

Birikimli Açık Araç Rotalama Problemi İçin Sezgisel Çözüm Yaklaşımları

Fehmi Burçin Özsoydan

**YÜKSEK LİSANS TEZİ**

Endüstri Mühendisliği Anabilim Dalı

Aralık 2011

Heuristic Solution Approaches For The Cumulative Open Vehicle Routing Problem

Fehmi Burçin Özsoydan

**MASTER SCIENCE THESIS**

Department of Industrial Engineering

December 2011

# Birikimli Açık Araç Rotalama Problemi İin Sezgisel özüm Yaklaşımları

Fehmi Burin Özsoydan

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

Danışman: Do. Dr. Aydın Sipahioğlu

Aralık 2011

## ONAY

Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Fehmi Burçin Özsoydan'ın YÜKSEK LİSANS tezi olarak hazırladığı “Birikimli Açık Araç Rotalama Problemi İçin Sezgisel Çözüm Yaklaşımları” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

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

**İkinci Danışman** : -

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

**Üye** : Doç. Dr. Aydın Sipahioğlu

**Üye** : Prof. Dr. Cevriye Gencer

**Üye** : Prof. Dr. Erdal Emel

**Üye** : Prof. Dr. Müjgan Sağır

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

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

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

## ÖZET

Birikimli Açık Araç Rotalama Problemi (BAARP), kullanılacak araçların kapasite kısıtları altında, bir serimde tüm düğümlere sadece bir kere uğranarak, düğümlere birikimli olarak ulaşma süreleri toplamının enküçüklenmeye çalışıldığı bir kombinatoriyel eniyileme problemidir. BAARP'ın yeni bir problem olması sebebiyle literatürde bu konuyla ilgili az sayıda çalışma vardır. Bu çalışmada, kombinatoriyel problemlerde başarılı sonuçlar elde edebildiği gösterilmiş olan Yasaklı Arama Algoritması ve Genetik Algoritma ile doğrusal olmayan sürekli eniyileme problemlerinde oldukça başarılı olduğu gösterilmiş Parçacık Sürüsü Optimizasyonu açık rotalı BAARP'a uyarlanmış, yöntemlerin birbirlerine göre üstünlükleri ve zayıf noktaları literatürden alınmış test problemleri üzerinde araştırılmıştır.

**Anahtar Kelimeler:** Birikimli açık araç rotalama problemi, kombinatoriyel problem, yasaklı arama algoritması, genetik algoritma, parçacık sürüsü optimizasyonu.

## SUMMARY

Cumulative Open Vehicle Routing Problem (COVRP) is a combinatorial problem of minimizing the summation of cumulative arrival times at nodes, subject to capacity constraints of vehicles and visiting each node exactly once. Because of being a recent subject, the literature contains few studies. In this study, Tabu Search Algorithm and Genetic Algorithm, which were proven to be capable of obtaining successful results on combinatorial problems, and Particle Swarm Optimization, which was shown to be successful on nonlinear continuous optimization, were adapted to open COVRP, and advantages and drawbacks of these methods were compared to each other on the test problems taken from the literature.

Keywords: Cumulative open vehicle routing problem, combinatorial problem, tabu search algorithm, genetic algorithm, particle swarm optimization.

## TEŐEKKÖR

Gerek bilimsel alıőmalarda, gerek geleceđimle ilgili hayati konular ile ilgili karar aőamalarında beni yönlendiren, sürekli fikir alışverişinde bulunduđum, beni akademik dünya ile tanıştıran ve bu alanda gelişimim ile yakından ilgilenip, her türlü destek ve imkânı sağlayan danışmanın Do. Dr. Aydın Sipahiođlu'na ve canımdan ok sevdiđim aileme teőekkürlerimi sunarım.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET .....</b>	<b>v</b>
<b>SUMMARY .....</b>	<b>vi</b>
<b>TEŞEKKÜR .....</b>	<b>vii</b>
<b>ŞEKİLLER DİZİNİ .....</b>	<b>x</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>xii</b>
<b>KISALTMALAR DİZİNİ .....</b>	<b>xiii</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. ROTALAMA PROBLEMLERİ .....</b>	<b>7</b>
2.1. Serim Kuramı ve Araç Rotalamadaki Bazı Gösterim ve Tanımlamalar .....	7
2.2. Düğüm Rotalama Problemleri .....	10
2.3. Ayrıt Rotalama Problemleri .....	13
2.4. Genel Rotalama Problemleri .....	15
<b>3. BİRİKİMLİ AÇIK ARAÇ ROTALAMA PROBLEMİ.....</b>	<b>16</b>
3.1. Yük Miktarına Bağlı Yakıt Tüketimli BARP Matematiksel Modeli.....	27
3.2 Düşümlere Ulaşma Süreleri Toplamının Enküçüklendiği BARP Matematiksel Modeli .....	31



**İÇİNDEKİLER (devam)****Sayfa**

<b>4. BAZI İLERİ SEZGİSEL ÇÖZÜM YAKLAŞIMLARI .....</b>	<b>36</b>
4.1. Yasaklı Arama Algoritması.....	37
4.2. Genetik Algoritma .....	42
4.3. Parçacık Sürüsü Optimizasyonu .....	45
<b>5. BAARP İÇİN GELİŞTİRİLEN ÇÖZÜM YAKLAŞIMLARI.....</b>	<b>55</b>
5.1. Başlangıç Çözüm Bulma Yöntemi.....	55
5.2 BAARP İçin Geliştirilen Yasaklı Arama Yaklaşımı .....	59
5.3 BAARP İçin Geliştirilen Genetik Algoritma Yaklaşımı.....	60
5.4 BAARP İçin Geliştirilen Parçacık Sürüsü Optimizasyonu Yaklaşımı .....	63
<b>6. TEST PROBLEMLERİ VE ELDE EDİLEN SONUÇLARI.....</b>	<b>67</b>
6.1. Test Problemlerinin Özellikleri .....	67
6.2 Amaç Fonksiyonu Değerleri ve Çözüm Sürelerine İlişkin Hesapsal Sonuçlar .....	71
6.3 Sonuçların Tekrar Edilebilirliği Ve Yakınsaklık .....	82
<b>7. SONUÇ VE ÖNERİLER.....</b>	<b>91</b>
<b>KAYNAKLAR DİZİNİ.....</b>	<b>93</b>
<b>EKLER DİZİNİ</b>	

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. a) Yönsüz b) Yönlü c) Karma serimlere örnekler.....	8
2.2. a) Tam bağlı ve ağırlıklandırılmamış b) Seyrek ve ağırlıklandırılmış serim örnekleri.....	9
2.3. a)Bağlı b)Kopuk serim örnekleri.....	10
2.4. a) Düzgün on iki yüzlü b) Karşı gelen uygun bir Hamilton turu.....	11
2.5. a) Köningsberg Köprüleri-1736 b) Köningsberg Köprüleri'nin serim gösterimi.....	13
3.1. ARP ve BAARP kıyaslaması için örnek bir serim.....	17
3.2. a) Klasik ARP için b) BARP için alt sınır.....	24
3.3. $n=3, k=1$ iken alt turların belirlenmesi.....	34
4.1. İkili yer değiştirme.....	38
4.2. Araya sokma.....	38
4.3. Tersine çevirme.....	39
4.4. Yasaklı Arama Algoritmasındaki hafıza tipleri.....	40
4.5. a) Tur ayrıçlı gösterim, b) Büyük tur gösterimi.....	43
4.6. Bir karınca kolonisinin besine erişimi ile ilgili bir gösterim.....	46
4.7. Arı kolonisi ile ilgili bir gösterim.....	47
4.8. Çeşitli hiperküpler.....	51
4.9. Örnek problemdeki noktaların yerleri.....	52
5.1. Tamir ve red yöntemlerinin zayıflıkları.....	58
5.2. Geliştirilen Yasaklı Arama Algoritması yaklaşımının akış şeması.....	59
5.3. Çaprazlamaya girecek bireyler.....	60
5.4. Rassal seçilen birey ve segment.....	61
5.5. Ara birey.....	61
5.6. Çaprazlama sonucu oluşan birey.....	61
5.7. Turnuva seçim yöntemi sözde kodları.....	62
5.8. Geliştirilen genetik algoritma yaklaşımının akış diyagramı.....	63
5.9. PSO çaprazlamasında kullanılan çaprazlama skalası.....	64

**ŞEKİLLER DİZİNİ (devam)**

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
<b>5.10.</b> Geliştirilen PSO yaklaşımındaki çaprazlama işleminin sözde kodları.....	65
<b>5.11.</b> Geliştirilen PSO yaklaşımındaki çaprazlama işleminin sözde kodları.....	66
<b>6.1.</b> A-n53-k7 için yöntemlerin yakınsama grafikleri.....	85
<b>6.2.</b> A-n80-k10 için yöntemlerin yakınsama grafikleri.....	85
<b>6.3.</b> B-n34-k5 için yöntemlerin yakınsama grafikleri.....	86
<b>6.4.</b> P-n76-k5 için yöntemlerin yakınsama grafikleri.....	86

## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
4.1. 0-1 tamsayılı değişkenlerden oluşan bir sürü örneği.....	50
6.1. Literatürden alınmış test problemlerinin özellikleri.....	69
6.2. PSO ile elde edilen amaç fonksiyonu değerleri.....	71
6.3. GA ile elde edilen amaç fonksiyonu değerleri.....	72
6.4. YAA ile elde edilen amaç fonksiyonu değerleri.....	73
6.5. PSO ile elde edilen çözüm süreleri (sn.).....	74
6.6. GA ile elde edilen çözüm süreleri (sn.).....	75
6.7. YAA ile elde edilen çözüm süreleri (sn.).....	76
6.8. PSO ile elde edilen amaç fonksiyonu değerleri için istatistikler.....	78
6.9. GA ile elde edilen amaç fonksiyonu değerleri için istatistikler.....	79
6.10. YAA ile elde edilen amaç fonksiyonu değerleri için istatistikler.....	80
6.11. PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerlerinin eniyi ve ortalama değerler açısından karşılaştırması.....	81
6.12. PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerleri için aralık araştırması .....	83
6.13. PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerleri için standart sapma araştırması .....	84
6.14. Yüzde uzaklıklar çizelgesi.....	87
6.15. % Sıklık ve boyut artışına bağlı analiz sonuçları.....	89

## KISALTMALAR DİZİNİ

### Kısaltmalar    Açıklama

AA	Arı Algoritması
ARP	Araç Rotalama Problemi
BARP	Birikimli Araç Rotalama Problemi
BAARP	Birikimli Açık Araç Rotalama Problemi
ÇDARP	Çok Depolu Araç Rotalama Problemi
ÇPP	Çinli Postacı Problemi
DARP	Dönemlik Araç Rotalama Problemi
DTARP	Dağıtımli-Toplamalı Araç Rotalama Problemi
enk-enb ARP	Enküçük-Enbüyük Araç Rotalama Problemi
enk-ort ARP	Enküçük-Ortalama Araç Rotalama Problemi
GA	Genetik Algoritma
GRP	Genel Rotalama Problemi
GSP	Gezgin Satıcı Problemi
GTARP	Geri Toplamalı Araç Rotalama Problemi
GTP	Gezgin Tamirci Problemi
HÇPP	Hiyerarşik Çinli Postacı Problemi
KARP	Kapasiteli Araç Rotalama Problemi
KKO	Karınca Kolonisi Optimizasyonu
KPP	Kırsal Postacı Problemi
KTARP	Kısmi Teslimatlı Araç Rotalama Problemi
<i>m</i> -GSP	<i>m</i> -Gezgin Satıcı Problemi

**KISALTMALAR DİZİNİ (devam)****Kısaltmalar   Açıklama**

OARP	Olasılıklı Araç Rotalama Problemi
PSO	Parçacık Sürüsü Optimizasyonu
RÇPP	Rüzgârlı Çinli Postacı Problemi
TB	Tavlama Benzetimi
YAA	Yasaklı Arama Algoritması
ZPARP	Zaman Pencerele Araç Rotama Problemi

## 1. GİRİŞ

Taşıma problemi, kaynak kullanımı verimliliğini ciddi boyutlarda etkileyebilen, bu sebeple tüm organizasyonların, hassasiyetle üzerinde durması gereken bir problemdir.

Bir tedarik zincirinde, hem hammaddenin tesise taşınması ve tamamlanmış nihai ürünlerin kullanıcıya ulaştırılması gibi dış lojistik, hem de tesis içindeki malzeme taşımalarının söz konusu olduğu iç lojistik faaliyetleri vardır ve taşıma işinin planlanması kritik öneme sahiptir.

Kamu kurumlarının faaliyetlerinde de önemli taşıma işlemleri vardır. Örneğin konut alanlarındaki çöplerin bir sıraya göre toplanması, kar yağışı yaşanan günlerde cadde ve sokaklarda biriken karların araçlar ya da kişiler tarafından kaldırılması, yollarda buzlanmayı önlemek amaçlı tuzlama faaliyetleri, elektrik ya da telefon tellerinin bakım onarım işlemleri, bir şehir belediyesi için önemli taşıma problemlerine örnek teşkil etmektedir.

Günümüzde, bir kargo şirketinin bir şehirde bir ya da birden fazla şubesi olabilmektedir. Büyük bir şehirde çok sayıda talep noktası ve ilgili kargo şirketinin ülke genelinde yine çok sayıda şubesi olabileceği düşünülürse, kapsama alanındaki düğümlerin taleplerinin hangi sırayla karşılanması gerektiğinin kararı, kargo şirketinin tüm şehirlerdeki tüm şubeleri için günlük ve en önemli operasyonel kararlarından bir tanesidir. Bu süreçte araçların on-line olarak izlenmesi ile gelen taleplere hızlı cevap verebilecek bir iletişim ve planlama bilgi sisteminin olması, taşıma verimliliğini ciddi şekilde arttıracaktır.

İlgili örneklerden hareketle şu söylenebilir: Taşımanın gerçekleştirilebilmesi için, bu işte kullanılan araçların ya da kişilerin izlemesi gereken rotalar vardır. Toplamda kat edilen uzaklık ve bu uzaklığa bağlı olarak oluşan maliyet, bir rotada bulunan noktalara hangi sırayla uğranması gerektiğine göre farklılık gösterebilmektedir. Böyle bir durumda amaç, bir veya birden fazla aracın toplamda kat edeceği mesafenin enküçüklenmesi olmalıdır. Araç ya da kişilerin izlemesi gereken eniyi rotanın

belirlenmeye çalışıldığı bu problem, Araç Rotalama Problemi (ARP) (Vehicle Routing Problem) olarak isimlendirilmektedir.

Rotalama problemleri genel olarak düğüm gezme (node routing) ve ayrıt gezme (arc routing) adı altında iki ana başlıkta incelenmektedir. Gezilecek yerlerin, noktalar ile gösterildiği durum, düğüm gezme problemi başlığı altında ele alınmaktadır. Düğüm gezme probleminin çok çeşitli türleri vardır, ancak, en çok bilineni ve üzerinde en fazla çalışma yapılmış türlerinden biri, Gezgin Satıcı Problemi (GSP)'dir (Travelling Salesman Problem). Bu problem, başlangıç düğümüne dönülmesi koşulu ile tüm düğümlere sadece bir kez uğranarak, toplam mesafenin enküçüklenmesi problemidir. Gezgin Tamirci Problemi (GTP) (Travelling Repairman Problem) de bu grupta incelenen bir problemdir. GSP'den amaç fonksiyonu bakımından farklılık gösterir. GTP'de amaç, tüm düğümlere ulaşma süreleri toplamının enküçüklenmesidir. Böylece bazı müşterileri daha erken ziyaret ederek müşteri memnuniyetinin artırılması hedeflenmektedir.

Rotalama problemlerinin diğer başlığını oluşturan ayrıt gezme problemlerinde, düğümlerin gezilme zorunluluğu yoktur. Burada, düğüm çiftleri arasındaki ayrıtların gezilmesi söz konusudur. Bu başlık altında incelenebilecek başlıca problem, Çinli Postacı Problemi (ÇPP)'dir (Chinese Postman Problem). Amaç, başlangıç düğümüne dönülmesi koşulu ile tüm ayrıtların en az bir kez ziyaret edilerek (zorunlu durumlarda birden fazla kez geçilebilir) toplam mesafeyi enküçük yapacak turun bulunmasıdır.

Bu iki başlıktan farklı olarak, hem düğümlerin hem de ayrıtların gezildiği probleme ise genel rotalama problemi denilmektedir.

