yer verilmiştir. Rotalar arası 1-1 değişiminde, *müşteri1* ve *müşteri2* farklı rotalardan seçilmektedir. Rota içi değişim 1-1’deki gibi sadece yerel aramada uygulandığında amaç fonksiyonu incelenmektedir. Yerel aramada ayrıca geçerli rotanın dışında kalan tüm rotaların tüm elemanları ile sırasıyla değiştirme işlemi gerçekleştirilmektedir. Araç kapasitesine ait kısıtın aşılp aşılmadığı silkelemede de, yerel aramada da kontrol edilir. Bu hareketin bir kere uygulanmasına ait önceki ve sonraki durumları Şekil 3.35’te gösterilmiştir.

1. rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
2. rotanın komşuluk yapısı uygulanmadan önceki hali	11	16	9	3	7	42	36	23	0	0	0
1. rotanın komşuluk yapısı uygulandıktan sonraki hali	36	3	65	72	39	15	61	16	44	87	94
2. rotanın komşuluk yapısı uygulandıktan sonraki hali	11	16	9	63	7	42	36	23	0	0	0

Şekil 3.35: Rotalar arası değişim 1-1 komşuluk yapısının örnek uygulaması.

3.7.4. Rotalar Arası Değişim 2-1

Sadece silkeleme aşamasında yer alan bu komşuluk yapısı, rotalar arası 1-1 değişim ile aynı esastadır. Fakat ilk rotadan seçilen *müşteri1* ile ardışık iki müşteri, ikinci rotadan seçilen *müşteri2* ile ise bir müşteri belirlenmektedir. Şekil 3.36'da bu harekete ait örnek gösterilmiştir.

1. rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
2. rotanın komşuluk yapısı uygulanmadan önceki hali	11	16	9	3	7	42	36	23	0	0	0
1. rotanın komşuluk yapısı uygulandıktan sonraki hali	36	3	72	39	15	61	16	44	87	94	0
2. rotanın komşuluk yapısı uygulandıktan sonraki hali	11	16	9	63	65	7	42	36	23	0	0

Şekil 3.36: Rotalar arası değişim 2-1 komşuluk yapısının örnek uygulaması.

3.7.5. Rota İçi Kaydırma 2-0

Bu komşuluk yapısı silkeleme ve yerel arama aşamalarında yer almaktadır. Yerel aramada geçerli rota içinde rassal olarak seçilen ardışık iki müşterinin (*müşteri*) aynı rotadaki tüm alternatif yerlere kaydırıldığında mevcut amaç fonksiyonlarında iyileşme olup olmadığı kontrol edilmektedir. Silkelemede ise iyileşmeye bakılmadan rassal olarak seçilen ardışık iki müşteri (*müşteri*), aynı rotada rassal olarak seçilen başka bir konuma kaydırılmaktadır. Şekil 3.37'de bu harekete ait örnek gösterilmiştir.

Rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
Rotanın komşuluk yapısı uygulandıktan sonraki hali	65	72	39	15	61	16	44	36	63	87	94

Şekil 3.37: Rota içi kaydırma 2-0 komşuluk yapısının örnek uygulaması.

3.7.6. Rotalar Arası Kaydırma 1-0

Sadece yerel aramada uygulanan bu komşuluk yapısında, geçerli rotadan seçilen rassal bir müşterinin sırayla tüm diğer rotalardaki tüm konumlara kaydırılması durumu incelenmektedir. Her incelemede ilk rotadaki müşteri sayısı bir azalırken diğer rotada bir artmaktadır. Şekil 3.38’de bu harekete ait örnek gösterilmiştir.

1. rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
2. rotanın komşuluk yapısı uygulanmadan önceki hali	11	16	9	3	7	42	36	23	0	0	0
1. rotanın komşuluk yapısı uygulandıktan sonraki hali	36	65	72	39	15	61	16	44	87	94	0
2. rotanın komşuluk yapısı uygulandıktan sonraki hali	11	16	9	3	7	42	36	63	23	0	0

Şekil 3.38: Rotalar arası kaydırma 1-0 komşuluk yapısının örnek uygulaması.

3.7.7. Rotalar Arası Kaydırma 2-0

Sadece yerel aramada uygulanan bu komşuluk yapısı, rotalar arası kaydırma 1-0 ile aynı esasta olup geçerli rotadan seçilen rassal bir müşteri yerine iki ardışık müşteri seçilmektedir. Ardından bu iki müşterinin sırayla tüm diğer rotalardaki tüm konumlara kaydırılması durumu incelenmektedir. Her incelemede ilk rotadaki müşteri sayısı iki artarken diğer rota da iki artmaktadır. Şekil 3.39’da bu harekete ait örnek gösterilmiştir

1. rotanın komşuluk yapısı uygulanmadan önceki hali	36	63	65	72	39	15	61	16	44	87	94
2. rotanın komşuluk yapısı uygulanmadan önceki hali	11	16	9	3	7	42	36	23	0	0	0
1. rotanın komşuluk yapısı uygulandıktan sonraki hali	36	63	65	72	61	16	44	87	94	0	0
2. rotanın komşuluk yapısı uygulandıktan sonraki hali	11	16	39	15	9	3	7	42	36	23	0

Şekil 3.39: Rotalar arası kaydırma 2-0 komşuluk yapısının örnek uygulaması.

3.7.8. Komşuluk Yapılarının Uygulama Sırasının Belirlenmesi

Rota sayısını azalttıktan sonra rotaların içinde iyileştirme yapmaya çalışmak daha uygun bir durumdur. Bu yüzden yerel arama aşamasında ilk olarak rotalar arası kaydırma ve değişim yapıları uygulanmıştır. Denemeler sırasında kaydırma yapılarından önce “rotalar arası değişim 1-1” yer aldığında sonuçlardaki iyileşme miktarının biraz daha fazla olması sebebi ile ilk sıraya alınmıştır. Yerel arama için belirlenen sıralama şu şekildedir: rotalar arası değişim 1-1, rotalar arası kaydırma 2-0, rotalar arası kaydırma 1-0, rota içi değişim 1-1, rota içi kaydırma 2-0 ve rota içi kaydırma 2-1. Silkeleme aşamasında istenen amaç çözümü çeşitlendirmek, yerel en iyilere takılmaktan kurtulmak olduğu için ve rassal çözümler üretilerek iyileşmeler kontrol edilmediğinden komşuluk yapılarının sıralamaları çok fazla bir önem göstermemektedir. O yüzden rota içi değişim 1-1, rota içi kaydırma 2-0, rota içi değişim 2-1, rotalar arası değişim 1-1 ve rotalar arası değişim 2-1 şeklinde sıralanmışlardır.

4. BULGULAR

4.1. PROGRAMIN ÇALIŞTIRILMASI

Bir önceki bölümde (geliştirilen yöntem) bahsedildiği gibi program iki şekilde ele alınmış ve farklı iki başlangıç çözüm yöntemi kullanılmıştır. Bu bölümde çok amaçlı genel değişken komşuluk arama (ÇAGDKA), süpürme algoritması başlangıç çözümü ile hem yüksek dış döngü hem de yüksek iç döngü alternatifleri kullanılarak çalıştırılmıştır. Yüksek iç döngü değeri kullanıldığında daha iyi sonuç verdiği gözlenmiştir. İİES başlangıç yöntemi olarak alındığında tüm problemler üzerinde çalıştırma yapmak yerine seçilen problem kümesi üzerindeki yüksek iç ve dış döngü değerleri ile elde edilen sonuçlar karşılaştırılmıştır. Yüksek iç döngü ile elde edilen sonuçlar İİES başlangıç çözümü seçildiğinde de daha iyi olduğu için, yüksek iç döngü değeri tercih edilmiştir.

Program çalıştırdıktan sonra elde edilen sonuçlar, Chand ve diğ.'nin (2010) çok amaçlı genetik algoritma (ÇAGA) yaklaşımı ile buldukları sonuçlar Bölüm 5 altında karşılaştırılmıştır.

4.1.1. Başlangıç Çözümü Oluşturmak İçin Süpürme Algoritmasının Kullanıldığı Durum

Süpürme algoritmasında bulunan başlangıç çözüm değerleri zaten en düşük araç sayısına eşit olduğu için ÇAGDKA yaklaşımının uygulanması sonucunda yine süpürme algoritmasında bulunan araç sayısına eşit araçlı rota sıralamaları elde edilmiştir. ÇAGDKA yaklaşımında Pareto optimallik fikrinin kullanılmasının sebebi baskın olmayan çözümler kümesini elde ederek alternatif sonuçlara ulaşmaktır. Bu sonuçlara ulaştıktan sonra alternatifler karar vericiye sunulmaktadır. Süpürme algoritmasının tercih edildiği durumda tüm sonuçlar için aynı araç sayısı söz konusu olduğundan farklı

alternatif sunumları gerçekleşmemiştir. Fakat başlangıç çözümünün bu yöntemle oluşturulması ile hemen hemen en iyi sonuç değerlerine ulaşılmıştır. Diğer başlangıç metodunun kullanıldığı durumda süpürme yönteminin kullanıldığı duruma kıyasla araç sayısının daha yüksek fakat katedilen mesafenin daha düşük olduğu tercih edilebilir alternatiflerin de bulunduğu gözlenmiştir. Bu duruma Bölüm 4.1.2’de değinilmiştir.

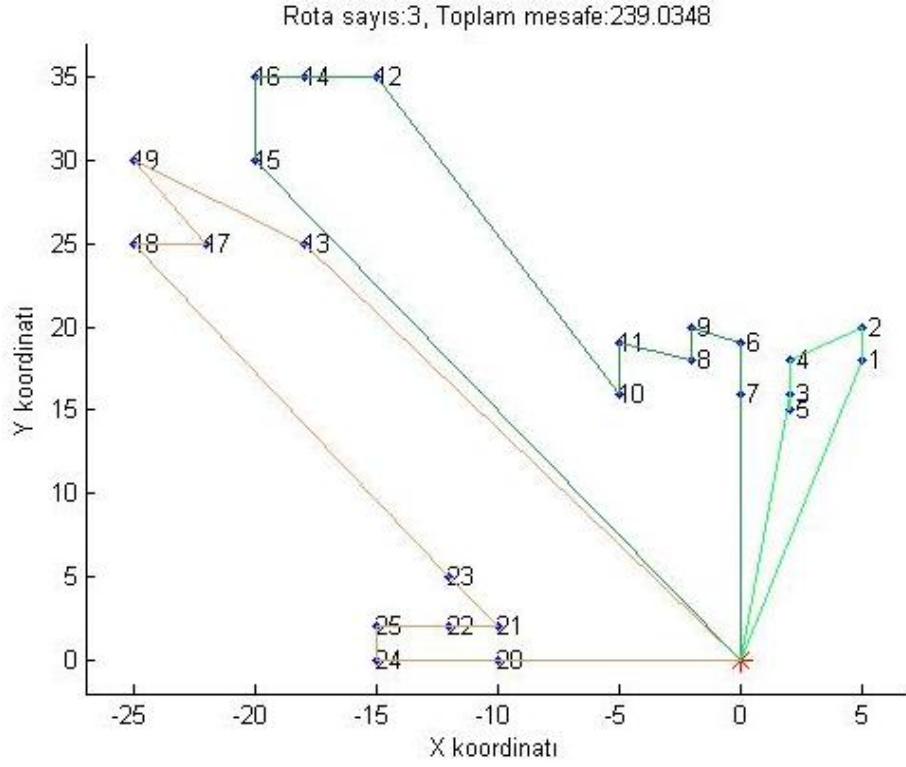
Başlangıç çözümlerinin anlatıldığı Bölüm 3.6.1’de süpürme algoritmasının uygulanması ile elde edilen rotalardaki müşteri sıralamalarına ve görsel olarak şekillerine yer verilmişti. Bu kısımda alt başlıklar halinde, Solomon’un 1983 yılında sunduğu test veri kümelerinin 25, 50 ve 100 müşterilik gruplarına ÇAGDKA yaklaşımının uygulanmasıyla elde edilen sonuçlar ele alınmıştır. ÇAGDKA yaklaşımının dış döngü değeri 200, iç döngü değeri 1 iken uygulanmasıyla elde edilen on sonuç ve dış döngü değeri 50, iç döngü değeri 20 iken elde edilen on sonuç sırasıyla tablolarda sunulmuştur. Ayrıca bu bölümde elde edilen en iyi değerli bazı rotaların şekilleri de tez içerisinde akışı bozmamasına dikkat edilerek örnek olarak sunulmuştur.

4.1.1.1. Geliştirilen Yaklaşımın C101 Problem Kümesi İçin Uygulanması

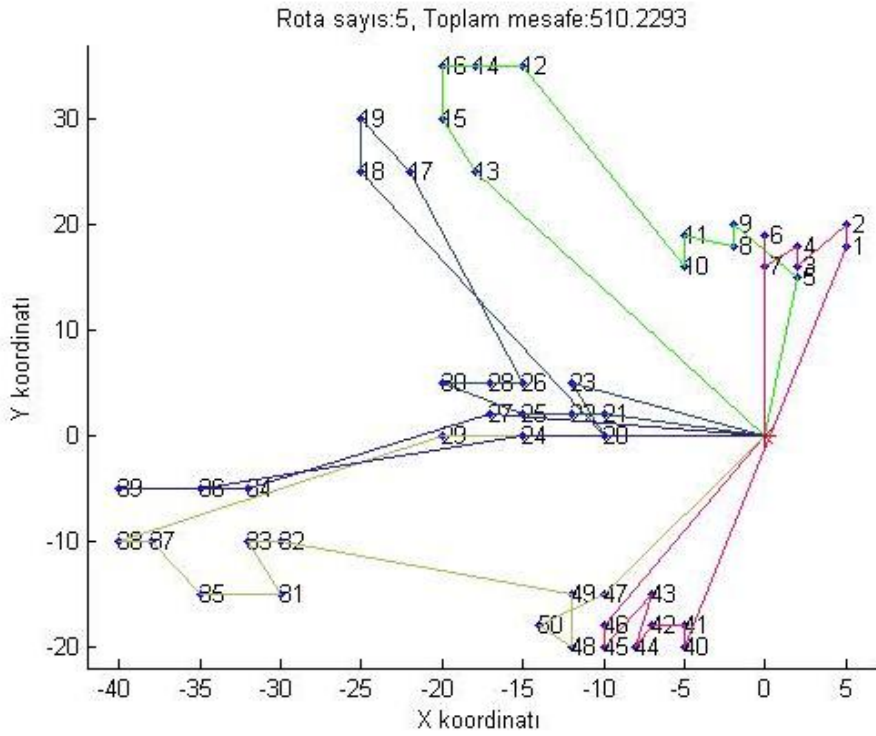
C101 problem kümesine ait verilerin 25, 50 ve 100’er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.1’de gösterilmiştir.

Tablo 4.1: 25, 50 ve 100 müşterilik C101 problem grupları için sonuç tablosu (yüksek dış döngü değerli)

Müşteri sayısı	Mesafe	Araç/rota sayısı
25	262,33	3
	263,85	3
	239,03	3
	263,32	3
	259,00	3
	254,96	3
	263,85	3
	263,85	3
	252,75	3
	263,85	3
50	525,51	5
	525,51	5
	523,95	5
	514,27	5
	525,51	5
	510,22	5
	525,51	5
	525,51	5
	525,51	5
	525,51	5
100	1192,8	10
	1147,2	10
	1173,8	10
	1146,9	10
	1184,2	10
	1156,8	10
	1184,9	10
	1180,8	10
	1170,8	10
	1190,6	10



Şekil 4.1: 25 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)



Şekil 4.2: 50 müşterilik C101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)

Tablo 4.2’de C101 problem kümesi için programda yüksek iç döngü değeri tercih edildiğinde bulunan sonuçlar ve bu sonuçlardan en iyi değerli ikisi Şekil 4.3 ve Şekil 4.4’te gösterilmiştir.

Tablo 4.2: C101 problem kümesi için sonuç tablosu (yüksek iç döngü değeri)

Problem kümesi	Mesafe	Araç/rota sayısı
25	251,26	3
	239,76	3
	257,73	3
	237,04	3
	238,07	3
	237,72	3
	250,16	3
	251,80	3
	251,32	3
	260,16	3
50	499,60	5
	525,51	5
	523,70	5
	525,51	5
	503,67	5
	505,36	5
	513,10	5
	525,51	5
	510,88	5
	503,69	5
100	1149,6	10
	1178,3	10
	1172,1	10
	1171,6	10
	1160,7	10
	1174,1	10
	1192,8	10
	1182,0	10
	1160,2	10
	1190,6	10

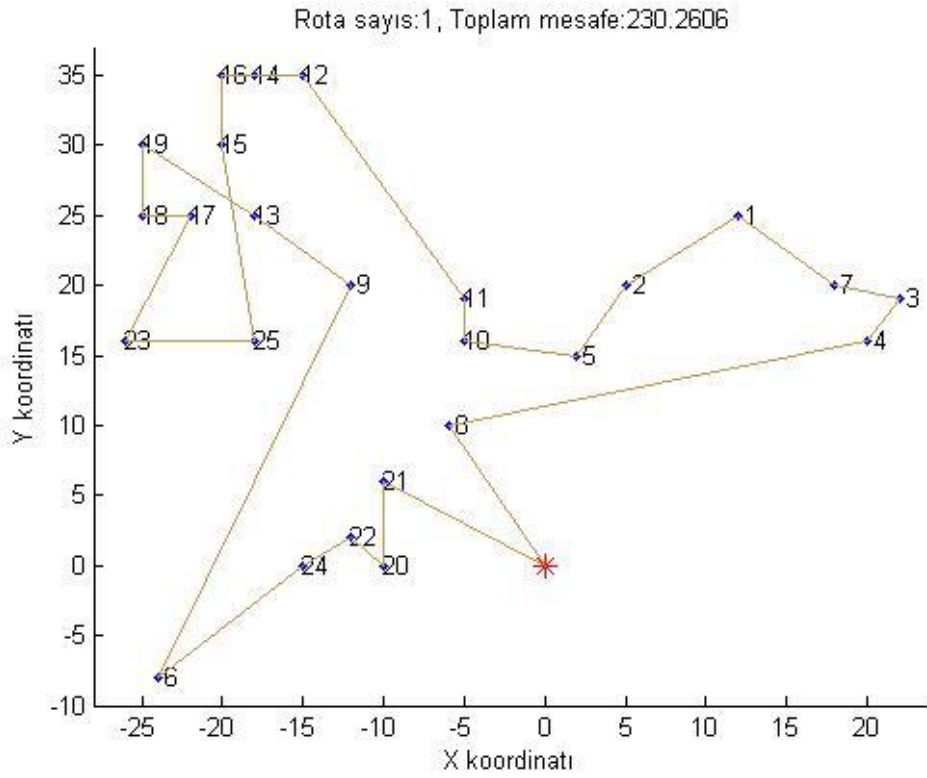
Tablo 4.1 ile Tablo 4.2’de sunulan deęerler incelendięinde sırasıyla 25,50 ve 100 müşteri için; yüksek dış döngü deęerinde elde edilen en iyi deęerler 239,510,1147, yüksek iç döngü deęerinde ise 237, 499, 1149 olarak elde edilmiştir. Daha önce de belirtildięi gibi deęerler çok yakındır.

4.1.1.2. Geliştirilen Yaklaşımın C201 Problem Kümesi İçin Uygulanması

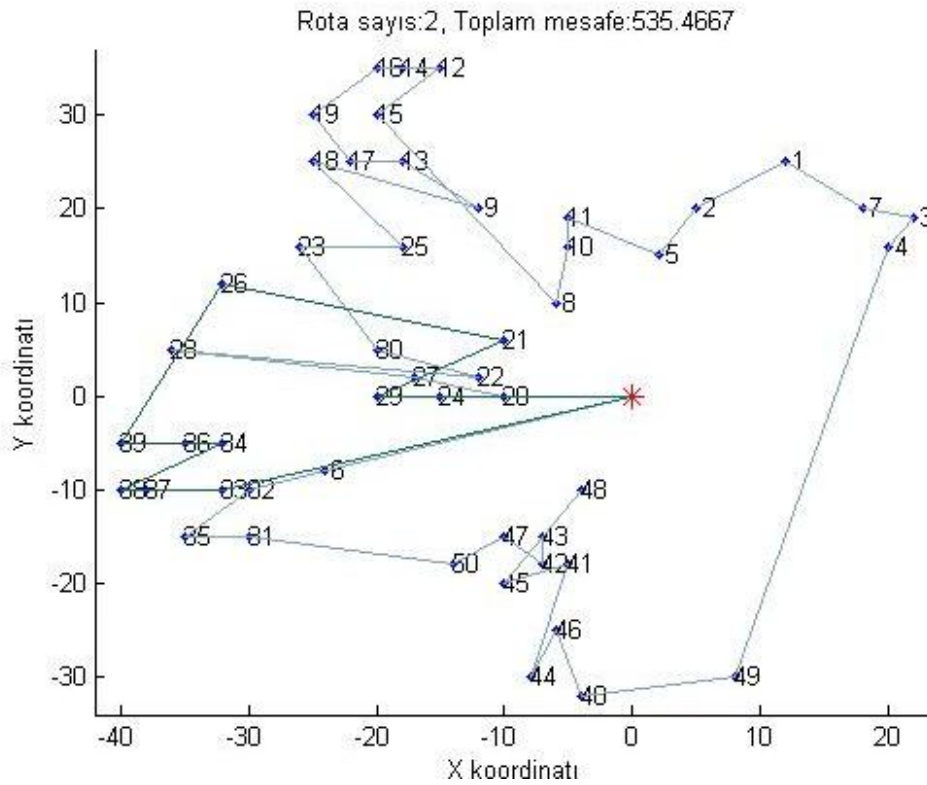
C201 problem kümesine ait verilerin 25, 50 ve 100’er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.3 ve Tablo 4.4’te gösterilmiştir. Tablo 4.3’ten iki görsele Şekil 4.5 ve Şekil 4.6’da, Tablo 4.4’ten iki görsele Şekil 4.7 ve Şekil 4.8’de yer verilmiştir.

