

**EŐZAMANLI TOPLA-DAĐIT ARAÇ ROTALAMA
PROBLEMİ İÇİN SEZGİSEL YAKLAŐIMLAR: GENETİK
ALGORİTMA VE KUŐ SÜRÜŐ ENİYİLEME**

Fatma Pınar GÖKSAL

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĐİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜŐÜ**

MAYIS 2010

ANKARA

Fatma Pınar GÖKSAL tarafından hazırlanan “EŞZAMANLI TOPLA-DAĞIT ARAÇ ROTALAMA PROBLEMİ İÇİN SEZGİSEL YAKLAŞIMLAR: GENETİK ALGORİTMA VE KUŞ SÜRÜSÜ ENİYİLEME” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Fulya ALTIPARMAK

Tez Danışmanı, Endüstri Mühendisliği A.B.D.

Bu çalışma, jürimiz tarafından oy birliği ile Endüstri Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Berna DENGİZ (Başkan)

Endüstri Mühendisliği A.B.D., Başkent Üniversitesi

Prof. Dr. Fulya ALTIPARMAK

Endüstri Mühendisliği A.B.D., Gazi Üniversitesi

Prof. Dr. Ömer Faruk BAYKOÇ

Endüstri Mühendisliği A.B.D., Gazi Üniversitesi

Tarih: 18/05/2010

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Bilal TOKLU

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Fatma Pınar GÖKSAL

**EŞZAMANLI TOPLA-DAĞIT ARAÇ ROTALAMA PROBLEMİ
İÇİN SEZGİSEL YAKLAŞIMLAR: GENETİK ALGORİTMA VE
KUŞ SÜRÜSÜ ENİYİLEME
(Yüksek Lisans Tezi)**

Fatma Pınar GÖKSAL

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Mayıs 2010**

ÖZET

Günümüz rekabet ortamında işletmelerin, tedarik zincirindeki süreçlerini eniyilemek ve etkin bir şekilde yönetmek amacıyla çeşitli stratejik ve operasyonel kararlar alması gerekmektedir. En önemli operasyonel kararlardan birisi araç rotalarının belirlenmesi ile ilgilidir. Klasik araç rotalama problemi, bir depodan müşterilerin dağıtım ya da toplama taleplerini karşılayacak en uygun rotaların belirlenmesi olarak tanımlanabilir. Pratikte karşılaşılan kısıtlamalara dayalı olarak literatürde farklı araç rotalama problemleri önerilmiştir. Bu problemlerden birisi Eşzamanlı Topla-Dağıt Araç Rotalama Problemidir (ETD_ARP). ETD_ARP depodan müşterilere ürün dağıtımının yanı sıra müşterilerden de depoya geri dönecek ürünlerin toplanması operasyonlarının eşzamanlı gerçekleştirildiği bir araç rotalama problemdir. ETD_ARP bir NP-zor problemdir. Bu nedenle problemin çözümü için literatürde çeşitli sezgisel algoritmalar geliştirilmiştir. Bu tez kapsamında ETD_ARP'nin çözümü için Genetik Algoritma (GA), Kuş Sürüsü Eniyileme (KSE) ve Değişken Komşu İniş (DKİ) algoritmasına dayalı iki melez algoritma (GA_DKİ ve KSE_DKİ) geliştirilmiştir. Melez algoritmalarda GA ve KSE çözüm uzayında araştırma yapmak, DKİ ise arama sırasında bulunan bir ya da birkaç çözüm etrafında derinlemesine aramayı gerçekleştirmek amacıyla kullanılmıştır. GA_DKİ ve KSE_DKİ'nin performansını araştırmak amacıyla

deneysel çalışma yapılmıştır. 76 test problemi ile yapılan deneysel çalışmada, GA_DKİ ve KSE_DKİ'nin çözüm kalitesi açısından literatürdeki diğer algoritmalar ile rekabet edebildiği ve bazı problemler için bilinen eniyi çözümü iyileştirdiği görülmüştür.

Bilim Kodu : 906.1.141
Anahtar Kelimeler : Eşzamanlı topla-dağıt araç rotalama problemi, genetik algoritma, kuş sürüsü eniyileme, değişken komşu iniş
Sayfa Adedi : 73
Tez Yöneticisi : Prof. Dr. Fulya ALTIPARMAK

**HEURISTIC APPROACHES FOR VEHICLE ROUTING PROBLEM WITH
SIMULTANEOUS PICKUP AND DELIVERY: GENETIC ALGORITHM
AND PARTICLE SWARM OPTIMIZATION**

(M.Sc. Thesis)

Fatma Pınar GÖKSAL

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
May 2010**

ABSTRACT

In today's competitive environment, it is obvious that companies should make strategic and operational decisions in order to optimize and manage the processes in their supply chain more efficiently. One of the most important operational decisions concerns to determine of vehicle routes. Classical vehicle routing problem can be defined as the problem of designing optimal delivery or pickup routes from one depot to a number of customers. Different variants of vehicle routing problem depending on some restrictions, which are faced in the practice, have been proposed in the literature. One of these variants is the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRP_SPD). In the VRP_SPD, pickup and delivery demands of customers in each route are met simultaneously. Since the VRP_SPD is an NP-hard problem, different heuristic solution algorithms have been proposed to solve the problem in the literature. In this thesis, two hybrid algorithms based on Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Variable Neighborhood Descent (VND) algorithm called GA_VND and PSO_VND are developed. While GA and PSO are used to explore solution space of the problem, VND is implemented to intensify around one or several good solutions found during search process of the hybrid algorithms. An experimental study is carried out to investigate the performances of GA_VND and PSO_VND. The computational results over 76

test instances indicate that the proposed hybrid algorithms compete with the heuristic approaches, which are proposed in the literature for the VRP_SPD, in terms of solution quality and also GA_VND improves known best solution for some instances.

Science Code	: 906.1.141
Key Words	: Vehicle routing problem with simultaneous pickup and delivery, genetik algorithm, particle swarm optimization, variable neighbourhood descent
Page Number	: 73
Adviser	: Prof. Dr. Fulya ALTIPARMAK

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarını hiçbir zaman esirgemeyerek beni yönlendiren tez danışmanım Sayın Prof. Dr. Fulya ALTIPARMAK'a sonsuz teşekkürlerimi sunarım.

Çalışmalarım sırasında bana vakit ayıran Gazi Üniversitesi Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü Arş. Gör. İsmail Karaoğlu'na şükranlarımı sunarım.

Son olarak tez çalışmalarımı maddi olarak destekleyen TÜBİTAK'a teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xii
ŞEKİLLERİN LİSTESİ	xiii
SİMGELER VE KISALTMALAR	xiv
1. GİRİŞ	1
2. ARAÇ ROTALAMA PROBLEMİ	4
2.1. Araç Rotalama Problemlerinin Sınıflandırması	7
2.1.1 Kapasite kısıtlı ARP	7
2.1.2. Mesafe kısıtlı ARP	8
2.1.3. Zaman pencereli ARP	8
2.1.4. Topla-Dağıt ARP	9
2.2. ARP için Çözüm Yöntemleri	9
2.2.1. Kesin yöntemler	11
2.2.2. Sezgisel yöntemler	12
2.2.3. Metasezgisel yöntemler	14
3. EŞZAMANLI TOPLA-DAĞIT ARAÇ ROTALAMA PROBLEMİ	15
3.1. ETD_ARP için Literatür Araştırması	19

Sayfa

4. ETD_ ARP İÇİN GELİŞTİRİLEN SEZGİSEL YAKLAŞIMLAR	23
4.1. Önerilen Genetik Algoritma	23
4.1.1. Çözüm gösterimi	24
4.1.2. Başlangıç yığınının oluşturulması	30
4.1.3. Seçim mekanizması ve genetik operatörler	31
4.1.4 Önerilen melez GA'nın adımları	37
4.2. Önerilen Kesikli Kuş Sürüsü Eniyileme Algoritması	39
4.2.1. Çözüm gösterimi	42
4.2.2. Başlangıç yığınının oluşturulması	42
4.2.3. Önerilen KSE'de pozisyonların güncellenmesi	43
4.2.4. Önerilen melez KSE algoritmasının adımları	46
5. DENEYSEL ÇALIŞMALAR	49
5.1. Test Problemleri	49
5.1.1. Dethloff'ın test problemleri	49
5.1.2. Salhi ve Nagy'nin test problemleri	50
5.2. DeneySEL Sonuçlar	51
5.2.1. Parametrelerin belirlenmesi	53
5.2.2. ETD_ ARP için deneySEL karşılaştırma sonuçları	55
5.2.3. KTD_ ARP için deneySEL karşılaştırma sonuçları	60
6. SONUÇ	62
KAYNAKLAR	65
EKLER	71

Sayfa

EK-1. GA_DKİ'nin parametreleri için varyans analizi sonuçları	70
EK-2. KSE_DKİ'nin parametreleri için varyans analizi sonuçları	71
ÖZGEÇMİŞ	73

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1 Araç rotalamanın kısa bir tarihçesi	11
Çizelge 3.1. ETD-ARP ile ilgili sezgisel çalışmalar	22
Çizelge 5.1. Farklı bilgisayarlar için Mflops değerleri ve dönüştürme faktörü	52
Çizelge 5.2. Dethloff'ın test problemleri için GA ile KSE'nin karşılaştırılması	55
Çizelge 5.3. Dethloff'ın test problemleri için deneysel sonuçlar	57
Çizelge 5.4. Salhi ve Nagy'nin ETD_ARP türündeki test problemleri için deneysel sonuçlar	59
Çizelge 5.5. Salhi ve Nagy'nin KTD_ARP türündeki test problemleri için deneysel sonuçlar	61

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Bir ARP'nin çözümünden elde edilen rotaların gösterimi	6
Şekil 4.1. 10 müşterili ETD_ARP için bir kromozom örneği	25
Şekil 4.2. ETD_ARP için ayırma algoritması	27
Şekil 4.3. Bir kromozomdan ETD_ARP için çözümün elde edilmesi	29
Şekil 4.4. Benzer iş 2-noktalı sıralı çaprazlama operatörünün uygulanması ...	33
Şekil 4.5. Önerilen GA'nın adımları	38
Şekil 4.6. Standart KSE algoritmasının adımları	41
Şekil 4.7. PMX çaprazlama operatörünün uygulanışı	45
Şekil 4.8. Önerilen KKSE algoritmasının sözde kodu	47

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simge ve kısaltmalar açıklamaları ile aşağıda verilmiştir.

Simgeler	Açıklama
Δ	Yığının çözüm değerleri arasındaki enküçük fark
$ N $	Problemdeki müşteri sayısı
$ Y $	Yığın genişliği
A	Düğümleeri birbirine bağlayan ayrıtlar kümesi
c_{ij}	$[i-j]$ ayrıtının uzunluğunu/maliyetini $[i-j]$
d_i	i müşterisinin dağıtım talebi ($i \in N_0$)
$f(s)$	s çözümünün amaç fonksiyonu değeri
G	Global eniyi pozisyon
h_1	Bilişsel bileşenin pozitif hızlanma sabiti
h_2	Sosyal bileşenin pozitif hızlanma sabiti
N	Düğüm kümesi $(0, \dots, N)$
$N(s)$	s çözümünün komşuluk yapısı
N_0	Müşteriler kümesi $(1, \dots, N)$
n_i	i müşterisi ($i \in N_0$)
P_i	i . kuşunun yerel eniyi pozisyonu
Q	Araç kapasitesi
q_i	i müşterisinin toplama talebi ($i \in N_0$)
r'	$[0,1]$ aralığında düzgün dağılımdan örneklenen rassal sayı
s	Problemin mümkün bir çözümü
T^0	Başlangıç sıcaklığı
V_i^t	i . kuşun t zamanındaki hızı
w	Atalet faktörü
X_i^t	i . kuşun t zamanındaki pozisyonu
Y	Yığın kümesi
z_{ij}	H grafindaki $[i-j]$ hattının maliyeti

Simgeler γ **Açıklama**

DKİ algoritmasının uygulama olasılığı

Kısaltmalar**Açıklama****ARP**

Araç Rotalama Problemi

BEÇ

Bilinen Eniyi Çözüm

BJA

Büyük Komşuluk Arama

CS

Çözüm Süresi

DKİ

Değişken Komşu İniş

EÇ

Eniyi Çözüm

ETD_arp

Eşzamanlı Topla-Dağıt Araç Rotalama Problemi

GA

Genetik Algoritma

GSP

Gezgin Satıcı Problemi

KARP

Klasik Araç Rotalama Problemi

KEP

Kombinatoriyal Eniyileme Problemi

KK

Kayıttan Kayıda

KKARP

Kapasite Kısıtlı Araç Rotalama Problemi

KKE

Karıncı Kolonisi Eniyileme

KKS

Karıncı Kolonisi Sistemi

KKSE

Kesikli Kuş Sürüsü Eniyileme

KSE

Kuş Sürüsü Eniyileme

KTD_arp

Karma Topla-Dağıt Araç Rotalama Problemi

OÇS

Ortalama Çözüm Süresi

ÖDST-arp

Önce Dağıt Sonra Topla Araç Rotalama Problemi

PYA

Paralel Yerel Arama

SY

Sapma Yüzdesi

TA

Tabu Arama

TB

Tavlama Benzetimi

TD_arp

Topla-Dağıt Araç Rotalama Problemi

TTA

Tepkisel Tabu Arama

Kısaltmalar**UHM****YA****Açıklama**

Uyarlamalı Hafıza Metodolojisi

Yerel Arama

1. GİRİŞ

Günümüzde küreselleşmenin etkisiyle işletmeler ürünlerini tüm dünyaya sunarken, tüketim bilincinin artması ve azalan kaynaklar nedeniyle global pazarın çetin rekabet koşullarına da uyum sağlamak zorunda kalmışlardır. Bu ortamda işletmelerin rakiplerine karşı rekabet avantajı elde ederek ayakta kalabilmeleri, müşteri memnuniyetini mümkün olan en yüksek karlılık seviyesinde sağlamaları ile mümkündür. Birbiriyle çelişen bu iki amaç doğrultusunda, işletme çıkarlarından olabildiğince ödün verilmeden müşteriye kaliteli malın ucuz ve hızlı şekilde ulaştırılmasına çalışılmalıdır. Diğer bir anlatımla, kaynakların verimli şekilde kullanımı ile müşteri taleplerinin ihtiyaçlar doğrultusunda etkin bir şekilde karşılanması gerekmektedir. Bu amaca ulaşmanın yollarından birisi de, tedarikçiden müşteriye kadar uzanan dağıtım ağının etkin bir şekilde tasarlanmasıdır.

Dağıtım ağlarının tasarımında hammadde, yarı ürün ve ürünün etkin şekilde depolanmasının yanı sıra verilmesi gereken kararlardan birisi de, müşterilere ürünlerin nasıl gönderileceğidir. İşletmelerde karşılaşılan bu problem, bazı sektörlerde oldukça yüksek maliyetlere neden olmaktadır [Erol, 2006]. Bu nedenle tüm müşterilere servis sunulabilmesi için bazı operasyonel kısıtları dikkate alarak enküçük maliyetli araç rotalarının belirlenmesi olarak tanımlanan Araç Rotalama Problemi (ARP) işletme tarafından çözülmelidir. İşletmelerdeki toplam lojistik ve dağıtım maliyetlerinin $1/3 - 2/3$ gibi büyük bir kısmı taşımadan kaynaklanmaktadır. Bu nedenle ürün kalitesine hiçbir katkısı olmayan bu masrafları azaltarak, tasarrufa dönüştürme fırsatı sağlayan ARP işletme yöneticileri açısından son yıllarda önemli bir ilgi alanı haline gelmiştir.

ARP'nin matematiksel modeli ve çözüm yaklaşımı ilk olarak 1959 yılında Dantzing ve Ramser tarafından literatüre kazandırılmıştır. Klasik Araç Rotalama Problemi (KARP) olarak bilinen bu problemde; bir işletme biriminden (örneğin depo) çıkarak, konumları ve talep miktarları önceden belli bir grup müşteriye servis sağlayan homojen (kapasite ve tipleri aynı olan araçlardan oluşmuş) bir araç filosunun rotaları kat ettikleri toplam mesafeyi en azlayacak şekilde belirlenmiştir. KARP'de, araç

kapasitelerinin aşılması, her müşteriye bir kere uğranılması ve araç rotalarının başladıkları noktada sonlandırılması kısıtları mevcuttur. Böylece KARP, kapasite kısıtının olduğu Gezgin Satıcı Problemi (GSP) olarak düşünülmektedir. Bu alandaki ilk sezgisel yaklaşım ise 1964 yılında Clarke ve Wright tarafından geliştirilmiştir.

Dünya kaynaklarının azalması, müşterilerin bilinçlenmesi gibi nedenlerden dolayı müşterilerden işletmelere gerçekleşen ürün taşımacılığı günümüz lojistik faaliyetlerinin önemli bir unsuru haline gelmiştir. Ürünün tesislerden müşterilere dağıtımı ile müşterilerden toplanması işlemlerinin aynı araçla gerçekleştirildiği problemler olan Topla-Dağıt Araç Rotalama Problemi (TD_ARP), 1980'li yıllardan bugüne kadar çeşitli çalışmaların yapıldığı bir ARP türüdür. Tüketimden kaynaklanan atıkların toplanıp geri dönüşüm tesislerinde ekonomik değerlere sahip ürünler haline dönüştürülmesi, müşterilerin satın aldıkları ürünleri iade etmesi gibi durumlar toplama faaliyetlerine örnek olarak verilebilir [Karaoğlu, 2009]. Bu tez kapsamında bir müşteriye uğranıldığında toplama ve dağıtımın aynı anda yapıldığı Eşzamanlı Topla-Dağıt Araç Rotalama Problemi (ETD_ARP) ele alınmıştır. Sağlık, gıda, otomotiv gibi farklı sektörlerde ETD_ARP'nin uygulamaları ile sıklıkla karşılaşılmaktadır.

ARP'nin bir türü olan ETD_ARP de NP-zor problemler sınıfında yer almaktadır. Literatürde bu problemin çözümü için geliştirilmiş birçok sezgisel yöntem mevcuttur. Bu tez kapsamında ETD_ARP için GA ve KSE olarak bilinen metasezgisellere dayalı olarak iki yeni algoritma geliştirilmiştir. Bu algoritmaların performansları literatürdeki test problemleri dikkate alınarak incelenmiştir. Son yıllarda araştırmacılar tarafından önerilen algoritmaların büyük bir kısmı melez algoritmadır. Melez algoritmalarda, metasezgisel algoritma basit Yerel Arama (YA) veya başka bir metasezgisel algoritma ile birleştirilmekte ve birden fazla metasezgiselin avantajından birlikte yararlanılmaktadır. Bu amaçla, önerilen metasezgisellerde de DKİ algoritması kullanılmıştır. Bu anlamda, ETD_ARP için ilk defa GA ve KSE'nin DKİ ile birleştirildiği GA_DKİ ve KSE_DKİ algoritmaları önerilmiş ve literatürdeki diğer yöntemlerle rekabet edebilecek nitelikte sonuçlar elde edilmiştir.

Tezin diğ er b l mleri řu řekilde d zenlenmiřtir: İkinci b l mde ARP tanımlanmıř, t rleri ve c z m y ntemleri detaylı olarak incelenmiřtir.     nc  b l mde tez kapsamında dikkate alınan ETD_ARP a ıklanarak, problemle ilgili literat r arařtırmasına yer verilmiřtir. D rd nc  b l mde sırasıyla, bu problemin c z m  i in geliřtirilen GA_DK  ve KSE_DK  algoritmaları anlatılmıřtır. Devam eden b l mde ise  nerilen algoritmalarla ilgili deneysel c alıřmalar yapılarak;  nce ETD_ARP i in geliřtirilen y ntemlerin performans deęerlendirilmesinde kullanılan test problemleri, ardından da bu problemlere iliřkin c z mler verilmiřtir. Tezin son b l m  olan sonu  kısmında ise, yapılan c alıřmalar ve elde edilen sonu lar  zetlenmiřtir.

2. ARAÇ ROTALAMA PROBLEMİ

Müşteri taleplerini en kısa zaman, en kısa yol veya en az maliyet gibi amaçlardan birini veya birkaçını en iyileyecek şekilde karşılayan araç rotalarının belirlenmesi günümüz işletmelerinin üzerinde durduğu problemlerden birisidir.

Bir ARP'yi oluşturan ana kavramlar; müşteriler, işletme birimi/birimleri, yol şebekesi, araçlar ve sürücülerdir. Bu kavramlar aşağıda kısaca açıklanmaktadır.

Yol Şebekesi: Müşteri ve işletme birimlerinin düğümlerle, bunlar arasındaki yolun ise hatlarla temsil edildiği bir graftır. Şebekedeki her hat bir maliyet ya da uzunlukla birlikte tanımlanmaktadır. Eğer yollar simetrikse, i noktasından j noktasına giden yol ile j 'den i 'ye giden yolun uzunluğu/maliyeti birbirine eşittir. Ancak yolların asimetrik olduğu şebekeler de söz konusu olmaktadır.

Müşteriler: Şebekede düğümlerde yer almaktadırlar. İhtiyacını karşılamak için her müşteriye araçlar tarafından en az bir kere uğranılır. Müşterinin talebi dağıtım, toplama ya da her ikisi birden olabilir. Taleplerine göre müşteriler dağıtım, toplama ya da hem dağıtım hem toplama müşterisi olmak üzere sınıflandırılabilir. ARP çözümünün uygun bir çözüm olarak kabul edilmesi için tüm müşterilerin ihtiyaçlarının karşılanması zorunludur. Ayrıca özelliklerine göre, her müşterinin yalnızca belirli bir zaman aralığında ziyaret edilmesi veya müşteriler arası önceliklerin sağlanması gibi başka kısıtlar da probleme eklenebilir.

İşletme Birimi: Hizmetin götürüldüğü işletmeye ait noktalardır ve şebekenin başlangıç düğüm/düğümüleri olarak gösterilirler. Araçlar işletme birimlerinden hareket ederek müşterilere uğrarlar. Depolar, dağıtım merkezleri ve bayi gibi üniteler gerçek hayattaki işletme birimleridirler. Genellikle bir tane işletme birimi olmasına rağmen, birden fazla işletme biriminin olduğu problem türleri de mevcuttur. Bu problemler, çok depolu ARP olarak adlandırılır. Çok depolu ARP'de birimlerin karşılayabileceği en büyük talep miktarı da bir kısıt olarak göz önüne

alınabilmektedir. Genelde araçların rotalarını işletme biriminde bitirmeleri kısıtı kullanılır, ancak rotaların işletme biriminde sonlanmadığı durumlar da söz konusu olabilmektedir.

Araçlar: Kullanılan araçların özelliklerine göre çok sayıda ARP türü söz konusudur. Tüm problemlerde araçlara yüklenen ürün miktarının araç kapasitesini aşmaması yani kapasite kısıtı söz konusudur. Kapasite kısıtının yanı sıra araçların gidebileceği en uzun yol veya süre gibi kısıtlar da olabilmektedir. Ayrıca, aracın tipi bazı sektörlerde özellikle önemli olup, aracın uğrayabileceği müşterileri belirlemektedir. Örneğin, gıda sektöründe farklı koşullarda taşınması gereken mallar aynı araçla taşınmaz. Ancak, kullanılan araç farklı taşıma koşuluna sahip bölümlere ayrılabilirse bu ürünlerin bir arada taşınması mümkün hale gelmektedir. Aynı şekilde, çift taraflı kapısı olan araçlarda aynı anda yükleme boşaltma işlemi yapılabilirken, tek kapılılarda bu mümkün olmamaktadır. Bazı problemlerde homojen yani birbiriyle tamamen özdeş araçlardan oluşan bir araç filosuyla, bazı problemlerde ise heterojen bir araç filosuyla rotalama yapılmaktadır. ARP’de genellikle araç sayısı değişkendir ve araçların sabit maliyeti olabilmektedir. Bu problemlerde, gidilen mesafeden kaynaklanan maliyete araç sabit maliyetinin eklenmesiyle elde edilen toplam maliyet en küçüklenmeye çalışılmaktadır.