Gerçek hayatta rotalama ve taşıma problemlerinde, araçların ve depoların kapasiteleri göz önüne alındığında, düğümlerin taleplerinin tek bir araç ile tek bir depodan karşılanması mümkün olmayabilmektedir. Bu sebeple taşıma işi, tek bir araç ve tek bir depo ile gerçekleştirilebileceği gibi birden fazla araç ve birden fazla depo ile de planlanabilmektedir. Bu durumda, araç filosunu oluşturan her bir araç için, araçların toplamda kat ettiği mesafeyi enküçükleyecek bir rota oluşturulması, bir başka ifadeyle hangi aracın, hangi sırayla, hangi düğümlere uğraması gerektiğinin belirlenmesi gerekmektedir. Ayrıca depoların da kapasiteleri göz önüne alınırsa, başlangıçta hangi



depoda kaç adet araç bulundurulması gerektiği ve herhangi bir aracın başlangıçta konumlandırılmış olduğu depoya dönüp dönmemesi zorunluluğu, rotalama probleminin ayrı karar aşamalarıdır. Bir araç, turuna başladığı depoya dönmüyor ise o aracın turu açık tur (open tour), dönüyor ise kapalı tur (closed tour) olarak isimlendirilmektedir. Unutulmamalıdır ki herhangi bir gerçek hayat problemi açık turlu rotalama gerektiriyor ise, izleyen dönemin rotalarında kullanılacak başlangıç düğümler, araçların mevcut dönemdeki turlarını tamamladıkları son düğümlerdir. Bu da açık turlu rotalamada, izleyen dönem rotalarının etkinliğinin önceki dönemin rotalarına bağlı olmasına sebep olmakta ve bütünsel etkinliğin sağlanabilmesi için, tüm dönem rotalarının birlikte incelenmesi gerekliliğini ortaya çıkarmaktadır. Bir başka ifadeyle problem çok periyotlu bir yapıya dönüşmektedir ve bu da problemin zorlaşması anlamına gelir. Buna benzer çeşitli farklılıklar nedeniyle farklı ARP türleri tanımlanmıştır.

Rotalama ve taşıma problemlerinde araç ve depo kapasitelerinden farklı olarak, bazı özel durumlarla karşılaşılabilmektedir. Örnek olarak, düğümlerin ya da ayrıtların taleplerinin karşılanması için bir zaman aralığının varlığı, problemi daha da karmaşık hale getirir. Bozulabilir gıdaların, kimyasal ürünlerin taşınması ya da hammaddenin bir tesisin ambarına ulaştırılmasında temin süresinin varlığı, bu duruma örnek olur. Öte yandan, taleplerin sadece bir araçla karşılanması zorunluluğu ya da birden fazla araç ile karşılanabilir olması veya dağıtım ile toplama işlerinin eş zamanlı olarak aynı araç ya da birden fazla araç tarafından gerçekleştirilebiliyor olması, bir başka özel duruma örnek olabilir. Günlük hayatta karşılaştığımız bir durum olan, damacaneler ile günlük su dağıtımını yapan araçların, talep noktalarının (siteler) günlük ihtiyacını karşılayacak şekilde hem su dağıtımını yapması hem de boş damacanelerin tekrar aynı araca yüklenmesi, bu duruma örnek olarak verilebilir. Görüldüğü gibi verilen örneklerle çerçevesi ana hatlarıyla çizilmeye çalışılmış rotalama ve taşıma problemlerinin her bir özel durumu, literatürde farklı bir rotalama problemi olarak tanımlanmıştır.

Birikimli Araç Rotalama Problemi (BARP) (Cumulative Capacitated Vehicle Routing Problem), zor ve yeni rotalama problemlerinden bir tanesidir. Bu problem düğüm gezme problemleri sınıfında yer almakta ve servis verilecek yerler düğümler ile gösterilmektedir. En az iki aracın yer aldığı ve tüm araçların tek bir çeşit ürünü taşıdığı varsayılan problemde araçlar homojen (homogeneous fleet) olabileceği gibi farklı

özelliklere sahip de (heterogeneous fleet) olabilmekte ve düğümlerin talepleri, sadece bir araç tarafından bir defada karşılanmaktadır. BARP'da genel olarak bir tane merkezi depo bulunmaktadır. Rotalar kapalı ise araçlar merkezi depoya dönmektedirler. Bu problemde amaç, klasik ARP kısıtları altında, her bir aracın düğümlere ulaşma süreleri toplamını enküçükleyerek, tüm düğümlerin taleplerini karşılamaktır. Böylece müşterilerin daha erken hizmet alması sağlanmakta ve müşteri memnuniyeti arttırılmaktadır. BARP'ın diğer araç rotalama problemlerinden farkı üçüncü bölümde ayrıntılı olarak anlatılmıştır.

Literatürde, *birikimli araç rotalama* ismi altında incelenen bir problem daha tanımlanmıştır. Bu problemin kısıtları yine klasik ARP'deki gibi olup, problem amaç fonksiyonu bakımından farklılık göstermektedir. Temel fikir, aracın yakıt tüketiminin, üzerindeki yük miktarına bağlı oluşudur. Buradan hareketle, aynı özellikteki iki araç, aynı hızda, aynı uzaklığı kat ederlerse, yük miktarı daha fazla olan aracın daha fazla yakıt tüketeceği beklenmektedir. Bu nedenle araçlar rotalanırken sadece düğümler arasındaki uzaklık değil, aynı zamanda düğümlerin talepleri sebebiyle araçta birikimli olarak artacak (toplama durumu) ya da birikimli olarak azalacak (dağıtım) yük miktarı da dikkate alınmaktadır. Açıkça anlaşılabilceği gibi bu problemde, dağıtım ve toplama için farklı rotalar oluşabilmektedir. Buradaki amaç toplam yakıt tüketimini enküçük yapacak turların belirlenmesidir.

Literatürde ARP için önerilmiş oldukça fazla sayıda çözüm yöntemi mevcuttur. Bu çözüm yöntemleri, genel olarak kesin çözüm yöntemleri ve sezgisel yaklaşımlar olarak sınıflandırılmaktadır. Kesin çözüm yöntemleri, fazla zaman alabilmekle birlikte eniyi çözümün garanti edildiği yaklaşımlardır. Buna karşılık sezgisel (heuristic) yöntemler eniyi çözümün garanti edilemediği fakat iyi bir çözümün makul çözüm süresi içerisinde elde edileceğinin beklendiği yöntemlerdir. Gerek çözüm süreleri, gerekse elde edilebilir çözüm değerleri açısından, her iki yaklaşımın da birbirine göre üstünlüklerinin olduğu durumlar mevcuttur.

Çözüm süresinin, parametre sayısındaki artışa bağlı olarak üstel bir artış gösterdiği zor problemlerin (NP-zor) (NP-hard) eniyi çözümünü, kesin çözüm yöntemleriyle polinom zamanda elde eden bir algoritma bilinmemektedir. Ek olarak, bu yöntemlerin bazı problemler üzerinde uzun süre çalışıp, tek bir uygun çözüm

üretemediği veya iyi bir çözüm veremediği durumlarla da karşılaşmak mümkün olabilmektedir. Öte yandan sezgisel yaklaşımlar kısa sürede bir çözüm üretebilmekte ve ilerleyen arđıřtırmalarda bu çözümü hızlıca iyileřtirebilmekte, hatta eniyi çözümü elde edebilmektedirler.

Yapay zekâ teknikleri olarak da bilinen ileri sezgisel (meta-heuristic) algoritmalar, sezgisel algoritmalarından farklı olarak, genellikle doğada karşılaşılan çeşitli davranışların bilgisayar ortamında benzetimini yaparak çözüm üretmeye çalışmaktadırlar. Parçacık Sürüsü Optimizasyonu (PSO) (Particle Swarm Optimization), Genetik Algoritma (GA) (Genetic Algorithm), Yasaklı Arama Algoritması (YAA) (Tabu Search Algorithm), Tavlama Benzetimi (TB) (Simulated Annealing), Arı Algoritması (AA) (Bees Algorithm), Karınca Kolonisi Optimizasyonu (KKO) (Ant Colony Optimization), literatürde sıkça karşılaşılan ileri sezgisel algoritmalarından bazılarıdır.

Literatürde, PSO'nun sürekli ve doğrusal olmayan eniyileme problemlerinde oldukça başarılı sonuçlar türetebildiği gösterilmiştir (Tseng and Liao, 2008). Ancak kesikli eniyileme problemlerindeki performansı konusunda daha az sayıda yayın vardır ve yapısal olarak diđer ileri sezgisel algoritmalarından oldukça farklılık göstermektedir. PSO'nun kesikli çözüm uzaylarına uygulanabilmesi için literatürde çeşitli yöntemler önerilmiştir. Bu konu günümüzde de araştırma ve geliřtirmeye açık bir alan olma özelliğini taşımaktadır. Ayrıca yapılan literatür araştırması sonrasında araç rotalama problemlerinin bazı türlerinde PSO'nun henüz detaylı olarak uygulanmadığı da gözlenmiştir.

Bu çalışmada, düğümlere ulaşma süreleri toplamının enküçüklendiği açık rotalı Birikimli Açık Araç Rotalama Problemi (BAARP) incelenmiş ve PSO'nun bu zor problemdeki başarısının incelenmesi çalışmanın temel motivasyon kaynaklarından biri olmuştur. Bunun yanı sıra ARP'de başarılı sonuçlar verdiği bilinen YAA ve GA'nın da BAARP üzerindeki performansları da literatürden alınmış test problemleri üzerinde karşılaştırılmıştır.

Bu amaçla çalışmanın izleyen bölümleri řu şekilde oluşturulmuştur: 2. bölümde rotalama problemleri tanıtılmış, 3. bölümde BARP ve bu konuda günümüze kadar

yapılmış çalışmalar detaylı bir şekilde aktarılmış, 4. bölümde çalışmada kullanılan 3 farklı ileri sezgisel algoritma hakkında bilgiler verilmiştir. 5. bölümde BAARP için geliştirilmiş bir sezgisel algoritma ve uygulanan ileri sezgisel algoritmalar açıklanmıştır. 6. bölümde türetilen test problemleri üzerinde yapılan çalışmalar sonrasında elde edilen sonuçlar verilmiş ve irdelenmiştir. Sonuç ve öneriler bölümünde ise elde edilen sonuçlar değerlendirilmiş, gelecek çalışmalara ışık tutacağı umulan önerilere yer verilmiştir.

## 2. ROTALAMA PROBLEMLERİ

Rotalama problemi, bir veya daha fazla depoda konumlandırılmış bir veya daha fazla aracın, çeşitli bölgelerin (düğüm, ayırıt) taleplerini karşılayabilecek şekilde, çeşitli kısıtlar (araç kapasitesi, zaman penceresi vb.) altında bir amacı (kat edilen toplam yol uzunluğunun enküçüklenmesi, bir rotadaki en uzun ayırıtın enküçüklenmesi vb.) eniyileyecek şekilde araç turlarının belirlenmesi problemidir. Rotalama problemleri, literatürde düğüm rotalama ve ayırıt rotalama olmak üzere iki ana başlık altına incelenmektedirler.

İzleyen alt bölümde rotalama problemlerine ilişkin bazı gösterim ve tanımlamalar aktarılmış, devamında ise örneklendirmeler ile düğüm rotalama ve ayırıt rotalama problemlerine yer verilmiştir.

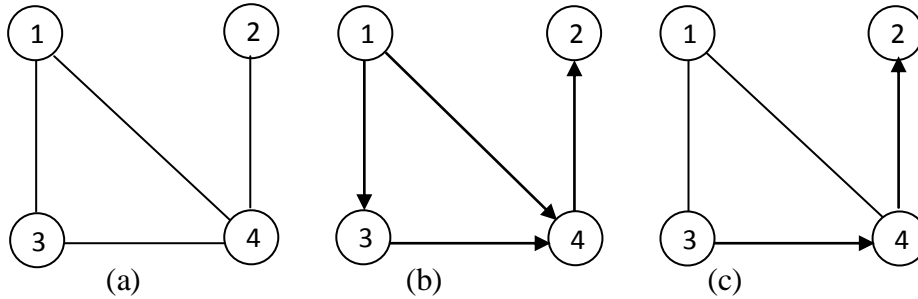
### 2.1. Serim Kuramı ve Araç Rotalamadaki Bazı Gösterim ve Tanımlamalar

Bir serim (network, graph)  $G=(V, E, A)$  ile gösterilebilir. Bu ifadedeki  $V$  düğümler (nodes, vertices) kümesini,  $E$  yönsüz ayırıtlar kümesini,  $A$  ise yönlü ayırıtlar kümesini temsil etmektedir. Bir başka ifadeyle bir serim düğümler, yönsüz ayırıtlar ve yönlü ayırıtlardan oluşmaktadır. Bir serimde düğümler noktaları, ayırıtlar düğümleri birbirine bağlayan uzantıları ifade etmektedir. Örneğin bir haritada kavşaklar, şehirler, depolar, boru hattı kesişim noktaları, tren yolu kesişim noktaları düğümleri, caddeler, sokaklar, telefon hatları, tren yolları, boru hatları ise ayırıtlar olarak görülebilir. Ayırıtlar, tek veya çift yönlü olabilirler, bu da ayırıtların yönlü ve yönsüz olması anlamına gelir. Buna bağlı olarak düğümler arasındaki uzaklıkları gösteren matris de simetrik veya asimetric olarak isimlendirilir.

Yönlü ve yönsüz ayırıt ifadeleri şu şekilde netleştirilebilir: Bir serimde birbirinden farklı herhangi  $v_i$  ve  $v_j$  düğümlerini birleştiren ayırıt  $(i,j)$  şeklinde gösterilmektedir.  $(i,j)$  ayırıtındaki akış hem  $v_i$ 'den  $v_j$ 'ye hem de  $v_j$ 'den  $v_i$ 'ye çift yönlü olarak gerçekleştirilebiliyorsa  $(i,j)$  yönsüz bir ayırıttır ve  $(i,j) \in E$  olarak tanımlanır. Eğer

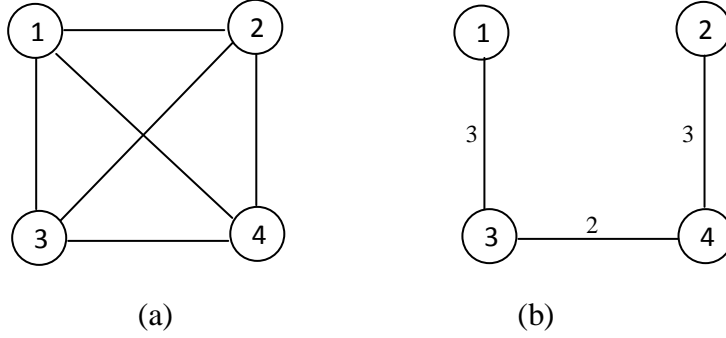
böyle bir durum söz konusu değil ve akış örnek olarak sadece  $v_i$ 'den  $v_j$ 'ye tek yönlü olarak mümkünse, o zaman  $(i,j)$  yönlü bir ayrıttır ve  $(i,j) \in A$  şeklinde tanımlanır.

Serimi oluşturan ayrıtların yönlü ve yönsüz olma durumlarına göre serim de isimlendirilebilmektedir. Bir serimdeki tüm ayrıtlar yönsüz ise serim yönsüz serim (undirected graph), tüm ayrıtlar yönlü ise serim yönlü serim (directed graph) olarak adlandırılmaktadır. Ancak ayrıtların bir kısmı yönlü bir kısmı da yönsüz olabilmektedir. Bu serime de karma serim (mixed graph) denilmektedir. Şekil 2.1'de yönsüz, yönlü ve karma serimlere örnekler verilmiştir.



**Şekil 2.1.** a) Yönsüz, b) Yönlü, c) Karma serimlere örnekler.

Öte yandan serimi oluşturan tüm düğümler kendisi haricindeki tüm düğümlere bağlı ise bu serime tam bağlı (complete) serim, değil ise seyrek (sparse) serim denilmektedir. Serimi oluşturan ayrıtlara, maliyet veya uzaklık gibi çeşitli birimler ile ifade edilebilen büyüklükler atanmış ise serim, ağırlıklandırılmış (weighted) serim, değil ise ağırlıklandırılmamış (unweighted) serim olarak adlandırılmaktadır (Ahuja, et al., 1993). Şekil 2.2'de tam bağlı, ağırlıklandırılmamış ve seyrek, ağırlıklandırılmış serimlere örnek verilmiştir.



**Şekil 2.2. a)** Tam bağlı ve ağırlıklandırılmamış, **b)** Seyrek ve ağırlıklandırılmış serim örnekleri.

Düğümler arasındaki ağırlıklar  $D$  maliyet matrisi yardımıyla gösterilebilmektedir.  $\forall (i,j) \in \{E,A\}$  ve  $i \neq j$  olmak üzere  $(i,j)$  ayrıtının maliyeti  $d_{ij}$  olarak gösterilebilir. Eğer serim yönsüz ise  $D$  matrisi simetriklik özelliği gösterir ise  $\forall i,j \in V$  ve  $i \neq j$  olmak üzere  $d_{ij}=d_{ji}$  olur. Eğer serim seyrek ve birbirinden farklı herhangi  $v_i$  ve  $v_j$  düğümlerini bağlayan bir ayrıt yok ise bu ayrıtın maliyeti uzaklık matrisinde  $d_{ij}=M$  (çok büyük bir sayı) olarak gösterilebilir. Servis bölgelerinin (düğüm ya da ayrıt) talep değerleri ise servis verilecek bölge bir  $v_i \in V$  düğümü ise  $q_i$ , ilgili bölge bir  $(i,j) \in (E, A)$  ve  $i \neq j$  ayrıtı ise  $q_{ij}$  şeklinde ifade edilebilir.