Tablo 4.3: C201 problem kümesi için sonuç tablosu (yüksek dış döngü deęerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
C201	25	266,29	1
		276,17	1
		276,17	1
		276,17	1
		276,17	1
		276,17	1
		230,26	1
		276,17	1
		276,17	1
		276,17	1
	50	564,15	2
		543,15	2
		535,46	2
		564,15	2
		544,32	2
		556,36	2
		550,98	2
		554,48	2
		562,10	2
		564,15	2
	100	1034,40	1
		1034,40	1
		1034,40	1
		1014,20	1
		1021,30	1
		1034,40	1
		1034,40	1
		1034,40	1
		1002,60	1
		1034,40	1



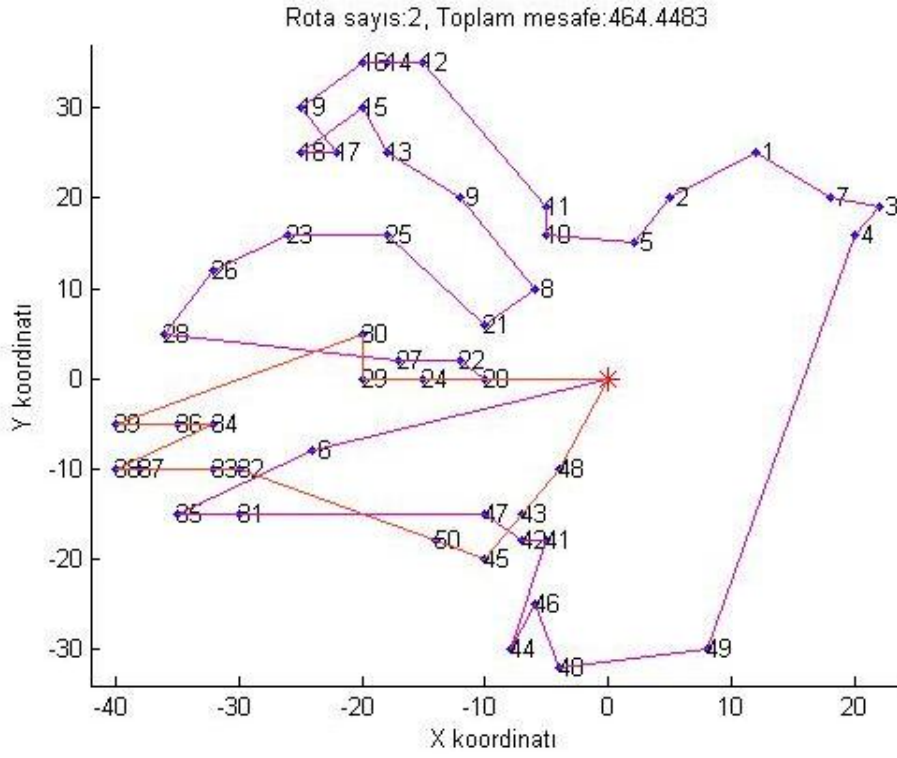
Şekil 4.5: 25 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)



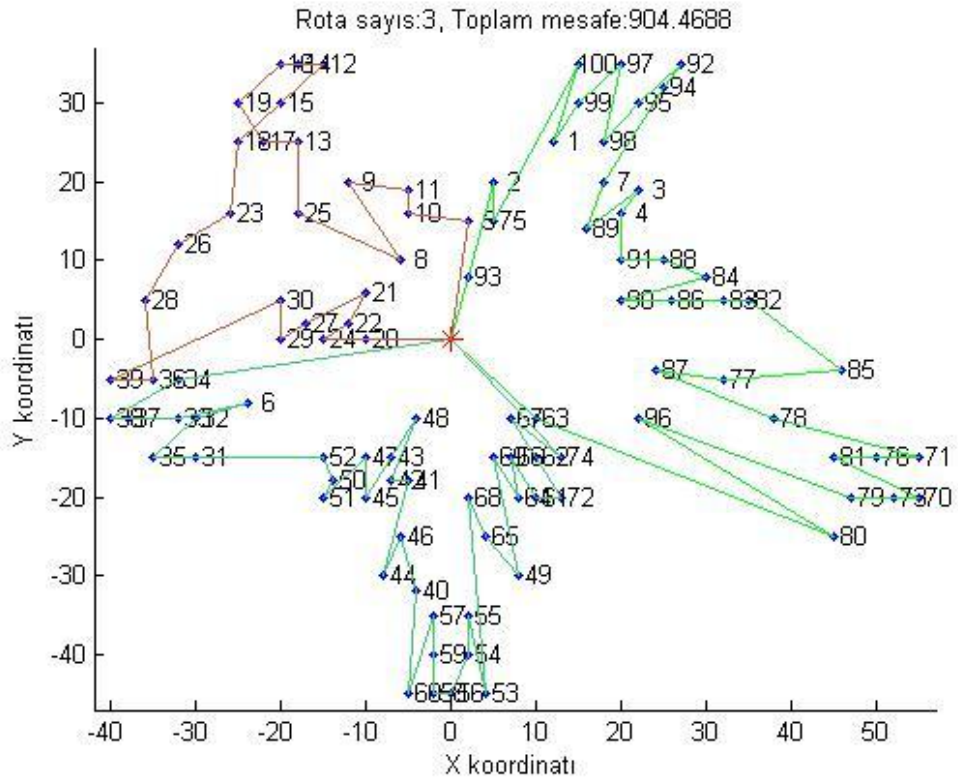
Şekil 4.6: 50 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)

Tablo 4.4: C201 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
C201	25	276,18	1
		229,06	1
		276,18	1
		276,18	1
		203,85	1
		276,18	1
		276,18	1
		237,90	1
		201,81	1
		226,48	1
	50	528,64	2
		520,18	2
		515,51	2
		464,45	2
		564,15	2
		501,71	2
		564,15	2
		508,16	2
		514,08	2
		500,47	2
	100	934,55	1
		1034,40	1
		1034,40	1
		1034,40	1
		986,39	1
		904,47	1
		1034,40	1
		984,17	1
		1010,50	1
		973,90	1



Şekil 4.7: 50 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değeri)



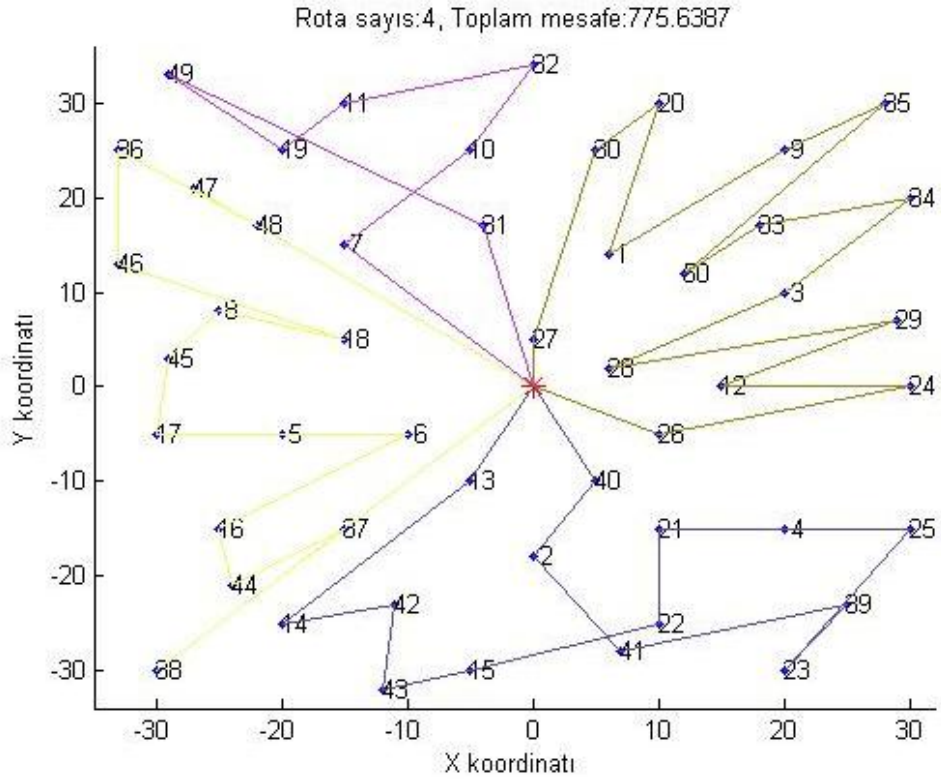
Şekil 4.8: 100 müşterilik C201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değeri)

4.1.1.3. Geliştirilen Yaklaşımın R101 Problem Kümesi İçin Uygulanması

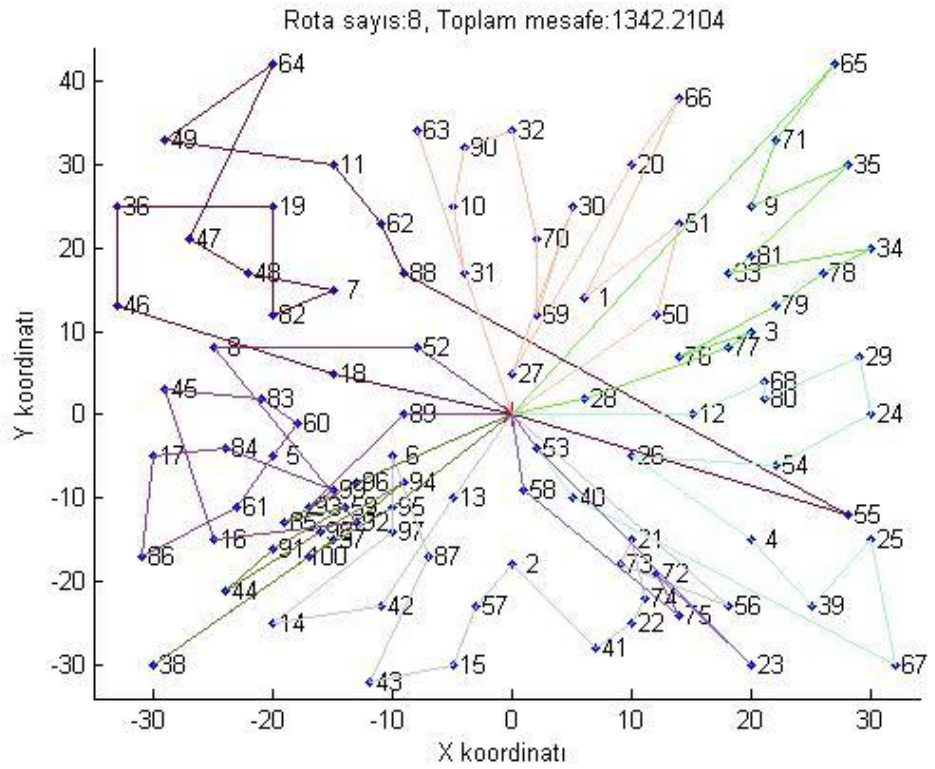
R101 problem kümesine ait verilerin 25, 50 ve 100'er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.5 ve Tablo 4.6'da gösterilmiştir. Tablo 4.5'ten iki görsele Şekil 4.9 ve Şekil 4.10'da, Tablo 4.6'dan iki görsele Şekil 4.11 ve Şekil 4.12'de yer verilmiştir.

Tablo 4.5: R101 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
R101	25	380,16	2
		377,28	2
		407,10	2
		407,10	2
		407,10	2
		395,18	2
		407,10	2
		392,43	2
		407,10	2
		407,10	2
	50	808,03	4
		814,65	4
		778,82	4
		810,22	4
		814,65	4
		775,64	4
		775,64	4
		780,45	4
		802,70	4
		790,83	4
	100	1465,8	8
		1317,8	8
		1465,8	8
		1396,8	8
		1359,2	8
		1465,8	8
		1465,8	8
		1430,7	8
		1411,5	8
		1465,8	8



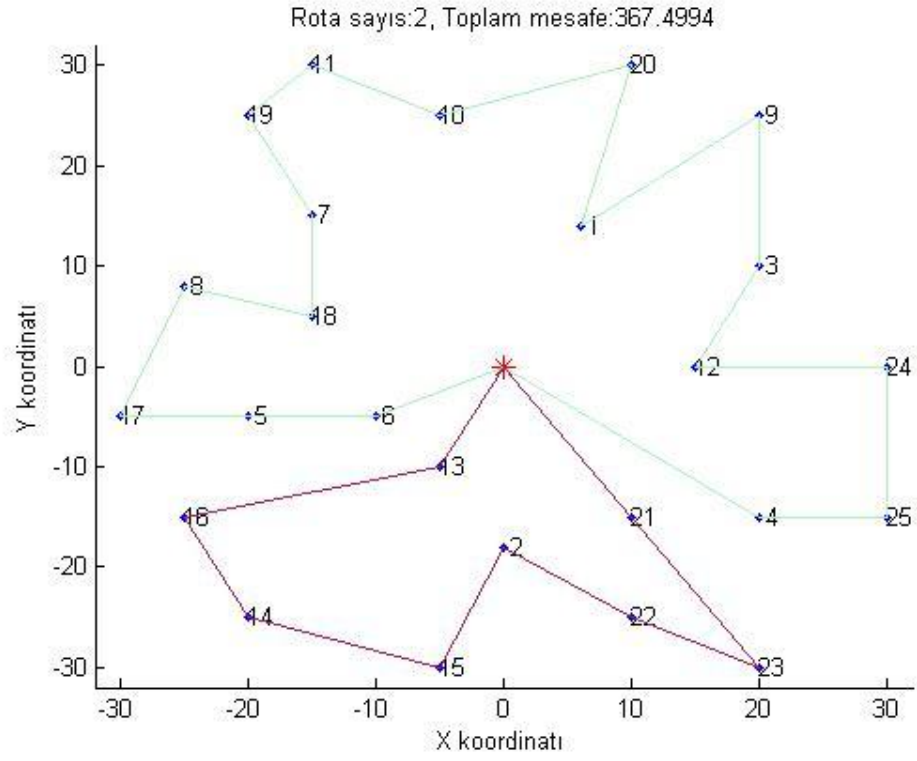
Şekil 4.9: 50 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)



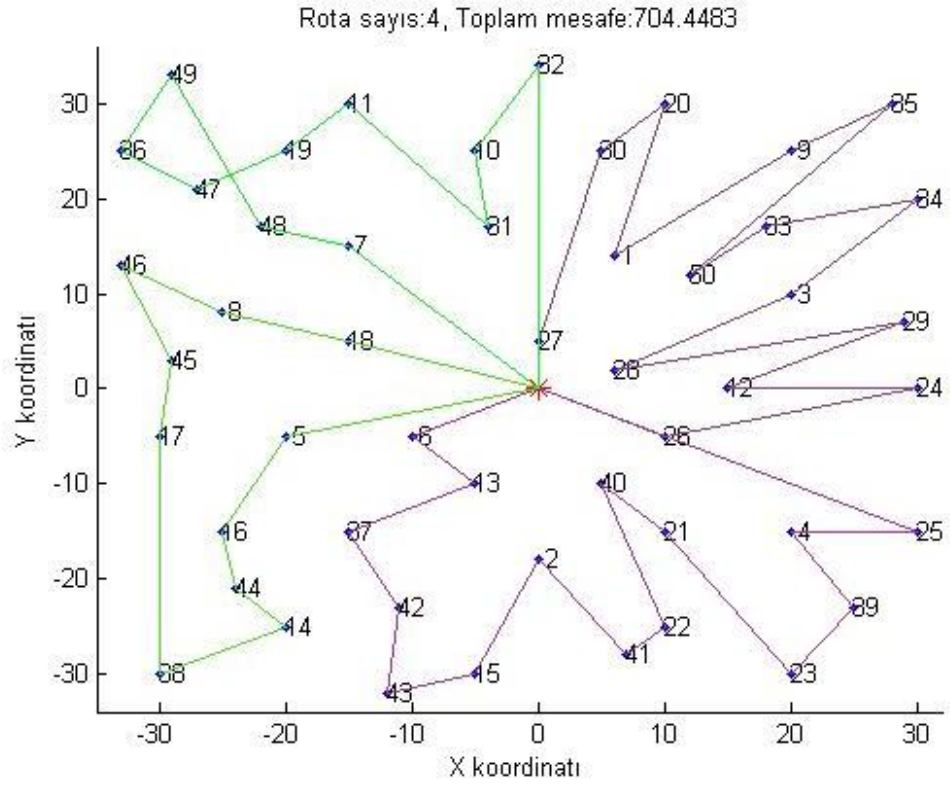
Şekil 4.10: 100 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)

Tablo 4.6: R101 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
R101	25	407,11	2
		368,52	2
		373,82	2
		407,11	2
		368,49	2
		407,11	2
		367,50	2
		369,68	2
		407,11	2
		367,50	2
	50	725,63	4
		814,65	4
		768,70	4
		806,66	4
		785,79	4
		814,65	4
		704,45	4
		730,94	4
		785,79	4
		767,64	4
	100	1465,8	8
		1419,9	8
		1397,6	8
		1321,0	8
		1355,6	8
		1428,0	8
		1465,8	8
		1347,6	8
		1465,8	8
		1465,8	8



Şekil 4.11: 25 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)



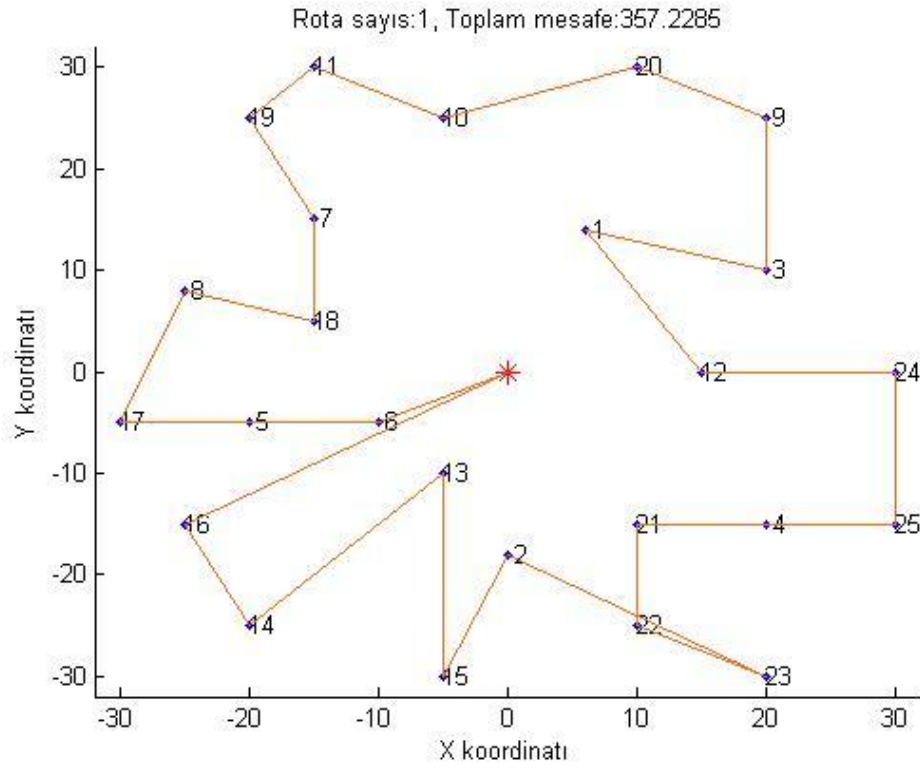
Şekil 4.12: 50 müşterilik R101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)

4.1.1.4. Geliştirilen Yaklaşımın R201 Problem Kümesi İçin Uygulanması

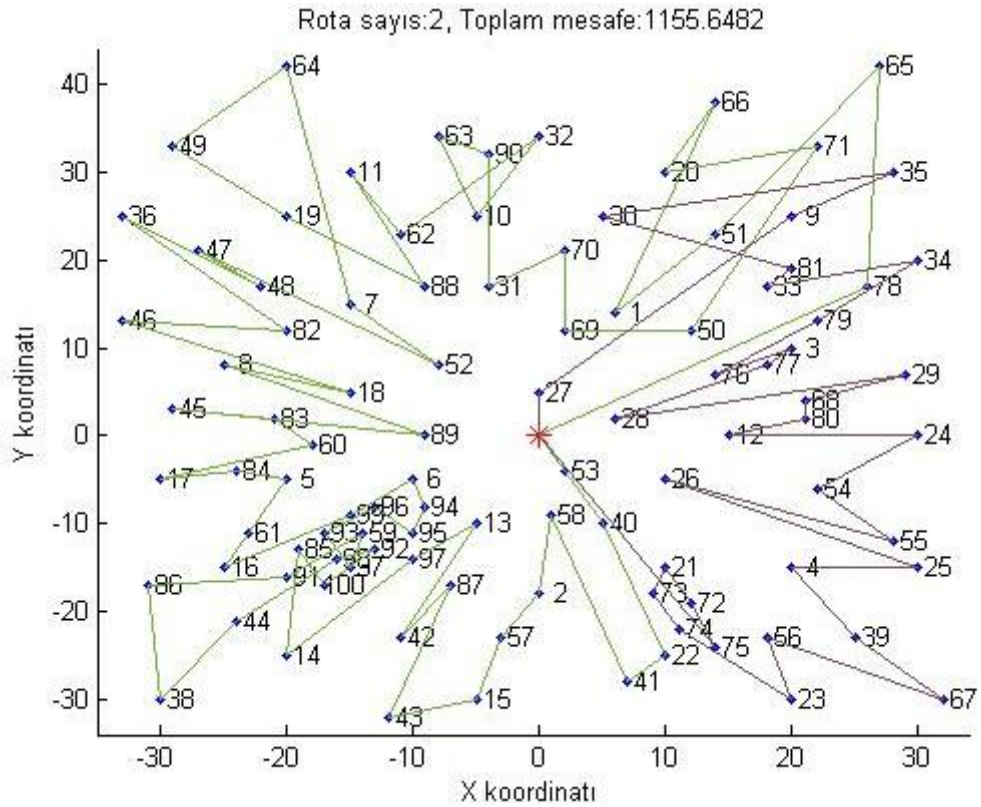
R201 problem kümesine ait verilerin 25, 50 ve 100'er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.7 ve Tablo 4.8'de gösterilmiştir. Tablo 4.7'den iki görsele Şekil 4.13 ve Şekil 4.14'de, Tablo 4.8'den iki görsele Şekil 4.15 ve Şekil 4.16'da yer verilmiştir.