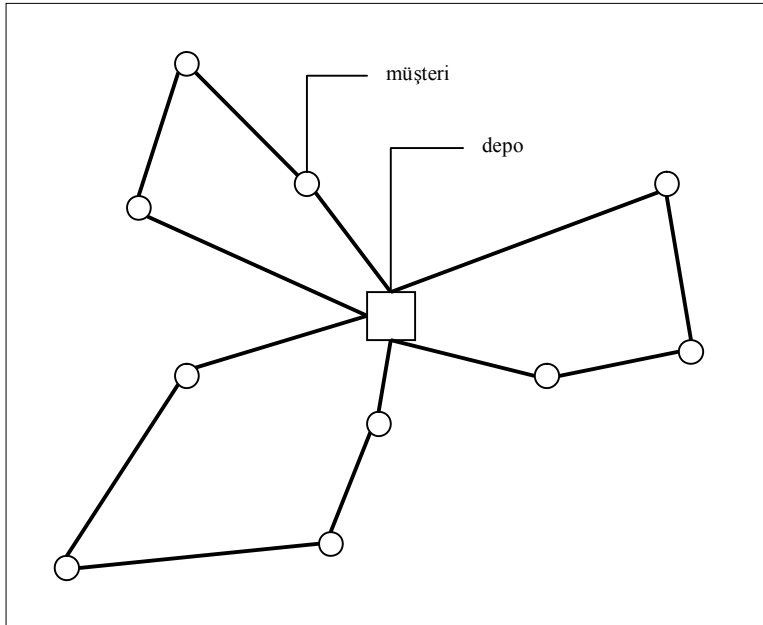
Sürücüler: Bazı ARP türlerinde sürücülere ait çalışma periyodu, fazla mesai, öğle arası gibi kısıtlar yer alabilmektedir.

ARP dağıtım ve/veya toplama faaliyetlerinin yönetimiyle uğraşan problemler bütününün genel bir adıdır. ARP’yi oluşturan kavramların özellikleri çözülecek problemin türünü belirlemektedir. KARP’de kapasite kısıtına sahip homojen bir araç filosu, araç filosunun servise başladığı bir depo, yerleri bilinen müşteriler ve bu müşterilerin miktarları önceden belli dağıtım/toplama talepleri vardır. Amaç aşağıdaki kısıtları sağlarken araçların katettiği toplam mesafeyi en küçükleyen rotalar kümesinin bulunmasıdır.

1. Tüm müşterilerin talepleri karşılanmalıdır.

2. Her müşteriye bir kez uğranılmalıdır.
3. Tüm rotalar depodan başlamalı ve tekrar depoda sonlanmalıdır.
4. Bir rota boyunca müşterilere dağıtılan toplam ürün miktarı araç kapasitesini geçmemelidir.

KARP en genel haliyle şu şekilde tanımlanabilir: $G=(N,A)$ tam bağlı (bütün düğümlerin bir ayrıtla birbirine doğrudan bağlı olduğu) bir graf olsun. Bu grafta $N=\{n_0,n_1,n_2,...,n_{|N|}\}$ ilk düğümün (n_0) ana depoyu diğerlerinin ise müşterileri gösterdiği düğüm kümesi; $A=\{(i,j)|i\neq j \text{ ve } i,j\in N\}$ ise bu düğümleri birbirine bağlayan ayrıtlar kümesidir. Bu grafta c_{ij} , (i,j) ayrıtının uzunluğunu (maliyetini) temsil etmekte ve üçgensel eşitsizlik $(c_{ij}\leq c_{ik}+c_{kj})$ sağlanmaktadır. Her müşterinin talep miktarı d_i 'dir ve taşımayı gerçekleştirecek Q kapasiteli yeterli sayıda araç depoda hazır olarak beklemektedir. Örnek bir ARP çözümü Şekil 2.1'de verilmektedir.



Şekil 2.1. Bir ARP'nin çözümünden elde edilen rotaların gösterimi

2.1. Araç Rotalama Problemlerinin Sınıflandırması

ARP belirli bir problemi göstermek yerine, KARP'nin yapısındaki geniş çeşitlilikten kaynaklanan birbirinden farklı problemleri kapsamaktadır. Pratikte farklı durumlar için geliştirilmiş çok sayıda ARP ile karşılaşilmektedir. Buna göre problemde yer alan işletme birimleri, araçlar, sürücüler, yollar ve müşterilerin özelliklerine göre çeşitli ARP türleri tanımlanmaktadır. Bu nedenle farklı kriterler dikkate alınarak ARP için çeşitli sınıflandırmalar yapılmıştır. Örneğin; çevre durumuna göre deterministik ve stokastik, rotaların durumuna göre açık uçlu ve kapalı uçlu, yolların durumuna göre simetrik ve asimetrik ARP'den bahsedilebilir.

Sınıflandırmada kullanılan kriterlerin en önemlisi kısıtlardır ve problem çoğu zaman ARP'de dikkate alınan kısıtın ismi ile beraber anılmaktadır. Ancak literatürde kısıtlarına göre sınıflandırma için farklı yaklaşımlar mevcuttur. Çünkü artan rekabet şartları ve değişen çevre koşulları sebebiyle rotalama problemlerine giderek daha fazla kısıt eklenmektedir. Probleme kısıt eklendikçe, tanımlanması ve çözümü zorlaşmaktadır. Pratikte kurulan lojistik sistemlerinde ARP'nin birebir modellenip çözümlenmesi zor olduğundan, bazı önemli kısıtlar seçilip diğerleri göz ardı edilerek problemler tanımlanmaya çalışılır. Aşağıda dikkate alınan önemli kısıtlara göre ARP türlerinden kısaca bahsedilmiştir.

2.1.1 Kapasite kısıtlı ARP

Kapasite kısıtlı ARP (KKARP), $|N|$ adet müşterinin isteklerini karşılamak üzere kullanılacak yükleme kapasitesi sınırına sahip araçların rotalanması problemidir. Hemen hemen tüm ARP uygulamalarında bu kısıt kullanılmaktadır.

Sadece dağıtım veya sadece toplama talebinin söz konusu olduğu bir problemde, rotadaki müşterilerin toplam dağıtım/toplama miktarının araç kapasitesi Q 'yu geçmemesi, bu kısıtın sağlanması için gerekli ve yeterli şarttır. Ancak dağıtımla beraber toplama işleminin de yapıldığı TD_arp'de, kapasite kısıtının sağlanması

(rotanın uygun olması) için sadece bu şartın yerine getirilmesi yeterli değildir. Her iki durumda da kapasite kısıtının sağlanması için; rota üzerindeki toplam dağıtım ve/veya toplama miktarı araç kapasitesi Q 'dan büyük olmamalıdır. Aksi halde, kapasite kısıtı altında problemin bir çözümü olabilmesi için, her müşteriye bir sefer uğranılması zorunluluğundan vazgeçilmesi, yani taleplerin araçlara bölünmesine izin verilmesi gerekmektedir. Bu durumda, ayrılabilir talepli KKARP'den söz edilebilir. Ancak bir müşterinin talebini parçalara bölmek çoğu zaman işletmeler için uygun bir çözüm olmadığından, akademik alanda ayrılabilir talepli problemlerle ilgili çalışmalar az sayıdadır. Çünkü taleplerin bölünmesi çoğu zaman fiziksel olarak imkansızdır veya müşteri hizmeti veya finansal yönetim güçlükleri yaratmaktadır.

2.1.2. Mesafe kısıtlı ARP

Mesafe kısıtlı ARP rotalara atanan her aracın gidebileceği maksimum mesafe kısıtının bulunduğu problemlerdir. Literatürde mesafe kısıtlı ARP'nin de farklı versiyonları bulunmaktadır. Örneğin, mesafe kısıtı yerine seyir süresi sınırlaması da konulabilir. Bu durumda her müşteri için servis süresi dikkate alınmaktadır. Ayrıca yükleme kapasitesi ile mesafe kısıtının birlikte kullanılması da karşılaşılan bir durumdur.

2.1.3. Zaman pencereci ARP

Her müşterinin hizmet görmesi gereken belli bir zaman aralığının $([a_i, b_i])$ olması, araçların müşteriye bu zaman aralıklarında ulaşmasını zorunlu hale getirmektedir. Zaman pencereci ARP, müşteriye belirli bir zaman aralığında uğranılması kısıtı altında araç rotalarının belirlendiği problemlerdir. Burada yer alan $[a_i, b_i]$ ifadesine *zaman penceresi* denilmektedir.

2.1.4. Topla-Dağıt ARP

TD_ARP’de talep edilen ürünlerin depodan müşterilere dağıtılması ile müşteriler tarafından depoya gönderilecek ürünlerin toplanması olmak üzere iki farklı işlem söz konusudur ve müşteri ihtiyaçlarını karşılamak için aynı araç hem dağıtım hem toplama yapmak durumundadır. Bu kısıt problemi planlamayı daha zor hale getirir ve araç kapasitelerinin iyi kullanılamamasına, yolun uzamasına ve daha fazla araç ihtiyacının doğmasına neden olur. Çünkü servis götürülen müşterilerin taleplerine göre araç yükü rota boyunca dalgalı olarak değişmektedir. Diğer bir ifade ile, aracın depodan çıktığında rotası boyunca ziyaret edeceği müşterilere dağıtılacak ürün miktarlarını karşılayacak ve müşterilerden topladığı tüm ürünleri taşıyabilecek kapasiteye sahip olması rotanın uygun (*feasible*) olması için yeterli değildir. Bu tez kapsamında incelenecek olan TD_ARP ile ilgili detaylı bilgi 3. bölümde verilmiştir.

2.2. ARP için Çözüm Yöntemleri

Eniyileme problemi; problemin kısıtlarını sağlayan tüm mümkün çözümlerden (s) oluşan *çözüm uzayı* (S) ve her çözüme ($s \in S$) bir değer atayan *amaç fonksiyonu* ($f(s)$) ile tanımlanabilir. Bir eniyileme problemi, amaç fonksiyonunu en iyileyen (duruma göre en küçükleyen veya en büyükleyen) s^* çözümünü (*eniye çözüm*) veren, problem bilinmeyenlerinin bulunmasını içermektedir. Dolayısıyla, çözüm uzayında yer alan her s çözümü, eniyileme algoritmaları için arama uzayında göz önüne alınması gereken bir noktadır. Karar değişkenlerinin alabileceği değerlerin sürekli olup olmamasına göre; iki sınıf eniyileme probleminden bahsedilmektedir. Karar değişkenlerinin kesikli olduğu Kombinatoryal Eniyileme Problemi (KEP); genelde ayrık niceliklerin optimal şekilde gruplanması, sıralanması ve seçilmesine çalışmaktadır. Buna göre; KEP çözümlerin yapısına göre gruplandırma problemleri, sıralama problemleri ve karma problemler olmak üzere 3’e ayrılmaktadır.

Gruplandırma (Sınıflandırma) problemleri

Bu tip problemlerde genel amaç bir küme içinde yer alan elemanların gruplandırılmasıdır. Örneğin tesis yeri belirleme probleminde, müşteriler gruplara ayrılarak, her gruba bir tesisin hizmet vermesi sağlanmaktadır.

Sıralama problemleri

Bu tip problemlerde genel amaç, bir grup elemanın belli bir sıraya göre permütasyon mantığında sıralanmasıdır. Sıralama problemlerine örnek olarak GSP verilebilir.

Karma problemler

Hem sıralama, hem de gruplandırmanın birlikte yapılmasının gerektiği, dolayısıyla çözüm uzayının oldukça geniş olduğu problemlerdir. ARP karma problemler kapsamına girmektedir.

ARP, NP-Zor yapıdaki bir KEP'dir. Çünkü problemin çözüm uzayı müşteri ve araç sayısı ile üstel olarak artmaktadır. Bu durum, çözüm uzayındaki eniyi çözümü bulan *kesin algoritmaların* problemin çözümü için kullanılmasını sınırlandırmaktadır. Çünkü kesin algoritmalarda çözüm uzayının her bir noktasının taranması gerektiğinden, bu algoritmalar hiçbir zaman NP-Zor problemlere polinom zamanlarda çözüm üretmeyi gerçekleştiremezler. Küçük boyutlu problemler hariç gerçek hayattaki problemlerin gelişmiş bilgisayarlarda dahi kesin yöntemlerle çözülmesi çok uzun sürelerde gerçekleşmektedir. *Yaklaşık algoritmalar* ise en iyi çözümü garanti etmezler, ancak hesaplama süreleri polinom zamanlıdır. Bu nedenle ARP için daha çok, kısa sürede eniyi çözüme yakın sonuçlar veren sezgisel algoritmalar geliştirilmiştir. Ancak çözüm uzayında yerel eniyi noktaların fazla olması ve arama sırasında bu noktalardan kurtulmanın zor olması *klasik sezgisel algoritmaların*, KEP'ye kaliteli çözümler bulmasını zorlaştırmaktadır. Bu sebeple ARP'nin çözümünde, çözüm uzayını etkin şekilde arayarak eniyiye oldukça yakın

çözümlere kısa sürelerde ulaşmayı başaran *metasezgisel yöntemler* son yıllarda büyük önem kazanmıştır [Cordeau ve ark., 2002].

ARP'nin çözümünde kullanılan yaklaşımların kronolojik sırası Çizelge 2.1'de verilmiştir. 1960 ve 1970'lerde; rota kurma, rota geliştirme ve iki aşamalı sezgiseller kullanılırken, 1980'lerde matematik programlama esaslı sezgiseller geliştirilmiştir. Bu sezgiseller yüksek kaliteli sonuçlar bulmakta, ancak fazla hesaplama zamanına gereksinim duymaktadırlar. 1980'lerin sonuna gelindiğinde yaklaşık 50 noktalı problemler için eniyi çözümler bulunmuştur. 1990'larda ise genel amaçlı metasezgisellere yönelme başlanmıştır.

Çizelge 2.1. Araç rotalamanın kısa bir tarihçesi [Şeker, 2007]

1950'ler:	ARP tamsayılı programlama olarak formüle edilmiş ve 10-20 müşterili küçük problemler çözülmüştür.
1960'lar:	Rota kurma sezgiselleri sunulmuş ve 30-100 müşterili problemler çözülmüştür.
1970'ler:	İki fazlı sezgiseller, interaktif (insan-makine) sezgiseller geliştirilmiş, yaklaşık 50 müşterili problemler eniyileme metotları ile çözülebilir hale gelmiştir.
1980'ler:	Matematiksel programlama esaslı prosedürler literatüre sunulmuştur. Etkileşimli (interaktif: insan-makine) sezgiseller geliştirilmiştir. Eniyileme yöntemleri kullanılarak yaklaşık 50 müşteriye sahip bazı problemler çözülmüştür.
1990'lar:	ARP'ye metasezgiseller uygulanmıştır. 50–100 müşteriye sahip bazı problemler için eniyi çözümler bulunmuştur.

ARP çözüm yöntemleri; kesin yöntemler ve sezgisel algoritmaları içeren yaklaşık yöntemler olmak üzere iki ana gruba ayrılabilir. Geliştirilen klasik sezgisel algoritmalar genel olarak yapısal, iyileştirmeli ve çok aşamalı olmak üzere üç tiptir. Son yıllarda hızlı bir yükselişe geçen çözüm yöntemi ise, yine en iyiyi garanti etmeyip yaklaşık çözümler bulan metasezgisellerdir.

2.2.1. Kesin yöntemler

Kesin yöntemler, problemin eniyi çözümünü bulan yöntemlerdir. Ancak kesin yöntemler, problem boyutu büyüdükçe ya da probleme başka kısıtlar eklendikçe

çözüm süresinin artmasıyla birlikte makul sürelerde iyi çözümler bulamamaktadır. Tamsayılı model olarak formüle edilen ARP'nin çözümü için dal-sınır (branch and bound), kesme düzlemi (cutting plane) ve dal-kesme (branch and cut) yöntemleri kullanılmaktadır. Ayrıca dinamik programlama, lagranj ayrıştırma (lagrangean relaxation), ağaç arama ve sütun yaratma (column generation) da kesin çözüm yöntemleri arasında yer almaktadır.

2.2.2. Sezgisel yöntemler

ARP'nin çözümünde iyi sonuçlara kısa sürelerde ulaşmak için genellikle sezgisel yöntemler kullanılmaktadır. Sezgisel algoritmalar, problemin çözüm uzayı çok büyük olduğunda çözümün aranmasını sınırlamak üzere herhangi kural, strateji, sadeleştirme gibi alternatif hareketlerden etkili olanını kullanmak amacıyla tanımlanan kriterler veya bilgisayar yöntemleri olarak tanımlanabilir. Sezgisel yöntemler arama uzayında oldukça kısıtlı bir alanda tarama yaparak, kısa sürede iyiye yakın çözümler üretmeyi amaçlarlar. Dolayısıyla bu tür algoritmalar yakınsama özelliğine sahip olsa da kesin çözümü garanti etmezler ve sadece eniyi çözümün yakınındaki bir çözümü garanti edebilirler [Erol, 2006].

ARP için geliştirilen klasik sezgisel algoritmalar genel olarak üç ana sınıfa ayrılmaktadır [Eryavuz ve Gencer, 2001] ve bunlar aşağıda anlatılmıştır:

- Tur kurucu sezgiseller
- Tur iyileştirici sezgiseller
- İki aşamalı metotlar

Tur kurucu sezgiseller

Bu yöntemlerde, genellikle her müşteri depoya atanarak basit turlar oluşturulur. Algoritmanın her aşamasında turlar araç kapasitesi dikkate alınarak birleştirilmekte uygun çözüm elde edilmektedir. Tur kurucu metotlar arasında en çok tercih edileni

Clarke ve Wright'ın (1964), Dantzig ve Ramser'in çalışmasından esinlenerek geliştirdikleri tasarruf algoritmasıdır. Tasarruf algoritmasının hem paralel hem de sıralı olmak iki versiyonu vardır. Bu algorithmadan yola çıkarak sayısız algoritma geliştirilmiştir [Eryavuz ve Gencer, 2001].

Tur iyileştirici sezgiseller

Bu yöntemlerin tur kurucu sezgisellerden en büyük farkı, uygun bir çözümle aramaya başlamalarıdır. Her iterasyonda dal kombinasyonlarında sınırlı değişiklikler yapılarak mevcut çözümden daha iyi amaç fonksiyonu değerine sahip başka bir çözüm elde edilmeye çalışılır, bulunduğu ise mevcut çözüm iyileştirilmiş olur. Bu şekildeki bir algoritma, en iyileme literatüründe YA algoritması olarak tanımlanmaktadır. Diğer bir ifade ile tur iyileştirici sezgiseller aslında birer YA algoritmalarıdır. YA'yı uygulayabilmek için öncelikle *komşuluk yapısının* tanımlanması gerekmektedir. Arama uzayında bir nokta olan s çözümünün komşuluk yapısı $N(s)$ ile gösterilir. $N(s)$ mevcut s çözümünden, tanımlanan bir hareketle ulaşılacak ve bir takım ölçülere göre s noktasının yakınında yer alan tüm çözümleri içeren kümedir. YA'nın her iterasyonunda mevcut çözüm, kendisinden daha iyi olan bir komşu çözüm ile yer değiştirilmektedir.

İki aşamalı sezgiseller

KKARP için geliştirilen çok aşamalı sezgiseller; müşterilerin kapasite kısıtına uygun gruplara ayrılması ve her grup içinde rotalama işleminin yapılması olmak üzere iki adımdan oluşmaktadır. Bu adımlardan hangisinin önce gerçekleştirildiğine göre, “önce grupta sonra rotala” veya “önce rotala sonra grupta” olmak üzere iki tipi vardır. Genellikle ikinci adımda yapılan işlem, ilk adımda yapılan işleme geri beslemeler ile sürdürülmektedir [Erol V., 2006].

GSP kapasitenin sonsuz olduğu bir KARP olduğu için, önce rotala sonra grupta yöntemlerinin ilk aşamasında oluşturulacak büyük tur GSP'yi çözen tur kurucu

sezgisellerle elde edilebilir. Ulusoy'un (1985) kapasiteli dal rotalama problemi (GSP'nin özel bir tipi) ile ilgili çalışmasında büyük bir turu kapasite kısıtına göre alt turlara bölmek için kullandığı yöntem, literatürde ARP'nin çözümünde oldukça fazla yer bulmuştur. Çünkü, ARP için geliştirilen genetik algoritmaların çoğunda çözümler tek bir müşteri dizisiyle (permütasyon kodlama) gösterildiğinden, bu dizinin her seferinde kısıtlara uygun rotalara ayrılması gerekmektedir. Bu tez çalışmasında ETD_ARP için geliştirilen algoritmalar, Ulusoy'un sözü edilen algoritmasından yararlanılmıştır.

2.2.3. Metasezgisel yöntemler

ETD_ARP'nin çözümü için bu tez kapsamında GA_DKİ ve KSE_DKİ algoritmaları geliştirilmiştir. Algoritmaların DKİ yöntemi ile birleştirilmesiyle hem yığın hem de YA stratejilerinin birlikte kullanılması sağlanmıştır. Son yıllarda ARP için araştırmacılar tarafından önerilen metasezgisel algoritmaların çoğu, iki sezgiselin avantajından birarada yararlanmayı amaçlayan *melez* algoritmalar. Özellikle ETD_ARP'nin çözümü için yığın tabanlı metasezgiseller YA'ya dayalı diğer sezgisellere nazaran çok iyi performans sergileyemediklerinden dolayı, literatürde pek fazla yer bulamamışlardır. Bu tez kapsamında önerilen GA_DKİ ve KSE_DKİ melez algoritmaları ise, literatürdeki bu boşluğu doldurabilmek amacıyla geliştirilmiştir.

3. EŞZAMANLI TOPLA-DAĞIT ARAÇ ROTALAMA PROBLEMİ

KARP'nin genelleştirilmiş bir hali olan TD_ARP, tesislerden müşterilere yapılan dağıtım işlemi ile müşterilerden tesislere yapılan toplama işleminin aynı araçlarla gerçekleştirildiği rotalama problemi olarak tanımlanmaktadır. Gerçek hayatta sağlık, gıda, otomotiv gibi birçok sektörde uygulama alanı olan TD_ARP, son yıllarda araştırmacıların üzerinde çeşitli çalışmalar yaptığı bir problem türü olmuştur. Kanların kan merkezlerinden hastanelere taşınması esnasında aynı zamanda hastanelerden kanların toplanması, yedek parçaların bayiliklere dağıtılırken bayiliklerden de kullanılmış parçaların geri dönüşüm için toplanması, gıdaların marketlere dağıtımı esnasında günü geçmiş ürünlerin toplanması, TD_ARP'nin günlük hayattaki uygulamalarından bazılarıdır.

TD_ARP en genel hali ile şu şekilde tanımlanabilir: $G = (N, A)$ tam bağlı (bütün düğümler arasında doğrudan bir ayırıtın mevcut olduğu durum) bir graf olsun. N düğümler kümesi ve A ise bu düğümler arasında tanımlanan ayırıtlar kümesidir, $A = \{(i, j) | i \neq j \text{ ve } i, j \in N\}$. Düğüm kümesindeki ilk düğüm depoyu, diğer düğümler müşterileri temsil etmektedir. Bu grafta c_{ij} , (i, j) ayırıtının uzunluğunu (maliyetini) göstermektedir ve üçgensel eşitsizlik $(c_{ij} \leq c_{ik} + c_{kj})$ sağlanmaktadır. Taşımayı gerçekleştirecek araçlar depoda hazır olarak beklemektedir ve araç kapasiteleri (Q) özdeşdir. TD_ARP, tanımlanan bu sistemde aşağıdaki kısıtları sağlayan enküçük maliyetli rotaların bulunması problemidir [Karaoğlu, 2009].

1. Her müşteriye kesinlikle bir kez uğranılmalı,
2. Bir rota depodan başlamalı ve tekrar depoda son bulmalı,
3. Rota üzerinde, aracın topladığı ve dağıtacağı yük miktarı toplamı araç kapasitesini geçmemelidir.