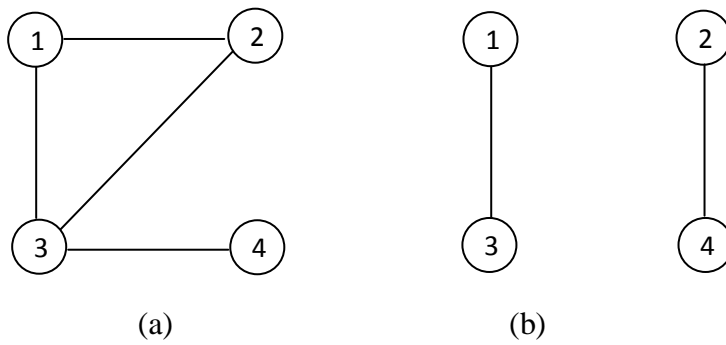
Öte yandan rotalanacak araçlar aynı (identical) özelliklere (kapasite, hız vb.) sahip olabileceği gibi farklı özellikleri de taşıyabilirler. Bu kavramdan hareketle aynı özelliklere sahip araçlardan oluşan filoya homojen filo (homogeneous fleet), farklı özelliklere sahip araçlardan oluşan filoya heterojen filo (heterogenous fleet) denilmektedir ve filodaki  $i$ . aracın kapasitesi  $Q_i$  olarak gösterilmektedir.

Serimlerde filonun başlangıçta konumlandırıldığı düğüm ya da düğümler mevcuttur. Bu düğüm(ler), depo (depot) düğümü olarak adlandırılmakta ve tek depolu bir serimde genellikle 0 ile gösterilmektedir.

Bir  $G$  seriminde ardışık olarak sıralanan düğümler ve ayrıtlar dizisi yürüyüş (walk) olarak adlandırılır.  $i=\{0,1,\dots,n\}$ ,  $k=\{0,1,\dots,(n-1)\}$ ,  $v_i \in V$  ve  $e_k \in (E, A)$  olmak üzere,  $v_0, e_0, v_1, e_1, v_2, e_2, \dots, e_{n-1}, v_{n-1}$  dizisi bir yürüyüşe örnektir.  $v_0=v_n$  ise buna kapalı yürüyüş (closed walk) denir. Yürüyüşteki herhangi bir düğümün tekrarlanmadığı

duruma yol (path) denilmektedir. Başladığı düğümde biten yol, çevrim (cycle) olarak adlandırılmaktadır. Herhangi bir ayrıtın tekrarlanmadığı yürüyüşe tur (tour, trail) denilir (Gross and Yellen, 2004).

Bir serimi oluşturan bütün düğümlerden diğerlerine en az bir yol ile ulaşılabilirse serim bağlıdır (connected), ulaşamıyorsa serime, kopuk (disconnected) serim denir. Bu duruma örnek Şekil 2.3'de verilmiştir.



**Şekil 2.3. a) Bağlı, b) Kopuk serim örnekleri.**

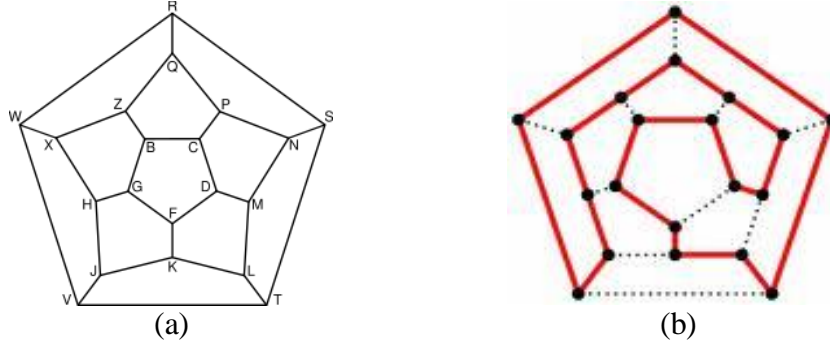
$v_i, v_j \in V, i \neq j$  olmak üzere, eğer  $v_i, v_j$  düğümlerini birleştiren bir  $(i, j) \in (A, E)$  ayrıtı var ise  $(i, j)$  ayrıtı  $v_i$  ve  $v_j$  düğümlerine komşudur (adjacent) (Larsen and Clausen, 2009). Kendisine komşu olan ayrıtıların sayısı, ilgili düğümün derecesini vermektedir. Bu değer tek ise düğüme tek dereceli düğüm (odd vertex), çift ise çift dereceli düğüm (even vertex) ismi verilmektedir. Örnek olarak Şekil 2.3 a'daki 1 ve 2 numaralı düğümler çift dereceli iken, 3 ve 4 numaralı düğümler tek derecelidir.

## 2.2. Düğüm Rotalama Problemleri

Düğüm rotalama problemleri, servis verilecek yerlerin düğümlerde tanımlandığı problemlerdir. Düğüm rotalama problemleri ilk olarak 1857 yılında William Rowan Hamilton tarafından önerilmiş *Icosian Game* isimli matematiksel oyun ile başlamıştır. Bu oyun, düzgün bir on iki yüzlünün (dodecahedron) herhangi bir noktasından başlanarak, tüm düğümlere sadece bir kez uğranıp, düğüm tekrarı yapmadan başlangıç



düğümüne dönülen bir tur bulunması oyunudur. Bu oyundan esinlenilerek söz konusu tura *Hamiltonian tur problemi* denilmiştir. Şekil 2.4’de düzgün bir on iki yüzlü ve ona karşı gelen uygun bir Hamilton turu örneği verilmiştir.



**Şekil 2.4.** a) Düzgün on iki yüzlü, b) Karşı gelen uygun bir Hamilton turu ([http://en.wikipedia.org/wiki/Icosian\\_game](http://en.wikipedia.org/wiki/Icosian_game) [http://en.wikipedia.org/wiki/Icosian\\_Calculus](http://en.wikipedia.org/wiki/Icosian_Calculus)).

Hamilton turu probleminden türeyen ve en yaygın olarak bilinen problem GSP’dir. GSP’de amaç, simetrik veya asimetrik bir serimde herhangi bir düğümden başlanıp, düğüm tekrarı yapılmadan, mümkün olan en düşük maliyetle tüm düğümlerin kapsandığı ve başlangıç düğüme dönüldüğü bir turun belirlenmesidir.

Literatürde GSP’den türemiş değişik problemler vardır. Bunlardan bir tanesi *m*-Gezgin Satıcı Problemi (*m*-GSP) (*m*-Travelling Salesman Problem)’dir. Burada, *m* adet sınırsız kapasiteli varsayılan araç için *m* farklı tur belirlenmeye çalışılmaktadır. Serim simetrik ya da asimetrik olabilmektedir. GSP ve türevleri Gutin ve Punnen (2004) tarafından ayrıntılı bir şekilde anlatılmıştır. Bir diğer problem ise literatürde yaygın olarak çalışılmış kombinatoriyel problemlerden biri olan Kapasiteli Araç Rotalama Problemidir (KARP) (Capacitated Vehicle Routing Problem) (Toth and Vigo, 2001).

Bu problem, genel varsayım altında, yönsüz bir serimde, başlangıçta depo olarak tanımlanmış bir düğüm üzerinde konumlandırılmış *m* adet sınırlı kapasiteli araç için, serimdeki tüm diğer düğümlerin taleplerini sadece bir kere uğrayarak karşılayıp, düğüm

tekrarı yapılmadan mümkün olan en düşük maliyetle depoya dönülen turların belirlenmesi problemidir.

KARP'nin de pek çok alt türü tanımlanmıştır. Aşağıda sıkça karşılaşılan türleri kısaca verilmiştir.

***Zaman Pencereci Araç Rotalama Problemi (ZPARP) (Vehicle Routing Problem With Time Windows):*** ZPARP'de düğümlerin talepleri belirli bir zaman aralığında karşılanmak zorundadır.

***Çok Depolu Araç Rotalama Problemi (ÇDARP) (Multiple Depot Vehicle Routing Problem):*** Düğüm taleplerinin birden fazla depo ile karşılandığı problemidir. Ek olarak bu problemde araçların yola çıktıkları depoya dönme zorunluluğu kaldırılırsa, açık rotalı problem türü ortaya çıkar.

***Olasılıklı Araç Rotalama Problemi (OARP) (Probabilistic Vehicle Routing Problem):*** Talebin veya uğranacak düğümlerin varlığının olasılıklı olması halidir.

***Kısmi Teslimatlı Araç Rotalama Problemi (KTARP) (Split Delivery Vehicle Routing Problem):*** Düğümlerin taleplerinin birden fazla araçla karşılanabilir olduğu durumdur.

***Dağıtım-Toplamalı Araç Rotalama Problemi (DTARP) (Vehicle Routing Problem With Pickup and Delivery):*** Düğüm taleplerinin (dağıtım ve toplama) eşzamanlı karşılanması gereken problemidir.

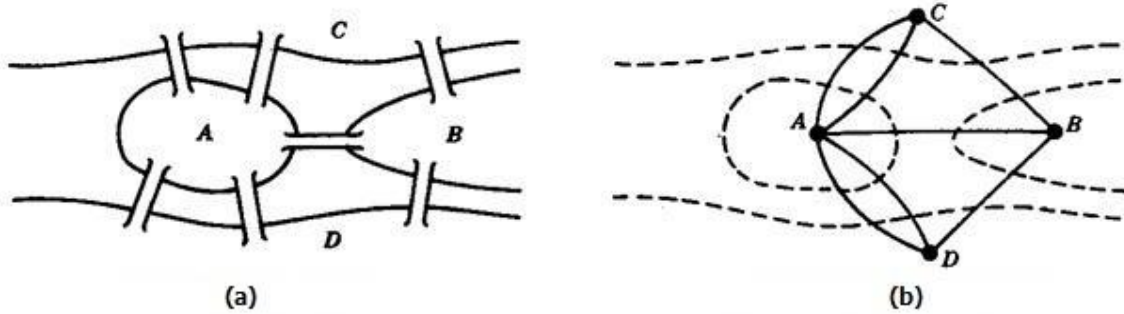
***Dönemlik Araç Rotalama Problemi (DARP) (Periodic Vehicle Routing Problem):***  $T$  gibi bir periyotta müşterilere  $p$  kez uğranması gereken problemidir.

***Geri Toplamalı Araç Rotalama Problemi (GTARP) (Vehicle Routing Problem With Backhauls):*** Müşterilere dağıtım yapıldıktan sonra, bazı müşterilerden toplama yapılması (tekrar uğranması) gereken problemidir.

Yukarıda tanımlanan KARP türlerinin tamamında amaç fonksiyonu, araçların tur uzunlukları toplamının enküçüklenmesi şeklinde tanımlanmıştır.

### 2.3. Ayrıt Rotalama Problemleri

Bir  $G=(V,E)$  serimindeki bütün ayrıtlardan en az bir kez geçip başlangıç düğüme en küçük maliyetle dönülmesi problemi, ayrıt rotalama problemidir. Ayrıt rotalama, ilk kez 1735’de Leonard Euler tarafından Köningsberg köprüleri problemi ile tanımlanmıştır (Euler, 1736; Gribkovskaia, et al., 2007). Bu problem Köningsberg şehrinde bulunan yedi adet köprüden sadece bir kez geçilip başlangıç yerine dönme problemidir. Euler, Köningsberg köprüleri problemini serim biçiminde tanımlayarak ilk kez serim kavramını ortaya atmış, ayrıca literatüre Euler Turu (Euler Tour) kavramını kazandırmıştır. Euler turu serimdeki her ayrıttan sadece bir kez geçerek başlangıç noktaya dönülmesi ile oluşan turdur. Euler, yönsüz bir serimde bütün düğümlerin dereceleri çift ise bir Euler turu olacağını aksi halde olamayacağını da ispatlamıştır. Köningsberg köprüleri (a) ve karşı gelen serim (b) Şekil 2.5’de verilmiştir. Serimden görülebileceği gibi düğümler tek derecelidir ve bu serimde bir Euler turu olamaz. Ancak bazı yollardan birkaç kere geçerek başlangıca dönülecek bir tur bulunabilir.



**Şekil 2.5.** a) Köningsberg Köprüleri-1736, b) Köningsberg Köprüleri'nin serim gösterimi (<http://www.transtutors.com/homework-help/Discrete+Mathematics/Graph+Theory/konisberg-multigraph-bridge.aspx>).

Tıpkı düğüm rotalama problemlerinde Hamilton turu probleminden türetilmiş problemler olduğu gibi, ayrıt rotalama problemlerinde de Euler turu probleminden türetilmiş çeşitli problemler vardır. Euler turu probleminden sonra tanımlanmış olan ayrıt rotalama problemi, Çinli Postacı Problemi (ÇPP) (Chinese Postman Problem)'dir.

ÇPP’de amaç, bir  $G=(V, E, A)$  serimindeki tüm ayrıtlardan en az bir kez geçilerek başlangıç düğümüne dönecek en küçük maliyetli turun belirlenmesidir. Bu problem ilk kez Çinli bir matematikçi olan Mei-Ko Kwan tarafından 1962 yılında önerilmiştir. Edmonds ve Johnson (1973) ve Eiselt’e (2000) göre serim yönlü veya yönsüz olsa da problem polinom zamanda çözülebilmektedir. Öte yandan, serim karma bir serim ise problem NP-zor olmakta (Papadimitriou, 1976) ve problemi polinom zamanlı çözebilen bir algoritma bilinmemektedir.

ÇPP’nin çeşitli türleri bulunmaktadır. Bunlardan en çok bilinenleri aşağıda verilmiştir.

**Rüzgârlı Çinli Postacı Problemi (RÇPP) (Windy CPP):**  $(i,j) \in E$  ve  $d_{i,j} \neq d_{j,i}$  durumu Rüzgârlı Çinli Postacı Problemi’ni tariflemektedir. Buna göre ayrıtların üzerindeki akış yönü, maliyeti değiştirebilmektedir. Aynı özellik ve ağırlıktaki iki araçtan, yokuş tırmanan araç daha fazla yakıt tüketir. Yokuş inen ise daha az yakıt tüketmektedir. Bu nedenle bu probleme göre, serim üzerindeki ayrıtlar akış yönüne göre farklı maliyetle ele alınırlar.

**Hiyerarşik Çinli Postacı Problemi (HÇPP) (Hierarchial CPP):** Gezilmesi gereken ayrıtların birbirlerine göre önceliğinin olduğu ÇPP türüdür (Eiselt et al., 1995a).

**Kırsal Postacı Problemi (KPP) (Rural Postman Problem):** KPP, Orlof (1974) tarafından önerilmiştir. ÇPP’ne benzemekte fakat serimdeki tüm ayrıtlar yerine, ayrıtların bir alt kümesinin gezilme zorunluluğu vardır. Bir başka ifadeyle KPP, bir serimdeki ayrıtların verilen bir alt kümesini kapsayan en küçük maliyetli turun bulunması problemidir. Lenstra ve Rinnooy Kan (1976) tarafından NP-zor olduğu gösterilmiştir. KPP’nin uygulama alanları ve çözüm yaklaşımlarına, Eiselt vd., (1995b)’nin çalışmalarında ayrıntılı olarak yer verilmiştir.

Bunlara ek olarak birden fazla sınırlı kapasiteli aracın olduğu ayrıt rotalama problemi ise Kapasiteli Ayrıtlar Rotalama Problemi (Capacitated Arc Routing Problem) olarak adlandırılmaktadır.

#### 2.4. Genel Rotalama Problemleri

Hem düğüm rotalama hem de ayrıt rotalama problemlerinin bir arada bulunduğu problemler de mevcuttur. Her iki rotalama problemlerini içeren problemler Genel Rotalama Problemi (GRP) (General Routing Problem) olarak adlandırılmaktadır. GRP verilen bir  $G=(V, E, A)$  seriminde tanımlı olan servis verilmesi gereken tüm ayrıtlara ve düğümlere en az bir kez uğranarak, en küçük maliyetli turun belirlenmesi problemidir. İlk olarak Orloff (1974) tarafından literatüre kazandırılan GRP,  $V=\emptyset$  durumunda ayrıt rotalama problemine indirgenmektedir. Bu sebeple NP-zor sınıfında yer alır.

### 3. BİRİKİMLİ AÇIK ARAÇ ROTALAMA PROBLEMİ

Literatürde BARP başlığı altında günümüze kadar yapılan çalışmalar incelendiğinde, bu çalışmaların iki ana başlık altında toplanabileceği açıkça görülmektedir.