Tablo 4.7: R201 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)

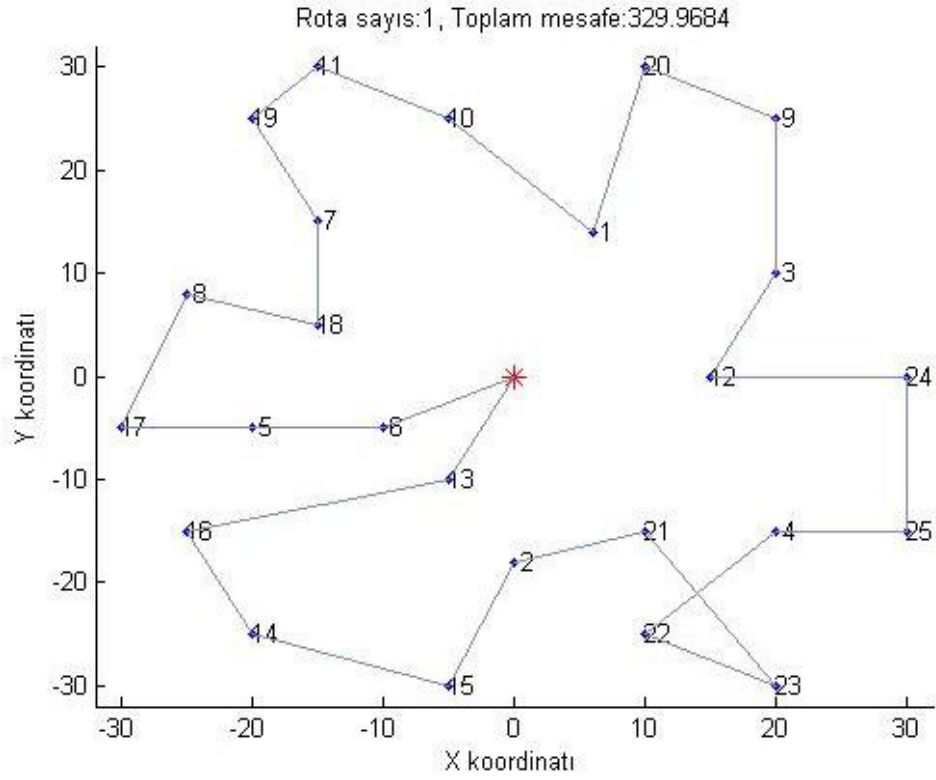
Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
R201	25	361,05	1
		361,05	1
		361,05	1
		361,05	1
		361,05	1
		361,05	1
		357,22	1
		361,05	1
		361,05	1
		361,05	1
	50	687,02	1
		675,21	1
		687,06	1
		687,02	1
		687,02	1
		687,02	1
		687,02	1
		645,91	1
		656,31	1
		687,02	1
	100	1159,6	2
		1279,2	2
		1194,7	2
		1252,8	2
		1279,2	2
		1279,2	2
		1244,7	2
		1279,2	2
		1270,1	2
		1155,6	2



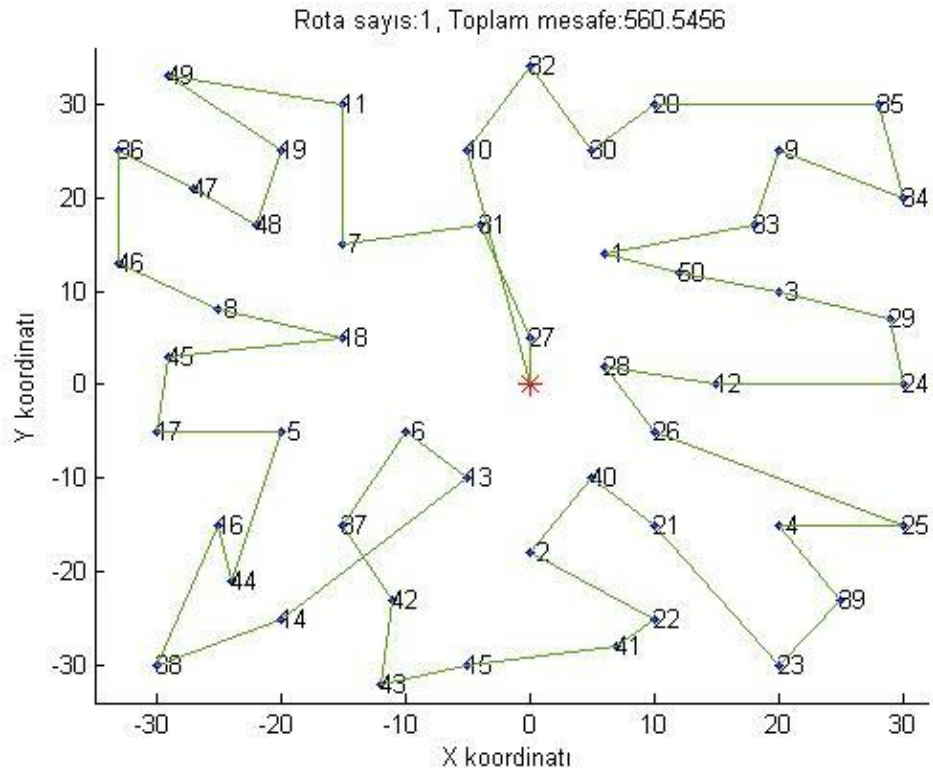
Şekil 4.13: 25 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)



Şekil 4.14: 100 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değeri)



Şekil 4.15: 25 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)



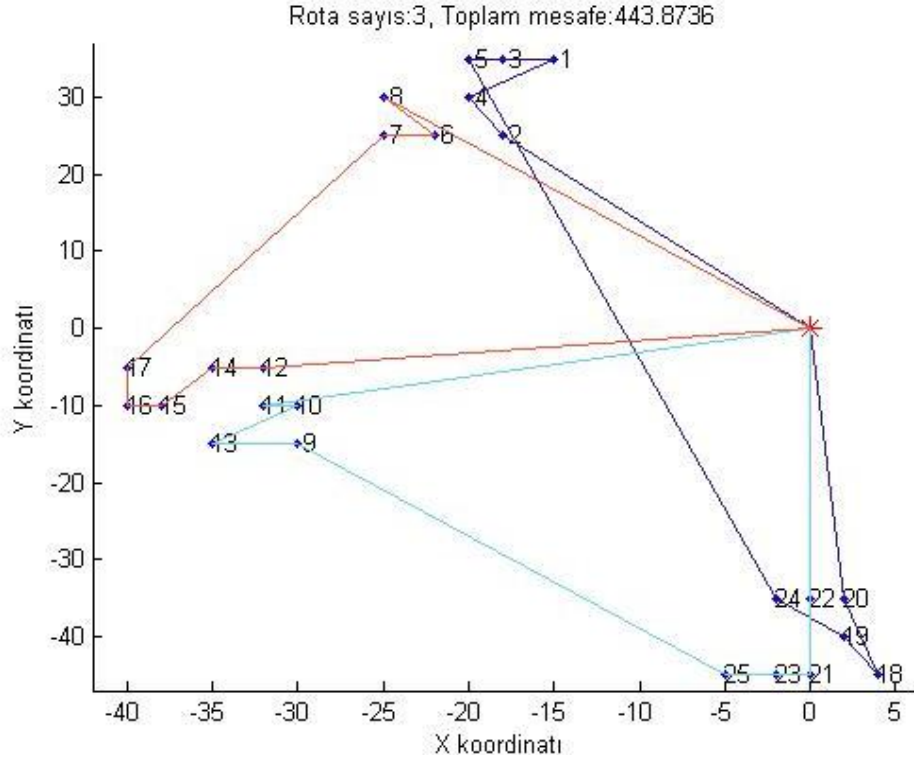
Şekil 4.16: 50 müşterilik R201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)

4.1.1.5. Geliştirilen Yaklaşımın RC101 Problem Kümesi İçin Uygulanması

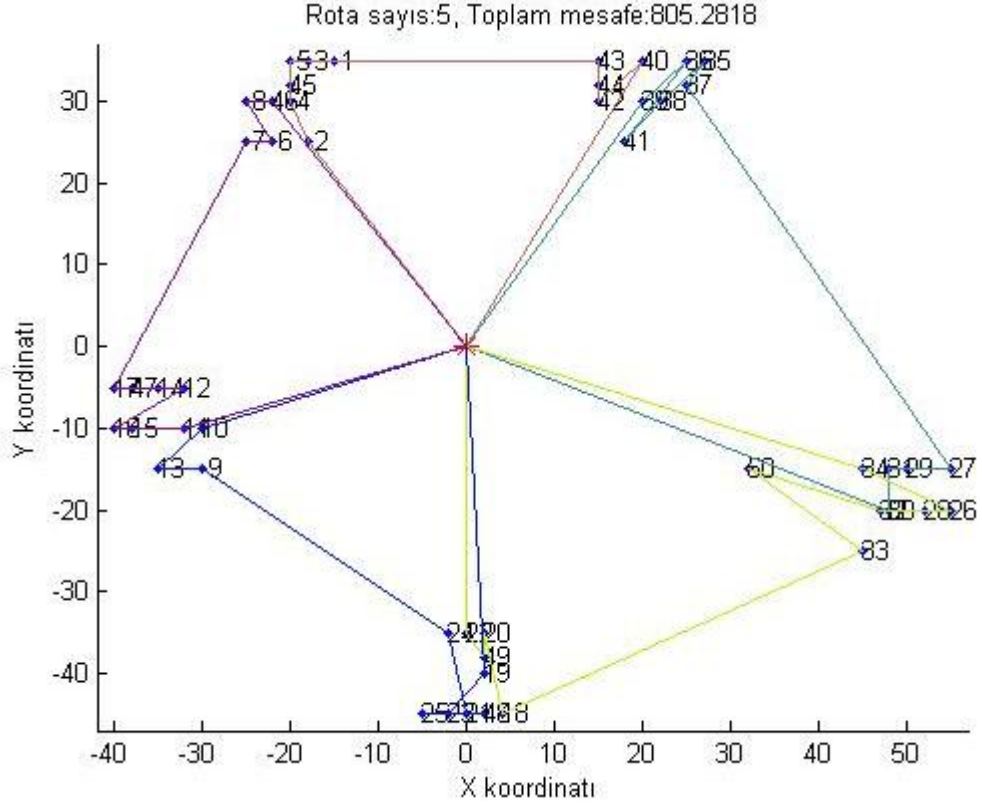
RC101 problem kümesine ait verilerin 25, 50 ve 100'er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.9 ve Tablo 4.10'da gösterilmiştir. Tablo 4.9'dan iki görsele Şekil 4.17 ve Şekil 4.18'de, Tablo 4.10'dan iki görsele Şekil 4.19 ve Şekil 4.20'de yer verilmiştir.

Tablo 4.9: RC101 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
RC101	25	461,54	3
		461,87	3
		452,56	3
		443,87	3
		461,87	3
		453,78	3
		458,84	3
		461,87	3
		461,87	3
		453,09	3
	50	826,40	5
		822,28	5
		832,40	5
		806,08	5
		805,28	5
		832,40	5
		832,40	5
		819,27	5
		820,96	5
		832,40	
	100	1490,1	9
		1564,3	9
		1536,9	9
		1654,7	9
		1624	9
		1680,9	9
		1680,9	9
		1573,2	9
		1680,9	9
		1526,6	9



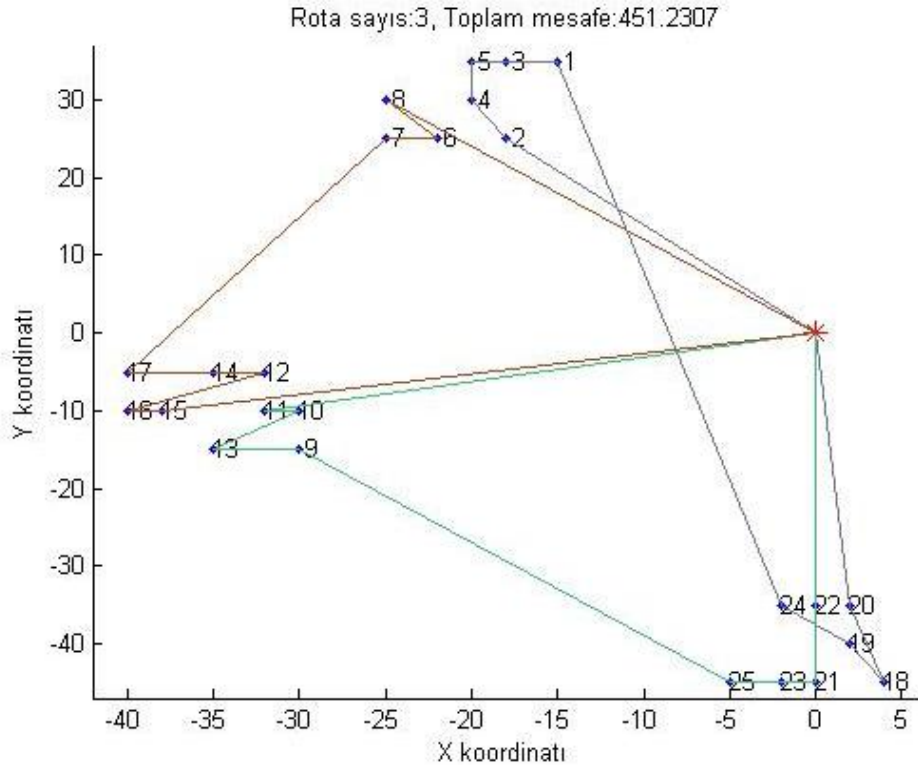
Şekil 4.17: 25 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)



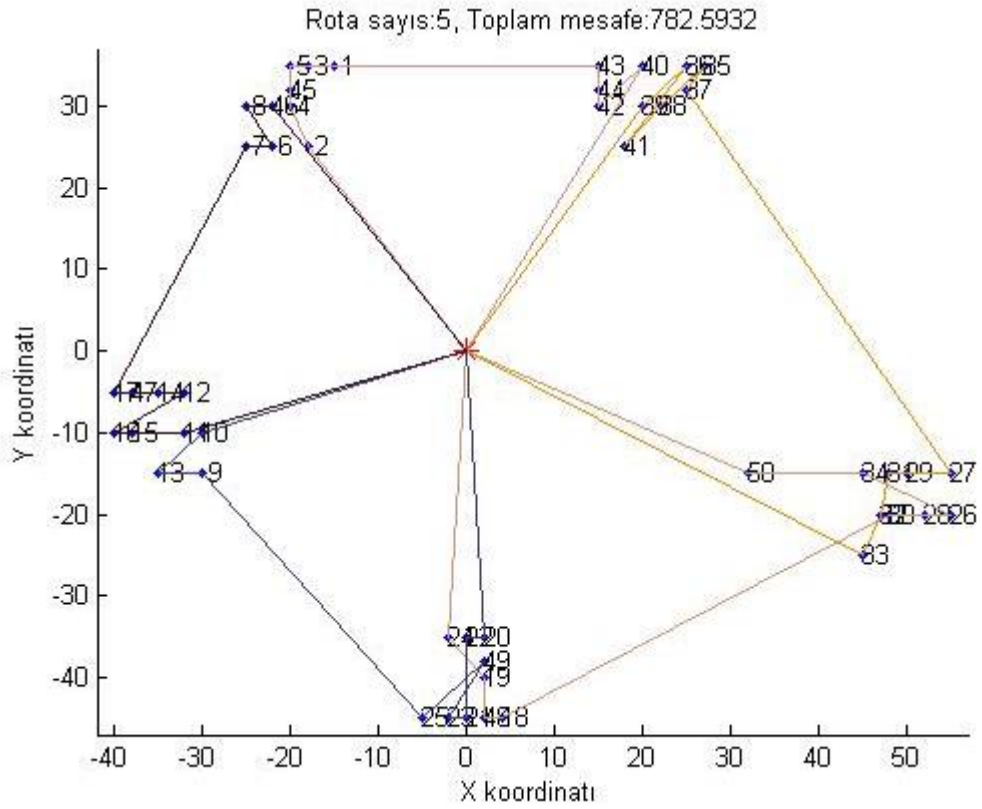
Şekil 4.18: 50 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)

Tablo 4.10: RC101 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
RC101	25	447,62	3
		451,76	3
		461,87	3
		461,87	3
		461,87	3
		452,54	3
		448,48	3
		461,87	3
		447,62	3
		451,23	3
	50	832,40	5
		782,59	5
		810,26	5
		783,50	5
		792,13	5
		832,40	5
		826,08	5
		803,81	5
		832,40	5
		832,40	5
	100	1591,9	9
		1611,2	9
		1486,3	9
		1635,2	9
		1680,9	9
		1587,7	9
		1563,6	9
		1637,6	9
		1680,9	9
		1528,5	9



Şekil 4.19: 25 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değeri)



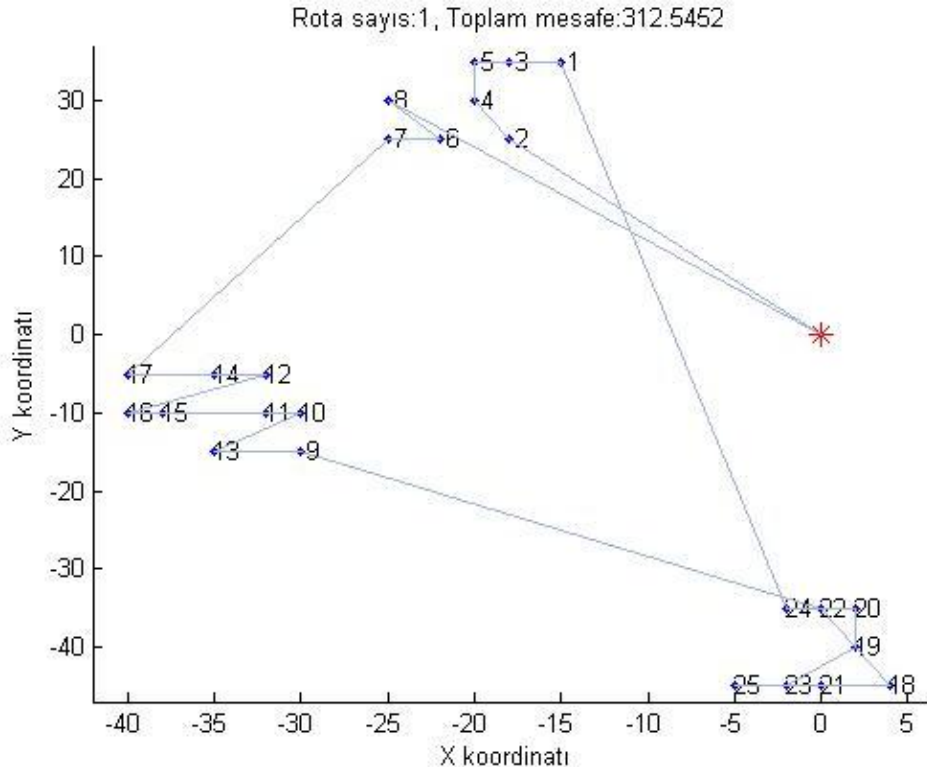
Şekil 4.20: 50 müşterilik RC101 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değeri)

4.1.1.6. Geliştirilen Yaklaşımın RC201 Problem Kümesi İçin Uygulanması

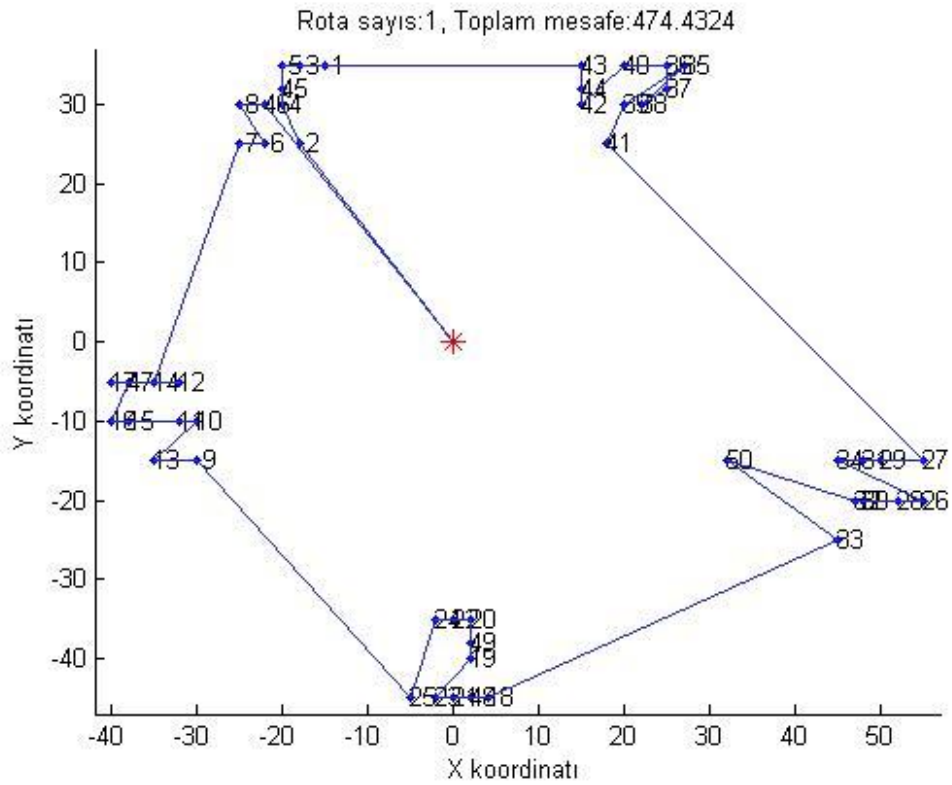
RC201 problem kümesine ait verilerin 25, 50 ve 100'er müşterilik gruplarına ÇAGDKA yaklaşımının ayrı ayrı on kere uygulanmasına ait sonuçlar Tablo 4.11 ve Tablo 4.12'de gösterilmiştir. Tablo 4.11'den iki görsele Şekil 4.21 ve Şekil 4.22'de, Tablo 4.12'dan iki görsele Şekil 4.23 ve Şekil 4.24'de yer verilmiştir.

