

Yeni Bir Kapasiteli Ayrıt Rotalama Problemi ve Çözüm Yaklaşımları

Gökhan Kirlik

**YÜKSEK LİSANS TEZİ**

Endüstri Mühendisliği Anabilim Dalı

Aralık 2009

A New Capacitated Arc Routing Problem & Solution Approaches

Gökhan Kirlik

**MASTER OF SCIENCE THESIS**

Department of Industrial Engineering

December 2009

Yeni Bir Kapasiteli Ayrıt Rotalama Problemi ve Çözüm Yaklaşımları

Gökhan Kirlik

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Endüstri Mühendisliği Anabilim Dalı  
Yöneylem Araştırması Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Aydın Sipahioğlu

Aralık 2009

## ONAY

Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Gökhan Kirlik'in YÜKSEK LİSANS tezi olarak hazırladığı “Yeni Bir Kapasiteli Ayrıt Rotalama Problemi ve Çözüm Yaklaşımları” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Yrd. Doç. Dr. Aydın Sipahioğlu

**İkinci Danışman** : -

### **Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Doç. Dr. Metin Türkay

**Üye** : Doç. Dr. Osman Parlaktuna

**Üye** : Doç. Dr. Müjgan Sağır Özdemir

**Üye** : Yrd. Doç. Dr. R. Aykut Arapoğlu

**Üye** :

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

## ÖZET

Kapasiteli ayrıt rotalama problemi (KARP), bir serimdeki gezilmesi gereken bütün ayrıtlara kapasite kısıtlı araçlarla toplamda en küçük maliyetli olacak şekilde servis verecek araç turlarının belirlenmesi olarak tanımlanır. KARP'ta araçlar gezilmesi gereken ayrıtlara servis verirken kapasite kullanmaktadır. Bu çalışmada, araçların servis verirken kullandığı kapasitenin yanı sıra servis vermeden geçtiği ayrıtlarda da kapasite kullandığı yeni bir KARP türü incelenmiştir. Tanımlanan KARP türündeki ayrıt talebi, ayrıta servis verilip verilmemesine göre değiştiğinden bu problem değişken ayrıt talepli kapasiteli ayrıt rotalama problemi (DAT-KARP) olarak adlandırılmıştır. Bu çalışmada, DAT-KARP için bir matematiksel model geliştirilmiştir. Ayrıca tanımlanan problemin büyük boyutlu problemlerde çözülebilmesi için KARP için önerilmiş olan Ulusoy'un bölümlendirme sezgiseli probleme uyumlandırılmış ve bir genetik algoritma önerilmiştir. KARP için önerilmiş test problemleri DAT-KARP'a uygun hale getirilmiş ve önerilen yöntemlerin etkinliği bu test problemleri üzerinde gösterilmiştir.

Anahtar Kelimeler: Kapasiteli ayrıt rotalama problemi, değişken ayrıt talebi, matematiksel model, Ulusoy'un bölümlendirme sezgiseli, genetik algoritma.

## SUMMARY

Capacitated arc routing problem (CARP) is the determination of vehicle tours that serve all required edges exactly in a network by a capacity constraint vehicles with minimization of total cost of all vehicle tours. In CARP, vehicles use capacity while servicing required edges. In this study, a new CARP variation is examined in which vehicles use capacity while traversing an edge whether servicing or not. In the new CARP formulation, the demand of an edge is varied depends on the service status, so this problem called as capacitated arc routing problem with varying arc demands (VAD-CARP). In this study, a mathematical model is proposed for the VAD-CARP. For the large scale problems Ulusoy's partitioning heuristic is modified and a genetic algorithm is proposed for the new CARP variation. Effectiveness of the proposed methods is shown on test problems for VAD-CARP which are obtained by modifying CARP test problem instances.

Keywords: Capacitated arc routing problem, varying arc demands, mathematical model, Ulusoy's partitioning heuristic, genetic algorithm.

## TEŞEKKÜR

Gerek derslerimde ve gerekse tez çalışmalarında, bana danışmanlık ederek, beni yönlendiren ve her türlü olanağı sağlayan danışmanım Yrd. Doç. Dr. Aydın Sipahioğlu'na teşekkürü bir borç bilirim. Ayrıca danışmalıklarından ve tezimdeki katkılarından dolayı Doç. Dr. Osman Parlaktuna, Yrd. Doç. Dr. Ahmet Yazıcı ve Yrd. Doç. Dr. Metin Özkan'a da teşekkürlerimi sunarım.

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'nun (TUBİTAK) 107E064 numaralı "Devinik İç Ortamların Tam Kapsanması Amacıyla Gezgin Robot Rota Planlaması" projesi tarafından desteklenmiştir.

## İÇİNDEKİLER

### Sayfa

<b>ÖZET .....</b>	<b>v</b>
<b>SUMMARY .....</b>	<b>vi</b>
<b>TEŞEKKÜR .....</b>	<b>vii</b>
<b>İÇİNDEKİLER .....</b>	<b>viii</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>xi</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>xiii</b>
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. ROTALAMA PROBLEMLERİ.....</b>	<b>4</b>
2.1. Gösterim ve Tanımlamalar.....	4
2.2. Düğüm Rotalama Problemleri .....	6
2.3. Ayırıt Rotalama Problemleri .....	8
2.4. Genel Rotalama Problemi .....	10
<b>3. YÖNSÜZ TEK ARAÇLI POSTACI PROBLEMLERİ.....</b>	<b>12</b>
3.1. Yönsüz Çinli Postacı Problemi .....	12
3.1.1. Yönsüz Çinli postacı probleminin matematiksel modeli.....	13
3.1.2. Yönsüz Çinli postacı problemi için sezgisel algoritma .....	15
3.1.2.1. En küçük maliyetli eşleştirme .....	16
3.1.2.2. Euler turu .....	19
3.2. Yönsüz Kırsal Postacı Problemi .....	20
3.2.1. Yönsüz KPP'nin matematiksel modeli.....	22
3.2.2. Yönsüz kırsal postacı problemi için sezgisel algoritma .....	24
<b>4. KAPASİTELİ AYRIT ROTALAMA PROBLEMİ .....</b>	<b>29</b>
4.1. Matematiksel Model .....	29



## İÇİNDEKİLER (devam)

### Sayfa

4.2. Çözüm Yöntemleri.....	33
4.2.1. Alt sınırlar ve kesin çözüm yöntemleri.....	34
4.2.2. Sezgisel algoritmalar .....	34
4.2.2.1. Yapıcı metotlar .....	35
4.2.2.2. İki fazlı oluşturma metotları .....	36
4.2.3. Metasezgisel algoritmalar.....	39
4.3. Kapasiteli Ayırıt Rotalama Problemi Türleri.....	41
4.3.1. Farklı serim türleri için KARP .....	41
4.3.2. Alternatif amaç fonksiyonlu KARP .....	42
4.3.3. Çoklu depolu KARP .....	44
4.3.4. Hareketli depolu KARP.....	45
4.3.5. Periyodik KARP .....	45
4.3.6. Stokastik KARP.....	46
4.3.7. Araç/Yön bağımlı KARP .....	46
4.3.8. Zaman Pencereleli KARP.....	47
4.3.9. Parçalı göndermeli KARP .....	48
4.3.10. Dinamik KARP .....	48
<b>5. YENİ BİR KAPASİTELİ AYIRIT ROTALAMA PROBLEMİ .....</b>	<b>50</b>
5.1. Motivasyon (Gezgin Robotlarla Kapsama Problemi).....	50
5.2. Yeni KARP İçin Bir Matematiksel Modeli.....	57
5.3. Yeni KARP İçin Önerilen Değiştirilmiş Ulusoy Algoritması .....	60
5.4. Yeni KARP İçin Geliştirilen Genetik Algoritma .....	63
5.4.1. Kodlama .....	64
5.4.2. Uyum değerinin hesaplanması .....	68
5.4.3. Genetik algoritma operatörleri .....	73

**İÇİNDEKİLER (devam)**

	<b><u>Sayfa</u></b>
<b>6. MOBİL ROBOTLARDA UYGULAMA .....</b>	<b>76</b>
6.1. Uygulamalardan Tanımlamalar.....	76
6.2. Uygulamalar ve Test Sonuçları.....	79
<b>7. ÖNERİLEN YÖNTEMLERİN TESTİ.....</b>	<b>89</b>
<b>8. SONUÇ VE ÖNERİLER.....</b>	<b>101</b>
<b>KAYNAKLAR DİZİNİ .....</b>	<b>103</b>

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Yönsüz ( <i>G</i> ), Yönlü ( <i>D</i> ) ve Karma serim örneği ( <i>M</i> ) (Martínez, 2003) .....	5
2.2. a) Düzgün on iki yüzlü, b) Düzgün on iki yüzlü serimine karşı gelen bir Hamilton turu .....	7
2.3. a) Köningsberg köprüleri, b) Köningsberg köprülerinin serim gösterimi .....	9
3.1. a) <i>G</i> serimi, b) <i>G</i> serime karşı gelen kolaylaştırılmış serim ( $G_S$ ) (Hertz and Mittaz, 2000). .....	22
4.1. a) Kural dışı alt turlar, b) Uygun alt turlar .....	32
5.1. Gezgin robot kontrol sisteminin genel gösterimi (Özkan, 2007). .....	50
5.2. a) Sınırları olan bir ortam, b) Ortamın belirli bir kısmı kapsandıktan sonraki görüntüsü (Choset, 2000). .....	52
5.3. BSA ile kapsama örneği (Gonzalez, et.al., 2005) .....	52
5.4. Örnek test ortamı için elde edilen yol planı (Ryerson and Zhang, 2007). .....	53
5.5. a) Örnek serim, b) Örnek serimin ayrıtlarının numaralarının gösterimi .....	66
6.1. Pioneer 3-DX gezgin robot platformu .....	77
6.2. Mimarinin blok diyagramı .....	79
6.3. Test ortamının fotoğrafı .....	80
6.4. Ortamın GVD tabanlı serimi .....	81
6.5. a) İki robotla yapılan uygulamada servis verilen ayrıtların robotlar arasında bölümlendirilmesi, b) Uygulama sırasında robotların takip ettiği yollar .....	83
6.6. a) Üç robotla MobileSim simülasyon ortamında yapılan uygulamada servis verilen ayrıtların robotlar arasında bölümlendirilmesi, b) Uygulama sırasında robotların takip ettiği yollar .....	84
6.7. Üç robotlu algılayıcı tabanlı kapsama uygulamasında MobileSim simülasyon ortamından alınan bir ekran görüntüsü .....	85
6.8. Laboratuar katına ait yerleşim haritası .....	86
6.9. Laboratuar katının GVD tabanlı serimi. ....	86

**ŞEKİLLER DİZİNİ (devam)****Sekil****Sayfa**

- 6.10. a) Ortalama tur uzunluğu, b) Ortalama harcanan enerji, c) En uzun tur uzunluğu, d) En fazla harcanan enerji..... 88

## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
5.1. Kapasiteli ayrıt rotalama problemi ve çok robotlu ortam kapsanması probleminin benzer ve farklı yönleri.....	56
5.2. Örnek serimin ayrıt numaraları, bu ayrıta karşı gelen bitiş düğümleri ve ayrıt maliyetleri .....	67
5.3. Örnek serimin ayrıt numaraları, bu ayrıta karşı gelen servis ve geçiş talepleri ..	72
5.4. Örnek kromozomun uyum değerinin hesaplanması adımları .....	73
5.5. Permütasyon tipi iki noktalı çaprazlama örneği.....	75
5.6. İkili yer değiştirme mutasyonu örneği .....	75
6.1. Laboratuvar katında gerçekleştirilen testler sonucu elde edilen sonuçlar.....	87
7.1. Önerilen genetik algoritma ile elde edilen çözümlerin amaç fonksiyonu ve çözüm süresi açısından ortalama, en küçük ve en büyük değerleri .....	92
7.2. Test problemlerinin GAMS/CPLEX kullanılarak 1 saat durdurma kriteri ile elde edilen sonuçlar.....	93
7.3. GAMS/CPLEX ile elde edilen sonuçlar ve genetik algoritma çözümleri ile karşılaştırması .....	95
7.4. DUA'nın test problemleri çözülmesi ile elde edilen test sonuçları .....	97
7.5. Kapasite artırımı sonucu önerilen genetik algoritma ile elde edilen çözümlerin amaç fonksiyonu ve çözüm süresi açısından ortalama, en küçük ve en büyük değerleri.....	98
7.6. Kapasitesi arttırılmış test problemlerinin DUA ile çözülmesiyle elde edilen sonuçlar .....	99
7.7. “d-eglese” test problemlerinin DUA kullanılarak çözülmesi sonucu elde edilen sonuçlar .....	100

## 1. GİRİŞ

İster kamu ister özel kuruluş olsun hemen hemen her organizasyonun taşıma ile ilgili işleri vardır. Örneğin ürün üretimi gerçekleştiren bir şirketin hammadde tedarikinden son ürünün kullanıcıya ulaştırılmasına kadar geçen süreçte birçok taşıma faaliyeti vardır. Benzer şekilde, bir şehrin belediyesi için şehir çöplerinin toplanması veya caddelerin temizlenmesi kamu sektörü için verilebilecek taşıma örnekleridir. Bu tür problemlerde araçların görevlerini etkin bir şekilde yerine getirmek için hangi rotayı izlemeleri gerektiğinin belirlenmesi gerekir. Çeşitli amaçlarla görevli araçların veya kişilerin izlemeleri gereken iş sırasını tanımlayan problemlere genel olarak rotalama problemi denilmektedir. Şirketlerin daha iyi rekabet edebilmesi, kamu sektörünün verimli olabilmesi için söz konusu taşıma işinin yani taşıma yönetiminin iyi yapılması gerekir. Bu nedenle son elli yılda yöneylem araştırması alanında yapılan çalışmalarda rotalama problemlerine olan ilginin sürekli arttığı görülmektedir.

Rotalama problemleri düğüm gezme ve ayırıt gezme problemleri olarak iki sınıfa ayrılmaktadır. Düğüm gezme probleminde servis verilecek müşteriler belirli noktalarda örneğin şehirler veya cadde kesişimlerinde bulunmaktadır. Ayırıt gezme probleminde ise servis verilecek müşteriler bir yol boyunca konumlanmıştır. Bir caddenin üzerindeki çöplerin toplanması veya elektrik tellerinin kontrol edilmesi buna örnek olarak verilebilir. Hem düğüm gezme problemini hem de ayırıt gezme problemini içeren probleme ise genel rotalama problemi denmektedir. Bu çalışmada ayırıt rotalama problemi sınıfında yer alan kapasiteli ayırıt rotalama probleminin bir türü incelenmiştir. Kapasiteli ayırıt rotalama problemi (KARP), kapasite kısıtlı araçlarla belirli bir noktadan başlayıp yine aynı noktada bitecek ve bütün gerekli ayırıtlara servis verilmesini sağlayacak yolun belirlenmesi problemidir. KARP'ın uygulama alanları arasında çöp toplama, kışın yolların tuzlanması, sokakların süpürülmesi verilebilir.

Günümüzün popüler çalışmalarından birisi de otonom gezgin robotlar (autonomous robots) ve bu robotlara görev atanmasıdır. Otonom gezgin robotlar (GR), yapısal olmayan ortamlarda sürekli bir insan güdümü olmaksızın istenen görevleri gerçekleştiren, fiziksel bir konuma sabitlenmemiş ve bulunduğu çevrede serbest olarak

dolaşabilen robotlardır. Günümüzde gezgin robotlar insan sağlığı için tehlikeli olan uygulamalardan (mayın tespit ve imhası, nükleer güç santrallerinde atıkların toplanması vb.) askeri uygulamalara (keşif görevi, mühimmat tedariki vb.) kadar birçok alanda kullanılmaktadır. Gezgin robotların kullanım alanlarından birisi de çok robotla ortam kapsanması görevidir.

Çok robotlu ortam kapsanması problemi, bir ortamın robot takımı ile her noktasının en az bir robot tarafından taranmasını sağlayacak yolların belirlenmesi problemidir. Ortam kapsanması görevi çok robot yerine tek robotla gerçekleştirebilir. Fakat çok robotun kullanılması hata toleransını, güvenliği ve gürbüzlüğü arttırmaktadır. Ortam kapsanması uygulamalarına örnek olarak süpürme, çim biçme, tarama-kurtarma vb. verilebilir. Bu çalışmada çok robotlu ortam kapsanması probleminin, kapasiteli ayırıt rotalama problemine dönüştürülebileceği gösterilerek robot turlarının belirlenmesine dönük yöntemler geliştirilmiştir. Buna göre, çok robotlu ortam kapsanması problemindeki robotlar, kapasiteli ayırıt rotalama problemindeki araçlara; taranması gereken yerler, gezilmesi gereken ayrıtlara; robotun harcadığı enerji, ayırıt taleplerine; robotun enerji kapasitesi, araç kapasitesine karşı gelmektedir. Fakat sözü edilen iki problem arasında kritik bir fark vardır. Çok robotlu ortam kapsanması probleminde, robotun servis vermeden geçtiği yerlerde harcadığı enerji göz ardı edilemez. Yani bir robot bir yoldan görev yapmadan sadece geçebilir. Fakat bu durumda da enerji harcamaya devam eder ve herhangi bir ayrıta servis vermediği halde kalan enerjisinden bir miktarını harcamış olur. Öte yandan görevini yaparak geçeceği bir yolda harcayacağı enerji ise daha farklı olacaktır ve bu durumun KARP'ta bir karşılığı bulunmamaktadır. Bu çalışmada, servis verilmeden geçilen ayrıtlarda kullanılan kapasitenin de göz önüne alındığı bir KARP türü önerilmiştir. Bu tür bir KARP, literatürde ilk kez tanımlandığı için “Yeni KARP” olarak adlandırılmıştır.

Bu çalışmada, “Yeni KARP” için matematiksel bir model geliştirilmiş ve problemin etkin çözümü için en çok bilinen KARP sezgisellerinden birisi olan Ulusoy'un sezgiseli geliştirilerek probleme uyarlanmıştır. Bu amaçla öncelikle kırsal postacı ve kapasiteli ayırıt rotalama problemlerini çözmek için gerekli olan ve literatürde daha önce tanımlanmış 10 farklı algoritma C++ ile kodlanmıştır. Daha sonra önerilen

yeni sezgisel yaklaşımın bu algoritmalarla uyumlu çalışmasını sağlayacak bir yapı tasarlanarak Yeni KARP için çözümlerin hızlı bir şekilde türetileceği gösterilmiştir. Önerilen yöntemin gerçek ortamdaki etkinliği Pioneer 3-DX robotlarla Eskişehir Osmangazi Üniversitesi Yapay Zeka ve Robotik laboratuvarında kurulan test ortamında, büyük ortamlardaki etkinliği için ise simülasyon ortamında Eskişehir Osmangazi Üniversitesi Elektronik Mühendisliği laboratuvarı 1. katında yapılan testlerle gösterilmiştir. Ayrıca farklı başka çözüm yaklaşımları da olabileceğini göstermek için bir de genetik algoritma önerilmiştir.

Çalışmanın 2. bölümünde rotalama problemleri detaylandırılmıştır. 3. bölümde Çinli Postacı ve Kırsal Postacı problemleri anlatılmıştır. 4. bölümde kapasiteli ayırıt rotalama problemi, türleri ve çözüm yöntemlerine değinilmiştir. 5. bölümde önerilen yeni kapasiteli ayırıt rotalama probleminin formülasyonu ve çözüm yaklaşımları verilmiş, 6. bölümde ise gezgin robotlarda yapılan uygulamalardan bahsedilmiştir. 7. bölümde önerilen yöntemler için elde edilen sonuçlara verilmiştir. Son olarak 8. bölümde sonuçlar ve değerlendirmeler anlatılmıştır.



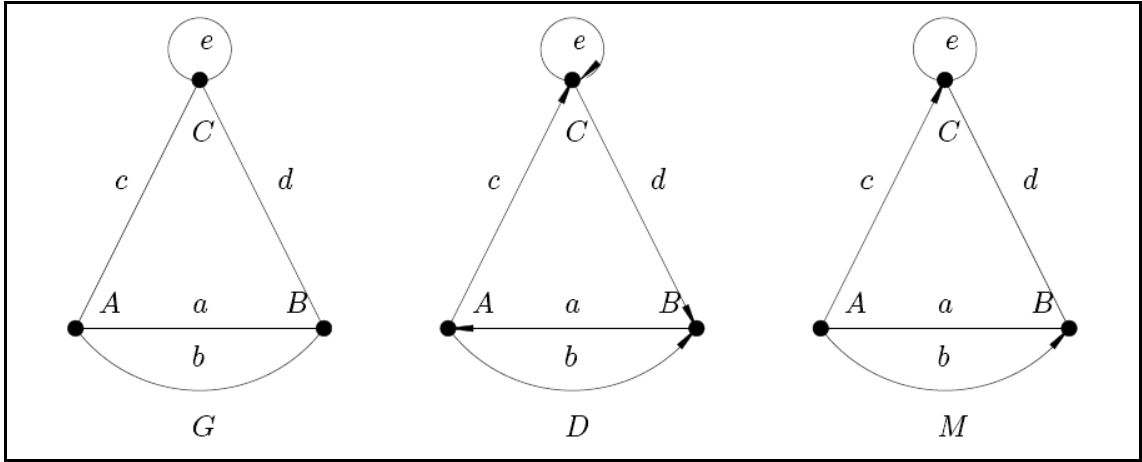
## 2. ROTALAMA PROBLEMLERİ

Rotalama problemi, bir veya daha fazla aracın belirli kısıtlar altında (maksimum yol uzunluğu, araç kapasitesi, zaman kısıtı gibi), tanımlanan bir görevi (Fabrikadan dağıtım birimlerine son ürünün gönderilmesi, bir şehirdeki çöplerin toplanması gibi) gerçekleştirirken, belirli bir amaç fonksiyonunu (kat edilen toplam yolun en küçüklenmesi, yüklerin dengeli dağıtılması gibi) eniyileyecek araç turlarının belirlenmesi problemidir. Rotalama problemleri, düğüm rotalama problemleri ve ayrıt rotalama problemleri olmak üzere ikiye ayrılır.

Rotalama problemleri serim açısından tanımlandığından izleyen alt bölümde öncelikle serim kuramındaki bazı gösterim ve tanımlamalara yer verilmiştir. Daha sonra sırasıyla düğüm rotalama problemleri ve ayrıt rotalama problemleri detaylandırılmıştır.

### 2.1. Gösterim ve Tanımlamalar

Bir serim (network, graph)  $G=(V,E\cup A)$  ile gösterilir.  $G$  seriminde  $V$  **düğüm**ler (vertices, nodes) kümesini,  $E$  yönsüz **ayrılar** (edges) kümesini ve  $A$  yönlü ayrıtlar (arcs) kümesini göstermektedir. Örneğin bir şehir yol haritasında düğümler, cadde kesişimi gibi yerleri, ayrıtlar ise bu yerleri birbirine bağlayan cadde veya yol bağlantılarını gösterir.  $v_i$  ve  $v_j$  düğümlerini birbirine bağlayan ayrıt  $(i, j)$  şeklinde gösterilir. Eğer bir ayrıt yönsüz ise bu iki düğümden istenen yönde  $v_i$ 'den  $v_j$ 'ye veya  $v_j$ 'den  $v_i$ 'ye geçilebilir. Yönlü ise ayrıt için bir yön tanımı vardır ve bu ayrıt üzerinden sadece bu yönde geçilebilir. Bir serimdeki bütün ayrıtlar yönsüz ise **yönsüz serim** (undirected graph), yönlü ise **yönlü serim** (directed graph) olarak adlandırılır. Serimdeki bazı ayrıtlar yönsüz bazı ayrıtlar yönlü ise bu tip serim **karma serim** (mixed graph) olarak adlandırılır (Şekil 2.1) (Gross and Yellen, 2004).



**Şekil 2.1.** Yönsüz ( $G$ ), Yönlü ( $D$ ) ve Karma serim örneği ( $M$ ) (Martínez, 2003)

Bir  $G$  serimindeki arka arkaya gelen düğüm ve ayrıtlar dizisi **yürüyüş** (walk) olarak adlandırılır. Örnek olarak  $v_0, e_0, v_1, e_1, v_2, \dots, e_{n-1}, v_n$  yürüyüşünde  $v_i \in V$ ,  $i = \{0, 1, \dots, n\}$  ve  $e_i \in (E \cup A)$ ,  $i = \{0, 1, \dots, (n-1)\}$ 'dir. Başladığı düğüme geri dönen yürüyüşe **kapalı yürüyüş** (closed walk) denir. Yürüyüş içerisindeki herhangi bir düğümün tekrarlanmadığı yürüyüş türüne **yol** (path) denilir. Başladığı düğümden biten yol **çevrim** (cycle) olarak adlandırılır. Herhangi bir iç ayrıtların tekrarlanmadığı kapalı yürüyüşe **tur** (tour, trail) denilir (Gross and Yellen, 2004).

Serimdeki bütün düğüm çiftleri arasında bağlantı varsa bu serime **tam bağlı serim** (complete graph), yoksa **seyrek serim** (sparse graph) denir. Bununla birlikte serimdeki ayrıtlar için mesafe veya maliyet gibi herhangi bir değer tanımlanmışsa, **ağırlıklandırılmış serim** (weighted graph), tanımlanmamışsa **ağırlıklandırılmamış serim** (unweighted graph) olarak adlandırılır (Ahuja, et. al., 1993). Ağırlıklandırılmış bir serim için bir  $C$  uzaklık matrisi tanımlanır.  $\forall (i, j) \in (E \cup A)$  olmak üzere  $(i, j)$  ayrıtlarından geçmenin maliyeti  $c_{ij}$ 'dir. Eğer  $v_i$  ve  $v_j$  düğümlerini bağlayan bir bağlantı yoksa bu durumda  $c_{ij} = \infty$  olarak atanır. Eğer serim yönsüz ise  $C$  matrisi simetriktir ( $c_{ij} = c_{ji} \forall i, j \in V$ ) diğer serim türleri için  $C$ , asimetrik bir matristir. Ayrıca serimdeki bağlantılar (yönlü ve yönsüz ayrıtlar) için bir talep değeri (Örneğin çöp toplama

problemi için bir cadde üzerindeki toplanacak çöp miktarı) tanımlanabilir. Serimdeki  $(i, j) \in (E \cup A)$  ayrıtının talep değeri  $q_{ij}$  ile gösterilir. Benzer şekilde serimdeki düğümler için de talep tanımlanabilir ve  $v_i \in V$  düğümün talebi  $q_i$  şeklinde gösterilir. Bununla birlikte serimlerde araçların yerleştirildiği özel bir düğüm vardır ve bu düğüme **depo düğümü** (depot vertex) denir. Serimlerde genellikle *1. düğüm* depo düğümü olarak seçilir.

Bir serimde bütün düğüm çiftleri arasında bir yol varsa bu serime **bağlı serim** (connected graph), diğer durumlardaki serimlere **kopuk serim** (disconnected graph) denir. Kopuk serimde birden fazla bağlı bileşen (connected components) bulunur (Ahuja, et. al., 1993).  $v_i$  düğümünden  $v_j$  düğümüne olan en küçük maliyetli yol, herhangi bir en kısa yol bulma algoritmasıyla belirlenebilir ve  $SP_{ij}$  şeklinde gösterilir.

Bir serimde bütün ayrıtıların yerine sadece bu kümenin bir alt kümesine servis verilmesi isteniyor olabilir. Servis verilmesi gereken ayrıtılar kümesi gezilmesi gereken ayrıtılar olarak adlandırılır ve  $R$  ( $R \subseteq E$ ) ile gösterilir. Kapasite kısıtlı problemler için gezilmesi gereken ayrıtılar ( $R$ ) talep değeri pozitif olan ayrıtılardır ( $R = \{(i, j) \mid q_{ij} > 0\}$ ,  $R \subseteq E$ ).

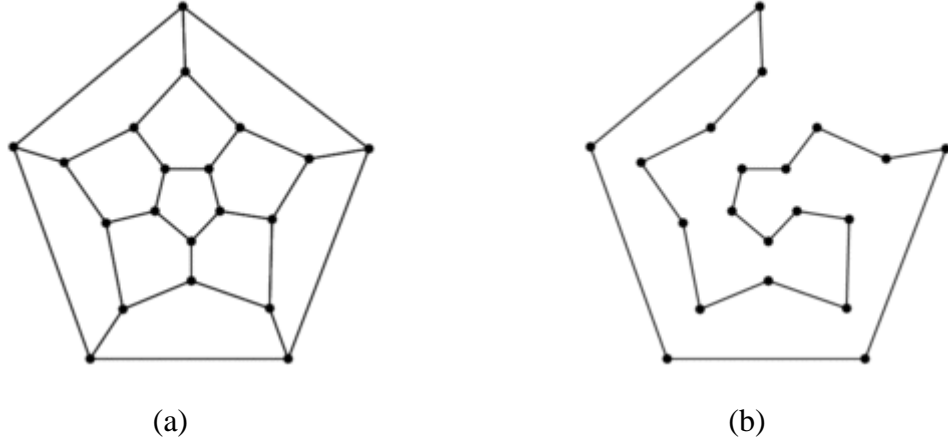
Son olarak, bir düğüme ( $i \in V$ ) komşu olan ayrıtı sayısı o düğümün derecesi olarak tanımlanır. Düğümün derecesi tek ise o düğüme **tek düğüm** (odd vertex), çift ise **çift düğüm** (even vertex) denir.

## 2.2. Düğüm Rotalama Problemleri

Düğüm rotalama probleminde servis verilecek yerler şehir veya caddelerin kesişimi gibi noktalarda yer alır. Amaç, araç turlarının belirlenmesidir. Düğüm rotalama problemleri ilk kez 1857 yılında William Rowan Hamilton tarafından bulunan “Icosian Game” isimli matematiksel oyun ile başlamıştır. Icosian oyunu, düzgün on iki yüzlünün

(dodecahedron) herhangi bir noktasından başlanarak bütün noktalara bir kez uğrayacak ve hiçbir düğüm tekrarı yapılmayacak çevrimin bulunmasıdır. Bu oyundan hareketle Hamiltonian tur problemi tanımlanmıştır.

Hamilton turu problemi (Hamiltonian tour problem): Serimdeki bir düğümden başlayarak bütün düğümlere bir kez uğrayacak ve başlanılan düğüme geri döndürecek turun bulunması problemidir. Düzgün on iki yüzlünün serim gösterimi (a) ve bu serime karşı gelen Hamilton turlarından bir tanesi (b) Şekil 2.2’de verilmiştir.



**Şekil 2.2.** a) Düzgün on iki yüzlü, b) Düzgün on iki yüzlü serimine karşı gelen bir Hamilton turu

Gezgin Satıcı Problemi (GSP) (Travelling Salesman Problem), Hamilton turunu temel alan ve en bilinen düğüm rotalama problemidir. GSP, düğüm sayısı  $n$  olmak üzere  $K_n = (V, E)$  yönsüz tam bağlı serimi üzerindeki en küçük maliyetli Hamilton turunun belirlenmesi problemidir. Problemin simetrik veya asimetrik olmasına göre tanımlanmış farklı türleri vardır. Gutin and Punnen, 2004 çalışmasında GSP ve türevlerini ayrıntılı olarak anlatılmıştır.

GSP'den türeyen diğer problemler ise m-Gezgin Satıcı Problemi (m-GSP) (m-Travelling Salesman Problem) ve Kapasiteli Araç Rotalama Problemidir (Kapasiteli-ARP) (Capacitated Vehicle Routing Problem).

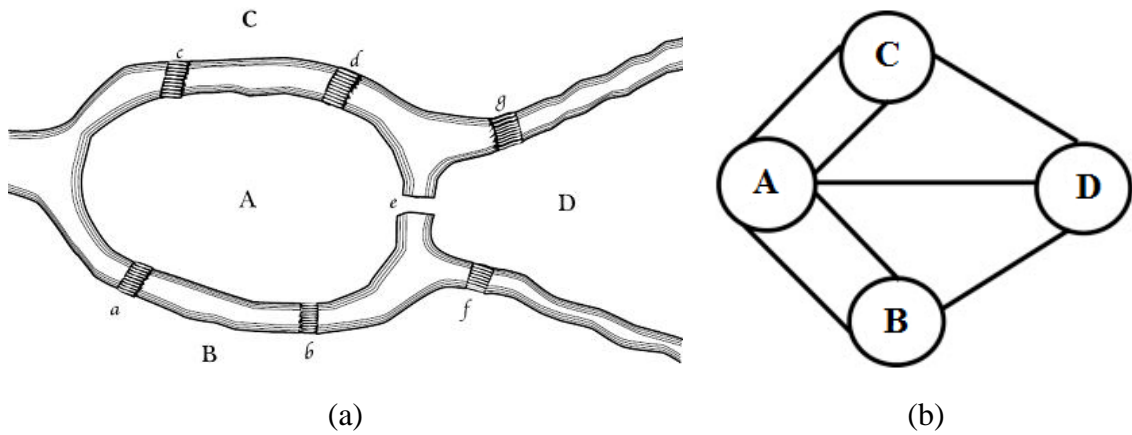
m-GSP,  $m$  tane kapasite kısıtı olmayan gezginin bulunduğu durumda gezginler için rotaların belirlenmesi problemdir. Kapasiteli Araç Rotalama Problemi ise  $m$  tane kapasite kısıtlı gezginin olduğu durum için rotaların belirlenmesi problemdir.

Buna göre yönsüz tam bağlı  $K_n = (V, E)$  serimi üzerinde  $v_1 \in V$  depo düğümünde bulunan  $m$  adet araç bütün düğümlere sadece bir kez uğrayarak turuna başladığı depo düğümüne dönecektir. Ayrıca herhangi bir araç uğradığı  $v_i$  düğümünün bütün talebini ( $q_i$ ) karşılamaktadır. Kapasiteli-ARP, araca yüklenen toplam talebin araç kapasitesini aşmadığı ve toplamda en küçük maliyetli olacak turların belirlenmesi problemdir. Kapasiteli-ARP literatürü (Toth and Vigo, 2002) çalışmasında detaylı olarak incelenmiştir.

### 2.3. Ayrıt Rotalama Problemleri

Ayrıt rotalama problemi, belirli kısıtlar altında bir serimdeki yönlü ve yönsüz ayrıtlara servis verecek en küçük maliyetli turun belirlenmesi problemdir. Ayrıt rotalama problemi, tarihi 1735'de Leonard Euler tarafından tanımlanan Köningsberg köprüleri problemiyle başlar (Euler, 1741; Gribkovskaia, et.al., 2007).

Bu problemde Köningsberg (şimdiki adıyla Kaliningrad) şehrinde bulunan yedi köprünün hepsinden sadece bir kez geçerek, başlanılan noktaya dönülecek bir yolun olup olmadığı araştırılmaktadır. Euler, Köningsberg problemini bir serim biçiminde tanımlamış ve literatüre Euler Turu kavramını katmıştır. İzleyen yıllarda da farklı problemler tanımlanmıştır. Köningsberg şehrinde bulunan yedi köprü (a) ve köprülere karşı gelen serim gösterimi (b) Şekil 2.3'de verilmiştir.



**Şekil 2.3. a) Köningsberg köprüleri, b) Köningsberg köprülerinin serim gösterimi**

### Euler Turu (Euler Tour) Problemi

Bağlı  $G = (V, E)$  serimindeki bütün ayrıtlardan sadece bir kez geçecek bir turun bulunması problemidir. Euler, bir  $G = (V, E)$  serimindeki bütün düğümlerin derecesi çift ise bu serimde Euler turunun bulunabileceğini ispatlamıştır. Daha sonraki yıllarda Euler turunun bulunması için sırasıyla Hierholzer 1873 yılında ve Fleury 1883 yılında birer algoritma önermiştir.

### Çinli Postacı Problemi (ÇPP) (Chinese Postman Problem-CPP)

Euler Turu probleminden sonra tanımlanan ayrıt rotalama problemi Çinli postacı problemidir. Verilen bir  $G = (V, E \cup A)$  serimindeki bütün yönlü ve yönsüz ayrıtlardan en az bir kez geçecek en küçük maliyetli turun bulunması problemidir. Bu problem Çinli bir matematikçi olan Mei-Ko Kwan tarafından 1962 yılında önerilmiştir. ÇPP'de en küçük maliyetli turun bulunacağı serim eğer yönsüz veya yönlü bir serimse ÇPP polinom zamanda çözülebilir (Edmonds & Johnson, 1973; Eiselt, 2000). Fakat serim karma hale dönüştüğünde problem  $NP$ -zor sınıfı bir probleme dönüşür (Papadimitriou, 1976). Literatürde ÇPP'nin birçok farklı türüyle ilgili yapılmış çalışmalar bulunmaktadır. Örnek olarak rüzgarlı Çinli postacı problemi (Windy CPP), hiyerarşik Çinli postacı problemi (Hierarchical CPP) verilebilir (Eiselt, et.al., 1995a).