TD_ARP için dikkate alınan varsayıma göre müşterilerden toplanan ürünlerin doğrudan diğer bir müşteriye taşınması söz konusu değildir. Yani bütün talepler ya

depodan müşteriye ulaştırılmakta ya da müşteriden depoya taşınmaktadır. [Nagy ve Salhi, 2005]. Bu varsayımlar altında TD_ARP'nin üç farklı tipi bulunmaktadır. Bunlar;

1. *Önce Dağıt Sonra Topla ARP (ÖDST_ARP)*: ÖDST_ARP müşterilerin dağıtım (*linehaul*) ve toplama (*backhaul*) müşterileri olmak üzere iki guruba ayrıldığı ve rota üzerinde dağıtım müşterilerine toplama müşterilerinden önce hizmet verildiği problem tipidir. Genelde, araç içerisinde dağıtılacak ürünler ile toplanan ürünlerin tekrar yerleştirilmesinin mümkün olmadığı durumlarda bu tür bir varsayıma ihtiyaç duyulur. Çünkü; bu durumda dağıtım müşterilerinin talepleri ile yüklenen araç, ancak bütün ürünler dağıtılıp boşaldıktan sonra toplama müşterilerine hizmet verebilir. Gıda sektöründe araçların ana depodan marketlere sebze-meyve dağıttıktan sonra, üreticilerden gelen ürünleri depoya taşımaları [Ropke ve Pissing, 2006] bu problem tipine örnek olarak verilebilir.
2. *Karma Topla-Dağıt ARP (KTD_ARP)*: ÖDST-ARP'de müşteri öncelikleriyle ilgili yapılan varsayımın kaldırılmasıyla elde edilen problem tipidir. Başka bir ifade ile, rota üzerinde toplama ve dağıtım müşterilerine karışık sırada hizmet verilebilmektedir. Dolayısıyla bu problem, araç içerisinde yükleme ve boşaltmanın birlikte yapılmasının mümkün olduğu durumlar için geçerlidir. Her iki yanında da kapısı olan araçlarla bu mümkün olmaktadır. Hizmet sektöründe, depodan müşterilere kargolar dağıtılırken, diğer müşterilerden depoya geri getirilmek üzere kargoların toplanması bu problem tipine örnek olarak verilebilir.
3. *Eşzamanlı Topla-Dağıt ARP*: ETD_ARP probleminde ÖDST_ARP ve KTD_ARP'den farklı olarak, bir müşteri aynı anda hem toplama hem de dağıtım müşterisi olabilmektedir. Aracın müşteriye uğradığında, hem dağıtılacak ürünü bırakıp hem de müşteriden depoya götürülecek ürünü toplayarak iki işlemi bir arada (önce verilecek ürünü bırakır sonra toplanacak ürünü alır) yaptığı problem tipidir. Gıda sektöründe, marketlere içeceklerin dağıtıldıktan sonra aynı marketten boş şişelerin geri dönüşüm amaçlı toplanması bu problem tipine örnek olarak verilebilir [Karaoğlu, 2009].

Tanımlarından da anlaşılabacağı gibi ETD_arp, ÖDST_arp ve KTD_arp'ın genel halidir. Dolayısıyla, ETD_arp için geliştirilen bir model, doğrudan ya da küçük değişiklikler ile ÖDST_arp ve KTD_arp için kullanılabilir.

Aşağıda ETD_arp'nin Karaoğlu (2010) tarafından önerilen, araç sayısından bağımsız (iki indisli) 0–1 karar değişkenlerinin kullanıldığı matematiksel model yer almaktadır [Karaoğlu, 2010].

Dizin kümeleri ve parametreler

N ;	tüm düğümlerin kümesi $(0, \dots, n)$,
N_0 ;	müşteriler kümesi $(1, \dots, n)$,
c_{ij} ;	i müşterisinden j müşterisine geçiş maliyeti,
p_i ;	i müşterisinin dağıtım talebi,
q_i ;	i müşterisinin toplama talebi,
Q ;	araç kapasitesi

Karar değişkenleri

$$x_{ij}; \quad \begin{cases} 1 & (i,j) \text{ ayrıtı herhangi bir tur üzerinde ise } (\forall i,j \in N) \\ 0 & \text{dd} \end{cases}$$

U_i ; i düğümüne girmeden hemen önce araçtaki dağıtılacak ürün miktarı $(\forall i \in N_0)$

B_i ; i düğümünün çıkışında araçtaki toplanan ürün miktarı $(\forall i \in N_0)$

$|R|$; rota sayısı,

Model

$$\text{Min} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (3.1)$$

Kısıtlar

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N_0 \quad (3.2)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N_0 \quad (3.3)$$

$$\sum_{i \in N} x_{i0} = |R| \quad (3.4)$$

$$\sum_{i \in N} x_{0j} = |R| \quad (3.5)$$

$$U_j - U_i + Qx_{ij} + (Q - d_j - d_i)x_{ji} \leq Q - d_i \quad \forall i, j \in N_0, d_i + d_j \leq Q, i \neq j \quad (3.6)$$

$$B_j - B_i + Qx_{ij} + (Q - q_j - q_i)x_{ji} \leq Q - q_j \quad \forall i, j \in N_0, q_i + q_j \leq Q, i \neq j \quad (3.7)$$

$$U_i + B_i - d_i \leq Q \quad \forall i \in N_0 \quad (3.8)$$

$$U_i \geq d_i + \sum_{j \in N_0, j \neq i} d_j x_{ij} \quad \forall i \in N_0 \quad (3.9)$$

$$U_i + (Q - d_i)x_{i0} \leq Q \quad \forall i \in N_0 \quad (3.10)$$

$$B_i \geq q_i + \sum_{j \in N_0, j \neq i} q_j x_{ji} \quad \forall i \in N_0 \quad (3.11)$$

$$B_i + (Q - q_i)x_{0i} \leq Q \quad \forall i \in N_0 \quad (3.12)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (3.13)$$

$$U_i \geq 0 \quad \forall i \in N_0 \quad (3.14)$$

$$B_i \geq 0 \quad \forall i \in N_0 \quad (3.15)$$

$$|R| \geq 0 \quad (3.16)$$

Matematiksel modelde amaç (Eş. 3.1) toplam taşıma maliyetinin en küçüklenmesidir. Eş 3.2–Eş. 3.5 arasındaki kısıtlar ARP için geliştirilmiş genel atama kısıtlarıdır ve bir müşteriye mutlaka bir kere servis verilmesini garantilemekte, buna ek olarak düğümlerin girdi çıktı dengesini sağlamaktadır. Eş. 3.6 ve Eş. 3.7 sırasıyla bir rota üzerinde dağıtım ve toplama taleplerinin toplamlarının ayrı ayrı olarak kapasiteyi geçmemesini sağlayan ve alt turların oluşmasını engelleyen kısıtlardır. Eş. 3.9 ile gösterilen kısıt her düğümden çıkıştan sonra aracın yükünün toplam kapasiteyi

geçmemesini sağlamaktadır. Eş. 3.9-Eş. 3.12 U ve B karar değişkenlerinin alt ve üst sınırlarını belirleyen kısıtlardır. Eş. 3.13-Eş. 3.16 ise işaret kısıtlardır.

3.1. ETD_ARP için Literatür Araştırması

ETD_ARP ilk olarak 1989 yılında Min tarafından, halk kütüphanesine ait kitapların dağıtımı ve toplanmasıyla ilgili gerçek bir hayat problemini çözmek üzere ortaya konulmuştur [Min, 1989]. Min problemi çözmek için bir matematiksel model ve önce grupla sonra rotala prensibine dayalı sezgisel bir algoritma önermiştir. Sezgisel algoritmada öncelikle müşteriler iki ayrık kümede gruplandırılmış, ardından her bir grup için GSP çözülmüş ve uygun çözüm elde edilene kadar araç kapasitesini aşan arklar cezalandırılarak GSP'nin çözümü sürdürülmüştür.

90'lar boyunca ETD-ARP ile ilgili pek fazla çalışma yapılmamıştır. Bu yıllarda Halse, problem için önce grupla sonra rotala yapısında iki aşamalı bir algoritma [Halse, 1992], Salhi ve Nagy ise ekleme yöntemine dayalı bir sezgisel [Salhi ve Nagy, 1999] önermiştir. Salhi ve Nagy, bu sezgiseli problemin çok depolu hali için de uygulamıştır.

2000'li yıllarda, pratikte toplama faaliyetleriyle daha çok karşılaşılmasıyla birlikte probleme ilgi de giderek artmıştır. Dethloff matematiksel bir model ve en ucuz ekleme, artık kapasite gibi dört farklı kritere göre ekleme stratejisini uygulayan tur kurucu bir sezgisel geliştirmiştir [Dethloff, 2001]. Bu çalışmasında ayrıca, kullandığı ekleme kriterlerinin algoritmanın performansı üzerindeki etkilerini ve problemin diğer ARP çeşitleriyle ilişkisini incelemiştir. Bir diğer matematiksel model ve eklemeye dayalı sezgisel de Nagy ve Salhi tarafından önerilmiştir [Nagy ve Salhi, 2005]. Bu sezgiselde, ARP'de uygun olmayan rotaların uygun rotalara dönüştürülmesi için önerilen yaklaşımlar ETD_ARP'de kullanılmıştır. Tang ve Galvao ise KARP için geliştirilen tur parçalama [Beasley, 1983] ve süpürme algoritmalarını ETD_ARP'ye uyarlamışlardır [Tang ve Galvao, 2002].

Son yıllarda yapılan çalışmalar incelendiğinde ise, ETD_ARP'nin çözümü için daha çok metasezgisel algoritmaların kullanıldığı görülmektedir [Karaoğlu, 2009]. Ropke ve Pisinger ETD-ARP'nin yanı sıra ARP'nin bir çok çeşidini çözmek için büyük komşuluk arama (large neighborhood search) sezgiselini geliştirmişlerdir [Ropke ve Pisinger, 2006]. Tabu arama (TA) algoritmasını Crispim ve Brandao DKİ algoritması ile [Crispim ve Brandao, 2005], Chen ise Tavlama Benzetimi (TB) [Chen, 2006] ile birleştirerek ETD_ARP'yi çözmek için melez algoritmalar elde etmişlerdir. Chen'in çalışmasında TB melez algoritmanın ana adımlarını oluşturmakta ve her sıcaklık değerinde kabul edilen çözüm tabu listesinde tutulmaktadır. Ayrıca bu algorithmada kullanılan komşuluk yapısı ardışık olarak değiştirilmiştir. Chen ve Wu, 2006 yılındaki bir diğer çalışmalarında öncekinden farklı olarak TB yerine Kayıttan Kayıda (KK) [Chen ve Wu, 2006] algoritmasını kullanmışlardır. TB'nin KK algoritması ile arasındaki tek fark; TB'de yeni çözümün kabul edilmesi belli bir olasılığa göre gerçekleşirken, KK algoritmasında o ana kadar elde edilmiş en iyi çözüme bağlı olarak belirlenen bir eşik değeri ile yapılan deterministik karşılaştırma sonucu belirlenmesidir. Montane ve Galvao tarafından, 1-1 yer değiştirme, 1-0 yer değiştirme, çaprazlama ve 2-opt olmak üzere dört farklı komşuluk yapısıyla birlikte uzun dönemli hafıza kullanan TA algoritması önerilmiştir [Montane ve Galvao, 2006]. Önerilen algoritmanın etkinliğini ölçmek için çözümlerini, geliştirdikleri matematiksel modelin 2 saat çalıştırılmasıyla elde ettikleri üst sınırlar ile karşılaştırmışlardır. Bianchessi ve Righini çalışmalarında, değişik çözüm kurucu ve yerel arama algoritmalarının yanı sıra düğüm değiştirme ve hat değiştirmeye dayalı değişen komşuluk yapısını kullanan bir TA algoritması önererek, algoritmanın performansını karşılaştırmalı olarak incelemişlerdir [Bianchessi ve Righini, 2007]. Wassan ve ark. bir noktanın yerleştirilmesi, iki noktanın farklı turlar arasına yer değiştirmesi ve rotanın yönünün ters çevrilmesi hareketlerinin oluşturduğu komşuluk yapılarını kullanan tepkisel TA algoritmasını geliştirmişlerdir [Wassan ve ark., 2007]. Tepkisel TA'da çeşitlendirme ve yoğunlaşma stratejileri arasında denge kurabilmek amacıyla, tabu süresi dinamik olarak değiştirilmiştir. Görüldüğü gibi ETD_ARP için geliştirilen metasezgisellerin çoğu TA algoritmasının saf veya melez versiyonlarıdır. Bunun nedeni, TA'nın bu problem türü için etkin bir algoritma olmasıdır.

ETD_ARP ile ilgili arařtırmalar 2009 yılında da sürmüřtür. En son yapılan alıřmalarda daha ok, bu problem iin bugüne kadar daha az uygulanmıř metasezgiseller kullanılmıřtır. Zachariadis ve ark., problemi TA ile rehberli YA'yı birleřtiren melez bir algoritma ile özmüřlerdir [Zachariadis ve ark., 2009a]. Ai ve Kachitvichyanukul tarafından KSE'ye [Jin Ai ve Kachitvichyanukul, 2009], Gajpal ve Apad tarafından ise karınca kolonisi eniyilemeye (KKE) [Gajpal ve Apad, 2009] dayalı birer algoritma geliřtirilmiřtir. KKE algoritması; (i) en yakın komřu sezgiseli kullanılarak elde edilen bařlangı özümüne göre yol yoğunluęu ve parametrelere bařlangı deęerlerinin belirlenmesi (ii) yol yoğunluklarına göre her karınca iin özümün elde edilip, yerel aramanın yapılması, ardından da yoğunlukların ve elit karıncaların güncellenmesi olmak üzere iki ana adımdan oluřmaktadır. Zachariadis ve ark., ayrıca Uyarlamalı Hafıza Metodolojisi (UHM) ve TA'ya dayalı bařka bir melez algoritma da geliřtirmiřlerdir [Zachariadis ve ark., 2009b]. Problemlle ilgili yapılan en son alıřma ise Subramanian ve ark. tarafından geliřtirilen paralel bir algoritmadır [Subramanian ve ark, 2009]. Bu algoritmada iteratif olarak yerel arama yapılmakta ve kullanılan komřuluk yapıları da rassal olarak deęiřtirilmektedir. Paralel bir yöntem olması nedeniyle kısa sürelerde özüme ulařması, algoritmanın en önemli özellięidir.

ETD-ARP ile ilgilenen iki yüksek lisans tezinde de, problemin özümü iin metasezgisel yöntemler geliřtirilmiřtir. Vural'ın alıřması, özümün Bean [Bean, 1994] tarafından önerilen rassal anahtarlama yaklařımı kullanılarak gösterildięi, iki farklı GA'yı iermektedir [Vural, 2003]. Performansının pek iyi olmadıęı belirtilen ilk algoritmada basit genetik operatörler kullanılmıřtır. İkincisinde ise izelgeleme problemi iin önerilen bir aprazlama operatörü, üç farklı mutasyon operatörü ve turların iyileřtirilmesi ařamasında da or-opt sezgiseli kullanılmıřtır. Göke'nin yüksek lisans tezinde ise, ETD-ARP ile birlikte eřitli ARP türleri iin KKE'ye dayalı bir algoritma geliřtirilmiřtir [Göke, 2004]. Bu algoritmada özümmler rota ii 2-opt sezgiseli kullanılarak iyileřtirilmiřtir.

Literatürde ETD-ARP iin Min [Min, 1989], Dethloff [Dethloff, 2001], Nagy ve Salhi [Nagy ve Salhi, 2005], Montane ve Galvao [Montane ve Galvao, 2006],

Dell'Amico ve ark. [Dell'Amico ve ark., 2006] ile Ai ve Kachitvicanukul [Ai ve Kachitvicanukul, 2009] tarafından matematiksel modeller geliştirilmiştir.

Çizelge 3.1 bu bölümde adı geçen ETD-ARP ile ilgili sezgisel yöntemleri genel özellikleriyle kronolojik olarak özetlemektedir. Görüldüğü gibi probleme olan ilgi giderek artmış ve çözümü için genellikle metasezgiseller kullanılmıştır. Bu metasezgisellerin çoğu ise TA'ya dayalı olup KSE, KKE ve GA'yı kullanan yalnızca bir veya iki algoritma geliştirilmiştir.

Çizelge 3.1. ETD-ARP ile ilgili sezgisel çalışmalar [Subramanian ve ark.,2009]

Yazarlar	Yıl	Yaklaşım
Min	1989	İlk çalışma
Halse	1992	Önce grupla sonra rotala stratejisi 3-opt prosedürü
Salhi ve Nagy	1999	Ekleme tabanlı sezgisel
Dethloff	2001	Dört farklı kritere göre ekleme tabanlı bir tur geliştirici sezgisel
Tang ve Galvao	2002	Tur parçalama ve süpürme algoritmaları probleme uyarlanmış
Vural	2002	GA
Gökçe	2003	Karınca Kolonisi
Salhi ve Nagy	2005	Farklı fizibilite seviyelerindeki sezgiseller
Crispim ve Brandao	2005	TA + DKİ
Ropke ve Pisinger	2006	Büyük Komşuluk Arama
Chen	2006	TB + TA
Chen ve Wu	2006	KK + tabu listesi
Montane ve Galvao	2006	Uzun dönemli hafıza kullanan TA
Bianchessi ve Righini	2007	Tur geliştirme ve yerel arama sezgiselleri TA + DKİ
Wassan ve ark.	2007	Tepkisel TA
Zachariadis ve ark.	2009	TA + rehberli YA
Ai ve Kachitvichyanukul	2009	KSE
Gajpal ve Apad	2009	KKE
Zachariadis ve ark.	2009	Uyarlamalı Hafıza Metodolojisi + TA
Subramanian ve ark.	2009	Paralel İteratif Yerel Arama

4. ETD_ARP İÇİN GELİŞTİRİLEN SEZGİSEL YAKLAŞIMLAR

Literatürde ETD_ARP'nin çözümü için geliştirilen metasezgisel yöntemlerin çoğu, bu tarz problemlere iyi sonuçlar üreten YA tabanlı algoritmalar. GA ve KSE gibi yığın tabanlı metasezgisellerin ETD_ARP'ye uygulamaları oldukça sınırlıdır. Bu tez kapsamında ETD_ARP için literatürde ilk kez DKİ ile melezleştirilmiş GA ve kesikli KSE (KKSE) metasezgisellerine dayalı melez algoritmalar (GA_DKİ ve KSE_DKİ) önerilmiştir. Ayrıca, ETD_ARP'nin KKSE ile çözülmesi ve bu algoritmanın hem DKİ, hem de TB ile melezleştirilmesi de literatürde bir ilktir. Önerilen algoritmaların performansı literatürdeki test problemleri üzerinde incelenmiştir.

Bu bölümde, önerilen sezgisel yöntemler detaylarıyla açıklanacaktır.

4.1. Önerilen Genetik Algoritma

Tez kapsamında ETD_ARP için geliştirilen ilk algoritma GA ve DKİ'e dayalı GA_DKİ algoritmasıdır. Bu melez algoritmada; çözüm uzayının iyi çözümlerin bulunduğu bölgelerine ulaşmak amacıyla GA, bu bölgelerde yoğun olarak arama yapmak içinse DKİ algoritması kullanılmıştır.

GA biyolojik sistemlerin gelişim sürecini taklit ederek, ilgilenilen probleme çözüm arayan bir metasezgiseldir. GA'nın ilk aşamasında, tüm mümkün çözümlerin alt kümesinden oluşan bir *başlangıç yığını* elde edilir. Yığının her elemanı (*birey*) çözümü ifade edecek şekilde kodlanır. Her çözüm biyolojik olarak bir *kromozoma* eşdeğerdir. GA'nın herhangi bir adımındaki çözümlerin bir kümesi ise, *yığın* olarak adlandırılmaktadır. Yığındaki her bireyin bir *uygunluk değeri* (*fitness value*) vardır. Hangi birey ya da bireylerin bir sonraki yığına taşınacağı, bu uygunluk değerlerine göre belirlenir. Kromozomun uygunluk değeri, problemin amaç fonksiyonu değerine dayalı olarak hesaplanmaktadır. İyi bir kromozom yani çözüm; problem enbüyükleme problemi ise yüksek, enküçükleme problemi ise düşük uygunluk değerine sahip olmalıdır.

Bir GA'nın her bir iterasyonu; yeniden üretim işlemi, genetik operatörlerin uygulanması ve uygunluk değerlerinin hesaplanması olmak üzere üç ana adımdan oluşmaktadır. *Yeniden üretim*, mevcut yığından gelecek yığına taşınacak kromozomların belirlenerek bir sonraki popülasyona kopyalanmasıdır. Algoritmada her yeni yığının bir öncekinden, daha uygun çözümlerden oluşması beklenir. Bu nedenle uygunluk değeri yüksek olan bireylerin yeni yığında bulunma şansı daha yüksektir. *Genetik operatörler*, mevcut yığından seçilen kromozomlara uygulanan *çaprazlama* ve *mutasyon* operatörleridir. Bu operatörler ile genetik bilgi kullanılarak mevcut yığından yeni kromozom veya kromozomlar oluşturulur. Çaprazlama operatörü farklı kromozomlar arasında bilgi değiş tokuşu yaparak yeni çözümleri (*çocuk*) elde ederken, mutasyon operatörü de kromozomun bir kısmını rassal olarak değiştirerek çözüm uzayının farklı bölgelerindeki çözümlere ulaşılmasını sağlamaktadır. *Uygunluk Değeri*, kromozomların değerlendirilerek yeni yığını oluşturacak kromozomların belirlenmesinde kullanılan bir araçtır. Bu nedenle, bir sonraki yığında bulunup bulunmayacağına karar vermek için yeni kromozomların uygunluk değerleri (amaç fonksiyonu değeri) hesaplanmalıdır. GA'da her zaman özel genetik yapıların nesiller boyu taşınması amaçlanmaktadır.

Bu bölümde; GA_DKİ'de kullanılan çözüm gösterimi, başlangıç yığını, seçim mekanizması, genetik operatörler ve DKİ algoritması detaylarıyla anlatılmaktadır.

4.1.1. Çözüm gösterimi

GA'nın performansı, çözümleri göstermek amacıyla kullanılan kodlama yapısına ve bu kodlama yapısı için kullanılan genetik operatörlere bağlıdır. Bu nedenle, öncelikle problem için uygun bir kodlama yapısı ya literatürde var olan kodlama yapıları arasından seçilmeli ya da geliştirilmelidir.

ETD_ARP ve ARP için GA'nın kullanıldığı çalışmalar incelendiğinde üç farklı kodlama yapısının kullanıldığı görülmektedir. Bunlardan ilk ikisi müşterilerin sıralanmasına dayalı olan kodlama tipidir. Bu kodlama tipi, "*permütasyon kodlama*" olarak adlandırılır. Permütasyon kodlamanın birinci tipinde her bir tur depo ile bir

diğerinden ayrılmaktadır. Van Breedam (1995), ARP için geliştirdiği GA'da bu kodlama tipini kullanmıştır [Van Breedam, 1995]. Permütasyon kodlamanın ikinci tipinde ise, tüm müşterilerden oluşan büyük bir tur elde edilir ve özel bir ayırma algoritması kullanılarak bu büyük turdan, araç kapasitelerini dikkate alan alt turlar elde edilir. Bu tür kodlama ARP için ilk kez, Prins (2004) tarafından geliştirilen GA'da kullanılmıştır [Prins, 2004]. Prins bu çalışmasında kod çözme (decoding) prosedürü olarak; ilk kez Ulusoy tarafından (1987) kapasiteli ark rotalama probleminde büyük bir turu uygun rotalara bölmek için uygulanan ayırma algoritmasını kullanmıştır. Bean ise (1994), söz konusu problemler için kullanılabilecek “*rassal anahtarlama*” olarak adlandırılan bir kodlama yapısı geliştirmiştir [Bean, 1994]. Rassal anahtarlama, her müşteriye tanımlanan bir aralıkta (genellikle 0 ile 1 arasında) bir rassal sayı atanır ve bu rassal sayılar tercihe göre küçükten büyüğe veya büyükten küçüğe sıralanarak müşterilerin ziyaret sırası elde edilir. Bu kodlama yapısının en önemli özelliği, tek-noktalı, iki-noktalı ve çok-noktalı gibi basit çaprazlama operatörleri ile uygun çözümlerin elde edilmesidir.

Genetik operatörlerin kolay uygulanması, uygun rotalara bölünebilmesi ve yapılan ön denemelerde iyi çözümler üretmesi nedeniyle GA_DKİ'de çözümler permütasyon kodlamanın ikinci tipiyle gösterilmiştir. Şekil 4.1'de 10 müşteriden oluşan bir ETD_ARP için permütasyon kodlamanın kullanıldığı bir çözüm gösterimi yer almaktadır.