Bunlardan ilki, klasik ARP kısıtlarının sağlandığı ve tüm düğümlere ulaşma süreleri toplamının enküçüklendiği birikimli araç rotalama problemidir (BARP). Bu problemde tüm araçların düğümlere ulaşma zamanlarının toplamı enküçüklenmeye çalışılmaktadır. Araçların düğümlere ulaşım sürelerinin toplamının birikimli olarak artış göstermesi nedeniyle de, problem bu şekilde isimlendirilmiştir.

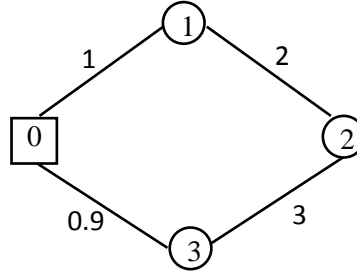
Birikimli sürelerin toplamını enküçükleme bazı müşterilere mümkün olan en kısa sürede ulaşmak anlamına gelmektedir. Bu da müşteri memnuniyetini arttıracığı için tercih edilen bir durumdur. BARP ile ilk olarak Ngueveu vd. (2010) çalışmışlardır.

Bu problem GTP'nin (Salehipour et al., 2008) geliştirilmiş halidir (Ngueveu et al., 2010). GTP, literatürde aynı zamanda Enküçük Gecikme Problemi (Minimum Latency Problem) olarak da bilinmektedir. Ribeiro ve Laporte (2012) ise BARP'ın, Dağıtıcı Problemi'nin (Deliveryman Problem) (Lucena, 1990; Fischetti et al., 1993) uzantısı olduğunu belirtmişlerdir. Bu çalışmada ise açık rotalı BAARP incelenmiştir.

BAARP, gerçek hayatta iş-makine çizelgeleme uygulamalarının yanı sıra, özellikle müşteri tatminine öncelik verildiği durumlarda ya da doğal felaketler sonrası hayati ihtiyaç ve ilk yardım malzemelerinin dağıtımında önem kazanmaktadır. Diğer araç rotalama problemlerinden farklılıkları Ngueveu vd. (2010) tarafından belirtilmiştir. Bu açıklamalar BAARP'in diğer benzer problemlerden farklılıklarını ortaya koymaktadır. Enk-ort Araç Rotalama Probleminde (Campbell et al., 2008) (enk-ort ARP) (min-avg VRP) düğümlere ulaşma süreleri ortalaması, enk-enb Araç Rotalama Probleminde (Arkin et al., 2006) (enk-enb ARP) (min-max VRP) ise herhangi bir düğüme, enbüyük ulaşma süresi enküçüklenmeye çalışılmaktadır. Ngueveu vd.'ne (2010) göre enk-ort ARP, BAARP ile eşdeğer sayılmaktadır fakat BAARP ile enk-enb

ARP oldukça farklı problemlerdir. BAARP'da tüm düğümlere ulaşma süreleri, enk-enb ARP'de ise herhangi bir düğüme enbüyük ulaşma süresi dikkate alınmaktadır.

BAARP'ın klasik araç rotalama ve açık araç rotalama probleminden farkı ise aşağıda, Şekil 3.1'de, 3 düğüm ve 1 merkezi depodan oluşan küçük bir serimde örneklendirilmiştir. Şekil 3.1'de 0 düğümü depoyu, 1, 2 ve 3 düğümleri ise uğranması gereken düğümleri göstermektedir. Ayrıtların uzunlukları ise serim üzerinde verilmiştir. Örnek olarak *depo* ile 1. düğüm arası uzaklık 1br'dir. Aracın tüm ayrıtlarda aynı hızda yol aldığını varsayarsak, örnek serimde verilen uzaklıklar, ilgili düğümler arası gerekli süre olarak da kullanılabilir. Son olarak bu örnekte taleplerin tek bir araç tarafından karşılanabildiği varsayılmıştır.



**Şekil 3.1.** ARP ve BAARP kıyaslaması için örnek bir serim.

Klasik ARP'ye göre eniyi tur, depoya döneceği için 0-1-2-3-0 ya da 0-3-2-1-0 olmaktadır. Bu iki farklı turun maliyeti aynı ve 6.9'dur. Açık ARP'ye göre ise eniyi tur 0-3-2-1 olup turun maliyeti 5.9'dur. Tez kapsamında incelenen BAARP'a göre ise eniyi tur, Açık ARP'den farklı olup 0-1-2-3 olmakta ve maliyeti  $1+3+6=10$  olmaktadır.

Bu serimin doğal felaket sonrası uğranması gereken düğümleri temsil ettiğini düşünürsek ve Açık ARP'nin eniyi turu (0-3-2-1) BAARP'a uygulandığında ilk düğüme uğranma zamanı 0.9, turdaki 2. düğüme uğranma zamanı 3.9 ve son düğüme varış zamanı ise 5.9, düğümlere ulaşma süreleri toplamı ise 10.7 olmaktadır. Fakat insani ihtiyaçların toplamda daha kısa sürede karşılanabildiği ya da müşteri tatmininin daha iyi sağlanabildiği 10 maliyetli farklı bir tur (0-1-2-3) vardır. Açık ARP'nin eniyi turunun

bu amaca uygun olmadığı ve böyle durumlar için farklı bir amaç fonksiyonuna ihtiyaç duyulduğu burada açıkça görülmektedir.

BARP'nin diğer türü ise, aracın üzerindeki yük miktarına bağlı olarak değişen yakıt tüketim miktarının da hesaba katıldığı ve yine klasik ARP kısıtlarının sağlanıp toplam tur maliyetinin enküçüklendiği problemleri kapsamaktadır. Literatürde bu ana başlık altında yapılmış çalışmaları da kendi arasında iki alt gruba ayırmak mümkündür.

Birinci alt grup çalışmalar daha çok makine mühendisliği ile endüstri mühendisliği ortak çalışma alanıyla ilintilidir. Burada araçların farklı ortamlarda (farklı hava ve yol koşulları, farklı sürücüler, farklı yük miktarı, trafik yoğunluğu vb.) yakıt tüketim miktarları çeşitli deneyler ile ölçülmektedir. Bu deneylerde, küresel konumlandırma sistemleri (global positioning systems-GPS) ve özellikle araçların egzoz çıkışlarında CO<sub>2</sub> ölçüm cihazları kullanılmaktadır. Bu konudaki ilk çalışmalar, erişilebildiği kadarıyla Rees'e aittir (2004). Diğer bir alt başlıkta ise araçların faaliyet gösterdiği çevre koşulların aynı olduğu ve sadece aracın üzerindeki yük miktarına bağlı olarak yakıt tüketiminin gerçekleştiği varsayılmaktadır. Literatüre erişilebildiği kadarıyla bu konudaki ilk çalışma Kara vd.'ne (2007) aittir.

Yukarıda anlatılanlardan da anlaşılacağı gibi BARP oldukça yeni ve araştırmaya açık bir konudur. Konunun güncelliği, bu çalışma kapsamında araştırılmasının başlıca sebeplerinden bir tanesidir.

Bu çalışmada öncelikle yakıt tüketimli BARP'a yer verilmiş, ardından çalışmanın asıl konusu olan, klasik ARP kısıtları altında düğümlere ulaşma süreleri toplamının enküçüklendiği BAARP ile ilgili çalışmalar aktarılmıştır.

Ress (2004), engebeli arazide rotalama problemi için yeni bir model ile birlikte anlık olarak bilgilerin işlenmesi ve modelin çözülmesi için bir yöntem önermiştir. Bu model aslında düğümler arasındaki fiziki mesafeyi önerdiği teknik ile yeniden hesaplayıp, bu verilere Dijkstra (1959) algoritmasını uygulamaktadır. Birleşik Devletler Ulusal Sağlık Enstitüsü'nde geliştirilen açık kaynak kodlu görüntü-işleyiciye (<http://rsb.info.nih.gov/nih-image/>) Dijkstra algoritmasının bilgisayar kodları yazılmış,



böylece sadece yatay eksen uzaklığı değil, eğim sebebiyle oluşan engebeli arazi koşulları da yakıt tüketiminde hesaba katılmıştır.

Tavares vd. (2009) atık toplama yönetimi konusunda yaptıkları çalışmada, atık toplayan araçların rotalanması işleminde, uğranması gereken düğümlerin çizelgelenmesinin yanı sıra aracın ağırlığı ve yolların eğimini de dikkate alan bir üç boyutlu coğrafi bilgi sistemi (3D-GIS) geliştirmişlerdir. Yakıt tüketimi hesaplamalarında ise Karayolu Taşımacılığında Salınım Hesaplayıcı Bilgisayar Programı'ndan (COPERT) (Ntziachristos and Samaras, 2000) faydalanmışlardır. Yaklaşım, Yeşil Burun Adaları'nın başkenti olan Praira ve Santiago Adası olmak üzere iki farklı bölgede uygulanmıştır. Sonuçta kat edilen mesafe %1,8 artarken, diğer taraftan yakıt tüketiminde %12'lere ulaşan bir azalma elde edilmiştir. Bu çalışmada ayrıca 3 boyutlu coğrafi bilgi sisteminin, 2 boyutlu uygulamalarına göre daha gerçekçi ve hassas sonuçlar üretebileceği gösterilmiştir.

Yong ve Xiaofeng (2009), BARP ile klasik ARP arasındaki farkın önemini vurgulayıp, bu iki problemin matematiksel modelleri ile elde edilen küçük boyutlu ( $n=7$ ) bir örnek problem üzerinde bile önemli farklılıklar oluşabileceğini vurgulamışlardır. Ayrıca toplam tur uzunluğu ve yakıt tüketiminin eniyilenmesi konusunun aynı anda iki amaçlı olarak ele alınması fikrini ortaya atmışlardır. Bunun CO<sub>2</sub> salınımı azaltılması ile hava kirliliği ve sera etkisi üzerinde olumlu etkileri olacağını belirtmişlerdir.

Hao (2010) BARP'ı çözmek için bir genetik algoritma önermiştir. Önerilen genetik algoritmanın bazı özellikleri şu şekildedir: Her bir araç için, uğradığı düğümlerin numaralarını sırası ile gösteren bir permütasyon oluşturulmuştur. Bütün permütasyonlar, bir başka ifade ile her bir aracın kat ettiği tur birbirine eklenmiş ve daha büyük bir zincir elde edilmiştir. Turları birbirinden ayırt etmek amacıyla her bir permütasyonun arasına depo düğümünü temsil eden bir işaret koyulmuştur. Özetle önerilen algoritmadaki kromozom gösterimi araçların turlarını parçalı olarak birbirlerine eklenmesiyle tasarlanmıştır. Hao, önerdiği algorithmada çaprazlama operatörü olarak kısmi eşleşmeli çaprazlama (Partial-Mapped Crossover) operatörünü kullanmıştır. Bu operatörle ilgili ayrıntılı bilgi Gen ve Cheng'in (1997) çalışmalarında yer almaktadır.

Mutasyon operatörü olarak ise ikili yer değiştirme tekniği benimsenmiştir. Bu teknikte bir kromozomdaki herhangi iki düğüm rassal olarak seçilir ve birbirleri ile yer değiştirirler. Kullanılan seçim tekniği ise aşağıdaki formülün yardımı ile gerçekleştirilir.

$$f_i' = \frac{f_i - f_{min}}{f_{max} - f_{min} + 0,0001} \quad (3.1)$$

Bu formülde  $f_i'$  i. bireyin uyum değerini,  $f_i$  i. bireyin amaç fonksiyonu değerini,  $f_{min}$  işlem yapılan popülasyondaki enküçük amaç fonksiyonu değerini ve benzer şekilde  $f_{max}$  aynı popülasyondaki enbüyük amaç fonksiyonu değerini temsil etmektedir. Her bir birey için  $f_i'$  değerleri hesaplandıktan sonra bir rassal sayı türetilir ve bu rassal sayının üzerinde kalan bireyler sonraki izleyen popülasyonda çaprazlama ve mutasyon işlemleri için seçilmiş olur.

Genetik algoritma özelliklerinden farklı olarak, bu çalışmada farklı bir nokta araç sayısı tayininin bir formül ile elde edilmesidir. İlgili formül aşağıda verilmiş olup  $m$ , gerekli olan araç sayısını,  $d_i$  i. müşterinin talebini,  $q$  araç kapasitesini,  $\alpha \in (0,1]$  olmak üzere, karmaşıklık katsayısını temsil etmektedir.

$$m = \left\lceil \frac{\sum d_i}{\alpha q} \right\rceil + 1 \quad (3.2)$$

Formülde yer alan  $\alpha$ , 1'e yaklaştıkça problemin zorluğu artmakta ve uygun bir çözüm bulmak zorlaşmaktadır.

Hao (2010), bir aracın  $i$  ayrıtındaki yakıt tüketimini hesaplamak için ise aşağıdaki formülleri kullanmıştır.

$$h_i = h_1 + r_i(h_2 - h_1) \quad (3.3)$$

$$r_i = \frac{g_i}{q} \quad (3.4)$$

Bu formülde  $h_i$  i. ayrıtta birim yolda tüketilen yakıt miktarını,  $h_1$  aracın yüksüz iken birim yolda tükettiği yakıt miktarını,  $h_2$  aracın tam yüklü iken birim yolda tükettiği

yakıt miktarını,  $r_i$  aracın  $i$ . ayrıttaki yük doluluk oranını,  $g_i$   $i$ . ayrıtta taşınan yük miktarını ve  $q$  ise aracın kapasitesini göstermektedir.

Tang vd. (2010), BARP'ı farklı isimle adlandırıp, aynı problemi çözmüşlerdir. BARP için kullandıkları isim Yükleme Maliyetli ARP'dir (Vehicle Routing With Loading Cost). Bu çalışmada Dağıtma Arama (Scatter Search) tekniği kullanılmıştır. Bu teknik, süpürme (Sweep) ve Optimal Bölme Prosedürü (Optimal Splitting Procedure) (Prins, 2004) ile güçlendirilmiştir.

Kuo (2010), aynı konudaki Zaman Pencere (With Time Windows) çalışmalarında çözüm yaklaşımı olarak Tavlama Benzetimi'ni (Kirkpatrick et al., 1983) kullanmıştır. Bu çalışmada aynı konudaki diğer çalışmalardan farklı olarak trafiğin gün içerisindeki durumu göz önünde bulundurulmuş ve trafik akışının günün farklı saatlerinde farklı hızlarda gerçekleşeceği düşüncesiyle ilgili zaman dilimlerinde araçlar tarafından farklı miktarlarda birim yakıt tüketimi gerçekleştirilebileceği varsayılmıştır.

Bunun için öncelikle, gün, farklı zaman dilimlerine bölünmüş ve bu zaman dilimlerinde (şehirdeki çeşitli kurumlardaki mesai başlangıç ve bitiş zamanları vb. faktörler de düşünülerek) araçların yol üzerinde yapabileceği düşük, orta ve yüksek olmak üzere üç farklı ortalama hız değeri hesaplanmıştır. Ortalama hız değerleri, birim hacim yakıt tüketiminde kat edilebilecek yola bölünerek ilgili aracın, ilgili ortalama hız değerinde, birim saatte beklenen yakıt tüketim miktarına ulaşılmıştır. Serimdeki mesafeler ile aracın ne kadar süre yolculuk edeceği ve yukarıda verilen yaklaşım ile de ne kadar yakıt tüketileceği hesaplanmıştır.

Aynı çalışmada aracın taşıdığı yük miktarına da değinilmiştir. Aracın taşıdığı yük miktarı aracın toplam kapasitesine bölünerek % doluluk oranı hesaplanmış ve yakıt tüketimi bu oran ile çarpılıp daha az yük taşıyan aracın daha az yakıt tüketimi gerçekleştireceği düşüncesi, çalışmaya yansıtılmıştır.

Tsang vd. (2011), çalışmalarında, aracın yükten bağımsız olarak farklı yol koşullarındaki birim yakıt tüketim miktarını araştırmışlardır. Bunun için Hong Kong'da mevcut olarak kullanılan dört farklı rota üzerinde araştırma yapılmıştır. Bunlar şehir içi, şehirlerarası, karışık ve dağlık rotalardır. Aynı tipteki aynı araçlar, aynı sürücü

tarafından bu 4 farklı ortamda sürülmüştür. Aracın egzoz borusuna yerleştirilen ve saniye temelinde anlık bilgi aktarımını sağlayan cihazlar ile aracın açığa çıkardığı CO<sub>2</sub> miktarı ile tüketilen yakıt miktarı belirlenmeye çalışılmıştır. Buna göre dağlık rotada ortalama birim yakıt tüketim miktarı en yüksek olarak hesaplanmış, onu takip eden şehir içi rotası olmuştur. Dağlık rotada özellikle pozitif eğimli yollardaki ve şehir içi rotada trafik faktörü sebebiyle oluşan durma/kalkma işlemleri anlarındaki yakıt tüketimi artışının göze çarpıcı boyutlara eriştiği, bu çalışmada belirtilmiştir. Bu rotaları sırasıyla karışık ve şehirlerarası rotalar takip etmektedir.