### Kırsal Postacı Problemi (KPP) (Rural Postman Problem)

KPP, Çinli postacı probleminden farklı olarak serimdeki ayrıtların bir alt kümesinin gezilmesi problemidir. KPP, verilen bir  $G = (V, E \cup A)$  seriminde tanımlanmış olan gezilmesi gereken ayrıtlar kümesindeki ( $R \subseteq E \cup A$ ) bütün ayrıtları kapsayan en küçük maliyetli turun bulunması problemidir. KPP, Orloff tarafından 1974 yılında önerilmiştir. Bu probleminin  $NP$ -zor bir problem olduğu Lenstra ve Rinnooy Kan tarafından 1976 yılında ispat edilmiştir. Kırsal postacı probleminin uygulama alanlarına ve çözüm yöntemlerine (Eiselt, et.al., 1995b) çalışmasında detaylı olarak yer verilmiştir.

### Kapasiteli Ayrıt Rotalama Problemi (KARP) (Capacitated Arc Routing Problem)

Kapasite kısıtlı ve çok araçlı KPP olarak adlandırılabilen KARP şu şekilde tanımlanmaktadır. Bağlı bir  $G = (V, E \cup A)$  serimindeki gezilmesi gereken bütün yönlü ve yönsüz ayrıtlara ( $R \subseteq E \cup A$ ) depo düğümünde bulunan kapasite kısıtlı araçlarla ( $Q$ ) araç kapasiteleri aşılmadan toplamda en küçük maliyetli olacak şekilde servis verecek araç turlarının bulunması problemidir. KARP ilk kez Golden ve Wong tarafından 1981 yılında tanımlanmıştır. Yine Golden ve Wong tarafından en iyi çözüme  $3/2$  yakınlıkta bir çözümün bulunmasının bile  $NP$ -zor olduğu gösterilmiştir (Golden ve Wong, 1981).

## **2.4. Genel Rotalama Problemi**

Hem düğüm gezme, hem de ayrıt gezme problemini içeren rotalama problemine genel rotalama problemi (GRP) (General Routing Problem) denmektedir. KPP'nin uzantısı olarak belirtilen GRP şu şekilde tanımlanmaktadır. Verilen bir  $G = (V, E \cup A)$  seriminde tanımlanmış olan servis verilmesi gereken bütün ayrıtlara ( $R \subseteq E \cup A$ ) ve düğümlere ( $V_R \subseteq V$ ) en az bir kez uğrayacak en küçük maliyetli turun bulunması problemidir. GRP, Orloff tarafından 1974 yılında tanıtılmıştır. GRP,  $V_R = \{\}$  olduğunda KPP'ye indirgenir. Bu yüzden  $NP$ -zor sınıfı bir problemdir. GRP için çok yüzlü (polyhedral) sonuçları (Letchford, 1997), (Corberan and Sanchis, 1998) ve (Letchford, 1999) çalışmalarında verilmiştir.

İzleyen bölümde yönsüz postacı problemleri olan ÇPP ve KPP anlatılmış, bir sonraki bölümde ise KARP ayrıntılandırılmıştır.



### 3. YÖNSÜZ TEK ARAÇLI POSTACI PROBLEMLERİ

Yönsüz tek araçlı postacı problemleri yönsüz Çinli postacı problemi ve yönsüz kırsal postacı problemi şeklindedir. Bu problemler aşağıda anlatılmıştır.

#### 3.1. Yönsüz Çinli Postacı Problemi

Yönsüz ve bağlı  $G = (V, E)$  serimindeki her ayrıntın  $(i, j) \in E$  pozitif maliyet değeri  $(c_{ij})$  olsun (Serim yönsüz olduğu için  $c_{ij} = c_{ji}, \forall (i, j) \in E$ ). Yönsüz Çinli postacı problemi serimdeki her  $(i, j)$  ayrıntından en az bir kez geçecek en küçük maliyetli turun bulunması problemi olarak tanımlanır (Eiselt, et.al., 1995a, Edmonds and Johnson, 1973).

ÇPP sadece yönsüz veya yönlü serimler için polinom zamanlı bir problemdir. Bunun dışında tanımlanmış olan ÇPP türleri  $NP$ -zor sınıfındadır. Örneğin bazı ayrıntların yönlü bazı ayrıntların yönsüz olduğu karma serim için ÇPP  $NP$ -zor bir problemdir (Papadimitriou, 1976). Serimdeki ayrıntların  $((i, j) \in E)$  maliyeti  $(c_{ij}, c_{ji} \geq 0)$  ayrıntın geçiş yönüne bağlı ise bu tip serime rüzgârlı serim denir. Rüzgârlı serim üzerinde tanımlanmış Çinli postacı problemi  $NP$ -zor bir problemdir (Minieka, 1979). Ayrıntlar arasında öncelik ilişkilerinin tanımlandığı ÇPP türüne hiyerarşik ÇPP denir ve 1987 yılında Dror tarafından literatüre katılmıştır. Hiyerarşik ÇPP de  $NP$ -zor sınıfı problemlerdendir (Dror, et.al., 1987). (Cabral, et.al., 2004) çalışmasında hiyerarşik ÇPP'yi kırsal postacı problemine dönüştüren bir yöntem anlatılmıştır. Ayrıca Çinli postacı probleminin, amaç fonksiyonun fayda yönlü tanımlandığı en büyük faydalı Çinli postacı problemi (Pearn and Wang, 2003) ve ayrınt üzerindeki tıkanıklığa bağlı olarak düğümler arasındaki erişimin olasılık dağılımıyla gösterildiği tipteki ÇPP'de stokastik Çinli postacı problemi olarak tanımlanır (Tan, et.al., 2005).

### 3.1.1. Yönsüz Çinli postacı probleminin matematiksel modeli

Edmonds ve Johnson'un (1973) Çinli postacı problemi için önerdiği matematiksel modele göre, tek dereceli düğümler en küçük maliyetli olacak şekilde eşleştirilmeli ve serim Euler serim haline dönüştürülmelidir. Bu yönüyle problem en küçük maliyetli eşleştirme problemine benzemektedir. Edmonds ve Johnson'un önermiş olduğu modelde karar değişkeni ve parametreler aşağıdaki gibidir (Benavent, et.al., 2000).

**Model 3.1.** *Edmonds ve Johnson'un ÇPP'nin çözümü için önermiş olduğu model*

#### Kümeler

- $V$       Düğümler kümesi
- $E$       Ayrıtlar kümesi
- $S$       Düğüm kümesinin herhangi bir alt kümesi ( $S \subseteq V$ )
- $\delta(S)$    Bir  $S$  düğüm alt kümesi ( $S \subseteq V$ ) için başlangıç noktası  $S$  kümesinde, bitiş noktası  $V \setminus S$  kümesinde olan ayrıtlar kümesi

#### İndisler

- $e$       Serimdeki ayrıtları gösteren indis

#### Parametreler

- $c_e$        $e$  ayrıtlarının maliyeti ( $e \in E$ )

#### Karar Değişkenleri

- $x_e$        $e$  ayrıtlarının  $G$  serimine eklenecek sayısı ( $e \in E$ )

#### Model:

$$x(\delta(S)) \geq 1, \forall S \subset V, |\delta(S)| \text{ tek} \quad (3.1)$$

$$x_e \geq 0, \forall e \in E \quad (3.2)$$

$$x_e \in \mathbb{Z}^+, \forall e \in E \quad (3.3)$$

kısıtları altında,

$$enk\left(\sum_{e \in E} c_e \cdot x_e\right) \quad (3.4)$$

(3.1) numaralı kısıt grubu tek dereceli düğümleri bağlayan ayrıtların en az bir kopyası olmasını garantiler (Tek-kesme eşitsizliği). (3.2) ve (3.3) numaralı kısıtlar  $x_e$ 'nin 0'dan büyük ve tamsayı olmasını garantiler. (3.4) numaralı ifade, modelin amaç fonksiyonudur ve  $G$  serimine eklenecek bütün ayrıtların toplam maliyetini en küçükler.

Edmonds ve Johnson'un önermiş olduğu matematiksel model, en küçük maliyetli eşleştirmeyi yapar ve serimi Euler serim haline dönüştürür. Fakat araç turunun belirlenmesi için Euler serim üzerinde herhangi bir Euler turu bulma algoritması çalıştırılmalıdır. Bunun yerine Model 3.2'de verilen matematiksel model ile serimdeki bütün ayrıtlardan en az bir kere geçecek araç turu belirlenebilir. Model parametreleri ve karar değişkeni aşağıdaki gibidir.

**Model 3.2.** *ÇPP'de sonucu Euler turu olan model*

Kümeler

$V$       Düğümler kümesi,  $\{1, 2, \dots, n\}$

$E$       Ayrıtlar kümesi

İndisler

$i, j$     Düğüm indisleri

Parametreler

$n$       Serimdeki düğüm sayısı

$c_{ij}$      $(i, j)$  ayrıtlarının maliyeti ( $c_{ij} = c_{ji}, \forall (i, j) \in E$ )

Karar Değişkenleri

$x_{ij}$     aracın  $i$ 'den  $j$ 'ye geçiş sayısı

Model:

$$\sum_{\substack{j=1 \\ (i,j) \in E}} x_{ij} - \sum_{\substack{j=1 \\ (i,j) \in E}} x_{ji} = 0 \quad i = \{1, 2, \dots, n\} \quad (3.5)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i, j) \in E \quad (3.6)$$

$$x_{ij} \in Z^+ \quad \forall (i, j) \in E \quad (3.7)$$

kısıtları altında,

$$enk \left( \sum_{(i,j) \in E} c_{ij} \cdot x_{ij} \right) \quad (3.8)$$

(3.5) numaralı kısıt grubu akışın devamlılığını sağlar. (3.6) numaralı kısıt grubu serimdeki bütün ayrıtların  $(i, j)$  ve  $(j, i)$  yönünde en az bir kez uğranmasını garantiler. (3.7) numaralı kısıt karar değişkenlerinin negatif olmamasını ve tamsayı olmasını sağlar. (3.8) numaraları denklem gidilen toplam yolu en küçüklemektedir.

### 3.1.2. Yönsüz Çinli postacı problemi için sezgisel algoritma

Yönsüz bir serimin bütün düğüm dereceleri çiftse bu, bir Euler serimdir. Euler olmayan bir  $G$  seriminin Euler bir  $G'$  serimine dönüştürülmesi için serime yeni ayrıtların eklenmesi gerekir.  $G'$  serimindeki Euler turu  $G$  serimindeki Çinli postacı turuna karşılık gelir. Çinli postacı probleminin eniyi çözümünü bulmak için  $G$  serimine eklenecek ayrıtlar en küçük maliyetli olmalıdır. (Edmonds and Johnson, 1973) çalışmasında gösterildiği gibi ÇPP çözümünün eniyi olabilmesi için serimdeki tek dereceli düğümler arasındaki en küçük maliyetli eşleştirmeler belirlenmelidir. Yönsüz serimler için Çinli postacı problemini çözen algoritma aşağıdaki gibidir.

**Algoritma 3.1.** Çinli Postacı Problemi için çözüm algoritması

Girdi: Yönsüz ve bağlı  $G(V, E)$  serimi

Çıktı:  $G$  üzerinde en küçük maliyetli Çinli postacı turu  $T$

*Adım 1:* Eğer  $G$  serimindeki bütün düğümlerin derecesi çift ( $G$ , Euler bir serim) ise Adım 5'e git.

*Adım 2:*  $G$  serimindeki tek dereceli düğümleri ( $V_0$ ) belirle. Düğüm kümesi  $V_0$  olan  $G_0$  tam serimini oluştur.  $G_0$  serimindeki  $(v_i, v_j)$  düğümleri arasındaki uzaklık,  $G'$  'de  $v_i$  ve  $v_j$  düğümleri arasındaki en kısa yolun uzunluğu ile belirlenir. ( $v_i$  ve  $v_j$  düğümleri arasındaki en kısa yol  $SP_{ij}$ ).

*Adım 3:*  $G_0$  'da minimum maliyetli en küçük eşleştirmeyi bul.

*Adım 4:*  $G' = G$ .  $G'$  serimine, her  $(v_i, v_j)$  optimal eşleşmesi için  $SP_{ij}$  'nin üzerinde bulunan ayrıtları ekle.

*Adım 5:*  $G'$  serimi üzerindeki  $T$  Euler turunu Euler Çevrimi algoritması ile belirle.

Çinli postacı probleminin çözümü için Edmonds ve Johnson tarafından önerilen algoritmanın iki temel adımı vardır. Bunlar, serimin Euler formuna dönüştürülmesi için gerekli olan tek dereceli düğümler arasındaki en küçük maliyetli eşleştirme (Adım 3) ve Euler bir serim üzerindeki Euler turunun belirlenmesidir (Adım 5). İzleyen alt bölümlerde algoritmanın bu iki adımı detaylı olarak anlatılmıştır.

### **3.1.2.1. En küçük maliyetli eşleştirme**

Verilen bir  $G(V, E)$  serimindeki bütün ayrıtların bir maliyet ( $c_e, \forall e \in E$ ) değeri olsun. En küçük maliyetli eşleştirme problemi,  $G$  serimi üzerindeki en küçük maliyetli eşleştirmenin ( $M$ ) bulunması olarak tanımlanır  $\sum(c_e : e \in M)$ . En küçük maliyetli eşleştirme probleminin çözümü için Edmonds (1965) tarafından polinom zamanlı olan çiçek algoritması (Blossom algorithm) önerilmiştir. Problem için ilerleyen yıllarda daha düşük çalışma zamanlarında çözüm bulan algoritmalar önerilmiştir (Cook and Rohe, 1999). Yakın zamanda Edmonds'un önermiş olduğu Blossom algoritmasının yeni bir uygulaması olan, en kötü çalışma zamanı iyileştirilen ve Blossom V algoritması olarak adlandırılan bir algoritma önerilmiştir (Kolmogorov, 2009). Bu çalışmada en küçük maliyetli eşleştirme algoritmasının çözümünde Blossom V algoritması kullanılmıştır.

Ayrıca en küçük maliyetli eşleştirme probleminin tamsayı matematiksel modeli Model 3.3'teki gibidir. Probleme ait indisler, parametreler ve model aşağıda verilmiştir.

**Model 3.3.** *En küçük maliyetli eşleştirme problemi için tamsayı model*

Kümeler

$V$  Serimdeki düğüm numaraları  $\{1, 2, \dots, n\}$

İndisler

$i, j$  Düğüm indisleri

Parametreler

$n$  Serimdeki düğüm sayısı

$c_{ij}$   $(i, j)$  düğümleri arasındaki uzaklık

Karar Değişkenleri

$$x_{ij} = \begin{cases} 1 & \text{Eger } i \text{ düğümü ile } j \text{ düğümü eşlesiyor} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

Model:

$$\sum_{\substack{j=1 \\ j>i}}^n x_{ij} + \sum_{\substack{j=1 \\ j<i}}^n x_{ji} = 1 \quad \forall i \in V \quad (3.9)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \quad (3.10)$$

kısıtları altında,

$$enk \left( \sum_{i=1}^n \sum_{\substack{j=1 \\ i<j}}^n c_{ij} \cdot x_{ij} \right) \quad (3.11)$$

(3.9) numaralı kısıt grubu serimdeki bütün düğümler arasında birebir eşleşme olmasını garantiler. (3.10) numaralı kısıt grubu karar değişkenlerinin 0-1 tamsayı değişken olmasını garantiler. (3.11) numaralı denklem modelin amaç fonksiyonudur ve eşleştirmedeki ayrıtların toplam maliyetinin en küçük olmasını sağlar.

Çinli postacı probleminin çözümünde, en küçük maliyetli eşleştirmenin kullanılmasının amacı serimi Euler hale dönüştürmektir. Algoritma 3.1'in 2. adımında

verildiği gibi tek dereceli düğümlerden oluşan  $G_0$  tam serim oluşturulduktan sonra bu serim üzerinde en küçük maliyetli eşleştirme gerçekleştirilir. Fakat eşleşmenin tam olarak gerçekleşmesi için  $G_0$  'daki düğüm sayısının yani serimdeki tek dereceli düğüm sayısının çift olması gerekir. Buna ilişkin el sıkışma ön kuramı (handshaking lemma) aşağıda verilmiştir (Diestel, 2005).

**Teorem 3.1.** Bir serimdeki bütün düğümlerin dereceleri toplamı çifttir (Diestel, 2005).

**İspat:** Denklem (3.12)'de de gösterildiği gibi her düğümün iki uç noktası olduğundan serimdeki bütün düğümlerin dereceleri toplamı çifttir.

$\deg(v)$  :  $v$ -inci düğümün derecesi

$$2m = \sum_{v \in V} \deg(v) \quad (3.12)$$

**Teorem 3.2.** Bir serimdeki tek dereceli düğüm sayısı çifttir (Diestel, 2005).

**İspat:** (3.13)'te verilen denklemdeki eşitliğin sol tarafı bütün serimdeki derecelerin toplamını, sağ tarafı ise tek dereceli ve çift dereceli düğümlerin toplamını göstermektedir. Eşitliğin sol tarafı çifttir (Teorem 3.1). Eşitliğin sağ tarafındaki çift dereceli düğümlerin toplamı da çifttir. Bu durum tek dereceli düğümlerin toplamının da çift olduğunu gösterir. O halde tek dereceli düğüm sayısı çifttir.

$$\sum_{v \in V} \deg(v) = \sum_{\substack{v \in V \\ v \text{ çift}}} \deg(v) + \sum_{\substack{v \in V \\ v \text{ tek}}} \deg(v) \quad (3.13)$$

Serim Euler formuna dönüştürüldükten sonra bu serim üzerindeki Euler turu belirlenebilir. İzleyen alt bölümde Euler turu belirleme algoritmaları anlatılmıştır.

### 3.1.2.2. Euler turu

Yönsüz  $G = (V, E)$  serimindeki bir düğümden başlayarak bütün ayrıtlardan ( $E$ ) bir kez geçerek başlangıç noktasına döndüren yola Euler turu (Euler çevrimi) denir. Euler'in yaptığı ispata göre serimde Euler turunun olabilmesi için serimdeki bütün düğümlerin derecesi çift olmalıdır (Euler, 1741).

Euler bir serim üzerindeki Euler turunun bulunabilmesi için 1873 yılında Hierholzer bir algoritma önermiştir. Bu algoritma serimdeki herhangi bir düğümden başlayarak bir çevrim oluşturur. Daha sonra oluşturulan bu çevrimdeki herhangi bir düğümden başlayarak yeni bir çevrim oluşturur ve bir önceki çevrime ekler. Bu işleme bütün ayrıtlara bir kez uğranana kadar devam edilir. Hierholzer'ın Euler turu algoritmasının adımları aşağıda verilmiştir.

**Algoritma 3.2.** *Hierholzer'in Euler çevrim algoritması*

Girdi: Bütün düğüm dereceleri çift olan bağlı bir  $G$  serimi

Çıktı:  $G$  üzerindeki Euler turu ( $T$ )

*Adım 1:*  $G$ 'deki herhangi bir  $v$  düğümünden başla ve bir  $T$  turu oluştur.

*Adım 2:* Eğer  $G$  serimindeki bütün ayrıtlar kullanılmışsa *Adım 5*'e git. Diğer durumda *Adım 3*'e git.

*Adım 3:*  $T$  turunda kullanılmamış bir ayrıta komşu olan herhangi bir  $w$  düğümünden başla ve  $D$  turunu oluştur.

*Adım 4:*  $D$  turunu ortak  $w$  düğümünden  $T$  turuna bağla. *Adım 2*'ye git.

*Adım 5:*  $T$  turu elde edilmiştir.

Yönsüz serimlerde Euler turun bulunması için önerilmiş diğer algoritma Fleury'nin algoritmasıdır. Bu algoritma, serim üzerindeki herhangi bir düğümden başlar ve düğüme komşu olan ve de köprü (Ayrıtların serimden çıkartılması iki veya daha fazla ayrık bağlı bileşen oluşmasına neden oluyorsa) olmayan bir ayrıtı seçer. Eğer böyle bir ayrıtı yoksa algoritmayı sonlandırır, diğer durumlarda ayrıtı Euler turuna ekler ve



serimden çıkarır. İşleme ayrıtın diğer bitiş noktasından devam edilir. Fleury'nin algoritmasının detayları aşağıda verilmiştir.

**Algoritma 3.3.** *Fleury'nin Euler çevrim algoritması*

Girdi: Bütün düğüm dereceleri çift olan bağlı bir  $G = (V, E)$  serimi

Çıktı:  $G$  üzerindeki Euler turu ( $T$ )

*Adım 1:*  $G$ 'deki herhangi bir  $v \in V$  düğümünden başla.

*Adım 2:* Seçilen düğümüne komşu olan ve köprü olmayan bir  $e \in E$  ayrıtı seç.

*Adım 3:*  $e$  ayrıtını  $T$  turuna ekle ve seçilen düğüm olarak  $e$  ayrıtının diğer bitiş düğümünü ata.

*Adım 4:*  $e$  ayrıtını  $G$ 'den sil. Eğer kopuk bir düğüm oluşuyorsa bu düğümü de  $G$ 'den sil.

*Adım 5:*  $G$ 'deki bütün düğümler silindiyse *Adım 6*'ya, diğer durumlarda *Adım 2*'ye git.

*Adım 6:*  $T$  turu elde edilmiştir.

### 3.2. Yönsüz Kırsal Postacı Problemi

Yönsüz ve bağlı  $G = (V, E)$  serimindeki her ayrıtın  $(i, j) \in E$  pozitif maliyet değeri ( $c_{ij}$ ) ve gezilmesi gereken ayrıtları gösteren  $R \subseteq E$  tanımlansın. Yönsüz kırsal postacı problemi  $R$ 'deki bütün ayrıtlardan en az bir kez geçecek en küçük maliyetli turun bulunması problemidir (Assad and Golden, 1995).  $G$  seriminde gezilmesi gereken ayrıtların ( $R$ ) bitiş noktalarına komşu olan düğüm kümesi  $V_R$  şeklinde gösterilir. Gezilmesi gereken alt serim  $G_R = (V_R, R)$ ,  $G$ 'de yer alan gezilmesi gereken ayrıtlardan elde edilir. Genel olarak KPP  $NP$ -zor bir problemdir (Lenstra and Rinnooy Kan, 1976). Fakat  $G_R$  bağlı bir serim ise problem polinom zamanlı olarak çözülebilir.

Kırsal postacı probleminin zorluğu turun, serimdeki farklı bileşenlerinin nasıl birleştirilmesi gerektiğinin belirlenmesinde ortaya çıkar (Wøhlk, 2005). Literatürde

KPP'nin çözülmesi için farklı sezgiseller önerilmiştir. Örnek olarak (Pearn and Wu, 1995) çalışmasında sezgisel algoritma, (Córdoba, et.al., 1998) çalışmasında Monte Carlo yöntemi tabanlı bir sezgisel algoritma ve (Ghiani, et.al., 2006) çalışmasında yapıcı bir algoritma ve çözümün iyileştirilmesi için yöntemler önerilmiştir. (Ghiani, 2006) çalışmasında elde edilen sonuçlar ile Frederickson'ın sezgiseli ile elde edilen sonuçlar 350 düğümlü test problemleri kullanılarak karşılaştırılmıştır. (Hertz, et.al., 1999) çalışmasında yönsüz KPP ile elde edilen sonuçların iyileştirilmesi için farklı iyileştirme algoritmaları (Kısaltma, çıkarma, ekleme, 2-OPT, TWIN-OPT) önerilmiştir. En iyi çözümün bulunması için önerilen yöntemlere (Corberán ve Sanchis, 1994) ve (Ghiani and Laporte, 2000) çalışmalarına bakılabilir. KPP detaylı olarak (Hertz, 2005) ve (Eiselt, et.al., 1995b) çalışmalarında anlatılmıştır.

Yönsüz KPP'nin çözümü için (Christofides, et.al., 1981: Hertz and Mittaz, 2000) çalışmasında  $G_R$  serimini başka kolaylaştırılmış serim olarak adlandırılan bir serime ( $G_S$ ) dönüştüren bir ön işlem prosedürü önerilmiştir. Kolaylaştırılmış serimi oluşturma algoritmasının adımları aşağıda verilmiştir.

**Algoritma 3.4.** *Kolaylaştırılmış serimi oluşturma algoritması*

Girdi: Yönsüz  $G = (V, E)$  serimi ve  $G$ 'nin gezilmesi gereken ayrıtlarından indirgenmiş  $G_R = (V_R, R)$  serimi.

Çıktı: Kolaylaştırılmış serim  $G_S = (V_R, R \cup E_S)$

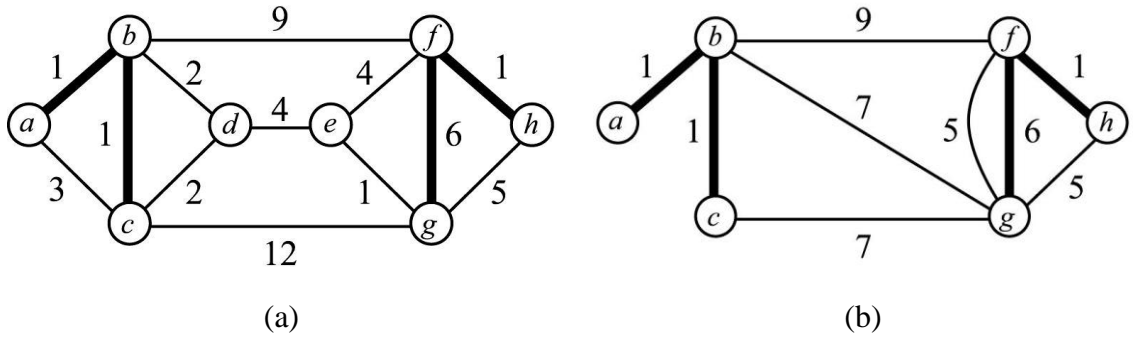
*Adım 1:*  $R$ 'de yer alan her ayrıt hiç bir değişikliğe uğratılmadan  $G_S$ 'e kopyala.

*Adım 2:*  $\forall v_i, v_j \in V_R$  olmak üzere  $(v_i, v_j)$  ayrıtı  $E_S$  kümesine ekle ve tam bağlı bir serim elde et.  $(v_i, v_j)$  ayrıtının uzunluğu  $G$ 'de  $v_i$  ve  $v_j$  düğümlerini bağlayan en kısa yolun uzunluğudur.

*Adım 3:*  $E_S$ 'e eklenen ayrıt aşağıdaki koşullardan herhangi birine uyuyorsa o ayrıtı serimden çıkart.

- i.  $\forall (v_i, v_j) \in E_S$  olmak üzere  $\exists v_k \in V_R$  için  $c_{ij} = c_{ik} + c_{kj}$  koşulu sağlanıyorsa
- ii. İki paralel ayrıt varsa ve bunların maliyet değerleri aynıysa.

Kırsal postacı problemi turu kolaylaştırılmış serim ( $G_s$ ) üzerinde belirlenir.  $G_s$  üzerinde elde edilen tur  $G$  üzerindeki tura en kısa yollardan bağlanarak elde edilir. Bundan sonraki kısımlarda yönsüz kırsal postacı probleminin çözümünde kullanılan yöntemlerde  $G$  seriminin  $G_s$  'e çevrildiği varsayılmaktadır. Şekil 3.1'de bir serim ve bu serime karşı gelen kolaylaştırılmış serim verilmiştir.



**Şekil 3.1.** a)  $G$  serimi, b)  $G$  serime karşı gelen kolaylaştırılmış serim ( $G_s$ ) (Hertz and Mittaz, 2000).

Bundan sonraki kısımlarda, kırsal postacı probleminin çözümü için kullanılan yöntemlerde serim ilk olarak kolaylaştırılmış serime dönüştürülmektedir. İzleyen alt bölümlerde yönsüz kırsal postacı problemi için matematiksel model ve sezgisel algoritma ile çözüm verilmiştir.

### 3.2.1. Yönsüz KPP'nin matematiksel modeli

Yönsüz kırsal postacı probleminin çözümü için Christofides, Campos ve Corberán (1981) tarafından önerilen matematiksel model aşağıda verilmiştir (Hertz, 2005).

**Model 3.4.** *Kırsal postacı probleminin matematiksel modeli*Kümeler

- $V$  Dğümler kümesi,  $\{1, 2, \dots, n\}$
- $E$  Ayrıtlar kümesi
- $S$   $V$  dğümler kümesinin herhangi bir alt kümesi
- $\delta(S)$  Bir  $S$  dğüm alt kümesi ( $S \subseteq V$ ) için başlangıç noktası  $S$  kümesinde, bitiş noktası  $V \setminus S$  kümesinde olan ayrıtlar kümesi
- $\delta(i)$   $S$  dğüm alt kümesi sadece  $i$  dğümünü içeriyor
- $P$   $G_R$  serimindeki bağılı bileşenler kümesi,  $\{1, 2, \dots, p\}$
- $V_k$   $G_R$  serimindeki bağılı bileşenlerin dğüm kümesi

İndisler

- $i$   $G$  serimdeki dğüm indisleri
- $k$   $G_R$  serimindeki bağılı bileşen indisleri
- $e$  Serimdeki ayrıtları gösteren indis ( $\forall e \in E$ )

Parametreler

- $n$  Serimdeki dğüm sayısı
- $p$  Bağılı bileşen sayısı
- $c_e$   $e$  ayrıtlının maliyeti ( $\forall e \in E$ )

Karar Değişkenleri

- $x_e$  Euler serim elde edilmesi için  $e \in E$  ayrıtlının kullanım sayısı
- $z_i$  Herhangi bir tamsayı

Model:

$$|R \cap \delta(i)| + \sum_{e \in \delta(v_i)} x_e = 2 \cdot z_i \quad i \in V \quad (3.14)$$

$$\sum_{e \in \delta(W)} x_e \geq 2 \quad W = \bigcup_{k \in P} V_k, P \subset \{1, \dots, p\}, P \neq \emptyset \quad (3.15)$$

$$x_e \geq 0 \text{ ve tamsayı} \quad e \in E \quad (3.16)$$

$$z_i \geq 0 \text{ ve tamsayı} \quad \forall i \in V \quad (3.17)$$

kısıtlar altında

$$\text{enk} \sum_{e \in E} c_e \cdot x_e \quad (3.18)$$

Modeldeki (3.14) numaralı kısıt grubu serimin Euler serim olmasını sağlar. İfadedeki eşitliğin sol tarafı  $v_i$  düğümüne komşu olan ayrıt sayısını ( $v_i$  düğümünün derecesini) gösterir. Sağ taraf ise çift bir tamsayıdır. (3.15) numaralı kısıt grubu çözümün bağlı olmasını garanti eder.  $G_R = (V, R)$  seriminde  $W = \bigcup_{k \in P} V_k$  olmak üzere  $W$   $k$ -ıncı bileşenin bütün düğümlerini gösterir. Bu kısıt  $W$  düğüm setinden  $V \setminus W$  düğüm setine en az iki bağlantı olmasını sağlar. Bu sayede çözümde bağlı bileşenleri birleştiren bir tur elde edilebilir. (3.16) ve (3.17) numaralı kısıt grupları  $x_e$  ve  $z_i$  karar değişkenlerinin tamsayı olmasını garantiler. (3.18) numaralı denklem modelin amaç fonksiyonudur ve kullanılacak ayrıtların toplam maliyetinin en küçük olmasını sağlar.

Problem genel olarak  $NP$ -zor bir problem olduğundan, matematiksel model ile büyük boyutlu problemleri çözmek mümkün değildir. Bu nedenle izleyen bölümde problem çözümü için önerilen bir sezgisel algoritma anlatılmıştır.

### 3.2.2. Yönsüz kırsal postacı problemi için sezgisel algoritma

Yönsüz kırsal postacı probleminin çözüm yöntemi girdi olan  $G_R$  seriminin bağlı olup olmamasına göre değişmektedir.  $G_R$  bağlı bir serim ise problemin çözümü için Çinli postacı problemi için verilen algoritmaya (Algoritma 3.1) benzer bir algoritma kullanılır ve en iyi çözümü garanti eder. Fakat  $G_R$  kopuk bir serim ise problem  $NP$ -zor bir problemdir. Bu nedenle Frederickson en iyi çözüme  $3/2$  yakınlıkta çözüm veren algoritma önermiştir (1979). Algoritmalara ait adımlar aşağıda verilmiştir.

**Algoritma 3.5.** *Bağlı Yönsüz KPP çözüm algoritması*

Girdi: Yönsüz  $G = (V, E)$  serimi ve  $R$ 'den elde edilmiş bağlı  $G_R = (V_R, R)$  serimi.

Çıktı:  $G$  üzerindeki en küçük maliyetli kırsal postacı turu ( $T$ ).

Adım 1:  $G' := G_R$ . Eğer  $G'$  seriminin bütün düğüm dereceleri çift ise *Adım 5*'e git.

Adım 2:  $G'$  serimindeki tek dereceli düğümleri ( $V_0$ ) belirle. Düğüm kümesi  $V_0$  olan  $G_0$  tam serimini oluştur.  $G_0$  serimindeki  $(v_i, v_j)$  düğümleri arasındaki uzaklık,  $G'$  'de  $v_i$  ve  $v_j$  düğümleri arasındaki en kısa yolun uzunluğu ile belirlenir.  $v_i$  ve  $v_j$  düğümleri arasındaki en kısa yol  $SP_{ij}$ .

Adım 3:  $G_0$  'da minimum maliyetli en küçük eşleştirmeyi bul.

Adım 4:  $G'$  serimine, her  $(v_i, v_j)$  optimal eşleşmesi için  $SP_{ij}$  'nin üzerinde bulunan ayrıtları ekle.

Adım 5: Euler tur algoritması ile  $G'$  serimi üzerindeki  $T$  Euler turunu belirle.  $T$   $G$  serimindeki eniyi KPP turudur.

Gezilmesi gereken ayrıtlardan indirgenerek belirlenen  $G_R$  serimi bağlı bir serim değilse problemin en iyi çözümü polinom zamanda belirlenemez. Bu yüzden Frederickson yönsüz kırsal postacı problemine Christofides'in gezgin satıcı problemi için önermiş olduğu algoritma aracılığıyla çözüm bulan bir algoritma önermiştir. Bu algoritma en kötü durumda en iyi çözümü  $3/2$  yakınlıkta çözüm bulmaktadır. Algoritmanın adımları aşağıda verilmiştir.

**Algoritma 3.6.** *Yönsüz KPP için çözüm algoritması (Frederickson'in sezgiseli)*

Girdi: Yönsüz  $G = (V, E)$  serimi ve  $R$  'den elde edilmiş  $G_R = (V_R, R)$  serimi.

Çıktı:  $G$  üzerindeki kırsal postacı turu ( $T$ ).

Adım 1:  $G' := G_R$ .  $C_1, \dots, C_c$   $G_R$  serimindeki bağlı bileşenler. Eğer  $c = 1$  ( $G_R$  'deki bileşen sayısı) ise *Adım 3*'e git. Düğüm kümesi  $\{1, \dots, c\}$  olan tam bağlı  $G''$  serimini oluştur.  $G''$  serimindeki  $(p, q)$  ayrıtlının uzunluğunu  $G$  seriminde  $C_p$  ve  $C_q$  arasındaki en kısa yolun uzunluğu ile belirle.

Adım 2:  $G''$  seriminde en küçük maliyetli kapsayan ağacı (minimum spanning tree) belirle. En iyi kapsar ağaçtaki her  $(p, q)$  ayrıtı için  $G$  seriminde  $C_p$  ve  $C_q$  düğümleri arasındaki en kısa yol üzerinde bulunan bütün ayrıtıları  $G'$  serimine ekle.

Adım 3: Algoritma 3.5'nin 2. adımından 5. adıma kadar olan kısmını uygula. Elde edilen  $T$  turu  $G$  serimi üzerinde kırsal postacı turudur.

Algoritma 3.6'da  $G_R$  serimindeki bağlı bileşenlerin ve en küçük maliyetli kapsar ağacın nasıl belirleneceği anlatılmamıştır. Aşağıda bu iki işleme ait algoritma detaylı olarak verilmiştir.

Bu çalışmada,  $G_R$  serimindeki bağlı bileşenleri belirlemek için Floyd-Warshall algoritması tabanlı bir yöntem kullanılmıştır. Floyd-Warshall algoritması düğümler arasındaki uzaklıkların tanımlı olduğu seyrek bağlı bir serimdeki bütün düğümler arasındaki en kısa yolların ve yol uzunluklarının belirlendiği bir algoritmadır. Algoritmanın karmaşıklığı  $O(|V|^3)$ 'dür.  $G_R$  serimindeki bağlı bileşenleri belirlemek için kullanılan algoritma adımları aşağıda verilmiştir.