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

Şekil 4.1. 10 müşterili ETD_ARP için bir kromozom örneği

Permütasyon kodlama ile çözüm genetik operatörlerin uygulanmasında ve genetik bilginin düzgün taşınmasında bazı kolaylıklar sağlayacak şekilde tüm müşterilerin içinde bulunduğu büyük bir tur sırası olarak gösterilir. Ancak bu büyük turun kapasite kısıtına uygun rotalara bölünmesi için bir ayırma işlemi gereklidir. Permütasyon kodlamanın gösterdiği büyük turun kapasite kısıtına uygun rotalara ayrılmasında, Prins'in 2004 yılında KARP için geliştirdiği GA'da kullandığı *ayırma*

algoritmasından yararlanılmıştır [Prins, 2004]. Tez kapsamında bu ayırma algoritması ilk defa ETD_ARP için uyarlanmıştır. Büyük turun rotalara ayrılması için kullanılan prosedür aşağıda ayrıntılı şekilde anlatılmaktadır.

Ayrma işleminde öncelikle ETD_ARP'nin bir $s = \{s_1, s_2, \dots, s_{|N|}\}$ çözümü için yardımcı $H = (Y, A)$ şebekesi elde edilir. Problemdeki müşteri sayısı $|N|$ olmak üzere bu şebekedeki düğüm sayısı $|N| + 1$ 'dir. Y düğümler kümesindeki ilk düğüm depoyu temsil eden aylak (dummy) düğüm olup "0" ile gösterilirken, diğerleri müşterileri gösteren 1 ile $|N|$ arasındaki sayılardan oluşan düğümlerdir.

A hatlar kümesi, aynı rotada hizmet alan ve rota boyunca taşınan en büyük yükün ($eb_yük$) araç kapasitesinden (Q) küçük olduğu alt müşteri sırasını, $[s_{i+1}, s_{i+2}, \dots, s_j] \in s$, gösteren (i, j) , $i < j$, hatlarından oluşmaktadır. Bir s için $(i, j) \in A$ hattı, $[0 - s_{i+1} - s_{i+2} - \dots - s_j - 0]$ rotasına karşılık gelmektedir. Yardımcı grafta (i, j) hattının ağırlığı olan z_{ij} , rotanın maliyetine eşittir ve Eş. 4.1 ile hesaplanır. Bu eşitlikte, c_{ij} , i ve j müşterileri arasındaki mesafe/maliyettir. Ayrma işleminde elde edilen yardımcı şebekedeki, ilk düğümden son düğüme en küçük maliyetli yolun bulunması ile büyük turdan elde edilen rotalar belirlenmiş olur.

$$z_{ij} = c_{(0, s_{i+1})} + \sum_{k=i+1}^{j-1} c_{(s_k, s_{k+1})} + c_{(s_j, 0)} \quad (4.1)$$

Prins (2004) ayırma işlemini iki aşamada gerçekleştirmiştir. Bu algoritmada H şebekesi açık bir şekilde oluşturulmamaktadır. Bunun yerine, algoritmanın ilk aşamasında Y düğümler kümesindeki her düğüm için V_j ve P_j olmak üzere iki etiket değeri hesaplanmaktadır. V_j ; H şebekesindeki 0 düğümünden j düğüme en kısa yolun maliyetini, P_j ise bu yoldaki j düğümünün öncülünü göstermektedir. İlk aşamanın çıktısı olan P vektörü, ikincinin girdisidir. Şekil 4.2'de bu algoritmanın ETD_ARP'ye uyarlanmış hali görülmektedir.

Algoritma: Ayırma işlemi AI Girdi: ETD_ARP'ye ait problem girdileri ve bir s çözümü $s = \{s_1, s_2, \dots, s_{ N }\}$ Çıktı: İlgilenilen s için rotalar kümesi R ve amaç fonksiyonu değeri $f(s) = V_{ N }$
Begin $R \leftarrow \emptyset; V_0 \leftarrow 0; V_j \leftarrow \infty; \text{for } \forall j \in \{1, 2, \dots, N \}$ for $i := 1, 2, \dots, N$ do $k \leftarrow 0; eb_yuk \leftarrow 0; yuk_{\emptyset} \leftarrow 0; yuk_s \leftarrow 0; j \leftarrow i$ repeat $yuk_{\emptyset} \leftarrow eb_yuk + d_{s_j}; \quad yuk_s \leftarrow yuk_s + p_{s_j};$ $eb_yuk \leftarrow \max(yuk_{\emptyset}, yuk_s)$ if $j = i$ then $cost \leftarrow c_{0, s_i} + c_{s_i, 0}$ else $cost \leftarrow cost - c_{s_{j-1}, 0} + c_{s_{j-1}, s_j} + c_{s_j, 0}$ if $eb_yuk \leq Q$ then if $V_{i-1} + cost < V_j$ then $V_j \leftarrow V_{i-1} + cost$ $P_j \leftarrow i - 1$ $j \leftarrow j + 1$ end if until $j > N$ or $eb_yuk > Q$ End for
For $i := 1, 2, \dots, N$ do $route(i) \leftarrow \emptyset; r \leftarrow 0; j \leftarrow N$ repeat $r \leftarrow r + 1; i \leftarrow P_j$ for $k := i + 1$ to j do $route(r) \leftarrow route(r) \cup \{s_k\}$ end for $R \leftarrow R \cup route(r); j \leftarrow i$ Until $i = 0$

Şekil 4.2. ETD_ARP için ayırma algoritması

Algoritmanın *repeat* döngüsünde (Bkz. Şekil 4.2), A kümesiyle belirtilen tüm uygun (feasible) alt rotalar incelenmektedir. İlgilenilen bir (i, j) hattı ile j düğümüne daha az bir maliyetle ulaşılabiliriyorsa; V_j 'ye ilgili maliyet değeri, P_j 'ye ise bu maliyeti veren rotayı tutacak şekilde i değeri ($P_j = i$) atanmaktadır. Bu şekilde her döngüde; mevcut rotaya sıradaki s_j müşterisi eklenerek veya rota boyunca taşınan enbüyük yük miktarının araç kapasitesini geçmesi halinde yeni rota açılarak H üzerindeki tüm uygun rotalar gözden geçirilmektedir. Rotaların enbüyük yük miktarları ise Eş. 4.2, Eş. 4.3 ve Eş. 4.4'den yararlanılarak bulunmaktadır. Eş. 4.2 ve Eş.4.3'deki yuk_o ve yuk_s ; halihazırdaki rotanın sonuna sıradaki s_j müşterisinin eklenmesiyle elde edilen yeni rotada aracın s_j 'ye gelinceye kadar ve s_j 'den sonra taşıdığı enbüyük yük miktarları iken, eb_yuk (Eş. 4.4) s_j müşterisini de içine alan bu yeni rotada taşınan enbüyük yük miktarıdır.

$$yuk_o = eb_yuk + d_{s_j} \quad (4.2)$$

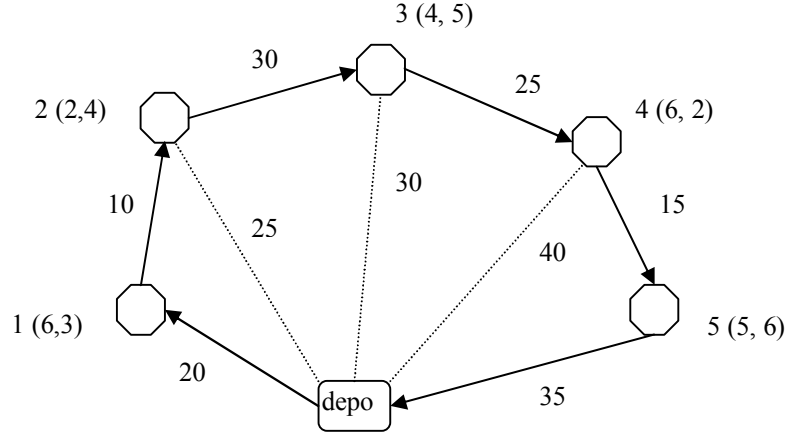
$$yuk_s = \sum_{k=i+1}^r q_{s_j} \quad (4.3)$$

$$eb_yuk = enb(yuk_o, yuk_s) \quad (4.4)$$

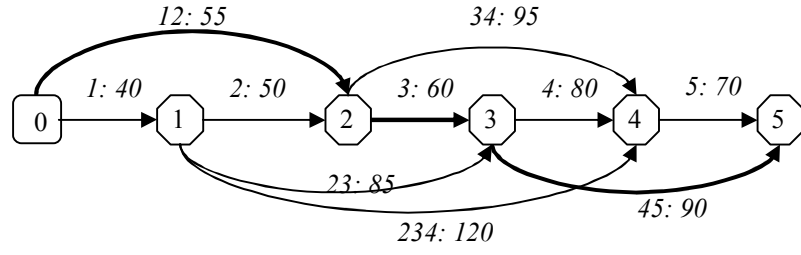
Algoritmada elde edilen en son $V_{|N|}$ değeri s çözümünün amaç fonksiyonu değerini vermektedir. Bu maliyete sahip çözümün rotaları ise ayırma algoritmasının ikinci aşamasında P vektöründen yararlanılarak bulunmaktadır.

Şekil 4.3'te büyük turdan ya da kromozomdan ayırma algoritması ile çözümün elde edilişi bir örnek üzerinde gösterilmektedir. Örnekteki beş müşterili ETD-ARP'ye ait kromozom $s=\{1, 2, 3, 4, 5\}$ olup, araç kapasitesi 12'dir. Şekil 4.3.a'da müşterilerin üzerindeki parantez içindeki sayılar sırasıyla toplama ve dağıtım taleplerini, hatlar üzerindeki sayılar ise noktalar arasındaki uzaklığı vermektedir. Bu problemin çözümünü gösteren büyük tur için elde edilen H yardımcı grafi ise Şekil 4.3.b'de yer almaktadır. H grafindaki $(0, 2)$ hattı 1'den 2'ye kadarki $\{1, 2\}$ müşterilerini içine

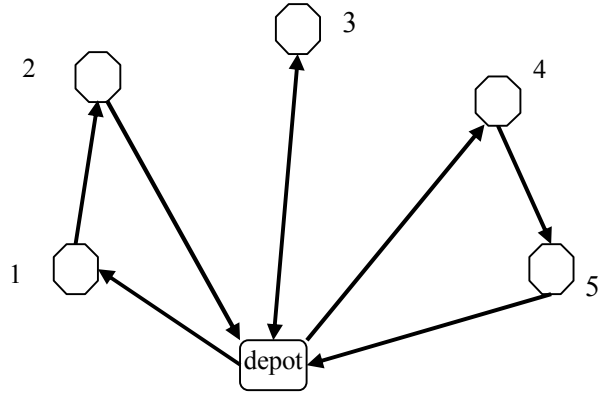
alan [0-1-2-0] turunu göstermektedir. Bu turun uzunluğu 55'dir.



a.



b.



c.

Şekil 4.3. Bir kromozomdan ETD_arp için çözümün elde edilmesi

- Beş noktalı bir ETD_arp örneği
- Ayrırma işleminin uygulanışı
- Elde edilen rotalar

Aynı şekilde (1, 4) hattı da, 2'den 4'e kadarki {2, 3, 4} müşterilerinden oluşan ve 120 birim uzunluğa sahip [0-2-3-4-0] turunu ifade etmektedir. Toplamda en küçük maliyeti veren hatlar koyu renk ile gösterilmiştir. H grafında ilk düğümden son düğüme kadar olan en kısa yol bulunduğunda, bu yolun değerinin 205 ($55+60+90=205$) birim olduğu görülmektedir. En kısa yola göre elde edilen rotalar (Bkz. Şekil 4.3.c)'de gösterilmiştir.

4.1.2. Başlangıç yığınının oluşturulması

Yapılan ön denemelerde, başlangıç yığının (Y^0) sezgisel algoritmalarla yararlanılarak üretilmesi ile tamamen rassal olarak üretilmesinin çözüm kalitesi ve algoritmanın yakınsaması üzerinde anlamlı bir etkisi olmadığı görülmüştür. Bu nedenle, GA'da başlangıç yığını tamamen rassal olarak üretilmiş $|Y^0|$ kromozomdan oluşturulmuştur. Başlangıç yığının rassal olarak üretimi aşamasında, yığının farklı çözümlerden oluşması ve yığındaki her çözümün birbirinden amaç fonksiyonu değeri açısından en az Δ kadar uzak olması sağlanmıştır. Amaç, yığındaki çeşitliliği artırarak çözüm uzayının etkin bir şekilde aranmasını gerçekleştirmek ve erken yakınsamayı önlemektir. Eğer ardışık 10 denemede bu kısıta uygun rassal bir k . çözüm elde edilememişse; başlangıç yığını $|Y^0| = k - 1$ genişliğinde bırakılmaktadır.

Yığında amaç fonksiyonu açısından tüm çözümlerin birbirinden en az Δ kadar uzak olması, algoritmanın her iterasyonunda elde edilen yığın için de geçerlidir. Δ değeri dinamiktir ve algortmada bulunan eniyi çözüm güncellendikçe bu değer de güncellenmektedir. s^* GA ile aramanın ilgili iterasyonuna kadar bulunan eniyi çözüm ve $f(s^*)$ bu çözümün amaç fonksiyonu değeri olmak üzere; Δ değeri $f(s^*)/1000$ eşitliği kullanılarak algoritma içinde azaltılmaktadır. Δ 'nın giderek daha küçük değerler alması, aramanın sonlarına doğru genetik operatörler ile elde edilen iyi çözümlerin yığına kabul edilme şansını artırmaktadır. Böylece algoritmanın sonlarına doğru, aramanın iyi çözümler etrafında devam etmesi sağlanmaktadır.

Yığındaki kromozomlar her zaman maliyetlerinin artan sırasına göre dizilmektedir. Dolayısıyla s_1 her zaman yığının eniyi, $s_{|Y^0|}$ ise enkötü kromozomudur. Yığının böyle bir dizilime sahip olması, hangi kromozomun yığını terk edeceğinin belirlenmesinde önemlidir.

4.1.3. Seçim mekanizması ve genetik operatörler

Çözüm gösterimi ve başlangıç yığını belirlendikten sonra durdurma koşulu sağlanıncaya kadar GA'nın her iterasyonunda; mevcut yığından seçim stratejisine göre seçilen kromozomlara genetik operatörler (*çaprazlama* ve *mutasyon* operatörleri) uygulanarak yeni kromozom(lar) üretilmekte ve bu kromozom(lar) belirlenen stratejilerle yığına eklenerek, her seferinde yeni bir yığın elde edilmektedir. Böylece biyolojik evrim sürecinde olduğu gibi, zamanla daha iyi kromozomların oluşturduğu bir yığın elde edilmeye çalışılmaktadır. Bu tez kapsamında geliştirilen GA, denge durumu (*steady-state*) GA'ya dayalıdır. Denge durumu GA'da her iterasyonunda, genetik operatörler seçim stratejisine göre seçilen bir çiftte uygulanmaktadır. Bu bölümde, ETD_ARNP için önerilen GA'da kullanılan seçim mekanizması ve genetik operatörler anlatılacaktır.

Seçim mekanizması

Çaprazlama operatörünün uygulanabilmesi için mevcut yığından seçilecek iki kromozoma ihtiyaç vardır. GA'da seçilen bu kromozomlar *ebeveynler* (*parents*), kromozomlara çaprazlama operatörünün uygulanmasıyla elde edilen yeni kromozomlar ise *çocuklar* (*offsprings*) olarak bilinmektedir. Belirlenen seçim stratejisine göre; ebeveynlerin ilki mevcut yığından ikili turnuva yöntemi ile seçilmektedir. Diğer ebeveyn ise GA'nın arama sırasında bulduğu eniyi çözümü temsil eden kromozom olarak belirlenmektedir. Böylece, yığındaki eniyi çözüme ait özelliklerin çocuklara taşınması amaçlanmaktadır. İkili turnuva, yığından rasgele seçilen iki kromozomdan daha düşük maliyetli olanının ebeveyn olarak dikkate alındığı yöntemdir.

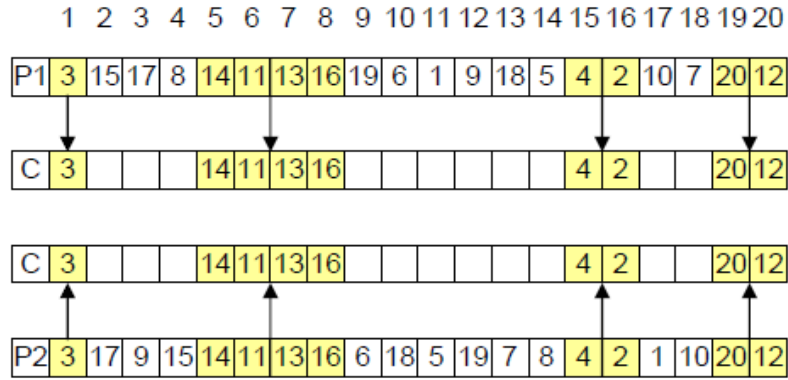
Genetik operatörler

Çaprazlama operatörü

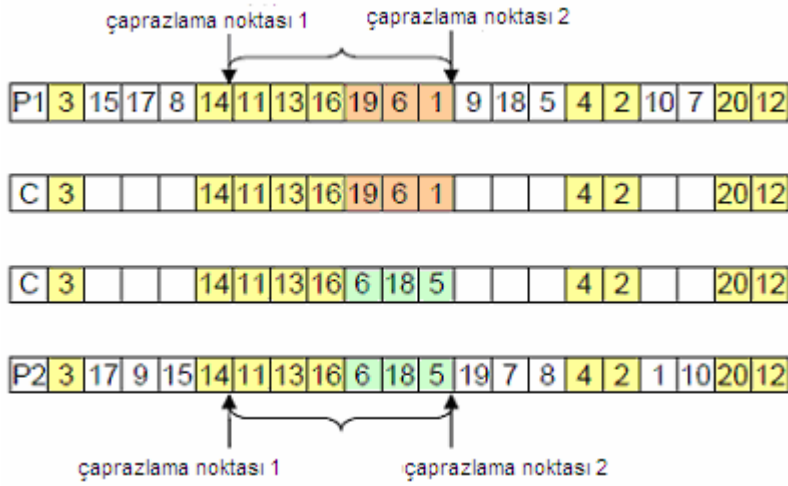
Permütasyon kodlamanın kullanıldığı algoritmalarda uygun olmayan çözümlerin elde edilmesini önlemek amacıyla, literatürde özel çaprazlama operatörleri geliştirilmiştir. Bu operatörler; kısmi çaprazlama (*partial mapped crossover*, PMX), sıralı çaprazlama (*order crossover*, OX) ve çevrim çaprazlamadır (*cycle crossover*, CX). Bunlardan literatürde en sık kullanılanları PMX ve OX'dir. Örneğin, Prins (2004) kapasiteli ARP için permütasyon kodlamaya dayalı geliştirdiği GA'da OX operatörünü kullanmıştır [Prins, 2004].

Ruiz ve ark. (2006), akış-tipi çizelgeleme problemini çözmek amacıyla permütasyon kodlamaya dayalı bir GA ve bu kodlama için dört farklı yeni çaprazlama operatörü geliştirmiştir [Ruiz ve ark., 2006]. Bu operatörler; *benzer iş sıralı çaprazlama* (*similar job order crossover-SJOX*), *benzer blok sıralı çaprazlama* (*similar block order crossover-SBOX*), *benzer iş 2-noktalı sıralı çaprazlama* (*similar job 2-point order crossover-SJ2OX*) ve *benzer blok 2-noktalı sıralı çaprazlama* (*similar block 2-point order crossover-SB2OX*) olarak adlandırılmıştır. Operatörlerin temel özelliği; ebeveynlerde birbirinin aynı olan genlerin olduğu gibi çocuklara taşınmasını sağlayarak, arama sırasında kazanılan temel yapı taşlarını (building blocks) koruyabilmektir. Ruiz ve ark. (2006), akış-tipi çizelgelemede bu operatörler ile literatürdeki operatörlerin (PMX, OX, vb.) performansını bir deneysel çalışma ile incelemişler ve SB2OX'in daha iyi bir performans sergilediğini göstermişlerdir.

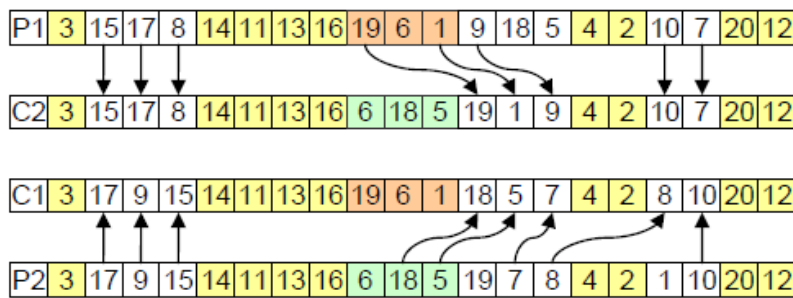
Permütasyon kodlamaya dayalı olarak geliştirilen GA'da kullanılacak çaprazlama operatörünün seçimi için yapılan ön denemelerde, literatürde çok sık kullanılan PMX operatörünün yanı sıra Ruiz ve ark. (2007) tarafından önerilen dört çaprazlama operatörü (*SJOX*, *SBOX*, *SJ2OX*, *SB2OX*) dikkate alınmıştır. Sonuçta, SJ2OX operatörünün kullanılmasına karar verilmiştir. SJ2OX operatörü ilgilenilen problem için, ilk defa bu çalışmada kullanılmıştır. *P1* ve *P2* ebeveynlerine SJ2OX'in uygulandığı ile *C1* ve *C2* çocuklarının elde edilişi Şekil 4.4'de verilmektedir.



a.



b.



c.

Şekil 4.4. Benzer iş 2-noktalı sıralı çaprazlama operatörünün uygulanması

- İki ebeveyndeki ortak genler çocuklara kopyalanması
- Çaprazlama noktaları arasındaki kısmın ebeveynlerden aktarılması
- Eksik genlerin diğer ebeveynden sırasıyla kopyalanması

(Bkz. Şekil 4.4.a)'da görüldüğü gibi ilk önce ebeveynlerdeki genler sırasıyla teker teker incelenerek, iki kromozomda da aynı olan genler olduğu gibi iki çocuğa da aktarılmaktadır. İkinci aşamada; $|N|$ problemdeki müşteri sayısı olmak üzere, 1 ile $|N|$ arasında iki nokta rassal olarak seçilmekte ve birinci ebeveynin bu iki nokta arasında kalan kısmı birinci çocuğa, ikinci ebeveynin ise ikinci çocuğa olduğu gibi aktarılmaktadır (Bkz. Şekil 4.4.b). Üçüncü ve son aşamada ise; birinci çocuğun eksik genleri baştan başlanarak; ikinci ebeveyndeki genlerden sırasıyla tamamlanmaktadır (Bkz. Şekil 4.4.c). Aynı işlem ikinci çocuk için de yapılarak, iki yeni kromozom üretilmektedir. Şekil 4.4'de görüldüğü gibi, uygulanan çaprazlama operatörü ile mevcut kromozomlardaki rassal noktalar arasında kalan genler ve iki kromozomdaki ortak genler olduğu gibi çocuklara geçmektedir.

Mutasyon operatörü

Permutasyon kodlamaya dayalı olarak geliştirilen GA_DKİ'de, çaprazlama operatörü ile elde edilen çocuklara mutasyon operatörü olarak DKİ algoritması kullanılmıştır.

Metasezgisel algoritmalarının performansının artırılmasında arama sırasında elde edilen çözümlerin YA algoritması ile iyileştirilmesi önemli bir etkiye sahiptir. Bir problemin çözümü için metasezgisel algoritmanın basit yerel arama ya da bir diğer metasezgiselle birlikte kullanıldığı algoritmalar *melez algoritmalar (hybrid algorithms)* olarak adlandırılmaktadır. Literatürde son yıllarda çeşitli KEP'nin çözümü için melez algoritmaların çok sık kullanıldığı görülmektedir. Melez metasezgisel algoritmaların sınıflandırılması ile ilgili bir çalışma Talbi (2002) tarafından yapılmıştır. Önerilen GA da DKİ algoritmasının kullanıldığı melez bir algoritmadır. Böylece iki metasezgiselin avantajlarının birleştirilmesiyle ETD_ARP için hem çözüm kalitesi hem de çözüm zamanı açısından literatürdeki diğer algoritmalarla rekabet edebilecek nitelikte bir GA geliştirilmiştir.