Tablo 4.11: RC201 problem kümesi için sonuç tablosu (yüksek dış döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
RC201	25	341,42	1
		341,42	1
		305,71	1
		341,42	1
		341,42	1
		312,55	1
		341,42	1
		318,38	1
		341,42	1
		341,42	1
	50	516,36	1
		512,83	1
		516,36	1
		516,36	1
		516,36	1
		516,36	1
		516,36	1
		516,36	1
		474,43	1
		516,36	1
	100	1354,6	2
		1354,6	2
		1354,6	2
		1354,6	2
		1228,2	2
		1354,6	2
		1354,6	2
		1354,6	2
		1354,6	2
		1311,8	2



Şekil 4.21: 25 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)

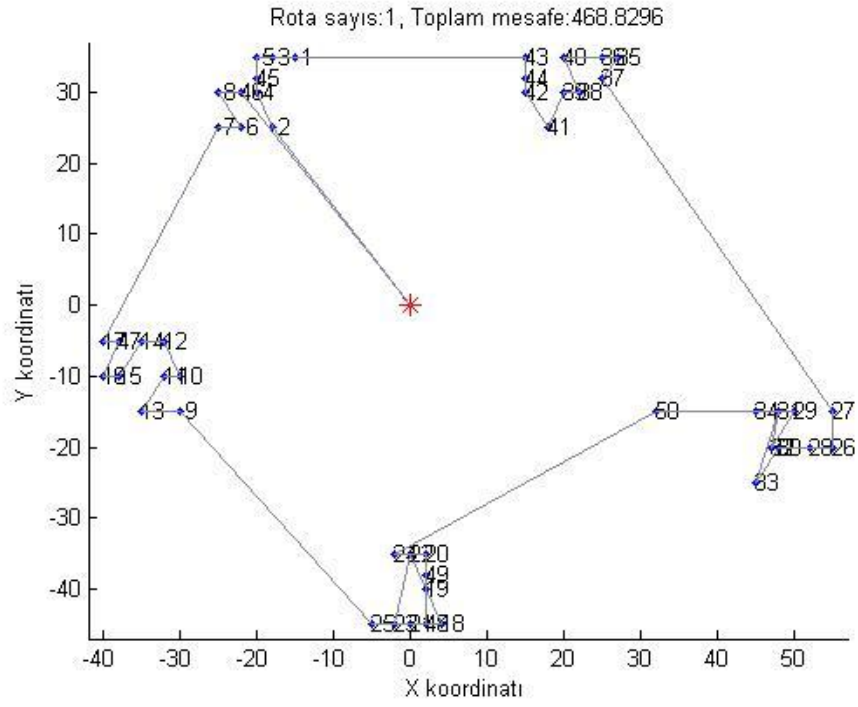


Şekil 4.22: 50 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek dış döngü değerli)

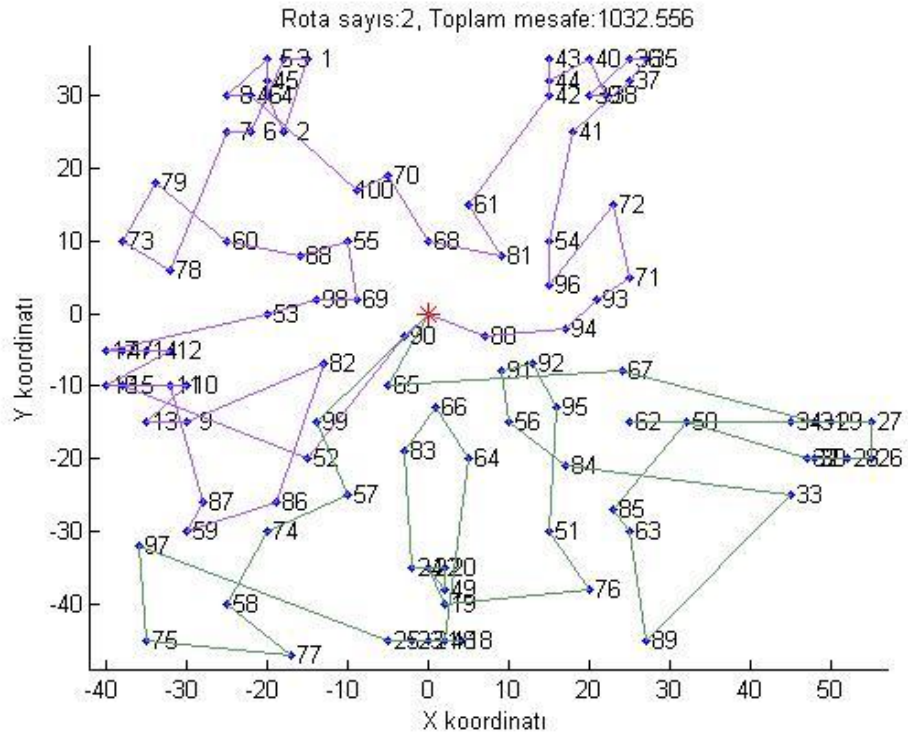
Tablo 4.12: RC201 problem kümesi için sonuç tablosu (yüksek iç döngü değerli)

Problem kümesi	Müşteri sayısı	Mesafe	Araç/rota sayısı
RC201	25	341,42	1
		294,72	1
		341,42	1
		341,42	1
		341,42	1
		341,42	1
		341,42	1
		341,42	1
		341,42	1
		341,42	1
	50	516,36	1
		516,36	1
		516,36	1
		516,36	1
		516,36	1
		516,36	1
		468,83	1
		516,36	1
		516,36	1
		516,36	1
	100	1354,6	2
		1354,6	2
		1354,6	2
		1354,6	2
		1354,6	2
		1032,6	2
		1161,3	2
		1153,7	2
		1110,1	2
		1354,6	2

Süpürme algoritmasından elde edilen başlangıç çözümlerine ait grafiklere 3.6.1’de yer verilmişti. ÇAGDKA uygulaması sonucu elde edilen iyileştirmeler, başlangıç çözümü grafiklerine bakıldığında da görülmektedir. Tablo 4.35’ten 4.46’ya kadar onar adet sonuçları sunulan ÇAGDKA yaklaşımının bu gözlemleri de kapsayan toplam yirmi beşer kere çalıştırılması sonucu elde edilen en iyi değerler ile yirmi beş değer ortalaması Tablo 4.13 ve 4.14’te gösterilmiştir. Bölüm 5’te ise Tablo 5.1 ve 5.2’de bu en iyi sonuçlar ile ÇAGA yaklaşımının en iyi sonuçları birlikte gösterilerek kıyaslamalar sunulmuştur. Ortalama değerler kıyaslama tablolarında kullanılmamış fakat daha güvenilir ortalamalar hesaplanması için 25 değer alınmıştır.



Şekil 4.23: 50 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)



Şekil 4.24: 100 müşterilik RC201 problem kümesi için elde edilen en iyi sonuç (yüksek iç döngü değerli)

Tablo 4.13: ÇAGDKA yaklaşımında dış döngü sayısı yüksek (200) tutularak elde edilen sonuçlar

Problem kümesi	Müşteri sayısı	Toplam talep	Araç kapasitesi	Toplam talep/kapasite	Gerekli olan en az araç/rota sayısı	Süpürme algoritmasının araç/rota sayıları	Bulunan en iyi değerler	25 kere programın çalıştırılması sonucu elde edilen ortalama değerler	Ortalama işletim süresi (sn)
C101	25	460	200	2,30	3	3	239,03	260,84	10,94
	50	860	200	4,30	5	5	510,23	523,62	17,79
	100	1810	200	9,05	10	10	1146,90	1178,05	29,99
C201	25	460	700	0,66	1	1	230,26	270,60	6,3
	50	860	700	1,23	2	2	535,47	553,93	27,8
	100	1810	700	2,59	3	3	1002,60	1027,89	39,6
R101	25	332	200	1,66	2	2	377,29	403,29	12,7
	50	721	200	3,61	4	4	764,62	798,86	15,9
	100	1458	200	7,29	8	8	1317,80	1413,58	30,1
R201	25	332	1000	0,33	1	1	357,23	360,86	6,5
	50	721	1000	0,72	1	1	645,92	682,84	10,3
	100	1458	1000	1,46	2	2	1155,60	1255,61	42,7
RC101	25	540	200	2,70	3	3	443,87	459,15	10,8
	50	970	200	4,85	5	5	805,28	826,41	15,6
	100	1724	200	8,62	9	9	1490,10	1617,69	30,2
RC201	25	540	1000	0,54	1	1	305,71	336,81	6,2
	50	970	1000	0,97	1	1	474,43	513,96	13
	100	1724	1000	1,72	2	2	1228,20	1337,68	53

Tablo 4.14: ÇAGDKA yaklaşımında iç döngü sayısı yüksek (20) tutularak elde edilen sonuçlar

Problem kümesi	Müşteri sayısı	Toplam talep	Araç kapasitesi	Toplam talep/kapasite	Gerekli olan en az araç/rota sayısı	Süpürme algoritmasının rota sayıları	Bulunan en iyi değerler	25 kere programın çalıştırılması sonucu elde edilen ortalama değerler	Ortalama işletim süresi (sn)
C101	25	460	200	2,30	3	3	237,04	252,23	6,23
	50	860	200	4,30	5	5	499,60	516,21	11,50
	100	1810	200	9,05	10	10	1149,60	1177,46	24,31
C201	25	460	700	0,66	1	1	201,81	251,00	6,69
	50	860	700	1,23	2	2	464,45	523,00	4,81
	100	1810	700	2,59	3	3	904,47	989,44	5,42
R101	25	332	200	1,66	2	2	367,50	385,95	2,31
	50	721	200	3,61	4	4	704,45	767,94	2,57
	100	1458	200	7,29	8	8	1321,00	1404,56	2,70
R201	25	332	1000	0,33	1	1	329,97	355,26	2,11
	50	721	1000	0,72	1	1	560,55	660,77	2,20
	100	1458	1000	1,46	2	2	991,40	1205,48	2,51
RC101	25	540	200	2,70	3	3	447,62	455,92	2,13
	50	970	200	4,85	5	5	782,59	812,95	2,26
	100	1724	200	8,62	9	9	1486,30	1597,39	2,57
RC201	25	540	1000	0,54	1	1	294,72	336,74922	2,01
	50	970	1000	0,97	1	1	468,83	511,60	2,14
	100	1724	1000	1,72	2	2	1032,6	1258,53	2,23

Tablo 4.15: ÇAGDKA yaklaşımının uygulanmasında yüksek iç döngü veya dış döngünün tercih edilmesinde elde edilen sonuçların kıyaslama tablosu

Araç/rota sayısı	Yüksek dış döngü değeri kullanımı sonucu elde edilen en iyi mesafe değerleri	Yüksek iç döngü değeri kullanımı sonucu elde edilen en iyi mesafe değerleri
3	239,03	237,04
5	510,23	499,60
10	1146,90	1149,60
1	230,26	201,81
2	535,47	464,45
3	1002,60	904,47
2	377,29	367,50
4	764,62	704,45
8	1317,80	1321,00
1	357,23	329,97
1	645,92	560,55
2	1155,60	991,40
3	443,87	447,62
5	805,28	782,59
9	1490,10	1486,30
1	305,71	294,72
1	474,43	468,83
2	1228,20	1032,6

Tablo 4.15 incelendiğinde yüksek iç döngü değerinin kullanılması durumunda mesafe değerlerinin biraz daha iyi olduğu görülmektedir. Bu durumda programın bir sonuç için geçirdiği sürelerin de kıyaslanması faydalıdır.

Tablo 4.16: ÇAGDKA yaklaşımında yüksek iç ve yüksek dış döngü kullanımı ile bulunan en iyi değerlerin işlem sürelerinin (sn) kıyaslama tablosu

Problem kümesi	Müşteri Sayısı	Yüksek dış döngü	Yüksek iç döngü
C101	25	10,94	6,23
	50	17,79	11,50
	100	29,01	24,31
C201	25	6,3	4,69
	50	27,8	4,8
	100	39,6	5,4
R101	25	12,7	2,3
	50	15,9	2,5
	100	30,1	2,7
R201	25	6,5	2,1
	50	10,3	2,2
	100	42,7	2,5
RC101	25	10,8	2,1
	50	15,6	2,2
	100	30,2	2,5
RC201	25	6,2	2
	50	13	2,1
	100	53	2,2

Tablo 4.13 ile Tablo 4.14 süpürme algoritmasının başlangıç çözümü olarak seçilmesi durumunda elde edilen sonuçlara ait verileri içermektedir. ÇAGDKA yaklaşımının kendi içinde iki farklı uygulaması kıyaslandığında Tablo 4.15 ve 4.16'dan de görüldüğü gibi, yüksek iç döngü değeri kullanıldığında, çok daha kısa sürelerde ve daha iyi sonuçlar sağlamıştır. Toplam on sekiz en iyi sonuç değerinin on altı tanesinde yüksek iç döngü değerinin daha iyi sonuçlar verdiği görülmüştür. Bu yüzden diğer başlangıç

yöntemi için, uygulama, önce bir veri kümesinde test edilmiştir. Bu test verileri de sonraki başlık altında sunulmuş ve değerlendirme sonucunda yüksek iç döngü değeri verilerek tüm problem kümelerinin çözülmesine karar verilmiştir.

4.1.2. Başlangıç Çözümü Oluşturmak İçin İleri İtmeli Ekleme Sezgisel Yönteminin Kullanıldığı Durum

Bu başlık altında her problem kümesi için elde edilen ve Bölüm 3.6.2’de sunulan başlangıç çözümlerinin ÇAGDKA yaklaşımı için kullanılmasıyla elde edilen sonuçlar üzerinde durulacaktır.

Yüksek iç döngü veya yüksek dış döngü kullanımını seçmek için C101 problem kümesinin yüz müşterilik grubu ele alınmıştır. Bulunan çözüm alternatifleri için rotalara karşılık gelen en kısa mesafe süreleri Tablo 4.17’de gösterilmiştir.

Tablo 4.17: Dış döngü ile iç döngü sayılarının farklılığına göre sonuçtaki başarı durumu

Problem kümesi	Araç/rota sayısı	Yüksek dış döngü		Yüksek iç döngü	
		Mesafe	Ortalama işletim süresi (sn)	Mesafe	Ortalama işletim süresi (sn)
C101/100	10	1666,6	47	1325,4	23,7
	11	1582,9		1162	
	12	1,4613		1115,1	

İç döngü sayısının yüksek tutulması durumunda daha kısa sürede daha iyi sonuçlar elde edildiği için kalan problemlere yüksek iç döngü uygulanacaktır.

ÇAGDKA yaklaşımının uygulanması sonucunda, tek bir sonuç değil, birbirine iki amaç açısından baskın olmayan çözüm değerlerine sahip Pareto çözüm kümesi sunulmaktadır. Programın her çalıştırılışında birden fazla çözüm sonucu vermesi şart değildir. Pareto kümesinin güncelleştirilmesi esnasında tek bir baskın çözüm varsa diğer çözümler kümede bulunamayacaklardır. Dolayısıyla tek bir çözüm elde edilecektir. Pareto kümesinin farklı alternatifler sunma özelliği bu başlangıç yönteminde görülmüştür. Rota sayısı düşük fakat mesafesi yüksek bir çözüm ile rota sayısı yüksek mesafesi düşük bir çözümü aynı anda sunabilmiştir. Aradaki değerler kadar da alternatif sayısı bulunabilmektedir. Buna örnek olarak programın bir kere çalıştırılmasıyla Pareto

çözüm kümesinde tutulan ve sonuç olarak verilen çözüm değerleri Tablo 4.18’de gösterilmiştir.

Tablo 4.18: ÇAGDKA uygulaması için programın bir kere çalıştırılması sonucu pareto çözüm kümesi

Programın çalıştırılma sırası	Problem kümesi	Araç sayısı	Mesafe
1	R101	5	1662,3
		3	1761,7
		2	1830,9

Geliştirilen yaklaşım, C101 probleminin yüz müşterilik grubuna on kere ayrı ayrı uygulanınca Tablo 4.19’deki on farklı pareto çözüm kümesi değerini sunmaktadır. Bazı Pareto çözüm kümelerinde bahsedildiği gibi tek çözüm değeri bulunmuştur. Bazılarında ise iki farklı alternatif sunmuştur.

Tablo 4.19: C101 problem verisinin 100 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu

Programın çalıştırılma sırası	Rota sayısı	Mesafe
1	11	1175,5
2	12	1169,8
3	12	1115,1
4	11	1162
5	13	1176,7
	12	1349,8
6	11	1340,6
	10	1561,5
7	15	1235,3
	12	1237,7
8	15	1250,8
	13	1322,6
9	14	1126
	13	1133,8
10	12	1196,9
	10	1325,4

Tablo 4.20: C101 problem verisinin 50 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu

Programın çalıştırılma sırası	Mesafe	Rota/araç sayısı
1	486,07	6
2	510,04	6
3	511,19	8
4	484,22	7
	484,87	6
5	485,91	7
6	465,08	6
7	556,59	7
	604,32	5
8	457,82	6
9	474,85	7
10	468,64	6
	486,42	1

Tablo 4.21: C101 problem verisinin 25 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu

Programın çalıştırılma sırası	Mesafe	Rota/araç sayısı
1	228,16	4
2	223,12	4
3	194,30	3
4	245,61	3
5	222,84	4
6	222,65	4
7	268,54	4
8	253,40	4
9	228,09	3
10	227,48	3

Tablo 4.22: R201 problem verisinin 100 müşterilik grubuna ÇAGDKA yaklaşımı uygulanmasının sonuç tablosu

Programın çalıştırılma sırası	Mesafe	Rota/araç sayısı
1	1662,3	5
	1761,7	3
	1830,9	2
2	1444,9	3
	2263,5	2
3	1531,8	2
4	1604	3
5	1737,3	8
	1786,9	4
6	1696,7	3
7	1591,4	9
	1652,8	4
8	1459	5
	1770,4	3
9	1459	5
	1770,4	3

Problem kümelerinin 25, 50 ve 100'er müşterilik gruplarına MATLAB programında ÇAGDKA yaklaşımı uygulandığında Tablo 4.17 ile 4.22 arasında verildiği gibi birden fazla alternatif çözüm barındıran veya tek baskın çözüme sahip Pareto kümesi sonuçları elde edilmektedir. Tüm sonuçlar için birden fazla sayıda alternatif bulunduğundan sonuç tablolarının hepsi ayrı ayrı sunulmamıştır. Onar sonuç değerini gösteren örnek olarak gösterilen bu tablolardaki gibi tüm problem kümeleri için, program yirmi beşer kez çalıştırılmıştır. Ardından, elde edilen sonuçların kıyaslanması açısından her bir yirmi beş farklı Pareto kümesi içerisinde bulunan rota sayılarına karşılık gelen en iyi mesafe değerleri seçilmiş ve sonuç tablosu (Tablo 4.23) hazırlanmıştır.

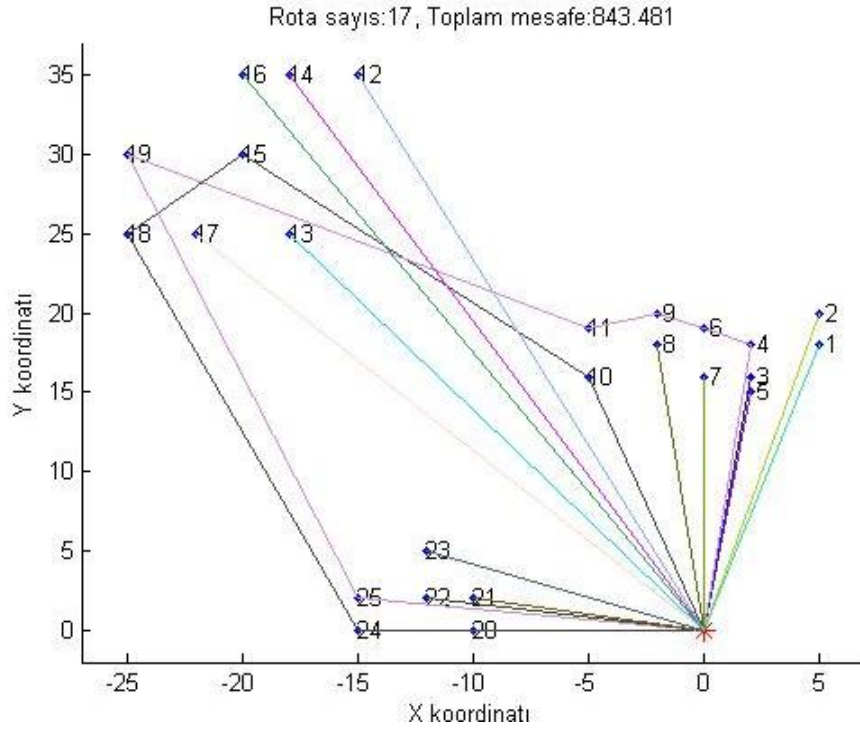
Tablo 4.23: ÇAGDKA yaklaşımı ile elde edilen sonuçların tablosu (İİES başlangıç çözüm yöntemi ve yüksek iç döngü değeri ile)

Problem kümesi	Müşteri sayısı	Araç kapasitesi	ÇAGDKA yaklaşımı ile elde edilen alternatifler	
			Araç/rota sayısı	Katedilen mesafe
C101	25	200	3	194,30
			4	221,40
	50	200	5	486,00
			6	413,81
	100	200	10	1325,40
			11	1162,00
12	1095,10			
C201	25	700	2	234,31
			50	700
	4	444,00		
	5	473,00		
	100	700	5	688,80
			6	843,11
8			1101,40	
R101	25	200	3	385,92
			4	411,84
			5	423,03
	50	200	5	652,28
			6	649,83
			7	674,40
			100	200
	10	1060		
	11	1100		
	12	1150		
13	1200			
14	1160			
R201	25	1000	3	482,78
			4	392,12
	50	1000	3	553,81
			4	592,55
			5	667,13
			7	650,91
	100	1000	4	982,22
			5	936,62
			6	1213,40
8			1512,30	
9			1478,50	

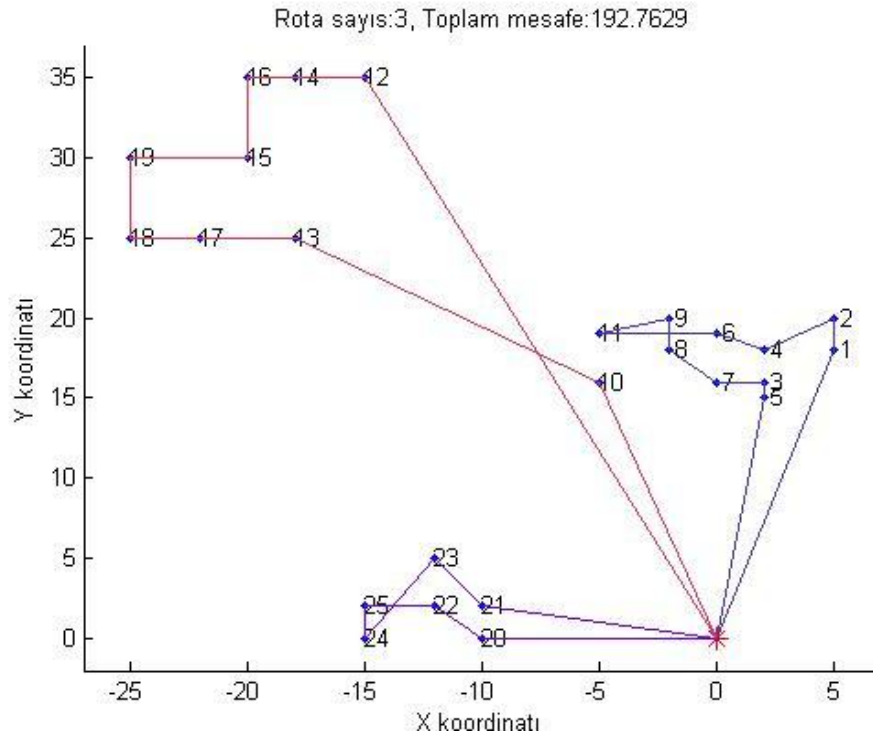
Tablo 4.23: devamı

Problem kümesi	Müşteri sayısı	Araç kapasitesi	ÇAGDKA yaklaşımı ile elde edilen alternatifler		
			Araç/rota sayısı	Katedilen mesafe	
RC101	25	200	3	312,28	
			4	373,45	
			5	447,07	
	50	200	5	536,15	
			6	628,74	
			7	768,80	
			8	794,64	
	100	200	11	1335,80	
			12	1377,40	
			13	1463,70	
			14	1605,20	
			15	1569,00	
			16	1490,50	
			17	1629,70	
	RC201	25	1000	2	333,82
				3	318,01
		50	1000	4	998,93
5				1248,60	
5				1429,60	
100		1000	6	1342,40	
			8	1468,20	
			9	1924,30	

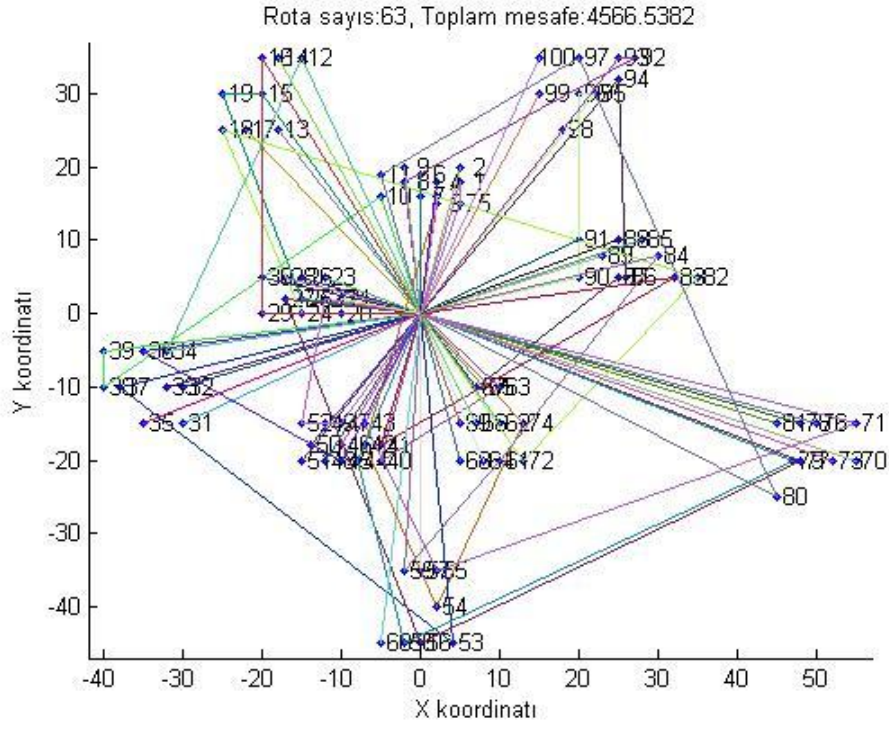
Tablo 4.23’de bulunan alternatif çözümler içinden en iyi sonuç değerleri için bazı görsellere Şekil 4.25 ile 4.34 arasında yer verilmiştir. İyileşmenin görülmesi için her bir örnekten önce başlangıç çözümüne ait görsel de verilmiştir.