**Algoritma 3.7.** Serimdeki bağlı bileşenlerin belirlenmesi

Girdi: Yönsüz  $G = (V, E)$  serimi ve  $R$ 'den elde edilmiş bağlı bileşenler içeren  $G_R = (V_R, R)$  serimi.

Çıktı:  $G_R$  üzerindeki bağlı bileşenler  $(C_1, \dots, C_c)$ .

Adım 1:  $G' := G_R$ . Floyd-Warshall algoritması ile  $G'$  serimindeki bütün düğümler arasındaki en kısa yolları belirle. Floyd-Warshall algoritması ile elde edilen uzaklık matrisinde bağlı düğümler için pozitif bağlı olmayan düğümler için "0" gözlenir.

Adım 2: Pozitif değer alan düğüm çiftlerini grupla. Her grup bağlı bir bileşendir.

Adım 3: Bağlı bileşenler elde edilmiştir.

Algoritma 3.6'nın 2. adımında bağlı bileşenlerden elde edilen tam bağlı  $G''$  seriminde en küçük maliyetli kapsar ağacın belirlenmesi gerekir. En küçük maliyetli

kapsar ağaç, üzerinden bir serimdeki tüm düğümlere ulaşılabilen ağaçlar içerisinde, toplam ağırlığı en düşük olan ağaç olarak tanımlanır. Bu tezde bir serim üzerindeki en küçük kapsar ağacın belirlenmesi için önerilen yöntemlerden Prim'in algoritması seçilmiştir.

**Algoritma 3.8.** *Prim'in en küçük maliyetli kapsar ağaç algoritması*

Girdi: Yönsüz bağlı  $G = (V, E)$  serimi ve serimdeki düğüm sayısı ( $N$ )

Çıktı:  $G$  üzerindeki en küçük maliyetli kapsar ağaç

Adım 1: Serim üzerindeki herhangi bir düğümü ağacı oluşturmaya başlamak için seç.

Adım 2: Oluşturulan ağaca eklemek için, şu ana kadar oluşturulmuş ağaç üzerinden erişilebilen ve daha önceden ağaca katılmamış olan en küçük ağırlıklı ayrıtı seç.

Adım 3: Eğer bu ayrıtın ağaca katılması, bir çevrim oluşmasına sebep olmuyorsa, ağaca ekle.

Adım 4: Ağaçtaki ayrıt sayısı  $(N-1)$ 'e ulaşana kadar Adım 2'ye dön.

Frederickson tarafından önerilen sezgisel algoritma en iyi çözümü garanti etmediğinden daha iyi çözümlerin elde edilmesi için (Hertz, et.al, 1999) ve (Hertz, et.al., 2000) çalışmalarında yönsüz kırsal postacı problemi için optimizasyon sonrası yöntemleri önerilmiştir. Bu çalışmada, önerilen algoritmalarından biri olan Shorten algoritması elde edilen yönsüz kırsal postacı turunu iyileştirmek için kullanılmıştır. Shorten algoritmasının adımları aşağıda verilmiştir.

**Algoritma 3.9.** *Shorten*

Girdi: Yönsüz bağlı  $G = (V, E)$  serimi, gezilmesi gereken ayrıtlar ( $R$ ) ve  $G$  serimindeki  $T$  kırsal potacı turu.

Çıktı: Olası daha iyi bir  $T$  kırsal potacı turu.

Adım 1:  $T$  üzerinde bir yön seç ve  $v_i$  tur üzerindeki herhangi bir düğüm olsun



Adım 2:  $T$  üzerinde  $v_i$  ve  $v_j$  düğümlerini bağlayan yol uzunluğunun mümkün olduğunca fazla olacağı ve  $v_j$ 'den  $v_i$ 'ye bütün gezilmesi gereken ayrıtların kapsanacağı bir  $v_j$  düğümü belirle.  $P$ ,  $v_i$ 'den  $v_j$ 'ye,  $Q$  ise  $v_j$ 'den  $v_i$ 'ye doğru giden yolu tanımlasın.

Adım 3:  $Q$ 'da  $v_j$  düğümüne giren bütün ayrıtlar servis verilen ayrıtlarsa *Adım 4*'e git. Diğer durumlarda,  $(v_k, v_j)$   $Q$  üzerindeki servis verilmeyen ayrıt olsun.  $Q$  üzerinde  $(v_k, v_j)$ 'ye kadar olan ayrıtlardan  $C = (v_j, \dots, v_k, v_j)$  çevrimini elde et. Çevrimin yönünü değiştir ve *Adım 2*'ye git.

Adım 4: Eğer  $v_i$  ve  $v_j$  düğümlerini bağlayan en kısa yolun  $(SP_{ij})$  uzunluğu,  $P$ 'den daha kısa ise tur üzerinde  $P$  yerine  $SP_{ij}$ 'yi yerleştir.

Adım 5: İyileşme elde edilmeyene kadar  $T$  turu üzerindeki her iki yönü ve her olası  $v_i$  düğümünü de göz önüne alarak tekrar tekrar *Adım 2*'den *Adım 4*'e algoritmayı uygula.

Kısaltma algoritması servis verilen ayrıtları turun bir tarafında toplayarak, diğer ucu mümkün olduğunca uzun tutacak alternatifler arar.

#### 4. KAPASİTELİ AYRIT ROTALAMA PROBLEMİ

Kapasiteli ayrıt rotalama problemi (KARP) izleyen şekilde tanımlanabilir. Yönsüz bağlı bir  $G(V, E)$  seriminde,  $V$  düğüm kümesini,  $E$  ayrıt kümesini ve  $R \subseteq E$  gezilmesi gereken ayrıt kümesini gösterebilir. Kapasitesi  $Q$  olan özdeş  $K$  adet araç (homojen filo) depo düğümünde bulunmaktadır. Serimdeki her  $(i, j)$  ayrıtının bir maliyeti  $(c_{ij})$  ve gezilmesi gereken ayrıtların  $(\forall(i, j) \in R)$  bir talep değeri  $(q_{ij})$  vardır. Bir araç, turuna depo düğümünden başlayıp depoda bitirmeli ve servis verdiği ayrıtların toplam talebi aracın kapasitesini aşmamalıdır. KARP'ın amaç fonksiyonu bütün araçların toplam tur maliyetinin en küçüklenmesidir. KARP'ın birçok pratik uygulaması vardır. Örnek olarak, posta dağıtımı, çöp toplama, yollardaki buzlanmanın engellenmesi için yolların tuzlanması vb. verilebilir.

Kapasiteli ayrıt rotalama problemi ilk kez Golden ve Wong tarafından tanımlanmıştır (1981). Aynı çalışmada Golden ve Wong kapasiteli Çinli postacı probleminin (KARP'ın özel bir durumu,  $R = E$ ) ayrıt maliyetleri üçgen eşitsizliğini sağlarken en iyi çözüme  $3/2$  yakınlıkta çözüm bulmanın  $NP$ -zor olduğunu göstermişlerdir. KARP'ta serimdeki bütün ayrıtların gezilmesi gerekiyorsa ( $R = E$ ) KARP'ın özel bir durumu oluşur. Bu probleme Kapasiteli Çinli Postacı Problemi (KÇPP) denir. KÇPP Christofides tarafından 1973 yılında tanımlanmıştır (Ahr, 2004).

İzleyen alt bölümlerde KARP için önerilmiş matematiksel modeller, çözüm yöntemleri ve literatürde yer alan KARP türleri anlatılmıştır.

##### 4.1. Matematiksel Model

KARP'ın ilk matematiksel modeli Golden ve Wong tarafından KARP'ın ilk kez tanımlandığı çalışmada verilmiştir (1981). Önerilen matematiksel model yönsüz serimlerdeki turların belirlenmesi için önerilmiştir  $(c_{ij} = c_{ji}, \forall(i, j) \in E)$ . Yönsüz KARP'ın 0-1 tamsayılı modeli aşağıda verilmiştir.

**Model 4.1.** *Yönsüz KARP'in tamsayılı modeli* (Golden and Wong, 1981).

Kümeler

$V$	Düğüm kümesi $\{1, 2, \dots, n\}$
$E$	Ayrıtlar kümesi
$V'$	Depo düğümü hariç düğüm kümesi $\{2, \dots, n\}$
$S$	$V'$ kümesinin herhangi bir alt kümesi
$\tilde{S}$	$S$ kümesinin tümleyen kümesi $V \setminus S$
$R$	Gezilmesi gereken ayrıtlar kümesi
$K$	Araçlar kümesi $\{1, 2, \dots, m\}$

İndisler

$i, j$	Düğüm indisleri
$k$	Araç indisleri
$s$	$S$ alt kümelerinin indisi

Parametreler

$n$	Serimdeki düğüm sayısı
$m$	Araç sayısı
$c_{ij}$	$(i, j)$ ayrıtlarının maliyeti $\forall (i, j) \in E$
$q_{ij}$	$(i, j)$ ayrıtlarının talep değeri $\forall (i, j) \in R$
$Q$	Araçın kapasitesi

Karar Değişkenleri

$$y_{ijk} = \begin{cases} 1 & \text{Eğer } (i, j) \in R \text{ ayrıtlarına } k \text{ turunda servis veriliyorsa} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{Eğer } (i, j) \in E \text{ ayrıtlarından } k \text{ turunda geçiliyorsa} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

Model

$$\sum_{p \in V} x_{pik} - \sum_{p \in V} x_{ipk} = 0 \quad \forall i \in V, \quad k = 1, 2, \dots, m \quad (4.1)$$

$$x_{ijk} \geq y_{ijk} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (4.2)$$

$$\sum_{k=1}^m (y_{ijk} + y_{jik}) = 1 \quad \forall (i, j) \in R, \quad R \subseteq E \quad (4.3)$$

$$\sum_{(i,j) \in R} q_{ij} \cdot y_{ijk} \leq Q, \quad k = 1, 2, \dots, m \quad (4.4)$$

$$\left. \begin{array}{l} \sum_{(i,j) \in (S,S)} x_{ijk} - n^2 \cdot t_{sk} \leq |S| - 1 \\ \sum_{(i,j) \in (S,\bar{S})} x_{ijk} + u_{sk} \geq 1 \\ t_{sk} + u_{sk} \leq 1 \end{array} \right\} \begin{array}{l} V' = \{2, 3, \dots, n\} \\ \forall S \subseteq V', S \neq \{ \} \\ k = 1, 2, \dots, m \end{array} \quad (4.5)$$

$$y_{ijk} \in \{0, 1\} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (4.6)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in E, \quad k = 1, 2, \dots, m \quad (4.7)$$

$$t_{sk} \in \{0, 1\} \quad \forall s, \quad k = 1, 2, \dots, m \quad (4.8)$$

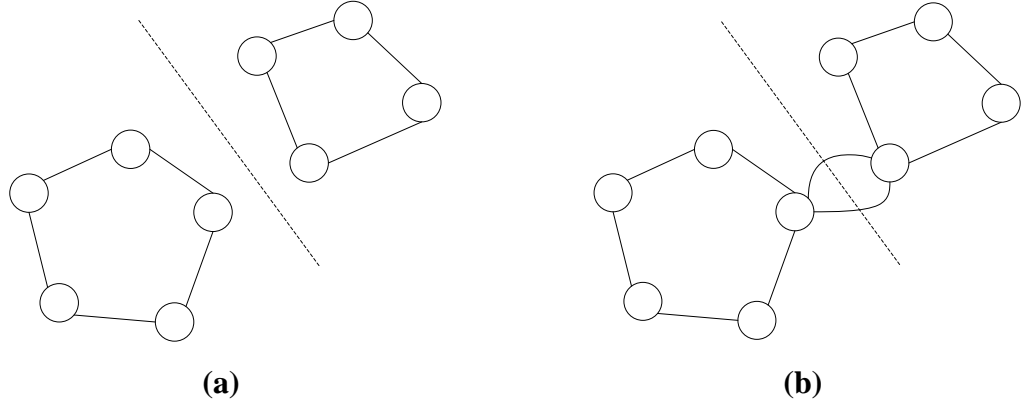
$$u_{sk} \in \{0, 1\} \quad \forall s, \quad k = 1, 2, \dots, m \quad (4.9)$$

kısıtları altında,

$$enk \left( \sum_{(i,j) \in E} \sum_{k=1}^m c_{ij} \cdot x_{ijk} \right) \quad (4.10)$$

(4.1) numaralı kısıt grubu akışın korunmasını sağlar (tur devamlılığı kısıtı). (4.2) numaralı kısıt ilişkilendirme kısıtıdır. Eğer bir  $(i, j)$  ayrıtına  $k$  – ncı araç servis verdiyse,  $k$  – ncı araç  $(i, j)$  ayrıtından geçmek zorundadır. (4.3) numaralı kısıt serimdeki bütün gezilmesi gereken ayrıtlara  $(\forall (i, j) \in R)$  bir kez servis verilmesini garantiler. (4.4) numaralı kısıt grubu araçların kapasite kısıtını aşmamasını garantiler. (4.5) numaralı kısıt grubu kural dışı alt turların oluşmasını engeller. (4.6), (4.7), (4.8) ve (4.9) numaralı denklemler karar değişkenlerinin 0-1 ve tamsayı olduğunu gösterir. (4.10) numaralı denklem modelin amaç fonksiyonudur ve toplam tur maliyetinin en küçüklenmesini sağlar. Modelde  $t_{sk}$  ve  $u_{sk}$  değişkenleri (4.5) numaralı kısıt grubunda yer alan kısıtlardan birinci ve ikincisinin sıkı veya gevşek olmasını sağlar.

Kapasiteli ayrıt rotalama probleminde, düğüm gezme probleminden farklı olarak alt turlara izin verilmektedir. Fakat her araca ait olan alt turlar başka ayrıtlarla birbirine bağlı olmalıdır. (4.5) numaralı kısıt grubu kopuk alt turların oluşmasını engeller. İzin verilen ve izin verilmeyen alt turlar Şekil 4.1’de verilmiştir.



**Şekil 4.1.** a) Kural dışı alt turlar, b) Uygun alt turlar

(4.5) numaralı kısıt grubu modele üstel sayıda kısıt eklenmesine neden olur. Bu kısıt grubu,  $n$  serimdeki düğüm sayısı olmak üzere  $V' = \{2,3,\dots,n\}$  kümesinin boş olmayan her alt kümesi ve araç sayısı için yazılmalıdır. Bu da modele  $(3m(2^{n-1} - 1))$  tane kısıt eklenmesi demektir.

Model 4.1'de verilen matematiksel model sadece yönsüz serimlerin çözümünde kullanılabilir. Karma ve yönlü serimlerdeki KARP'ın çözülebilmesi için Dror ve Leung (1998) tarafından yeni bir matematiksel model önerilmiştir (Dror and Langevin, 2000).

**Model 4.2.** *KARP'ın karma tamsayı modeli.*

Kümeler ve Parametreler

$M$  Büyük sabit bir sayı

$V[S]$   $S$  ayrıt alt kümesine komşu olan düğüm kümesi

Karar Değişkenleri

$$y_{ijk} = \begin{cases} 1 & \text{Eger } (i,j) \in R \text{ ayrıtına } k \text{ turunda servis veriliy orsa} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

$x_{ijk}$   $k$  turunda  $(i, j)$  ayrıtından geçiş sayısı

Model

$$\sum_{p \in V} x_{pik} - \sum_{p \in V} x_{ipk} = 0 \quad \forall i \in V, \quad k = 1, 2, \dots, m \quad (4.1)$$

$$\sum_{k=1}^m y_{ijk} = 1 \quad \forall (i, j) \in R, R \subseteq E \quad (4.11)$$

$$\sum_{(i,j) \in R} q_{ij} \cdot y_{ijk} \leq Q, \quad k = 1, 2, \dots, m \quad (4.4)$$

$$x_{ijk} \geq y_{ijk} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (4.2)$$

$$M \sum_{i \notin V[S], j \in V[S]} x_{ijk} \geq \sum_{(j,p) \in S} x_{jpk} \quad \forall S \subseteq R, \quad 0 \notin V[S], \quad k = 1, 2, \dots, m \quad (4.12)$$

$$y_{ijk} \in \{0,1\} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (4.6)$$

$$x_{ijk} \in Z^+ \quad \forall (i, j) \in E, \quad k = 1, 2, \dots, m \quad (4.13)$$

kısıtları altında,

$$enk \left( \sum_{(i,j) \in E} \sum_{k=1}^m c_{ij} \cdot x_{ijk} \right) \quad (4.10)$$

(4.11) numaralı kısıt gezilmesi gereken bütün ayrıtlara bir kez servis verilmesini garantiler. (4.12) numaralı kısıt grubu kural dışı alt tur oluşmasını engeller. (4.13) numaralı ifade  $x_{ijk}$  karar değişkeninin tamsayı olduğunu gösterir.

Model 4.2 verilen matematiksel modelin Golden ve Wong tarafından önerilen modelden en büyük farkı  $x_{ijk}$  karar değişkenidir. Golden ve Wong tarafından önerilen modeldeki  $x_{ijk}$  karar değişkeni 0-1 tamsayı iken bu modelde tamsayıdır. Bu da bir ayrıttın birden fazla sefer ziyaret edilmesine izin vermek anlamına gelmektedir. Ayrıca bu modelde tamsayı değişkenlere uygun kuraldışı alt turları engelleyecek kısıt grubu önerilmiştir.

## 4.2. Çözüm Yöntemleri

KARP'ın çözüm yöntemleri; eniyi çözümün garanti edildiği kesin çözüm yöntemleri, uygun çözümlerin elde edildiği sezgisel yöntemler ve son olarak en iyiye

mümkün olduğunca yakın çözüm bulmak için kullanılan metasezgisel yöntemlerdir. İzleyen alt bölümlerde çözüm yöntemleri detaylandırılmıştır.

#### 4.2.1. Alt sınırlar ve kesin çözüm yöntemleri

Literatürde KARP için alt sınırların türetilmesinde beş farklı alt sınır türetme yöntemi kullanılmıştır. Bunlar MLB (Matching lower bound), NSLB (Node scanning lower bound), MNSLB (Matching-Node scanning lower bound), NDLB (Node duplication lower bound) ve MLB'nin geliştirilmiş hali olan LB1 (Lower Bound-1) şeklinde tanımlanabilir (Wøhlk, 2006).

KARP'ın kesin çözümünün elde edilmesi için ilk çalışma Hirabayashi vd. tarafından 1992 yılında yapılmıştır (Hirabayashi, et.al., 1992a). Belenguer ve Benavent tarafından 1998 yılında geçerli eşitsizlikler (valid inequalities) ve 2003 yılında kesme düzlemleri (cutting plane) yaklaşımlarıyla iki farklı kesin çözüm yöntemi önerilmiştir.

Kesin çözüm yöntemlerinin elde edilmesinde diğer bir yaklaşım da bir KARP'ın eş değer düğüm gezme problemine dönüştürülmesidir. Bunu için ilk yöntem önerisi (Pearn, et.al, 1987) çalışmasında yapılmıştır. Daha sonrasında (Baldacci and Maniezzo, 2006) ve (Aragao, et.al, 2006) çalışmalarında dönüştürme sırasında her ayrıt için kullanılacak düğüm sayısı azaltılmıştır.

#### 4.2.2. Sezgisel algoritmalar

Bu alt bölümde KARP'ın çözülmesi için önerilen sezgisel yöntemler anlatılmıştır. KARP sezgiselleri için (Hertz and Mittaz, 2000), (Wøhlk, 2006) ve (Assad and Golden, 1995) çalışmalarına bakılabilir. Bu sezgiseller yapıcı metotlar (constructive methods) ve iki fazlı yapıcı metotlar olmak üzere ikiye ayrılır. İki fazlı yapıcı metotlar ise önce rotala sonra paylaşır (route first cluster second) ve önce paylaşır sonra rotala (cluster first route second) olmak üzere ikiye ayrılmaktadır.

#### 4.2.2.1. Yapıcı metotlar

Christofides tarafından 1973 yılında Construct-Strike (CS) algoritması önerilmiştir. Bu algorithmada servis verilmesi gereken ayrıtlardan henüz seçilmemiş, kapasite kısıtını aşmayan ve çıkarıldığında serimin geri kalanı kopuk olmayacak şekilde ayrıt seçilir. Kalan serim en küçük maliyetli eşleştirme algoritması ile çift serim haline dönüştürülür. Serim üzerindeki bütün servis verilmesi gereken ayrıtlar oluşturulan turların içindeyse algoritma durdurulur, diğer durumlarda ayrıt seçme adımına geri dönülür.

Pearn tarafından 1989 yılında CS algoritmasının değiştirilmiş halini önermiştir. Bu algoritma Modified-Construct-Strike (MCS) olarak adlandırılır. MCS algoritmasında serimin bağlı olma zorunluluğu kaldırılmıştır. Bu algorithmada ayrıtlar art arda en yakın depoya dönüşünü en büyükleyecek şekilde turlara eklenir. Serim ayrıt koparıldığında kopuk oluyorsa en küçük maliyetli eşleştirme öncesi en küçük kapsar ağaç ile serim bağlı hale getirilir.

Golden, DeArmon ve Baker tarafında 1983 yılında Path-Scanning (PS) algoritması önerilmiştir. PS algoritması, (Golden, et.al., 1983) çalışmasında verilen beş farklı kural ile henüz eklenmemiş ayrıtlar listesinden araç kapasitesini aşmayacak şekilde ayrıt seçer. Araç kapasitesi aşıldığında kalınan noktadan en kısa yol ile çevrim depo düğümüne bağlanır. Sezgisel sonucunda elde edilen beş farklı sonuçtan en küçük maliyetli olan uygulanır.

Araç rotalama problemi için önerilen Clarke ve Wright algoritmasından hareketle (Golden, et.al., 1983) çalışmasında KARP için Augment-Merge (AM) algoritması önerilmiştir. Algoritma iki fazdan oluşur. İlk fazda her seferinde bir servis verilmesi gereken ayrıt eklenecek şekilde çevrim oluşturulur. İkinci fazda aracın kapasitesini aşmayacak şekilde bir önceki fazda oluşturulan çevrimler birleştirilir.

Chapleau, vd. 1984 yılında GSP'deki ekleme prosedürüne benzeyen Parallel-Insert (PI) isimli sezgiseli önermişlerdir. Bu çalışmada toplam tur uzunluğunun



minimize edilmesinin yanı sıra tur uzunluklarının dengeli olması istenmektedir. Kaç tane tura ihtiyaç olacağını belirlenir ve depoya en uzakta olan ayrıta servis verecek çevrim oluşturulur. Bundan sonra birinci ekleme stratejisi seçilen ayrıtın hangi tura ekleneceği, ikinci ekleme stratejisi ise verilen tura hangi ayrıtın ekleneceğinin belirlenmesidir. Algoritma bu iki ekleme stratejisi ile sürdürülür.

Pearn tarafından 1991 yılında AM ve PI algoritmalarındaki fikirleri birleştiren Augement-Insert isimli algoritma önerilmiştir. AM algoritmasının çoğaltma fazında olduğu gibi uygun araç turları türetilir. İkinci fazda ise PI algoritmasının birinci ekleme fazına benzer şekilde türetilen uygun araç turlarına ayrıt eklemeleri yapılır.

(Hirabayashi, et.al., 1992b) çalışmasında önerilen düğüm çoğaltma alt sınırı (node duplication lower bound) yöntemi temel alınarak (Wøhlc, 2006) çalışmasında düğüm çoğaltma sezgiseli önerilmiştir. Aynı çalışmada, AM ve PS sezgisellerinde kullanılan fikir birleştirilerek Double Outer Scan isimli sezgisel önerilmiştir. Bu sezgisel turları AM sezgiselindeki gibi en uzaktaki servis verilmemiş ayrıta göre oluşturulur. Daha sonra kısa turlar birleştirilir. Ayrıca PS sezgiselinde olduğu gibi tura dâhil edilecek ayrıt, ayrıt seçim kurallarına göre belirlenir.

(Santos, et.al., 2009) çalışmasında KARP'ın çözümü için elips kuralı tabanlı bir sezgisel önerilmiştir. Bu yaklaşım PS sezgiseli temel alınarak geliştirilmiştir. PS sezgiselinde turlar seçim kurallarına göre oluşturulur. Bu çalışmada ayrıt elipsin içerisindeyse eklenmektedir.

#### **4.2.2.2. İki fazlı oluşturma metotları**

İki fazlı oluşturma yöntemleri iki kategoriye ayrılır. Önce rotala sonra paylaşır stratejisinde ilk olarak bütün gezilmesi gereken yönlü ve yönsüz ayrıtları içeren dev bir Euler turu oluşturulur. Daha sonraki fazda ise bu tur uygun araç turlarına paylaşılır. Ters olarak önce paylaşır-sonra rotala sezgisellerinde ilk olarak gezilmesi gereken

yönlü ve yönsüz ayrıtlar araç kapasitesini aşmayacak şekilde bölümlendirilir. Daha sonra her küme için KPP algoritması ile araç turları belirlenir.

#### Önce Rotala-Sonra Paylaştır

Önce rotala sonra paylaştır algoritmalarının ilk fazında yer alan Euler turun oluşturulması için Bölüm 3'te verilen yöntemler kullanılabilir. Ulusoy tarafından 1985 yılında toplam tur uzunluğunun en küçüklenmesinin yanı sıra kullanılan araç sayısını da en küçükleyen bir algoritma önerilmiştir. Algoritma ilk olarak türetilen Euler turunu uygun araç turlarına böler. Daha sonra her ayrıtı türetilen bir tura karşı gelen yönlü bir serim oluşturulur. Her yönlü ayrıtın maliyeti o ayrıta karşı gelen turun uzunluğudur. Elde edilen yönlü serim üzerinde başlangıç düğümünden bitiş düğümüne en kısa yolun belirlenmesiyle türetilen turlar içerisinde minimum maliyetli ve bütün servis verilmesi gereken ayrıtları içeren araç turları belirlenir (Ulusoy, 1985). Bu tezde önerilen çözüm yöntemlerinden birisi Ulusoy'un sezgiselinin değiştirilmiş hali olduğundan bu algoritma detaylı olarak verilmiştir. Ulusoy'un algoritmasının adımları aşağıdaki gibidir.

#### **Algoritma 4.1.** *Ulusoy'un bölümlendirme algoritması*

**Girdi:** Yönsüz  $G = (V, E)$  serimi ve  $R$ 'yi kapsayan  $T$  turu.

**Çıktı:**  $G$ 'de tanımlı  $R$ 'yi kapsayan uygun araç turları.

*Adım 0:* Eğer gezilmesi gereken bir ayrıt birden fazla sefer ziyaret ediliyorsa, bu ayrıta ilk geçişte servis ver.

*Adım 1:*  $G$ 'deki düğümleri  $T$  turuna göre yeniden etiketle.  $v_0$  düğümü depo düğümü olmak üzere,  $T$  turu  $(v_0, v_1, \dots, v_t = v_0)$ 'a eşittir.  $r$ ,  $T$  turunda en son servis verilen ayrıta komşu olan düğümün indeksi olsun. Düğüm kümesi  $V' = \{v_0, v_1, \dots, v_r\}$  olan  $G' = (V', A)$  yönlü serimini oluştur.  $T$  turu üzerinde yol parçası olan  $(v_a, \dots, v_b)$ 'deki toplam yük araç kapasitesini ( $Q$ ) aşmıyorsa,  $G'$  seriminin ayrıt kümesine  $A(b > a)$ ,  $(a, b)$  ayrıtını ekle. Eğer  $b > a + 1$  ve  $(v_a, v_{a+1})$  veya  $(v_{b-1}, v_b)$   $T$  turu üzerinde servis verilen bir ayrıt değilse  $G'$  seriminden  $(a, b)$  ayrıtını çıkar.  $G'$  serimindeki ayrıtların  $(a, b)$  maliyetleri  $(c'_{ab})$  aşağıdaki gibi hesaplanmaktadır.

- Eğer  $b = a + 1$  ve  $(v_a, v_{a+1})$   $T$  üzerinde servis verilen bir ayrıt değilse,  $c'_{ab} = 0$ .
- Eğer  $b > a + 1$  veya  $(v_a, v_{a+1})$   $T$  üzerinde servis verilen bir ayrıt ise  $(v_a, \dots, v_b)$  aralığındaki tur parçasını gösteren  $P_{ab}$  zinciri için aşağıdaki durumları kontrol et.
  - $P_{ab} = (v_a, \dots, v_0, \dots, v_b)$  zinciri depo düğümünü içeriyorsa;  $P_{ab}$  zincirine  $v_b$  ve  $v_a$  düğümleri arasındaki en kısa yolu gösteren  $SP_{ab}$  en kısa yolunu ekle. Bu durumda  $C_{ab} = (v_0, \dots, v_a, SP_{ab}, v_b, \dots, v_0)$  çevrimi elde edilir.  $C_{ab}$  çevrimin uzunluğu  $c'_{ab}$ 'yi tanımlar.
  - $P_{ab} = (v_a, \dots, v_b)$  zinciri depo düğümünü içermiyorsa,  $P_{ab}$  zincirine  $SP_{v_0 a}$  ve  $SP_{b v_0}$  en kısa yollarını ekle. Bu durumda  $C_{ab} = (SP_{v_0 a}, v_a, \dots, v_b, SP_{b v_0})$  çevrimi elde edilir.  $C_{ab}$  çevrimin uzunluğu  $c'_{ab}$ 'yi tanımlar.

*Adım 2:*  $G'$  seriminde  $O'$ 'dan  $r'$ 'ye en kısa yolu bul. En kısa yol üzerinde kullanılan her  $(a, b)$  ayrıtı  $G$  üzerindeki uygun bir araç turunu gösterir.

(Hertz, et.al., 2000) çalışmasında bir Euler turunu uygun araç turlarına dönüştüren CUT isimli bir sezgisel algoritma önerilmiştir. Bu algoritma toplam talep değeri  $D$  olan ve araç kapasitesini ( $Q$ ) aşan bir  $T = v_0, v_1, \dots, v_t = v_0$  turunu uygun araç turlarına böler. İlk olarak tur üzerinde  $P = (v_0, v_1, \dots, v_r)$  aralığındaki toplam yükün araç kapasitesini aşmayacağı ve  $S = (v_r, \dots, v_t = v_0)$  aralığında yükün  $Q \lceil D/Q \rceil$  değerini aşmayacağı bir  $v_r$  düğümü belirlenir. Daha sonra  $P$  tur parçası  $v_r$  düğümünden depo düğümüne bağlanarak bir çevrim elde edilir.  $(v_r, v_{r+1})$  ayrıtı  $T$  üzerinde servis verilen bir ayrıt olması koşuluyla en küçük indeksli  $r' \geq r$  belirlenir. Daha sonra  $(v_0, v_1, \dots, v_{r'})$  tur parçası yerine  $v_{r'}$  ve depo düğümü arasındaki en kısa uzaklık konulur ve bu yeni tur parçası için algoritma tekrar çalıştırılır.

### Önce Paylaştır-Sonra Rotala

Önce paylaşır-sonra rotala algoritmalarında ilk olarak gezilmesi gereken ayrıtlar kümelerine paylaştırılır. Daha sonra her küme için araç rotaları oluşturulur. (Benavent, et.al., 1990) çalışmasında Cycle-Assignment isimli önce paylaşır-sonra rotala sınıfına

giren bir algoritma önerilmiştir. Bu algoritmada, ilk olarak kapasite kısıtına uygun kümeler sezgisel algoritma ile oluşturulur. Daha sonra ayırıt taleplerinin araçlara atanması genelleştirilmiş atama modelinin çözümü ile belirlenir. Son olarak her araca ait rota, Bölüm 3'te verilen KPP çözüm yöntemleriyle belirlenebilir.

#### 4.2.3. Metasezgisel algoritmalar

Metasezgisellerin NP-zor olan KARP'a kısa sürelerde iyi çözümler (türetilen alt sınırlara yakın çözümler) vermesi nedeniyle, son yıllarda KARP ile ilgili yapılmış çalışmaların büyük kısmında çözüm yöntemi olarak metasezgisel algoritmalar kullanılmaktadır. Metasezgisel algoritmalarla ilgili detaylı bir derleme olan (Glover and Kochenberger, 2003) çalışmasına bakılabilir.

Hertz vd. 2000 yılında CARPET isimli bir yasaklı arama (tabu search) algoritması önermiştir. Başlangıç çözümü, Frederickson'ın sezgiseli (Bkz. Bölüm 3) ile gezilmesi gereken ayırıtları kapsayan Euler turunun CUT algoritması kullanılarak parçalanmasıyla elde edilir. Bu çalışmada uygun olmayan çözümlere cezalandırılması koşuluyla izin verilmektedir. (Amberg, et.al., 2000) çalışmasında çoklu depolu KARP için yasaklı arama algoritması önerilmiştir. Greistorfer tarafından 2003 yılında dağıtık arama, yasaklı arama ile birleştirilerek KARP'ın çözümü için yasaklı dağıtık arama (tabu scatter search) algoritması önerilmiştir. (Brandao and Eglese, 2008) çalışmasında deterministik yasaklı arama (deterministic tabu search) algoritması ile klasik KARP'a çözüm önerilmiştir. Bu çalışmada uygun olmayan çözümlere de izin verilmektedir. Başlangıçta belirtilen kurallara göre başlangıç çözümleri türetilir ve yasaklı arama arđıřtırmaları da rastsallık içermediğinden bir problem örneğİ için aynı çözüm tekrar elde edilebilir. (Mei, et.al., 2009a) çalışmasında düzeltme tabanlı olan yasaklı arama algoritmalarında uygun çözüm üretmek için bir operatör önerilmiştir.

KARP'ın çözümü için Lacomme vd. tarafından 2001 yılında bir genetik algoritma önerilmiştir. Bu çalışmada kromozom, servis verilmesi gereken ayırıtlara hangi sırada servis verileceğİ bilgisini içermektedir. Daha sonra bu bilgi bir

bölümlendirme algoritması kullanılarak uygun araç turlarına dönüştürülür. Aynı yazarlar tarafından çalışmanın devamı olarak 2004 yılında bir memetik algoritma önerilmiştir. Bu çalışmada mutasyona yerel arama eklenmiştir (Lacomme, et.al, 2004). (Lacomme, et.al, 2006) çalışmasında iki amaçlı KARP (toplam tur uzunluğunun en küçüklenmesi ve en uzun tur uzunluğunun en küçüklenmesi) için NSGA (Non-dominated sorted genetic algorithm) yapısı kullanılmıştır. Sulama araçlarının rotalanmasından hareketle KARP'ın çözümü için (Deng, et.al., 2007) çalışmasında mutasyonu yerel arama olan bir genetik algoritma önerilmiştir. Aynı motivasyondan hareketle (Zhu, et.al., 2007) çalışmasında çoklu depolu KARP için melez genetik algoritma önerilmiştir.

(Doerner, et.al., 2003) çalışmasında KARP'ın çözümü için karınca kolonisi optimizasyonu (ant colony optimization) yöntemi kullanılmıştır. Bu çalışmada karıncaların kararı bir sonraki servis verilecek gezilmesi gereken ayrıt şeklindedir. Sezgisel bilgi ayrıt oryantasyonuna göre en kısa yol bilgisidir. Feromon bilgisi ayrıt oryantasyonlarını yok sayarak servis verilecek ayrıtların sırasını içerir. Bu çalışmanın devamında (Doerner, et.al., 2004) çalışmasında yine karınca kolonisi optimizasyonu yöntemi kullanılmıştır. (Lacomme, et.al., 2004) çalışmasında ise karınca kolonisi optimizasyonu yöntemi yerel arama algoritması ile birleştirilmiştir.

(Prins, et.al., 2009) çalışmasında kapasiteli araç rotalama ve KARP için önce rotala sonra paylaşır tabanlı sezgisel algoritmaların GRASP (Greedy randomized adaptive search) ve ILS (Iterated local search) ile birleştirilmesi anlatılmıştır. Zaman pencereli KARP'ın incelendiği (Reghioui, 2009) çalışmasında rassallaştırılmış PS ve rassallaştırılmış önce rotala-sonra bölümlendir algoritmalarının kullanıldığı bir GRASP önerilmiştir.

KARP için (Beullens, et.al., 2003) çalışmasında GLS (Guided local search) sezgiseli kullanılarak bir çözüm yöntemi önerilmiştir. GLS sezgiselinde tek rotalı hareketler ve iki rotalı hareketler olmak üzere iki farklı komşuluk hareketi tanımlanmıştır. Çözüm kalitesini arttırmak ve çözüm zamanından kazanmak için komşuluk listesi ve ayrıt işaretleme mekanizmaları kullanılmıştır.