Geliştirilen GA_DKİ algoritmasının her iterasyonunda çaprazlama ile elde edilen iki yeni çözüm yani çocuklar ($C1$ ve $C2$) DKİ algoritması ile iyileştirilmektedir. Bunun için, öncelikle çocukların karşılık geldiği rotalar ayırma algoritması (Bkz. Şekil 4.2)

ile bulunmalıdır. Rotalar DKİ ile iyileştirildikten sonra, çözümlerin tekrar birer kromozom olarak kodlanması gerekmektedir. Ardından mutasyona uğramış kromozomların ($M1$ ve $M2$) amaç fonksiyonu değerleri $f(M1)$ ve $f(M2)$ tekrar hesaplanmaktadır. Çünkü ayırma algoritmasına göre büyük bir turun rotalara ayrılması, bazen rotaların toplam maliyetinin düşmesine neden olmaktadır.

Mladenovic ve Hansen tarafından önerilen DKİ algoritması, arama sırasında sistematik bir şekilde kullanılan komşuluk yapısını değiştirerek iyi ya da en iyi çözüme ulaşmaya çalışan metasezgisel bir algoritmadır [Mladenovic ve Hansen, 1997]. DKİ'nin uygulanabilmesi için öncelikle problem için kullanılacak komşuluk yapıları kümesinin, $K = \{N^1(.), N^2(.), \dots, N^{|K|}(.)\}$ belirlenmesi gerekmektedir. Bu komşuluk yapıları, belirlenen sırayla kullanılarak çözüm uzayında arama gerçekleştirilir. Aramaya ilk sıradaki $N^1(.)$ komşuluk yapısı ile başlanır. Herhangi bir anda kullanılan komşuluk yapısı ile mevcut çözüm iyileştirilmiş ise o ana kadar bulunan eniyi çözüm güncellenir ve ilk sıradaki komşuluk yapısına tekrar dönülür. Aksi takdirde bir sonraki komşuluk yapısı ile aramaya devam edilir. Tüm komşuluk yapıları sırasıyla uygulanıp bir iyileşme gerçekleşmemiş ise algoritma sonlandırılır.

GA_DKİ'de mutasyon operatörü olarak kullanılan DKİ algoritmasında, 3'ü rotalar arası ve 3'ü rota içi olmak üzere toplam 6 farklı komşuluk yapısıyla çalışılmıştır. Rotalararası komşuluk yapıları; çaprazlama, 1-1 yer değiştirme ve 1-0 yer değiştirme; rota içi komşuluk yapıları ise 2-opt, yer değiştirme ve ters çevirmedir. Aşağıda bu hareket mekanizmaları açıklanmıştır.

Çaprazlama: Bu hareket tipinde, r_1 rotasına ait ardışık n_1 ve n_2 müşterileri arasındaki hat ile, r_2 rotasına ait ardışık n_3 ve n_4 müşterileri arasındaki hat kaldırılarak her iki rota iki parçaya bölünür. Ardından n_1 ile n_4 ve n_3 ile n_2 arasına yeni hatlar eklenerek; r_1 rotasının ilk kısmı r_2 rotasının ikinci kısmıyla, r_2 rotasının ilk kısmı da r_1 rotasının ikinci kısmıyla birleştirilmiş olur. Bu komşuluk yapısında tüm rota çiftleri ve ayırma noktaları dikkate alınmaktadır.

1-1 Yer deęiřtirme: Bu hareket tipinde, r_1 rotasından n_1 ve r_2 rotasından da n_2 müşterisi kaldırılır. n_1 müşterisi r_2 rotasındaki, n_2 müşterisi ise r_1 rotasındaki mümkün olan eniyi yere yerleştirilir. n_1 ve n_2 müşterileri geleneksel yer deęiřtirme hareketinde olduęu gibi birbirlerinin pozisyonuna konulmaması nedeniyle, tüm komřuluęun aranması çok fazla sayıda hareketin denenmesini gerektirir. Zamandan tasarruf etmek için, sadece r_2 rotasında n_1 'e, r_1 rotasında da n_2 'ye uzaklık olarak en yakın 10 müşteriden en az birisinin bulunması şartını saęlayan n_1 ve n_2 müşterileri için bu hareket uygulanmıřtır.

1-0 Yer deęiřtirme: r_1 rotasına ait n_1 müşterisi, r_2 rotasındaki en iyi yere yerleştirilir. Tüm müşteriler için dięer rotalarda yerleřebilecekleri tüm pozisyonlar dikkate alınmıřtır.

2-opt: Bu hareket tipinde aynı rotadaki, n_{i-1} ile n_i ve n_{j-1} ile n_j müşterileri arasındaki birbiriyle ardışık olmayan 2 hat kaldırılarak, yerine n_{i-1} ile n_{j-1} ve n_i ile n_j müşterileri arasında yeni hatlar oluřturulur. Böylece, rota içinde n_{i-1} ile n_j müşterileri arasında kalan kısım ters çevrilmiř olur.

Yer deęiřtirme: Rota içinden seęilen n_1 ve n_2 müşterilerinin pozisyonları deęiřtirilir.

Ters Çevirme: Rotanın yönü ters çevrilir. Bu hareket ile rota uzunluęu deęiřmedięinden dolayı çözüm kalitesi iyileřmez. Ancak, rotada tařınan en büyük yük miktarında bir azalma saęlanabilir. Dolayısıyla, ileriki iterasyonlarda toplam uzunluęu daha az olan rotaların oluřma řansını arttırmak amacıyla rotanın en büyük yükünde bir azalma olduęunda bu hareket geręekleřtirilir.

DKİ algoritmasının performansında bu komřuluk yapılarının uygulanma sırasının etkisi büyüktür. Yapılan ön denemeler sonucunda, komřuluk yapılarının burada aęıklandığı sırada kullanılmasına karar verilmiřtir.

DKİ, her bir komşuluk yapısında eniyi iyileştirme stratejisine göre aramayı gerçekleştirmektedir. Arama sırasında sadece uygun çözümler dikkate alınmaktadır. Dolayısıyla, yeni çözümün kapasite kısıtına uygun olup olmadığını anlamak için her hareketin ilgili olduğu rota veya rotaların enbüyük yükünde neden oldukları değişimin bulunması gerekmektedir. Bu hesabın kolay ve kısa sürede gerçekleştirilmesinde Bianchessi ve Righini (2007) tarafından tanımlanan özel matematiksel bağıntılardan yararlanılmıştır [Bianchessi ve Righini, 2007].

DKİ'nin çözüm zamanını azaltmak amacıyla, en fazla zaman gerektiren 1-1 *yer değiştirme* komşuluk yapısının uygulanmasında tabu listesi kullanılmıştır. Tabu listesinde birbirleriyle rotalarını değiştirmiş olan müşteri çiftleri, sabit 7 iterasyon boyunca tabu olarak tutulmuştur. Tabu yıkma kriteri, arama sırasında bulunan eniyi çözümden daha iyi bir çözümün elde edilmesidir.

4.1.3. Önerilen melez GA'nın adımları

GA_DKİ'nin çalışma mekanizması Şekil 4.5'de kısaca özetlenmiştir. Başlangıç yığını oluşturulduktan sonra; algoritmanın her iterasyonunda, tanımlanan seçim mekanizmasına göre iki kromozom seçilir ve SJ2OX operatörü ile bu kromozomlardan iki yeni kromozom ($C1$ ve $C2$) elde edilir. Bu kromozomların her ikisi de DKİ algoritması kullanılarak iyileştirilir. Çocuklardan rassal olarak seçilen bir tanesi; yığındaki tüm çözümlere en az Δ kadar uzaklıktaysa yığına alınır ve karşılığında yığının en kötü $|Y'|/2$ kromozomu arasından rassal olarak seçilen bir tanesi yığından çıkartılır. Böylece, yeni yığın elde edilmiş olur ve durdurma koşulu sağlanıncaya kadar adımlar tekrarlanır. Yığına kabul edilen kromozomlar; kromozomların artan maliyetlere göre sıralamasını bozmayacak şekilde yerleştirilirler. Öyleki yığında her zaman $i < j$ ise; $s_i, s_j \in Y'$ olmak üzere, $f(s_i) \leq f(s_j)$ olmalıdır. Yeni yığına, üretilen yeni çözümler arasından rassal olarak seçilen bir tanesinin alınması ve bu çözümün yığındaki tüm çözümlere en az Δ uzaklıkta olması şartının aranması, erken yakınsamayı önlemek amacıyla kullanılmaktadır. Ancak, üretilen çözümlerden birisi ya da her ikisi, arama sırasında

bulunan eniyi çözümden daha iyi ise; bunlardan küçük maliyetli olanı bir şart aranmaksızın mevcut yığına eklenir. Bulunan eniyi çözümün iyileşmediği ardışık iterasyon sayısı ve toplam iterasyon sayısı algoritmanın durdurma koşulları olarak dikkate alınmıştır. GA_DKİ algoritması önce sağlanan koşula göre durdurulmaktadır.

Algoritma: ETD_ARP için <i>GA_DKİ</i> Girdi: ETD_ARP için problem girdileri ve GA'ya ait parametreler Çıktı: Bulunan iyi çözüm
Begin $t \leftarrow 0$; Başlangıç yığının oluştur: Y^t Yığının bireylerini değerlendir. while not (durdurma kriterinin sağlanması) do Y^t 'den ikili turnuva seçim yöntemi ile ilk ebeveyni elde et; $P1 \in Y^t$ Y^t 'den eniyi kromozomu ikinci ebeveyn $P2$ olarak dikkate al; $P1$ ve $P2$ 'den <i>SJ2OX</i> operatörü ile $C1$ ve $C2$ çocuklarını elde et; $C1$ ve $C2$ 'yi DKİ algoritmasıyla iyileştir. Eğer $C1$ ve $C2$ 'den iyi olanı Y^t 'deki eniyi kromozomdan daha iyi ise, bu kromozomla; Aksi halde, $C1$ ve $C2$ 'den rasgele seçilen ve Δ uzaklığını sağlayan kromozom ile mevcut Y^t yığının güncelle $t \leftarrow t + 1$; end Bulunan en iyi çözümü rapor et. End

Şekil 4.5. Önerilen GA'nın adımları

4.2. Önerilen Kesikli Kuş Sürüsü Eniyileme Algoritması

Tez kapsamında ETD_ARP için geliştirilen ikinci algoritma KSE_DKİ'dir. Bu algoritma kesikli KSE (KKSE) ve DKİ'e dayalı melez bir algoritmadır. ETD_ARP'nin çözümü için genellikle YA tabanlı algoritmalar önerilmiştir. Hem bu nedenle hem de yeni bir metasezgisel olmasıyla nedeniyle; literatürde ilgili problemin çözümü için standart KSE'yi kullanan yalnızca tek çalışma mevcuttur [Jin Ai ve Kachitvichyanukul 2009]. ETD_ARP için KKSE'ye dayalı ilk algoritma ise bu tez kapsamında geliştirilen melez KSE_DKİ algoritmasıdır.

KSE_DKİ'de çözüm uzayının iyi çözümlerin bulunduğu bölgelerine ulaşmak için KKSE, bu bölgelerin yoğun olarak taranması için ise DKİ algoritması kullanılmıştır. Algoritmanın her iterasyonunda güncellenen çözümlerin iyileştirilmesi için DKİ algoritması belli bir olasılıkla uygulanmaktadır. Aynı zamanda bulunan iyi çözümlerin kabul edilip edilmemesine karar vermek amacıyla, TB kullanılmıştır. Amaç *yoğunlaşmanın* (DKİ) yanı sıra *çeşitlendirme* stratejisinin de (TB) kullanılarak, çözüm uzayının etkin şekilde taranmasını gerçekleştirmektedir.

Standart KSE algoritması, evrimsel algoritmalar sınıfında yer alan yığın tabanlı bir algoritmadır. KSE, Kennedy ve Eberhart (1995) tarafından orijinalinde sürekli eniyileme problemlerinin çözümü için önerilmiştir [Kennedy ve Eberhart, 1995]. KSE'nin bazı özellikleri şu şekilde sıralanabilir: (i) Hafızadan dolayı iyi çözümlerin bilgisine sürüdeki tüm kuşlar sahiptir, (ii) sürüdeki kuşlar arasında bilgi paylaşımı söz konusudur ve (iii) uygulanması kolaydır.

KSE, kuş veya balık sürüsü gibi grup halindeki canlıların hareketini taklit eden bir metasezgiseldir. Bir KSE algoritmasında, yığının her bir üyesi bir *kuş* olarak adlandırılır. Her kuş, çok boyutlu arama uzayında belirli bir *hız* ile hareket ederek, pozisyonunu değiştirmektedir. Her *pozisyon* ilgilenilen problem için belli bir çözüme karşılık gelmektedir. KSE algoritması, çözüm uzayından seçilmiş çözümlerin bir yığını ile aramaya başlar ve algoritmanın her iterasyonunda tüm kuşların pozisyonu güncellenir. Her kuş yeni pozisyonunu kendi tecrübesine (*bilişsel bilgi*) ve sürüden

öğrendiği bilgiye (*sosyal bilgi*) göre belirleyerek, yiyeceğe yani iyi çözüme ulaşmayı amaçlamaktadır. Burada, bilişsel bilgi kuşun iterasyonlar boyunca bulduğu eniyi pozisyon, sosyal bilgi ise yığındaki tüm kuşların eniyi pozisyonudur. Dolayısıyla kuşun pozisyonunu güncelleyecek olan hız vektörü, kuşun mevcut hızı ile bilişsel ve sosyal bileşenlerden oluşmaktadır.

Standart KSE algoritması Şekil 4.6’da açıklanmıştır. KSE algoritmasında; d -boyutlu arama uzayındaki i . kuşun t zamanındaki pozisyonu ve hızı sırasıyla $X_i^t = \{x_{i1}^t, x_{i2}^t, \dots, x_{id}^t\}$ ve $V_i^t = \{v_{i1}^t, v_{i2}^t, \dots, v_{id}^t\}$ ile gösterilir. Yerel eniyi pozisyon ($pbest$), bir kuşun tüm arama boyunca sahip olduğu eniyi pozisyonudur ($P_i = \{p_{i1}, p_{i2}, \dots, p_{id}\}$). Global eniyi pozisyon ($gbest$) yani sürünün tüm t zamanına kadarki arama boyunca sahip olduğu eniyi pozisyon ise $G = \{g_1, g_2, \dots, g_d\}$ ile gösterilir. $(t+1)$. iterasyonda sürüdeki her kuşun yeni hızı Eş. 4.5 kullanılarak hesaplanmaktadır:

$$v_{ij}^{t+1} = wv_{ij}^t + h_1 r_1' [p_{ij} - x_{ij}^t] + h_2 r_2' [g_j - x_{ij}^t] \quad j = 1, 2, \dots, d \quad (4.5)$$

Bu eşitlikte h_1 ve h_2 , bilişsel ve sosyal bileşenlerin katkılarını ölçekleyen pozitif hızlanma sabitleri; w atalet (inertia) faktörü, r_1' ve r_2' ise $[0,1]$ aralığında düzgün dağılımdan örneklenen rassal sayılardır. Her iterasyonda sürüdeki her i . kuşunun pozisyonu Eş. 4.6 kullanılarak belirlenmektedir.

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad j = 1, 2, \dots, d \quad (4.6)$$

Her iterasyonda sürüdeki kuşlar belirlenen yeni pozisyonlarına uçarlar ve bu süreç önceden belirlenen bir durdurma koşulu sağlanıncaya kadar tekrarlanır. Bulunan her yeni çözümün amaç fonksiyonu değeri $f(X_i^t)$, P_i ve G ‘nin amaç fonksiyonu değerleri ($f(P_i)$ ve $f(G)$) ile karşılaştırılarak; P_i ve G iyileştirilir.

```

Begin
 $t \leftarrow 0$ 
Başlangıç sürüsünü elde et      :  $Y^t = \{X_1^t, X_2^t, \dots, X_{|Y|}^t\}$ 
Her kuş için en iyi pozisyonu tut :  $P_i \leftarrow X_i^t$  for  $\forall i \in \{1, \dots, |Y|\}$ 
Global en iyi pozisyonu tut      :  $G \leftarrow X_i^t, f(G) \leq X_i^t$  for  $\forall i \in \{1, \dots, |Y|\}$ 
while not (durdurma koşulu) do
    for  $i := 1, 2, \dots, |Y|$  do
         $i$  kuşunun pozisyonunu Eş. 4.5 ve Eş 4.6'ya göre güncelle.
         $i$  kuşunun yerel eniyi pozisyonunu bul : if  $f(X_i^t) \leq f(P_i)$  then  $P_i \leftarrow X_i^t$ 
        Global en iyi pozisyonu bul          : if  $f(X_i^t) \leq f(G)$  then  $G \leftarrow X_i^t$ 
    end for
 $t \leftarrow t + 1$ 
end while
 $G$  ve  $f(G)$  'yi rapor et.
End

```

Şekil 4.6. Standart KSE algoritmasının adımları

Standart KSE sürekli problemlerin çözümü için önerilmiş bir metasezgisel olduğundan (Bkz. Eş. 4.5 ve Eş. 4.6), bir KEP olan ETD_ARP için diğer algoritmalar kadar iyi çözümler üretememektedir. Bu durum, Ai ve Kachitvichyanukul'un (2009) çalışmasında da görülmektedir [Ai ve Kachitvichyanukul, 2009]. Standart KSE'nin KEP'e uyarlanmış bir versiyonu olan KKSE ile ilgili literatürdeki çalışmaların çoğu ise çeşitli çizelgeleme problemlerinin çözümü üzerinedir [Sha ve Hsu (2008), Anghinolti and Paolucci (2009), Jarboui ve ark. (2008), Pan ve ark. (2008) ve Zhang ve ark. (2009)]. Bu tez kapsamında ETD_ARP'nin çözümleri permutasyon kodlama ile gösterildiğinden, geliştirilen KSE_DKİ algoritmasında standart KSE'nin kesikli problemler için bir versiyonu olan KKSE kullanılmıştır.

Bu bölümde; ETD_ARP için önerilen KSE_DKİ algoritmasında çözümlerin gösterilmesi, başlangıç yığınının oluşturulması ve yığının güncellenmesi detaylarıyla anlatılmaktadır. Son olarak da önerilen melez KKSE algoritmasının adımları verilmektedir.

4.2.1. Çözüm gösterimi

KKSE'nin performansı, çözümleri göstermek amacıyla kullanılan kodlama yapısına oldukça bağlıdır. GA'da bahsedildiği gibi ETD_ARP ve ARP ile ilgili çalışmalar incelendiğinde, çözümlerin gösteriminde “*permütasyon kodlama*” ve “*rassal anahtarlama*”ya dayalı farklı kodlama tiplerinin kullanıldığı görülmektedir. Önerilen KSE_DKİ algoritmasında da GA_DKİ'deki gibi çözümler, tüm müşterilerin büyük bir tur sırasını gösteren *permütasyon kodlama* (Bkz. Şekil 4.1) ile ifade edilmektedir. Büyük turu kapasite kısıtına uygun rotalara bölmek için yine Prins'in geliştirdiği ve bu tez kapsamında ETD_ARP'ye uyarlanan *ayırma algoritması* (Bkz. Şekil 4.2 ve 4.3) kullanılmaktadır.

4.2.2. Başlangıç yığınının oluşturulması

Başlangıç yığınındaki turların biri en yakın komşu algoritması ile diğerleri ise rassal olarak üretilmiştir. Başlangıç yığının rassal olarak üretimi sırasında, çeşitliliği sağlamak için yığındaki her çözümün amaç fonksiyonu değeri açısından birbirlerinden en az Δ kadar uzak olmasına dikkat edilmiştir. Sabit Δ uzaklığı, $\Delta = 2$ olarak alınmıştır. Amaç, çözüm uzayını etkin bir şekilde arayarak, zamansız yakınsamayı önlemektir. Eğer ardışık 10 denemede bu kısıta uygun rassal bir k . çözüm elde edilememişse; başlangıç yığını $|Y| = k - 1$ genişliğinde bırakılmaktadır. Bu kısıt diğer iterasyonlarda aranmamaktadır.

4.2.3. Önerilen KSE’de pozisyonların güncellenmesi

KSE_DKİ için başlangıç yığını oluşturulduktan sonra; durdurma koşulu sağlanıncaya kadar standart KSE’de olduğu gibi, KKSE algoritmasının her iterasyonunda da tüm kuşların pozisyonu, her kuşun yerel eniyi pozisyonu P_i ve global eniyi pozisyon G güncellenmektedir (Bkz. Şekil 4.6). KKSE’de yeni pozisyonlar Pan ve ark.’nın (2008) çalışmasında yer alan Eş. 4.7’ye göre belirlenmektedir [Pan ve ark., 2008].

$$X_i^t = h_2 \otimes F_3(h_1 \otimes F_2(w \otimes F_1(X_i^{t-1}), P_i), G) \quad (4.7)$$

Eş. 4.7’de, F_1 mutasyon, F_2 ve F_3 ise çaprazlama operatörünü göstermektedir. w , h_1 ve h_2 sırasıyla bu operatörlerin uygulanma olasılığıdır. Standart KSE’de olduğu gibi Eş. 4.7’e göre kuşun yeni pozisyon belirlenirken, kuşun hızı, yerel eniyi pozisyon ve global eniyi pozisyon olmak üzere üç bileşenden belirli olasılıklarda etkilenilmektedir. İlk bileşen $\lambda_i^t = w \otimes F_1(X_i^{t-1})$ olup, i . kuşun $(t-1)$. iterasyondaki pozisyonuna (X_i^{t-1}) w olasılığı ile mutasyon operatörünün uygulanmasını ifade etmektedir. Bu operatör kuşun kendi yörüngesinde uçmasını modellemektedir. r' , $[0,1]$ aralığında düzgün dağılımdan üretilen rassal sayı olmak üzere λ_i^t Eş. 4.8’de verildiği gibi elde edilmektedir. Bu eşitliğe göre üretilen rassal sayı r' , w ’den küçük yada eşitse kuşun mevcut pozisyonu mutasyona uğratılmakta, aksi durumda (X_i^{t-1}) aynı kalmaktadır ($\lambda_i^t = X_i^{t-1}$).

$$\lambda_i^t = \begin{cases} F_1(X_i^{t-1}) & \text{eger } r' \leq w \\ X_i^{t-1} & d.d. \end{cases} \quad (4.8)$$

Eş. 4.7’deki ikinci bileşen $(\delta_i^t = h_1 \otimes F_2(\lambda_i^t, P_i))$, λ_i^t çözümü ile i . kuşun $((t-1)$. iterasyona kadar bulunan) yerel eniyi çözümüne (P_i) , h_1 olasılığı ile çaprazlama operatörünün uygulanmasını ifade etmektedir. Bu ifadeye göre δ_i^t , Eş. 4.9 kullanılarak elde edilir. Burada r' , $[0,1]$ aralığında düzgün dağılımdan üretilen rassal

sayıyı göstermektedir. Eş. 4.9'a göre, ya h_1 olasılığıyla λ_i^t ile P_i çözümleri çaprazlanarak kuş için yeni bir pozisyon elde edilmekte ya da kuşun pozisyonu $(1-h_1)$ olasılığıyla aynı $(\delta_i^t = \lambda_i^t)$ kalmaktadır. Bu operatör; yeni pozisyon belirlenirken kuşun bulunan yerel eniyi pozisyonu bilgisinin belirli bir olasılıkla kullanılmasını modellemektedir.

$$\delta_i^t = \begin{cases} F_2(\lambda_i^t, P_i) & \text{eger } r' \leq h_1 \\ \lambda_i^t & d.d. \end{cases} \quad (4.9)$$

Eş. 4.7'deki son bileşen $(X_i^t = h_2 \otimes F_3(\delta_i^t, G))$ ise; δ_i^t çözümü ile global eniyi çözümüne $(gbest, G)$, h_2 olasılığı ile çaprazlama operatörünün uygulanmasını ifade etmektedir. Bu ifadeye göre X_i^t , Eş. 4.10 kullanılarak elde edilir. Burada r' , $[0,1]$ aralığında düzgün dağılımdan üretilen rassal sayıyı göstermektedir. Eş. 4.10'a göre, ya h_2 olasılığıyla δ_i^t ile G 'ye çaprazlama operatörü kullanılarak kuşun yeni pozisyonu belirlenmekte ya da $(1-h_2)$ olasılığıyla kuşun pozisyonu $X_i^t = \delta_i^t$ olarak kalmaktadır. Bu operatör kuşun yeni pozisyonu belirlenirken, belli bir olasılıkla sürünün global eniyi pozisyon bilgisinin kullanılmasını modellemektedir. Bu yaklaşım, Pan ve ark. (2008) tarafından akış-tipi çizelgeleme problemine uygulanmış [Pan ve ark., 2008] ve algoritmanın performansı literatürdeki algoritmalar ile karşılaştırmalı olarak incelenmiştir. İlerleyen kısımda çaprazlama ve mutasyon operatörleri anlatılmaktadır.