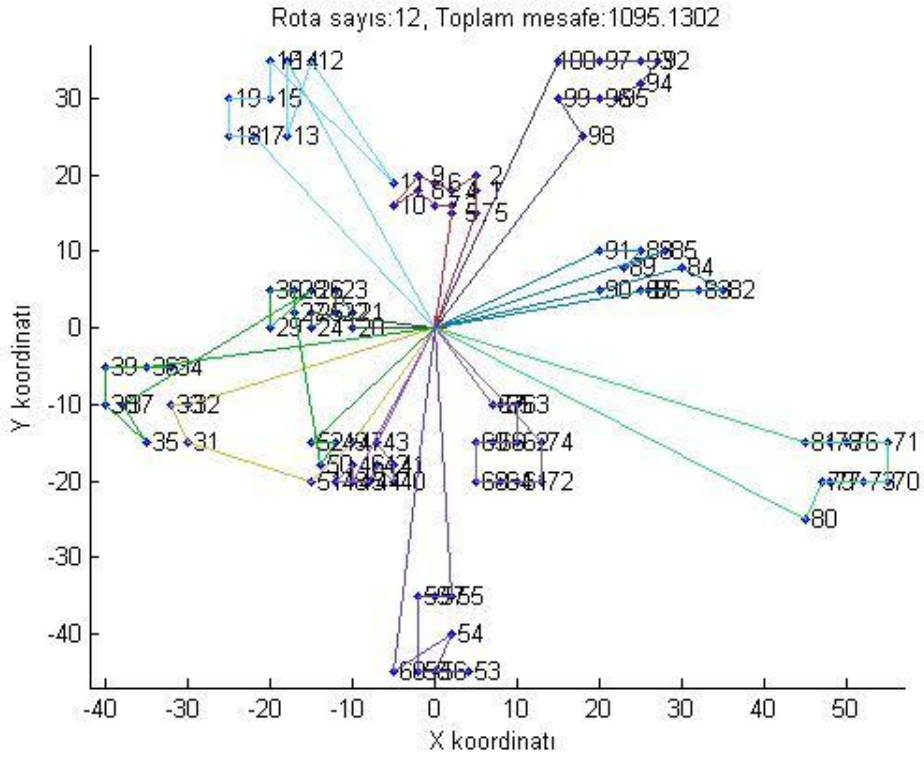
Şekil 4.25: 25 müşterilik C101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözümü grafiği



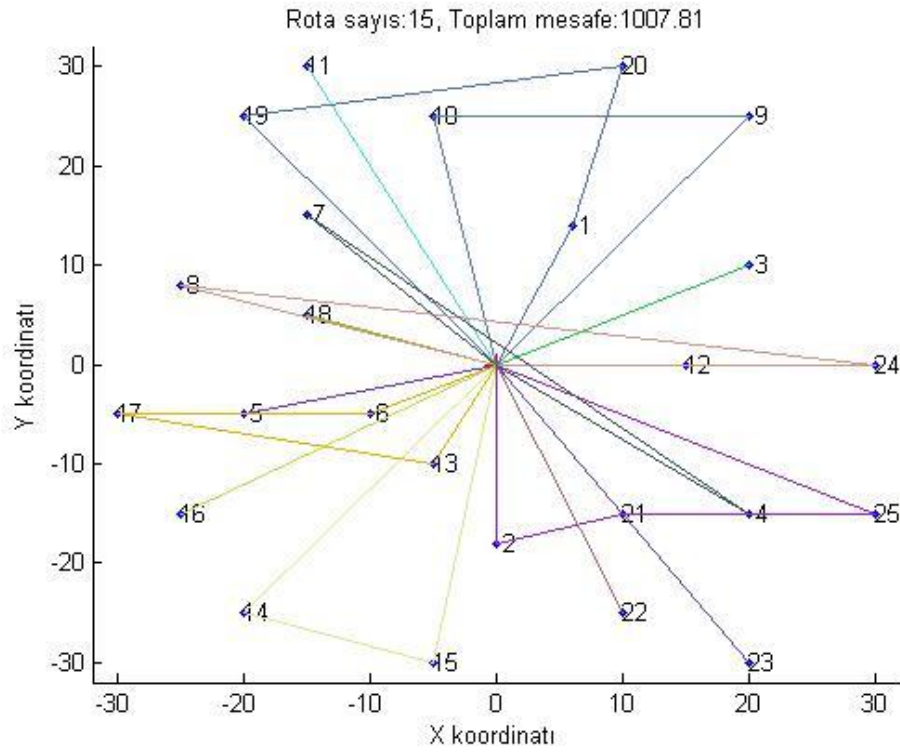
Şekil 4.26: 25 müşterilik C101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik



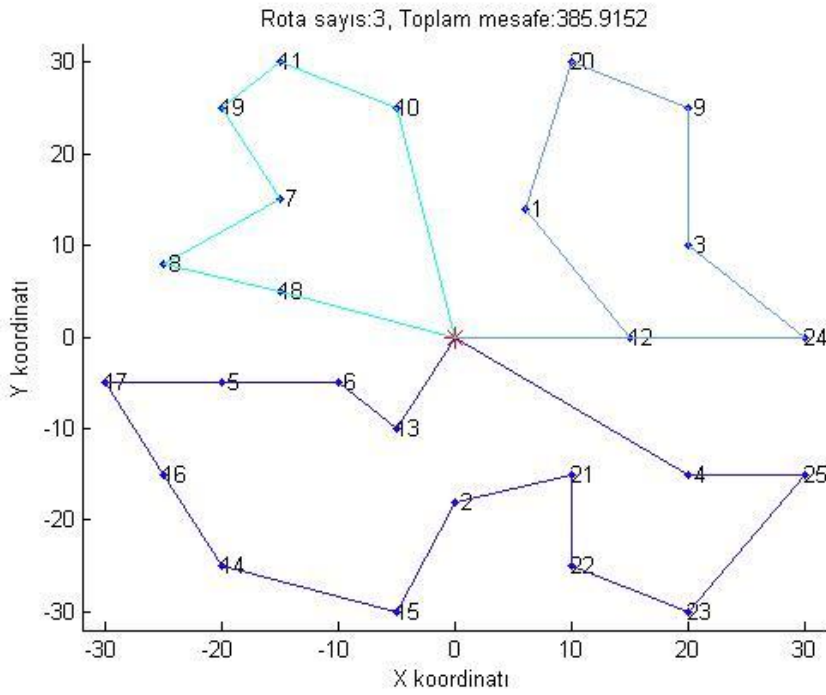
Şekil 4.27: 100 müşterilik C101 problem kümesi için İES yöntemi ile elde edilen başlangıç çözümü grafiği



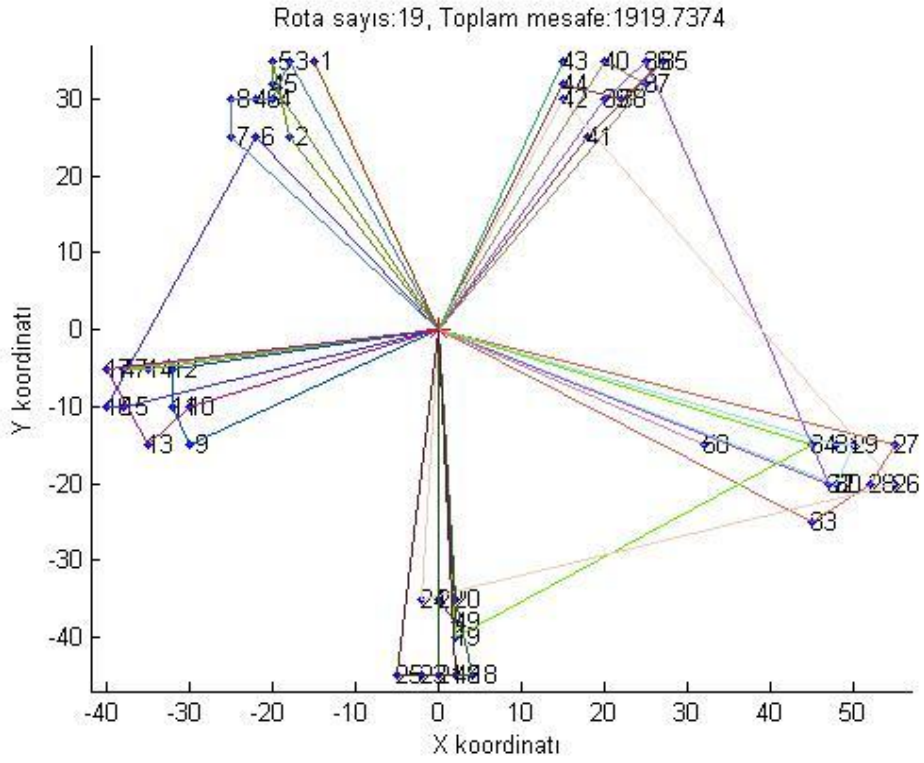
Şekil 4.28: 100 müşterilik C101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik



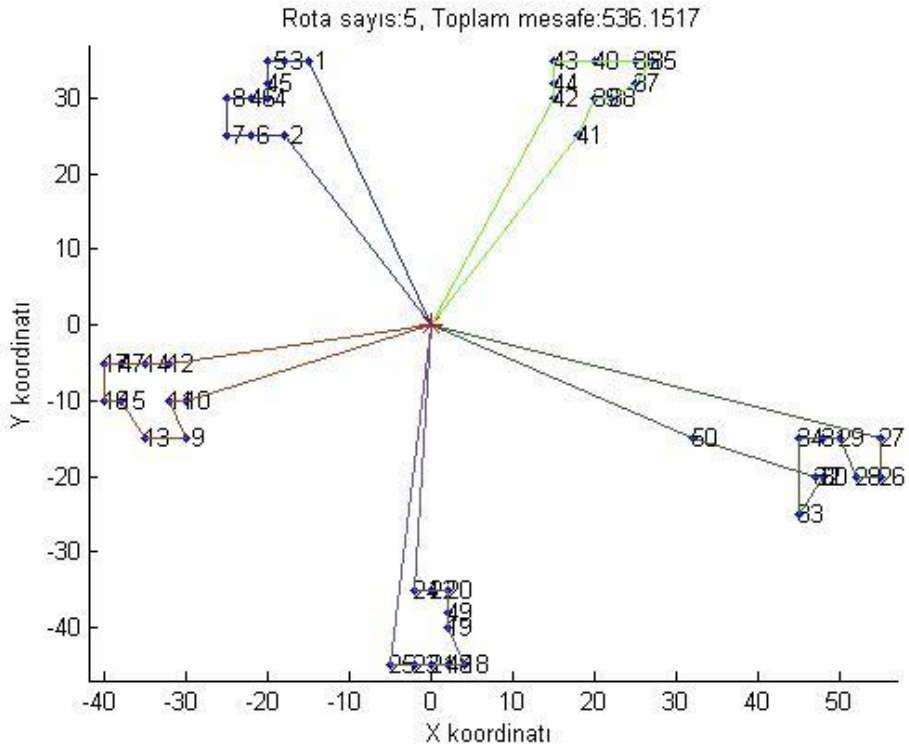
Şekil 4.29: 25 müşterilik R101 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözümü grafiği



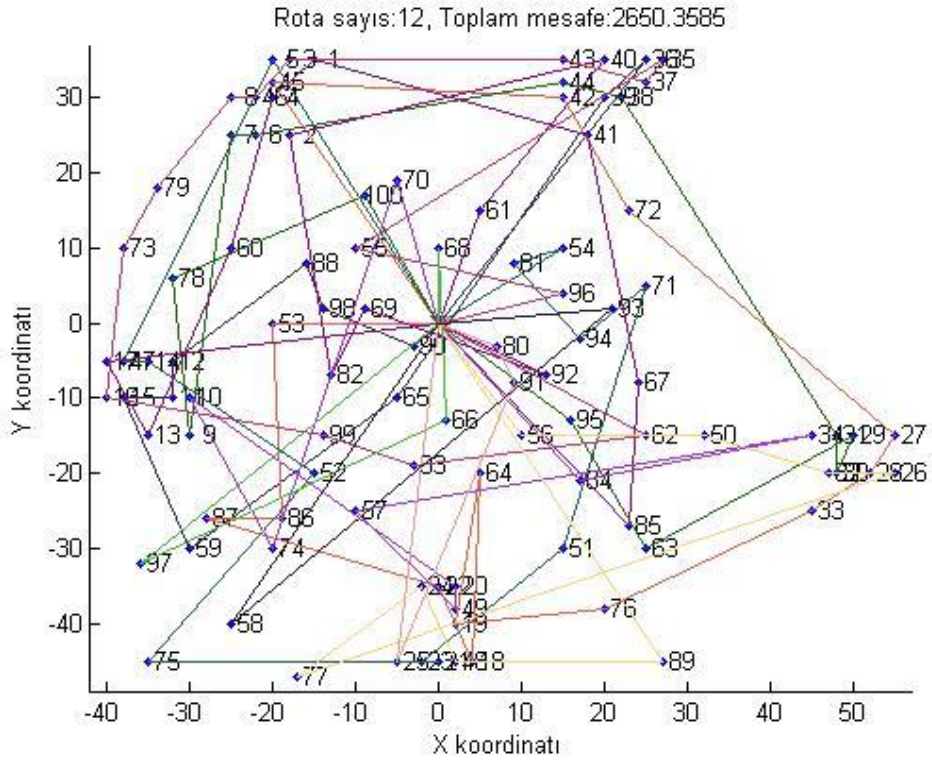
Şekil 4.30: 25 müşterilik R101 problem kümesi için elde edilen alternatiflerden 3 rotalı için en iyi değerli grafik



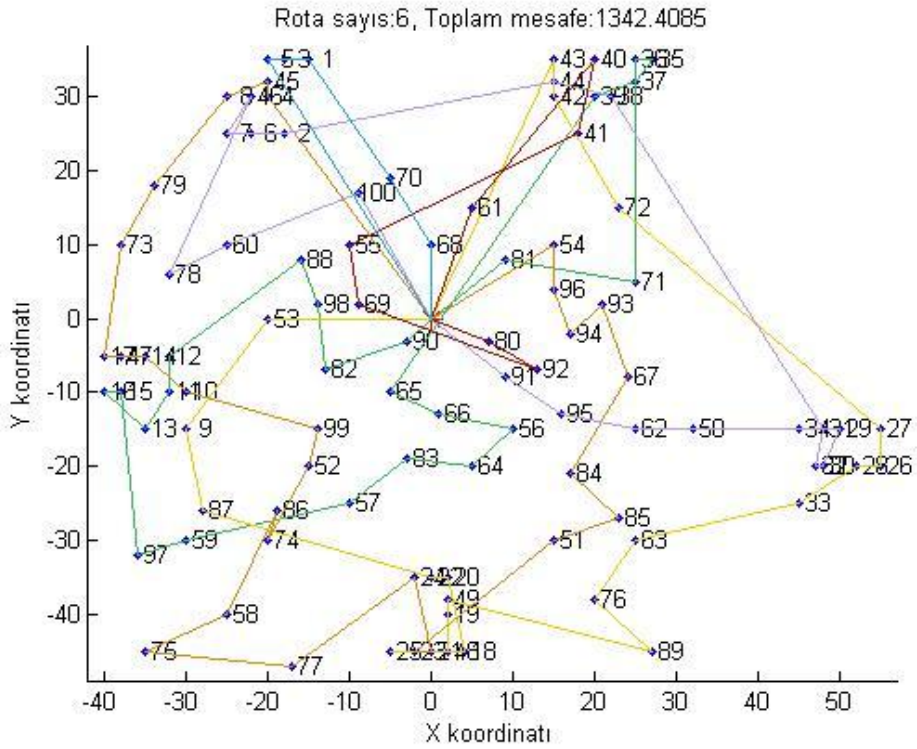
Şekil 4.31: 50 müşterilik RC101 problem kümesi için İES yöntemi ile elde edilen başlangıç çözümü grafiği



Şekil 4.32: 50 müşterilik RC101 problem kümesi için elde edilen alternatiflerden 5 rotalı için en iyi değerli grafik



Şekil 4.33: 100 müşterilik RC201 problem kümesi için İİES yöntemi ile elde edilen başlangıç çözümü grafiği



Şekil 4.34: 100 müşterilik RC201 problem kümesi için elde edilen alternatiflerden 5 rotalı için en iyi değerli grafik

Sunulan tablolar incelendiğinde, ÇAGA'dan elde edilmiş sonuçlara kıyasla C101 problem setinin 25 müşterilik grubu dışında kalan veri kümelerinde, ÇAGDKA yaklaşımı ile çok iyi değerler elde edildiği görülmüştür. Bu kıyaslama tabloları beşinci bölümde gösterilmiştir. Bu bölüm altında C101 problemi için daha iyi sonuç elde etmek için program üzerinde gerçekleştirilen değişim incelenmiştir.

4.1.3. Geliştirilen Yaklaşımın C101 Problem Kümesi İçin İncelenmesi

İİES ve süpürme algoritması başlangıç çözüm yöntemlerine ÇAGDKA yaklaşımının uygulanması sonucunda C101 problem setinin 25 müşterilik grubunda genetik algoritmanın sonuçlarından iyi değerler elde edilememiştir. Bu yüzden 25 müşterilik veri kümesi için program üzerinde döngü sayıları değiştirilerek iyileştirme yapmaya çalışılmasına karar verilmiştir. İlk olarak kullanılan durdurma kriteri (dış döngü) sayısı $s=50$ olarak sabit tutulmuş, iç döngü sayısı olan $l=20$ ise $l=30$ olarak alınmıştır. Bu durumda Tablo 4.59'daki rota sayısı=3, mesafe=194,30 değerlerinden daha iyi sonuçlar elde edilmemiştir. İç ve dış döngü değerleri 10'ar birim arttırılarak denemelere devam edilmiştir. $s=100$, $l=80$ olarak alındığında iyi daha iyi değerler elde edilmiştir. bu değerler için program ilk çalıştırıldığında rota sayısı=3, mesafe=192,03; ikinci kez çalıştırıldığında rota sayısı=3, mesafe=191,59 elde edilmiştir. Genetik algoritmanın verdiği sonuçla kıyaslandığında ÇAGDKA'nın ikinci değeri daha iyidir. Program sekiz kere daha çalıştırılmış ve Tablo 4.24'te tüm sonuçlar gösterilmiştir. Beşinci çalıştırmada Pareto çözüm kümesi iki alternatif çözüm değeri sunmuştur. Dördüncü çalıştırılışında en iyi değer bulunduğuna görülmektedir.