Hertz ve Mittaz tarafından klasik KARP'ın çözümü için VND (Variable neighborhood descent) algoritması önerilmiştir (2001). Bu çalışmada komşuluk hareketleri ADD, DROP ve SWITCH şeklindedir. Daha sonra bu turlar CUT algoritması kullanılarak uygun araç turlarına dönüştürülmektedir. Bu çalışmada kullanılan algoritmalar (Hertz, et.al., 2000) çalışmasında önerilmiştir. (Tagmouti, et.al., 2008) çalışmasında zamana bağlı servis maliyetlerinin değiştiği KARP'ın çözümü için VND algoritması önerilmiştir. Bu çalışmada başlangıç çözümü Clarke ve Wright tarafından önerilen kazanım algoritması temel alınarak geliştirilen bir sezgisel ve ekleme sezgiseli kullanılarak elde edilmektedir. Komşuluk hareketleri ise ayır hareketi, karşılıklı yer değiştirme ve blok halinde yer değiştirme şeklindedir. KARP'ta ara tesislerin de göz önüne alındığı (Polacek, et.al., 2008) çalışmasında tanımlanan problemin çözümü için VNS (variable neighborhood search) algoritması önerilmiştir. Bu çalışmada başlangıç çözümü gezilmesi gereken ayrıtlara kapasite kısıtı altında sırasıyla hizmet verecek şekilde veya Ulusoy'un sezgiseli kullanılarak türetilmiştir.

### **4.3. Kapasiteli Ayır Rotalama Problemi Türleri**

Kapasiteli ayır rotalama problemi, en genel haliyle yönsüz bir serim üzerinde, araç filosunun homojen olduğu ve amaç fonksiyonun toplam gidilen yolun en küçüklenmesi olarak tanımlanır. Fakat uygulama alanına bağlı olarak serimin tipi, araç filosu, çok amaçlılık veya ek kısıtlar tanımlanabilir. İzleyen kısımda KARP'ın türleri hakkında yapılmış çalışmaların detayları verilmiştir.

#### **4.3.1. Farklı serim türleri için KARP**

Klasik kapasiteli ayır rotalama problemi yönsüz serim üzerinde tanımlanır. Fakat gerçek hayat uygulamalarında caddelerin tek yön olması veya caddenin iki yanına paralel olarak servis verilmesi gibi kısıtlarla karşılaşılabılır. Bu durum KARP'ın yönlü veya karma serimler üzerinde tanımlanmasını gerektirir. Ayrıca ayrıtların maliyet değeri

ayrıtı geçiş yönüne bağlı olarak değişiyorsa bu tip ayrıtlara rüzgârlı ayrıt olarak adlandırılır. Bazı KARP uygulamalarında bu tür ayrıtlarla karşılaşmaktadır.

(Belenguer et.al., 2006) çalışmasında karma serimler üzerinde tanımlanan KARP detaylı olarak incelenmiştir. Klasik KARP'ın çözümünde kullanılan Ulusoy'un sezgiseli, AM ve PS sezgiselleri karma KARP'ı çözebilecek şekilde uyumlandırılmıştır. Ayrıca bu çalışmada karma KARP için test problemleri türetilmiş ve bu test problemlerine ait alt sınır ve üst sınır değerleri elde edilmiştir. Karma KARP'ın alt sınırları (Gouveia, et.al., 2009) çalışmasında da incelenmiştir.

(Lacomme, et.al., 2004) çalışmasında Genişletilmiş-KARP (GKARP) olarak adlandırılan bir problem tanımlanmış ve memetik algoritma ile çözüm bulunmuştur. (Lacomme, et.al., 2001) çalışmasında GKARP için tamsayılı doğrusal bir model önerilmiş ve (Golden, et.al., 1983) çalışmasında türetilen test problemlerinden 11 tanesini XPRESS ile çözülmüş ve en iyi çözümler elde edilmiştir. (Lacomme, et.al., 2002) çalışmasında ise GKARP'ın çok amaçlı halinin (en fazla yük ve toplam tur uzunluğu) de göz önüne alındığı durum için kesme düzlemleri algoritması ve genetik algoritma önerilmiştir. Şehirdeki gündelik çöplerin toplanması uygulamasından hareketle (Amado and Mourão, 2005) çalışmasında karma serimlerdeki GKARP için sezgisel algoritma önerilmiştir. (Maniezzo, 2004) çalışmasında da çöplerin toplanma uygulaması el alınmıştır. Bu çalışmada serim tamamen yönlü bir serime dönüştürülerek çözülmüştür.

#### **4.3.2. Alternatif amaç fonksiyonlu KARP**

Kapasiteli ayrıt rotalama problemlerinde çoğunlukla amaç fonksiyonu filodaki bütün araçların gezdiği toplam yolun en küçüklenmesidir. Fakat birtakım uygulamalarda başka amaç fonksiyonlarının da göz önüne alınması gerekebilir. Örnek olarak kullanılan araç sayısının en küçüklenmesi, tur yüklerinin dengelenmesi, en uzun tur uzunluğunun en küçüklenmesi verilebilir.

(Lacomme, et.al., 2003) ve (Lacomme, et.al., 2006) çalışmalarında çok amaçlı KARP göz önüne alınmıştır. Klasik KARP'ta yer alan toplam tur uzunluğunun en küçüklenmesinin yanı sıra en uzun tur uzunluğu da en küçüklenmektedir. Yazarlar, çok amaçlı KARP'ın çözümü için genetik algoritma önermişler ve literatürdeki test problemleriyle karşılaştırmışlardır.

Ulusoy tarafından 1985 yılında önerilen sezgisel algoritma da toplam tur uzunluğu ile birlikte kullanılan araç sayısı da en küçüklenmektedir. Bu çalışmada, ilk olarak bütün gezilmesi gereken ayrıtları kapsayan bir Euler turu elde edilir ve bu tur uygun araç turlarına bölünür. Heterojen filonun da göz önüne alındığı çalışmada uygun araç turları maliyetleri içerisinde sabit araç tahsis maliyetleri de dâhil edilmektedir. Bu nedenle önerilen sezgisel algoritma toplam tur uzunluğunun yanı sıra araç atama maliyetini de en küçüklemeyi amaçlamaktadır. Uygun araç turları türetildikten sonra en kısa yol problemi ile hangi araç turlarının kullanılacağına karar verilir (Ulusoy, 1985).

Mini-Maks k-Çinli postacı problemi (MM k-ÇPP), araçların sınırsız kapasitesi olduğu varsayımıyla en uzun tur uzunluğunu en küçüklemeyi amaçlayan postacı problemi türüdür. Bu problem ilk kez (Frederickson, et.al., 1978) çalışmasında tanımlanmıştır. MM k-ÇPP (Ahr, 2004) çalışmasında detaylı olarak incelenmiştir. Bu çalışmada MM k-ÇPP çözümü için Çoğalt-Birleştir algoritmasını temel alan bir sezgisel algoritma önerilmiştir. (Ahr and Reinelt, 2006) çalışmasında MM k-ÇPP için bir yasaklı arama algoritması önerilmiş ve büyük serimlerde (392 ayrıta kadar) test edilmiştir.

Yolların tuzlanması uygulamasında zamanında müdahale kritik olduğundan Gendreau vd. zamana bağlı servis maliyetlerinin göz önüne alındığı kapasiteli kısıtlı bir ayrıt rotalama problemi önermişlerdir (2007). Tanımlanan bu problemi, ayrıt rotalama problemini düğüm gezme problemine dönüştürüp sütun türetme yöntemi ile çözmüşlerdir.

(Archetti, et.al., 2009) çalışmasındaki amaç fonksiyonu diğer kapasiteli ayrıt rotalama problemlerinden farklı olarak en büyük şekilde tanımlanmıştır. Bu çalışmada



servis verilmesi gereken ayrıtlar için bir fayda değeri tanımlanmış ve toplam fayda değeri en büyüklenmektedir.

### 4.3.3. Çoklu depolu KARP

Çoklu depolu KARP (ÇD-KARP), KARP'ın kısıtları ile birlikte, her araç serim üzerinde tanımlanmış depoların birinde bulunmaktadır ve her araç turunu başladığı depoda bitirmek zorundadır. (Cattrysse, et.al., 2002) ve (Cattrysse, et.al., 2003) çalışmalarında serim alt serimlere (bölgelere) ayrılmıştır ve her bölgede bir yerel merkez (depo) bulunmaktadır. Bu durumda problem çoklu depo (multi-depot) problemine dönüşmektedir. (Amberg, et.al., 2000) çalışmasında ÇD-KARP çok merkezli ayrıt kısıtlı kapasiteli en küçük kapsar ağaç problemine dönüştürülerek çözülmüştür. (Zhu, et.al., 2007) çalışmasında sulama araçlarının rotalama probleminin ÇD-KARP'a benzediği anlatılmıştır. Bu çalışmada ÇD-KARP'ın çözümü için ise melez genetik algoritma önerilmiştir.

(Ghiani, et.al., 2001) çalışmasında bir ÇD-KARP sınıfı içerisinde yer alabilecek olan ara tesisli KARP (AT-KARP) tanımlanmıştır. Ara tesisli KARP, klasik KARP ile birlikte aracın ara noktalarda boşaltmaların veya tekrar doldurmaların göz önüne alındığı problem türü olarak tanımlanır. AT-KARP için kırsal postacı problemi temel alınarak ve problemin tamsayı doğrusal modeli gevşetilerek alt sınırlar elde edilmiştir. (Ghiani, et.al., 2004) çalışmasında AT-KARP'a filodaki araçlar için en uzun tur uzunluğu kısıtı eklenmiştir. Bu çalışmada tanımlanan problemin çözümü için üç farklı sezgisel önerilmiştir. İlki parçalama yaklaşımı tabanlı bir yapıcı algoritma, ikinci ve üçüncü sezgiseller ise yasaklı arama algoritmasının adaptasyonu şeklindedir. (Polacek, et.al., 2008) çalışmasında da AT-KARP ve tur uzunluğu kısıtlı AT-KARP incelenmiştir. Problemin çözümü için VNS algoritması önerilmiştir.

#### 4.3.4. Hareketli depolu KARP

(Filippi and Pia, 2006) çalışmasında İtalya'nın kuzeyinde bulunan Due Carre şehrindeki çöp toplama problemi incelenmiştir. Bu uygulamada karşılaştıkları problemi hareketli depolu (mobile-depot) KARP olarak isimlendirmişlerdir. Bunun sebebi çöp toplama işleminde uşak ve kompaktör adı verilen iki tip aracın kullanılması ve problemde klasik KARP'tan farklı olarak yeni kısıtların bulunmasıdır. Bir uşak tipi aracın kapasitesi dolduğunda filodaki herhangi bir kompaktör ile bir noktada buluşmakta ve çöplerini o araca aktarmaktadır Yazarlar, tanımlanan bu problemi Hertz ve Mittaz (2001) tarafından önerilen inişli VNS algoritmasını değiştirerek çözmüşlerdir.

Hareketli depolu KARP örneği olarak tanımlanabilecek yeniden doldurma noktalı KARP (Amaya, et.al., 2007) çalışmasında önerilmiştir. Bu çalışmada da filoda iki tip araç bulunmaktadır. Bunlardan bir tanesi kapasite kısıtlı bir servis aracı iken diğer tip araç bu araca ikmal ile görevlidir. Servis aracı ihtiyaç duyduğunda ikmal aracıyla bir noktada buluşup kapasitesini doldurur ve servis görevine devam eder. Tanımlanan problem için matematiksel model verilmiştir ve kesme düzlemleri algoritması ile çözülmüştür.

#### 4.3.5. Periyodik KARP

Uzun zaman ufkunun göze alındığı klasik KARP'ta her ayrıt talebine birden fazla sefer servis verilmesi gerekir. Bu problem Periyodik KARP (PKARP) olarak adlandırılır. Örneğin çöp toplama probleminde planlanan bir zaman ufkunda (haftada 2 veya 3 kez) çöplerin evlerden toplanması gerekebilir. (Chu, et.al, 2005) çalışmasında PKARP'ın matematiksel modeli verilmiştir. Ayrıca bu çalışmada problemin çözümü için sıklık tabanlı en yakın ekleme sezgiseli, ekleme sezgiseli ve iki fazlı sezgisel algoritmaları önerilmiştir. (Lacomme, et.al., 2005) çalışmasında PKARP için (Lacomme, et.al., 2004) çalışmasında önerilen memetik algoritmanın uzantısı olan bir çözüm yöntemi önerilmiştir. (Chu, et.al., 2006) çalışmasında ise PKARP'ın çözümü için

dağıtık arama (scatter search) algoritması önerilmiştir. (Chu, et.al., 2003) çalışmasında ise PKARP için iki farklı alt sınır türetme yöntemi önerilmiştir.

#### 4.3.6. Stokastik KARP

Stokastik KARP'ın (SKARP) klasik KARP'tan tek farkı ayrıt taleplerinin rassal değişken olmasıdır. Pratikte bu problemle kesin taleplerin belli olmadığı çöp toplama, posta dağıtımı ve kar küreme uygulamalarında karşılaşılır. SKARP ilk kez (Fleury, et.al, 2002) çalışmasında tanımlanmıştır. Problemin ileriki uzantılarına (Fleury, et.al, 2005) çalışmasında yer verilmiştir. (Fleury, et.al, 2004) çalışmasında ise SKARP'ın çözümü için (Lacomme, et.al., 2004) çalışmasında önerilen memetik algoritmanın genişletilmiş hali kullanılmıştır.

(Fleury, et.al., 2008) çalışmasında iki amaçlı SKARP incelenmiştir. Amaç fonksiyonlarından birisi toplam tur uzunluğunun en küçüklenmesi iken diğer amaç fonksiyonu en uzun tur uzunluğunun en küçüklenmesi şeklindedir. Tanımlanan iki amaçlı SKARP'ın çözümü için bir genetik algoritma önerilmiştir.

Yakın zamanda ise (Christiansen, et.al., 2009) çalışmasında ayrıt taleplerinin Poisson dağılımına uyduğu bir SKARP tanımlanmıştır. Problemin çözümü için B&P (branch and price) algoritması kullanılmıştır.

#### 4.3.7. Araç/Yön bağımlı KARP

Çöp toplama probleminde kullanılan araçlar farklı kapasite, boyut ve şekillerde olabilir. Her tip araç bütün ayrıtlara servis veremediği durumda, servis verilen ayrıtlar araç tipine bağımlı olmaktadır. Örneğin her araç dar vadi ve köprülerden geçemez. Bu nedenle problem araç/yön bağımlı KARP (Vehicle/Site Dependency) olarak adlandırılmıştır. Bir çöp toplama uygulaması olan (Ball, et.al., 2001) çalışmasında, yazarlar çözüm yöntemi olarak araç ayrıştırma algoritması önermişlerdir. (Bodin and

Snizek, 2006) çalışmasında ise tanımlanan problemin çözümü için karma tamsayılı matematiksel modeller önerilmiştir.

#### 4.3.8. Zaman Pencereleli KARP

Zaman pencereleli (time window) KARP (ZP-KARP), klasik KARP'ın kısıtları ile birlikte her servis verilmesi gereken ayrıt için daha önceden belirlenmiş bir zaman penceresi kısıtının olduğu problem türüdür. Örnek olarak, kışın yolların tuzlanması uygulamasında bazı caddeler iki saat içinde, diğerleri dört saat içerisinde tuzlanmalı gibi kısıtların olduğu problemde gözlenmektedir (Wøhlk, 2008).

(Ramdane-Cherif, 2006) çalışmasında GKARP zaman penceresi kısıtlarıyla incelenmiştir. Bu çalışmada tanımlanan problemin çözümü için iki farklı ekleme algoritması ve memetik algoritma önerilmiştir. ZP-KARP farklı görüş açılarından (Wøhlk, 2005) çalışmasında incelenmiştir. Bu çalışmada ZP-KARP için iki farklı matematiksel model önerilmiştir. Bunlardan ilki problemin tamsayılı doğrusal modeli diğeri ise düğüm gezme problemine karşı gelen modelidir. (Reghioui, et.al., 2007) çalışmasında ZP-KARP için GRASP ve yeniden yol bağlama sezgiselleri kullanılmıştır. Önerilen GRASP algoritmasında rassallaştırılmış yol tarama sezgiseli ve rassallaştırılmış önce tur oluştur sonra paylaşır sezgiselleri kullanılmıştır. Elde edilen çözümlerin iyileştirilmesi için ise OR-OPT, SWAP, ve 2-OPT yer arama algoritmaları kullanılmıştır. Johnson ve Wøhlk tarafından ZP-KARP'ın optimal olarak çözümü için sütun türetme yöntemi önerilmiştir. Bu çalışma, literatürde ZP-KARP'ın optimal olarak çözüldüğü ilk çalışmadır (2009).

Sayısal harita çıkarma maliyetlerinin en küçüklenmesinde karşılaşılan problem (Vansteenwegen, et.al., 2009) çalışmasında karma ZP-KARP olarak tanımlanmıştır. Gün içerisinde resimlerin ne zaman çekildiği kritik olduğundan probleme zaman penceresi kısıtları eklenmiştir. Tanımlanan problem kapasiteli araç rotalama problemine dönüştürülmüştür. Gün sayısını azaltacak bir yerel arama algoritması ve zaman penceresi aşımını en küçükleyecek bir melez metasezgisel önerilmiştir. Son olarak

aracın sokaklara ideal servis başlangıç zamanının belirlenmesi için doğrusal programlama modeli kullanılmıştır.

#### **4.3.9. Parçalı göndermeli KARP**

Parçalı göndermeli KARP (PG-KARP), klasik KARP'ın herhangi bir ayrıta farklı turlarda servis verilmesine izin verilmesiyle gevşetilmiş şeklidir. Parçalı gönderme (split delivery) ile ilgili çalışmaların büyük kısmı parçalı göndermeli araç rotalama problemi ile ilgilidir. Bu problem ilk kez Dror ve Trudeau tarafından tanımlanmıştır (1989, 1990). Yazarlar, parçalı göndermelere izin verilmesinin toplam maliyette ve kullanılan araç sayısında büyük kazanımlar yarattığını göstermişlerdir.

PG-KARP'ın çözümü için (Mullaseril, et.al., 1997) çalışmasında Dror ve Trudeau tarafından parçalı göndermeli araç rotalama problemi için önerilen sezgisel uyarlanmıştır. (Guéguen, 1999; Labadi, et.al., 2008) çalışmasında zaman pencereli PG-KARP'ın düğüm gezme problemine dönüştürülmesi ve sütun türetme yaklaşımıyla çözülmesi anlatılmaktadır. (Labadi, et.al., 2007) çalışmasında PG-KARP'ın matematiksel modeli verilmiş ve problemin çözümü için bir memetik algoritma önerilmiştir. (Labadi, et.al., 2008)'de ise (Labadi, et.al., 2007) çalışmasında yapılanlar genişletilmiştir.

#### **4.3.10. Dinamik KARP**

Dinamik kapasiteli ayrıt rotalama problemi (D-KARP) ilk kez (Tagmouti, 2009) çalışmasında tanımlanmıştır. Kışın yolların tuzlanması probleminden hareketle D-KARP türetilmiştir. Çalışmada, hava raporlarındaki herhangi bir güncelleme ayrıtlara yapılacak optimal servis başlangıç zamanını değiştirmektedir. Bu nedenle servis maliyeti, servisin başlangıç zamanına bağlı parçalı bir fonksiyonla ifade edilmiştir. Bu da problemi dinamik hale dönüştürmektedir. Problemin statik versiyonu (servis maliyetlerinin daha önceden bilindiği ve değişmediği varsayımı) için azalan değişken

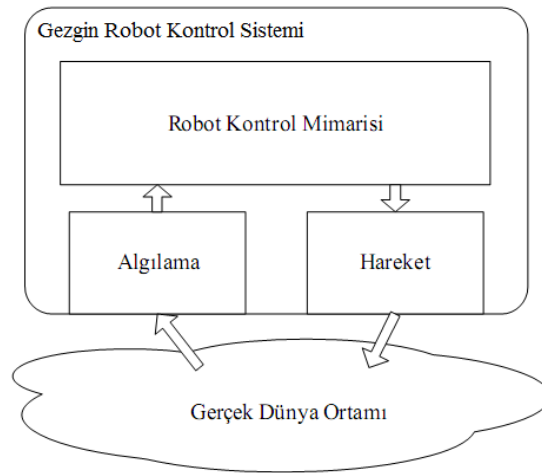
komşuluk arama sezgiseli önerilmiştir. Daha sonra bu sezgisel D-KARP'a uyarlanmıştır.

## 5. YENİ BİR KAPASİTELİ AYRIT ROTALAMA PROBLEMİ

Çalışmanın bu bölümünde klasik KARP'tan farklı olarak gezilmesi gerekmeyen ayrıtlarda da kapasitenin kullanıldığı bir KARP türüne niçin ihtiyaç duyulduğu açıklanmıştır. Ayrıca bu bölümde tanımlanan “Yeni KARP” türü için bir matematiksel model ve çözümü için ise bir sezgisel algoritma önerilmiştir.

### 5.1. Motivasyon (Gezgin Robotlarla Kapsama Problemi)

Gezgin robotlar, fiziksel ortamlarda hareket edebilmeleri ve ortama etki edebilmeleri için hareket sistemleriyle ve içinde buldukları ortamı algılayabilmeleri içinse algılama sistemleriyle donatılan; kendilerinden beklenen görevleri gerçekleştirmek üzere bu sistemlerin koordinasyon ve kontrolünü gerçekleştirecek kontrol mimarisine sahip araçlardır (Özkan, 2007). Şekil 5.1’de bir gezgin robot kontrol sisteminin genel gösterimi verilmiştir. Gezgin robotlar ve gezgin robot sistemlerini oluşturan parçalar için (Arkin, 2002), (Murphy, 2000) ve (Siegwart and Nourbakhsh, 2004) kaynaklarına bakılabilir.



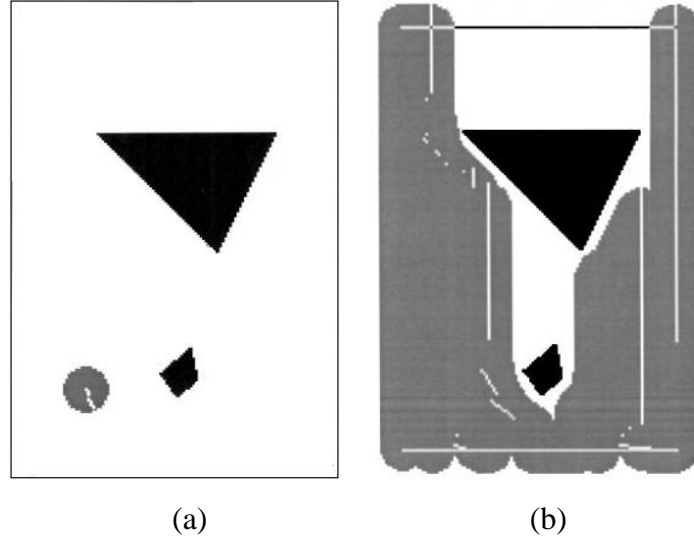
Şekil 5.1. Gezgin robot kontrol sisteminin genel gösterimi (Özkan, 2007).

Gezgin robotlar birçok uygulamada kullanılmaktadır. Bunlar uzay keşifleri, yerlerin temizlenmesi, çim biçme, servis vb. olarak sıralanabilir. Bu uygulamaların içerisinde yerlerin temizlenmesi (Prassler, et.al., 2000), çim biçme (Arkin, et.al., 2000) gezgin robot literatüründe kapsama problemi (Coverage Problem) olarak bilinen problem sınıfına girmektedir. Kapsama problemi, bir ortamdaki her noktanın en az bir sefer taranmasının garanti edilmesi olarak tanımlanır (Choset, 2001). Kapsama problemi için yol planlaması ise her noktanın kapsanmasını garanti edecek yol planının belirlenmesi olarak tanımlanır.

Klasik kapsama problemi için yol planlamasında robot aparatının ortamdaki her noktadan geçecek şekilde bir yolun belirlenmesi gerekir. Örneğin, yerlerin temizlenmesi, çim biçme, kar küreme, boyama, ekin biçme vs. Fakat bazı uygulamalarda robotun ortamdaki bütün noktaları taraması yerine algılayıcısı ile ortamdaki bütün noktaları tarayacak şekilde geçmesi yeterlidir. Bu kapsama türüne algılayıcı tabanlı kapsama denir (Acar, et.al., 2006). Algılayıcı tabanlı kapsama probleminin uygulama alanlarına devriye gezme, arama-kurtarma operasyonları (Trevai, 2007), haritalama vb. örnek olarak verilebilir.

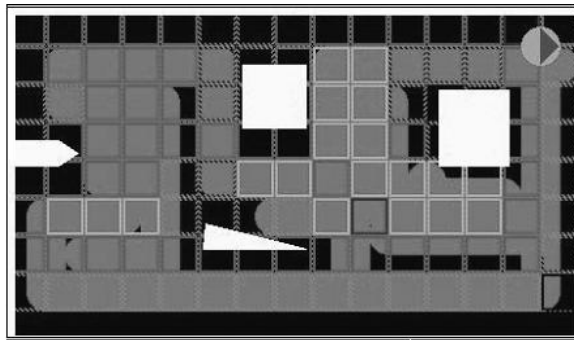
Algılayıcı tabanlı kapsama problemi için literatürde önerilmiş yöntemler bulunmaktadır. (Choset, 2000) çalışmasında ortamdaki boş alanlar üst üste gelmeyecek şekilde hücre (cell) adı verilen parçalara bölünür ve bu hücreler arasında bir komşuluk matrisi tanımlanır. Daha sonra her hücredeki kapsama robotun çapı genişliğinde aralıklarla ileri geri hareket etmesiyle gerçekleştirilir. Şekil 5.2’de hücre ayrıştırma yöntemiyle yapılan kapsama için uygulama örneği verilmiştir. Şeklin birinci kısmında uygulama ortamı verilmiştir. Şekilde siyah poligonlar engelleri, gri daire ise robotu göstermektedir. Robot simgesinin üzerinde bulunan ve yarıçap uzunluğunda olan beyaz çizgi ise robotun yönelimini göstermektedir. Şeklin ikinci kısmında kapsamanın belirli bir zamanında alınan ekran görüntüsü verilmiştir.





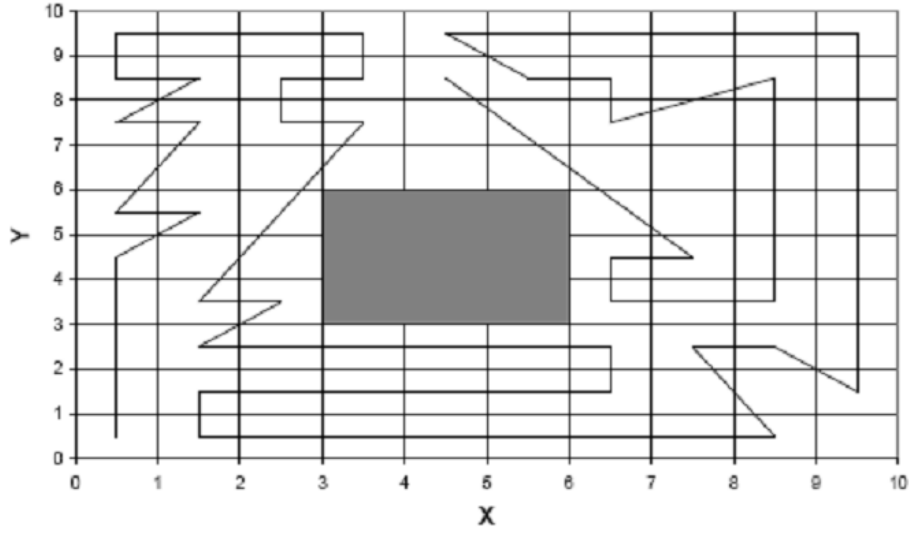
**Şekil 5.2.** a) Sınırları olan bir ortam, b) Ortamın belirli bir kısmı kapsandıktan sonraki görüntüsü (Choset, 2000).

Gonzalez, vd.'nin çalışmasında ortam, ızgara (grid) tabanlı olarak modellenmiştir (2005). Bu çalışmada “zig-zag” şeklindeki yollar yerine BSA (Backtracking Spiral Algorithm) temel alınarak spiral şeklindeki yollar elde edilmiştir. Şekil 5.3'te BSA ile yapılmış bir kapsama örneği verilmiştir. Bu şekilde beyaz poligonlar engelleri, içerisinde bir üçgen bulunan daire robotu, gri alanlar ise robotun geçtiği yolları göstermektedir.



**Şekil 5.3.** BSA ile kapsama örneği (Gonzalez, et.al., 2005)

Ryerson ve Zhang'in çalışmasında otonom tarım ekipmanının bilinen engellerden sakınarak bütün alanı kapsayacak yolun belirlenmesi incelenmiştir (2007). Bu çalışmada ortam robotun araç genişliğinde olan dikdörtgen şekilli hücrelere bölünmüştür. Şekil 5.4'te örnek test ortamı için önerilen yöntem ile elde edilen bir yol planı verilmiştir. Şekildeki gri dikdörtgen engeli, devamlı olan çizgi de yolu göstermektedir.



**Şekil 5.4.** Örnek test ortamı için elde edilen yol planı (Ryerson and Zhang, 2007).

Bu çalışmalarda önerilen yöntemler geniş ortamlarda algılama tabanlı kapsama probleminde uygulanabilir. Fakat, ortamdaki engeller tarafından algılayıcı genişliği sınırlanmış (dar ortam) ise bu metotlar uygulanabilir durumda değildir veya etkin planlar türetemezler (Acar, et.al., 2006). Bu durumda ortam Genelleştirilmiş Voronoi Diyagramı (GVD-Generalized Voronoi Diagram) ile modellenebilir. Voronoi diyagramı, düzlemde iki noktaya olan mesafenin eşit olduğu noktalar kümesi olarak tanımlanır (Çınar vd., 2008). Gerçek ortamdaki engeller basit noktalarla gösterilemeyeceği için, robotikte kullanılan VD tanımını GVD şeklinde genişletilmiştir (Choset, et.al., 2005).

Dar ortamlar GVD kullanılarak modellendikten sonra, elde edilen diyagramın gezilmesi ortamın algılayıcı tabanlı olarak kapsanmasını garantiler. Ortama ait GVD oluşturulduktan sonra bu diyagramın düğüm ve yönsüz ayrıtlardan oluşan serim gösterimi türetilebilir. Bu durumda elde edilen serimdeki bütün ayrıtlara en az bir kez uğrayacak şekilde oluşturulan rota, ortamın kapsanması için yeterlidir. Fakat verimli bir algılayıcı tabanlı kapsama için gereksiz ayrıt (Örnek: Bir ayrıttan iki kez geçmek) geçişleri en küçüklenmelidir. Toplamda en küçük maliyetli olacak şekilde tanımlanan bu problem aslında bir ayrıt rotalama problemidir. Bir serimdeki bütün ayrıtların en küçük maliyetli olacak şekilde en az bir kez kapsanmasını sağlayacak turun belirlenmesi problemi, ayrıt rotalama problemi olarak tanımlanır (Dror, 2000). Bu durumda, GVD-tabanlı olarak oluşturulmuş serim üzerinde çözülecek ayrıt rotalama problemi ortamın kapsanmasını garantiler.

Kapsama problemi uygulamalarında, çok robotun kapsama süresini düşürmesi (bir robot yerine çok robot ile kapsama görevi daha kısa sürede bitirilebilir), hata toleransının olması (robot takımından birisi işlevsiz hale geldiğinden geri kalan robotlar görevi tamamlayabilir) vb. nedenlerle daha çok tercih edilir. Kapsama görevinin çok robotla gerçekleştirilmesi problemine çok robotlu ortam kapsanması problemi (Multi-Robot Coverage Problem) denir. Bu problemle ilgili literatürde bazı çalışmalar bulunmaktadır. (Kurabayashi, et.al., 1996) çalışmasında ortam alanını tanımlamak için ayrıtları, robotların yollarını belirlemek için VD kullanmışlardır. Elde edilecek maliyet değerleri ise robotun aldığı yol cinsinden tanımlanmıştır. İlk olarak bütün yolları gezecek bir tur elde edilir. Daha sonra bu turun uygun parçaları maliyet değerlendirmesine göre robotlara atanır. Fakat bu çalışmada robotun enerji kapasitesi göz önüne alınmamıştır. (Latimer, et.al., 2002) çalışmasında tek robot için önerilen hücre ayrıştırma yaklaşımı çok robot için uyumlandırılmıştır. Bu çalışmada robot takımı homojen olsa da robotlar için keşif ve tarama şeklinde iki farklı rol tanımlanmıştır. Keşif rolündeki bir veya birden fazla robot hücreleri oluşturduktan sonra robot takımı küçük gruplara ayrılır ve uygun bir formasyonda hücreyi tararlar. (Mei, et.al., 2006) çalışmasında gezgin robot dağıtım problemi incelenmiştir. Dağıtım problemi, taşıyıcı tarafından bırakılacak robot grubu sayısı, her gruptaki robot sayısı ve her robotun başlangıç pozisyonunun belirlenmesi problemi olarak tanımlanır. Robotların hem

zamanlama hem de enerji kısıtları göz önüne alınmıştır. Bu çalışmada robotların kapsama rotaları dikdörtgen tarama çizgileri kullanılarak türetilmektedir. Ayrıca tanımlanan dağıtım problemi az sayıda robotla ve geniş ortamlar için çözülmüştür. Dar ortamlarda dağıtım problemi için bir çözüm yöntemi önerilmemiştir. Fakat ortam GVD kullanılarak modellenirse çok robot için dar ortamlarda algılayıcı tabanlı kapsama yapılabilir.

Bu çalışmada, dar ortamlar için çok robotlu algılayıcı tabanlı ortam kapsanması probleminde robot enerji kısıtının da göz önüne alındığı bir çözüm yöntemi önerilmiştir. İlk olarak ortama ait GVD tabanlı bir serim çıkartılır. Bu serimdeki bütün ayrıtları, robotların enerji kısıtı göz önüne alarak kapsayacak robot rotalarının belirlenmesi ortamın kapsanması için yeterlidir. Fakat etkin bir kapsama için robotlar tarafından alınan toplam yol uzunluğunun en küçüklenmesi istenir. Bu problem yönsüz kapasiteli ayrıt rotalama problemidir. KARP'ta tanımlanan ayrıt maliyeti, çok robotlu algılayıcı tabanlı ortam kapsanması probleminde GVD tabanlı oluşturulan serimdeki ayrıt uzunluğuna, aracın ayrıtlardan topladığı talep değeri, robotun kapsama görevini gerçekleştirirken harcadığı enerjiye (örneğin izleme görevi için robotun kamerasının harcadığı enerji) karşı gelmektedir. Farklı olarak çok robotlu ortam kapsanması probleminde, robot kapsama görevini gerçekleştirirken harcadığı enerjinin yanı sıra motoru, algılayıcıları, üzerinde bulunan bilgisayarı vs. çalıştırmak için de enerji harcamaktadır ve bu enerji göz ardı edilemez. Bu harcanan enerji servis verilen ayrıtlarda harcanan enerjiye dâhil edilebilir. Fakat robot servis verilmiş bir ayrıttan geçtiğinde de bu enerjiyi harcamaya devam etmektedir. Bu nedenle servis verilmeyen ayrıtlardan geçerken de bir enerji harcanmaktadır. Bu çalışmada servis verilirken harcanan enerji kapsama enerjisi ( $q_{ij}$ ), ayrıtlardan geçerken harcadığı enerji de geçiş enerjisi ( $e_{ij}$ ) olarak adlandırılmıştır. Robotun servis vermeden de geçtiği ayrıtlar nedeniyle bu çalışmada tanımlanan KARP, klasik KARP'tan farklıdır. Bu yüzden bu problem Yeni KARP olarak adlandırılmıştır. Klasik KARP ve çok robotlu ortam kapsanması probleminin (Yeni KARP) benzer ve farklı yönleri detaylı olarak Çizelge 5.1'de verilmiştir.

**Çizelge 5.1.** Kapasiteli ayrıt rotalama problemi ve çok robotlu ortam kapsanması probleminin benzer ve farklı yönleri

	Çok Robotlu Algılayıcı Tabanlı Ortam Kapsanması Problemi (Yeni KARP)	Kapasiteli Ayrıt Rotalama Problemi (KARP)
<b>Benzer Yönler</b>		
	GVD tabanlı olarak elde edilen yönsüz serim.	$G = (V, E)$ serimi
	Robot enerji kapasitesi	Araç kapasitesi ( $Q$ )
	Robot takımı	Araç filosu
	Şarj noktası veya robotların belirlenmiş bir toplanma noktası	Depo düğümü
	Robot takımının ortamın sadece belirli bir bölgesini kapsaması	Gezilmesi gereken ayrıtlar kümesi ( $R$ )
	GVD tabanlı serimdeki ayrıtların uzunluğu	Ayrıt maliyeti ( $c_{ij}$ )
	Robotun GVD tabanlı serim üzerindeki bir ayrıta servis verirken harcadığı kapsama enerjisi	Ayrıtların talebi ( $q_{ij}$ )
	GVD-tabanlı serimin en küçük maliyetle kapsanması.	Toplam tur maliyetinin en küçüklenmesi.
<b>Farklı Yön</b>		
	Servis verilmeden geçilen bir ayrıtta robotun enerji harcaması ( $e_{ij}$ )	Karşılığı yok.