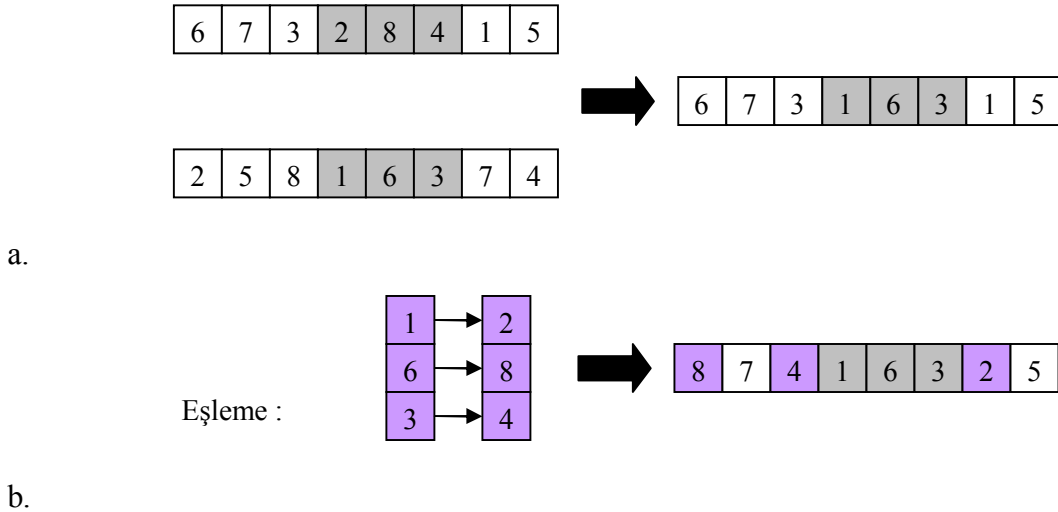
$$X_i^t = \begin{cases} F_2(\delta_i^t, G) & \text{eger } r' \leq h_2 \\ \delta_i^t & d.d. \end{cases} \quad (4.10)$$

Önerilen KSE_DKİ'de, güncellenen her pozisyon belli bir olasılıkla DKİ algoritması kullanılarak iyileştirilmektedir. Ayrıca, P_i pozisyonu her zaman daha iyisiyle değiştirilmemektedir. Bunun yerine, amaç fonksiyonu farkına bağlı ve algoritma boyunca gittikçe azalan bir olasılıkla P_i , daha kötü çözümlerle de değiştirilmektedir.

P_i pozisyonunun kendisinden daha kötü bir çözümle güncellenip güncellenmeyeceğine karar vermek için TB algoritması işletilmektedir.

Çaprazlama operatörü

KKSE'ye dayalı olarak geliştirilen KSE_DKİ algoritmasında, yeni çözümleri elde etmek üzere (Bkz. Eş. 4.7) kullanılacak çaprazlama operatörü deneysel çalışmalar sonucunda belirlenmiştir. Deneysel çalışmada permütasyon kodlamada çok sık tercih edilen PMX, bu tezde önerilen GA_DKİ'de kullanılan SJ2OX ve Pan ve ark.'nın (2008) akış tipi çizelgeleme problemi için önerdikleri KKSE algoritmasında kullandıkları PTL [Pan ve ark., 2008] operatörleri dikkate alınmıştır. Çalışma sonucunda, PMX operatörünün önerilen algoritmanın performansını artırdığı görülmüştür. Bu nedenle, önerilen KSE_DKİ'de PMX operatörü kullanılmıştır. Şekil 4.7'de iki çözümünden PMX operatörü kullanılarak yeni bir çözümün elde edilmesi örnek üzerinde gösterilmektedir.



Şekil 4.7. PMX çaprazlama operatörünün uygulanişı

- Tur üzerinde rasgele seçilen iki nokta arasında kalan kısmın kopyalanması
- Eşleşmenin yapılarak tekrarların ortadan kaldırılması

PMX’de önce rasgele 2 nokta belirlenir (k_1 ve k_2) ve çaprazlanacak turlardan birinci turun kopyası alınarak yeni çözüm elde edilir. Daha sonra, ikinci turun k_1 ve k_2 noktaları arasında kalan kısmı yeni turda aynı noktalar arasına olduğu gibi kopyalanır (Bkz. Şekil 4.7.a). Bu işlem sonucunda uygun olmayan bir tur (bazı müşteriler turda iki kez bulunurken, bazıları hiç bulunmamaktadır) elde edilmektedir. Bu uygun olmayan turu uygun hale getirmek için, çaprazlanacak turlarda k_1 ve k_2 noktaları arasındaki müşteriler karşılıklı olarak eşleştirilir (Bkz. Şekil 4.7.b). Son olarak yeni turun, k_1 ve k_2 noktaları dışında kalan kısmı tek tek incelenerek, tekrar eden müşteriler eşleriyle yer değiştirilir (Bkz. Şekil 4.7.b).

Mutasyon operatörü

Önerilen KSE_DKİ algoritmasında kullanılan mutasyon operatöründe; tur üzerinde herhangi bir konumdaki müşteri kaldırılarak, aynı tur üzerinde rasgele seçilen bir başka konuma yerleştirilmektedir.

4.2.4. Önerilen melez KSE algoritmasının adımları

Aşağıda, KKSE’nin DKİ ve TB sezgiselleriyle melezleştirilmesiyle elde edilen KSE_DKİ algoritmasının adımları genel olarak anlatılmış ve Şekil 4.8’de verilmiştir. KSE_DKİ bir tanesi enyakın komşu sezgiselinden, diğerleri rassal olarak üretilmiş büyük turlardan (permütasyon) oluşan bir sürü yığını ile aramaya başlamaktadır. Başlangıç sürüsünü farklılaştırmak için; yığındaki herhangi bir kuşun pozisyonunun amaç fonksiyonu değeri açısından, sürüdeki diğer kuşların pozisyonundan en az $\Delta = 2$ kadar uzak olmasına dikkat edilmiştir..

Algoritmanın her iterasyonunda tüm kuşların pozisyonları güncellenmekte (Bkz. Eş. 4.7) ve ardından γ olasılığı ile YA tabanlı DKİ algoritmasıyla (Bkz. Şekil 4.2) iyileştirilmektedir. Güncelleme esnasında; her i kuşunun pozisyonu (X_i^t) , w olasılığı ile bir müşterinin tur üzerinde rasgele seçilen başka bir yere yeniden

Algoritma: ETD_ ARP için KSE_DKİ Girdi: Problem girdileri ve $KKSE$ 'ye ait parametreler; $\gamma, w, h_1, h_2, maxI, T^0, \mu$ Çıktı: İyi çözümü; G	
Begin $t \leftarrow 0; k \leftarrow 0$ Başlangıç sürüsünü elde et $: Y^t = \{X_1^t, X_2^t, \dots, X_{ Y }^t\}$ Her i kuşu için yerel eniyi pozisyonu bul $: P_i \leftarrow X_i^t$ for $\forall i \in \{1, \dots, Y \}$ Global en iyi pozisyonu bul $: G \leftarrow X_i^t, f(G) \leq X_i^t$ for $\forall i \in \{1, \dots, Y \}$ while not $(k \leq maxI)$ do for $i := 1, 2, \dots, Y $ do i kuşunun $: r' \leftarrow u \sim [0, 1]$; if $(r' \leq w)$ then $X_i^t \leftarrow M(X_i^t)$ pozisyonunu $: r' \leftarrow u \sim [0, 1]$; if $(r' \leq h_1)$ then $X_i^t \leftarrow PMX(X_i^t, P_i)$ güncelle $: r' \leftarrow u \sim [0, 1]$; if $(r' \leq h_2)$ then $X_i^t \leftarrow PMX(X_i^t, G)$ (Bkz. Eş. 4.7) i kuşunun pozisyonunu $DKİ$ algoritmasıyla iyileştir: $r' \leftarrow u \sim [0, 1]$; if $(r' < \gamma)$ then $X_i^t \leftarrow DKİ(X_i^t)$ i pozisyonunu değerlendir $: if (f(X_i^t) < f(G))$ then $G \leftarrow X_i^t; K \leftarrow 0$ else; $k \leftarrow k + 1$ $: if (f(X_i^t) < f(P_i))$ then $P_i \leftarrow X_i^t$ else; $D = \frac{[f(X_i^t) - f(P_i)]}{f(P_i)} * 100$ $r' \leftarrow u \sim [0, 1]$ if $\left(r' \leq e^{\frac{-D}{T}}\right)$ then $P_i \leftarrow X_i^t$ end for $t \leftarrow t + 1$ $T^{t-1} = T^t * \mu$ end while G ve $f(G)$ 'yi raporla End	

Şekil 4.8. Önerilen $KKSE$ algoritmasının sözde kodu

yerleştirilmesi $(M(X_i^t))$ suretiyle değiştirilir. Ardından elde edilen pozisyon, h_1 olasılığıyla i . kuşun yerel eniyi pozisyonu ile PMX operatörü (Bkz. Şekil 4.7) kullanılarak çaprazlanır $(PMX(X_i^t, P_i))$. Son olarak da elde edilen pozisyon h_2 olasılığı ile global eniyi pozisyonla çaprazlanır $(PMX(X_i^t, G))$.

Standart KSE algoritmasından farklı olarak KSE_DKI’de, her i . kuşun yerel eniyi pozisyonu (P_i) güncellenirken TB’ndeki kabul olasılığı kullanılmıştır. Bu durumda i . kuşun yeni pozisyonunun amaç fonksiyonu değeri, yerel eniyi pozisyonundan daha iyi ise yerel eniyi pozisyon güncellenmekte, kötü ise güncelleme Eş. 4.11’deki kabul olasılığına göre yapılmaktadır. Amaç, kuşun eniyi pozisyonunun arasına kötü pozisyonlar ile yerdeğiştirmesini sağlayarak arama sırasında değişkenliği artırmaktır.

$$D = \frac{-[f(X_i^t) - f(P_i)]}{f(P_i)} * 100 \quad (4.11)$$

Algoritma önceden belirlenen sayıdaki ($maxI$) ardışık iterasyon boyunca global en iyi pozisyonda bir iyileşme gerçekleşmezse sonlandırılmaktadır.

5. DENEYSEL ÇALIŞMALAR

Önerilen algoritmaların performansının belirlenmesi amacıyla; literatürde var olan çeşitli test problemi kümeleri kullanılarak deneysel çalışma yapılmıştır. Deneysel çalışmada; her test problemi geliştirilen algoritmalar ile 10 kez çözülerek elde edilen çözüm değerleri ve hesaplama zamanları tutulmuştur. Denemeler Windows Vista Bussiness işletim sistemi altında çalışan, 1 GB RAM'e sahip, Intel Xeon 3.16 GHz hızındaki bir bilgisayarda gerçekleştirilmiş olup, algoritmalar Visual C# programlama dilinde kodlanmıştır. Sonuçlar; hem çözüm kalitesi hem de çözüm zamanına dayalı performans kriterleri açısından literatürdeki diğer algoritmalarla karşılaştırmıştır. Bu bölümde, önce kullanılan test problemleri hakkında kısa bilgi verilip, ardından deneysel çalışma sonuçları sunulmuştur.

5.1. Test Problemleri

Deneysel çalışmalarda, iki farklı test problemi kümesi kullanılmıştır.

5.1.1. Dethloff'ın test problemleri

Birinci problem kümesi, Dethloff (2001) tarafından üretilmiştir ve her biri 50 müşterili, 40 ETD_ARP'nden oluşmaktadır. Bu problemlerde, müşterilerin test yüzeyi üzerine yerleştirilmesinde iki farklı coğrafik senaryo kullanılmıştır. İlk senaryoda müşteriler $[100 \times 100]$ 'lük ızgara yüzey üzerine tamamen rassal olarak yerleştirilmiştir (SCA tipi). Diğer senaryoda; müşterilerin yarısı ilk senaryodaki gibi, diğer yarısı ise $[100/3 \times 100/3]$ 'lük ızgara yüzey üzerine rassal olarak yerleştirilmiş. Böylece, müşterilerden bazılarının yerleşim bölgesinin bir bölümünde yoğunlaşması sağlanmıştır (CON tipi). Müşterilerin dağıtım talepleri, $[0, 100]$ aralığında rassal olarak üretilmiştir. Toplama talepleri ise her müşterinin kendi dağıtım talebine bağlı olarak, $p_i = q_i * (0,5 + r'_i)$ şeklinde belirlenmiştir. Burada; r'_i , $[0,1]$ aralığında rassal bir sayıdır. Dethloff (2001), farklı araç kapasitesine bağlı olarak, her senaryo için iki, toplamda ise dört problem grubu (SCA3, CON3, SCA8, CON8 grupları)

tanımlamıştır. Böylece, her grup için 10 olmak üzere toplamda 40 test problemi üretmiştir.

5.1.2. Salhi ve Nagy'nin test problemleri

İkinci test problemi kümesi Salhi ve Nagy (1999) tarafından, Christofides, Mignozzi ve Toth'un (1979) literatürde CMT problemleri olarak bilinen 14 KARP'nden türetilmiştir. Bu problemlerden 7 tanesi araç kapasitesi kısıtının yanı sıra, enbüyük rota uzunluğu kısıtına da sahiptir. Bu kısıtın olduğu test problemleri deneysel çalışmanın dışında tutulmuştur. CMT problemlerinin müşteri sayısı 50 ile 200 arasında değişmektedir. Salhi ve Nagy 5 farklı yaklaşım kullanarak, 14 problemten 5 grup problem (70 adet) elde etmiştir. Bunlardan; X ve Y grubundakiler (CMTX ve CMTY problemleri) ETD_ARP türünde iken, T, Q, H grubundakiler (CMTT, CMTQ, CMTH problemleri) ise KTD_ARP'dir. X tipi problemler şu yöntemle elde edilmiştir. Önce her i müşterisi için koordinatlarına (x_i, y_i) bağlı olarak r_i oranı, $r_i = \text{enküçük}(x_i/y_i, y_i/x_i)$, hesaplanmıştır. Ardından, orijinal talep değerleri bu oran kullanılarak toplama ve dağıtım talebi olarak ayrılmıştır. Öyleki; o_i i müşterisinin orijinal talebi olmak üzere dağıtım ve toplama talepleri sırasıyla $q_i = r_i * o_i$ ve $p_i = o_i - q_i$ olarak belirlenmiştir. Benzer şekilde her müşterinin elde edilen taleplerinin yerleri değiştirilerek ($p_i = r_i * o_i$ ve $q_i = o_i - p_i$) Y tipi problemler elde edilmiştir. T, Q, H serileri ise sırasıyla; orijinal problemin her 10, 4 ve 2. müşterisinin toplama müşterisi, dağıtım talebinin de toplama talebi olarak belirlenmesiyle oluşturulmuştur. KTD_ARP, her müşterinin toplama veya dağıtım taleplerinden birinin sıfıra eşit olduğu ETD_ARP olarak kabul edilebilir. Bu nedenle, ETD_ARP için önerilen algoritmalarda bir değişiklik yapmadan KTD_ARP türündeki problemler de çözülebilmektedir.

Yürütülen deneysel çalışmada 21 tanesi KTD_ARP, geri kalanı ise ETD_ARP olmak üzere toplam 75 problem çözülmüştür. 54 adet ETD_ARP türündeki problemin 40 tanesi Dethloff'a, 14 tanesi ise Salhi ve Nagy'ye aittir.

5.2. Deneysel Sonuçlar

ETD_ARP veya KTD_ARP tipindeki sözü edilen test problemleri, önerilen GA_DKİ ve KSE_DKİ algoritmalarıyla 10 defa çözülmüştür. 10 denemede bulunan eniyi çözüm (EÇ) ve 10 denemenin bilgisayar hesaplama zamanı ortalamasından elde edilen çözüm süresi (ÇS), literatürdeki diğer algoritmaların sunduğu değerlerle karşılaştırılmıştır. Karşılaştırmada kullanılan algoritmalar literatürdeki çalışmalardan derlenmiş ve aşağıda sıralanmıştır. Bu algoritmalar ETD_ARP için son yıllarda önerilen, test problemlerinin literatürde bilinen eniyi çözümlerini (BEÇ) üreten veya bu tez kapsamında geliştirilen GA_DKİ ve KSE_DKİ gibi yığın tabanlı olan metasezgisellerdir.

- Tepkisel Tabu Arama (TTA) [Wassan ve ark., 2007]
- Büyük Komşuluk Arama (BKA) [Ropke ve Pisinger, 2006]
- Kuş Sürüsü Eniyileme (KSE) [Ai ve Kachitvichyanukul, 2009]
- Karınca Kolonisi Sistemi (KKS) [Gajpal and Apad, 2009]
- Paralel Yerel Arama (PYA) [Subramanian ve ark. 2009]
- Uyarlamalı Hafıza Metodolojisi (UHM) [Zachariadis ve ark., 2009]

Her problemin EÇ ve ÇS değerlerinin yanı sıra; her tabloda EÇ'nin kaç problemde literatürde bilinen eniyi çözümü (BEÇ) sağladığı, sapma yüzdesi (SY) ortalaması ve ortalama çözüm süresi (OÇS) değerine yer verilmiştir. Probleme ait SY'nin hesaplanmasında Eş.5.1 kullanılmaktadır.

$$SY = (EÇ - BEÇ) * 100 / BEÇ \quad (5.1)$$

Algoritmaların farklı programlama dillerinde kodlanması ve farklı bilgisayar ortamlarında koşturulmasından dolayı, çözüm zamanları açısından karşılaştırılmaları genellikle zor olmaktadır. Ayrıca, programlama becerisi de algoritmaların çözüm zamanı üzerinde büyük etkiye sahiptir. Dogarra (2006), standart doğrusal eşitlik yazılımı kullanarak farklı bilgisayarların performansını değerlendirmek için bir

çalışma yapmıştır. Dongarra'nın (2006) çalışmasında, birçok bilgisayarın *Mflops* (million floating point calculation per second) değeri verilmiştir. Son yıllarda araştırmacılar, farklı bilgisayar ortamlarında koşturulan algoritmaları çözüm zamanı açısından kabaca karşılaştırmak için Dongarra'nın (2006) çalışmasından faydalanmaktadır [Gajpal ve Abad, 2009]. Bu çalışmada bazı bilgisayarlar yer almamaktadır. Böyle bir durumda, ilgili bilgisayarın özelliklerine yakın bilgisayar dikkate alınmaktadır. Çizelge 5.1'nin sütunlarında sırasıyla deneysel çalışmada karşılaştırılan algoritmalar, her algoritmanın koşturulduğu bilgisayar ortamı, yaklaşık bilgisayar ortamı, yaklaşık bilgisayar için *Mflops* değeri ve Eş. 5.2'ye göre hesaplanan ölçeklendirme faktörü ω yer almaktadır. Eş. 5.2'de $Mflops_x$ ve $Mflops_a$ sırasıyla, ω değeri bulunacak ve bu çalışmada kullanılan bilgisayar ortamı için *Mflops* değerleridir. Karşılaştırmada kullanılan algoritmanların çözüm zamanlarının, bu çalışmada kullanılan bilgisayar ortamına göre (Intel Xeon 3.16 GHz hızında ve 1 GB RAM'e sahip Windows Vista Bussiness işletim sistemi) ölçeklendirilmesinde ω değerleri kullanılmaktadır.

$$\omega = Mflops_x / Mflops_a \quad (5.2)$$

Çizelge 5.1. Farklı bilgisayarlar için Mflops değerleri ve dönüştürme faktörü (ω)

Algoritma	Bilgisayar Ortamı	Yaklaşık Bilgisayar Ortamı	Mflops	ω
GA_DKİ & KSE_DKİ	Inter Xeon 3.16 Ghz	Intel Xeon 3.16 Ghz	981	1
TTA	Sun Fire V440 server-Ultra SPARC – III 1062 MHz	IBM eServer pSeries 630 6C4 1 proc (1000 MHz)	842	0.86
BKA	Pentium IV 1.5 GHz	Intel Pentium M 1.6 GHz	326	0.33
KSE	Intel P4 3.4 GHz	Intel Pentium 4 2.8 GHz	1317	1.34
KKS	Intel Xeon 2.4 GHz	Intel Xeon 2.4 GHz	884	0.90
UHM	Intel P4 2.4 GHz	Intel Pentium 4 2.53 GHz	1033	1.05
PYA*	Intel Xeon 2.66 GHz (256 Paralel İşlemci)	*	*	*

5.2.1. Parametrelerin belirlenmesi

Bu bölümde önerilen algoritmalara ait bazı parametreler için uygun değerlerin belirlenmesi amacıyla, test problemlerinin çözülmesinden önce yapılan çalışmadan bahsedilmektedir.

GA_DKİ'in parametreleri; yığın genişliği, başlangıç yığını ve çaprazlama operatörü olarak belirlenmiştir. Deneysel çalışmada yığın genişliği için 50, 100 ve 150 değerleri; çaprazlama operatörü olarak PMX, SJOX, SBOX, SJ2OX ve SB2OX dikkate alınmıştır. Başlangıç yığının oluşturulmasında ise 3 farklı yaklaşım denenmiştir. İlk yaklaşımda; başlangıç yığınının ilk iki kromozomu en yakın komşu (nearest neighbour) ve kazanç (saving) algoritmalarından, geri kalanı ise rassal olarak elde edilmiştir. Tüm yığın ikinci yaklaşımda Zachariadis ve ark. (2009) tarafından önerilen sezgisel yöntem ile, üçüncü yaklaşımda ise tamamen rassal olarak üretilmiştir. KSE_DKİ'in parametreleri ise; yığın genişliği, çaprazlama operatörü, başlangıç sıcaklığı ve γ , w , h_1 , h_2 olasılıkları olmak üzere 7 tanedir. Başlangıç sıcaklığı için 5, 10, 20, 30, 58 değerleri dikkate alınmıştır. Diğer parametrelerin her biri için 3'er düzey belirlenmiştir. Bunlar; yığın genişliği için 20, 30, 60, γ için 0.1, 0.3, 0.5, w için 0.1, 0.2, 0.3, h_1 ve h_2 içinse 0.6, 0.8, 1.0 değerleridir. Dikkate alınan çaprazlama operatörleri ise PMX, önerilen GA'da kullanılan SJ2OX ve Pan ve ark. (2008)'nin KKSE algoritmalarında kullandıkları PTL [Pan ve ark., 2008] operatörleridir.

Parametrelerin eniyi kombinasyonunu belirlemek amacıyla Dethloff'ın test problemlerinden 3'ü GA_DKİ'nin her parametre kombinasyonu için 10 kez çözülmüştür. Parametre kombinasyonunun fazla olması nedeniyle; bu problemlerden yalnızca 2 tanesi KSE_DKİ'nin deneysel tasarımında kullanılmış ve her parametre kombinasyonu 5 defa çözülmüştür. Dolayısıyla deneysel çalışmada $1350 (3 \times 3 \times 5 \times 3 \times 10)$ GA_DKİ, $36450 (5 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 2 \times 5)$ KSE_DKİ için olmak üzere toplam 37800 deneme yapılmıştır. Her denemede algoritma, ardışık 350 iterasyonda eniyi çözüm iyileşmediğinde durdurulmuştur.