Tablo 4.24: ÇAGDKA yaklaşımının C101/25 problem kümesine $s=100$, $l=80$ değerleri ile uygulanmasının sonuçları

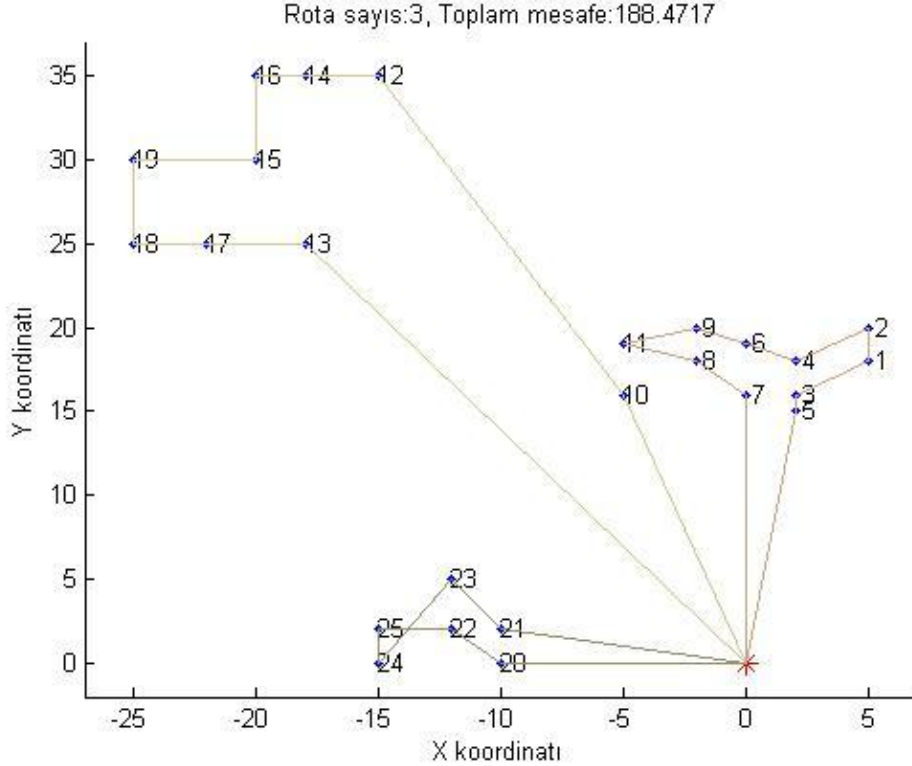
Programın çalıştırılma numarası	Araç/rota sayısı	Mesafe	Ortalama işlem süresi (sn)
1	3	192,03	36,6
2	3	191,59	
3	4	253,12	
4	3	188,47	
5	4	217,75	
	3	235,51	
6	3	253,24	
7	3	195,44	
8	4	230,60	
9	4	229,04	
10	3	230,90	

Program üzerinde döngü sayısı artırıldığında iyi sonuç etme imkanı artmaktadır. Fakat on değer için geçen süreler incelendiğinde ortalama 36,6 sn bir sonucun elde edilmesi için geçen süre olmuştur. Elde edilen çözümün kalitesine göre bu süre kabul edilebilir düzeydedir. Döngü sayısı değiştirildikçe problem çözümü için geçen ortalama süreler ait bazı değerler Tablo 4.25'te belirtilmiştir.

Tablo 4.25: ÇAGDKA yaklaşımının C101/25 problem kümesine uygulanmasında s ve l değerlerinin değiştirilmesiyle değişen çözüme ulaşma süreleri tablosu

Dış döngü sayısı (s)	İç döngü sayısı (l)	Çözüme ulaşılmasında geçen ortalama süre (sn)
50	20	8,2
50	40	10,6
100	20	9,4
100	40	12,08
100	80	36,6

Döngü değerlerinin değiştirilmesi sonucu elde edilen en iyi değere ait görsel Şekil 4.35'te sunulmuştur.



Şekil 4.35: 25 müşterilik C101 problem kümesi için elde edilen en iyi sonucun grafiği.

Süpürme algoritmasından elde edilen başlangıç çözümlerinin kullanılarak programların çalıştırılmasında yüksek dış döngü ve yüksek iç döngü tercih edilmesinin sonuçlar üzerinde çok fazla etkisi olmadığı gözlenmiştir. Bunun sebebi, zaten iyi sonuçlar üretmiş olan süpürme algoritmasının değerlerinin iyileştirilmeye çalışılmasıdır. Fakat iç döngü değeri yüksek tutulduğunda daha kısa sürede kaliteli çözümlere ulaşılmıştır. Kıyaslama tablolarında da (Tablo 4.15 ve Tablo 4.16) zaten daha kısa sürede biraz daha iyi çözümlere ulaşıldığı görülmektedir. Bu yüzden yüksek iç döngü değeri öncelikli olarak tercih edilmektedir. Başlangıç çözümlerinin İİES ile elde edilmesinde ise bu fark daha fazla göze çarpmaktadır. Yüksek iç döngünün kullanıldığı durumlarda daha iyi sonuçlar elde edilmiş olması normaldir. Çünkü GDKA yaklaşımının yerel arama aşamasında değişken komşuluk azalma yapısı uygulanmaktadır ve rassallık kullanmadan iyileştirme yapmaya çalışmaktadır. Çözümün herhangi bir yerel aramaya takılı kalması durumuna karşı iç döngü tamamlandıktan sonra stokastik bir yapıda olan silkeleme aşamasına geçilmektedir.

5. TARTIŞMA VE SONUÇ

Gerçek hayatta karşılaşılan problemler genelde birden fazla amaca sahiptir ve bunlar genellikle birbirleriyle çelişen yapıdadırlar. Bilgisayarların fazla olmadığı yıllarda çok amaçlı problemlerin çözümü üzerine fazla yoğunlaşılmamıştır. Fakat teknolojinin gelişmesiyle bu alandaki çalışmalar da hız kazanmış ve çözümü zor olan kombinatoryal problemler de çok amaçlı olarak ele alınmıştır. Klasik yöntemler gelişen teknolojiye rağmen büyük boyuttaki problemlerin çözümünde çok geç sonuç bulabilmekte veya sonuca ulaşamadığı durumlar da olabilmektedir. Bu yüzden kısa sürede kabul edilebilir iyiye yakın sonuçlar veren metasezgisel yaklaşımlar geliştirilmiştir. Bu yaklaşımlardan birisi de sistematik olarak komşuluk değişimini esas alan değişken komşuluk arama (DKA) metasezgiselidir. Tedarik zinciri içinde karşılaşılan önemli bir problem olan araç rotalama problemleri (ARP) de çözülmesi zor olan problemler sınıfına girmektedir. Tek amaçlı araç rotalama problemlerinde sıklıkla uygulanmış olan DKA metasezgiselinin çok amaçlı araç rotalama problemi (ÇARP) için bir uygulaması ile karşılaşılmamıştır. Fakat çizelgeleme, atama gibi bazı problemlerde uygulamaları olduğu görülmüştür. Bu çalışmaların bazılarında birden fazla amaç fonksiyonları çeşitli katsayılar verilerek tek amaç fonksiyonuna indirgenerek çözülmüştür. İki amaç fonksiyonunun belirlendiği bu tezde ise baskın olmayan çözümlerin, yani iki farklı amaç fonksiyonu için farklı çözüm alternatifleri sunarak karar vericiye tercih hakkı sunan pareto yaklaşımının uygulanması tercih edilmiştir. Pareto yaklaşımını kullanarak Geiger (2004) atölye tipi çizelgeleme problemleri için çok amaçlı değişken komşuluk arama (ÇADKA) yaklaşımını geliştirmiştir. Bu yöntemde çeşitli değişiklikler yapılmış ve ARP'ne uygun hale getirilmiştir. Literatüre daha farklı bir yaklaşım kazandırmak için DKA yaklaşımının yerel arama aşaması içinde, değişken komşuluk azalmanın uygulanması ile geliştirilmiş olan GDKA yaklaşımının seçilmesine karar verilmiştir. Bu yaklaşımın tek amaçlı ARP üzerinde de az sayıda uygulaması mevcuttur ve herhangi bir çok amaçlı problem üzerinde de henüz bir uygulaması mevcut değildir. Bu açıdan ele alındığında ilk olarak bu tezde, çok amaçlı genel değişken komşuluk arama yaklaşımı ARP üzerinde uygulanmıştır.

Çalışmanın giriş bölümünde tezin ana hatları ve literatüre sağlanmak istenen katkılar ele alınmıştır. Ardından ikinci bölümde optimizasyon, yerel arama, komşuluk yapısı, tedarik zinciri yönetimi kavramları ile tedarik zinciri yönetiminde araç rotalama problemi, çok amaçlı programlama ve detaylı olarak çok amaçlı araç rotalama problemleri tanımlanmış, çözüm için geliştirilen yöntemler incelenmiştir.

Üçüncü bölümde DKA yaklaşımı anlatılmış, yöntemin temel algoritması, çalışma prensipleri ve geliştirilen çeşitleri incelenmiştir. Ayrıca bu bölümde DKA yaklaşımının ARP üzerinde uygulamalarına ait çalışmalar ve uygulama aşamaları hakkında bilgi verilmiştir. Çok amaçlı ARP üzerinde DKA yaklaşımının uygulaması bulunmamaktadır. Fakat çözüm yönteminin geliştirilmesinde çok az sayıda bulunan ÇADKA uygulamalarına yer verilmiştir.

Dördüncü bölümde GDKA yaklaşımının genel olarak çok amaçlı araç rotalama problemlerinde uygulanması ele alınarak, sonuçlar sunulmuştur. Uygulamada MATLAB programında hazırlanan kodlar çalıştırılmıştır. Geliştirilen algoritmanın başarısı, Chand ve diğ.'nin (2010) çok amaçlı genetik algoritma (ÇAGA) yaklaşımında elde ettikleri sonuçlar ile karşılaştırılarak incelenmiştir. Tablo 3.1'de minimum araç sayıları ifade edilmişti, uygulama sonucunda da doğal olarak bu değerlerden daha düşük sonuçlar elde edilmemiştir. Tablo 5.1 ve 5.3 arası bu karşılaştırmalara yer verilmiştir. Bu tablolarda Chand ve diğ.'nin (2010) uygulamalarında elde ettikleri sonuçlara kıyasla daha iyi olan sonuçlar koyu renkte gösterilmiştir. Tablo 5.1 ile Tablo 5.2 süpürme algoritmasının başlangıç çözümü olarak seçilmesi durumunda elde edilen sonuçların tüm kıyaslama verilerini içermektedir. ÇAGDKA yaklaşımın uygulanması sonucu elde edilen sonuçların neredeyse tümünün genetik algoritma yaklaşımının uygulanması ile elde edilen sonuçlardan daha iyi olduğu görülmüştür. Yüksek iç döngü değeri kullanıldığında C101 problem kümesinin küçük ölçekli, yani 25 ve 50 müşterilik gruplarında, ÇAGA daha iyi mesafe değeri vermiştir. Ayrıca yüksek dış döngü değeri seçilerek uygulama yapıldığında C201 problem kümesinin 50 ve 100 müşterilik gruplarında da, ÇAGA mesafe açısından daha iyi değerlere sahiptir. Diğer problem kümeleri için elde edilen sonuçlarda ÇAGDKA yaklaşımının her iki amaç fonksiyonunda daha iyi değerler vermesiyle baskın özellikte sonuçlara sahip olduğu görülmektedir.

Tablo 5.1: ÇAGDKA yaklaşımı ile ÇAGA yaklaşımının verdiği en iyi değerlerin kıyaslama tablosu(yüksek dışdöngü)

Problme ait genel veriler			Chand ve diğ.'nin (2010) ÇAGA ile elde ettikleri sonuçlar		ÇAGDKA yaklaşımının uygulanması ile elde edilen sonuçlar	
Problem kümesi	Müşteri sayısı	Araç kapasitesi	Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe
C101	25	200	3	191.81	3	239,03
	50	200	6	429.63	5	510,23
	100	200	12	1467.47	10	1170,74
C201	25	700	2	237.15	1	230,26
	50	700	2	495.45	2	535,47
	100	700	3	773.27	3	1002,60
R101	25	200	8	623.52	2	377,29
	50	200	11	1223.11	4	764,62
	100	200	19	1997.87	8	1317,80
R201	25	1000	2	597.22	1	357,23
	50	1000	2	1069.32	1	645,92
	100	1000	5	1700.59	2	1155,60
RC101	25	200	4	473.50	3	443,87
	50	200	8	951.79	5	805,28
	100	200	15	1986.58	9	1490,10
RC201	25	1000	2	526.86	1	305,71
	50	1000	3	1135.48	1	474,43
	100	1000	5	2027.62	2	1228,20

Tablo 5.2: ÇAGDKA yaklaşımı ile ÇAGA yaklaşımının verdiği en iyi değerlerin kıyaslama tablosu(yüksek iç döngü)

Problme ait genel veriler			Chand ve diğ.'nin (2010) ÇAGA ile elde ettikleri sonuçlar		ÇAGDKA yaklaşımının uygulanması ile elde edilen sonuçlar	
Problem kümesi	Müşteri sayısı	Araç kapasitesi	Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe
C101	25	200	3	191.81	3	237,04
	50	200	6	429.63	5	499,60
	100	200	12	1467.47	10	1149,60
C201	25	700	2	237.15	1	201,81
	50	700	2	495.45	2	464,45
	100	700	3	773.27	3	904,47
R101	25	200	8	623.52	2	367,50
	50	200	11	1223.11	4	704,45
	100	200	19	1997.87	8	1321,00
R201	25	1000	2	597.22	1	329,97
	50	1000	2	1069.32	1	560,55
	100	1000	5	1700.59	2	991,40
RC101	25	200	4	473.50	3	447,62
	50	200	8	951.79	5	782,59
	100	200	15	1986.58	9	1486,30
RC201	25	1000	2	526.86	1	294,72
	50	1000	3	1135.48	1	468,83
	100	1000	5	2027.62	2	1032,6

Tablo 5.3: ÇAGDKA yaklaşımı ile elde edilne sonuçların tablosu (İİES başlangıç çözüm yöntemi ve yüksek iç döngü değeri ile)

Problem kümesi	Müşteri sayısı	Araç kapasitesi	ÇAGDKA yaklaşımı ile elde edilen alternatifler		ÇAGDKA yaklaşımı alternatiflerinin en iyi değerleri (kıyaslama için)		Chand ve diğ.'nin (2010) ÇAGA ile elde ettikleri sonuçlar	
			Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe
C101	25	200	3	194,30	3	194,30	3	191.81
			4	221,40				
	50	200	5	486,00	6	413,81	6	429.63
			6	413,81				
	100	200	10	1325,40	10	1325,40	12	1467.47
			11	1059,00				
12			1043,91					
C201	25	700	2	234,31	2	234,31	2	237.15
			3	478,00				
	50	700	4	444,00	4	444,00	2	495.45
			5	473,00				
			5	688,80				
	100	700	6	843,11	5	688,80	3	773.27
8			1101,40					
8			1101,40					
R101	25	200	3	385,92	3	385,92	8	623.52
			4	411,84				
			5	423,03				
	50	200	5	652,28	5	652,28	11	1223.11
			6	649,83				
			7	674,40				
	100	200	9	1,21	9	1210,00	19	1997.87
			10	1,06				
			11	1,10				
			12	1,15				
			13	1,20				
			14	1,16				

Tablo 5.3: devamı

Problem kümesi	Müşteri sayısı	Araç kapasitesi	GA ile kıyaslama için en iyi alternatif değerleri		GA ile kıyaslama için kullanılan en iyi değerler		Chand ve diğ.'nin (2010) ÇAGA ile elde ettikleri sonuçlar	
			Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe
R201	25	1000	3	482,78	3	482,78	2	597.22
			4	392,12	4	392,12		
	50	1000	3	553,81	3	553,81	2	1069.32
			4	592,55				
			5	667,13				
			7	650,91				
	100	1000	4	982,22	4	982,22	5	1700.59
			5	936,62				
			6	1213,40				
8			1512,30					
9			1478,50					
RC101	25	200	3	312,28	3	312,28	4	473.50
			4	373,45				
			5	447,07				
	50	200	5	536,15	5	536,15	8	951.79
			6	628,74				
			7	768,80				
			8	794,64				
	100	200	11	1335,80	11	1335,80	15	1986.58
			12	1377,40				
			13	1463,70				
			14	1605,20				
			15	1569,00				
			16	1490,50				
17			1629,70					

Tablo 5.3: devamı

Problem kümesi	Müşteri sayısı	Araç kapasitesi	GA ile kıyaslama için en iyi alternatif değerleri		GA ile kıyaslama için kullanılan en iyi değerler		Chand ve diğ.'nin (2010) ÇAGA ile elde ettikleri sonuçlar	
			Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe	Araç/rota sayısı	Katedilen mesafe
RC201	25	1000	2	333,82	2	333,82	2	526.86
			3	318,01	3	318,01		
	50	1000	4	998,93	4	998,93	3	1135.48
			5	1248,60				
	100	1000	5	1429,60	5	1429,60	5	2027.62
			6	1342,40				
			8	1468,20				
			9	1924,30				

Tablo 5.3'te başlangıç yöntemi olarak İİES yöntemi uygulandığında ÇAGDKA sonuçları ile Chand ve diğ.'nin (2010) ÇAGA yaklaşımının sonuçları sunulmuştur. Farklı başlangıç çözümleri uygulanmasının sebebi algoritmanın başarısını ölçmek içindir. Ayrıca programdaki döngülere ait sayılarda değişiklikler yapılmış ve elde edilen sonuçlar incelenmiştir. İlk başlangıç çözümü olarak süpürme algoritması seçilmiştir. MATLAB programının microsoft office excel dosyalarından veri okuyabilme özelliğinden faydalanarak excel ortamında hazırlanan makroların çalıştırılmasıyla elde edilen başlangıç rota çözümleri kullanılmıştır. Süpürme algoritmasında elde edilen sonuçların kullanılmasıyla çalıştırılan programda Chand ve diğ.'nin (2010) elde ettikleri sonuçlara kıyasla, daha başarılı sonuçlar elde edilmiştir (Tablo 5.1 ve 5.2). Süpürme algoritması zaten en düşük araç sayısını vererek başlangıç çözümlerini oluşturduğu için, ÇAGDKA yaklaşımının uygulanması sonucunda başlangıç çözümüyle aynı araç sayısında daha düşük mesafeli sonuçlar elde edilmiştir. Geliştirilen pareto yaklaşımının sunduğu alternatif sonuçlara da ulaşabilmek için başka başlangıç çözümü yöntemi de uygulanmış ve program çalıştırılmıştır. Rota sayısını yüksek veren, iyi sonuçlara sahip olmayan bu başlangıç çözüm değerleri ile program tekrar çalıştırılmıştır. Kötü başlangıç çözümlerinin kullanıldığı bu durumda da, yine Chand ve diğ.'nin (2010) elde ettikleri sonuçlara kıyasla daha iyi sonuçlar elde edilmiştir.

Solomon veri kümelerinin 25, 50 ve 100'er müşterilik grupları üzerinde yapılan uygulamalar sonucunda amaç fonksiyonlarına göre ÇAGDKA yaklaşımının en iyi değerleri ve ortalama değerleri bir önceki bölümde tablolar halinde sunulmuştur. Bu bölümdeki kıyaslama tablolarında da görüldüğü gibi süpürme algoritması ile başlangıç çözümleri elde edilerek ÇAGDKA yöntemi uygulandığında, C101 problem kümesinin 25 ve 50 müşterilik grupları dışında kalan tüm veri setlerinde çok amaçlı genetik algortima (ÇAGA) yaklaşımı ile elde edilen sonuçlara göre daha iyi değerler elde edildiği görülmüştür. ÇAGDKA yaklaşımına ait programı üzerinde yüksek iç döngü değeri veya yüksek dış döngü değeri kullanılarak oluşturulan iki farklı çalışma sürecinde, süpürme algoritmasının başlangıç çözümlerinin kullanılması durumunda elde edilen sonuçlar birbirine oldukça yakındır. Bunun sebebi, süpürme algoritmasında olabilecek en düşük araç (rota) sayısının kullanılması ve ÇAGDKA yaklaşımının sadece mesafe değerlerini iyileştirmesidir. ÇAGDKA yaklaşımına alternatif sonuçları sunan Pareto kümesi dahil edilmiş olmasına rağmen, süpürme algoritmasının verdiği sonuçlar

ile ÇAGDKA uygulandığında alternatif çözümler elde edilememiştir. Pareto uyarlamasının çalışmasını görebilmek ve karar vericiye alternatif çözümler sağlanabilmesi amacıyla ikinci bir alternatif başlangıç yöntemi olarak ileri itmeli ekleme sezgisel yöntemi kullanılmıştır. Yüksek sayıda rotalar veren bu başlangıç çözüm değerlerinin ÇAGDKA yaklaşımında kullanılmasıyla elde edilen alternatif çözümler ve pareto kümelerine ait örnekler bir önceki bölümde ele alınmıştır. Uygulama sonucu elde edilen alternatifler incelendiğinde ve en iyi değerler ele alındığında yine ÇAGA uygulamasının sonucunda elde edilen değerlerden daha iyi sonuçlar bulunmuştur (Tablo 5.3). Bu sonuçlar içinde sadece C101 problem kümesine ait 25 müşterilik veri grubunda ÇAGA yaklaşımı baskın bir çözüm olmuştur. Bu grup dışında kalan C101(50 ve 100 müşteri için), C201(25müşteri için), R101(25, 50 ve 100 müşteri için), R201(25, 50 ve 100 müşteri için), RC101(25, 50 ve 100 müşteri için), R201(25 ve 100 müşteri için) setlerinin hepsinde ÇAGDKA yaklaşımı ÇAGA'dan daha iyi değerler vermiştir. Yani baskın olan çözümler elde edilmiştir. C201(50 ve 100 müşteri için) ve R201(50 müşteri için) elde edilen sonuçlarda bir amaç fonksiyonu ÇAGA'nın verdiği sonuçtan daha iyi bulunmuştur. Bu durum kabul edilebilir baskın olmayan çözümler olduklarını göstermektedir. Her iki başlangıç yönteminin verdiği sonuçlara ÇAGDKA yaklaşımı uygulanmasıyla sadece C101 problem kümesinin 25 müşterili veri grubunda ÇAGDKA iyi sonuç elde edememiştir. Fakat ikinci başlangıç yöntemi kullanılarak elde edilen sonuçlar incelendiğinde ilk amaç fonksiyonunun üç araç, ikinci amaç fonksiyonunun 194,3 birim mesafe değeri verdiği, ÇAGA'da ise üç araç ile 191,81 mesafe sonuçlarının bulunduğu dikkat edilmiştir. ÇAGDKA için hazırlanan programda döngü sayılarının arttırılması ile çözümün iyileşeceği düşünülerek denemeler yapılmış ve bu problem için de Tablo 4.24'te gösterildiği gibi daha iyi sonuçlara ulaşılmıştır (Tablo 4.24 bir önceki bölümde sunulmuştur).

Programın çalıştırılmasında işlem sürelerine ait tablolar incelendiğinde yüksek iç döngü değeri kullanıldığında daha kısa sürede daha iyi sonuçlar elde etme durumu görülmüştür. Ayrıca verilen sürelerin tümü incelendiğinde elde edilen çok iyi sonuçlara göre çok kısa süreler olması DKA yaklaşımının tanımında ifade edilen basit ve iyi sonuçlar verme özelliğini göstermektedir.

GDKA yaklaşımının tek amaçlı araç rotalama problemleri üzerinde başarılı bir algoritma olduğu bilinmekte iken bu çalışma sonucunda çok amaçlı araç rotalama problemleri üzerinde de başarılı sonuçlarına ulaşılmıştır. Çalışma ARP'nin diğer çeşitlerinin çok amaçlı olarak ele alınması şeklinde devam ettirilecektir. Örneğin zaman heterojen araçlı ARP için amaç fonksiyonları belirlenerek ÇAGDKA uygulaması yapılacaktır. Günümüzde çevresel amaçlar rotalama problemlerinde de göz önüne alındığı için, amaç fonksiyonları değiştirilerek de çalışma devam ettirilecektir. Genel değişken komşuluk arama algoritmasının temel alınması yerine diğer DKA yaklaşımı çeşitleri ile de incelemelerin yapılması daha sonraki çalışmalarının konusunu oluşturmaktadır. Ayrıca yöntemi sabit tutarak diğer çok amaçlı problemlere uygulama yapma imkanı da mevcuttur. ÇAGDKA yaklaşımının iyi sonuçlar vermesi bu çalışmaların gerçekleştirilmesi için temel olmaktadır.