İzleyen alt bölümde tanımlanan Yeni KARP için bir matematiksel model önerilmiştir. Problemin çözümü için ise KARP sezgiselleri içerisinde anlatılan

Ulusoy'un sezgiseli adapte edilmiş ve kullanılmıştır. Son olarak da Yeni KARP çözümü için bir genetik algoritma önerilmiştir.

## 5.2. Yeni KARP İçin Bir Matematiksel Modeli

GVD tabanlı olarak elde edilen serim, yönsüz bir serimdir. Bu nedenle önerilen matematiksel model ve çözüm yöntemleri yönsüz bir serim için geliştirilmiştir. Yönsüz KARP'ın en iyi çözümünde aynı ayrıta, aynı yönde birden fazla sefer uğranmaz (Lacomme, et.al., 2001). Bu yüzden araç turunu belirleyecek karar değişkenleri için tamsayı yerine 0-1 tamsayı kısıtı konulmuştur. Yeni KARP'ın matematiksel modeli Model 5.1'de verilmiştir.

### Model 5.1. Yeni KARP matematiksel modeli

#### Kümeler

$V$	Düğüm kümesi, $\{1, 2, \dots, n\}$
$E$	Ayrıtlar kümesi
$R$	Kapsanması istenen ayrıtlar kümesi
$V'$	Depo düğümü hariç bütün düğümler kümesi $V \setminus \{1\}$
$S$	$V'$ kümesinin olası bütün alt kümeleri

#### İndisler

$i, j$	Düğüm indisleri
$k$	Robot (araç) indisi

#### Parametreler

$n$	Serimdeki düğüm sayısı
$m$	Robot (araç) sayısı
$c_{ij}$	$(i, j)$ ayrıtlarının uzunluğu $\forall (i, j) \in E$
$q_{ij}$	$(i, j)$ ayrıtlarına servis verirken harcanacak enerji $\forall (i, j) \in R$
$e_{ij}$	$(i, j)$ ayrıtlarından geçişinde harcanacak enerji $\forall (i, j) \in E$
$Q$	Robotun (aracın) enerji kapasitesi

$M$  Büyük sabit bir sayı

### Karar Değişkenleri

$$y_{ijk} = \begin{cases} 1 & (i,j) \in R \text{ ayrıtına } k \text{ turunda servis veriliyor} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & i' \text{den } j' \text{ye } k \text{ turunda geçiliyor, } \forall (i,j) \in E \\ 0 & \text{Diğer durumlarda} \end{cases}$$

### Model

$$\sum_{p \in V} x_{pik} - \sum_{p \in V} x_{ipk} = 0 \quad \forall i \in V, \quad k = 1, 2, \dots, m \quad (5.1)$$

$$\sum_{k=1}^m y_{ijk} = 1 \quad \forall (i, j) \in R, \quad R \subseteq E \quad (5.2)$$

$$\sum_{(i,j) \in E} e_{ij} \cdot x_{ijk} + \sum_{(i,j) \in R} q_{ij} \cdot y_{ijk} \leq Q \quad k = 1, 2, \dots, m \quad (5.3)$$

$$x_{ijk} + x_{jik} \geq y_{ijk} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (5.4)$$

$$M \sum_{i \notin S, j \in S} x_{ijk} \geq \sum_{(j,p) \in S} x_{jpk} \left\{ \begin{array}{l} V' = \{2, 3, \dots, n\} \\ \forall S \subseteq V', S \neq \emptyset \\ k = 1, 2, \dots, m \end{array} \right. \quad (5.5)$$

$$y_{ijk} \in \{0, 1\} \quad \forall (i, j) \in R, \quad k = 1, 2, \dots, m \quad (5.6)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in E, \quad k = 1, 2, \dots, m \quad (5.7)$$

kısıtlar altında

$$enk \left( \sum_{(i,j) \in E} \sum_{k=1}^m c_{ij} \cdot x_{ijk} \right) \quad (5.8)$$

(5.1) numaralı kısıt grubu tur devamlılığını sağlar. (5.2) numaralı kısıt serimdeki kapsanması gereken ayrıtlara ( $\forall (i, j) \in R$ ) bir robot tarafından bir kez servis verilmesini sağlar. Modelde parçalı gönderime izin verilmemektedir. (5.3) numaralı kısıt grubunun sol tarafı robotun servis sırasında ve gezerken harcadığı enerjinin toplamıdır. Bu kısıt grubu harcanan toplam enerjinin robotun enerji kapasitesini aşmamasını garantiler. (5.4) numaralı kısıt grubu, bir robotun serimdeki herhangi bir ( $i, j$ ) ayrıtına servis veriyorsa ( $i, j$ ) veya ( $j, i$ ) yönünde bu ayrıttan geçmesini

garantiler (İlişkilendirme kısıtı). (5.5) numaralı kısıt grubu kural dışı alt turların oluşmasını engeller. (5.6) ve (5.7) numaralı denklemler karar değişkenlerinin 0-1 ve tamsayı olduğunu gösterir. (5.8) modelin amaç fonksiyonunu gösterir ve robotların kapsama sırasında kat ettikleri toplam yolun en küçüklenmesini sağlar.

Yeni KARP için önerilen matematiksel modelin KARP'ın matematiksel modelinden en temel farkı (5.3) numaralı kapasite kısıtıdır. Aracın toplam kapasite değeri, KARP'ta sadece gezilmesi gereken ayrıtlarda tanımlıyken, Yeni KARP'ta (5.3) numaralı kısıtından da görüldüğü gibi hem gezilmesi gereken ayrıtlarda hem de geçişlerde tanımlanır.

Modeldeki (5.1) numaralı tur devamlılığı kısıtı her düğüm ve araç için yazıldığından modele  $n \cdot k$  adet kısıt eklentisine neden olmaktadır. (5.2) numaralı her gezilmesi gereken ayrıta servis verilmeli kısıtı bütün gezilmesi gereken ayrıtlar için yazıldığından bu kısıt grubu toplamda gezilmesi gereken ayrıt sayısı  $(|R|)$  kadar kısıt içermektedir. (5.3) numaralı kapasite kısıtı bütün araçlar için yazıldığından modele  $k$  adet kısıt eklentisine neden olur. (5.4) numaralı ilişkilendirme kısıtı ise modele  $|R| \cdot k$  adet kısıt ekler. Modelde  $|E| \cdot k$  adet  $x$  karar değişkeni,  $|R| \cdot k$  adet  $y$  karar değişkeni bulunmaktadır. Son olarak (5.5) numaralı kural dışı alt turları engelleme kısıtı modele depo düğümü hariç bütün düğüm alt kümeleri için yazılacağından modele  $(m(2^{n-1} - 1))$  adet kısıt eklentisine neden olur. Bu kısıt grubu (Dror and Leung, 1998: Dror and Langevin, 2000) çalışmasında önerilmiş kısıt grubundan farklıdır. Dror ve Leung tarafından önerilen kısıt grubu serimdeki gezilmesi gereken ayrıtların bütün alt kümeleri için yazılmaktadır. Fakat bu modelde ise aynı kısıt grubu bütün düğüm alt kümeleri için yazılmıştır. Bir serimdeki ayrıt sayısı için  $n$  düğüm sayısı olmak üzere  $|E| \approx 4n$  kabulü yapılmaktadır. Bir serimdeki ayrıtların yarısının gezilmesi gerektiği varsayılırsa bu çalışmadaki kural dışı alt tur engelleme kısıtı modele daha az sayıda kısıt eklenmesini sağlayabilir. Fakat her iki grup kısıtta üstel sayıda kısıt eklentisine neden olduğu için büyük serimler için model ile çözüm bulunması hesapsal olarak uygun değildir. Bu nedenle Yeni KARP'ın çözümü için bir sezgisel ve bir genetik algoritma önerilmiştir.



### 5.3. Yeni KARP İçin Önerilen Değiştirilmiş Ulusoy Algoritması

Literatürde KARP'ın çözümü için birçok sezgisel algoritma önerilmiştir (Bkz. Bölüm 4.3.2). Ancak bu algoritmalar sadece gezilmesi gereken ayrıtlarda aracın kapasiteni harcadığını göz önüne aldıklarından Yeni KARP'ta direkt olarak kullanılamaz. Ayrıca, bu sezgisel algoritmalar yapıcı sezgisel sınıfındakiler, araç serimin herhangi bir düğümündeyken kapasitesi aşırsa bu noktadan sonra hiçbir ayrıta servis vermeden en kısa yol kullanılarak depo düğümüne bağlar ve turu oluşturur. Fakat bu tipte olan bir sezgisel Yeni KARP'ta kullanılamaz. Çünkü aracın (veya robotun) böyle bir durum ortaya çıktığında depo düğümüne dönecek kadar enerjisi olmalıdır. Depo düğümüne varmadan önce herhangi bir düğümde enerjisinin bittiği bir çözüm Yeni KARP için uygun bir çözüm değildir. Bu nedenlerden dolayı, literatürde de çok kullanılmış olan Ulusoy'un sezgiseli Yeni KARP için uyarlanarak kullanılmıştır. Ulusoy'un sezgiseli Bölüm 4.3.2'de önce rotala sonra bölümlendir grubu sezgisellerin içerisinde detaylı olarak anlatılmıştır.

Ulusoy'un bölümlendirme algoritmasında ilk olarak bütün gezilmesi gereken ayrıtları kapsayan bir tur (dev tur) oluşturulur ve bu tur uygun araç turlarına bölümlendirilerek bir yönlü serim oluşturulur. Yönlü serimdeki her bir ayrıt asıl serimdeki bir tura karşılık gelir. Son olarak yönlü serim üzerinde en kısa yol problemi çözülür ve hangi turların kullanılması gerektiği belirlenmiş olur. Ulusoy'un algoritmasında dev turdan alınan parçanın ilk olarak araç kapasitesini aşıp aşmadığı kontrol edilir ve eğer aşmıyorsa yönlü serime eklenir. Çünkü burada sadece hangi ayrıtlara servis verildiği önemlidir. Fakat Yeni KARP'ta hem geçiş hem de servis için kapasite kullanıldığı için tur parçasının tam bir tur olup olmadığı önem kazanır. Tur değilse tur parçası önce tam bir tura dönüştürülür sonrasında turun kapasitesi kontrol edilir. Eğer araç kapasitesi aşılmıyorsa en kısa yol probleminin çözüleceği yönlü serime eklenir. Ulusoy'un bölümlendirme algoritmasının Yeni KARP için uyumlandırılmış hali bu çalışmada Değiştirilmiş Ulusoy Algoritması (DUA) olarak adlandırılmıştır.

Değiştirilmiş Ulusoy algoritması bütün gezilmesi gereken ayrıtları kapsayan Euler turunu parçalar. Her parçayı depo düğümünde başlayıp tamamlanacak şekilde bir

çevrime dönüştürür. Daha sonra bu çevrim üzerindeki toplam enerji talebini, servis verilmeyen ayrıtlarda enerji harcandığını da göz önünde bulundurarak hesaplar. Eğer bu enerji talebi aracın (veya robotun) enerji kapasitesini aşmıyorsa tanımlanan yönlü serime bu çevrim bir ayrıt olarak eklenir. Daha sonra başlangıç düğümünden bitiş düğümüne en kısa yol belirlenir. En kısa yol üzerinde kullanılan her ayrıt bir çevrimi göstermektedir. Elde edilen çevrimler, türetilmiş çevrimler içerisinde bütün gezilmesi gereken ayrıtları kapsayan ve en küçük maliyetli olanlardır.

DUA çalıştırılmadan önce ilk olarak ortamın GVD tabanlı serimi çıkartılır. Bu serim yönsüz bir serimdir ve DUA'ya girdi olarak verilir. DUA'nın ikinci girdisi gezilmesi gereken ayrıtları kapsayan  $T$  turudur. Bu tur herhangi bir yapıcı yöntem, yerel arama ve sezgisel kullanılarak oluşturulabilir. Bu çalışmada dev Euler turu Frederickson'nın yönsüz KPP'nin çözümü için önermiş olduğu sezgisel algoritma kullanılarak oluşturulmaktadır. Serimdeki bütün ayrıtların kapsanması istenirse bu durumda problem yönsüz ÇPP'ye indirgenir. Fakat bu durum da KPP çözüm yöntemi kullanılabilir. DUA'nın detayları Algoritma 5.1'de verilmiştir.

**Algoritma 5.1.** *Yeni KARP için Değiştirilmiş Ulusoy Algoritması*

Girdi: GVD tabanlı olarak oluşturulmuş yönsüz  $G = (V, E)$  serimi ve gezilmesi gereken ayrıtlar kümesi olan  $R$ 'yi kapsayan  $T$  turu.

Çıktı:  $G$ 'de tanımlı  $R$ 'yi kapsayan uygun araç turları.

*Adım 0:* Eğer  $T$  turunda gezilmesi gereken bir ayrıt birden fazla sefer ziyaret ediliyorsa, bu ayrıta ilk geçişte servis ver.

*Adım 1:*  $G$ 'deki düğümleri  $T$  turuna göre yeniden etiketle.  $v_0$  düğümü depo düğüm olmak üzere,  $T$  turu  $(v_0, v_1, \dots, v_r = v_0)$ 'a eşittir.  $r$ ,  $T$  turunda en son servis verilen ayrıta komşu olan düğümün indeksi olsun. Düğüm kümesi  $V' = \{v_0, v_1, \dots, v_r\}$  olan  $G' = (V', A)$  yönlü serimini oluştur (Başlangıçta;  $A = \{\}$ ).  $G'$  seriminin ayrıt kümesine  $(A)$   $a, b = \{1, 2, 3, \dots, r\}$  için  $b > a$  olmak koşuluyla  $(a, b)$  ayrıtını ekle. Eğer  $b > a + 1$  ve  $(v_a, v_{a+1})$  veya  $(v_{b-1}, v_b)$   $T$  turu üzerinde servis verilen bir ayrıt değilse

$G'$  seriminden  $(a,b)$  ayrıtını çıkar.  $G$  seriminde  $v_a$  ve  $v_b$  düğümleri arasında tanımlı olan  $C_{ab}$  çevrimi, çevrimin maliyeti ( $c'_{ab}$ ) ve çevrimde harcanan toplam enerjisi aşağıdaki durumlara göre hesapla. Hesaplanacak olan  $c'_{ab}$  değeri  $G'$  serimindeki  $(a,b)$  ayrıtının maliyetine karşı gelmektedir.

- Eğer  $b = a + 1$  ve  $(v_a, v_{a+1})$   $T$  üzerinde servis verilen bir ayrıt değilse,  $c'_{ab} = 0$ .
- Eğer  $b > a + 1$  veya  $(v_a, v_{a+1})$   $T$  üzerinde servis verilen bir ayrıt ise  $(v_a, \dots, v_b)$  aralığındaki tur parçasını gösteren  $P_{ab}$  zinciri için aşağıdaki durumları kontrol et;
  - $P_{ab}$  zinciri depo düğümünü içeriyorsa,  $P_{ab} = (v_a, \dots, v_0, \dots, v_b)$  zincirine  $v_b$  ve  $v_a$  düğümleri arasındaki en kısa yolu gösteren  $SP_{ab}$  en kısa yolunu ekle. Bu durumda  $C_{ab} = (v_0, \dots, v_a, SP_{ab}, v_b, \dots, v_0)$  çevrimi elde edilir.
  - $P_{ab}$  zinciri depo düğümünü içermiyorsa,  $P_{ab}$  zincirine  $SP_{v_0 a}$  ve  $SP_{b v_0}$  en kısa yollarını ekle. Bu durumda  $C_{ab} = (SP_{v_0 a}, v_a, \dots, v_b, SP_{b v_0})$  çevrimi elde edilir.

$C_{ab}$  çevrimi üzerindeki toplam enerji talebini servis verilen ayrıtlar için  $q_{ij}$ , ayrıtlardan geçmek için ise  $e_{ij}$  değerini kullanarak hesapla.

- Eğer  $C_{ab}$  çevriminin toplam enerji değeri kapasiteyi ( $Q$ ) aşmıyorsa,  $G'$ 'de tanımlanmış  $C_{ab}$  çevrimin uzunluğu ile  $c'_{ab}$ 'nin değeri hesaplanır.
- Diğer durumda  $G'$  seriminden  $(a,b)$  ayrıtını çıkar

*Adım 2:*  $G'$  seriminde  $0$ 'dan  $r$ 'ye en kısa yolu bul. En kısa yol üzerinde kullanılan her  $(a,b)$  ayrıtı  $G$  üzerinde farklı bir araç tarafından gerçekleştirilecek uygun bir araç turunu gösterir.

#### 5.4. Yeni KARP İçin Geliştirilen Genetik Algoritma

Yeni KARP'ta tanımlanmış olan gezilmesi gerekmeyen ayrıtlarda aracın kapasite kullanması göz ardı edilirse problem klasik KARP'a indirgenir. KARP  $NP$ -zor bir problem olduğundan Yeni KARP da  $NP$ -zor bir problemdir. Bu durum büyük boyutlu problemler için Yeni KARP'ın çözümünün anlamlı bir sürede elde edilmesini imkânsız kılmaktadır. Bu nedenle, bu çalışmada Yeni KARP'ın çözümü için literatüre kombinatorik eniyileme problemlerinde başarıyla uygulanmış olan genetik algoritma kullanılmıştır. Çalışmanın bu alt bölümünde Yeni KARP için geliştirilmiş bir genetik algoritma anlatılmıştır.

Genetik algoritmalar (GA), doğal seçim ve doğal genetik mekanikleri temel alan güçlü ve yaygın olarak kullanılan bir stokastik arama yöntemidir (Goldberg, 1989). GA ilk kez 1975 yılında Holland tarafından önerilmiştir (Holland, 1975). Genellikle, bilinen eniyileme yöntemleri ile çözülemeyen ya da çözüm zamanı problemin büyüklüğü ile üstel artan problemlerde kullanılan GA, en iyi ya da en iyiye yakın çözüm vermektedir.

GA, geleneksel arama tekniklerinden farklı olarak *popülasyon* (population) adı verilen rassal başlangıç çözüm seti ile başlar. Popülasyondaki her birey *kromozom* (chromosome) olarak adlandırılır ve problem için bir çözümü gösterir. Kromozomun parçası *gen* (gene), genin kromozomdaki pozisyonu *lokus* (locus) ve genin kromozomdaki değeri *alel* (allele) olarak adlandırılır. Kromozomlar *nesil* (generation) adı verilen başarılı ardıştırmalar sonucu evrim geçirir. Her nesilde kromozomlar, *uyumun* (fitness) bir ölçüsü olarak *değerlendirilir* (evaluate). Yeni neslin yaratılması için *çocuk* (offspring) olarak adlandırılan yeni kromozomlar, ya *çaprazlama* (crossover) operatörü kullanılarak şimdiki nesilde yer alan iki kromozomun birleştirilmesiyle ya da *mutasyon* (mutation) operatörü kullanılarak kromozomun değiştirilmesiyle elde edilir. Yeni nesil, uyum değerlerine göre bazı ebeveyn ve çocuklar seçilerek ve popülasyon büyüklüğünü sabit tutmak için diğerleri reddedilerek şekillendirilir. Uyum değeri yüksek olan kromozomların seçilme olasılıkları daha büyüktür. Birkaç nesilden sonra algoritma en iyi kromozoma yakınsar. Bu kromozom problemin en iyi çözümünü ya da alt en iyi çözümünü temsil eder.  $P(t)$  ve  $C(t)$  sırasıyla,  $t$  neslindeki ebeveyn ve çocuk

kromozomlarını göstermek koşuluyla genetik algoritmanın genel yapısı aşağıda verilmiştir (Gen and Chen, 1997).

*Adım 1:*  $t = 0$ ,  $P(t)$ 'nin ilk değerlerini ata,  $P(t)$ 'deki tüm bireylerin uyum değerlerini hesapla.

*Adım 2:*  $C(t)$ 'yi elde etmek için  $P(t)$ 'ye çaprazlama ve mutasyon operatörlerini uygula.

*Adım 3:*  $C(t)$ 'deki bireylerin uyum değerlerini hesapla.

*Adım 4:*  $P(t+1)$ 'i  $P(t)$  ve  $C(t)$ 'den seç.

*Adım 5:*  $t = t + 1$

*Adım 6:* Durma kriteri sağlanıyorsa dur, sağlanmıyorsa Adım 2'ye git.

Yeni KARP'ın çözümü için önerilen genetik algoritmadaki kromozomun kodlanması, değerlendirilmesi (uyum değerinin hesaplanması) ve genetik operatörler (çaprazlama ve mutasyon operatörleri) aşağıda anlatılmıştır.

#### 5.4.1. Kodlama

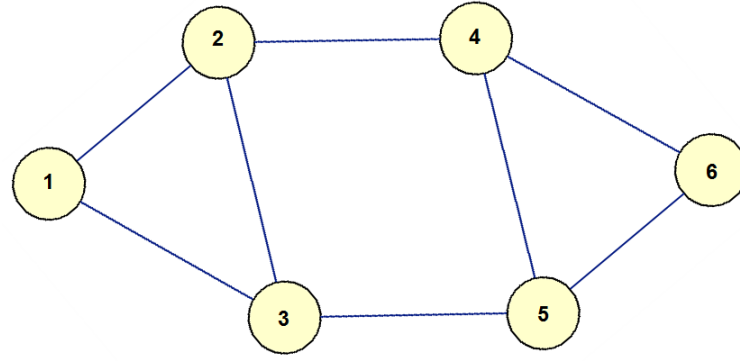
KARP'ın çözümü için önerilen genetik algoritmanın kromozom gösteriminde permütasyon tipi kodlama kullanılmıştır.  $G = (V, E)$  serimindeki ayrıtlar, gezilmesi gereken ayrıtlar ( $R$ ) ve gezilmesi gerekmeyen ayrıtlar ( $E \setminus R$ ) olarak ikiye ayrılabilir.  $|R|$  gezilmesi gereken ayrıtlar kümesinin eleman sayısı olmak üzere, kromozomun uzunluğu  $|R|$ 'ye eşittir. Serimdeki gezilmesi gereken ayrıtlara hepsi birbirinden farklı olacak şekilde numara verilir. Ayrıca kromozomda araçlara atanan ayrıtları belirlemek için sabit araçlar kullanılmıştır.  $m$  eldeki araç ( $k = 1, 2, \dots, m$ ) sayısı olmak üzere kromozomda kullanılan araç sayısı  $m - 1$ 'dir.

Yeni KARP'ta tanımlanan araç filosu klasik KARP'ta olduğu gibi homojendir. Bu nedenle araçların servis verdiği ayrıt sayısı mümkün olduğu kadar birbirine yakın

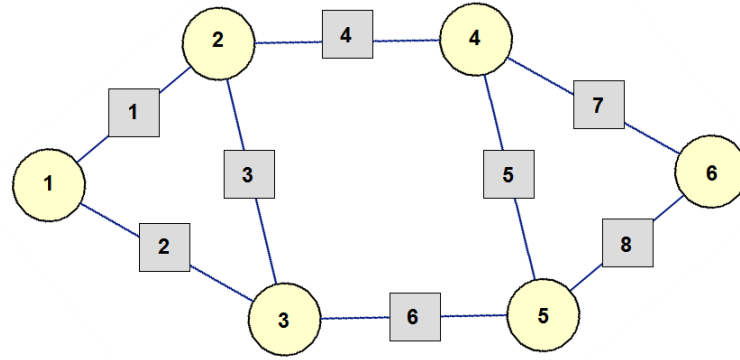
olmalıdır. Araçların servis vereceği ayırıt sayısını belirlemek için ilk olarak bütün araçlara  $\lfloor R/m \rfloor$  adet ayırıt ayrılır. Daha sonrasında geriye kalan  $(|R| \bmod m)$  adet ayırıtlik yer her araca en fazla bir ayırıt gelecek şekilde rassal olarak ayrılır. Son olarak araçlara ayrılan ayırıt sayılarından hareketle ayraç indeksleri belirlenir.

Bütün araçlara servis vermesi gereken ayırıtlar belirlendikten sonra her aracın servis vereceği ayırıtı kapsayan tur Frederickson'ın sezgiseli (Algoritma 3.6) kullanılarak belirlenmiştir. Bu algoritma, gezilmesi gereken ayırıtlar bağlı olduğunda en iyi çözümü, bağlı değilse de en iyi çözüme  $3/2$  yakınlıkta bir çözümü bulmaktadır.

Kromozomun gösterimini ve araç turlarının oluşturulmasını daha detaylı olarak açıklamak için Şekil 5.5-a'da örnek bir serim verilmiştir. Bu serimdeki bütün ayırıtlar gezilmesi gereken ayırıtlar listesindedir ( $R = E$ ). Bu nedenle bütün ayırıtlar Şekil 5.5-b'deki gibi numaralandırılmıştır. Şekil 5.5'te daire içerisindeki numaralar düğüm numaralarını, kare içerisindeki numaralar ayırıt numaralarını göstermektedir. "1" numaralı düğüm depo düğümünü göstermektedir.



(a)



(b)

**Şekil 5.5.** a) Örnek serim, b) Örnek serimin ayrıtlarının numaralarının gösterimi

Şekil 5.5'te verilen serim örneği için tanımlanacak herhangi bir kromozomun araç turlarına dönüştürülmesinde kullanılacak, ayrıtlara karşı gelen bitiş düğümleri listesi Çizelge 5.2'de verilmiştir.

**Çizelge 5.2.** Örnek serimin ayırıt numaraları, bu ayırta karşı gelen bitiş düğümleri ve ayırıt maliyetleri

Ayırıt Numarası	1. Bitiş Düğümü	2. Bitiş Düğümü	Maliyeti
1	1	2	2
2	1	3	3
3	2	3	4
4	2	4	3
5	4	5	5
6	3	5	4
7	4	6	6
8	5	6	4

Örnek serim üzerinde tanımlı ve depo düğümünde 2 adet aracın bulunduğu bir kromozom örneği aşağıda verilmiştir. Verilen kromozomda gezilmesi gereken ayırıt sayısı  $|R|=8$  ve araç sayısı  $m=2$ 'dir. Bu bilgilerden hareketle kromozomdaki tek aracın  $(m-1=1)$  indeksi 4 olarak belirlenir.

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

Bu kromozomda 1. araca atanan ayırıtlar 1, 6, 7 ve 8; 2. araca atanan ayırıtlar 2, 3, 4 ve 5'tir. 1. aracın servis vereceği ayırıtlar listesi (1,2), (3,5), (4,6) ve (5,6)'dır. Algoritma 3.6 ile bu ayırıtı kapsayan tur 1-2-4-6-5-3-1 şeklinde elde edilmiştir. Benzer şekilde 2. aracın servis vereceği ayırıt listesi (1,3), (2,3), (2,4), (4,5) ve turu ise 1-3-5-4-2-3-1 olarak elde edilmiştir.

Önerilen genetik algoritmada başlangıç popülasyonu rastsal olarak türetilmiştir. Her ayırıtın kromozomda  $i$ . sırada bulunmasının olasılığı eşittir. Başlangıç popülasyonu türetildikten sonra bu popülasyon yukarıda anlatılan prosedür ile çözümlenmektedir.



#### 5.4.2. Uyum deęerinin hesaplanması

Kromozomların çözümlenmesinden sonra yapılması gereken çözümlenen kromozomların deęerlendirilmesidir. Deęerlendirme işlemi popülasyondaki bireylerin uyum deęerinin hesaplanması olarak da adlandırılır. Bu işlem kısıtsız eniyileme problemleri için tanımlanan amaç fonksiyonuna göre kromozomun deęerlendirilmesidir. Fakat Yeni KARP'ta kural dıőı alt turların engellenmesi ve kapasite kısıtı olmak üzere iki önemli kısıt grubu bulunmaktadır. Önerilen kromozom kodlaması ve çözümlene stratejisi kural dıőı alt turların oluşmasını kromozomun çözümlenmesi sırasında engeller. Fakat kapasite kısıtı çözümlene işleminden sonra uygun olmayan çözümlerin oluşmasına neden olabilir. Bu problemi çözmek için genetik algoritma literatüründe dört farklı yöntem önerilmiştir. Bunlar aőađıdaki gibi sıralanabilir:

- Reddetme stratejisi: Evrim sürecinden uygun olmayan (kısıtları saęlamayan) bireylerin çıkartılması.
- Onarma stratejisi: Bir tamir prosedürü ile uygun olmayan çözümlerin uygun hale getirilmesi. Bu yöntem uygun çözüm alanı dıőındaki bir çözümün çözüm alanı içerisine girecek şekilde düzenlenmesi olarak da tanımlanabilir.
- Genetik operatörlerin deęiőtirilmesi stratejisi: Kromozomun uygunluęunun garanti edilmesi için probleme uygun genetik operatörlerin tasarlanmasıdır.
- Ceza stratejisi: Uygun olmayan çözümlerin tanımlanan oranda cezalandırılmasıdır. Özellikle uygun çözüm bulunmasının zor olduęu problemlerde kullanılır.

Bu çalışmada kısıtların ele alınabilmesi için ceza stratejisi uygulanmıştır. Bu strateji, düzeltme veya ekstra işlem sırasındaki hesaplama zamanından kaçınmak ve uygun olmayan fakat iyi çözümlerin kaybedilmesini önlemek amacıyla kullanılmıştır. Bu çalışmada kullanılan ceza fonksiyonu  $k$ . aracın kapasite aőım miktarı olarak tanımlanmıştır.

Kromozomdaki  $k$ . aracın kapasite aşımı varsa bunu belirlemek için ilk olarak  $k$ . araca yüklenen toplam kapasite hesaplanmalıdır. Toplam kapasite, servis sırasındaki ve gezme sırasındaki kapasitenin toplamıdır.  $k$ . aracın toplam kapasitesinin hesaplanmasıyla ilgili olarak aşağıdaki tanımlamalar yapılmıştır.

- $d_{v_i, v_j}$   $(v_i, v_j)$  ayrıtının maliyeti
- $e_{v_i, v_j}$   $v_i$  düğümünden  $v_j$  düğümüne geçiş sırasında harcanan kapasite
- $q_{v_i, v_j}$   $(v_i, v_j)$  ayrıtında servis için kullanılan kapasite miktarı
- $F$  Kromozomun uyum değeri
- $U_k$   $k$ . aracın tur uzunluğu
- $E_k$   $k$ . aracın toplam talebi
- $T_k$   $k$ . aracın turu
- $l_k$   $k$ . araç turundaki toplam düğüm sayısı
- $T_{k,i}$   $k$ . araç turundaki  $i$  numaralı düğüm,  $i \in \{1, 2, \dots, l_k\}$
- $R_k$   $k$ . aracın servis vereceği ayrıt listesi
- $|R_k|$   $k$ . aracın servis vereceği toplam ayrıt sayısı
- $R_{k,j}$   $k$ . aracın servis vereceği  $j$  numaralı ayrıt.  $i \in \{1, 2, \dots, l_k\}$
- $R_{(k,j)-1}$   $k$ . aracın servis vereceği  $j$  numaralı ayrıtın 1. bitiş düğümü
- $R_{(k,j)-2}$   $k$ . aracın servis vereceği  $j$  numaralı ayrıtın 2. bitiş düğümü.

Bu tanımlamalardan sonra  $k$ . aracın toplam kapasitesi olan  $E_k$  (5.9) numaralı denklemde verildiği gibi hesaplanır. Denklemin birinci toplam kısmı gezme sırasında kullanılan kapasiteyi, ikinci toplam ise ayrıtlara servis sırasında kullanılan kapasiteyi göstermektedir.

$$E_k = \sum_{i=1}^{l_k-1} e_{T_{k,i}, T_{k,(i+1)}} + \sum_{j=1}^{|R_k|} q_{R_{(k,j)-1}, R_{(k,j)-2}} \quad (5.9)$$

Aracın toplam kapasite miktarı hesaplandıktan sonra kromozomun uyum değerine uygulanacak ceza değeri belirlenebilir. Bu çalışmada ceza fonksiyonu denklem (5.10)'da verildiği gibi tanımlanmıştır. Denklem aracın kapasite aşım miktarı oranında ceza uygulamaktadır.

$$p_k = \begin{cases} \frac{E_k}{Q} & \text{Eger } E_k > Q \text{ ise} \\ 1 & \text{Diğer Durumlarda} \end{cases} \quad (5.10)$$

Yeni KARP'ın amaç fonksiyonu da KARP'ta olduğu gibi toplam tur uzunluğunun en küçüklenmesi şeklinde tanımlanmıştır. Kromozom çözümlendikten sonra her aracın tur uzunluğu denklem (5.11)'de verildiği gibi hesaplanır.

$$U_k = \sum_{i=1}^{l_k-1} d_{T_{k,i}, T_{k,(i+1)}} \quad (5.11)$$

Kromozomdaki araçların tur uzunlukları belirlendikten sonra denklem (5.9) ile aracın toplam kapasitesi belirlenir ve denklem (5.10) ile araç kapasitesine göre  $p_k$  değeri tespit edilir. Kromozomun uyum değeri ( $F$ ) ise denklem (5.12)'de verildiği gibi her aracın cezalandırılmamış veya cezalandırılmış tur uzunluğunun toplamıdır.

$$F = \sum_{k=1}^m U_k \cdot p_k \quad (5.12)$$

Çözömlenen bir kromozomun uyum değerinin hesaplanması adımları Algoritma 5.2'de verilmiştir.

**Algoritma 5.2.** *Çözümlenen bir kromozomun cezalandırılmış toplam tur uzunluğunun hesaplanması*

Girdi: Kromozomun çözümlenmesi sonucu elde edilen araç turları, serimin uzaklık matrisi, geçişler sırasında harcanan kapasite değerleri ve servis sırasında kullanılan kapasite değerleri.

Çıktı: Cezalandırılmış toplam tur uzunluğu

*Adım 1.* Her aracın tur uzunluğunu hesapla

*Adım 2.* Her aracın servis ve gezme sırasında kullandığı kapasite değerini hesapla

*Adım 3.* Eğer araca yüklenen toplam kapasite araç kapasitesini aşmıyorsa Adım 5'e git. Diğer durumlarda Adım 4'e git

*Adım 4.* Aracın kapasite aşım miktarı oranı ile tur uzunluğunu çarp (Ceza fonksiyonu).

*Adım 5.* Bütün araçların tur uzunluklarını topla.

Şekil 5.5'te verilen serim için örnek bir kromozomun Çizelge 5.2'de verilen ayrıt bitiş düğümleri ve maliyet bilgileri kullanılarak çözümlenmesi gösterilmiştir. Bu örnek kromozomun uyum değerinin hesaplanması için kapasite kısıtını sağlayıp sağlamadığı kontrol edilmelidir. Bu nedenle serimdeki ayrıtların servis talepleri ( $e_{ij} = 1, \forall (i, j) \in E$ ) ve geçiş talepleri ( $q_{ij}, \forall (i, j) \in R$ ) tanımlamalıdır. Bu bilgiler Çizelge 5.3'de verilmiştir.

**Çizelge 5.3.** Örnek serimin ayrit numaraları, bu ayrita karşı gelen servis ve geçiş talepleri

Ayrit Numarası	Servis Talebi	Geçiş Talebi
1	3	1
2	4	1
3	2	1
4	6	1
5	3	1
6	4	1
7	5	1
8	2	1

Örnek kromozom;

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

şeklinde tanımlanmıştır.  $m = 2$  ve her aracın kapasitesi 20 olarak verilmiştir ( $Q = 20$ ).

Bu kromozomun çözümlenmesi sonucu elde edilen araç turları aşağıdaki gibidir.

- 1. Araç Turu: 1-2-4-6-5-3-1
- 2. Araç Turu: 1-3-5-4-2-3-1

Araç turları elde edildikten sonra turlar Çizelge 5.4'te verildiği gibi değerlendirilir. Tablodan görüleceği gibi 2. aracın toplam talebi araç kapasitesini aşmaktadır. Bu nedenle ceza değeri birden büyüktür. Gerekli işlemler yapıldıktan sonra kromozomun amaç fonksiyonu değeri 45,1 olarak bulunmuştur.