GA_DKİ'nin varyans analizine (ANOVA) göre %5 anlamlılık düzeyinde algoritmanın ortalama çözüm kalitesinde yığın genişliğinin anlamlı bir etkisinin olduğu görülmüştür. Ancak, başlangıç yığının ve çaprazlama operatörünün çözüm kalitesi üzerinde anlamlı bir etkisi çıkmamıştır. Bunun en önemli nedeni çözümlerin DKİ algoritması ile iyileştirilmiş olmasıdır. Duncan'ın çoklu aralık testi yığın genişliği için uygulandığında 100 ve 150 aynı grupta yer almış ve bu genişlikler için ortalama çözüm kalitesinin yığın genişliği 50 seçildiğinde elde edilen değerden daha iyi olduğu görülmüştür. Sonuçta önerilen GA'da; başlangıç yığının tamamen rassal olarak üretilmesine, çaprazlama operatörü olarak SJ2OX kullanılmasına ve yığın genişliğinin 100 olmasına karar verilmiştir. KSE_DKİ için yapılan varyans analizine göre %5 anlamlılık düzeyinde başlangıç sıcaklığının (T^0) ve w değerinin çözüm kalitesi üzerinde anlamlı bir etkisinin olmadığı görülmüştür. Bu değerler sırasıyla $T^0 = 5$ ve $w = 0,2$ olarak kararlaştırılmıştır. Yığın genişliği ve γ parametrelerin değerleri beklendiği gibi çözüm kalitesiyle doğru orantılıdır. Ancak bu parametrelerin yüksek değerlerde olması çözüm kalitesinin yanı sıra hesaplama süresini de artırmaktadır. Bu nedenle $|Y^0|$ ve γ değerleri sırasıyla $|Y^0| = 60$ ve $\gamma = 0,3$ olarak belirlenmiştir. Çaprazlama operatörü sırasıyla PMX, SJ2OX ve PTL seçildiğinde, Duncan'ın çoklu aralık testine göre çözüm kalitesinin giderek azaldığı ortaya çıkmıştır. Bu nedenle çaprazlama operatörü ilk grupta yer alan PMX olarak belirlenmiştir. h_1 ve h_2 ikilisi dikkate alındığında çoklu aralık testine göre birinci grupta sırasıyla 1,0 ve 0,6; ikinci grupta h_1 için 0,8, h_2 için 0,8 ve 1,0 değerleri yer almaktadır. h_1 'in üçüncü gruptaki değeri ise 0.6'dır. Bu parametrelerin birbiriyle etkileşimleri dikkate alındığında, değerleri $h_1 = 0,6$ ve $h_2 = 0,6$ olarak belirlenmiştir. GA_DKİ ve KSE_DKİ için varyans analizi sonuçları sırasıyla EK-1 ve EK-2'de verilmiştir.

5.2.2. ETD_ARP için deneysel karşılaştırma sonuçları

Dethloff'ın test problemleri (2001) için sonuçlar

Literatürde dikkate alınan yöntemlerden Dethloff'ın test problemlerini kullananlar BKA, KSE, KKS, UHM ve PYA algoritmalarıdır. Dolayısıyla, Dethloff'ın test problemleri için bu beş algoritma kullanılarak GA_DKİ ve KSE_DKİ'nin performansı değerlendirilmiştir. Denemelerde GA_DKİ ve KSE_DKİ algoritmaları ardışık 350 iterasyon boyunca bulunan eniyi çözüm iyileştirilmezse sonlandırılmıştır.

Jin Ai ve Kachitvichyanukul (2009) çalışmalarında, Dethloff'ın 10 problemten oluşan her bir test problemi sınıfı için buldukları eniyi çözüm ortalamalarını sunmuşlardır [Jin Ai ve Kachitvichyanukul, 2009]. Bu nedenle, önerilen algoritmalarla KSE'nin karşılaştırması ayrı bir tabloda yapılmıştır. Çizelge 5.2'de, her problem grubu için 10 denemede KSE, GA_DKİ ve KSE_DKİ'nin elde ettiği ortalama eniyi çözüm değerleri yer almaktadır. Görüldüğü gibi GA_DKİ ve KSE_DKİ aynı sonuçları elde etmiş ve her ikisi de tüm problem grupları için KSE tarafından elde edilen sonuçları iyileştirmiştir. Bu nedenle önerilen algoritmalarla ait değerler tabloda (*) ile işaretlenmiştir. İyileştirme miktarı %0.35 ile %3.34 arasında değişmektedir. Ai ve Kachitvichyanukul (2009), KSE'nin her problem için ortalama 30 sn'de çözüme ulaştığını belirtmişlerdir. Ölçeklendirilmiş (Bkz. Çizelge 5.1) ortalama çözüm zamanı ise 40.2 sn'ye (30×1,34) denk gelmektedir. Çizelge 5.3'den görüleceği gibi GA_DKİ ve KSE_DKİ'nin ortalama çözüm zamanları ise sırasıyla 3,8 ve 3,05 sn'dir. Bu sonuçlar; önerilen GA_DKİ ve KSE_DKİ algoritmalarının her ikisinin de hem çözüm kalitesi hem de çözüm zamanı açısından, kendileri gibi yığın tabanlı olan KSE'den daha iyi bir performansa sahip olduğunu göstermektedir.

Çizelge 5.2. Dethloff'ın test problemleri için GA ile KSE'nin karşılaştırılması

Test Problemi Sınıfı	KSE	GA_DKİ	KSE_DKİ
SCA3	675,8	673,4*	673,4*
SCA8	1041,8	1028,15*	1028,15*
CON3	569,6	560,95*	560,95*
CON8	798,3	771,65*	771,65*

Çizelge 5.3’de, Dethloff’ın test kümesindeki tüm problemler için BKA, KKS, UHM, PYA ve önerilen algoritmalar tarafından elde edilen sonuçlar verilmektedir. Çizelge 5.3’de ilk sütun problemi vermektedir. Diğer sütunlar ise her problem için ilgilenilen algoritmanın EÇ ve ÇS değerlerini vermektedir. Tabloda literatürde bilinen en iyi çözümü sağlayan EÇ değerleri (*) ile işaretlenmiştir. Ropke ve Pisinger (2006), BKA ile elde ettikleri EÇ değerlerini ondalıktan sonra tek basamak kullanarak raporlamışlardır. Bu nedenle; BEÇ ondalıktan sonra tek değer olacak şekilde yuvarlandığında BKA’nın bulduğu EÇ değeri sağlanıyorsa, BEÇ’in bulunduğu kabul edilmiştir.

Çizelge 5.3’de görüldüğü gibi, GA_DKİ ve KSE_DKİ algoritmalarının her ikisi de UHM ve PYA gibi Dethloff’ın tüm test problemleri için BEÇ’e ulaşmıştır. Önerilen algoritmalar gibi yığın tabanlı olan KKS ise bu problemlerin ikisinde BEÇ’i bulamamıştır. BEÇ’in bulunamadığı problemler için KKS’nin SY değerleri sırasıyla %0,29 ve %0,08’dir. Ortalama SY değeri 0 olan algoritmalar çözüm süreleri açısından karşılaştırıldığında; dördünün ölçeklendirilmiş OÇS’nin diğerlerine göre çok kısa ve birbirine yakın ve görülmektedir. PYA’nın süresi 2,92 sn.’dir. Ancak bilindiği gibi PYA, 256 adet çekirdekte aynı anda çalıştırılan paralel algoritmadır. Bu nedenle bu algoritmanın çözüm zamanı, ölçeklendirilememiş ve dikkate alınamamıştır. Ayrıca, PYA paralel ortamda çalıştırıldığından dolayı OÇS’nin diğerlerinden daha küçük olması doğaldır. En kısa ölçeklendirilmiş OÇS 3,05 sn. ile önerilen KSE_DKİ’ye aittir. Onu sırasıyla 3,3 sn. ile UHM, 3,8 sn. ile GA_DKİ izlemektedir. KKS çözüm süresi açısından da 4. sırada olup, diğerlerinin yaklaşık 3 katı uzunluğunda süreye sahiptir. BKA hem çözüm süresi hem de çözüm kalitesi açısından diğerlerinden çok geridedir. BKA %0,32’lik ortalama SY ile 18 problemde BEÇ’i sağlayamadığı gibi, çözüm süresi de (52,1 sn.) diğerler algoritmalarından oldukça fazladır. Sonuç olarak; önerilen yığın tabanlı algoritmaların her ikisi de, 50 noktalı Dethloff problemleri için kaliteli çözümleri kısa sürelerde üretmekte ve literatürdeki yığın tabanlı diğer algoritmaalrı da hem çözüm kalitesi hem de süre açısından geçmektedir.

Çizelge 5.3. Dethloff'ın test problemleri için deneysel sonuçlar

Problem	BKA		KKS		UHM		PYA		GA_DKİ		KSE_DKİ	
	EÇ	ÇS	EÇ	ÇS	EÇ	EÇ	ÇS	ÇS	EÇ	ÇS	EÇ	ÇS
SCA3-0	636,1	232	635,62*	6	635,62*	2,5	635,62*	2,3	635,62*	6,7	635,62*	4,89
SCA3-1	697,8*	218	697,84*	6	697,84*	2,5	697,84*	2,3	697,84*	1,1	697,84*	0,77
SCA3-2	659,3*	203	659,34*	6	659,34*	2,9	659,34*	2,1	659,34*	0,5	659,34*	0,35
SCA3-3	680,6	241	680,04*	6,1	680,04*	2,3	680,04*	2,5	680,04*	2,2	680,04*	1,01
SCA3-4	690,5*	208	690,5*	5,7	690,5*	2,9	690,5*	2,2	690,5*	0,3	690,5*	0,26
SCA3-5	659,9*	226	659,9*	5,1	659,9*	3	659,9*	2,2	659,9*	3,1	659,9*	2,02
SCA3-6	651,1*	233	651,09*	6,1	651,09*	3,1	651,09*	2,5	651,09*	1,6	651,09*	0,75
SCA3-7	666,1	206	659,17*	6,8	659,17*	2,8	659,17*	2,4	659,17*	5	659,17*	1,56
SCA3-8	719,5*	190	719,47*	5,4	719,47*	3,5	719,48*	2,3	719,47*	0,5	719,47*	0,52
SCA3-9	681*	220	681*	6	681*	4,7	681*	1,9	681*	0,4	681*	0,81
SCA8-0	975,1	98	961,5*	11	961,5*	2,7	961,5*	3,4	961,5*	5,5	961,5*	4,83
SCA8-1	1052,4	95	1049,65*	11,5	1049,65*	3,8	1049,65*	2,9	1049,65*	7,8	1049,65*	6,81
SCA8-2	1044,5	94	1042,69	11,9	1039,64*	3,9	1039,64*	2,4	1039,64*	12,7	1039,64*	10,18
SCA8-3	999,1	94	983,34*	11,3	983,34*	2,6	983,34*	3	983,34*	4,7	983,34*	13,03
SCA8-4	1065,5*	93	1065,49*	11,1	1065,49*	2,4	1065,49*	2,8	1065,49*	4,4	1065,49*	2,95
SCA8-5	1027,1*	96	1027,08*	11,3	1027,08*	3,4	1027,08*	3,3	1027,08*	1,9	1027,08*	4,12
SCA8-6	977	94	971,82*	12	971,82*	2,7	971,82*	3,5	971,82*	2	971,82*	1,63
SCA8-7	1061	92	1052,17	12,5	1051,28*	5,1	1051,28*	3,1	1051,28*	6,5	1051,28*	3,43
SCA8-8	1071,2*	98	1071,18*	11	1071,18*	3,6	1071,18*	2,9	1071,18*	0,9	1071,18*	0,83
SCA8-9	1060,5*	92	1060,5*	11,5	1060,5*	4,8	1060,5*	2,2	1060,5*	6	1060,5*	7,29
CON3-0	616,5*	215	616,52*	8,3	616,52*	4,7	616,52*	3,1	616,52*	1,8	616,52*	2,10
CON3-1	554,5*	245	554,47*	7,1	554,47*	2,2	554,47*	2,8	554,47*	1,1	554,47*	1,27
CON3-2	521,4	232	518*	6,9	518*	3,1	518*	2,8	518*	6,6	518*	1,34
CON3-3	591,2*	231	591,19*	7,2	591,19*	3,2	591,19*	2,3	591,19*	0,6	591,19*	0,48
CON3-4	588,8*	221	588,79*	6	588,79*	2,3	588,79*	2,6	588,79*	4,5	588,79*	3,24
CON3-5	563,7*	209	563,7*	6,9	563,7*	3,7	563,7*	2,7	563,7*	0,3	563,7*	0,43
CON3-6	500,8	225	499,05*	7,3	499,05*	3,7	499,05*	2,8	499,05*	4,1	499,05*	2,34
CON3-7	576,5*	227	576,48*	7	576,48*	1,9	576,48*	2,8	576,48*	2,3	576,48*	2,57
CON3-8	523,1*	237	523,05*	7,4	523,05*	3,8	523,05*	2,5	523,05*	0,7	523,05*	0,95
CON3-9	586,4	207	578,25*	6,8	578,25*	2,2	578,25*	3,4	578,25*	1,8	578,25*	2,89
CON8-0	857,2*	94	857,17*	12,3	857,17*	4,4	857,17*	3,7	857,17*	7,3	857,17*	5,17
CON8-1	740,9*	94	740,85*	12	740,85*	3,3	740,85*	3	740,85*	1,6	740,85*	2,87
CON8-2	716	94	712,89*	13	712,89*	2,7	712,89*	3,1	712,89*	3	712,89*	2,12
CON8-3	811,1*	98	811,07*	13,9	811,07*	2,8	811,07*	4	811,07*	3,5	811,07*	2,77
CON8-4	772,3*	95	772,25*	11,9	772,25*	2,8	772,25*	3,7	772,25*	3,5	772,25*	3,63
CON8-5	755,7	94	754,88*	12,4	754,88*	5,7	754,88*	4,2	754,88*	9,5	754,88*	3,41
CON8-6	693,1	96	678,92*	12,4	678,92*	3,4	678,92*	4,1	678,92*	8	678,92*	7,94
CON8-7	814,8	94	811,96*	13	811,96*	2,5	811,96*	4	811,96*	3,8	811,96*	3,00
CON8-8	774	94	767,53*	12,5	767,53*	3,2	767,53*	3,4	767,53*	3,7	767,53*	3,20
CON8-9	809,3	92	809*	12,9	809*	3,8	809*	3,5	809*	1,9	809*	2,43
ortalama SY	0,32		0,01		0		0		0			0
BEÇ sayısı	22		38		40		40		40			40
OÇS (sn)		157,9		9,3		3,3		2,92		3,8		3,05
Ölçeklendirilmiş OÇS (sn)		52,1		8,4		3,5		—		3,8		3,05

Salhi ve Nagy'nin test problemleri (1999) için sonuçlar

Çizelge 5.4'de Salhi ve Nagy'nin (1999) test problemleri için elde edilen sonuçlar görülmektedir. Bu test problemleri için önerilen algoritmaların performansı TTA, BKA, KSE, KKS, UMH ve PYA olmak üzere literatürde bulunan 6 farklı algoritma ile karşılaştırılmalı olarak incelenmiştir. GA_DKİ ve KSE_DKİ iyileşme olmayan ardışık 350 iterasyonda durdurulmuştur. TTA ile CMT1Y problemi için elde edilen eniyi çözümün yanlış olduğu belirtilmiştir [Subramanian ve ark., 2009]. Bu nedenle, tabloda bu problem için TTA tarafından elde edilen sonuca yer verilmemiştir.

Çizelge 5.4'e göre; çözüm kalitesi açısından en iyi üç algoritma sırasıyla PYA, KSE_DKİ ve GA_DKİ olup, bu algoritmaların ulaştıkları BEÇ sayıları ve ortalama SY miktarları birbirlerine oldukça yakındır. Aynı algoritmalar çözüm süreleri açısından karşılaştırılmak istendiğinde, paralel bir algoritma olması nedeniyle PYA değerlendirmeye alınamamıştır. GA_DKİ tarafından bulunan EÇ değerlerini ufak bir farkla iyileştirse de, KSE_DKİ'nin çözüm süresinin çok uzun olduğu görülmektedir. GA_DKİ çözüm süresi açısından 142,1 sn. ile en iyi 3. algoritma iken, KSE_DKİ sonda yer almaktadır. Çözüm kalitesi açısından 4. sırada 5 BEÇ sayısı ile UMH yer almaktadır ve 19,3 sn. ile en kısa ölçeklendirilmiş OÇS'ne sahiptir. En kısa ikinci süre (87,8 sn.) ise önerilen algoritmalar gibi yığın tabanlı olan KKS'ye aittir. Ancak KKS sadece 2 problem için BEÇ'i bulabildiğinden, ortalama 0,92'lik sapma yüzdesiyle çözüm kalitesi açısından 5. sırada yer alabilmiştir. Önerilen algoritmalar dışında, tabloda yer alan ikinci yığın tabanlı algoritma olan KSE ise; hem 6,09'lık en büyük SY değeri ile en kötü algoritmadır, hem de çözüm süresi GA_DKİ'ninkinden uzundur. Kalan TTA ve BKA algoritmaları da çözüm kalitesi açısından sonlarda yer almaktadır. Genel olarak; GA_DKİ literatürdeki yığın tabanlı diğer algoritmalarından çok daha kaliteli, iyi algoritmalarla rekabet edebilen nitelikte çözümler üretmiştir. Çözüm kalitesine bağlı olarak hesaplama süresi diğer iki algoritmadan fazla çıksa da; eniyi çözüm süresine sahip UMH hesaplama süresini kısaltmakla birlikte kodlanması ve uygulanması çok zor olan özel bir kodlama yapısı kullanmaktadır. KSE_DKİ ise çözüm kalitesi açısından GA_DKİ'e göre küçük iyileştirmeler sağlasa da, bu durum çözüm sürelerinde büyük artışlara neden olmuştur.

Çizelge 5.4. Salhi ve Nagy'nin ETD_ARP türündeki test problemleri için deneysel sonuçlar

Problem	TTA		BKA		KSE		KKS		UMH		PYA		GA_DKİ		KSE_DKİ	
	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS
CMT1X	468,30	48	467*	221	467*	40	466,77*	5	469,80	2,1	466,77*	2,3	466,77*	3,1	466,77*	1,26
CMT1Y	-	-	467*	235	467*	40	466,77*	5	469,80	3,8	466,77*	2,3	466,77*	3,8	466,77*	1,40
CMT2X	668,77*	94	704	294	710	54	684,21	20,8	684,21	5,4	684,21	6,4	684,61	28,2	684,21	35,58
CMT2Y	663,25*	102	685	331	710	54	684,94	22,3	684,21	6,8	684,21	6,4	684,61	23,8	684,21	36,76
CMT3X	729,63	294	731	863	738	114	721,40	41,3	721,27*	11,9	721,27*	12,1	721,27*	64,8	721,27*	41,72
CMT12X	644,70*	242	685	684	691	115	663,01	36,3	662,22	9,3	662,22	10,3	662,22	68,4	662,94	141,36
CMT3Y	745,46	285	738	708	740	113	721,40	43,8	721,27*	11	721,27*	12,3	721,27*	64,5	721,27*	55,48
CMT12Y	659,52	254	675	539	697	114	663,50	39,3	662,22	4,8	662,22	10,8	663,50	71,1	663,50	105,44
CMT11X	861,97	504	837	1821	895	226	839,66	57,3	833,92*	21,2	833,92*	18,9	833,92*	126	833,92*	244,85
CMT11Y	830,39*	325	920	1376	900	238	840,19	52,8	833,92	14,4	833,92	19	833,92	193,7	833,92	368,92
CMT4X	876,50	558	879	1676	912	207	854,12	131,8	852,46*	29,6	852,46*	30,9	853,70	242,5	852,83	380,16
CMT4Y	870,44	405	876	1788	913	204	855,76	140,3	852,46*	27,4	852,46*	31,6	852,46*	356,3	852,46*	414,61
CMT5X	1044,51	483	1108	2340	1167	285	1034,87	377,5	1030,55	62,8	1029,25*	71,5	1029,25*	438,6	1029,25*	1539,17
CMT5Y	1054,46	533	1146	2177	1142	286	1037,34	393,5	1030,55	47,7	1029,25*	69,6	1030,97	304,7	1029,25*	1795,57
Ortalama SY	1,31		4,07		6,09		0,92		0,76		0,65		0,69		0,67	
BEÇ sayısı	5		2		2		2		5		9		7		8	
OÇS		299,7		1075,2		148,6		97,6		18,4		21,7		142,1		368,73
Ölçeklendirilmiş OÇS		257,7		354,8		199,1		87,8		19,3		*		142,1		368,73

5.2.3. KTD_ARP için deneysel karşılaştırma sonuçları

KTD_ARP için önerilen algoritmaların performansını incelemek amacıyla literatürde bu problemi çözmüş olan üç algoritma dikkate alınmıştır. Bu algoritmalar, KSE, BKA ve KKS'dir. Çizelge 5.5'da sonuçlar verilmektedir. KSE ve BKA algoritmaları tarafından, EÇ değerleri tamsayı olarak raporlamışlardır. Bu nedenle; BEÇ değeri tamsayıya yuvarlandığında bu algoritmaların ilgili problem için bulduğu EÇ elde ediliyorsa, BEÇ'in bulunduğu kabul edilmiştir. Ancak bu algoritmalara ait ortalama SY ve BEÇ sayılarının karşılaştırılması çok anlamlı olmamaktadır. Jin Ai ve Kachitvichyanukul (2009), KSE için çözüm zamanlarını raporlamamışlardır. Bu nedenle, ölçeklendirilmiş OÇS açısından KSE karşılaştırılamamıştır.

Çizelge 5.5. incelendiğinde, KSE_DKİ 16 BEÇ sayısı ve %0,12 ortalama SY, GA_DKİ ise 13 BEÇ sayısı ve %0,08 ortalama SY ile çözüm kalitesi açısından eniyi algoritmadır. KSE_DKİ 350 yerine iyileşme olmayan ardışık 250 iterasyon çalıştırılmasına rağmen en uzun çözüm süresine sahiptir. GA_DKİ ise 119,7 sn. ile KSE_DKİ'ye göre oldukça düşük bir ölçeklendirilmiş OÇS'ye sahiptir. Her iki algoritmayı diğer yığınlı tabanlı algoritmalarla karşılaştırdığımızda; beklediği gibi GA_DKİ ve KSE_DKİ'nin, ortalama 0,23 SY'ne sahip KKS'yi ve 2,37 SY ile sonda yer alan KSE'yi çözüm kalitesi açısından geçtiği görülmektedir. KSE'nin çözüm zamanları bulunmadığı için bir karşılaştırma yapılamamaktadır. Ancak GA_DKİ'nin ölçeklendirilmiş OÇS (119,7 sn) değeri, KKS'ye ait ölçeklendirilmiş OÇS'den (95,5) fazla, ancak yakındır. BKA algoritması ise 59,6 sn. ölçeklendirilmiş OÇS ile en kısa hesaplama süresine sahip algoritmadır.

GA_DKİ, CMT2Q ve CMT4Q problemleri için yeni BEÇ değerleri bulmuştur. KSE_DKİ ise hem CMT4Q probleminin yeni BEÇ değerlerini sağlamış, hem de eniyi çözüm değerlerini tamsayı olarak veren KSE ve BKA algoritmaları daha önce bulmamışsa CMT12Q ve CMT3H'nin bilinen eniyi çözüm değerlerini de iyileştirmiştir.