Demir vd. (2011), araçların yakıt tüketimi konusundaki daha önce önerilmiş farklı modelleri inceleyip karşılaştırmalı analiz yapmışlardır. Bu modeller sırasıyla, anlık yakıt tüketimi modeli (instantaneous fuel consumption model) (Bowyer et al., 1985), 4-modlu temel yakıt tüketim modeli (four-mode elemental fuel consumption model) (Akçelik, 1982), seyir hızı yakıt tüketim modeli (running speed fuel consumption model) (Bowyer, 1985), toplu biçimli tüketim modeli (comprehensive modal emission model) (Barth et al., 2000), (Barth and Boriboonsin, 2008), ulaştırma salınım ve enerji tüketimi yöntemi (methodology for calculating transportation emissions and energy consumption) (Hickman, 1999) ve karayolu taşımacılığında salınım hesaplayıcı bilgisayar programıdır (Computer programme to Calculate Emissions From Road Transportation) (Ntziachristos and Samaras, 2000).

4-Modlu Temel Yakıt Tüketim Modeli'nde, hızlanma, yavaşlama, seyretme ve boşta bekleme gibi dört farklı eylemde farklı yakıt tüketimi gerçekleşeceği varsayımıyla her biri için farklı bir yakıt tüketim fonksiyonu önerilmiştir. Toplu Biçimli Tüketim Modeli'nde ise yakıt tüketimi üç farklı kriterin toplamı şeklinde hesaplanmaktadır. Bunlar, motor gücü, motor hızı ve saniyelik yakıt tüketim oranıdır.

Yukarıda verilen altı farklı model, araç, sürücü, çevresel durumlar ve trafik akışı olmak üzere dört farklı faktörü içeren farklı senaryolarda birbirleri ile kıyaslanmıştır. Sonuçta, yol eğiminin, araç büyüklüğünün ve seyir hızının yakıt tüketimine ciddi etkileri olduğu gözlemlenmiştir.

Kara vd. (2007), Enerji Enküçüklemeli Araç Rotalama Problemi (EEARP) (Energy Minimizing Vehicle Routing Problem) adı altında aynı konu üzerine

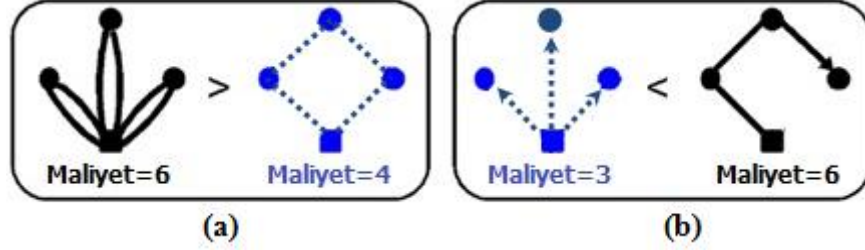
çalışmışlardır. Diğer çalışmalarla benzer yaklaşım ve uygulama farklı isimle adlandırılmıştır. Bu çalışmada klasik ARP ile toplam uzaklığın enküçüklendiği sonuçlar ve EEARP yaklaşımında kullanılan, araç yükü ve düğümler arası uzaklık kaynaklı toplam yakıt tüketiminin enküçüklendiği sonuçlar kıyaslanmıştır. Hesaplamalarda sadece araçtaki yük göz önünde bulundurulmuş, diğer faktörler (hız, yolun durumu, eğim, vb.) göz ardı edilmiştir.

Bektaş ve Laporte (2011), benzer konularda çalışmışlar, bir matematiksel model önermişler ve problemi üç farklı bakış açısından ele alıp, gerçek veriler ile üç farklı modeli çözüp, eniyi rotaları belirlemişlerdir. Bunlardan birincisi hız ve yük faktörünü göz önünde bulundurmakta, ikincisi zaman pencereyi ya da zaman penceresiz olmak üzere CO<sub>2</sub> salınımını enküçüklemekte, üçüncüsü ise BARP'daki çeşitli bileşenler (uzaklık, yük, salınım, maliyet, vb.) arasında ödünleşme analizi yapmaktadır.

Daha önce de bahsedildiği gibi Nguèveu vd. (2010), BARP başlığı altında, yukarıda özetlenen literatürden oldukça farklı olarak, maliyet bileşeni olarak yakıt tüketimini dikkate alan BARP yerine, klasik ARP kısıtlarının yer aldığı fakat amaç fonksiyonunun, tüm düğümlere ulaşım süreleri toplamının enküçüklenmesi şeklinde oluşturulmuş bir problemi işaret edip, önerdikleri bu problem için, Genetik Algoritma tabanlı bir Memetik Algoritma (Memetic Algorithm) geliştirmişlerdir. Önerilen bu algoritma, erişebildiği kadarıyla BARP'ın bu türü için literatürde önerilmiş *ilk sezgisel algoritma* özelliğini taşımaktadır. Çalışmada ayrıca BARP'nin önemli özellikleri vurgulanmış, problemin yapısından kaynaklanan üç farklı alt sınır elde etme tekniği önerilmiştir. Bu özellikler ve alt sınır elde etme teknikleri aşağıda verilmiştir.

**Özellik 1:** GTP, BARP için bir altı sınır oluşturamamaktadır. Buna ek olarak ARP'de bir aracın mümkün olduğu kadar çok düğümü kapsaması ve daha az araç kullanılması istenirken, BARP'ta eğer mümkünse (yeteri kadar araç varsa) depodan her düğüme sadece bir araç gönderilerek eniyi sonuç elde edilebilmektedir. Bu özellik 1. alt sınır elde etme tekniğini oluşturmaktadır. Özetle bu özellik şu duruma işaret etmektedir: BARP'ta kullanılacak araç sayısı artarken amaç fonksiyonu iyileşmektedir. ARP için durumun tersi söz konusudur. Mümkünse tüm düğümlerin taleplerinin tek bir araç ile

karşılanması istenmektedir. Bu nedenle GSP, KARP için bir alt sınır oluştururken GTP, BARP için bir alt sınır oluşturamamaktadır. Bu durum Şekil 3.2’de gösterilmiştir.



Şekil 3.2. a) Klasik ARP için, b) BARP için alt sınır (Ngueveu et al., 2010).

Yukarıdaki şekilde kare düğümler depo iken diğer düğümler müşterileri temsil etmekte ve her düğüm arası uzaklığın 1 br olduğu varsayılmaktadır. Şekil 3.2 (a)’da, KARP için, üç düğümün taleplerinin üç araç tarafından karşılanması (her düğüme tek bir araç) ile aynı üç düğümün taleplerinin tek bir araç tarafından karşılanması durumları karşılaştırılmıştır. Tek araçla karşılanan durumda 4br’lik maliyet oluşurken ((a)’da kesikli çizgi ile gösterilmiştir), KARP için alt sınır oluşturmuştur. Her düğüme bir araç servis verdiğinde ise ((a)’da sürekli çizgi ile gösterilmiştir) her araç depodan çıkıp, müşteriye uğrayıp, depoya döneceği için 6 br’lik maliyet oluşacaktır. Şekil 3.2 (a) KARP için araç sayısı azaldıkça amaç fonksiyonunun iyileşeceğini göstermektedir.

Şekil 3.2 (b)’de ise BARP için aynı karşılaştırma yapılmıştır fakat sonuç tam tersini göstermektedir. Burada kesikli çizgi ile gösterilen durum, her müşterinin tek bir araç tarafından servis edildiği durumdur. 3 br’lik maliyetle alt sınır oluşturmaktadır. Siyah ile gösterilen rotada ise müşterilere ulaşmanın maliyetleri sırasıyla 1 br, 2 br ve 3 br olacağından amaç fonksiyonu değeri 6 br olacaktır. Buradan da BARP’da araç sayısı arttıkça amaç fonksiyonu değeri iyileşeceği sonucuna varılmaktadır.

**Özellik 2:**  $R$ , toplam araç sayısını,  $n$ , depo haricindeki toplam düğüm sayısını gösteriyor iken, BARP’da eniyi çözüm  $enk(n, R)$  kadar araç kullanılarak elde edilebilir. Ayrıca açıkça görülmektedir ki araç sayısı artıyor iken çözüm değeri azalır, araç sayısı azalıyor iken çözüm değeri artar. Bunun en önemli sonucu, BARP için eniyi çözüm (varsa) sadece  $R$  adet araç kullanılarak elde edilebilir. KARP’da ise durum tam tersidir.

Toplamda en fazla  $R$  adet araç ile (mümkünse  $R$ 'den daha az sayıda araç ile) rotalama gerçekleştirilir.

**Özellik 3:** Serim simetrik olsa da, bir  $k$  rotasının tersine çevrilmesi ile farklı bir amaç fonksiyonu değeri elde edilebilmektedir. Oysa GSP ve KARP'da, simetrik serimler ile çalışılırken rotanın tersine çevrilip uygulanması amaç fonksiyonu değerini etkilememektedir.

Ngueveu vd. (2010) tarafından üç farklı alt sınır elde etme tekniği önerilmiştir. Bu tekniklerin aktarılması amacıyla aşağıda, öncelikle küme ve gösterimler tanımlanmış ve alt sınır elde etme yöntemleri sırasıyla verilmiştir.

$G=(V,E,W)$  seriminde tanımlı,  $V=\{0, \dots, n, n+1\}$  ( $0$  ve  $n+1$  depo düğümler) kümesi, düğüm kümesini tariflesin. Böylece müşteriler kümesi  $V' = V \setminus \{0, n+1\}$  olarak gösterilebilir.  $E$  ayrıt kümesidir ve  $w_{ij} = w_{ji} \in W$ ,  $(i,j) \in E$  ayrıtının maliyetini ( $i$ 'den  $j$ 'ye ulaşım süresini) vermektedir.  $Q$  araç kapasitesini,  $R$  araç sayısını ve  $q_i$   $i \in V'$ , her bir müşterinin talebini tariflemektedir.

**Alt sınır 1 (sınırsız araç sayısı):** Özellik 1 ve Özellik 2'de belirtildiği gibi, araç sayısı artıyor iken amaç fonksiyonu değeri azalır, araç sayısı azalıyor iken çözüm değeri artar. Buradan hareketle,  $R \geq n$  iken her bir müşteri doğrudan depoya tek araç ile bağlandığında eniyi çözüm aşağıdaki denkleme göre elde edilmiş olur.

$$AS_1 = \sum_{j \in V'} w_{0j} \quad (3.5)$$

Diğer iki alt sınır teknikleri ise  $R < n$  durumu için tariflenmiştir.

**Alt sınır 2 (çözüm değeri tahmini):** Bu yöntemde göre  $G$  serimindeki ayrıtlar artış sırasına göre küçükten büyüğe dizilirler.  $w_e, e$ . en küçük ayrıt iken Alt sınır 2 aşağıdaki gibi elde edilir.

$$AS_2 = \sum_{e=1}^n \left( \left\lceil \frac{R + n - e - (n \bmod R)}{R} \right\rceil \right) w_e \quad (3.6)$$

**Alt sınır 3 (çözüm değeri tahmini):** Bu yöntemle  $AS_2$ 'ye göre daha gelişmiş bir alt sınır üretmektedir. Burada  $G$  serimindeki ayrıtlar, depo-müşteri arasında ya da müşteri-müşteri arasında olup olmama durumuna göre gruplandırılırlar ve  $AS_2$ 'deki gibi artış sırasına göre küçükten büyüğe dizilirler.  $w'_e$  depo-müşteri arasındaki  $e$ . en küçük ayrıtı iken,  $w''_e$  iki müşteri arasındaki  $e$ . en küçük ayrıtı gösterir. Alt sınır 3 aşağıdaki gibi elde edilir.

$$AS_3 = \sum_{e=1}^R \left( \left\lceil \frac{R + n - e - (n \bmod R)}{R} \right\rceil \right) w'_e + \sum_{e=1}^{n-R} \left( \left\lceil \frac{n - e - (n \bmod R)}{R} \right\rceil \right) w''_e \quad (3.7)$$

Ngueveu vd. (2010) BARP için bir matematiksel model ve çözüm yöntemi önermişlerdir. Bunlara izleyen kısımlarda ayrıntılı bir şekilde yer verilmiştir.

Ngueveu vd.'den (2010) sonra, Ribeiro ve Laporte (2012) literatürde BARP konusundaki ikinci çalışmayı yayınlamışlardır. Bu çalışma teorik olarak Ngueveu vd.'nin (2010) çalışmalarını desteklemektedir. Matematiksel model ya da alt sınırlar bakımından bir eklenti yapılmamış, çözüm yöntemi üzerinde durulmuştur. Bu çalışmada farklı çözüm yaklaşımı olarak Uyarlamalı Geniş Komşuluk Arama sezgiseli (Adaptive Large Neighborhood Search) kullanılmıştır. Sonuçlar Ngueveu vd.'nin (2010) sonuçlarıyla karşılaştırılmış ve 27 test probleminin 21 tanesinde daha iyi sonuçların elde edildiği gösterilmiştir.

Bu tez çalışmasının kapsamı, düğümlere ulaşma süreleri toplamının enküçüklendiği problem olarak belirlenmesine rağmen, literatürde benzer isimde çalışmalar yer alması nedeniyle her iki tip BARP problemine ait matematiksel modeller izleyen alt kısımlarda verilmiştir.



### 3.1. Yük Miktarına Bağlı Yakıt Tüketimli BARP Matematiksel Modeli

Yük miktarına bağlı yakıt tüketimli BARP modeli Kara vd. (2008) tarafından önerilmiştir. Matematiksel modelin tanımlanmasından önce aracın izleyeceği turun yönü hakkında bir açıklama yapılması gerekmektedir.

Klasik ARP'de dağıtım için oluşturulmuş bir rota, toplama rotası amacı ile de kullanılabilir (tersi de geçerli) ve serimin simetriklik ya da asimetriklik özelliği bunu etkilememekte, rotalar aynı amaç fonksiyonu değerini vermektedir. Oysa araçtaki yük miktarına bağlı yakıt tüketimli BARP'da serimin simetriklik durumuna göre eniyi dağıtım ve toplama rotaları ve değerleri birbirinden farklı olabilmektedirler.

Bu problemde, dağıtım yapan bir aracın yükü her düğüme uğradıktan sonra parçalı bir şekilde azalma eğilimindedir. Bu nedenle aracın rotasındaki sonlara doğru kümelenmiş ayrıtlarda oluşan maliyet daha az olabilmektedir. Tersisi de düşünülebilir. Toplama yapan bir aracın ise yük miktarı artış göstereceğinden aracın rotasındaki sonlara doğru kümelenmiş ayrıtlarda oluşan maliyetin daha fazla olabileceği durumlarla da karşılaşılabilir. Bu nedenle dağıtım için oluşturulan bir rota, toplama amaçlı kullanım için uygun olmayabilir.

Aşağıda sırayla, toplama ve dağıtım için Kara vd. (2008) tarafından yakıt tüketimli BARP için önerilen karma tamsayılı matematiksel model verilmiştir.

Modelde kullanılan çeşitli gösterimler aşağıda verilmiştir.

$G=(V,A)$  seriminde  $0$  depo düğüm olmak üzere, düğümler kümesi  $V=\{0,1,2,\dots,n\}$ , ayrıtlar kümesi ise  $A= \{ (i,j): i, j \in V, i \neq j \}$  olarak gösterilsin.