**Çizelge 5.4.** Örnek kromozomun uyum değerinin hesaplanması adımları

	Tur	Tur Uzunluğu	Toplam Talep (Servis+Gezme)	Ceza Değeri	Cezalandırılmış Tur Uzunluğu
1. Araç	1-2-4-6-5-3-1	22	14+6=20	1	22
2. Araç	1-3-5-4-2-3-1	22	15+6=21	1,05	23,1

### 5.4.3. Genetik algoritma operatörleri

**Kopyalama.** Kopyalama, iyi uyum değerlerine sahip kromozomların yüksek olasılıkla yeni nesillere aktarılmasını sağlayarak başarısız özelliklerin elenmesi ve başarılı özelliklerin yaşatılmasına olanak sağlamaktadır. Bu çalışmada kopyalama işlemcisi olarak, Brindle'in deterministik seçim kuralı kullanılmıştır. Bu yöntemde, ilk olarak popülasyondaki her bireyin yeni jenerasyondaki beklenen kopya sayıları hesaplanır. Daha sonra beklenen sayıların tamsayı kısımları sayısınca bir sonraki jenerasyona kopyalanır. Son olarak popülasyonun geri kalan kısmı için beklenen sayıların kesirli olan kısımları büyükten küçüğe sıralanır. Popülasyon sayısı değişmeyecek şekilde bireyler yeni jenerasyona kopyalanır. Uyum değeri yüksek olan bireylerin beklenen sayıları da büyük olacağından bu bireylerin seçim olasılıkları daha yüksektir.

Popülasyonda diğer kromozomlara uyum değeri olarak baskınlık sağlayan bir kromozom (süper kromozom) varsa kopyalama sonrası bu kromozomdan fazla sayıda türetilir. Aynı kromozomdan fazla sayıda olması genetik algoritmanın yerel en iyiye takılmasına neden olur. Bu durumu engellemek için uyum değerleri ölçeklendirilir. Bu çalışmada ölçeklendirme için "Normalleştirme ölçeklendirmesi" kullanılmıştır. Aşağıda verilen tanımlamalara göre (5.13)'de uyum değerlerinin normalleştirilmesi için kullanılan denklem verilmiştir.

$$f_k \quad k. \text{ kromozomun cezalandırılmış toplam tur uzunluğu}$$

$f_{maks}$	Popülasyondaki en büyük cezalandırılmış toplam tur uzunluğu değeri
$f_{min}$	Popülasyondaki en küçük cezalandırılmış toplam tur uzunluğu değeri
$f'_k$	$k$ . kromozomun uyum değeri
$\gamma$	Küçük bir sayı (Sıfıra bölünmeyi engeller)

$$f'_k = \frac{f_{maks} - f_k + \gamma}{f_{maks} - f_{min} + \gamma} \quad (5.13)$$

Ölçeklendirme için kullanılan (5.13)'deki denklem aynı zamanda maliyet yönlü olan amaç fonksiyonu değerini fayda yönlü hale dönüştürür. Bir kromozomun cezalandırılmış toplam tur uzunluğu değeri ne kadar küçükse uyum değeri o kadar büyüktür.

**Çaprazlama.** Çaprazlama operatörü olarak permütasyon tipi kodlama için Murata ve Ishibuchi tarafından önerilmiş olan iki noktalı çaprazlama kullanılmıştır (1994).  $C_1$  ve  $C_2$  çocuk kromozomları  $P_1$  ve  $P_2$  ebeveyn kromozomları gösterebilir. Kromozomun uzunluğu  $n$  iken  $n_1 = [1, n]$  ve  $n_2 = [n_1, n]$  olmak üzere iki çaprazlama noktası rastsal olarak belirlenir. Daha sonra  $P_1$ 'de yer alan  $[n_1, n_2]$  aralığındaki kromozom parçası yine  $[n_1, n_2]$  aralığında  $C_2$ 'ye atanır.  $P_2$  kromozomundan  $P_1$ 'in  $[n_1, n_2]$  aralığında yer alan genler çıkartılır.  $P_2$  kromozomunda geriye kalan genler sırasıyla  $C_2$  kromozomunda boş kalan yerlere atanır. Aynı işlem  $C_1$  kromozomu için de tekrarlanır. Örnek iki kromozomun permütasyon tipi iki noktalı çaprazlanması örneği Çizelge 5.5'te verilmiştir.

**Çizelge 5.5.** Permütasyon tipi iki noktalı çaprazlama örneği

Ebeveyn-1	2	7	6	3	1	5	4	8
Ebeveyn-2	4	5	6	2	8	7	3	1
Çocuk-1	4	6	2	3	1	5	8	7
Çocuk-2	6	3	1	2	8	7	5	4

**Mutasyon.** Önerilen GA'da mutasyon operatörü olarak yer değiştirme işlemi uygulanmıştır. Kromozomun uzunluğu  $n$  iken  $n_1 = [1, n]$  ve  $n_2 = [1, n]$  olmak üzere iki gen pozisyonu seçilir. Daha sonra seçilen bu pozisyonlardan genlerin içerikleri yer değiştirilir. Örnek bir kromozom üzerinde gerçekleştirilen mutasyon işlemi Çizelge 5.6'da gösterilmiştir. Verilen kromozomda rassal olarak  $n_1 = 4$  ve  $n_2 = 8$  olarak seçilmiştir. Daha sonra bu pozisyonda bulunan genler yer değiştirilmiştir.

**Çizelge 5.6.** İkili yer değiştirme mutasyonu örneği

Mutasyon Öncesi	2	7	6	3	1	5	4	8
Mutasyon Sonrası	2	7	6	8	1	5	4	3



## 6. MOBİL ROBOTLARDA UYGULAMA

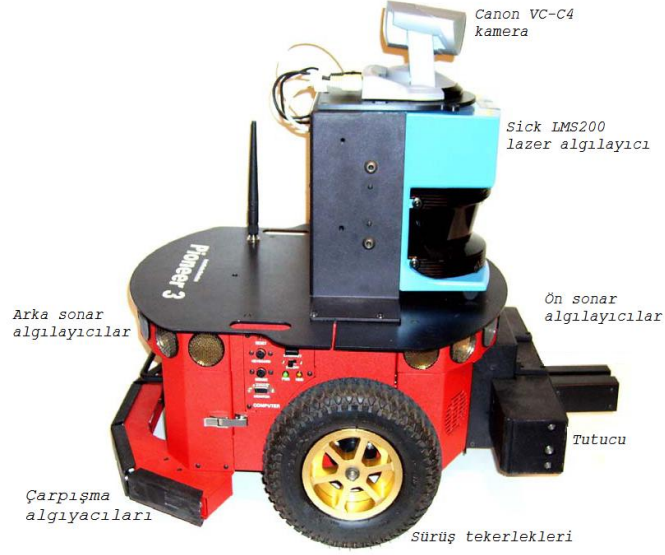
Yeni KARP, çok robotlu algılayıcı tabanlı kapsama probleminden hareketle tanımlanmış bir problemdir. Bu nedenle çalışmanın bu bölümünde çok robotlu algılayıcı tabanlı ortam kapsanması probleminde, gezgin robotların yol planlarının Yeni KARP'ın çözüm yöntemi olarak önerilen DUA'nın kullanıldığı çok robotlu gezgin robot uygulamaları anlatılmıştır. Çoklu gezgin robot uygulamaları Pioneer 3-DX robotlar ve MobileSim simülasyon ortamı kullanılarak Eskişehir Osmangazi Üniversitesi Yapay Zeka ve Robotik laboratuvarında (YZR, 2009) oluşturulmuş deney ortamı ve Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği laboratuvarı 1. katında gerçekleştirilmiştir. Bu bölümde ilk olarak uygulama ile ilgili algılayıcı, haberleşme vs. tanımlamaları, daha sonrasında ise gerçekleştirilen uygulamaların detayları anlatılmıştır.

### 6.1. Uygulamalardan Tanımlamalar

Bu çalışmada gerçekleştirilen uygulamalar hem fiziksel robotlarla hem de MobileSim olarak adlandırılan gezgin robot simülasyon ortamında test edilmiştir. MobileSim, MobileRobots Inc. tarafından geliştirilmiş olan robotların simülasyonun gerçekleştirilebildiği bir yazılımdır (ActivMedia Robotics, 2007; Whitbrook, 2006).

Bu çalışmada gerçekleştirilen uygulamalarda Pioneer 3-DX model gezgin robot platformu kullanılmıştır. Pioneer 3-DX gezgin robotunun resmi Şekil 6.1'de gösterilmektedir. Bu robot platformu, 44,5 cm. x 39,3 cm. x 23,7 cm. boyutlarında, iki adet diferansiyel sürüş tekerleğine sahip bir tabana sahiptir. Ayrıca, çevrenin algılanmasında kullanılmak üzere çeşitli algılayıcılar ile donatılmıştır. Mesafe algılamak üzere ön ve arkasında 8'er adet sonardan oluşan sonar dizisi mevcuttur. Ayrıca, bağıl konum hesaplamaları için, tekerleklere monte edilmiş sayma algılayıcıları (odometry); çarpışmaları algılamak üzere arka çarpışma algılayıcıları (bumper); yön tayini için elektronik pusula; görsel algılamalarda kullanılmak üzere dönme-eğilme yakınlaştırma özellikli kamera (Canon VC-C4 pan-tilt-zoom camera) ve lazer algılayıcısı ile

donatılmış durumdadır. Bunların yanı sıra nesne taşıma amacıyla kullanılmak üzere bir tutucu mekanizmaya sahiptir.



**Şekil 6.1.** Pioneer 3-DX gezgin robot platformu

Çok robotlu algılayıcı tabanlı kapsama probleminde, planlama fazında karşılaşılan kapsama sırasında kat edilen yolu en küçüklenmesi, robotlara yüklerin dengeli dağıtılması gibi problemlerinin yanı sıra kapsama çok robotla gerçekleştirildiğinden robotların koordinasyon problemiyle karşılaşılmaktadır. Bu nedenle bu çalışmada çok robotun kontrolü için erkin (agent) tabanlı kontrol mimarisi önerilmiştir. Erkin, bir çevre içinde bulunan ve tasarım amaçlarına hizmet edecek şekilde bu çevrede otonom hareket edebilen bilgisayar sistemi olarak tanımlanır (Weiss, 2000; Wooldridge, 2002). Erkin tabanlı programlama etkinlik, güvenilirlik, esneklik ve yeniden kullanılabilirlik sağladığı için kullanılmıştır. Önerilen erkin tabanlı kontrol mimarisinde, dört tip erkin kullanılmıştır. Bunlar kullanıcı arayüzü erkini, planlama erkini, hareket erkini ve haberleşme erkini.

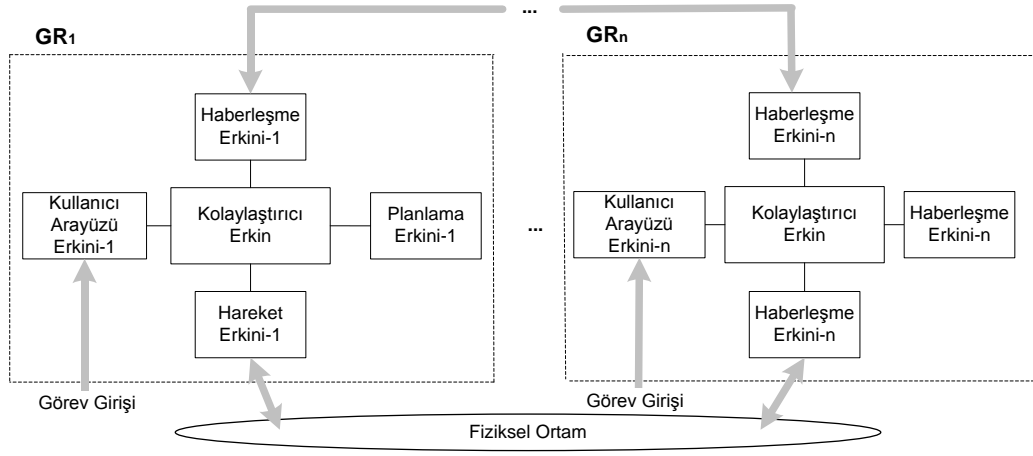
Kullanıcı Arayüzü Erkini: Bu erkinin temel görevi kullanıcı ile robot arasındaki etkileşimi sağlamaktır. Bu çalışmada kullanıcı ile robot arasındaki iletişim terminal ekranı üzerinden sağlanmaktadır.

Planlama Erkini: Bu erkinin hareket planı, yol planı, görev planı gibi planların türetilmesi ile görevlidir. Bu çalışmada planlama erkini DUA'yı kullanarak gezgin robot için yol planı türetmektedir. DUA'nın yerini önerilen genetik algoritma veya herhangi bir planlama algoritması alabilir.

Hareket Erkini: Bu erkin, robotun hareketi, algılaması gibi donanımı ile ilgili görevleri gerçekleştirmesini sağlar. Bu çalışmada hareket erkini, planlama erkininden gelen yol planını ortama ait bilgileri kullanarak düğümlere hareket görevlerine dönüştürür ve sırasıyla bu düğümlere gitme davranışını gerçekleştirir. Ayrıca ortamdaki engellerden sakınılması, ortamın kapsanması veya konumlandırma amacıyla da algılayıcılardan alınan bilgiler yorumlanır.

Haberleşme Erkini: Bu erkin robotlar arası haberleşme ile görevlidir. Önerilen kontrol mimarisinde robotlar haberleşme erkinleri aracılığı ile birbirlerine mesaj göndermektedir. Bu çalışmada haberleşme erkinleri arasındaki iletişim TCP/IP ile gerçekleşmiştir.

Robot takımında yer alan bütün robotlar tanımlanan bu 4 yazılım erkinine sahiptir. Robot içerisinde bu dört erkin birbirleriyle Açık Erkin Mimarisinde (Open Agent Architecture) bulunan kolaylaştırıcı erkin (facilitator) aracılığıyla haberleşmektedir (Martin, et.al., 1999). Önerilen mimarinin blok diyagramı Şekil 6.2'de verilmiştir.



Şekil 6.2. Mimarinin blok diyagramı

## 6.2. Uygulamalar ve Test Sonuçları

Test ortamlarında gerçekleştirilen uygulamalarda kapsama algılayıcısı olarak lazer algılayıcı kullanılmıştır. Uygulamalarda kullanılan lazer algılayıcısının 50 m. alanı tarayabilecek donanımdadır. Fakat uygulamaların gerçekleştirildiği ortamlar dar olduğundan lazer tarama alanı 3m. ile sınırlandırılmıştır. Robot bir noktadan geçtiğinde lazer algılayıcı 3 m. çapındaki yarım dairenin içerisinde kalan her noktayı taramış olur.

Daha önce anlatıldığı gibi çok robotlu algılayıcı tabanlı ortam kapsaması probleminde iki tip ayırıt talebi tanımlanmıştır. Bunlar, kapsama talebi ve gezinme talebi şeklindedir. Gezgin robotlarda ayırıt talebi ayırıtlarda harcanan enerjiye karşı gelmektedir. Robot gezinme sırasında motoru, üzerindeki bilgisayarı, mikro denetleyici kartı ve engellerden sakınmak için ise sonar dizisini çalıştırmaktadır. Bu durumda robotun gezinme enerjisi bu birimler tarafından harcanan enerjinin toplamıdır. Tanımlanan enerjinin belirlenmesi ile ilgili (Mei, et.al., 2006) çalışmasında Pioneer 3-DX robotunun hıza bağımlı enerji modeli çıkartılmıştır. Burada çıkartılan modele göre robot gezinme sırasında  $e_{ij} = (17.49 + (7.4 \cdot (\text{hiz} / 1000))) \cdot t_{ij}$  kadarlık enerji harcamaktadır. Kapsamada ise lazer algılayıcı kullanılmaktadır. Kapsama enerjisi ise

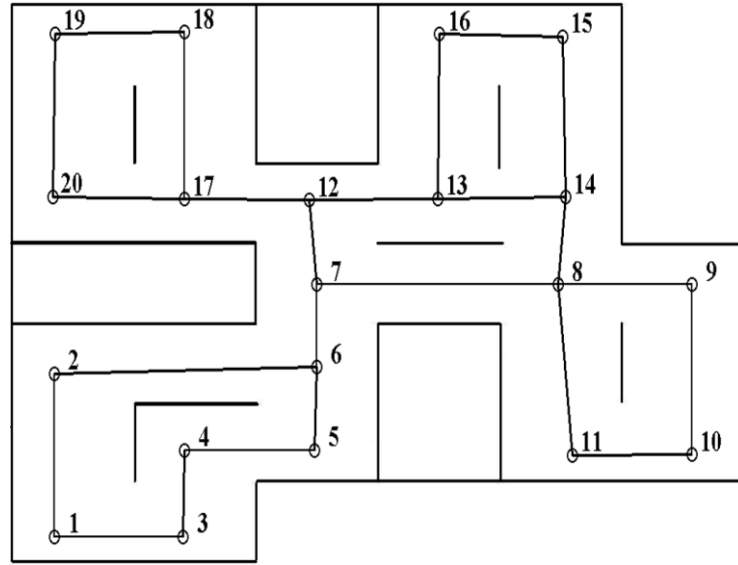
lazer algılayıcı tarafından harcanan enerjidir. Gezgin robot üzerinde bulunan Sick LMS-200 lazer algılayıcısının gücü 20 Watt'dır. Buradan hareketle kapsama enerjisi  $q_{ij} = 20 \cdot t_{ij}$  olarak elde edilir. Uygulamalarda robotun hızı sabit ve 400mm/s olarak ayarlanmıştır. Buradan robot gezinme için 0,051 joule/mm'lik, kapsama için ise 0,050 joule/mm'lik enerji harcamaktadır. Robot kapsama sırasında hareketine devam ettiğinden robot servis verdiği bir ayrıttan geçerken 0,101 joule/mm'lik enerji harcamaktadır.

Önerilen yöntemin etkinliğinin gösterilmesi için Eskişehir Osmangazi Üniversitesi, Yapay Zeka ve Robotik laboratuvarında kurulan deney ortamında test edilmiştir. Test ortamına ait fotoğraf Şekil 6.3'te verilmiştir. İlk olarak ortam GVD tabanlı serim ile modellenmiştir. Modelleme sonucu elde edilen serim Şekil 6.4'te verilmiştir. Serimdeki 1 numaralı düğüm depo düğümüdür.



**Şekil 6.3.** Test ortamının fotoğrafı

Ortamın GVD tabanlı serimi robotlarda komşuluk matrisi ve düğüm koordinatları bilgisi ile saklanmaktadır. Düğüm sayısı  $n$  olmak üzere komşuluk matrisi  $n \times n$ 'lik bir matristir. Eğer serimdeki  $v_i$  ve  $v_j$  düğümleri arasında bir bağlantı varsa komşuluk matrisinin  $(i, j)$ . ve  $(j, i)$ . elemanı 1 değerini, diğer durumlarda 0 değerini alır. Düğüm koordinatları matrisi ise  $n \times 2$ 'lik bir matristir. Bu matriste her düğümün koordinat sistemindeki  $X$  ve  $Y$  koordinatları yer almaktadır. Ayrıt maliyetleri ise iki düğüm arasında Öklidyen uzaklık ile belirlenmektedir.  $v_i$  ve  $v_j$  düğümlerinin koordinatları sırasıyla  $(X_i, Y_i)$  ve  $(X_j, Y_j)$  olmak üzere  $v_i$  ve  $v_j$  düğümleri bağlı ise bu iki düğüm arasındaki uzaklık  $d_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$  şeklinde hesaplanır.



Şekil 6.4. Ortamın GVD tabanlı serimi

Kapasite kısıtlı robotların ortamı kapsayan turları DUA kullanılarak türetilenir. DUA'nın girdisi  $G = (V, E)$  serimi,  $R$  gezilmesi gereken ayrıtlar kümesi ve  $R$ 'yi kapsayan Euler turudur.  $G$  serimi GVD tabanlı olarak elde edilen yönsüz serimdir. Bütün ortam kapsanması istediği için gezilmesi gereken ayrıtlar kümesi

serimdeki bütün ayrıtlardır ( $R = E$ ). Son olarak  $R$ 'yi kapsayan Euler turu KPP çözüm algoritması olan Frederickson'ın sezgiseli ile belirlenebilir.  $R = E$  olduğundan Euler turu CPP çözüm algoritması ile de elde edilebilir.

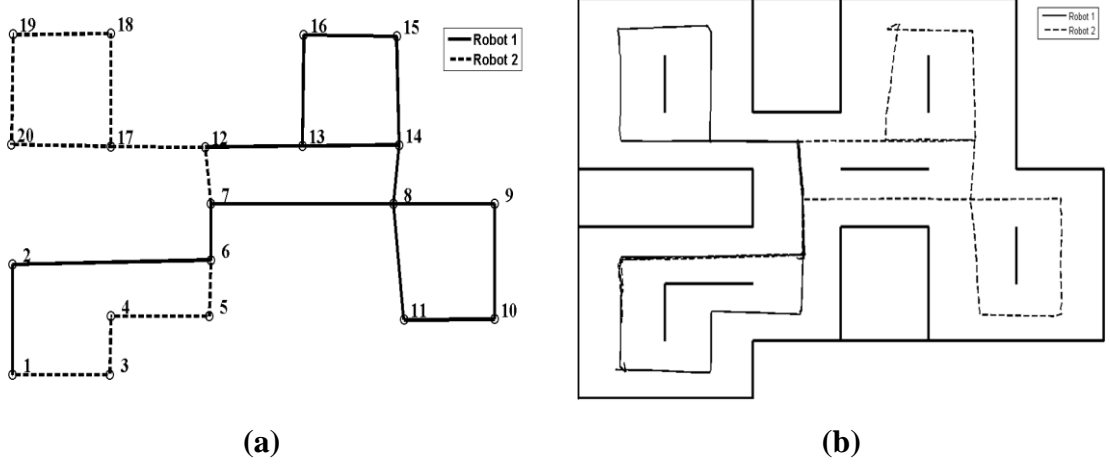
Frederickson'ın sezgiseli ile belirlenen Euler turu: **(1-2), (2-6), (6-7) (7-8), (8-9), (9-10), (10-11), (11-8), (8-14), (14-15), (15-16), (16-13), (13-14), (14,13), (13-12), (12-17), (17-18), (18-19), (19-20), (20-17), (17-12), (12-7), (7-6), (6-5), (5-4), (4-3), (3-1)** şeklindedir. Her ayrıta ilk uğrandığında servis verildiği kabul edilirse kalın olarak yazılmış ayrıtlar servis verilerek geçilen ayrıtları, italik yazılan ayrıtlar servis verilmeden (geçiş için kullanılan ayrıt) geçilen ayrıtları göstermektedir. Ortamın kapasite kısıtı olmayan bir robotla tarandığı varsayılır ise Euler turunun gezilmesi ortamın kapsanması için yeterlidir. Bu durumda robotun tur uzunluğu 42931,5 mm, harcadığı enerji 4162,94 joule olur. Ortamın kapsanma süresi ise 282 s.'dir.

Enerji kapasitesi 3200 joule olarak kabul edilen 2 robot 1. düğümde bulunmaktadır. Bu durumda DUA ile elde edilen araç turları aşağıdaki gibidir.

1. Robot turu: **(1-2), (2-6), (6-7) (7-8), (8-9), (9-10), (10-11), (11-8), (8-14), (14-15), (15-16), (16-13), (13-14), (14,13), (13-12), (12-7), (7-6), (6-2), (2-1)**
2. Robot turu: *(1-2), (2-6), (6-7), (7-12), (12-17), (17-18), (18-19), (19-20), (20-17), (17-12), (12-7), (7-6), (6-5), (5-4), (4-3), (3-1).*

DUA ile elde edilen robot turlarında servis verilen ayrıtlar Şekil 6.5.a'da verilmiştir. Sırasıyla, robotların tur uzunlukları 33352 mm. ve 23946,9 mm. ve robotlar tarafından harcanan enerji 2953,03 ve 1944,44 joule'dür. Toplam tur uzunluğu 57298,9 mm ve toplam harcanan enerji 4897,47 joule olarak elde edilmiştir. 2 robotla yapılan uygulamada kapsama zamanı 228 s.'dir. Kapsama zamanı turunu en geç bitiren robotun kapsama zamanına eşittir. Tek robotlu durum ile karşılaştırıldığında kapsama zamanı %19 oranında azalmıştır. Buna rağmen uzaklık değeri %33, harcanan enerji miktarı %17 oranında artmıştır. Uygulama sırasında gerçek robotların konum bilgileri

kullanılarak robotların takip ettiği yollar Şekil 6.5.b'deki gibi elde edilmiştir. Uygulamaya ait videoya [www.ai-robotlab.ogu.edu.tr](http://www.ai-robotlab.ogu.edu.tr) sitesinden ulaşılabilir.



**Şekil 6.5. a)** İki robotla yapılan uygulamada servis verilen ayrıtların robotlar arasında bölümlendirilmesi, **b)** Uygulama sırasında robotların takip ettiği yollar

İkinci uygulamada 3 robot depo düğümü olan 1. düğümde bulunmaktadır. Bu uygulama MobileSim simülasyon ortamında gerçekleştirilmiştir. DUA'ya aynı Euler tur verilmiş ve robotların enerji kapasitesinin 2000 joule olduğu kabul edilmiştir. Bu durumda robot turları aşağıdaki gibi elde edilmiştir.

1. Robot Turu: **(1-2), (2-6), (6-7), (7-8), (8-9), (9-10), (10-11), (11-8), (8-7), (7-6), (6-2)** ve **(2-1)** (Noktalı Çizgi).

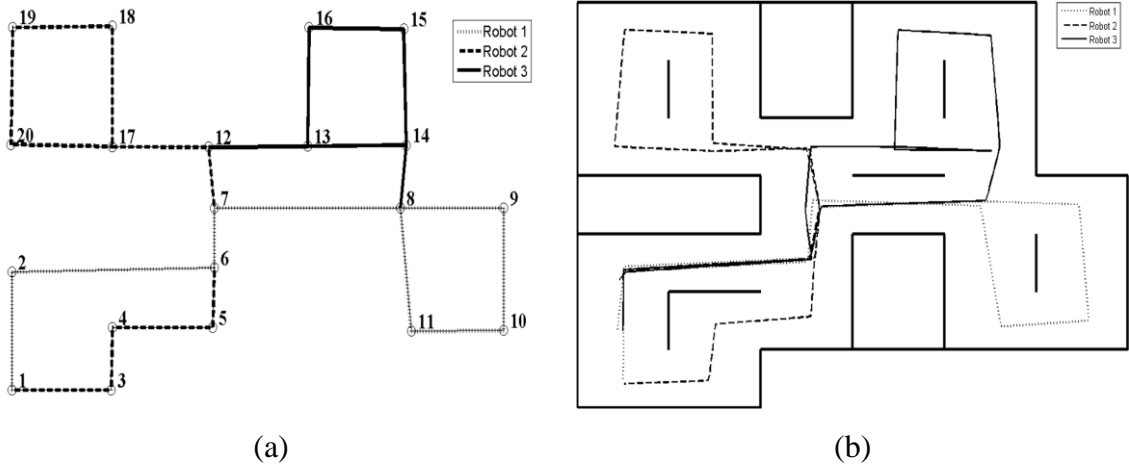
2. Robot Turu: **(1-2), (2-6), (6-7), (7-12), (12-17), (17-18), (18-19), (19-20), (20-17), (17-12), (12-7), (7-6), (6-5), (5-4), (4-3)**, ve **(3-1)** (Kesikli çizgi).

3. Robot Turu: **(1-2), (2-6), (6-7), (7-8), (8-14), (14-15), (15-16), (16-13), (13-14), (14-13), (13-12), (12-7), (7-6), (6-2)** ve **(2-1)** (Devamlı Çizgi).

Ayrıtların 3 robot kullanılarak önerilen algoritma ile bölümlendirilmesi ile elde edilen sonuç Şekil 6.7.a'da gösterilmiştir. Robotlar kapsama sırasında sırasıyla 1990,33;

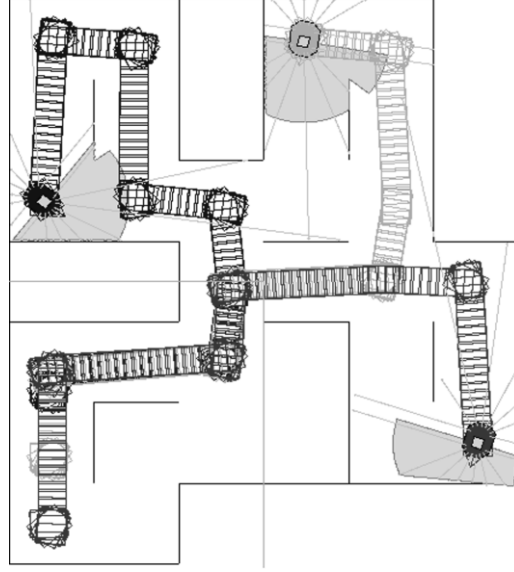


1944,44 ve 1807,44 joule enerji harcamıştır. Robotların tur uzunlukları ise 23466.8 mm, 23946,9 mm ve 26108,4 mm olarak elde edilmiştir. Robotlar tarafından toplam harcanan enerji 5742,21 joule, toplam tur ise 73520,8 mm'dir. Bu uygulamadaki kapsama zamanı ise 168 s olarak gözlenmiştir. Tek robotlu durum ile karşılaştırılırsa kapsama zamanı %42 oranında azalmıştır. Fakat toplam harcanan enerji ve toplam tur uzunluğu değerleri sırasıyla %71 ve %37 oranında artmıştır.



**Şekil 6.6.** a) Üç robotla MobileSim simülasyon ortamında yapılan uygulamada servis verilen ayrıtların robotlar arasında bölümlendirilmesi, b) Uygulama sırasında robotların takip ettiği yollar

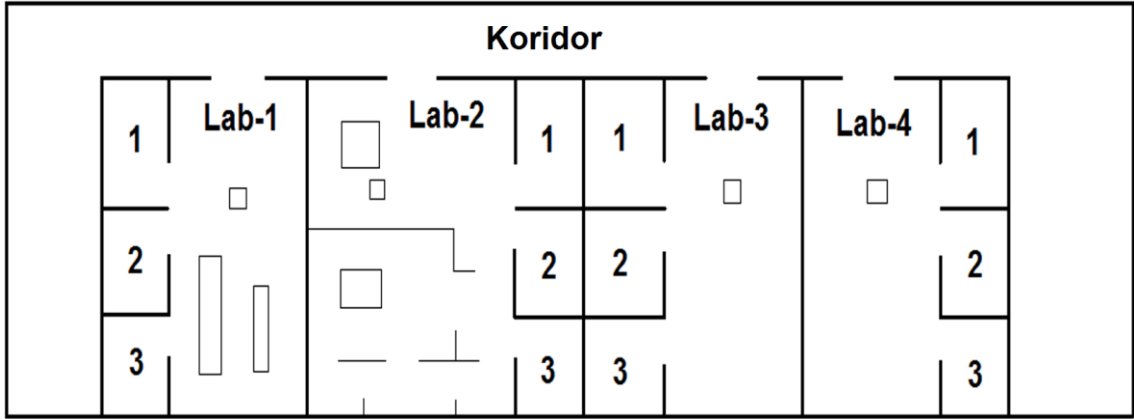
Üç robotla yapılan uygulama iki robotlu yapılan uygulama ile karşılaştırıldığında toplam harcanan enerji %18 artarken en uzun tur uzunluğu %22 oranında azaldığı gözlenmiştir. Şekil 6.7'da uygulamanın gerçekleştirildiği MobileSim simülasyon ortamından alınan bir ekran görüntüsü verilmiştir.



**Şekil 6.7.** Üç robotlu algılayıcı tabanlı kapsama uygulamasında MobileSim simülasyon ortamından alınan bir ekran görüntüsü

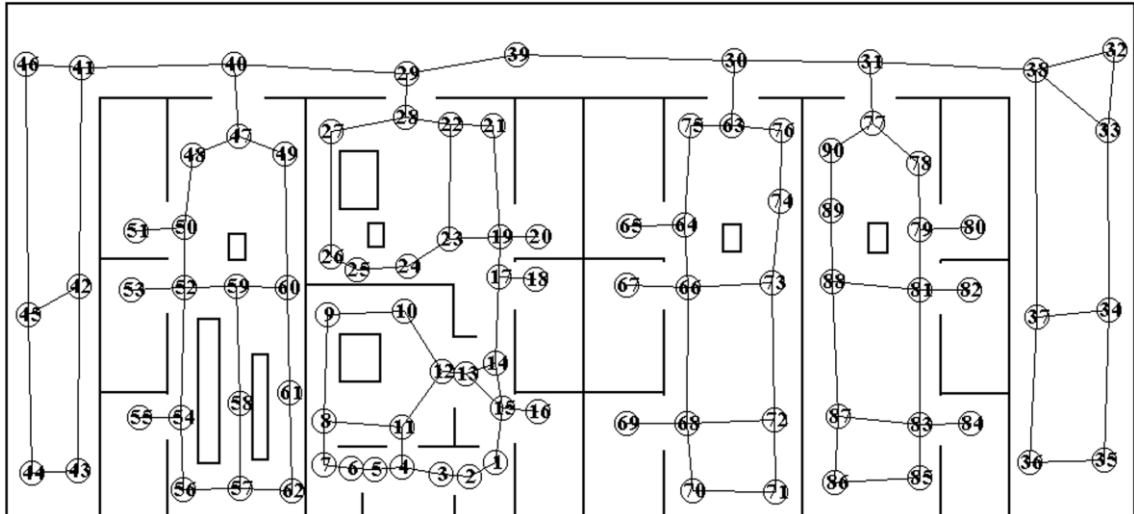
Test sonuçlarından da gözlendiği gibi robot sayısı arttıkça servis verilmeden geçilen ayırıt sayısı arttığından toplam tur uzunluğu ve toplam harcanan enerji miktarı artmaktadır. Bununla birlikte beklenen bir sonuç olarak da kapsama süresi azalmaktadır. Bu noktada klasik KARP ve Yeni KARP'ın arasındaki fark daha iyi gözlenmektedir. Klasik KARP'ta sadece servis verilmesi gereken ayırıtlarda kapasite kullanıldığından uygulama kaç araçla yapılırsa yapılsın toplam harcanan kapasite miktarı değişmez. Fakat Yeni KARP'ta gezinme sırasında da enerji harcandığından servis verilmeden geçilen ayırıt sayısı arttığında toplam harcanan enerji artar.

Önerilen DUA'nın büyük serimlerdeki etkinliğini göstermek için Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Mühendisliği laboratuvarının 1. katında test ortamı olarak seçilmiştir. Bu katta dört laboratuvar bulunmaktadır. Her laboratuvarın içerisinde üç oda, masalar, depolama kabinleri ve kolonlar bulunmaktadır. Laboratuvarları koridor birbirine bağlamaktadır. Bu kata ait yerleşim haritası Şekil 6.8'de verilmiştir.



**Şekil 6.8.** Laboratuvar katına ait yerleşim haritası.

Şekil 6.8’de verilen ortamın GVD tabanlı oluşturulmuş serimi Şekil 6.9’da verilmiştir. Bu serim 90 düğüm ve 109 ayrıttan oluşmaktadır. Serimin depo düğümü 44 numaralı düğüm olarak belirlenmiştir.



**Şekil 6.9.** Laboratuvar katının GVD tabanlı serimi.

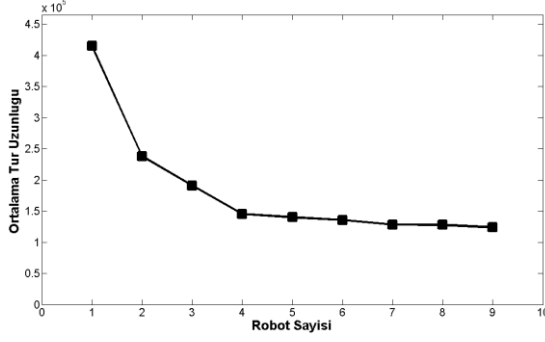
Şekil 6.9’da verilen serim için önerilen DUA’nın bir robottan dokuz robota kadar testleri gerçekleştirilmiş, testler sonucu elde edilen sonuçlar Çizelge 6.1’de verilmiştir.

Bu tabloda robot sayısına karşılık testler sonucu elde edilen toplam tur uzunluğu (TTU), toplam harcanan enerji (THE), ortalama tur uzunluğu (OTU), ortalama harcanan enerji (OHE), en uzun tur uzunluğu (EUTU), en fazla harcanan enerji (EFHE) ve çözüm zamanı değerleri verilmiştir. Robot sayısından bağımsız olarak çözüm zamanı yaklaşık olarak 0,9 s olarak gözlenmiştir.