Çizelge 5.5. Salhi ve Nagy'nin KTD_ARP türündeki test problemleri için deneysel sonuçlar

Problem	KSE	BKA		KKS		GA_DKİ		KSE_DKİ	
	EÇ	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS	EÇ	ÇS
CMT1T	520*	520*	34	520,06*	7	520,06*	3.3	520,06*	1,53
CMT2T	810	783*	57	782,77*	26	782,77*	17.1	782,77*	14,73
CMT3T	827	798*	109	798,07*	42.6	798,07*	48	798,07*	88,77
CMT12T	792	788*	96	787,52*	52	787,52*	38.6	787,52*	26,48
CMT11T	1026	1000	164	998,8*	70.2	998,86	65	998,86*	65,31
CMT4T	1014	1000	212	990,39*	166.8	990,39*	165.5	990,39*	306,53
CMT5T	1297	1227*	333	1232,08	460.8	1229,67	687.2	1228,06	1541,31
CMT1Q	490*	490*	41	489,74*	6	489,74*	2.6	489,74*	1,09
CMT2Q	739	733	65	732,76	26.2	726,27*	19.5	732,76	19,96
CMT3Q	768	747*	128	747,15*	39.8	747,15*	44.3	747,15*	55,70
CMT12Q	733	729*	108	729,46	42	730,62	49.7	729,25*	121,10
CMT11Q	964	939*	196	939,36*	66.2	939,36*	63.9	939,36*	79,26
CMT4Q	938	918	244	913,93	153	913,63*	178.3	913,63*	455,20
CMT5Q	1174	1119*	381	1134,72	451.8	1120,78	241.4	1123,46	1326,58
CMT1H	464*	465	51	465,02	5.6	465,02	3.2	465,02	2,08
CMT2H	668	663*	78	662,63*	22	662,63*	17.9	662,63*	22,93
CMT3H	701*	701*	186	701,31	35.6	701,31	44.5	700,94*	60,01
CMT12H	635	629*	150	629,37	32.8	629,02*	52.4	629,02*	135,25
CMT11H	830	818*	303	820,35	45.8	818,05*	76.1	818,05*	128,18
CMT4H	883	829*	345	831,39	125.4	831,39	150.5	831,04	273,74
CMT5H	1044	983*	514	992,37	351.4	986,2	544.4	988,72	1855,36
Ortalama SY	2.37	0.14		0.23		0.08		0.12	
BEÇ sayısı	4	16		10		13		15	
OÇS			180.7		106.1		119.7		313,38
Ölçeklendirilmiş OÇS			59.6		95.5		119.7		313,38

Vural'ın (2003) yüksek lisans tezindeki GA'ya dayalı yaklaşımı [Vural, 2003] ile bu tez kapsamında önerilen GA_DKİ'nin performanslarının kısaca değerlendirilmesinde fayda vardır. Vural algoritmasının performansını Min'in (1989) ve Dethloff'ın (2001) test problemleri üzerinde incelemiştir. Tezde Min'in (1989) problemi için elde edilen eniyi çözümün 89 olduğu ve bu çözüme 1 dakikadan (Pentium III 500 MHz) daha az sürede ulaşıldığı rapor edilmiştir [Vural, 2003]. Dethloff'ın test problemleri içinse SY'nin 10 civarında olduğu sunulmuş, ancak çözüm zamanı bilgisi verilmemiştir. Önerilen GA_DKİ ise, Dethloff'ın tüm problemlerinde eniyi çözüme (Bkz. Çizelge 5.3) ulaşmaktadır.

6. SONUÇ

Küreselleşmenin etkisiyle son yıllarda işletmeler dünya pazarlarına açılma çabası içine girmişlerdir. İşletmelerin global pazarın çetin rekabet koşullarında varlıklarını sürdürebilmeleri ise planlama, tasarım, çizelgeleme, maliyetlendirme gibi faaliyetlerde eniyileme kavramını uygulayabilmelerine bağlıdır.

Pazarın genişlemesiyle, lojistik faaliyetlerinin etkin bir şekilde gerçekleştirilmesi giderek önem kazanmıştır. Çünkü etkin bir lojistik yönetimi; maliyetlerin düşmesini, üretimin artmasını, kalitenin yükselmesini ve müşteri memnuniyetini sağlamaktadır. İyi tasarlanmadığında yüklü maliyetlere neden olan lojistik faaliyetlerinden birisi de “*Araç Rotalama*” olup, küresel rekabet ortamında işletmeler için stratejik bir önem arz etmektedir. Araç rotalama sisteminin görevi tedarik noktasından alınan ürünlerin işletmeye getirilmesi ve ürünlerin depo ya da müşteriye ulaştırılmasıdır. Ancak son zamanlarda müşterilerin bilinçlenmesi, geri dönüşümün artması, müşteri memnuniyetini sağlamak adına ekstra hizmetlerin verilmeye başlanması, devletin çevresel kaygılar sonucu atıkların toplanması konusunda yaptırımlar uygulaması gibi nedenlerle araç rotalama faaliyetlerinde müşterilerden ürünlerin toplanması da söz konusu olmaktadır. Bu problem literatürde Topla Dağıt Araç Rotalama Problemi (TD_ARP) olarak adlandırılır. TD_ARP toplama ve dağıtım faaliyetlerinin gerçekleştirilmesine göre 3 sınıfa ayrılmaktadır. Bunlardan birisi Eşzamanlı Topla-Dağıt Araç Rotalama Problemidir (ETD_ARP). Tez kapsamında dikkate alınan ETD_ARP’de bir müşteri ziyaret edildiğinde hem dağıtım hem de toplama faaliyeti gerçekleştirilir ve amaç bu faaliyetleri içeren enküçük maliyetli araç rotalarının belirlenmesidir.

Literatürde ETD_ARP’nin çözümü için eniyi çözümü veren kesin yöntemler ve sezgisel yaklaşımlar önerilmiştir. Ancak ETD_ARP NP-zor yapıda bir problem olduğu için eniyi çözümü veren yaklaşımlar sadece küçük boyutlu problemlere uygulanabilmektedir. Bu nedenle problemin çözümünde daha az işlem, daha kısa hesaplama süresi ile eniyi çözüme yakın, iyi kalitede çözümler üreten yaklaşık yöntemler tercih edilmektedir. Özellikle tasarruf ve süpürme algoritması gibi klasik

sezgisel yöntemler yerine *metasezgisel* olarak adlandırılan ve ARP için iyi çözümler üretebilen yaklaşık yöntemler araştırmacıların ilgi odağı haline gelmiştir. Bu nedenle, literatürde ARP'nin çeşitli türleri için farklı metasezgisellere dayalı olarak çözüm yöntemleri geliştirilmiştir.

Literatürdeki ETD_ARP için geliştirilen metasezgisellerin çoğu tabu aramayı kullanmaktadır. Tabu arama, son çalışmalarda daha çok diğer algoritmalarla melezleştirilerek uygulanmıştır. Çünkü bir metasezgiselin başka bir metasezgisel veya basit bir yerel arama algoritması ile beraber kullanılması olarak ifade edilen melez algoritmalar daha iyi performansa sahip olabilmektedir. Literatür incelendiğinde Genetik Algoritma (GA) ve Kuş Sürüsü Eniyileme (KSE) gibi popülasyon tabanlı metasezgisellerin kaliteli sonuçlar üretmemesi nedeniyle ETD_ARP'nin çözümü için çok tercih edilmediği görülmektedir. Bu tez kapsamında ETD_ARP'nin çözümü için; Değişken Komşu İniş (DKİ) ile melezleştirilmiş GA ve KSE'e dayalı GA_DKİ ve KSE_DKİ algoritmaları geliştirmiştir. Amaç çözüm uzayını daha etkin aramak için yığın ve yerel arama tabanlı iki ayrı metasezgiselin avantajından aynı anda faydalanmaktır. ETD_ARP için GA, KSE ve DKİ algoritmalarına dayalı melez algoritmalar ilk defa bu tez kapsamında önerilmiştir.

Yapılan deneysel çalışmalarla önerilen GA_DKİ algoritmasının performansı ETD_ARP için önerilen literatürdeki diğer algoritmalarla çözüm kalitesi ve çözüm zamanı açısından karşılaştırmalı olarak incelenmiştir. Karşılaştırmalar sonucunda GA_DKİ'nin çözüm kalitesi açısından son yıllarda yayınlanan ve oldukça iyi çözümler bulabilen PYA ve UHM algoritmaları kadar iyi olduğu görülmüştür. PYA'nın paralel bilgisayar mimarisinde çalıştığı göz önüne alınırsa, GA_DKİ hesaplama süresi daha uzun olsa da uygulanması çok daha kolay bir algoritmadır. UHM algoritmasında hesaplama süresinin kısa olması için özel bir kodlama yapısı kullanılması çözüm zamanını azaltmıştır. Ancak bu kodlama yapısının kullanılması pratikte oldukça güçtür. Çözüm zamanı açısından ise, GA_DKİ'in kabaca TTA, BKA ve KSE algoritmalarından daha kısa zamanlarda, çok daha kaliteli sonuçlara ulaştığı görülmektedir. Önerilen KSE_DKİ algoritması ise en az GA_DKİ kadar iyi

sonuçlar elde etse de, bu durum algoritmanın çözüm zamanında büyük artışlara neden olmuştur.

Genel olarak önerilen algoritmaların her ikisi de, kendileri gibi yığın tabanlı olan literatürdeki KSE ve KKS algoritmalarından çok daha kaliteli çözümler üretmektedir. Ancak GA_DKİ kabul edilebilir bir çözüm zamanına sahip olsa da, KSE_DKİ'nin çözüm zamanı oldukça fazladır. Bunun en önemli nedeni, KSE'nin her iterasyonunda yığındaki tüm çözümlerin güncellenmesidir. Buna rağmen KSE_DKİ literatürdeki diğer KSE algoritmasıyla rekabet edebilecek güçtedir. İlk kez bu tez kapsamında ETD_ARP için kesikli KSE yöntemi kullanılmış ve KSE ile Tavlama Benzetimi (TB) algoritmaları melezleştirilmiştir.

Sonuç olarak bu tezde literatüre ETD_ARP'nin çözümü için DKİ ile melezlenmiş GA ve KSE'ye dayalı iki algoritma sunulmuştur. Çözümlerin permütasyon kodlama ile gösterildiği bu algoritmalarda, Prins (2004) tarafından KARP için önerilen ayırma algoritması, yine ilk kez ETD_ARP için uyarlanmıştır.

Gelecek çalışmalar

- Genetik Algoritma ve Kuş Sürüsü Eniyileme algoritmaları, performansı artırmak amacıyla başka metasezgisel yöntemlerle melezleştirilebilir.
- Zaman penceresi ve/veya rotanın gidebileceği enbüyük mesafe gibi daha fazla operasyonel kısıt probleme eklenerek, geliştirilen algoritmalar çözümü için uyarlanabilir.
- Önerilen algoritmalar problemin heterojen araç filolu veya çok depolu hali için uygulanabilir.
- ETD_ARP 'nin Karınca Kolonisi Eniyilemesi ve Diferansiyel Gelişim Algoritması gibi başka yığın tabanlı metasezgisellerle çözümü için çalışılabilir.

KAYNAKLAR

Anghinolfi, D., Paolucci, M., “A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times”, *European Journal of Operational Research*, 193: 73-85 (2009).

Bean, J.C., “Genetic algorithms and random keys for sequencing and optimization”, *ORSA Journal on Computing*, 6: 154-160 (1994).

Beasley, J. E., “Route first-cluster second methods for vehicle routing”, *Omega*, 11: 403-408 (1983).

Bianchessi, N., Righini G., “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery”, *Computers and Operations Research*, 34(2): 578 (2007).

Brandao, J., “A tabu search algorithm for the open vehicle routing problem”, *European Journal of Operation Research*, 157: 552-564 (2004).

Breedam, A.V., “A parametric analysis of heuristics for the vehicle routing problem”, *European Journal of Operation Research*, 137: 348-370 (2002).

Chen, J. F., Wu., T. H., “Vehicle routing problem with simultaneous deliveries and pickups”, *The journal of the Operational Research Society*, 57(5): 579-587 (2006).

Chen, J.F., “Approaches for the vehicle routing problem with simultaneous deliveries and pickups”, *Journal of the Chinese Institute of Industrial Engineers*, 23 (2): 141-150 (2006).

Christofides, N., Mingozzi, A., Toth, P., “Contributions to the quadratic assignment problem”, *European Journal of Operational Research*, 4: 243-247 (1980).

Clarke, G., Wright, W., “Scheduling of vehicles from a central depot to a number of deliver points”, *Operational Research*, 12: 568-581 (1964).

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

Crispim, J., Brandao, J., “Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls”, *Journal of the Operational Research Society*, 56: 1296-1302 (2005).

DellAmico, M., Righini, G., Salani, M., “A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection”, *Transportation science*, 40(2): 235-247 (2006).

Dethloff, J., “Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up”, *OR Spektrum*, 23(1): 79-96 (2001).

Erol, V., “Araç rotalama problemleri için popülasyon ve komşuluk tabanlı metasezgisel bir algoritmanın tasarımı ve uygulaması”, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Sistem Mühendisliği*, İstanbul, 3-120 (2006).

Eryavuz, M., Gencer, C., “Araç rotalama problemine ait bir uygulama”, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi*, 139-155 (2001).

Gajpal, Y., Abad, P., “An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup”, *Computers & Operations Research*, 36(12): 3215–3223 (2009).

Gillet, B., Miller, L., “A heuristic for the vehicle dispatching problem”, *Operations Research*, 22: 340-349 (1974).

Gökçe, E.I., “A revised ant colony system approach to vehicle routing problems”, Yüksek Lisans Tezi, *Sabancı Üniversitesi Fen ve Doğa Bilimleri Enstitüsü*, Ankara, 55-65 (2004).

Hertz, A., Widmer, M., “Guidelines For The Use Of Meta-Heuristics In Combinatorial Optimization”, *European Journal of Operation Research*, 151 (2), 247-252 (2003).

Jarboui, B., Ibrahim, S., Siarry, P., Rebai, A., “A combinatorial particle swarm optimization for solving permutation flowshop problems”, *Computers and Industrial Engineering*, 54: 526-538 (2008).

Jin Ai, T., Kachitvichyanukul, V., “A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery”, *Computers and Operations Research*, 36:1693-1702 (2009).

Karaoğlu, İ., “Dağıtım ağları tasarımında yer seçimi ve eşzamanlı topla-dağıt araç rotalama problemleri”, Doktora Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 1-40 (2009).

Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F., “Classical And Modern Heuristics For The Vehicle Routing Problem”, *International Transactions in Operational Research*, 7: 285-300 (2000).

Min, H., “The multiple vehicle-routing problem with simultaneous delivery and pick-up Points”, *Transportation Research. Part A, Policy and Practice*, 23(5): 377–386 (1989).

Montane, F.A.T., Galvao, R.D., “A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service”, *Computers & Operations Research*, 33:595-619 (2006).

Mladenovi, N., Hansen, P., “Variable neighborhood search”, *Computers & Operations Research*, 24(11):1097-1100 (1997).

Nagy G, Salhi S., “Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries”, *European Journal of Operational Research*, 162:126–141 (2005).

Pan, W.K., Tasgetiren, M.F., Liang, Y.C., “A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem”, *Computers and Operations Research*, 35(9): 2807-2839 (2008).

Prins, S., “A simple and effective evolutionary algorithm for the vehicle routing problem”, *Computers & Operations Research*, 31: 1985–2002, (2004).

Ropke, S., D. Pisinger., “A unified heuristic for a large class of vehicle routing problems with backhauls”, *European Journal of Operational Research*, 171(3): 750-775 (2006).

Ruiz, R., Maroto, C., Alcaraz, J., “Two new robust genetic algorithms for the flowshop scheduling problem”, *Omega*, 34: 461-476 (2006).

Salhi, S., Nagy, G., “A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling”, *Journal of the Operational Research Society*, 50(10):1034–1042 (1999).

Sha, D.Y., Hsu, C.Y., “A new particle swarm optimization for the open shop scheduling problem”, *Computers and Operations Research*, 35: 3243-3261 (2008).

Subramanian, A., Drummonda, L.M.A., Bentes, C., Ochi, L.S., Farias, R., “A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery”, *Computers & Operations Research*, 37(11): 1899-1911 (2009)

Şeker, Ş., “Araç rotalama problemi ve zaman pencereli stokastik araç Rotalama problemine genetik algoritma yaklaşımı”, Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi FBE Endüstri Mühendisliği Anabilim Dalında Endüstri Mühendisliği Bölümü*, İstanbul, 31-95 (2007).

Tang, F.A., Galvao, R.D., “Vehicle routing problem with simultaneous pick-up and delivery service”, *Journal of the Operational Research Society of India*, 39:19-33 (2002).

Taşgetiren, M.F., Liang, Y.C., Şevkli, M., Gençyilmaz, G., “Particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem”, *European Journal of Operational Research*, 177:1930-1947 (2007).

Ulusoy, G., “The fleet size and mixed problem for capacitated arc routing”, *European Journal of Operational Research*, 22: 833-839 (1987).

Van Breedam, A., “Improvement heuristics for the vehicle routing problem based on simulated annealing”, *European Journal of Operational Research*, 86(3):480–490 (1995).

Vural, A., “A GA based meta-heuristic for capacited vehicle routing problem with simultaneous pick-up and deliveries”, Yüksek Lisans Tezi, *Sabancı Üniversitesi Fen ve Doğa Bilimleri Enstitüsü*, Ankara, 1-40 (2003).

Wassan, N. A., Wassan, A. H., Nagy, G., “A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries”, *Journal of Combinatorial Optimization*, 15(4):368–86 (2007).

Waters, C. D. J., “A solution procedure for the vehicle scheduling problem based on iterative route improvement”, *Journal of Operational Research Society*, 38: 833-839 (1987).

Zachariadis, E. E., Tarantilis, C. D., Kiranoudis, C. T., “A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service”, *Expert Systems with Applications*, 36 (2): 1070-1081 (2009a).

Zachariadis, E. E., Tarantilis, C. D., Kiranoudis, C. T., “An adaptive memory methodology optimization for the vehicle routing problem with simultaneous pick-ups and deliveries”, *European Journal of Operational Research*, 202(2): 401-411 (2009b).

Zhang, C., Ning, J., Ouyang, D., “A hybrid alternate two phases particle swarm optimization algorithm for flowshop scheduling problem”, *Computers and Industrial Engineering*, 58(1): 1-11 (2010).

EKLER

EK-1. GA_DKİ'nin parametreleri için varyans analizi sonuçları

Çizelge 1.1. GA_DKİ'nin parametreleri için varyans analizi sonuçları

Source	Type III Sum of Squares	DF	Mean Square	F-values	P-values
Corrected Model	163681,919 ^a	46	3558,303	147,728	0
Intercept	1,49E+09	1	1,49E+09	6,17E+07	0
Başlangıç yığını	19,172	2	9,586	0,398	0,672
çaprazlama	137,431	4	34,358	1,426	0,223
Yığın genişliği	175,093	2	87,547	3,635	0,027
problem	162071,48	2	81035,74	3,36E+03	0
Başlangıç yığını *	241,191	8	30,149	1,252	0,265
çaprazlama	267,943	4	66,986	2,781	0,026
Başlangıç yığını *	97,632	4	24,408	1,013	0,399
Yığın genişliği	260,912	8	32,614	1,354	0,213
Başlangıç yığını *	135,735	8	16,967	0,704	0,688
çaprazlama *	275,33	4	68,833	2,858	0,023
problem	31385,127	1303	24,087		
Error	1,49E+09	1350			
Total	195067,046	1349			
Corrected Total					

a. R Squared = ,839 (Adjusted R Squared = ,833)

Çizelge 1.2. GA_DKİ'nin parametreleri için Duncan'ın çoklu aralık testi

Factors	N	Mean	Duncan grouping	Levels
Başlangıç yığını	450	1,05E+03	A	Y2
	450	1,05E+03	A	Y1
	450	1,05E+03	A	Y3
çaprazlama	270	1,05E+03	A	SJ2OX
	270	1,05E+03	A	SB2OX
	270	1,05E+03	A	SBOX
	270	1,05E+03	A	PMX
	270	1,05E+03	A	SJOC
Yığın genişliği	450	1,05E+03	A	150
	450	1,05E+03	A	100
	450	1,05E+03	B	50
problem	450	1,04E+03	A	P2
	450	1,05E+03	B	P1
	450	1,06E+03	C	P3

EK-2. KSE_DKİ'nin parametreleri için varyans analizi sonuçları

Çizelge 2.1. KSE_DKİ'nin parametreleri için varyans analizi sonuçları

Source	Type III Sum of Squares	DF	Mean Square	F-values	P-values
Model	2,744E10 ^a	142	1,932E8	3,682E7	,000
problem	1,538E9	1	1,538E9	2,931E8	,000
çaprazlama	4426,931	2	2213,466	421,788	,000
sıcaklık	11,538	4	2,885	,550	,699
yığıngenışliğı	12582,812	2	6291,406	1,199E3	,000
olasılık	29756,831	2	14878,415	2,835E3	,000
w	1,191	2	,595	,113	,893
h ₁	102,832	2	51,416	9,798	,000
h ₂	88,897	2	44,448	8,470	,000
problem * çaprazlama	291,775	2	145,887	27,800	,000
problem * sıcaklık	13,724	4	3,431	,654	,624
problem * yığıngenışliğı	331,614	2	165,807	31,595	,000
problem * olasılık	388,367	2	194,184	37,003	,000
problem * w	1,448	2	,724	,138	,871
problem * h ₁	30,533	2	15,266	2,909	,055
problem * h ₂	12,497	2	6,248	1,191	,304
çaprazlama * sıcaklık	28,463	8	3,558	,678	,711
çaprazlama * yığıngenışliğı	84,210	4	21,052	4,012	,003
çaprazlama * olasılık	688,826	4	172,207	32,815	,000
çaprazlama * w	31,720	4	7,930	1,511	,196
çaprazlama * h ₁	21,036	4	5,259	1,002	,405
çaprazlama * h ₂	38,610	4	9,653	1,839	,118
sıcaklık * yığıngenışliğı	27,746	8	3,468	,661	,726
sıcaklık * olasılık	46,940	8	5,868	1,118	,347
sıcaklık * w	37,211	8	4,651	,886	,527
sıcaklık * h ₁	19,425	8	2,428	,463	,883
sıcaklık * h ₂	27,249	8	3,406	,649	,737
yığıngenışliğı * olasılık	720,593	4	180,148	34,328	,000
yığıngenışliğı * w	42,853	4	10,713	2,041	,086
yığıngenışliğı * h ₁	52,397	4	13,099	2,496	,041
yığıngenışliğı * h ₂	34,080	4	8,520	1,624	,165
olasılık * w	15,003	4	3,751	,715	,582
olasılık * h ₁	23,519	4	5,880	1,120	,345
olasılık * h ₂	54,460	4	13,615	2,594	,035
w * h ₁	35,431	4	8,858	1,688	,150
w * h ₂	44,364	4	11,091	2,113	,076
h ₁ * h ₂	16,098	4	4,025	,767	,547
Error	190537,608	36308	5,248		
Total	2,744E10	36450			
a. R Squared = 1,000 (Adjusted R Squared = 1,000)					

EK-2. (Devam) KKSE'nin parametreleri için varyans analizi sonuçları

Çizelge 2.2. KKSE'nin parametreleri için Duncan'ın çoklu aralık testi

Factors	N	Mean	Duncan grouping	Levels
çaprazlama	12150	8,4257E2	A	PMX
	12150	8,4295E2	B	SJ2OX
	12150	8,4342E2	C	PTL
başlangıç sıcaklığı	7290	8,4295E2	A	58
	7290	8,4297E2	A	10
	7290	8,4299E2	A	5
	7290	8,4299E2	A	20
	7290	8,4300E2	A	30
yığın genişliği	12150	8,4219E2	A	60
	12150	8,4315E2	B	30
	12150	8,4360E2	C	20
olasılık	12150	8,4195E2	A	0.5
	12150	8,4284E2	B	0.3
	12150	8,4415E2	C	0.1
w	12150	8,4298E2	A	0.2
	12150	8,42979E2	A	0.3
	12150	8,4299E2	A	0.1
h ₁	12150	8,4292E2	A	1.0
	12150	8,4298E2	B	0.8
	12150	8,43051E2	C	0.6
h ₂	12150	8,4292E2	A	0.6
	12150	8,4299E2	B	0.8
	12150	8,4304E2	B	1.0

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÖKSAL, Fatma Pınar
Uyruğu : T.C.
Doğum tarihi ve yeri : 15.02.1983 Ankara
Medeni hali : Bekar
Telefon : 0 (533) 747 32 35
e-mail : pinargoksal@yahoo.com.

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /Endüstri Müh.	2010
Lisans	Gazi Üniversitesi /Endüstri Müh	2006
Lise	Ankara Atatürk Lisesi	2001

Yabancı Dil

İngilizce