#### Parametreler:

$d_{ij}$   $i$  ve  $j$  düğümleri arasındaki uzaklık

$q_i$   $i$ . müşterinin talebi

$m$  homojen araç sayısı

$Q_0$  toplama durumunda depo ve ilk müşteri arasındaki, dağıtım durumunda son müşteri ile depo arasındaki akış miktarı (yüksüz araç ağırlığı)

$M$   $G$ 'deki herhangi bir ayırıt için enbüyük akış (yüksüz araç ağırlığı ve araç kapasitesi toplamı olarak alınabilir)

Karar Değişkenleri:

$$x_{ij} = \begin{cases} 1 & \text{herhangi bir araç } (i, j) \text{ ayırıtını kullanıyorsa} \\ 0 & \text{diğer} \end{cases}$$

$y_{ij} = (i, j)$  ayırıtı kullanıldığında  $i$  ile  $j$  arasında oluşacak akış miktarı

olmak üzere model aşağıdaki gibidir.

$$\text{Min} = \sum_{i=0}^n \sum_{j=0}^n d_{ij} y_{ij} \quad (3.8)$$

$$\sum_{i=1}^n x_{0i} = m \quad (3.9)$$

$$\sum_{i=1}^n x_{i0} = m \quad (3.10)$$

$$\sum_{i=0}^n x_{ij} = 1 \quad \forall j, (i, j) \in A \quad (3.11)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \forall i, (i, j) \in A \quad (3.12)$$

$$\sum_{j=0, j \neq i}^n y_{ij} - \sum_{j=0, j \neq i}^n y_{ji} = q_i \quad i = 1, 2, \dots, n \quad (3.13)$$

$$q_0 = 0 \quad (3.14)$$

$$y_{0i} = Q_0 x_{0i} \quad i = 1, 2, \dots, n \quad (3.15)$$

$$y_{ij} \leq (M - q_j)x_{ij} \quad \forall (i, j) \in A \quad (3.16)$$

$$y_{ij} \geq (Q_0 + q_i)x_{ij} \quad \forall (i, j) \in A \quad (3.17)$$

$$x_{ij} \in \{0,1\}, \quad y_{ij} \geq 0 \quad \forall (i, j) \in A \quad (3.18)$$

Amaç fonksiyonu (3.8),  $i$  ve  $j$  düğümleri arasındaki  $\forall (i, j) \in A$  ayrıtındaki akış ve aynı düğümler arasındaki uzaklığın çarpımları toplamının enküçüklenmesi biçimindedir. (3.9) ve (3.10),  $m$  adet aracı, depo düğüme giriş-çıkışına zorlayıp toplamda  $m$  adet aracın kullanmasını sağlamaktadır. Bu kısıtlar  $\leq$  biçiminde eşitsizlikler halinde de yazılabilir. (3.11) ve (3.12) klasik atama kısıtlarıdır ve her bir düğüme giriş varsa düğümden çıkış da olmasını sağlarlar. (3.13) depo düğümü haricinde bir düğüme giren akış ile ilgili düğümden çıkan akışın korunumunu sağlamaktadır. Bu kısıta göre her bir turun akış değişkenleri artan bir parçalı fonksiyon grafiği sergilemektedir ve ayrıca bu kısıtlar oluşabilecek alt turları engellemektedirler. Depo düğümünün talebi olmaması (3.14) kısıt tarafından sağlanmaktadır. Toplama probleminde, depo ile ilk müşteri arasındaki herhangi bir ayrıtın kullanıldığı bir turda, araç üzerinde yük olmayacağından dolayı, buradaki maliyetin, sadece ilgili ayrıtın uzunluğu ve aracın ağırlığı çarpımı şeklinde gerçekleşmesi gerekir. (3.15) bu durumu sağlamaktadır. (3.16)  $(i, j) \in A$  ayrıtı kullanılmadığında  $y_{ij}$  değerini 0 olmaya, aksi halde kalan kapasite kadar değer almaya zorlar. (3.17),  $y_{ij}$  için alt sınır oluşturur. (3.18) ise değişkenlere ait işaret kısıtlarıdır.

Dağıtım durumunda ise aşağıdaki model önerilmiştir.

$$Min = \sum_{i=0}^n \sum_{j=0}^n d_{ij} y_{ij} \quad (3.8)$$

$$\sum_{i=1}^n x_{0i} = m \quad (3.9)$$

$$\sum_{i=1}^n x_{i0} = m \quad (3.10)$$

$$\sum_{i=0}^n x_{ij} = 1 \quad \forall j, (i, j) \in A \quad (3.11)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \forall i, (i, j) \in A \quad (3.12)$$

$$\sum_{j=0, j \neq i}^n y_{ji} - \sum_{j=0, j \neq i}^n y_{ij} = q_i \quad i = 1, 2, \dots, n \quad (3.19)$$

$$q_0 = 0 \quad (3.14)$$

$$y_{i0} = Q_0 x_{i0} \quad i = 1, 2, \dots, n \quad (3.20)$$

$$y_{ij} \leq (M - q_i) x_{ij} \quad \forall (i, j) \in A \quad (3.21)$$

$$y_{ij} \geq (Q_0 + q_j) x_{ij} \quad \forall (i, j) \in A \quad (3.22)$$

$$x_{ij} \in \{0, 1\}, \quad y_{ij} \geq 0 \quad \forall (i, j) \in A \quad (3.18)$$

Dağıtım modeli, toplama modelinden (3.19), (3.20), (3.21) ve (3.22) numaralı kısıtlar aynı işlemlere sahip olmalarına rağmen gösterim olarak farklılıklar göstermektedir. (3.19), (3.13) ile aynı işleve sahiptir ve depo düğümü haricinde bir düğümden çıkan akış ile aynı düğüme giren akışın korunumunu sağlamaktadır. Dağıtım modelinde depoya dönen aracın son ayrıttaki maliyetinin pozitif değer alabilmesi için, üzerinde yük olmayacağından dolayı sadece aracın kendi ağırlığı ile son ayrıttın çarpımı şeklinde (3.20)'de verilmiştir. (3.21),  $(i, j) \in A$  ayrıtı kullanılmadığında  $y_{ij}$  değerini 0 olmaya aksi halde dağıtımdan sonraki akış kadar değer almaya zorlar. (3.22),  $y_{ij}$  için alt sınır oluşturur.

Toplama ve dağıtım modelleri farklı olmalarına rağmen serim simetrik ise, dağıtım ve toplama rotaları birbirlerinin yerlerine kullanılabilirler. Bunun ispatı Kara vd.'nin (2008) çalışmasında verilmiştir.

Her iki modelde de  $n^2+n$  0-1 tamsayı,  $n^2+n$  sürekli değişken,  $2n^2+6n+2$  tane kısıt vardır.

### 3.2 Dğümlere Ulaşma Süreleri Toplamının Enküçüklendiği BARP Matematiksel Modeli

Dğümlere ulaşma süreleri toplamının enküçüklendiği BARP matematiksel modeli Ngueveu vd.'nin (2010) çalışmalarında önerilmiştir. Bu model 2.0.28.0 versiyonlu GAMS IDE yazılımı kullanılarak CPLEX ile çözdürüldüğünde modelin feasible (olurlu) turlar oluşturmadığı görülmüş ve modelde ileride açıklanacak olan değişiklikler yapılmıştır. Ngueveu vd.'nin (2010) önerdiği model şu şekildedir:

$G=(V,E,W)$  tam bağlı yönsüz bir serim,  $0$  ve  $(n+1)$  düğümleri aynı depo düğüm olmak üzere  $V=\{0,1,2,\dots,(n-1),n,(n+1)\}$  tüm düğümler kümesi,  $V'=\mathcal{V}\setminus\{0,(n+1)\}$  müşteriler kümesi iken,

#### Parametreler:

$w_{ij}$   $i$  ve  $j$  düğümleri arasındaki ulaşım süresi

$q_i$   $i$ . müşterinin talebi

$R$  homojen araç sayısı

$Q$  araç kapasitesi

$T$  çok büyük bir sayı

#### Karar Değişkenleri:

$x_{ij}^k = \begin{cases} 1 & k. \text{ araç } (i, j) \text{ ayrıtını kullanıyorsa} \\ 0 & \text{dd.} \end{cases}$

$t_i^k = k. \text{ aracın } i. \text{ müşteriye varış zamanı}$

Model:

$$Min = \sum_{k=1}^R \sum_{i \in V'} t_i^k \quad (3.23)$$

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \quad \forall j \in V', \quad \forall k \in [1, \dots, R] \quad (3.24)$$

$$\sum_{k=1}^R \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V' \quad (3.25)$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^k q_i \leq Q, \quad \forall k \in [1, \dots, R] \quad (3.26)$$

$$\sum_{j \in V} x_{0j}^k = 1, \quad \forall k \in [1, \dots, R] \quad (3.27)$$

$$\sum_{i \in V} x_{i,n+1}^k = 1, \quad \forall k \in [1, \dots, R] \quad (3.28)$$

$$t_i^k + w_{ij} - (1 - x_{ij}^k)T \leq t_j^k, \quad \forall i \in V \setminus \{n+1\}, \quad \forall j \in V, \quad \forall k \in [1, \dots, R] \quad (3.29)$$

$$t_i^k \geq 0, \quad \forall i \in V, \quad \forall k \in [1, \dots, R] \quad (3.30)$$

$$x_{ij}^k \in \{0,1\}, \quad \forall i \in V, \quad \forall j \in V, \quad i \neq j, \quad \forall k \in [1, \dots, R] \quad (3.31)$$

Bu modelde klasik ARP'den farklı olarak göze çarpan kısım, amaç fonksiyonu ve alt tur engelleme kısıtlarıdır. Amaç fonksiyonu (3.23)  $n$  tane müşteriye varış zamanları toplamını enküçüklemektedir. (3.24)  $i$ . müşteriye giriş olduğunda, aynı müşteriden çıkış olmasını sağlamaktadır. (3.25) her müşteriye sadece bir kez uğranmasını garanti ederken araç kapasitelerinin aşılması (3.26) tarafından sağlanmaktadır. (3.27) ve (3.28) turların depoda başlayıp depoda tamamlanmasını garantilemekte, (3.29) müşterilere varış zamanını hesaplamakta ve aynı zamanda alt tur oluşmasını engellemektedir. (3.30) ve (3.31) ise değişkenlere ait işaret kısıtlarıdır.

Bu modelde karşılaşılan önemli bir durumun açıklanmasına gerek duyulmuştur. Model, kısıtları nedeniyle araçlar için kapalı turlar bulacak şekilde tasarlanmıştır. Öte yandan amaç fonksiyonuna, doğal olarak, bitiş düğümü olan  $x_{n+1}$ . düğüme ulaşma

zamanı dâhil edilmediği için amaç fonksiyonu değeri sanki açık rotalı bir probleme ait çözüm değeriymiş gibi algılanmaktadır. Bu nedenle de ilgili model BAARP gibi düşünülebilir.

Modele dair vurgulanması gereken bir başka önemli nokta ise olurlu (feasible) çözüm türetememesidir. Bu sorunu aşabilmek için aşağıda verilen (3.32) kısıtı modele eklenmiş ve (3.25) numaralı kısıt da (3.33)'de olduğu gibi değiştirilmiştir.

$$\sum_{j \in V} x_{(n+1)jk} = 0 \quad \forall k \quad (3.32)$$

$$\sum_{k=1}^R \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V', \quad i \neq j \quad (3.33)$$

Modele eklenen (3.32) kısıtı, bitiş düğümünden çıkış olmamasını garantilemektedir. Değiştirilerek düzenlenen (3.33) kısıtı ise yine her müşteriye sadece bir kez uğranmasını garanti etmektedir. Öncekinden tek farkı  $i \neq j$  koşulunun eklenmiş olmasıdır. Bu değişikliklerden sonra model, olurlu çözümler türetebilir ve böylece BAARP için kullanılabilir hale getirilmiştir.

Modelde kullanılan alltur engelleme kısıtları aynı zamanda  $t_i^k$  değerlerinin hesaplanmasına da yardımcı olmaktadır. (3.29)'un alt turları nasıl engellediğinin ispatı şu şekildedir:

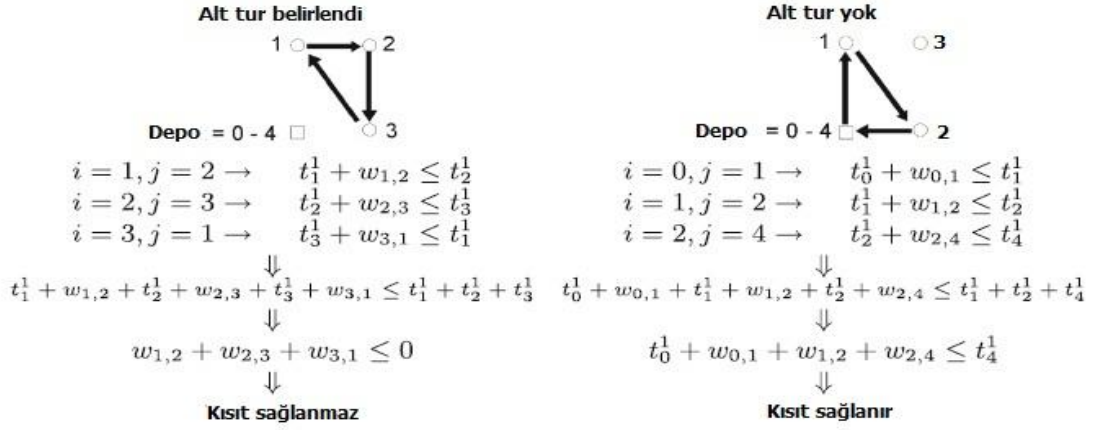
Herhangi bir  $k$  turunda,  $i \in \{n+1\}$   $j$ 'den önce geliyorsa  $x_{ij}^k=1$  değerini almıştır ve kısıt aşağıdaki hali alır.

$$t_i^k + w_{ij} \leq t_j^k \quad (3.34)$$

Bu durumda bir sonraki düğüme ulaşma zamanı kendinden önceki düğümden  $w_{ij}$  kadar fazla olacaktır ve bu artış nedeniyle alt turun oluşması mümkün olamaz. Eğer herhangi bir  $k$  turunda,  $i \in \{n+1\}$   $j$ 'den önce gelmiyorsa  $x_{ij}^k=0$  değerini almıştır ve kısıt aşağıdaki hali alır.

$$t_i^k + w_{ij} - T \leq t_j^k \quad (3.35)$$

$T$  çok büyük bir değer olarak kabul edildiğinden dolayı bu kısıt da sağlanmış olur. Böylece alt turlar engellenirken, bir turda birbirini takip eden düğümlere ulaşım süreleri hesaplanmış olur. Şekil 3.3’de alt turların nasıl saptandığı gösterilmiştir.



**Şekil 3.3.**  $n=3, k=1$  iken alt turların belirlenmesi (Ngueveu et al., 2010).

Şekil 3.3’de, Ngueveu vd.’nin (2010) önerdiği matematiksel modelin karar değişkenleri tanımına göre örnekler verilmiştir. Verilen iki farklı örneğe göre alt turların engellenmesi şu şekilde olmaktadır.

Birinci durumda araç depoya uğramadan 3 farklı düğümü gezmiştir ve alt tur oluşturmuştur. Bu durumda (3.29) kısıtı, Şekil 3.3’de alt turun belirlendiği gibi oluşmuştur ve taraf tarafa toplandığında,

$$w_{12} + w_{23} + w_{31} \leq 0 \quad (3.36)$$

yukarıdaki eşitsizliğini oluşturmaktadır. Bu eşitsizlik sağlanamayacağından dolayı böyle bir durum (alt tur) oluşamaz. Böylece alt turlar engellenir.

Aynı şeklin 2. örneğinde ise alt turun oluşmadığı normal duruma yer verilmiştir. Araç depodan çıkıp, 1 ve 2 düğümlerini gezmiş ve depoya dönmüştür. (3.29) kısıtı yine taraf tarafa toplandığında,



$$t_0^1 + w_{01} + w_{12} + w_{24} \leq t_4^1 \quad (3.37)$$

eşitsizliğini oluşturmaktadır. Bu eşitsizlik sağlandığından kısıt uyumlu bir şekilde çalışmaktadır.

Hem yakıt tüketimli, hem de düğümlere ulaşma süreleri toplamının enküçüklendiği BARP, NP-zor sınıfında yer almaktadır (Kara vd., 2008; Ngueveu, et al.,2010). Bu nedenle büyük boyutlu bir problemin matematiksel modellerle kesin çözümünün bulunması mümkün olamayabilmektedir. İzleyen bölümde, NP-zor problemler için literatürde sıkça kullanılmış ve BAARP'da performansları incelenmiş ileri sezgisel çözüm yaklaşımlarına yer verilmiş, bu yöntemlerin etkinliği üzerinde yapılan araştırmalar aktarılmıştır.

#### 4. BAZI İLERİ SEZGİSEL ÇÖZÜM YAKLAŞIMLARI

Yöneylem araştırması alanındaki problemler polinom ve üstel zamanda çözülebilen problemler olarak ikiye ayrılabilir. Polinom zamanda çözülebilen problemler için sorun yoktur ancak üstel zamanda çözülebilen problemlerde ilgilenilen parametre değeri büyüdükçe çözümün makul sürede bulunması zorlaşır ve hatta imkânsızlaşabilir. NP-zor olarak isimlendirilen bu problem grubunda büyük boyutlu durumlar için sezgisel algoritma kullanmak, günümüzde neredeyse bir zorunluluktur.

Gerçek hayat problemlerinde, hızlı bir şekilde çözüm üretebilmek çoğu zaman kritiktir. Örneğin imalat yapan bir işletmenin, işleri amaca yönelik olarak eniyi şekilde çizelgelemesi problemi için gereken bekleme süresinin saatler biriminde olması, işletmenin imalat yapmadan beklemesi anlamına gelmektedir. Bu durum da büyük maddi kayıplara sebep olur. Benzer şekilde çeşitli noktalara hammadde ya da ürün ulaştırmak amacıyla araç rotalama probleminin eniyi olarak çözülmesini bekleyen bir işletme için, ürünleri zamanında ulaştıramama olasılığı oluşmakta bu da tedarik zincirini olumsuz şekilde etkileyebilmektedir.