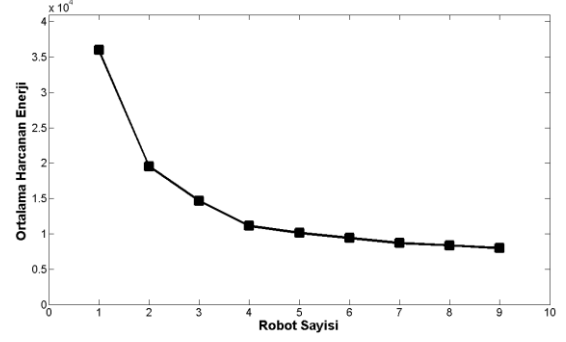
**Çizelge 6.1.** Laboratuvar katında gerçekleştirilen testler sonucu elde edilen sonuçlar

Robot Sayısı	TTU (mm)	THE (joule)	OTU (mm)	OHE (joule)	EUTU (mm)	EFHE (joule)	Çözüm Zamanı (ms)
1	415106.0	35975.9	415106.0	35975.9	415106.0	35975.9	912.0
2	476105.0	39094.5	238052.5	19547.3	334207.0	27364.4	913.0
3	572074.0	44000.8	190691.3	14666.9	210503.0	15852.8	935.0
4	581271.2	44471.1	145317.8	11117.8	184193.0	12843.6	908.0
5	700750.0	50579.6	140150.0	10115.9	158249.0	11730.1	902.0
6	813464.0	56342.0	135577.3	9390.3	151877.0	9920.4	883.0
7	899075.5	60718.9	128439.4	8674.1	151530.0	9699.6	910.0
8	1021670.1	66986.5	127708.8	8373.3	153215.0	9087.0	892.0
9	1112978.5	71654.6	123664.3	7961.6	141433.0	8433.5	910.0

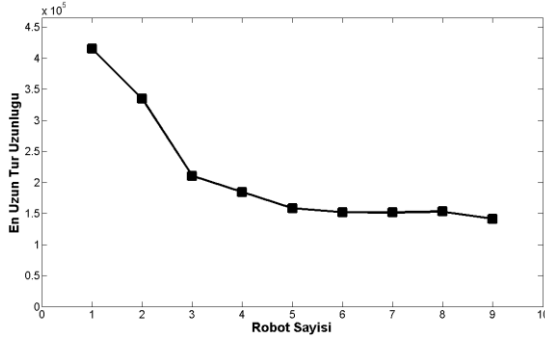
Tablodan görüldüğü gibi robot sayısı arttıkça servis verilmeden gezilen ayrıt sayısı arttığından hem TTU hem de THE artmaktadır. Diğer taraftan, OTU, OHE, EUTU ve EFHE değerleri azalmaktadır. Robot sayısına bağlı olarak OTU, OHE, EUTU ve EFHE değerlerinin değişimini gözlemlemek için Şekil 6.10'da yer alan dört grafik verilmiştir.



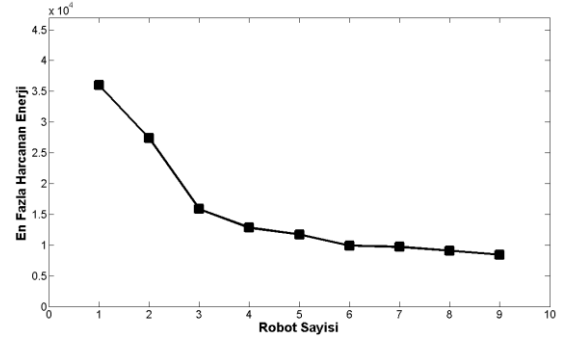
(a)



(b)



(c)



(d)

**Şekil 6.10.** a) Ortalama tur uzunluğu, b) Ortalama harcanan enerji, c) En uzun tur uzunluğu, d) En fazla harcanan enerji

Şekil 5.10'dan görüleceği gibi 4 robottan sonra yeni robot eklentisinin robot başına ortalama tur üzerindeki marjinal faydası azalmaktadır. Benzer şekilde diğer durumlarda da 5 robottan sonra yeni robot eklentisinin marjinal faydası azalmaktadır. Bu nedenle verilen ortamdaki kapsama görevi 5 robota atanması anlamlıdır. Bu tür bir analiz herhangi bir ortam kapsama görevi yapacak en fazla robot sayısının belirlenmesinde kullanılabilir.

## 7. ÖNERİLEN YÖNTEMLERİN TESTİ

Bu çalışmada önerilen Yeni KARP literatürde ilk kez tanımlandığından ne test problemleri ne de problem için elde edilmiş alt sınır ya da en iyi çözüm bulunmaktadır. Bu nedenle literatürde klasik KARP için önerilmiş olan test problemleri Yeni KARP için uyarlanmıştır. Önerilen genetik algoritma ve DUA'nın etkinliğinin gösterilmesi için probleme ait ya bir alt sınır ya da en iyi çözüm olmalıdır. Bu nedenle bu çalışmada Yeni KARP için önerilen matematiksel model kullanılarak türetilen bazı yeni test problemleri için en iyi çözümler elde edilmiştir. Fakat kuraldışı alt turları engelleme kısıtı modele üstel sayıda kısıt eklenmesine neden olduğundan dolayı klasik KARP için önerilmiş test problemlerinden sadece küçük boyutlu olanlar kullanılmıştır.

Literatürde KARP için <http://www.uv.es/~belengue/carp.html> adresinde test problemleri verilmiştir. Bu çalışmada, KARP test problemlerinden, küçük boyutlu problem setleri olması nedeniyle *kshs* ve *gdb* test problemleri Yeni KARP için uyarlanmıştır. *kshs* sınıfı test problemindeki düğüm sayısı 8, 9 veya 10, ayrıt sayısı 15 ve araç sayısı da 3 veya 4'tür. Serimdeki bütün ayrıtlar gezilmesi gereken ayrıt listesindedir (Kiuchi, et.al., 1995). *gdb* sınıfındaki test problemlerinde düğüm sayısı 7-27 aralığında, ayrıt sayısı 11-55 aralığında, araç sayısı ise 3-10 aralığındadır (Baker, et.al., 1983). Bu iki sınıf test problemi örneklerinde bütün ayrıtlar gezilmesi gereken ayrıtlar listesindedir. Test problemi örneklerinde bütün serim gezilmesi gerekiyor ( $R = E$ ) olsa da bu çalışmada önerilen yöntemler  $R \subseteq E$  olması durumunda da çalışmaktadır. Bununla birlikte büyük boyutlu test problemlerinin de çözülebildiğinin gösterilmesi için (Eglese, 1994 and Eglese and Li, 1996: Belenguer and Benavent) çalışmalarında önerilmiş olan *eglese* isimli test problemleri kullanılmıştır. Bu test problemlerinde serimdeki düğüm sayısı 77 veya 140, ayrıt sayısı ise 98 veya 190 olarak tanımlanmıştır. “*eglese*” test problemleri de KARP için türetilmiş test problemleri olduğundan geçiş için herhangi bir talep tanımlanmamıştır. “*eglese*” test problemlerinde gezilmesi gereken ayrıtlar için talep değeri ayrıt maliyetlerine eşit olarak tanımlanmıştır ( $q_{ij} = c_{ij}, \forall (i, j) \in R$ ). Bu nedenle geçiş talepleri de ayrıt maliyetlerine eşit olarak belirlenmiştir ( $e_{ij} = c_{ij}, \forall (i, j) \in E$ ). Araçlar geçiş sırasında da kapasite kullanacağından

araç kapasite değerleri arttırılmıştır. “eglese” test problemi setindeki araç kapasite değerlerinin arttırılması ve geçiş talepleri tanımlanması nedeniyle değiştirilmiş “eglese” anlamında “d-eglese” olarak adlandırılmıştır.

*Kshs* ve *gdb* sınıfı test problemleri klasik KARP için önerilmiş olduğundan aracın gezme sırasında harcadığı kapasite tanımlanmamıştır. Bu nedenle ilk olarak, aracın gezme sırasında harcayacağı kapasitelerin  $(e_{ij}, \forall(i, j) \in E)$  tanımlanması gerekir. Gezgin robotlarda yapılan uygulamalarda gezgin robotun gezinme sırasında  $(e_{ij})$  ve kapsama sırasında harcadığı enerji  $(q_{ij})$  oldukça yakın olarak hesaplanmıştır. (Yeni KARP’ta araç kapsama sırasında  $e_{ij} + q_{ij}$  gezinme sırasında ise  $e_{ij}$  miktarında kapasite kullanılmaktadır). Bu nedenle test problemlerinde bulunan  $q_{ij}$  değerlerine yakın  $e_{ij}$  değerleri türetilmiştir. Test problemlerindeki  $q_{ij}$  değerlerinin nasıl türetildiği bilinmediğinden, her test probleminin  $q_{ij}$  değerleri için Kolmogorov-Smirnov (KS) dağılım uygunluk testi yapılmıştır. Testler sonucunda sabit talep değerine sahip olmayan test problemleri haricindeki test problemlerinden bir tanesi Poisson dağılımına diğerleri ise kesikli düzgün dağılıma uygun elde edilmiştir. Elde edilen dağılımlar ve bu dağılımlara ait parametreler Ekler bölümünde verilmiştir.

Dağılımların belirlenmesinden sonra bu dağılımlara uygun olarak  $e_{ij}$  değerleri türetilmiştir. Var olan test problemlerinde gözden geçirilmesi gereken diğer parametre ise araç kapasiteleridir. Klasik KARP’ta araç sadece servis sırasında kapasite kullanıldığından eldeki araç sayısına bağlı olarak araç kapasiteleri belirlenebilir. Fakat Yeni KARP’ta araç gezinme sırasında da kapasite kullandığından tur tamamen oluşturulmadan aracın harcadığı kapasite değeri tam belirlenememektedir. Bu nedenle var olan test problemlerindeki araç kapasite değerleri 2.5 katına çıkartılarak bütün test problemleri matematiksel model ile 3 dakika çözüm süresi durdurma kriteri ile çözülmüştür. Bu test ile belirlenmek istenen, bir serimdeki gezilmesi gereken ayrıtlar kapsanırken hangi oranda ayrıtlardan servis verilmeden geçildiğini belirlemektir. Model iki test probleminde 3 dakika içerisinde uygun çözüm bulamamıştır. Diğer elde edilen sonuçlara göre bir serimdeki ayrıtlar en az 0, en fazla 0,89 ve ortalamada 0,35 kat fazla

sefer ziyaret edilmektedir. Elde edilen çözümlerden hareketle araç kapasitesi,  $m$  eldeki araç sayısı olmak üzere Denklem (7.1) ile belirlenmektedir.

$$Q = \frac{\left[ \left( \sum_{\forall(i,j) \in E} e_{ij} \right) \cdot 1,7 + \sum_{\forall(i,j) \in R} q_{ij} \right]}{m} \quad (7.1)$$

Test problemi örneklerindeki en büyük boyutlu problem 13 düğümlü olduğundan Frederickson'ın sezgiselinde yer alan en küçük maliyetli eşleştirme problemi matematiksel model yerine tam sayımlama ile gerçekleştirilmiştir. Sayımlanacak eşleştirme sayısı serimdeki tek dereceli düğüm sayısına bağlıdır. Serimdeki tek dereceli düğüm sayısı  $|V_{tek}|$  ise, tam sayımlamada  $\binom{|V_{tek}|}{2}$  adet kombinasyonun denenmesi gerekir.

Testler gerçekleştirilirken önerilen genetik algorithmada popülasyon büyüklüğü 30, çaprazlama oranı 0,8, mutasyon 0,01 olarak seçilmiştir. Ayrıca genetik algoritma iki farklı durdurma kriteri kullanılmıştır. Bunlar;

- Uygun bir çözüm bulunduktan sonra 300 ardıştırma boyunca elde edilen en iyi çözüm iyileştirilmiyorsa durdur veya
- 3000 ardıştırma boyunca uygun bir çözüm bulunmazsa durdur, şeklindedir.

Değiştirilmiş test problemleri önerilen genetik algoritma kullanılarak çözülmüştür. Bu çalışmada çözülen test problemleri literatürdeki test problemlerinden geçiş sırasında kapasite kullanması yönüyle farklı olduğundan test problemleri isimlerin başına değiştirilmiş anlamında “d-“ öneki konulmuştur. Her test problemi 10 kez çözülmüştür. Her çözümde elde edilen en iyi çözüm ve çözüm süresi Ekler bölümünde verilmiştir. Çizelge 7.1'de ise testler sonucunda elde edilen amaç fonksiyonu ve çözüm süresini ortalama, en küçük ve en büyük değerleri verilmiştir. Bu tabloda  $|V|$  düğüm sayısını,  $|E|$  ayrıt sayısını,  $|R|$  gezilmesi gereken ayrıt sayısını ve  $m$  eldeki araç sayısını göstermektedir.



**Çizelge 7.1.** Önerilen genetik algoritma ile elde edilen çözümlerin amaç fonksiyonu ve çözüm süresi açısından ortalama, en küçük ve en büyük değerleri

Test Problemi	V	E	R	m	Ortalama Amaç Fonksiyonu Değeri	En Küçük Amaç Fonksiyonu Değeri	En Büyük Amaç Fonksiyonu Değeri	Ortalama Çözüm Süresi	En Kısa Çözüm Süresi	En Uzun Çözüm Süresi
d-kshs1	8	15	15	4	14813.8	14729	15272	7.26	5.35	11.46
d-kshs2	10	15	15	4	9930.8	9863	10022	8.57	5.41	20.66
d-kshs3	6	15	15	4	9354.6	9320	9666	4.97	3.96	7.25
d-kshs4	8	15	15	4	11196.4	11098	11812	9.45	5.16	32
d-kshs5	8	15	15	3	11071.6	10957	11715	5.48	4.36	8.62
d-kshs6	9	15	15	3	10284.7	10197	10966	5.4	4.62	7.84
d-gdb1	12	22	22	5	332	323	362	39.28	14.83	73.58
d-gdb2	12	26	26	6	364.7	349	408	42.47	17.47	88.02
d-gdb3	12	22	22	5	292.1	275	317	32.97	7.73	75.5
d-gdb4	11	19	19	4	289.8	287	311	11.72	6.97	18.65
d-gdb5	13	26	26	6	410.9	395	433	32.81	11.98	94.66
d-gdb6	12	22	22	5	336.3	310	392	26.58	10.72	63.59
d-gdb7	12	22	22	5	344.1	325	373	26.87	10.3	62.13
d-gdb8	27	46	46	10	-	-	-	209.56	203.29	219.05
d-gdb9	27	51	51	10	-	-	-	240.06	233.53	248.85
d-gdb10	12	25	25	4	292.7	275	313	12.23	8.62	18.51
d-gdb11	22	45	45	5	426.4	405	444	70.76	55.27	96.72
d-gdb12	13	23	23	7	-	-	-	91.73	88.32	93.35
d-gdb13	10	28	28	6	545.6	540	558	20.94	14.66	29.03
d-gdb14	7	21	21	5	101.9	100	103	8.87	6.26	12.49
d-gdb15	7	21	21	4	58.8	58	60	5.71	4.63	10.03
d-gdb16	8	28	28	5	127.2	127	129	10.33	7.38	15.52
d-gdb17	8	28	28	5	91	91	91	8.48	7.69	10.21
d-gdb18	9	36	36	5	167	166	168	14.18	11.08	21.53
d-gdb19	8	11	11	3	58.6	57	67	3.3	3.23	3.41
d-gdb20	11	22	22	4	121.7	121	124	9.74	6.42	17.03
d-gdb21	11	33	33	6	162.4	158	171	17.1	12.17	21.2
d-gdb22	11	44	44	8	204.8	202	210	37.34	18.21	56.83
d-gdb23	11	55	55	10	245.2	239	251	46.81	35.28	60.97

Genetik algoritma ile elde edilen sonuçlarda “d-gdb8”, “d-gdb9” ve “d-gdb12” test problemleri için uygun çözüm bulunamamıştır. Büyük olasılıkla kapasite kısıtı uygun çözüm bulunmasını engellemiştir. Genetik algoritma ile elde edilen çözümlerin

kalitesini göstermek için aynı test problemleri GAMS/CPLEX kullanılarak 1 saat çözüm süresi durdurma kriteri ile çözülmüştür. GAMS ile elde edilen çözüm, çözüm süresi, alt sınır (elde edilebilecek en iyi çözüm) ve aralık Çizelge 7.2’de verilmiştir.

**Çizelge 7.2.** Test problemlerinin GAMS/CPLEX kullanılarak 1 saat durdurma kriteri ile elde edilen sonuçlar

	V	E	R	m	Model Amaç Fonksiyonu Değeri	Çözüm Süresi (s)	Alt Sınır	Aralık (%)
d-kshs1	8	15	15	4	14661	1398.39	14661	0
d-kshs2	10	15	15	4	9863	1350.17	9863	0
d-kshs3	6	15	15	4	9320	422.77	9320	0
d-kshs4	8	15	15	4	11098	258.83	11098	0
d-kshs5	8	15	15	3	10957	88.41	10957	0
d-kshs6	9	15	15	3	10197	2.13	10197	0
d-gdb1	12	22	22	5	317	3600.25	261	17.67
d-gdb2	12	26	26	6	331	3600.33	291	12.08
d-gdb3	12	22	22	5	275	3600.25	233	15.27
d-gdb4	11	19	19	4	287	3600.08	272	5.23
d-gdb5	13	26	26	6	432	3601.06	316	26.85
d-gdb6	12	22	22	5	290	3600.3	260	10.34
d-gdb7	12	22	22	5	325	3600.25	262	19.38
d-gdb8	27	46	46	10		Çözülemedi		
d-gdb9	27	51	51	10		Çözülemedi		
d-gdb10	12	25	25	4	275	3600.31	254	7.64
d-gdb11	22	45	45	5		Çözülemedi		
d-gdb12	13	23	23	7	-	3600.38	-	-
d-gdb13	10	28	28	6	528	3600.11	509	3.6
d-gdb14	7	21	21	5	98	2022.66	98	0
d-gdb15	7	21	21	4	56	102.52	56	0
d-gdb16	8	28	28	5	125	3600.03	119	4.8
d-gdb17	8	28	28	5	91	3600.05	84	7.69
d-gdb18	9	36	36	5	158	30.81	158	0
d-gdb19	8	11	11	3	55	0.28	55	0
d-gdb20	11	22	22	4	121	3600.14	116	4.13
d-gdb21	11	33	33	6	154	3600.17	149	3.25
d-gdb22	11	44	44	8	202	3600.34	191	5.45
d-gdb23	11	55	55	10	241	3600.8	223	7.47

Çizelge 7.2’de aralık sütunu 1 saatlik çözüm süresi sonunda elde edilen amaç fonksiyonu değeri ile gevşetilmiş modelin amaç fonksiyonu değeri (alt sınır sütunu) arasındaki farkı göstermektedir. Dolayısıyla aralık değerinin 0 olduğu testler eniyi çözümün elde edildiğini göstermektedir. “d-gdb8”, “d-gdb9” ve “d-gdb11” test problemlerinde düğüm sayısı 20’nin üzerinde olduğundan dolayı bu problemler GAMS ile çözülmemiştir. Çizelge 7.2’den de görüleceği gibi “d-gdb12” problemi için 1 saatte çözüm bulunamamıştır. Daha sonra bu problem 3 saat durdurma kriteri ile çalıştırılmış fakat yine uygun çözüm bulunamamıştır.

GAMS ile elde edilen sonuçlar ve genetik algoritma çözümleri ile karşılaştırması Çizelge 7.3’de verilmiştir.

**Çizelge 7.3.** GAMS/CPLEX ile elde edilen sonuçlar ve genetik algoritma çözümleri ile karşılaştırması

Test Problemi	V	E	R	m	Model Amaç Fonksiyonu Değeri	Çözüm Süresi	GA ile Elde Edilen Çözüm	Çözüm Süresi	Amaç Fonksiyonu Farkı (%)	Çözüm Süresi Farkı (sn)
d-kshs1	8	15	15	4	14661	1398.39	14729	8.56	0.46	-1389.83
d-kshs2	10	15	15	4	9863	1350.17	9863	6.25	0.00	-1343.92
d-kshs3	6	15	15	4	9320	422.77	9320	4.47	0.00	-418.30
d-kshs4	8	15	15	4	11098	258.83	11098	6.84	0.00	-251.99
d-kshs5	8	15	15	3	10957	88.41	10957	5	0.00	-83.41
d-kshs6	9	15	15	3	10197	2.13	10197	5.6	0.00	3.48
d-gdb1	12	22	22	5	317	3600.25	323	21.35	1.86	-3578.90
d-gdb2	12	26	26	6	331	3600.33	349	31.47	5.16	-3568.86
d-gdb3	12	22	22	5	275	3600.25	275	75.5	0.00	-3524.75
d-gdb4	11	19	19	4	287	3600.08	287	8.99	0.00	-3591.09
d-gdb5	13	26	26	6	432	3601.06	395	17.25	-9.37	-3583.81
d-gdb6	12	22	22	5	290	3600.30	310	63.59	6.45	-3536.71
d-gdb7	12	22	22	5	325	3600.25	325	20.68	0.00	-3579.57
d-gdb8	27	46	46	10	Çözülmedi		-	210	-	-
d-gdb9	27	51	51	10	Çözülmedi		-	233.53	-	-
d-gdb10	12	25	25	4	275	3600.31	275	11.71	0.00	-3588.60
d-gdb11	22	45	45	5	Çözülmedi		405	63.36	100.00	-
d-gdb12	13	23	23	7	-	3600.38	-	90.47	-	-3509.91
d-gdb13	10	28	28	6	528	3600.11	540	26.07	2.22	-3574.04
d-gdb14	7	21	21	5	98	2022.66	100	10.67	2.00	-2011.99
d-gdb15	7	21	21	4	56	102.52	58	5.1	3.45	-97.42
d-gdb16	8	28	28	5	125	3600.03	127	8.62	1.57	-3591.41
d-gdb17	8	28	28	5	91	3600.05	91	8.4	0.00	-3591.65
d-gdb18	9	36	36	5	158	30.81	166	13.58	4.82	-17.23
d-gdb19	8	11	11	3	55	0.28	57	3.27	3.51	2.99
d-gdb20	11	22	22	4	121	3600.14	121	7.88	0.00	-3592.26
d-gdb21	11	33	33	6	154	3600.17	158	15.87	2.53	-3584.30
d-gdb22	11	44	44	8	202	3600.34	202	56.83	0.00	-3543.51
d-gdb23	11	55	55	10	241	3600.80	239	42.39	-0.84	-3558.41

Çizelge 7.3'te yer alan amaç fonksiyonu farkı sütunu genetik algoritma ile elde edilen çözümlerin model ile elde edilen çözümlerden ne kadar daha iyi (negatif) veya kötü (pozitif) olduğunu göstermektedir. Çözüm süresi farkı sütunu da genetik

algoritmanın çalışma süresi ile modelin çalışma süresi farkını göstermektedir. Bu değerin negatif olması genetik algoritmanın çözüm süresinin daha kısa olduğunu göstermektedir. Çizelge 7.3'ten görüleceği gibi GAMS/CPLEX ile 1 saatten daha kısa sürede en iyi çözümü bulunan problemlerde önerilen genetik algoritma ya en iyi çözümü bulmuş ya da küçük bir fark vardır. Hatta “d-gdb5” ve “d-gdb23” test problemlerinde genetik algoritma GAMS'ten daha iyi çözümler elde etmiştir.

Türetilen test problemleri ayrıca Yeni KARP'ın çözümü için önerilen DUA kullanılarak da çözülmüştür. DUA bütün gezilmesi gereken ayrıtları kapsayacak araç kapasitesini aşmayan turları belirler. Bu işlemi gerçekleştirirken elde kaç tane araç olduğunu kontrol etmez. Bu nedenle DUA ile elde edilen sonuçlar GAMS ve genetik algoritma ile elde edilen sonuçlarla karşılaştırılmamıştır. DUA'nın test problemleri çözülmesi ile elde edilen test sonuçları Çizelge 7.4'te verilmiştir. Ayrıca DUA ile elde edilen çözümlerin iyileştirilmesi için Frederickson'ın sezgiseli kullanılmıştır. DUA ile elde edilen araçların servis vereceği ayrıtlar kullanılarak Frederickson'ın sezgiseli ile bu ayrıtları kapsayan tur belirlenmiştir. Eğer elde edilen tur kapasite kısıtını aşmıyorsa ve DUA ile elde edilen turdan daha iyiyse Frederickson'ın sezgiseli ile elde edilen tur kullanılır.

Çizelge 7.4. DUA'nın test problemleri çözülmesi ile elde edilen test sonuçları

	m	Araç Kapasitesi	Kullanılan Araç Sayısı	İyileştirme Öncesi Toplam Tur Maliyeti	İyileştirme Sonrası Toplam Tur Maliyeti	En Büyük Tur Maliyeti	En Fazla Harcanan Kapasite Miktarı	Çözüm Zamanı (sn)
d-kshs1	4	377	5	17993	17542	4360	365	0
d-kshs2	4	388	6	13605	13438	2621	388	0.01
<i>d-kshs3</i>	4	382	4	9871	9843	3374	367	0
d-kshs4	4	393	6	13980	13980	3182	375	0
d-kshs5	3	372	4	13769	13669	4528	357	0
<i>d-kshs6</i>	3	323	3	11177	11177	5006	323	0
d-gdb1	5	12	7	450	413	78	11	0
d-gdb2	6	12	8	381	375	73	12	0.01
d-gdb3	5	12	7	375	373	74	12	0
d-gdb4	4	13	5	314	304	86	13	0.01
d-gdb5	6	12	8	478	475	80	11	0
d-gdb6	5	12	6	335	335	91	12	0
<i>d-gdb7</i>	5	12	5	359	335	84	12	0
d-gdb8	10	72	-	-	-	-	-	1.25
d-gdb9	10	76	-	-	-	-	-	1.21
<i>d-gdb10</i>	4	24	4	289	289	82	24	0
<i>d-gdb11</i>	5	122	5	413	413	107	118	1.11
d-gdb12	7	82	-	-	-	-	-	0
d-gdb13	6	105	7	548	548	138	97	0
<i>d-gdb14</i>	5	42	5	105	105	27	42	0.01
<i>d-gdb15</i>	4	71	4	58	58	22	63	0
<i>d-gdb16</i>	5	68	5	127	127	36	64	0
<i>d-gdb17</i>	5	84	5	97	95	24	82	0.01
<i>d-gdb18</i>	5	87	5	168	168	45	83	0
d-gdb19	3	57	4	65	65	25	56	0
<i>d-gdb20</i>	4	75	4	123	121	44	66	0
<i>d-gdb21</i>	6	72	6	158	158	47	71	0
<i>d-gdb22</i>	8	65	8	203	203	37	64	0.01
<i>d-gdb23</i>	10	71	9	231	231	38	71	0

Çizelge 7.4'ten görüleceği üzere yine “d-gdb8”, “d-gdb9” ve “d-gdb12” test problemlerinde uygun çözüm bulunamamıştır. DUA ile elde edilen çözümlerde eğer kullanılan araç sayısı eldeki araç sayısına eşit veya küçükse bu aynı zamanda test probleminin de bir uygun çözümdür. Bu durumu sağlayan testler italik olarak

yazılmıştır. DUA, iki aşamalı yapıcı bir sezgisel olsa da “d-gdb23” problemi için çok kısa sürede hem GAMS’ten hem de genetik algorithmadan daha iyi bir çözüm bulmuştur.

“d-gdb8”, “d-gdb9” ve “d-gdb12” test problemlerinin uygun çözümü olmadığından araçların kapasite kısıtı denklem (7.2)’deki gibi arttırılarak bu 3 test problemi genetik algoritma, GAMS ve DUA ile yeniden çözülmüştür.

$$Q = \frac{\left[ \left( \sum_{\forall(i,j) \in E} e_{ij} \right) \cdot 3 + \sum_{\forall(i,j) \in R} q_{ij} \right]}{m} \quad (7.2)$$

Arttırılmış kapasite değerleri ile değiştirilmiş test problemleri önerilen genetik algoritmayla 10 sefer çözülmüştür. Her çözüme ait en iyi amaç fonksiyonu değeri ve çözüm süresi Ekler bölümünde verilmiştir. Elde edilen çözümlere ait özet çizelge ise Çizelge 7.5’te verilmiştir.

**Çizelge 7.5.** Kapasite artırımını sonucu önerilen genetik algoritma ile elde edilen çözümlerin amaç fonksiyonu ve çözüm süresi açısından ortalama, en küçük ve en büyük değerleri

Test Problemi	Ortalama Amaç Fonksiyonu Değeri	En Küçük Amaç Fonksiyonu Değeri	En Büyük Amaç Fonksiyonu Değeri	Ortalama Çözüm Süresi	En Kısa Çözüm Süresi	En Uzun Çözüm Süresi
d-gdb8	397.20	372.00	430.00	67.87	42.59	110.59
d-gdb9	365.50	342.00	394.00	80.40	47.83	109.16
d-gdb12	521.20	474.00	556.00	16.96	10.02	21.71

“d-gdb12” test problemi GAMS kullanılarak 1 saat kısıtıyla çözülmüş ve 498 sonucu elde edilmiştir. Elde edilebilecek en iyi amaç fonksiyonu değeri 336, aralık ise 0,33’tür. Son olarak yeni kapasite değerleri ile üç test problemi DUA kullanılarak çözülmüştür. Elde edilen sonuçlar Çizelge 7.6’da verilmiştir.

**Çizelge 7.6.** Kapasitesi arttırılmış test problemlerinin DUA ile çözülmesiyle elde edilen sonuçlar

	m	Araç Kapasitesi	Kullanılan Araç Sayısı	İyileştirme Öncesi Toplam Tur Maliyeti	İyileştirme Sonrası Toplam Tur Maliyeti	En Büyük Tur Maliyeti	En Fazla Harcanan Kapasite Miktarı	Çözüm Zamanı (sn)
<i>d-gdb8</i>	10	107	8	378	347	58	104	1.24
<i>d-gdb9</i>	10	114	9	333	323	54	109	1.20
<i>d-gdb12</i>	5	121	8	639	611	96	112	0

“d-gdb8” ve “d-gdb9” problemlerindeki araç kapasiteleri bol olduğundan dolayı DUA ile hem uygun hem de iyi sonuçlar elde edilmiştir.

Büyük boyutlu problemlerin de çözülebildiğinin gösterilmesi için DUA kullanılarak “d-eglese” isimli test problemleri çözülmüştür. Bu test problemlerinin çözümü sonucu elde edilen sonuçlar



**Çizelge 7.7.** “d-eglese” test problemlerinin DUA kullanılarak çözülmesi sonucu elde edilen sonuçlar

	V	E	R	m	Kullanılan Araç Sayısı	İyileştirme Öncesi Toplam Tur Maliyeti	İyileştirme Sonrası Toplam Tur Maliyeti	En Büyük Tur Maliyeti	En Fazla Harcanan Kapasite Miktarı	Çözüm Zamanı (sn)
d-egl-e1-A	77	98	51	5	4	5470	4744	1549	1962	0.07
d-egl-e1-B	77	98	51	7	5	5684	5113	1329	1589	0.07
d-egl-e1-C	77	98	51	10	10	7530	6535	863	1083	0.07
d-egl-e2-A	77	98	72	7	8	8197	7634	1328	1646	0.10
d-egl-e2-B	77	98	72	10	12	9839	9107	967	1073	0.10
d-egl-e2-C	77	98	72	14	10	9015	8132	1134	1348	0.10
d-egl-e3-A	77	98	87	8	13	13656	10907	1173	1459	0.13
d-egl-e3-B	77	98	87	12	11	12590	10219	1442	1605	0.13
d-egl-e3-C	77	98	87	17	10	11728	9143	1446	1679	0.13
d-egl-e4-A	77	98	98	9	16	15626	12081	1024	1315	0.12
d-egl-e4-B	77	98	98	14	15	14808	11349	1024	1372	0.14
d-egl-e4-C	77	98	98	19	8	11644	9125	1508	2009	0.14
d-egl-s1-A	140	190	75	7	4	7974	5946	1897	2396	0.27
d-egl-s1-B	140	190	75	10	9	10214	7579	1340	1498	0.27
d-egl-s1-C	140	190	75	14	6	9192	6832	1565	1892	0.27
d-egl-s2-A	140	190	147	14	13	20317	13361	1629	1966	0.67
d-egl-s2-B	140	190	147	20	8	18060	11513	2073	2642	0.67
d-egl-s2-C	140	190	147	27	12	20056	12821	1662	2036	0.67
d-egl-s3-A	140	190	159	15	8	18796	14005	2718	3278	0.76
d-egl-s3-B	140	190	159	22	9	19308	12223	2376	2936	0.76
d-egl-s3-C	140	190	159	29	15	22270	14471	1625	1930	0.80
d-egl-s4-A	140	190	190	19	8	19760	12070	2574	3275	0.93
d-egl-s4-B	140	190	190	27	12	21744	13999	1906	2393	0.90
d-egl-s4-C	140	190	190	35	19	25550	16927	1467	1814	0.89

“d-eglese” test problemlerindeki düğüm sayısı ve ayrıt sayısı fazla olmasına rağmen DUA kullanılarak bu test problemleri 1 sn’in altında çözülmüştür. Ayrıca DUA’nın çözümünü iyileştirmede kullanılan Frederickson’ın algoritması tur maliyetlerini oldukça aşağıya çektiği gözlenmiştir.

## 8. SONUÇ VE ÖNERİLER

Bu çalışmada klasik KARP'tan farklı olarak araçların ayrıtlara servis verilmesi sırasında kullandığı kapasitenin yanı sıra geçiş sırasında kullandığı kapasitenin de göz önüne alındığı yeni bir KARP türü tanımlanmıştır. Bu problem ile gezgin robotlardaki çok robotlu algılayıcı tabanlı ortam kapsanması problemi ile karşılaşılmıştır. Çalışmada öncelikle ortam kapsanması probleminin KARP şeklinde tanımlanabileceği gösterilmiştir. Daha sonra gezgin robotun ortamı tararken harcadığı enerji ile birlikte gezinme sırasında harcadığı enerjinin göz ardı edilemeyecek kadar büyük olması nedeniyle yeni bir KARP türünün tanımlanması gerektiği ifade edilmiştir. Çalışmada yeni KARP'ın kesin çözümünü bulabilmek için bir matematiksel model önerilmiş, büyük boyutlu problemlerde etkin çözümleri bulmak için de genetik algoritma ve Ulusoy algoritması tabanlı iki farklı çözüm yöntemi geliştirilmiştir. Önerilen yaklaşımlar var olan test problemlerinden değiştirilerek oluşturulan yeni test problemleri üzerinde denenmiş ve başarıları gösterilmiştir.

Bu çalışmada değinilen çok robotlu algılayıcı tabanlı ortam kapsanması probleminin uygulanması için çok erkinli bir mimari geliştirilmiş ve bu mimarinin planlama fazında Değiştirilmiş Ulusoy Algoritması (DUA) kullanılmıştır. Bu algoritma kullanılarak çok robotlu gezgin robot uygulamaları için hem gerçek ortamda hem MobileSim simülasyon ortamında deneyler gerçekleştirilmiştir. Ayrıca önerilen mimari ve planlama algoritması, 90 düğümlü bir serim üzerinde 9 robota kadar test edilerek başarısı gösterilmiş, robot sayısındaki değişime göre elde edilen tur uzunluğu incelenerek bu tür çalışmalarda kaç robotla çalışılması gerektiğinin nasıl belirleneceğine dair bir yöntem gösterilmiştir.

Tanımlanan yeni KARP türü için önerilen matematiksel model, probleme dair eniyi çözümlerin elde edilmesi açısından önemli bir katkıdır. Ancak tıpkı daha önce KARP için önerilmiş matematiksel modellerde olduğu gibi bu modelin en önemli sıkıntısı uygun olmayan alt turları engellemek için yazılan kısıtların üstel sayıda artmasıdır. Bu nedenle sadece küçük boyutlu problemler için GAMS/CPLEX programı kullanılarak eniyi çözümler bulunabilmiştir. Ancak günümüzde KARP için de büyük

boyutlu problemlerin eniyi çözümünü makul sürede bulabilen bir matematiksel modelin geliştirilmemiş olduğu unutulmamalıdır. Matematiksel modelin 15 düğümden daha büyük problemleri çözememesi nedeniyle çalışmada genetik algoritma ve DUA sezgiselleri önerilmiş, genetik algoritma ile elde edilen sonuçlar matematiksel model sonuçları karşılaştırılmıştır. Bu karşılaştırma sonrasında elde edilen çözümlerin eniyi çözüme oldukça yakın olduğu gözlenmiştir. Genetik algoritmanın çözüm süresi de 50 saniye mertebesinde gayet kabul edilebilir düzeydedir. Önerilen DUA sezgiseli ise bütün test problemlerinde uygun çözüm üretmese de çok kısa sürelerde çözüm bulduğundan tercih edilebilir bir algoritma olduğunu göstermiştir.