KAYNAKLAR

Adibi, M. A., Zandieh, M., Amiri, M., 2010, Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems With Applications*, Vol.37(1), 282-287.

Aldous, M., Wilson, R.J., 2003, *Graphs and applications: an introductory approach*. Springer: Printed in Great Britain.

Alp, A., Çerçioğlu, H., Tokaylı, M.A., Dengiz, B., 2001, Stokastik montaj hattı dengeleme: Bir tavlama benzetimi algoritması. *Endüstri Mühendisliği Dergisi*, Cilt:12, Sayı:3-4, 32-51, Makina Mühendisleri Odası.

Alpay, Ş., 2008, Tam zamanlı montaj hatlarında çok amaçlı karışık model sıralama için rota birleştirilmiş ağgözlü rassallaştırılmış uyarlamalı arama yordamı. *Eskişehir Osmangazi Üniversitesi Müh.Mim.Fak.Dergisi*, Cilt:11, Sayı:1.

Anderson, D., McNeill, 1992, G. Artificial neural networks technology. Rome Laboratory, *A DACS State of the Art Report*, ELIN: A011, New York.

Atlas, M., 2008, Çok amaçlı programlama çözüm tekniklerinin sınıflandırılması. *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, Cilt:8, Sayı:1, 47-68.

Attila, K.A., 2005, Traveling salesman problem for dynamic graphs. *Computational Intelligence Reports*, University of Cluj-Napoca. <http://www.cir.cs.ubbcluj.ro/TechnicalReports/No052005.pdf> [Erişim: 20.05.2010]

Bağış, A., 2002, Tabu araştırma algoritmasına dayalı sistem kimliklendirme işleminde kontrol parametrelerinin çözüme etkisi. *Elektrik-Elektronik-Bilgisayar Müh. Sempozyumu (ELECO 2002)*, 317-321.

Balakrishnan, R., Ranganathan, K., 1999, *A textbook of graph theory*. Springer:New York.

Balakrishnan, V.K., 1997, *Schaum's outline of theory and problems of graph theory*. McGraw-Hill Companies: USA.

Bayzan, Ş., Çetin, M., Uğur, A., 2002, Araç rotalama probleminde araç rotalarının tespitinde en kısa yol yaklaşımı: Denizli örneği. <http://ab.org.tr/ab06/bildiri/152.pdf> [Erişim: 25.12.2008]

Belhaiza, S.J., 2010, Hybrid variable neighborhood-tabu search algorithm for the site dependent vehicle routing problem with time windows. <http://faculty.kfupm.edu.sa/MATH/slimb/SDVRP/LongAbstract-Roadef2011.pdf> [Erişim: 30.07.2011]

Bowersox, D.J., Closs, D.J., Cooper, M.B., 2002, *Supply chain logistics management*. New York: McGrawHill.

Braysy, O., 2000, Local search and variable neighborhood search algorithms for the vehicle routing with time windows. Doktora tezi. Universitas Wasaenis, Vaasa, Finlandiya.

Bräysy, O., 2003, A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing* ©2002INFORMS, Vol.15, No.4, 347-368.

Burke, E.K., Curtois, T., Post, G., Qu, R., Veltman, B., 2008, A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, Vol.188, Issue 2, 330-341.

Caccetta, L, Hill, S.P., 2001, Branch and cut methods for network optimization. *Mathematical And Computer Modeling*, 33, 517-532.

Carson, T., 2001, Empirical and analytic approaches to understanding local search heuristics. Doktora Tezi. Bilgisayar Bilimi. California Üniversitesi, San Diego.

Chand, P., Mishra, B.S.P., Dehuri, S., 2010, A multi objective genetic algorithm for solving vehicle routing problem. *International Journal of Information Technology and Knowledge Management*, Vol.2, No.2, 503-506.

Chandra, C., Tumanyan, A., 2003, Supply chain system taxonomy: Development and application. *Conference Proceedings. 12th Annual Industrial Engineering Research Conference (IERC 2003)*, Portland, Oregon.

Chen, P., Huang, H-K., Dong, X-Y., 2010, Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, Vol.37, 1620–1627.

Christofides, N., Mingozzi, A., Toth, P., 1981, Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, Vol.20, 255-282.

Christopher, M., 1998, *Logistics and supply chain management*. Prentice Hall, Second Edition.

Civalek, Ö., Ülker, M., 2004, Dikdörtgen plakların doğrusal olmayan analizinde yapay sinir ağı yaklaşımı. *İMO Teknik Dergi*, Yazı 213, 3171-3190.

Clark, J., Holton, D.A., 1991, *A first look at graph theory*. World Scientific Publishing: Singapore.

Clarke, G., Wright, J.W., 1964, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, Vol.12, No.4, 568-581.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F., 2002, A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, Vol.53, 512-522.

Crainic, T.G., Gendreau, M., Hansen, P., Mladenovic, N., Parallel VNS for the p-median problem, Sunum. http://www.lsi.upc.edu/~dbaneres/Resources/PVSp_median.ppt [Eriřim: 27.12.2008]

Crainic, T.G., Gendreau, M., Hansen, P., Mladenovic, N., 2004, Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, Vol.10, No.3, 293-314.

Crispim, J., Brandao, J., 2001, Reactive tabu search and variable neighbourhood descent applied to the vehicle routing problem with backhauls. *MIC'2001-4th Metaheuristics International Conference*, 631-636.

Çolakođlu, E., 2010, Graf teorisi (graph theory) örüntü tanıma <http://akademik.maltepe.edu.tr/~ttbilgin/BIL518/presentations/EsraCOLAKOGLU/Sunum1/Esra%20%C7olako%F0lu.pdf> [Eriřim: 30.07.2011]

Çölkesen, R., 2002, *Bilgisayar programlama ve yazılım mühendisliğinde: Veri yapıları ve algoritmalar*. Papatya Yayıncılık: İstanbul.

Dabia, S., Woensel, T.-V., Kok, A.G.-D., 2010, *A dynamic programming approach to multi-objective time-dependent capacitated single vehicle routing problems with time window*. Beta Working Paper series 313. ISBN 978-90-386-2240-8.

Dreo, J., Petrowski, A., Siarry, P., Taillard., 2006, *Metaheuristics for hard optimization: Methods and case studies*. E. Translator : Amitava Chatterj, Springer: Berlin. (Originally published in French by Eyrolles, Paris, France (2003) under the title: "M'etaheuristiques pour l'optimisation difficile")

Eksioglu, B., Vural, A.V., Reisman, A., 2009, Survey: The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, Vol.57, 1472–1483.

Erengüç, ř.S., Simpson, N.C., Vakharia, A.J., 1999, Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, Vol.115, Issue 2, 219-236.

Evans, G., 1984, An overview of techniques for solving multiobjective mathematical programs. *Management Science*, 30(11), 1268-1282.

Farahani, R. Zanjirani., S, Maryam. S., Asgari, N., 2010, Multiple criteria facility location problems: A survey. *Applied Mathematical Modelling*, Vol.34, Issue: 7, Publisher: Elsevier, 1689-1709

Ferreira, F., 2002, Combinatorial optimization by gene expression programming: Inversion revisited. (In J. M. Santos and A. Zapico, eds.) *Proceedings of the Argentine Symposium on Artificial Intelligence*, Santa Fe, Argentina, 160-174.

Fleszar, K., Hindi, K.S., 2004, Solving the resource-constrained project scheduling problem by a variable neighbourhood search, *European Journal of Operational Research*, Vol.155, 402–413.

Fleszar, K., Hindi, K.S., 2008, An effective VNS for the capacitated p -median problem. *European Journal of Operational Research*, Vol.191, Issue 3, 612-622.

Fleszar, K., Osman, I H., Hindi, K.S., 2009, A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, Vol.195, Issue 3, 803-809.

Floudas, C.A., Pardalos, P.M. (Eds.), 2009, *Encyclopedia of optimization*. 2. Ed., Springer Science+Buisness Media, LLC:New York. XXXIV, 4626.

Fonlupt, C., Robilliard, D., Preux, P., 1997, Fitness landscape and the behavior of heuristics. *Evolution Articielle (EA'97)*.

Ganeshan, R., Harrison, T.P., 1995, An introduction to supply chain management. In: supply chain management. Penn State University. Version 1.0. http://lcm.csa.iisc.ernet.in/scm/supply_chain_intro.html [Eriřim tarihi: 16 Ekim 2011]

Geiger, M.J., 2001, Genetic algoritms for multi-objective vehicle routing. *Proceedings of the Metaheuristics International Conference MIC'2001*, Porto, Portugal, 349–353.

Geiger, M.J., 2004, Randomised variable neighbourhood search for multi objective optimisation. In *Proceedings of the 4th EU/ME Workshop: Design and Evaluation of Advanced Hybrid Meta-Heuristics*, Nottingham, United Kingdom, 34-42.

Gencer, C., Yařa, Ö., 2007, Ulařtırma komutanlıęı ring seferlerinin eř zamanlı daęıtım toplama karar destek sistemi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Cilt: 22, No.3, 437-449.

Geyik, F., Cedimoęlu, İ.H., 2010, Atölye tipi çizelgelemede komşuluk yapılarının tabu arama teknięi ile karşılařtırılması. *Journal of Polytechnic Faculty of Technical Education Gazi University*, Vol.4, No.1, 95-103. <http://www.politeknik.gazi.edu.tr/index-s.php?q=4&s=12>

Ghoseiri, K., Ghannadpour, S.F., 2010, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing*, Vol.10, Issue 4, 1096-1107.

Glover, F., Kochenberger, G.A., 2003, *Handbook of metaheuristics*. Dordrecht: Kluwer Academic Publishers.

Gözüdeli, Y., Akcayol, M.A., 2007, Genetik algoritma ile web sayfası düzeninin gerçek zamanlı optimizasyonu. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Cilt:22, No.2, 431-439.

Gözüpek, D., Genç, G., 2009, Hücresel aęlarda kanal planlama problemine tabu araması yaklařımı. *Akademik Biliřim'09 - XI. Akademik Biliřim Konferansı Bildirileri*. Harran Üniversitesi, řanlıurfa.

- Gutin, G., Punnen, A.P. (edited by), 2002, *The traveling salesman problem and its variations*. Kluwer Academic Publishers New York, Boston.
- Hansen, P., Mladenovic, N., Perez-Brito, D., 2001, Variable neighborhood decomposition search. *Journal of Heuristics*, Vol.7, 335–350.
- Hansen, P., Mladenovic, N., 2001, Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130, 449-467.
- Hansen, P., Mladenovic, N., 2003, A tutorial on variable neighborhood search, *GERAD*.
- Hansen, P., Mladenovic, N., Urosevic, D., 2006, Variable neighborhood search and local branching. *Computers & Operations Research*, Vol.33, Issue 10, 3034-3045.
- Hemmelmayr, V.C., Doerner, K.F., Hartl, R.F., 2009, A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, Vol.195, 791–802.
- Hansen, P., Mladenovic, N., Perez Moreno, J.A., 2010, Variable neighborhood search: Methods and applications. *Ann Oper Res*, Vol.175, 367–407.
- Hwang, C.-P., 1996, *Global and local search heuristics for the symmetric traveling salesman problems*. Doktora Tezi. Mississippi Üniversitesi.
- Imran, A., Salhi, S., Wassan, N.A., 2009, A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, Vol.197, 509–518.
- Ishibuchi, H., Yoshida, T., Murata, T., 2003, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2), 204-223.
- Johnson, D.S., McGeoch, L.A., 1995, The traveling salesman problem: A case study in local optimization. (E.H.L. Aarts ve J.K. Lenstra tarafından düzenlenen *Local Search in Combinatorial Optimization* kitabında bir bölüm)
- Jozefowicz, N., Semet, F., Talbi, E.-G., 2008a, *From single-objective to multi-objective vehicle routing problems: Motivations, case studies, and method*. (in: The vehicle routing problem: Latest advances and new challenges-operations research/computer science interfaces series), Vol.43, Part II, 445-471.
- Jozefowicz, N., Semet, F., Talbi, E.-G., 2008b, Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189 (2), 293-309.
- Jozefowicz, N., Semet, F., Talbi, E.-G., 2009, An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research*, Vol.195, Issue 3, 761-769.
- Karaboğa, D., 2011, *Yapay zeka optimizasyon algoritmaları*. 2. Basım. Nobel Yayın Dağıtım: Ankara. XII

Keçek, G., 2005, Bir dişli fabrikasında tamsayı hedef programlama uygulama denemesi. *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, Sayı:13.

Korpela, J., Lehmusvaara, A., 1999, A customer oriented approach to warehouse network evaluation and design. *International Journal of Production Economics*, Vol.59, Issue 1-3, 135-146.

Korpela, J., Lehmusvaara, A., Tuominen, M., 2001, Customer service based design of the supply chain. *International Journal of Production Economics*, Vol.69, 2, 193-204.

Küçükdeniz, T., 2004, Tedarik Zinciri Yönetiminde Bulanık Modelleme. Yüksek Lisans Tezi. İstanbul Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü. İstanbul.

Kytöjoki, J., Nuortio, T., Bräysy, O., Gendreau, M., 2007, An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, Vol.34, 2743–2757.

Lambert, D.M., Cooper, M.C., Pagh, J.D., 1998, Supply chain management: Implementation issues and research opportunities. *The International Journal of Logistics Management*, 9(2), 1-12.

Laporte, G., 2006, A short history of the traveling salesman problem. HEC Montreal, Canada. <http://neumann.hec.ca/chairedistributique/common/laporte-short.pdf> [Erişim tarihi: 6 Eylül 2011]

Laporte, G., 2009, Fifty years of vehicle routing. *Transportation Science*. Vol.43, No.4, 408-416. Publisher:INFORMS

Larsen, J., 1999, Parallelization of the vehicle routing problem with time windows. Doktora Tezi. LYNGBY.NO. 62. IMM - DTU Technical University of Denmark

Lei, H., Laporte, G., Guo, B., 2011, A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *Sociedad de Estadística e Investigación Operativa*.

Li, J.-Q., Pan, Q.-K., Xie, S.-X., 2010a, A hybrid variable neighborhood search algorithm for solving multi-objective flexible job shop problems. *Computer Science and Information Systems/ComSIS*, Vol.7(4), 907-930.

Li, J.-Q., Pan, Q.-K., Liang, Y.-C., 2010b, An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, Vol.59(4), 647-662.

Liang, Y.-C., Lo, M.-H., 2010, Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm. *Journal of Heuristics*, Vol.16(135), 511-535.

Liang, Y.-C., Chen, A.H.-L., Tien, C.-Y., 2009a, Variable neighborhood search for multi-objective parallel machine scheduling problems. *Proceeding of the 8th International Conference on Information and Management Sciences (IMS 2009)*, Kunming-Çin, 519-522.

Liang, Y.-C., Chuang, C.-Y., Lo, M.-H., 2009b, Variable neighborhood search for multi-objective resource allocation problems. *Proceedings of the 10th Asia Pacific Industrial Engineering & Management System Conference (APIEMS 2009)*, Kitakyushu, Japan, 2175-2179.

Liberti, L., Maculan, N., 2006, *Global optimization: From theory to implementation*. Springer: Berlin.

Liu, H., Abraham, A., Grosan, C., 2007, A novel variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. *IEEE International Conference on Digital Information Management, 1(Lyon, France, IEEE Press, USA)*, 138-145.

Liu, H., Abraham, A., Choi, O., Moon, S.H., 2006, Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. *In Proceedings of SEAL'200*, 197-204.

Mastrolilli M., Gambardella L.M., 2002, Effective neighborhood functions for the flexible job shop problem. *Journal of Scheduling*, 3(1):3-20.

Marinakis, Y., Migdalas, A., 2007, Annotated bibliography in vehicle routing. *Operational Research. An International Journal*, Vol.7, No.1, 27-46.

Mladenovic, N., Hansen, P., 1997, Variable neighborhood search. *Computers & Operations Research*, Vol.24, Issue 11, 1097-1100.

Nabiyev, V.V., 2010, *Yapay zeka - Problemler yöntemler algoritmalar*, 3. Baskı. Seçkin Yayınevi, Ankara.

Nilsson, C., 2003, Heuristics for the traveling salesman problem. Tech. Report, Linköping University, Sweden, 2003.

http://www.ida.liu.se/~TDDDB19/reports_2003/htsp.pdf [Erişim: 30.07.2011]

O'Brien, R., 2008, Ant algorithm hyperheuristic approaches for scheduling problems. MPhil thesis, University of Nottingham.

Ombuki, B., Ross, B.J., Hanshar, F., 2006, Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, Vol.24(1), 17-30.

Osman, I.H., Kelly, J.P., 1996, *Meta-heuristics: theory & applications*. Kluwer Academic Publishers: London.

Osman, I.H., Laporte, G., 1996, Metaheuristics: A bibliography. *Annals of Operations Research*, Vol.63, 513-623.

Özaslan, B., Saka, B., 2004, GSM'de frekans atama problemleri ve çözüm algoritmaları. *URSI-Türkiye'2004 İkinci Ulusal Kongresi*
http://www.ursi.org.tr/2004_kongre/BU92.PDF [Erişim: 10.06.2011]

Özdemir, A., Gümüšođlu, Ő., 2006, Dinamik amaç programlama modelinin ürünleri makinelere atama problemine uygulanması. *Review of Social, Economic & Business Studies*, Vol.7/8, 359-383

Pacheco, J., Mart, R., 2006, Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society*, Vol.57, No.1, 29-37.

Paraskevopoulos, D.C., Repoussis, P.P., Tarantilis, C.D., Ioannou, G., Prastacos, G.P. (t.y.). Reactive variable neighborhood tabu search for heterogeneous fleet vehicle routing and scheduling problems. <http://leeds-faculty.colorado.edu/glover/TS%20-%20vignettes%20-%20ReVN%20TS%20Tarantilis%20et%20al.doc> [Eriřim: 20.05.2010]

Paraskevopoulos, D.C., Repoussis, P.P., Tarantilis, C.D., Ioannou, G., Prastacos, G.P., 2008, A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal Heuristics*. Vol.14, 425–455.

Perez, J.A.M., Mladenovic, N., Batista, B.M., Amo, I.J.G., 2006, *Variable neighborhood search*. Chapter in: *Metaheuristic procedures for training neural networks*. Springer: US.

Pirkwieser, S., Raidl, G.R. Multiple variable neighborhood search enriched with ILP techniques for the periodic vehicle routing problem with time windows. <https://www.ads.tuwien.ac.at/publications/bib/pdf/pirkwieser-09b.pdf> [Eriřim:1.07.2011]

Pirkwieser, S., Raidl, G.R., 2008, A variable neighborhood search for the periodic vehicle routing problem with time Windows. *Institute of Computer Graphics and Algorithm. EU/MEeting 2008*, Troyes, France.

Pirkwieser, S., Raidl, G.R., 2009, Boosting a variable neighborhood search for the periodic vehicle routing problem with time windows by ILP techniques. *MIC 2009: The VIII Metaheuristics International Conference*. Hamburg, Germany, July 13-16, 2009.

Pisinger, D., Ropke, S., 2007, A general heuristic for vehicle routing problems, *Computers & Operations Research*, Cilt:34, Sayı:8, 2403-2435.

Polacek, M., Benkner, S., Doerner, K.F., Hartl, R.F., 2008, A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. *BuR—BusinessResearch Official Open Access Journal of VHB*, Vol.1, Issue 2, 207-218.

Polacek, M., Hartl, R.F., Doerner, K.F., Reimann, M., 2004, A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, Vol.10, 613–627.

Porter, M., 1987, *From competitive advantage to corporate strategy*. Harvard Business Review

Rezgui, J., Hafid, A., Ali, R.B., Gendreau, M., 2010, A variable neighborhood search method for multi-objective channel assignment problem in multi-radio WMNs. *35th Annual IEEE Conference on Local Computer Networks LCN*, 64-71.

- Ribas, S., 2011, Um algoritmo híbrido para o problema de roteamento de veículos com janelas de tempo. Phd, Fluminense Federal University. Rio de Janeiro, Brezilya
- Roshanaei, V., Naderia, B., Jolaib, F., Khalilic, M., 2009, A variable neighborhood search for job shop scheduling with set-up times to minimize makespan. *Future Generation Computer Systems*, Vol.25, 654-661.
- Rousseau, L-M., Gendreau, M., Pesant, G., 2002, Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, Vol.8, 43-58.
- Sağ, T., Çunkaş, M., 2009, Çok amaçlı genetik algoritmalar için bir çevrimdışı performans değerlendirmesi. 5. *Uluslararası İleri Teknolojiler Sempozyumu (IATS'09)*, Karabük.
- Salvendy, G., 2001, *Handbook of industrial engineering: Technology and operations management*. A Wiley-Interscience Publication: Canada.
- Schneider, H., Reinholz, A., 2008, Integrating variable neighborhood search into a hybrid evolutionary strategy for the open vehicle routing problem. *EU/MEeting 2008 - Troyes, France*.
- Shor, P.W., 1996, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J.Sci.Statist.Comput.* Vol.41, No.2, 303-332.
- Smith, K.A., Gupta, J.N.D., 2002, *Neural networks in business: Techniques and applications*. Idea Group Publishing: USA.
- Sule, D.R., 2001, *Logistics of facility location and allocation*, Marcel Dekker Inc, New York.
- Şahin, R., Türkbey, O., 2010, Çok amaçlı tesis yerleşim problemi için yeni bir melez sezgisel algoritma. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Cilt:25, No:1, 119-130.
- Şevkli, M., Aydın, M.E., 2007, Parallel variable neighbourhood search algorithms for job shop scheduling problems. *IMA Journal of Management Mathematics*, Vol.18, No.2, 117-133.
- Taha, H.A., 2000, *Yöneylem araştırması*. 6. basımdan çeviri, Çevirenler ve Uyarlayanlar: Ş. Alp Baray, Şakir Esnaf, Literatür Yayıncılık: İstanbul.
- Talbi, E.-G., 2009, *Metaheuristics - From design to implementation*. John Wiley & Sons Inc.: United States of America.
- Tan, K.-C., 2001a, A framework of supply chain management literature. *European Journal of Purchasing & Supply Management*, Vol.7, 39-48.
- Tan, K.C., Lee, L.H., Ou, K., 2001b, Artificial intelligence heuristics in solving vehicle routing problems with time window constraints . *Engineering Applications of Artificial Intelligence*, Vol.14, 825-837.