Bunun gibi sorunları aşabilmek için sezgisel algoritmalar önerilmiştir. Sezgisel algoritmalar bilgisayarlı problem çözümlerinde birkaç farklı yaklaşımı kullanan, zeki arama stratejileri tabanlı metotlardır (Pearl, 1984).

Sezgisel algoritmalar eniyi çözüm garantisi olmadan makul sürelerde iyi çözümler üretebilmek amacıyla kullanılmaktadırlar. Sezgisel yaklaşımlar genellikle probleme özel olarak türetilmişlerdir. Örneğin Kazanım Algoritması (Savings Algorithm) (Clarke and Wright, 1964) araç rotalama problemi için önerilmiştir. Üzerinde çeşitli değişiklikler yapılarak benzer yapıdaki problemlere uyarlanabilmektedir.

Öte yandan, ileri sezgisel algoritmalar (meta-heuristic) olarak da adlandırılan ve yapay zekâ teknikleri arasında yer alan algoritmalar, sezgisel yaklaşımı bir adım ileriye taşıyıp, yaklaşımı probleme özgü olmaktan çıkarmaktadırlar. Bir ileri-sezgisel

algoritma, uygulamada çeşitli değişikliklerle çok farklı yapılarıdaki problemlere uyarlanabilmektedirler. Bu durum, bu yaklaşımların yaygın olarak kullanılma sebepleri arasında gösterilebilir.

İleri-sezgisel algoritmalar genel olarak tek noktalı ve sürü tabanlı olmak üzere iki ana başlık altında sınıflandırılabilirler. İzleyen bölümde tek noktalı ileri-sezgisellerden Yasaklı Arama Algoritması, sürü tabanlı ileri-sezgisel algoritmalarından ise Genetik Algoritma ve Parçacık Sürüsü Optimizasyonu Algoritmaları hakkında bilgi verilmiştir.

#### **4.1. Yasaklı Arama Algoritması**

Yasaklı Arama Algoritması (YAA) (Tabu Search), eniyileme problemlerinin çözümünde kullanılan ve yerel eniyilere takılmayı engelleyen yüksek dereceden bir sezgisel algoritmadır (Glover, 1989, 1990a, 1990b).

YAA araştırmalarda sıkça kullanılan tek noktalı ileri sezgisel algoritmalar arasında gösterilebilir ve çok sayıda kombinatoriyel probleme uygulanabildiği gibi, doğrusal olmayan sürekli problemlere de uyarlanabilmektedir.

Çalışmada uygulanan YAA'nın aktarılmasından önce, bu konuda kullanılan bazı terimler aşağıda verilmiştir.

***Ardıştırma Sayısı (# of iterations)***: Algoritmanın toplam koşturma sayısı.

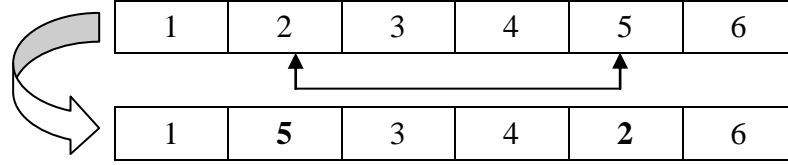
***Başlangıç Çözüm (initial solution)***: Algoritmanın birinci ardıştırmada kullanılan çözümdür. İzleyen diğer tüm çözümlere bu çözümdeki bilgilerin kullanımı ile ulaşılır. Bu nedenle başlangıç çözüm kritiktir. Başlangıç çözüm tamamen rassal olarak oluşturulabileceği gibi, daha iyi bir noktadan başlamak amacıyla, farklı bir tekniğin çıktısı olan bir çözüm de başlangıç çözüm olarak kabul edilebilir.

***Komşuluk sayısı (# of neighborhood)***: Çözüm uzayında, bulunan mevcut noktanın komşuluğunda incelenecek aday çözüm (aday nokta) sayısı.

**Aday Çözüm (candidate solution):** Çözüm uzayında, bulunulan mevcut noktadan rassal bir hareketle elde edilmiş çözüm.

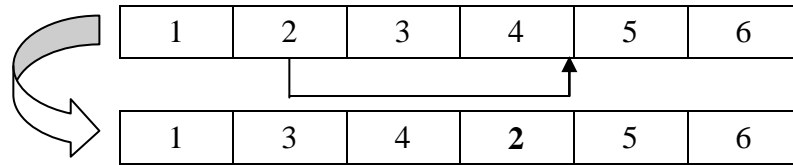
**Hareket (move):** Çözüm uzayında bir noktadan başka bir noktaya ulaşmak için kullanılan mekanizmadır. Başka bir ifadeyle başka bir çözüm elde etme tekniğidir. Literatürde günümüze kadar uygulanmış pek çok *hareket* mevcuttur. Bunlardan sıklıkla kullanılanları aşağıda verilmiştir.

**İkili Yer Değiştirme (Swap):** Örnek olarak bir tur tarifleyen permütasyondaki iki farklı düğümün birbirleri ile yer değiştirmesidir. İkili yer değişikliği çeşitli tekniklerde tam sayımlamanın gerektirdiği kadar yapıyor iken, literatürde genellikle yer değiştirecek düğümlerin rassal olarak seçilmesi benimsenmiştir. Şekil 4.1’de buna ilişkin bir şekil verilmiştir.



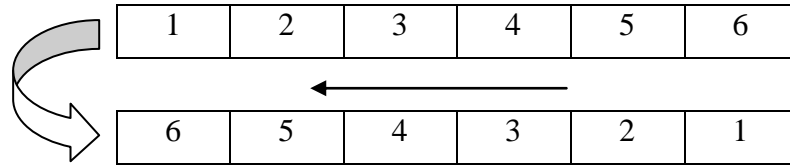
**Şekil 4.1.** İkili yer değiştirme.

**Araya Sokma (Insertion):** Bu teknikte rassal seçilen bir düğüm yine rassal seçilen bir aralığa yerleştirilir. Şekil 4.2’de buna ilişkin bir örnek verilmiştir.



**Şekil 4.2.** Araya sokma.

**Tersine çevirme (Inversion):** Bir turun tersine çevrilmesi işlemidir. Şekil 4.3’de buna ilişkin bir örnek verilmiştir.



**Şekil 4.3.** Tersine çevirme.

**Yasak Statüsü (tabu status):** Yapılacak hareketin yasaklı ya da yasaksız olduğu bilgisini barındıran değişken.

**Yasak Süresi (tabu time):** Yapılmış bir hareketin yasaklı statüsünde kalacağı zaman dilimi (ardıştırmaya sayısı).

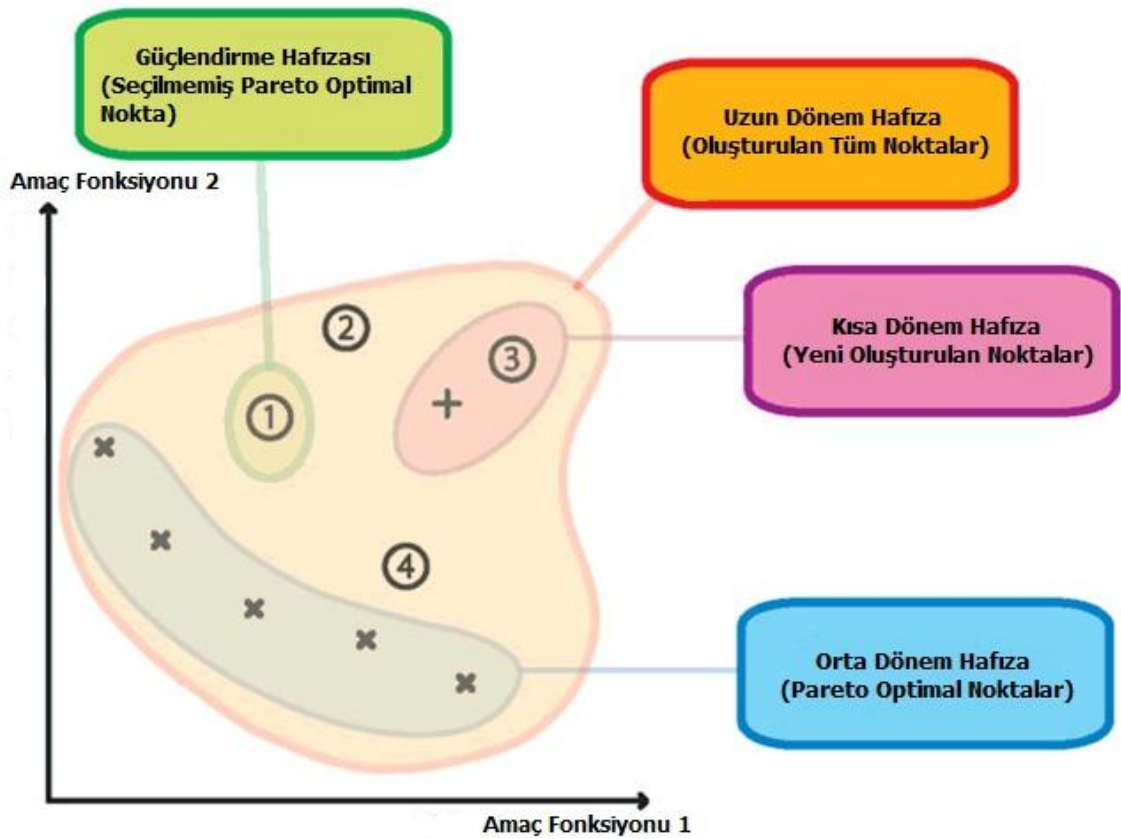
**Yasak Listesi (tabu list):** Yapılan hareketlerin (yasak statüsü kazanmış hareketlerin) ardışık olarak hafızada tutulduğu liste.

**Yasak Listesi Kapasitesi (tabu list capacity):** Yasak listesine dâhil edilebilecek en fazla eleman sayısı.

**Algoritma Eniyi (incumbent):** Algoritmanın herhangi bir ardıştırmaya kadar bulabildiği eniyi çözümdür. Problemin eniyi çözümü olmak zorunda değildir.  $z^*$  problemin eniyi çözümü iken, enbüyükleme problemi için  $incumbent \leq z^*$ , enküçükleme problemi için  $incumbent \geq z^*$  olur.

YAA’da genel olarak kısa dönemli (short-term) ve uzun dönemli (long-term) olmak üzere iki farklı hafıza yapısı vardır. Bazı kaynaklarda bu sınıflandırma arttırılabilmektedir. Şekil 4.4’de buna ilişkin bir gösterim verilmiştir. Kısa dönemli hafıza, çeşitli yasaklamalar ve kontrollere bağlı olarak bulunulan noktanın komşuluğu ile ilişkilendirilir. Yasak listesi kapasitesinin küçük, aday çözüm sayısının büyük

olması, algoritmanın toplam işlem süresinde kısa dönemli hafızadan kaynaklı işlem süresinin artmasına sebep olabilmektedir. Tersisi durumda ise kısa dönemli hafıza kaynaklı gerekli işlem süresi azalma eğiliminde olmasına rağmen, bulunulan noktanın komşuluklarının yeterli hassasiyetle incelenememe ihtimali algoritmanın başarısını etkileyebilmektedir.



Şekil 4.4. Yasaklı Arama Algoritmasındaki hafıza tipleri (<https://www-edc.eng.cam.ac.uk/projects/tabusearch/>).

Öte yandan ardıştırma sayısı direkt olarak uzun dönemli hafıza ile ilişkilendirilmektedir. Uzun dönem hafıza ile algoritmanın, başlangıcından sonuna kadar elde edilen çözüm değerlerine işaret edilmektedir.

Bu iki farklı hafıza tipinin dengelenbilmesi ve algoritmanın makul çözüm sürelerinde iyi çözümler üretebilmesi, algoritmada kullanılan parametrelerin uygun değerlerinin belirlenebilmesi ile sıkı bir şekilde ilintilidir. Bu sebeple çeşitli sonuçlara ilişkin yorumlar ileri sürülmeden önce, genellikle algoritma için farklı parametreler altında denemeler yapılır (trial-run). Bu denemelerden elde edilen verilere göre çeşitli istatistiksel tekniklerin yardımıyla parametrelerin eniyi değerleri belirlenebilmektedir. Ancak ileri-sezgisel algoritmaların rassallık özelliği nedeniyle aynı parametreler altında farklı sonuçlar elde edilmesi de olası bir durumdur.

Algoritmanın tüm adımları aşağıda ayrıntılı olarak verilmiştir.

**Adım 1:** Ardıştırma sayacını ( $k$ ), tabu aşma kuralını, tabu listesi uzunluğunu, tabu listesinde tutma süresini, aday listesi sayacını ( $i$ ) ve hareket mekanizmasını (araya sokma veya ikili yer değiştirme gibi) belirle.

**Adım 2:**  $algoritma\_eniyi = baslangic\_cozum$ ,  $k=0$ .

**Adım 3:**  $k=k+1$ ,  $i=0$ .

**Adım 4:**  $yasak\_sayaci=0$ ,  $k$ . ardıştırma için aday çözümler oluştur.

**Adım 5:**  $i=i+1$ .

**Adım 6:**  $i$ . aday çözümün değerini hesapla,  $i=komşuluk\_sayısı$  ise 7. adıma geç, aksi halde 5. adıma dön.

**Adım 7:** Mevcut komşuluklardan eniyi amaç fonksiyonu değeri vereni hesapla ( $p$ ).

**Adım 8:**  $komşuluk(p)$ ,  $algoritma\_eniyi$ 'den daha iyi bir sonuç veriyor ise  $algoritma\_eniyi=komşuluk(p)$ .  $p$ . aday çözümü elde etmede kullanılan hareket yasak ise hareketi yap, yasak- süresini yeniden başlat, değil ise hareketi yap, yasak listesine ekle ve 9. adıma geç.

$aday\_çözüm(p)$   $algoritma\_eniyi$ 'den daha iyi bir sonuç vermiyor ve hareket yasaklı ise hareketi yapma,  $yasak\_sayaci=yasak\_sayaci+1$ ,  $komşuluk(p)$ 'yi komşuluk listesinden çıkar adım 7'ye dön.

$aday\_çözüm(p)$ ,  $algoritma\_eniyi$ 'den daha iyi bir sonuç vermiyor ve hareket yasaklı değil ise hareketi yap ve yasak listesine ekle.

**Adım 9:** Tüm hareketlerin  $yasak\_sürelerini$  1 azalt ve  $yasak\_süresi$  tamamlanan bir hareket varsa, ilgili hareketi yasak listesinden çıkar.

$yasak\_sayaci=komşuluk\_sayısı$  ise veya  $k=ardıştırma\_sayısı$  ise dur. Eniyi çözümü listele.

Yukarıda genel hatlarıyla anlatılan YAA bu çalışmadaki uyarlaması ayrıntılı olarak 4.4 bölümünde verilmiştir.

## 4.2. Genetik Algoritma

Günümüzde, doğada var olan çeşitli organizmaların yapılarının ve işleyişlerinin, bilgisayar ortamındaki benzetimi ile çözümü zor olan çeşitli problemlere çözüm üretilebilmektedir. Genetik algoritmalar da sözü edilen doğada var olan yapılardan, Charles Darwin'in evrim teorisini temel almış, sürü tabanlı ileri-sezgisel algoritmalarındandır.

GA, doğal seçilim ve doğal genetik mekanizmalarını temel alan güçlü ve yaygın olarak kullanılan bir rassal arama yöntemidir (Goldberg, 1989). Holland (1975) yılında ilk olarak literatüre kazandırılan GA, çözümü zor olan ve çözüm süresi problem boyutundaki artışa bağlı olarak üstel artış gösteren problemlere sıklıkla uygulanmaktadır. Literatürdeki sonuçlara bakıldığında ise GA'ların kombinatoryel eniyilemede, genellikle başarılı (eniyi ya da eniyiye yakın) olabildiği gözlenebilmektedir. GA'lar ayrıca sürekli ve doğrusal olmayan eniyileme problemlerine de uygulanabilmektedir (Gen and Cheng, 1997).

Aşağıda GA ile ilgili sıkça kullanılan terimlerin anlamları verilmiştir.

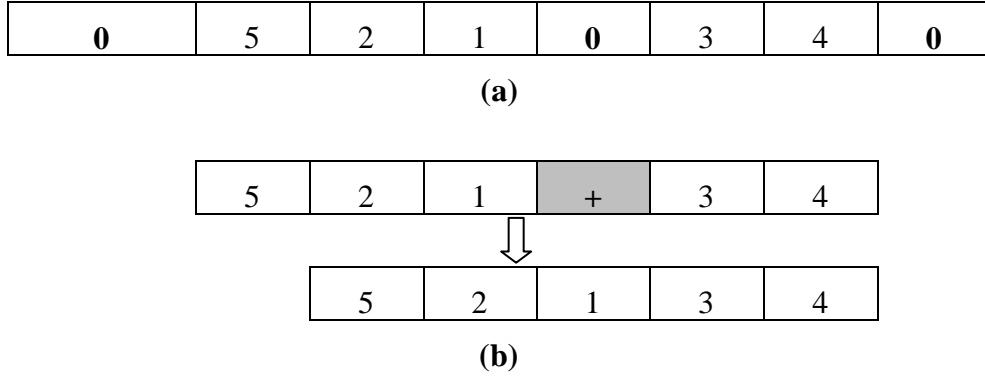
**Kromozom (*chromosome*):** Bir çözüme karşı gelen bireydir.