İlginç bir sonuç da önerilen genetik algoritma ve DUA arasındaki farktır. Araç kapasite değerinin sıkı olduğu durumlarda problem için uygun çözüm bile bulmak oldukça zordur. Bu tür durumlarda genetik algoritma uygun çözüm bulup, bu çözümü iyileştirebilmektedir. Diğer bir yönden araç kapasite değerinin sıkı olmadığı durumlarda ise DUA eldeki araç sayısından daha az araçla bütün gezilmesi gereken ayrıtlara servis verecek araç turlarını belirleyebilmektedir.

Sonuç olarak çalışma yeni bir problem tanımı ve çözüm yöntemleri önererek literatüre katkıda bulunmaktadır. Bundan sonraki çalışmaların heterojen filo ve alternatif depoların olması hali için yeni algoritmaların geliştirilmesi düşünülmektedir.

**KAYNAKLAR DİZİNİ**

- Acar, E.U., Choset, H. and Lee, J.Y., 2006, Sensor-based coverage with extended range detectors, *IEEE Transactions on Robotics*, vol. 22 (1), pp. 189-198.
- Ahr, D. and Reinelt, G., 2006, A tabu search algorithm for the min-max k-Chinese postman problem. *Computers and Operations Research*, vol. 33(12), pp. 3403–3422.
- Ahr, D., 2004, Contributions to multiple postmen problems, PhD thesis, University of Heidelberg, Germany, p. 222.
- Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., 1993, *Network Flows: Theory, Algorithms, and Applications*, Printice Hall, New Jersey.
- Amado L., and Mourão M.C., 2005, Heuristic method for a mixed capacitated arc routing problem: A refuse collection application, *European Journal of Operational Research*, vol. 160(1), pp. 139–153.
- Amaya, A., Langevin, A. and Trépanier, M., 2007, The capacitated arc routing problem with refill points, *Operations Research Letters*, vol. 35(1), pp. 45–53.
- Amberg, A., Domschke, W. and Voß. S., 2000, Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees, *European Journal of Operational Research*, vol. 124, pp. 360–376.
- Aragao, M.P., Longo, H. and Uchoa, E., 2006, Solving capacitated arc routing problem using a transformation to the CVRP, *Computers and Operations Research*, vol. 33(6), pp. 1823–1837.
- Archetti, C., Feillet, D., Hertz, A. Speranza, M.G., 2009, The undirected capacitated arc routing problem with profits, *Computers & Operations Research*, (in press).
- Arkin, E.M., Fekete, S.P. and Mitchell, J.S.B., 2000, Approximation algorithms for lawn mowing and milling, *Computational Geometry*, vol. 17, pp. 25-50.

**KAYNAKLAR DİZİNİ (devam)**

- Arkin, R.C., 2002, Behavior-Based Robotics, John Wiley and Sons Press, England.
- Assad A. and Golden, B.L., 1995, Arc Routing Methods and Applications. In Ball, M.G., Magnanti, T.L., Monma, C.L. and Nemhauser, G.L. (eds.), Network Routing, Volume 8 of Handbooks in Operations Research and Management Science, chapter 5, pp. 375-483, Elsevier.
- Baldacci R. and Maniezzo, V., 2006, Exact methods based on node routing formulations for undirected arc routing problems. *Networks*, vol. 47(1), pp. 52–60.
- Ball, M., Bodin, L., Levy, L. and Sniezek, J., 2001, Capacitated arc routing problem with vehicle-site dependencies: The Philadelphia experience. In: Toth, P. and Vigo, D., The vehicle routing problem, Society for Industrial and Applied Mathematics, Philadelphia, Chapter 11, pp. 287-330.
- Belenguer, J.M. and Benavent, E., 1998, The capacitated arc routing problem: Valid inequalities and facets, *Computational Optimization and Applications*, vol. 10, pp. 165–187.
- Belenguer, J.M. and Benavent, E., 2003, A cutting plane algorithm for the capacitated arc routing problem, *Computers and Operations Research*, vol. 30(5), pp. 705–728.
- Belenguer, J.M., Benavent, E., Lacomme, P. and Prins, C., 2006, Lower and upper bounds for the mixed capacitated arc routing problem, *Computers and Operations Research*, vol. 33(12), pp. 3363–3383.
- Benavent, E., Campos, V., Corberán, A. and Mota, E., 1990, The Capacitated Arc Routing Problem: A Heuristic Algorithm, *Qüestió*, vol. 14(1-3), pp. 107-122,
- Benavent, E., Corberán, A., Sanchis, J.M., 2000, Linear programming based methods for solving arc routing problems. In: Dror, M. (ed.), *Arc Routing: Theory, Solutions and Applications*, Boston, Kluwer Academic Publishers, pp. 231-275.

**KAYNAKLAR DİZİNİ (devam)**

- Beullens, P., Muyldermans, L., Cattrysse, D., Van Oudheusden, D., 2003, A guided local search heuristic for the capacitated arc routing problem, *European Journal of Operational Research*, vol. 147, pp. 629-643.
- Bodin, L. and Sniezek, J., 2006, Using mixed integer programming for solving the capacitated arc routing problem with vehicle/site dependencies with an application to the routing of residential sanitation collection vehicles, *Annals of Operations Research*, vol. 144(1), pp. 33–58.
- Brandao, J., Eglese, R., 2008, A deterministic tabu search algorithm for the capacitated arc routing problem, *Computers & Operations Research*, vol. 35, pp. 1112-1126.
- Cabral, E.A., Gendreau, M., Ghiani, G., Laporte, G., 2004, Solving the Hierarchical Chinese Postman Problem as a Rural Postman Problem, *European Journal of Operational Research*, vol. 155, pp. 44-50.
- Cattrysse, D., Lotan, T., Muyldermans, L., Van Oudheusden, D., 2002, Districting for salt spreading operations, *European Journal of Operational Research*, vol. 139, pp. 521–532.
- Cattrysse, D., Muyldermans, L., Van Oudheusden, D., 2003, District design for arc routing applications, *Journal of the Operational Research Society*, vol. 54, pp. 209–221.
- Choset, H., 2000, Coverage of known spaces: The boustrophedon cellular decomposition, *Autonomous Robots*, vol. 9, 247–253.
- Choset, H., 2001, Coverage for robotics – A survey of recent results, 2001, *Annals of Mathematics and Artificial Intelligence*, vol.31, pp.113–126.
- Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E. and Thrun, S., 2005, *Principles of Robot Motion Theory, Algorithms and Implementations*, MIT Press, Cambridge, MA.

**KAYNAKLAR DİZİNİ (devam)**

- Christiansen, C.H., Lysgaard, J. and Wøhlk, S., 2009, A Branch-and-Price Algorithm for the Capacitated Arc Routing Problem with Stochastic Demands, *Operations Research Letters*, (in press).
- Chu, F., Labadi, N. and Prins, C., 2003, Lower bounds for the periodic capacitated arc routing problem, *The Odysseus 2003 Conference*.
- Chu, F., Labadi, N. and Prins, C., 2005, Heuristics for the periodic capacitated arc routing problem, *Journal of Intelligent Manufacturing*, vol. 16, pp. 243–251.
- Chu, F., Labadi, N. and Prins, C., 2006, A scatter search for the periodic capacitated arc routing problem, *European Journal of Operational Research*, vol. 169(2), pp. 586–605.
- Çınar, E., Parlaktuna, O., Yazıcı, A., 2008, Küçük Ortamların Tam Kapsanması Amacıyla Genelleştirilmiş Voronoi Diyagramı Modellemesi, *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, 19-21 Haziran 2008, Isparta, Türkiye.
- Cook, W. and Rohe, A., 1999, Computing minimum weight perfect matchings, *INFORMS Journal on Computing*, vol. 11(2), pp. 138-148.
- Corberán, A., and Sanchis, J.M., 1994, A polyhedral approach to the rural postman problem, *European Journal of Operational Research*, vol. 79(1), pp. 95-114.
- Corberán, A., and Sanchis, J.M., 1998, The General Routing Problem polyhedron: facets from the RPP and GTSP polyhedra. *European Journal of Operational Research*, vol. 108(3), pp. 538-550.
- Córdoba, P.F., Raffi, L.M.G. and Sanchis, J.M., 1998, A Heuristic Algorithm based on Monte Carlo Methods for the Rural Postman Problem, *Computers & Operations Research*, vol. 25, no. 12, pp. 1097–1106.
- CPLEX Optimization Inc., CPLEX Version 11.0, 2007.

### KAYNAKLAR DİZİNİ (devam)

- Deng, X., Zhu, Z., Yang, Y., Li, X., Tian, Y., Xia, M., 2007, A Genetic Algorithm for the Capacitated Arc Routing Problem, Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, pp. 1551-1556.
- Doerner, K.F., Hartl, R.F., Maniezzo, V. and Reimann, M., 2003, An ant system metaheuristic for the capacitated arc routing problem, The 5th Metaheuristics International Conference (MIC2003), Kyoto, pp. 1-6.
- Doerner, K.F., Hartl, R.F., Maniezzo, V. and Reimann, M., 2004, Applying Ant Colony Optimization to the Capacitated Arc Routing Problem, Lecture Notes in Computer Science, vol. 3172, pp. 420–421.
- Dror, M. (ed), 2000, Arc Routing Theory, Solution and Application, Boston, Kluwer Academic Publishers.
- Dror, M. and Langevin, A., 2000, Transformation and exact node routing solutions by column generation. In: Dror, M. (ed.), Arc Routing: Theory, Solutions and Applications, Boston, Kluwer Academic Publishers, pp. 277-326.
- Dror, M. and Trudeau, P., 1989, Savings by split delivery routing, Transportation Science, vol. 23, pp. 141-145.
- Dror, M. and Trudeau, P., 1990, Split delivery routing, Naval Research Logistics, vol. 37, pp. 383-402.
- Dror, M., Stern, H.I., Trudeau, P., 1987, Postman Tour on a Graph with Precedence Relation On Arcs, Networks, vol.17, pp. 283–294.
- Edmonds, J. and Johnson, E.L., 1973, Matching, Euler Tours and the Chinese postman, Mathematical Programming, vol. 5(1), pp. 88–124.
- Edmonds, J., 1965, Paths, Trees and Flowers, Canadian Journal of Mathematics, vol. 17, pp. 449–467.



**KAYNAKLAR DİZİNİ (devam)**

- Eiselt, H.A., 2000, A historical perspective on arc routing. In: Dror, M. (ed.), Arc Routing: Theory, Solutions and Applications, Boston, Kluwer Academic Publishers, pp. 1-16.
- Eiselt, H.A., Gendreau, M., Laporte, G., (1995), Arc Routing Problems, Part I: The Chinese Postman Problem, Operations Research, vol. 43(2), pp. 231–242 (a).
- Eiselt, H.A., Gendreau, M., Laporte, G., (1995), Arc Routing Problems, Part II: The Rural Postman Problem, Operations Research, vol. 43(3), pp. 39–414 (b).
- Eskişehir Osmangazi Üniversitesi, 2009, Yapay Zeka ve Robotik Laboratuvarı, [www.ai-robotlab.ogu.edu.tr](http://www.ai-robotlab.ogu.edu.tr).
- Euler, L., 1741, Solvatio Problematis ad Geometriam Sitvs Pertinentis, Commentarii Academiae Scientiarum Imperialis Petropolitanae, 8: 128–140, Opera Omnia. Series 1, Volume 7, pp. 1–10. Internet: <http://www.eulerarchive.com/>
- Filippi C. and Pia, A.D., 2006, A variable neighborhood descent algorithm for a real waste collection problem with mobile depots, International Transactions in Operational Research, vol. 13, pp. 125–141.
- Fleury, G., Lacomme, P. and Prins, C., 2005, Stochastic capacitated arc routing problems, Research Report LIMOS/RR-05-12, 2005.
- Fleury, G., Lacomme, P. and Prins, C., and Ramdane-Chérif, W., 2005, Improving robustness of solutions to arc routing problems, Journal of the Operational Research Society, vol. 56, pp. 526–538.
- Fleury, G., Lacomme, P., Prins, C. and Ramdana-Chérif, W., 2002, Robustness evaluation of solutions for the capacitated arc routing problem, AI Simulation and Planning in High Autonomy Systems Conference, ISBN 1-56555-242-3, pp. 290–295.

**KAYNAKLAR DİZİNİ (devam)**

- Fleury, G., Lacomme, P., Prins, C. and Sevaux, M., 2008, A Bi-Objective Stochastic Approach for the Stochastic CARP, Research Report LIMOS/ RR-08-06.
- Fleury, G., Lacomme, P., Prins, C., 2004, Evolutionary algorithms for stochastic arc routing problems, Lecture Notes in Computer Science, vol. 3005, pp. 501–512.
- Frederickson, G.N., 1979, Approximation Algorithms for Some Postman Problems, Journal of Association for Computing Machinery, vol. 26(3), pp. 538–554.
- Frederickson, G.N., Hecht, M.S. and Kim, C.E., 1978, Approximation algorithms for some routing problems, SIAM Journal on Computing, vol. 7(2), pp. 178-193.
- Gen, M., Cheng, R., 1997, Genetic Algorithms and Engineering Design, John Wiley & Sons, New York.
- Gendreau, M., Potvin, J.Y. and Tagmouti, M., 2007, Arc routing problems with time dependent service costs, European Journal of Operational Research, vol. 181(1), pp. 30– 39.
- Ghiani G. and Laporte, G., 2000, A branch and cut algorithm for the Undirected Rural Postman Problem, Mathematical Programming, vol. 87(3), pp. 467-481.
- Ghiani, G., Guerriero, F., Laporte G. and Musmanno, R., 2004, Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and length restrictions, Journal of Mathematical Modelling and Algorithms, vol. 3, pp. 209–223.
- Ghiani, G., Improta, G. and Laporte, G., 2001, The capacitated arc routing problem with intermediate facilities, Networks, vol. 37(3), pp. 134–143, 2001.
- Ghiani, G., Laganà D., Musmanno, R., 2006, A constructive heuristic for the Undirected Rural Postman Problem, Computers & Operations Research, vol. 33, pp. 3450–3457.

**KAYNAKLAR DİZİNİ (devam)**

- Glover, F.W. and Kochenberger, G.A. (eds.), 2003, Handbook of metaheuristics, Kluwer Academic Publishers.
- Goldberg, D.E., 1989, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, Reading, Massachusetts.
- Golden B.L., DeArmon, J.S., and Baker, E., 1983, Computational experiments with algorithms for a class of routing problems, Computers and Operations Research, vol. 10(1), pp. 47–59.
- Golden, B.L. and Wong, R., 1981, Capacitated arc routing problems, Networks, vol. 11, pp. 305-315.
- Gonzalez, E., Alvarez, O., Diaz, Y., Parra, C. and Bustacara, C., 2005, BSA: A complete coverage algorithm, Robotics and Automation, ICRA 2005, Proceedings of the 2005 IEEE International Conference on 18-22 April 2005, pp. 2040 – 2044.
- Gouveia, L., Mourã, M.C. and Pinto, L.S., 2009, Lower bounds for the mixed capacitated arc routing problem, Computer and Operations Research, (in press).
- Greistorfer, P., 2003, A tabu scatter search metaheuristic for the arc routing problem, Computers and Industrial Engineering, vol. 44(2), pp. 249–266.
- Gribkovskaia, I., Halskau, O., Laporte, G., 2007, The Bridges of Königsberg—A Historical Perspective, Networks, vol. 49, pp. 199-203.
- Gross, J.L. and Yellen, J. (eds.), 2004, Handbook of Graph Theory, CRC Press, Boca Raton, 1115 p.
- Gutin, G. and Punnen, A.P. (eds.), 2004, The Travelling salesman problem and Its variation, Kluwer Academic Publishers, Dordrecht.

**KAYNAKLAR DİZİNİ (devam)**

- Hertz A. and Mittaz, M., 2001, A variable neighborhood descent algorithm for the undirected capacitated arc routing problem, *Transportation Science*, vol. 35(4), pp. 425–434.
- Hertz, A., Laporte, G. and Hugo. P.N., 1999, Improvement procedures for the undirected rural postman problem, *INFORMS Journal on Computing*, vol. 11(1), pp. 53-62.
- Hertz, A., Laporte, G. and Mittaz, M., 2000, A tabu search heuristic for the capacitated arc routing problem, *Operations Research*, vol. 48, pp. 129–135.
- Hertz, A., Mittaz, M., 2000, Heuristic algorithms. In: Dror, M. (ed.), *Arc Routing: Theory, Solutions and Applications*, Boston, Kluwer Academic Publishers, pp. 327-386.
- Hertz. A., 2005, Recent trends in arc routing. In: Golombic, M.C. and Hartman, I.B.A. (eds.), *Graph theory, Combinatorics and algorithms: Operations Research/Computer Science Interfaces Series*, vol. 34, Springer, New York, pp. 215-236.
- Hirabayashi, R., Nishida, N. and Saruwatari, 1992, Y., Tour construction algorithm for the capacitated arc routing problem, *Asia-Pacific Journal of Operational Research*, vol. 9, pp. 155–175 (a).
- Hirabayashi, R. Nishida, N. and Saruwatari, Y., 1992, Node Duplication Lower Bounds for the Capacitated Arc Routing Problems, *Journal of Operations Research Society of Japan*, vol. 35(2), pp. 119–133 (b).
- Holland, J.H., 1975, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Michigan.
- Johnson, E.L. and Wøhlk, S., 2009, Solving The Capacitated Arc Routing Problem With Time Windows Using Column Generation, Working Paper L-2008-09, Centre for Operations Research Applications in Logistics.

**KAYNAKLAR DİZİNİ (devam)**

- Kolmogorov, V., 2009, Blossom V: A new implementation of a minimum cost perfect matching algorithm, *Mathematical Programming Computation*, vol. 1(1), pp. 43–67.
- Kurabayashi, D., Ota, J., Arai, T. and Yoshida E., 1996, Cooperative sweeping by multiple mobile robots, *Proceedings of the IEEE International Conference on Robotics Automation*, Minneapolis, Minnesota, April 1996, vol. 3, pp. 1744–1749.
- Labadi, N., Prins, C. and Reghioui M., 2008, An Evolutionary Algorithm with Distance Measure for the Split Delivery Capacitated Arc Routing Problem. In: Cotta C. and Hemert J.V. (eds.), *Recent Advances in Evolutionary Computation for Combinatorial Optimization*, vol. 153, pp. 275–294.
- Labadi, N., Prins, C. and Reghioui, M., 2007, A Memetic algorithm with population management for the capacitated arc routing problem with split demands, *Proceedings of the 37th International Conference on Computers and Industrial Engineering*, Alexandria, Egypt.
- Lacomme, P., Christian, P. and Ramdane-Chérif, W., 2001, A genetic algorithm for the capacitated arc routing problem and its extensions, *Lecture Notes in Computer Science*, vol. 2037, pp. 473–483.
- Lacomme, P., Christian, P. and Ramdane-Cherif, W., 2003, Multiobjective capacitated arc routing problem, *Lecture Notes in Computer Science*, vol. 2632, pp. 550–564.
- Lacomme, P., Christian, P. and Ramdane-Cherif, W., 2004, Competitive memetic algorithms for arc routing problems, *Annals of Operations Research*, vol. 131, pp. 159–185.
- Lacomme, P., Christian, P. and Ramdane-Chérif, W., 2005, Evolutionary algorithms for periodic arc routing problems, *European Journal of Operational Research*, vol. 165, pp. 535–553.

**KAYNAKLAR DİZİNİ (devam)**

- Lacomme, P., Christian, P. and Ramdane-Cherif, W., 2006, A genetic algorithm for a bi-objective capacitated arc routing problem, *Computers and Operations Research*, vol. 33(12), pp. 3473–3493.
- Lacomme, P., Prins, C. and Ramdane-Chérif, W., 2001, An integer linear model for general arc routing problems, 13th European Simulation Symposium, *Simulation in Industry*, Marseille, France, pp. 427-431, October 18-20, ISBN 90-77039-02-3.
- Lacomme, P., Prins, C. and Ramdane-Chérif, W., 2002, General Arc Routing Problems Solved by a Cutting Plane Algorithm and a Genetic Algorithm, 15th Triennial World Congress of the International Federation of Automatic Control (IFAC 2002), Barcelona, pp. 21-26.
- Lacomme, P., Prins, C. and Tanguy, A., 2004, First competitive ant colony scheme for the CARP, *Lecture Notes in Computer Science*, vol. 3172, pp. 426–427.
- Latimer, D., Srinivasa, S., Sonne, V.L.S.S., Choset, H. and Hurst, A., 2002, Towards sensor based coverage with robot teams, *Proceedings of the IEEE International Conference on Robotics & Automation*, Washington DC, May 2002, pp. pp. 961-967.
- Lenstra J.K. and Rinnooy Kan A.H.G., 1976, On General Routing Problems, *Networks*, vol. 6(3), pp. 273–280, 1976.
- Letchford, A.N., 1997, New inequalities for the general routing problem, *European Journal of Operational Research*, vol. 96(2), pp. 317-322.
- Letchford, A.N., 1999, The general routing polyhedron: A unifying framework, *European Journal of Operational Research*, vol. 112(1), pp. 122-133, 1999.
- Maniezzo, V., 2004, Algorithms for large directed CARP instances: urban solid waste collection operational support, Technical Report UBLCS-2004-16, University of Bologna, Bologna.

**KAYNAKLAR DİZİNİ (devam)**

- Martínez, F.J.Z., 2003, Postman problems on mixed graphs, PhD thesis, University of Waterloo, 176 p.
- Mei, Y., Tang and Yao, X., 2009, A global repair operator for capacitated arc routing problem, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 39(3), pp. 723-734.
- Mei, Y., Yung-Hsiang, Y., Charlie, H.Y. and George, L.C.S., 2006, Deployment of mobile robots with energy and timing constraints, *IEEE Transactions on Robotics and Automation*, vol. 22(3), pp. 507- 522.
- Minieka, E., 1979, The Chinese postman problem for mixed networks, *Management Science*, vol. 25(7), pp. 643–648.
- Mullaseril, P.A., 1997, Capacitated Rural Postman Problem with Time Windows and Split Delivery, PhD thesis, MIS Department, University of Arizona, Tucson, Arizona 85721, 1997.
- Mullaseril, P.A., Dror, M., and Leung, J., 1997, Split delivery routing heuristics in livestock feed distribution. *Journal of the Operational Research Society*, vol. 48, pp. 107-116.
- Murata, T. and Ishibuchi, H., 1994, Performance evaluation of genetic algorithms for flow shop scheduling problems, *Proceedings of the First IEEE Conference on Evolutionary Computation*, vol. 2, pp. 812-817.
- Murphy, R. R., 2000, *Introduction to AI Robotics*, MIT Press, London, England.
- Orloff, C.S., 1974, A Fundamental Problem in Vehicle Routing, *Networks*, vol. 4, pp. 35-64.
- Özkan, M., 2007, Farklı özelliklere sahip gezgin robot grubunun dağıtık kontrolü, Eskişehir Osmangazi Üniversitesi, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Doktora Tezi.

**KAYNAKLAR DİZİNİ (devam)**

- Papadimitriou, C.H., 1976, On the Complexity of Edge Traversing, *Journal of the Association for Computing Machinery*, vol. 23(3), pp. 544-554.
- Pearn, W.L. and Wang, K.H., 2003, On the Maximum Benefit Chinese Postman Problem, *Omega*, vol. 31, pp. 269 – 273.
- Pearn, W.L. and Wu, T.C., 1995, Algorithms for the Rural Postman Problem, *Computers & Operations Research*, vol. 22, no. 8, pp. 819–828.
- Pearn, W.L., Assad, A. and Golden, B.L., 1987, Transforming Arc Routing into Node Routing Problems, *Computers & Operations Research*, vol. 14(4), pp. 285-288.
- Polacek, M., Doerner, K.F., Hartl, R. and Maniezzo, V., 2008, A variable neighborhood search for the capacitated arc routing problem with intermediate facilities, *Journal of Heuristics*, vol. 14, pp. 405-423.
- Prassler, E., Ritter, A., Schaeffer, C. and Fiorini, P., 2000, A short history of cleaning robots, *Autonomous Robots*, vol. 9, pp. 211-226.
- Prins, C., Labadi, N. and Reghioui, M., 2009, Tour splitting algorithms for vehicle routing problems, *International Journal of Production Research*, vol. 47(2), pp. 507-535.
- Ramdane-Cherif, W., 2006, Evolutionary algorithms for capacitated arc routing problems with time windows, *12th IFAC Symposium on Information Control Problems in Manufacturing*, Ecole des Mines, Saint Etienne, France.
- Reghioui, M., Prins, C. and Labadi, N., 2007, GRASP with path relinking for the capacitated arc routing problem with time windows, *Lecture Notes in Computer Science*, vol. 4448, pp. 722–731, 2007.
- Ryerson, A.E.F. and Zhang, Q., 2007, Vehicle path planning for complete field coverage using genetic algorithms, *Agricultural Engineering International: the CIGR Ejournal*. Manuscript, ATOE 07 014, Vol. 9, July, 2007.



**KAYNAKLAR DİZİNİ (devam)**

- Santos, L., Coutinho-Rodrigues, J. and Current, J.R., 2009, An improved heuristic for the capacitated arc routing problem, *Computers and Operations Research*, vol. 36(9), pp. 2632-2637.
- Siegwart, R. and Nourbakhsh, I.R., 2004, *Introduction to Autonomous Mobile Robots*, MIT Press, London, England.
- Tagmouti, M., Gendreau, M., Potvin, J.Y., 2008, A variable neighbourhood descent for arc routing problems with time dependent service cost, *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation*, Technical Report (CIRRELT-2008-27), Montréal, Canada.
- Tagmouti, M., Gendreau, M. and Potvin, J.Y., 2009, A dynamic capacitated arc routing problem with time-dependent service costs, *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation*, Technical Report (CIRRELT-2009-19), Montréal, Canada.
- Tan, G., Cui, X. and Zhang, Y., 2005, Chinese Postman Problem in Stochastic Networks, *Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services*, pp. 78-78.
- Toth P. and Vigo, D. (eds.), 2002, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications.
- Treval, C., Ota, J. and Arai, T., 2007, Multiple mobile robot surveillance in unknown environments, *Advanced Robotics*, vol. 21, pp. 729-749.
- Ulusoy, G., 1985, The fleet size and mix problem for capacitated arc routing, *European Journal of Operation Research*, vol. 22, pp. 329-337.
- Vansteenwegen, P., Souffriau, W. and Sörensen, K., 2009, Solving the mobile mapping van problem: A hybrid metaheuristic for capacitated arc routing with soft time windows, *Computers & Operations Research*, (in press).

**KAYNAKLAR DİZİNİ (devam)**

- Wøhlk, S., 2005, Contributions to arc routing, PhD thesis, University of Southern Denmark, 269 p.
- Wøhlk, S., 2006, New lower bound for the capacitated arc routing problem, *Computers and Operations Research*, vol. 33(12), pp. 3458–3472.
- Wøhlk, S., A Decade of capacitated arc routing, 2008. In: Golden, B., Raghavan, S. and Wasil, E., *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, New York, pp. 29-48.
- Zhu, Z., Li, X., Yang, Y., Deng, X., Xia, M., Xie, Z., Liu J., 2007, A Hybrid Genetic Algorithm for the Multiple Depot Capacitated Arc Routing Problem, *Proceedings of the IEEE International Conference on Automation and Logistics*, Jinan, China, pp. 2253-2258.

## **EKLER**

**Ek.1.** KSHS ve GDB sınıfı test problemlerindeki talep deęerlerinin KS testi sonucu elde edilen daęılımlar.

**Ek.2.** Önerilen genetik algoritma ile elde edilen 10 testin en iyi amaç fonksiyonu deęeri ve çözüm süresi.

**Ek.3.** Üç test probleminin kapasiteleri artırıldıktan sonra önerilen genetik algoritma ile elde edilen 10 testin en iyi amaç fonksiyonu deęeri ve çözüm süresi.

**Ek.1. KSHS ve GDB sınıfı test problemlerindeki talep değerlerinin KS testi sonucu elde edilen dağılımlar**

- *Kesikli Düzgün Dağılım (KDD(a,b))*
- *Sabit*: Bütün talep değerleri aynı
- *Poisson Dağılımı (Poisson( $\lambda$ ))*

Test Problemi	Dağılım	1. Parametre	2. Parametre
kshs1	Kesikli Düzgün Dağılım	6	66
kshs2	Kesikli Düzgün Dağılım	0	66
kshs3	Kesikli Düzgün Dağılım	2	73
kshs4	Kesikli Düzgün Dağılım	9	70
kshs5	Kesikli Düzgün Dağılım	1	58
kshs6	Kesikli Düzgün Dağılım	-7	59
gdb1	Sabit	1	-
gdb2	Sabit	1	-
gdb3	Sabit	1	-
gdb4	Sabit	1	-
gdb5	Sabit	1	-
gdb6	Sabit	1	-
gdb7	Sabit	1	-
gdb8	Kesikli Düzgün Dağılım	1	10
gdb9	Kesikli Düzgün Dağılım	1	10
gdb10	Kesikli Düzgün Dağılım	1	2
gdb11	Kesikli Düzgün Dağılım	1	9
gdb12	Kesikli Düzgün Dağılım	1	18
gdb13	Kesikli Düzgün Dağılım	2	16
gdb14	Kesikli Düzgün Dağılım	1	7
gdb15	Poisson Dağılımı	5.333	-
gdb16	Kesikli Düzgün Dağılım	0	8
gdb17	Kesikli Düzgün Dağılım	3	9
gdb18	Kesikli Düzgün Dağılım	1	8
gdb19	Kesikli Düzgün Dağılım	2	10
gdb20	Kesikli Düzgün Dağılım	0	9
gdb21	Kesikli Düzgün Dağılım	1	9
gdb22	Kesikli Düzgün Dağılım	1	9
gdb23	Kesikli Düzgün Dağılım	1	9

**Ek.2. Önerilen genetik algoritma ile elde edilen 10 testin en iyi amaç fonksiyonu değeri ve çözüm süresi**

	Test-1		Test-2		Test-3		Test-4	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-kshs1	14729	8.56	14980	5.4	15272	5.35	14729	5.68
d-kshs2	9863	6.25	10022	9.59	9953	9.16	9953	5.66
d-kshs3	9320	4.47	9666	3.96	9320	5.36	9320	4.2
d-kshs4	11098	6.84	11098	7.34	11098	32	11368	5.5
d-kshs5	10957	5	10957	4.36	10957	6.26	11715	4.54
d-kshs6	10197	5.6	10197	4.62	10197	4.65	10305	4.66
d-gdb1	327	21.74	323	21.35	348	40.96	335	44.45
d-gdb2	355	43.4	352	34.03	355	22.02	408	78.71
d-gdb3	275	75.5	294	47.57	287	25.79	297	50.18
d-gdb4	287	8.99	287	13.55	287	11.22	287	18.65
d-gdb5	401	21.33	417	16.9	395	17.25	405	23.51
d-gdb6	319	36.78	363	10.72	310	63.59	336	44.27
d-gdb7	361	12.47	325	20.68	330	62.13	345	21.73
d-gdb8	0	210	0	219.05	0	208.84	0	208.72
d-gdb9	0	233.53	0	237.75	0	240.98	0	244.83
d-gdb10	283	10.44	281	8.81	281	15.92	301	9.1
d-gdb11	420	93.81	436	56.03	405	63.36	409	96.72
d-gdb12	0	90.47	0	93.35	0	91.78	0	88.32
d-gdb13	540	26.07	553	16.75	543	21.53	542	23.17
d-gdb14	102	7.19	103	9.43	103	6.94	102	6.26
d-gdb15	60	4.63	60	5.59	58	5.1	60	4.76
d-gdb16	127	8.62	127	15.52	127	11.86	129	7.38
d-gdb17	91	8.4	91	7.69	91	10.21	91	8.13
d-gdb18	168	11.08	166	13.58	166	15	166	21.53
d-gdb19	57	3.27	57	3.26	57	3.23	57	3.32
d-gdb20	123	12.37	124	17.03	123	6.9	121	7.88
d-gdb21	162	15.93	159	19.91	158	15.87	160	18.68
d-gdb22	204	34.67	210	27.38	204	51.51	202	56.83
d-gdb23	248	35.28	239	42.39	250	57.25	251	41.3

	Test-5		Test-6		Test-7	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-kshs1	14729	5.4	14783	6.19	14729	11.46
d-kshs2	9863	6.58	10022	6.25	9923	10.23
d-kshs3	9320	4.14	9320	6.89	9320	4.35
d-kshs4	11098	5.16	11098	5.71	11098	9.53
d-kshs5	10957	4.39	11151	4.41	10957	7.63
d-kshs6	10197	7.84	10197	5.21	10197	5.86
d-gdb1	331	23.48	362	56.72	325	73.34
d-gdb2	372	88.02	353	33.76	382	40.75
d-gdb3	289	21	294	48.61	275	17.01
d-gdb4	287	6.97	287	7.16	287	10.7
d-gdb5	433	18.77	401	94.66	423	15.71
d-gdb6	316	18.14	359	16.91	319	21.16
d-gdb7	373	46.61	350	42.6	352	19.47
d-gdb8	0	206.78	0	207.71	0	212.65
d-gdb9	0	234.69	0	241.33	0	238.66
d-gdb10	284	18.51	275	11.71	300	10.44
d-gdb11	437	86.56	440	82.04	444	55.27
d-gdb12	0	90.03	0	92.15	0	92.8
d-gdb13	558	29.03	540	14.66	548	20.87
d-gdb14	100	10.67	103	6.75	102	10.1
d-gdb15	58	5.29	58	5.51	58	5.9
d-gdb16	127	8.95	127	9.62	127	12.56
d-gdb17	91	8.19	91	8.39	91	8.78
d-gdb18	168	11.56	168	12.97	168	17.76
d-gdb19	57	3.24	57	3.3	67	3.41
d-gdb20	121	12.13	121	6.91	121	10.88
d-gdb21	160	14.39	165	20.99	161	16.65
d-gdb22	204	32.1	209	23.79	203	51.14
d-gdb23	242	50.94	244	48.38	242	45.94

	Test-8		Test-9		Test-10	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-kshs1	14729	8.68	14729	10.2	14729	5.64
d-kshs2	9863	5.94	9923	5.41	9923	20.66
d-kshs3	9320	7.25	9320	4.85	9320	4.27
d-kshs4	11098	6.23	11098	10.32	11812	5.88
d-kshs5	10957	8.62	10957	4.47	11151	5.09
d-kshs6	10197	4.76	10197	5.34	10966	5.41
d-gdb1	323	14.83	323	22.38	323	73.58
d-gdb2	349	31.47	357	35.02	364	17.47
d-gdb3	287	21.79	306	14.52	317	7.73
d-gdb4	287	15.55	311	7.65	291	16.8
d-gdb5	401	69.72	410	38.23	423	11.98
d-gdb6	392	14.86	327	18.6	322	20.76
d-gdb7	335	13.38	340	10.3	330	19.31
d-gdb8	0	203.29	0	206.39	0	212.2
d-gdb9	0	238.6	0	241.42	0	248.85
d-gdb10	311	8.62	313	10.75	298	18.02
d-gdb11	426	57.3	417	60.13	430	56.34
d-gdb12	0	92.69	0	92.67	0	93
d-gdb13	546	17.69	546	20.08	540	19.58
d-gdb14	102	12.01	102	6.84	100	12.49
d-gdb15	58	10.03	58	5.11	60	5.15
d-gdb16	127	8.06	127	12.27	127	8.47
d-gdb17	91	8.6	91	7.98	91	8.43
d-gdb18	166	13.06	168	13.3	166	11.97
d-gdb19	57	3.29	63	3.36	57	3.29
d-gdb20	121	6.42	121	8.05	121	8.82
d-gdb21	171	12.17	168	15.2	160	21.2
d-gdb22	205	37.67	204	18.21	203	40.12
d-gdb23	240	60.97	249	42.33	247	43.33

**Ek.3. Üç test probleminin kapasiteleri artırıldıktan sonra önerilen genetik algoritma ile elde edilen 10 testin en iyi amaç fonksiyonu değeri ve çözüm süresi**

	Test-1		Test-2		Test-3		Test-4	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-gdb8	372	110.59	373	75.38	426	42.59	414	47.31
d-gdb9	342	109.16	363	81.52	394	53.59	386	62.24
d-gdb12	556	10.02	531	15.42	497	12.03	552	14.59

	Test-5		Test-6		Test-7	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-gdb8	372	72.52	393	84.01	394	72.85
d-gdb9	355	99.83	367	47.83	360	72.23
d-gdb12	538	20.72	474	21.13	532	19.24

	Test-8		Test-9		Test-10	
	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi	GA Amaç Fonksiyonu Değeri	Çözüm Süresi
d-gdb8	384	75.19	430	49.61	414	48.67
d-gdb9	353	96.91	360	105.88	375	74.77
d-gdb12	534	19.73	524	14.98	474	21.71