Tan, K.C., Chew, Y.H., Lee, L.H., 2006, A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, Vol.172, 855–885.

Tavakkoli-Moghaddam, R., Azarkish, M., Sadeghnejad-Barkousaraie, A., 2011, A new hybrid multi-objective pareto archive PSO algorithm for a bi-objective.job shop scheduling problem. *Expert Systems with Applications*, Vol.38, 10812–10821.

Thomas, D.J., Griffin, P.M., 1996, Coordinated supply chain management. *European Journal of Operational Research*, Vol.94, 1-15.

Toth, P., Vigo, D., 2002, *The vehicle routing problem*. SIAM: Philadelphia.

Wei, L., Zhuo, F., 2010, A variable neighborhood tabu search algorithm for the heterogeneous fleet vehicle routing problem with time windows. *Logistics Engineering and Intelligent Transportation Systems (LEITS)International Conference*.

Xu, Y., Qu, R., 2010, Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighborhoods. *Journal of Operational Research Society*, Vol.62, 313-325.

Yavuz, Y., Tosun, S., 2008, Foton haritalama algoritmasının evrensel aydınlatma amaçlı paralelleştirilmesi. *ELECO'2008 Elektrik - Elektronik - Bilgisayar Mühendisliği*.

Yumurtacı, H., Küçükdeniz, T., Esnaf, Ş., 2008, Doğrusal programlama ile araç rotalama ve bir uygulama. *V. Kobi'ler ve Verimlilik Kongresi*. İstanbul.

Zhang, G., Gao, L., Shi, Y., 2010, A novel variable neighborhood genetic algorithm for multi-objective flexible job-shop scheduling problems. *Advanced Materials Research*, Vol.118, 369-373.

www.baskent.edu.tr/~ayyuce/END407%20Ders3a.pdf, Aydemir, A. (t.y.). END 543 Sezgisel Optimizasyon, Meta Sezgisel Yöntemler Ders Notu. [Erişim tarihi: 12 Temmuz 2011].

EKLER

EK A: ARAÇ ROTALAMA PROBLEMİ İÇİN SOLOMON TEST VERİLERİ.

**EK B: GELİŞTİRİLEN ÇOK AMAÇLI GENEL DEĞİŞKEN KOMŞULUK
ARAMA YAKLAŞIMI İÇİN HAZIRLANAN KODLAR.**

EK A

ARAÇ ROTALAMA PROBLEMİ İÇİN SOLOMON TEST VERİLERİ

C101 Problemine Ait Veri Kümesi

C201 Problemine Ait Veri Kümesi

R101 ve R201 Problemine Ait Veri Kümesi

RC101 ve RC201 Problemine Ait Veri Kümesi

Tablo A.1: Solomon C-101 Problem Kümesine ait 100 noktanın koordinat ve talep verisi

Müşteri Numarası	X koordinatı	Y koordinatı	Talep	Müşteri Numarası	X koordinatı	Y koordinatı	Talep
Depo	0	0	0	39	-40	-5	20
1	5	18	10	40	-5	-20	10
2	5	20	30	41	-5	-18	10
3	2	16	10	42	-7	-18	20
4	2	18	10	43	-7	-15	10
5	2	15	10	44	-8	-20	10
6	0	19	20	45	-10	-20	10
7	0	16	20	46	-10	-18	30
8	-2	18	20	47	-10	-15	10
9	-2	20	10	48	-12	-20	10
10	-5	16	10	49	-12	-15	10
11	-5	19	10	50	-14	-18	10
12	-15	35	20	51	-15	-20	10
13	-18	25	30	52	-15	-15	10
14	-18	35	10	53	4	-45	20
15	-20	30	40	54	2	-40	40
16	-20	35	40	55	2	-35	10
17	-22	25	20	56	0	-45	30
18	-25	25	20	57	0	-35	40
19	-25	30	10	58	-2	-45	30
20	-10	0	10	59	-2	-35	10
21	-10	2	20	60	-5	-45	20
22	-12	2	20	61	10	-20	10
23	-12	5	10	62	10	-15	20
24	-15	0	10	63	10	-10	50
25	-15	2	40	64	8	-20	10
26	-15	5	10	65	8	-10	10
27	-17	2	10	66	7	-15	10
28	-17	5	20	67	7	-10	10
29	-20	0	10	68	5	-20	10
30	-20	5	10	69	5	-15	10
31	-30	-15	20	70	55	-20	30
32	-30	-10	30	71	55	-15	20
33	-32	-10	40	72	13	-20	10
34	-32	-5	20	73	52	-20	10
35	-35	-15	10	74	13	-15	50
36	-35	-5	10	75	5	15	20
37	-38	-10	20	76	50	-15	10
38	-40	-10	30	77	48	-20	10

78	48	-15	20	90	20	5	10
79	47	-20	10	91	20	10	10
80	45	-25	10	92	27	35	20
81	45	-15	30	93	25	35	40
82	35	5	20	94	25	32	10
83	32	5	10	95	22	30	30
84	30	8	20	96	20	30	10
85	28	10	30	97	20	35	30
86	26	5	10	98	18	25	20
87	25	5	20	99	15	30	10
88	25	10	30	100	15	35	20
89	23	8	10				

Tablo A.2: Solomon C-201 Problem Kümesine ait 100 noktanın koordinat ve talep verisi

Müşteri Numarası	X koordinatı	Y koordinatı	Talep	Müşteri Numarası	X koordinatı	Y koordinatı	Talep
Depo	0	0	0	26	-32	12	10
1	12	25	10	27	-17	2	10
2	5	20	30	28	-36	5	20
3	22	19	10	29	-20	0	10
4	20	16	10	30	-20	5	10
5	2	15	10	31	-30	-15	20
6	-24	-8	20	32	-30	-10	30
7	18	20	20	33	-32	-10	40
8	-6	10	20	34	-32	-5	20
9	-12	20	10	35	-35	-15	10
10	-5	16	10	36	-35	-5	10
11	-5	19	10	37	-38	-10	20
12	-15	35	20	38	-40	-10	30
13	-18	25	30	39	-40	-5	20
14	-18	35	10	40	-4	-32	10
15	-20	30	40	41	-5	-18	10
16	-20	35	40	42	-7	-18	20
17	-22	25	20	43	-7	-15	10
18	-25	25	20	44	-8	-30	10
19	-25	30	10	45	-10	-20	10
20	-10	0	10	46	-6	-25	30
21	-10	6	20	47	-10	-15	10
22	-12	2	20	48	-4	-10	10
23	-26	16	10	49	8	-30	10
24	-15	0	10	50	-14	-18	10
25	-18	16	40	51	-15	-20	10

52	-15	-15	10	77	32	-5	10
53	4	-45	20	78	38	-10	20
54	2	-40	40	79	47	-20	10
55	2	-35	10	80	45	-25	10
56	0	-45	30	81	45	-15	30
57	-2	-35	40	82	35	5	20
58	-2	-45	30	83	32	5	10
59	-2	-40	10	84	30	8	20
60	-5	-45	20	85	46	-4	30
61	10	-20	10	86	26	5	10
62	10	-15	20	87	24	-4	20
63	10	-10	50	88	25	10	30
64	8	-20	10	89	16	14	10
65	4	-25	10	90	20	5	10
66	7	-15	10	91	20	10	10
67	7	-10	10	92	27	35	20
68	2	-20	10	93	2	8	40
69	5	-15	10	94	25	32	10
70	55	-20	30	95	22	30	30
71	55	-15	20	96	22	-10	10
72	13	-20	10	97	20	35	30
73	52	-20	10	98	18	25	20
74	13	-15	50	99	15	30	10
75	5	15	20	100	15	35	20
76	50	-15	10				

Tablo A.3: Solomon R-101 ve R201 Problem Kümelerine ait 100 noktanın koordinat ve talep verisi

Müşteri Numarası	X koordinatı	Y koordinatı	Talep	Müşteri Numarası	X koordinatı	Y koordinatı	Talep
Depo	0	0	0	51	14	23	10
1	6	14	10	52	-8	8	9
2	0	-18	7	53	2	-4	14
3	20	10	13	54	22	-6	18
4	20	-15	19	55	28	-12	2
5	-20	-5	26	56	18	-23	6
6	-10	-5	3	57	-3	-23	7
7	-15	15	5	58	1	-9	18
8	-25	8	9	59	-14	-11	28
9	20	25	16	60	-18	-1	3
10	-5	25	16	61	-23	-11	13
11	-15	30	12	62	-11	23	19
12	15	0	19	63	-8	34	10
13	-5	-10	23	64	-20	42	9

14	-20	-25	20	65	27	42	20
15	-5	-30	8	66	14	38	25
16	-25	-15	19	67	32	-30	25
17	-30	-5	2	68	21	4	36
18	-15	5	12	69	2	12	6
19	-20	25	17	70	2	21	5
20	10	30	9	71	22	33	15
21	10	-15	11	72	12	-19	25
22	10	-25	18	73	9	-18	9
23	20	-30	29	74	11	-22	8
24	30	0	3	75	14	-24	18
25	30	-15	6	76	14	7	13
26	10	-5	17	77	18	8	14
27	0	5	16	78	26	17	3
28	6	2	16	79	22	13	23
29	29	7	9	80	21	2	6
30	5	25	21	81	20	19	26
31	-4	17	27	82	-20	12	16
32	0	34	23	83	-21	2	11
33	18	17	11	84	-24	-4	7
34	30	20	14	85	-19	-13	41
35	28	30	8	86	-31	-17	35
36	-33	25	5	87	-7	-17	26
37	-15	-15	8	88	-9	17	9
38	-30	-30	16	89	-9	0	15
39	25	-23	31	90	-4	32	3
40	5	-10	9	91	-20	-16	1
41	7	-28	5	92	-13	-13	2
42	-11	-23	5	93	-17	-11	22
43	-12	-32	7	94	-9	-8	27
44	-24	-21	18	95	-10	-11	20
45	-29	3	16	96	-13	-8	11
46	-33	13	1	97	-10	-14	12
47	-27	21	27	98	-16	-14	10
48	-22	17	36	99	-15	-9	9
49	-29	33	30	100	-17	-17	17
50	12	12	13				

Tablo A.4: Solomon RC-101 ve RC201 Problem Kümelerine Problem Kümesine ait 100 noktanın koordinat ve talep verisi

Müşteri Numarası	X koordinatı	Y koordinatı	Talep	Müşteri Numarası	X koordinatı	Y koordinatı	Talep
1	0	0	0	52	15	-30	19
2	-15	35	20	53	-15	-20	3
3	-18	25	30	54	-20	0	5
4	-18	35	10	55	15	10	16
5	-20	30	40	56	-10	10	16
6	-20	35	20	57	10	-15	19
7	-22	25	20	58	-10	-25	23
8	-25	25	20	59	-25	-40	20
9	-25	30	10	60	-30	-30	19
10	-30	-15	20	61	-25	10	17
11	-30	-10	30	62	5	15	9
12	-32	-10	40	63	25	-15	3
13	-32	-5	20	64	25	-30	6
14	-35	-15	10	65	5	-20	17
15	-35	-5	10	66	-5	-10	16
16	-38	-10	20	67	1	-13	16
17	-40	-10	20	68	24	-8	9
18	-40	-5	20	69	0	10	21
19	4	-45	20	70	-9	2	27
20	2	-40	40	71	-5	19	23
21	2	-35	10	72	25	5	14
22	0	-45	10	73	23	15	8
23	0	-35	40	74	-38	10	5
24	-2	-45	30	75	-20	-30	8
25	-2	-35	10	76	-35	-45	16
26	-5	-45	20	77	20	-38	31
27	55	-20	30	78	-17	-47	7
28	55	-15	20	79	-32	6	27
29	52	-20	10	80	-34	18	30
30	50	-15	10	81	7	-3	13
31	48	-20	10	82	9	8	10
32	48	-15	20	83	-13	-7	9
33	47	-20	10	84	-3	-19	14
34	45	-25	10	85	17	-21	18
35	45	-15	30	86	23	-27	2
36	27	35	20	87	-19	-26	28
37	25	35	40	88	-28	-26	13

38	25	32	10	89	-16	8	19
39	22	30	30	90	27	-45	25
40	20	30	10	91	-3	-3	6
41	20	35	30	92	9	-8	13
42	18	25	20	93	13	-7	14
43	15	30	10	94	21	2	3
44	15	35	20	95	17	-2	23
45	15	32	10	96	16	-13	6
46	-20	32	10	97	15	4	26
47	-22	30	10	98	-36	-32	35
48	-38	-5	10	99	-14	2	9
49	2	-45	10	100	-14	-15	15
50	2	-38	10	101	-9	17	3
51	32	-15	30				

EK B**GELİŞTİRİLEN ÇOK AMAÇLI GENEL DEĞİŞKEN KOMŞULUK ARAMA
YAKLAŞIMI İÇİN HAZIRLANAN KODLAR***

*Bu tezde, kodların içindeki bazı satırlar (Şekil 3.11’de yer alan kodlar da dahil olmak üzere) silinmiştir. Kodların tamamına ulaşmak için İ.Ü. Endüstri Mühendisliği Bölümü’nden iletişime geçilebilir.

```

function [paretoCozum paretoR y]=gvns4_3C101(kapasite)
tic
load A
load R
kapasite=200;

ilkMesafe=toplamMesafeHesapla(R,B);
paretoCozum=[ paretoCozum; ilkMesafe size(R,1) 1];

    for s=1:1
        k=1;
        while k<=7
            if k==1
                [yeniR rotaN]=s_Degisim1_1(B,R);
            end
            if k==2
                [yeniR rotaN]=s_kaydirmaT_0(B, R, T);
            end
            if k==3
                [yeniR rotaN]=s_degisimT_Z(B, R, T, Z);
            end
            if k==4
                [yeniR rotaN rota2N]=s_interDegisim1_1(A,B,R,kapasite);
            end
            if k==5
                [yeniR rotaN rota2N]=s_interDegisimT_Z(A,B, R, T, Z,kapasite);
            end
            for t=1:5
                for l=1:6
                    if l==4
                        [yeni2R paretoCozum paretoR] = ya_Degisim1_1 (B, yeniR, rotaN, rota2N, paretoCozum, paretoR);
                    elseif l==5
                        [yeni2R paretoCozum paretoR]=ya_kaydirmaT_0(B, yeniR, T, rotaN, rota2N, paretoCozum, paretoR);
                    elseif l==6
                        [yeni2R paretoCozum paretoR]=ya_degisimT_Z(B, yeniR, T, Z,rotaN,rota2N,paretoCozum,paretoR);
                    elseif l==1
                        [yeni2R paretoCozum paretoR]=ya_interDegisim1_1(A, B, yeniR, rotaN, rota2N, kapasite,
paretoCozum, paretoR);
                    elseif l==2
                        [yeni2R paretoCozum paretoR]=ya_interKaydirmaT_0(A, B,yeniR, T, rotaN, rota2N, kapasite,
paretoCozum, paretoR);

                            end

                        end
                    end
                end
            end

            for i=1:r

                figure
                rotaCiz(A,tempR,B)
            end
            y=toc;
        end
    end

```

Şekil B.1: Çok amaçlı genel değişken komşuluk arama genel kodu.


```

function talep=talepHesapla(R1,A)

talep=0;
for i=1:length(R1)
    talep=talep+A(R1(i),3);
end

end

```

Şekil B.2: Talep hesaplama fonksiyonunun kodu.

```

function mesafe=toplamMesafeHesapla(R,B)
mesafe=0;
for j=1:size(R,1)
    mesafe=mesafe+mesafeHesapla(R(j,:),B);
end
end

```

Şekil B.3: Toplam mesafe hesaplama fonksiyonunun kodu.

```

function B=uzaklik(A)

for i=1:n
    for j=i:n
        x = A(i,1) - A(j,1);
        y = A(i,2) - A(j,2);
        B(i+1,j+1) = sqrt((x)^2 + (y)^2);
    end
end

for i=1:n
    x = A(i,1)^2;
    y = A(i,2)^2;
    B(i+1,1) = sqrt(x + y);
end

end

```

Şekil B.4: Uzaklık hesaplama fonksiyonunun kodu.

```

function [paretoCozum paretoR]=paretoKarsilastir(mesafe, ... ..)
m=size(paretoCozum,1);
cozumeal=0;
for i=1:m
    if mesafe>paretoCozum(i,1) && rotaSayisi>paretoCozum(i,2)
        cozumeal=0;
        break;
    elseif mesafe<paretoCozum(i,1) && rotaSayisi<paretoCozum(i,2)
        cozumeal=1;
        iler=[iler i];
    elseif mesafe==paretoCozum(i,1)

        cozumeal=0;
        break;

    end
end
end
if cozumeal==1

end
end

```

Şekil B.5: Pareto çözümün karşılaştırma fonksiyonunun kodu.

```

function [yeniR rotaN]=s_Degisim1_1(B,R)

yeniR=R;

rotaN=randi(n);

m=length(rota);
if m>1
    musterin=randi(m);

    for i=1:m

        uretR(i,:)=rota;
        uretR(i,i)=rota(musterin);

    end
end
enR=rota;

end
end

```

Şekil B.6: Rota içi değişim 1-1 silkeleme komşuluk uygulaması fonksiyonunun kodu.


```
function [yeniR rotaN]=s_kaydirmaT_0(B,R, T)
yeniR=R;
n=size(R,1);
rotaN=randi(n);
rota=R(rotaN,:);

m=length(rota);
if m>T
    musterin=randi(m-(...));

    for i=1:m
        if i>musterin+(...)

            mesafe(i)=mesafeHesapla(uretR(i,:),B);
        end
        if i<musterin

            mesafe(i)=mesafeHesapla(uretR(i,:),B);
        end
    end
    enR=rota;

    yeniR(rotaN,1:m)=enR;
end
end
```

Şekil B.8: Rotalar arası kaydırma 2-0 (silkeleme) komşuluk uygulaması fonksiyonunun kodu.

```

function [yeniR paretoCozum_c paretoR_c]=ya_degisimT_Z(B,R,T,Z,rotaN,rota2N,paretoCozum,paretoR)
Z=1;
n=size(R,1);
yeniR=R;
for r=1:2
    if r==1
        rotaN=rotaN;
    else
        R=yeniR;
    end
    if rotaN~=0
        rota=R(rotaN,:);
        ilkmesafe=mesafeHesapla(rota,B);
        m=length(rota);
        if m>T
            musterin=randi(m-( ));
            for i=1:m
                if i>musterin+( )
                    mesafe(i)=mesafeHesapla(uretR(i,:),B);
                end
                if i<musterin
                    uretR(i,:)=rota;
                    for k=i+1:1:musterin-1
                        end
                        mesafe(i)=mesafeHesapla(uretR(i,:),B);
                    [paretoCozum,paretoR]=paretoKarsilastir (toplamMesafeHesapla (yeniR rotaK, B), size (yeniR rotaK,
1), yeniR rotaK,paretoCozum,paretoR);
                end
            end
        end
    end
    end
    end
    end
    end
    paretoCozum_c=paretoCozum;
    paretoR_c=paretoR;
end

```

Şekil B.9: Rota içi değişim 1-1 ve değişim 2-1(yerel arama) komşuluk uygulaması fonksiyonunun kodu.


```

function [yeniR_paretoCozum_c paretoR_c]=ya_kaydirmaT_0(B,R, T,rotaN,rota2N,paretoCozum,paretoR)
n=size(R,1);
for r=1:2
    if r==1
        rotaN=rotaN;

    end
    if rotaN~=0
        rota=R(rotaN,:);

        ilkmesafe=mesafeHesapla(rota,B);
        m=length(rota);
        if m>T
            musterin=randi(m-(T-1));

            for i=1:m
                if i>musterin+(T-1)

                    yeniRrotaK=R;

                    [paretoCozum,paretoR]=paretoKarsilastir (toplamMesafeHesapla(yeniRrotaK,B)
, size(yeniRrotaK,1),yeniRrotaK,paretoCozum,paretoR);
                    end
                    if i<musterin
                        uretR(i,:)=rota;
                        mesafe(i)=mesafeHesapla(uretR(i,:),B);
                        yeniRrotaK=R;

                        [paretoCozum,paretoR]=paretoKarsilastir (toplamMesafeHesapla(yeniRrotaK,B)
, size(yeniRrotaK,1),yeniRrotaK,paretoCozum,paretoR);
                        end
                    end

                end
            end
        end

        paretoCozum_c=paretoCozum;
        paretoR_c=paretoR;
    end
end

```

Şekil B.11: Rota içi kaydırma 1-0 ve değişim 2-0 (yerel arama) komşuluk uygulaması fonksiyonunun kodu.

```

function [yeniR paretoCozum_c paretoR_c]=ya_interKaydirmaT_0(A, B,R, T,rotaN,rota2N,kapasite, paretoCozum,paretoR)
[n n2]=size(R);
for r=1:2
    if r==1
        rotaN=rotaN;
    else

end

rota=R(rotaN,:);

ilkMesafe=mesafeHesapla(rota,B);
m=length(rota);
if m>=T
    musteriN=randi(m-( ));
    for i=1:n
        if i~=rotaN

            mesafeT=mesafeHesapla( ,B);
            ilkmesT=mesafeT+ilkMesafe;
            for j=1:mT

                if ilkTalep<=kapasite

                    if ilkTalep==0
                        yeniRrotaK=R;

                        [paretoCozum,paretoR]=paretoKarsilastir (toplamsesafeHesapla(yeniRrotaK,B)
,size(yeniRrotaK,1),yeniRrotaK,paretoCozum,paretoR);
                    else
                        yeniRrotaK=R;

                        [paretoCozum,paretoR]=paretoKarsilastir (toplamsesafeHesapla(yeniRrotaK,B)
,size(yeniRrotaK,1),yeniRrotaK,paretoCozum,paretoR);
                    end
                end
            end
        end
    end
end
end
end
end

paretoCozum_c=paretoCozum;
paretoR_c=paretoR;
end

```

Şekil B.12: Rotalar arası kaydırma 1-0 ve kaydırma 2-0 (yerel arama) komşuluk uygulaması fonksiyonunun kodu.

ÖZGEÇMİŞ

Hacer Yumurtacı, 1979 yılında Kütahya'da doğdu. İlk ve ortaöğreniminin ardından 1997 yılında Atatürk Lisesi'nden mezun oldu. Aynı yıl girdiği Selçuk Üniversitesi Endüstri Mühendisliği Bölümü'nde lisans eğitimini 2001 yılında tamamladıktan sonra Yıldız Teknik Üniversitesi İşletme Yönetimi Anabilim Dalı'nda başladığı yüksek lisans programını 2005 yılında bitirdi. 2005 yılında İstanbul Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı Endüstri Mühendisliği Doktora Programına başladı. 2001 yılından bu güne dek İstanbul Üniversitesi Endüstri Mühendisliği Bölümü'nde araştırma görevlisi olarak görev yapmaktadır.