Literatürde farklı gösterim şekilleri mevcuttur. Çalışmanın kapsamıyla ilintili olması açısından KARP için sıklıkla kullanılan iki farklı kromozom gösterimi aşağıda, bir örnek üzerinde gösterilmiştir.

Birinci kromozom gösteriminde tur ayracı (tour splitter) kullanılmıştır. Bu gösterime göre her iki araç arası bir aracın rotasını oluşturmaktadır. Şekil 4.5 (a)'da bu gösterime ilişkin bir kromozom örneği verilmiştir.



Diğer gösterimde ise araçların rotaları ayrı permütasyonlar olarak elde edilip, bu turlar birbirine eklenerek büyük tur (gigantic tour) oluşturulur. Şekil 4.5 (b)'de bu gösterime ilişkin bir kromozom örneği verilmiştir.



**Şekil 4.5. a)** Tur ayrıçlı gösterim, **b)** Büyük tur gösterimi.

Hem (a) hem (b)'ye göre birinci araç depodan çıkıp 5-2-1 rotasını izleyip tekrar depoya dönecektir. İkinci araç ise yine depodan çıkıp 3-4 rotasını izleyip depoya dönecektir.

Bu çalışmada (b)'deki gösterim tercih edilmiştir.

**Popülasyon (population):** Kromozomların oluşturduğu topluluktur. Başlangıç popülasyonu rassal oluşturulabileceği gibi çeşitli algoritmalar yardımıyla daha iyi bir çözüm setinden de oluşabilir.

**Popülasyon büyüklüğü:** Popülasyondaki kromozom sayısıdır.

**Gen (gene):** Kromozomun her bir hücreğini temsil eder.

**Alel (allele):** İlgili genin değeridir.

**Lokus (locus):** Bir genin kromozomdaki pozisyonudur.

**Nesil (generation):** Ardıştırmaya anlamına gelmektedir.

**Uyum (fitness):** Bir kromozomun amaç fonksiyonu değeridir.

**Çaprazlama (cross-over):** İki kromozom arasındaki bilgi aktarımıdır. Var olan bilgi kullanılarak yeni bilgiye erişim sağlanır (exploitation).

**Çaprazlama oranı (cross-over rate):** Kromozomların çaprazlanmasına ilişkin kararın verilmesini sağlayan bir parametredir.

**Ebeveyn (parent):** Çaprazlanacak kromozomlara verilen addır.

**Çocuk (offspring):** Çaprazlama sonrası oluşan kromozomdur.

**Mutasyon (mutation):** Gen(ler)de meydana gelen değişikliktir. Bu operatörle henüz var olmayan bilgiye erişim amaçlanır (exploration).

**Mutasyon oranı (mutation rate):** Gen(ler)de mutasyon yapılmasına ilişkin kararın verilmesini sağlayan bir parametredir.

**Seçme ve kopyalama (selection & copy):** Kromozomların uyum değerlerine göre bir seçim prosedürü ile izleyen nesle aktarılması işidir.

**Elitizm (elitism):** Eniyi uyum değerine sahip kromozomun izleyen nesle doğrudan kopyalanmasıdır.

Yeni popülasyon oluşumunda kullanılan, literatürde kayıtlı çeşitli seçim teknikleri mevcuttur. Rulet tekeri ve turnuva seçim yöntemleri bunlardan sıkça kullanılanlarıdır. Rulet tekeri tekniğinde, kromozomların uyum değerleri normalize edildikten sonra, bu değerlerin birikimli toplamı yardımıyla her bir kromozomun dahil olduğu  $[0,1]$  aralığında aralıklar elde edilir. Ardından her bir seçim işlemi için rassal sayı türetilir. Rassal sayının isabet ettiği aralığa ait kromozom sonraki nesle kopyalanır. Böylece daha iyi uyum değerine sahip kromozomun sonraki nesle kopyalanma olasılığı göreceli olarak yüksek olur.

Turnuva (tournament) seçim yönteminde ise turnuva büyüklüğü (turnuvaya girecek kromozom sayısı) kadar rassal olarak belirlenen adaylar arasından eniyi uyum değerine sahip olan kromozom, izleyen nesle aktarılır. Her iki seçim yöntemi de Gen ve Cheng'in (1997) kitabında ayrıntılı olarak anlatılmıştır.

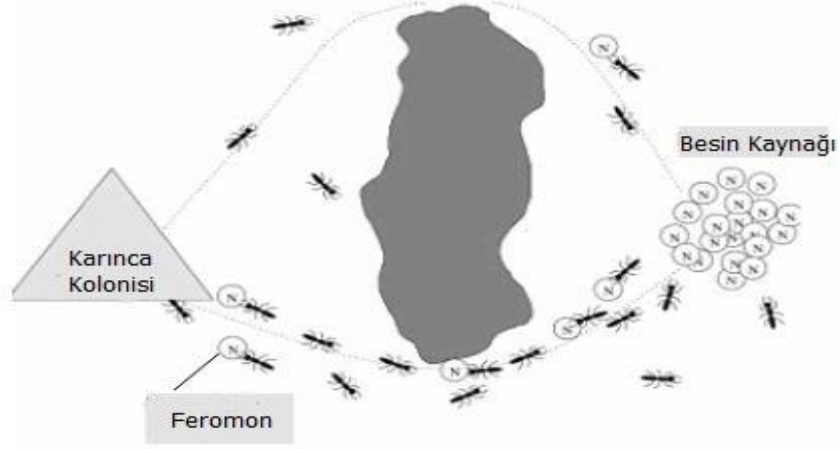
Bu çalışmada turnuva seçim yöntemi uygulanmış ve turnuva büyüklüğü iki olarak seçilmiştir. GA'nın adımları ise aşağıda verilmiştir.

- Adım 1:** Başlangıç parametre değerlerini ve başlangıç popülasyonu belirle,  $t=0$ .
- Adım 2:** Çaprazlanacak kromozomları ve mutasyona uğrayacak genleri belirleyip, çaprazlama ve mutasyon yap.
- Adım 3:** Tüm popülasyonun uyum değerlerini hesapla, eniyi kromozomu ve uyum değerini sakla.
- Adım 4:** Seçim işlemini gerçekleştir ve izleyen nesle kopyala.
- Adım 5:**  $t=t+1$
- Adım 6:** Durma kriteri sağlanıyorsa dur. Eniyi çözümü ve uyum değerini yazdır. Aksi halde Adım 2'ye dön.

### 4.3. Parçacık Sürüsü Optimizasyonu

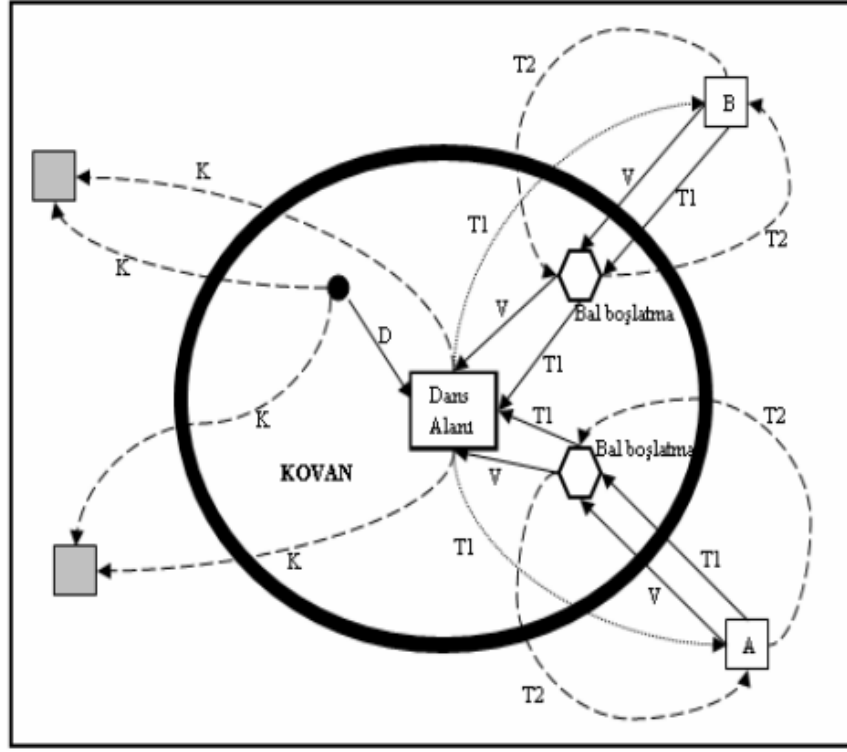
Sürü zekâsı ile hareket eden organizmalarda, kolektif zekâ ve dolayısıyla bireyler arasında gerçekleştirilen bilgi aktarımı oldukça önemli bir paya sahiptir. Bu aktarım, farklı organizmalarda farklı şekillerde gerçekleştirilebilir.

Örneğin karınca kolonilerinde, karıncalar besin kaynaklarına, diğer karıncaların etrafa yaydıkları feromon kokusunun daha fazla duyulduğu yere doğru hareket ederek ulaşmaya çalışırlar. Başka bir ifadeyle besin kaynağı civarındaki karınca sayısı zaman ilerledikçe artmaya başlar. Kaynak tüketildiğinde ise karıncalar tekrar feromon salınımını dikkate alarak farklı besin kaynaklarına yönelme eğilimindedirler. Karıncaların besin kaynağı arayışındaki davranışlar Şekil 4.6'da gösterilmiştir. Bu davranış mekanizmasını temel alarak, çeşitli zor problemlere çözüm üretebilmek amacıyla Karınca Kolonisi Optimizasyonu (Dorigo et al., 1991; Dorigo et al., 1996) literatüre kazandırılmıştır.



**Şekil 4.6.** Bir karınca kolonisinin besine erişimi ile ilgili bir gösterim (<http://www.ankurkhetrapal.com/research/proteus.htm>).

Benzer yaklaşım arı kolonilerine de uygulanmıştır. Arı kolonilerinde ise bilgi aktarımı kovan (nest) içerisinde yapılan salınım dansı (waggle dance) ile gerçekleştirilir. Bu dans ile besin kaynağının miktarı ve yeri hakkında arılar arasında bilgi aktarımı olur ve kaynaktaki besin miktarı yüksek ise o kaynağa daha fazla arı atanır. Şekil 4.7’de bu durum özetlenmiştir.



**Şekil 4.7.** Arı kolonisinin ile ilgili bir gösterim  
([http://www.emo.org.tr/ekler/5a2d64e6d533a97\\_ek.pdf](http://www.emo.org.tr/ekler/5a2d64e6d533a97_ek.pdf))

Bu şekilde *A* ve *B* olmak üzere iki farklı besin kaynağı vardır. *T2* ile gösterilen rotada besin kaynağından kovana besin taşınır. Ancak besin miktarı fazla ise *T1* rotaları ile dans alanında salınım dansı yapılır ve bu bilgi boşta olan diğer arılara aktarılarak o besin kaynağına daha fazla arı atanır. Besin kaynağı tükendikten sonra ise *V* rotasıyla besin toplayan arılar boşta kalarak salınım dansı izlemeye başlarlar. Son olarak *K* ile gösterilen rota ile kâşif arılar (scout bees), rassal olarak besin kaynağı ararlar. Bulunan besin kaynaklarının yerlerini ve miktarlarını ise yine salınım dansı ile boşta olan arılar ile paylaşırlar. Arıların besin arama konusundaki bu mekanizması bilgisayar ortamına uyarlanmış ve Arı Algoritması (Bees Algorithm) (Karaboğa, 2005; Pham et al., 2006; Karaboğa ve Baştürk, 2007a; Karaboğa ve Baştürk, 2007b; Karaboğa ve Baştürk, 2007c; Karaboğa ve Akay, 2009; Karaboğa, 2010) adını almıştır.

Benzer şekilde balık ve kuş sürülerinde de aynı durum geçerlidir. Bu sürülerde de besin kaynağına erişim kolektif zekânın bir ürünüdür.

PSO, kuş ve balık sürüleri gibi topluca hareket eden sosyal organizmaların davranışlarının benzetimini yapan, sürü zekâsı tabanlı algoritmalardan bir tanesidir (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995; Eberhart et al., 1996; Kennedy, 1997; Angeline, 1998; Eberhart, et al., 2001).

PSO ile ilgili ayrıntılı bilgi verilmeden önce aşağıda önemli olduğu düşünülen bazı tanımlar verilmiştir.

**Parçacık (*particle*):** GA'daki kromozoma karşılık gelir ve her bir parçacık bir çözümü ifade eder.

**Sürü (*swarm*):** GA'daki popülasyona karşılık gelir ve parçacıkların oluşturduğu sürüyü ifade eder.

**Sürü büyüklüğü (*swarm size*):** Sürüdeki parçacık sayısıdır. GA'daki popülasyon büyüklüğüne karşılık gelir.

**Atalet bileşeni (*inertia component*):** Parçacık hızını azaltan yerel çözümleri hassaslaştıran bileşendir.

**Atalet katsayısı (*inertia coefficient*):** Atalet bileşenin şiddetini ayarlamakta kullanılan katsayıdır.

**Bireysel bileşen (*cognitive component*):** Parçacığın geçmiş tecrübelerini sakladığı ve bu tecrübelerden faydalanarak hızını ayarladığı bileşendir.

**Bireysel katsayı (*cognitive coefficient*):** Bireysel bileşenin şiddetini ayarlamakta kullanılan katsayıdır.

**Sosyal bileşen (*social component*):** Tüm sürünün eriştiği eniyi çözümün bireyler arasında kullanılarak ilgili parçacığın hızının ayarlandığı bileşendir.

**Sosyal katsayı (*social coefficient*):** Sosyal bileşenin şiddetini ayarlamakta kullanılan katsayıdır.

**Bireysel eniyi (*individual best*):** Bir parçacığın bireysel olarak herhangi bir  $t$ . ardıştırmaya kadar erişebildiği eniyi çözümdür.

**Algoritma eniyi (*incumbent*):** Sürünün herhangi bir  $t$ . ardıştırmaya kadar erişebildiği eniyi çözümdür.

**Hız vektörü (velocity vector):** herhangi bir  $t$  anında parçacık bazında saklanabildiği gibi sürü için de saklanabilen,  $n$ , problemdeki değişken sayısı,  $S$ , sürü büyüklüğü iken sırasıyla  $1 \times n$  ya da  $S \times n$  (matris) boyutlarında olabilen parçacıkların  $(t+1)$ . ardıştırmadaki konumlarının hesaplanmasında kullanılan vektör ya da matris.

**Konum vektörü (position vector):** Problemdaki değişkenlerin tanım aralığı içerisindeki koordinatlarının saklandığı vektördür. Hız vektöründe olduğu gibi sürü bazında ya da parçacık bazında da tutulabilir.

PSO ilk olarak Kennedy ve Eberhart (1995) tarafından, sürekli doğrusal olmayan eniyileme (continuous nonlinear optimization) için önerilmiştir. Bu teknikle parçacıkların hızını ve konumunu ayarlamak için aşağıdaki denklemler kullanılır.

$$V_{i(t+1)} = wV_{i(t)} + c_1\Omega_1[X_{iind(t)}^* - X_{i(t)}] + c_2\Omega_2[X_{gl(t)}^* - X_{i(t)}] \quad (4.1)$$

$$X_{i(t+1)} = X_{i(t)} + V_{i(t+1)} \quad (4.2)$$

Bu denklemlerde,  $V_{i(t)}$   $i$ . parçacığın  $t$  anındaki hızını,  $X_{i(t)}$   $i$ . parçacığın  $t$  anındaki konumunu göstermektedir.  $w$ ,  $c_1$  ve  $c_2$  sırasıyla atalet katsayısını, bireysel katsayıyı ve sosyal katsayıyı temsil etmekte,  $\Omega_i$  ise  $[0,1]$  aralığında sürekli düzgün dağılmış (continuous uniform distributed) rassal sayıları göstermektedir.  $X_{iind(t)}^*$   $i$ . parçacığın  $t$  anına kadar bulabildiği eniyi çözüm,  $X_{gl(t)}^*$  ise sürü tarafından  $t$  anına kadar bulunabilen eniyi çözümdür.

(4.1)'den görülebildiği gibi  $i$ . parçacık  $(t+1)$ . anındaki hızını  $t$  anındaki hızı, bireysel eniyi ve algoritma eniyiye göre ayarlamaktadır.  $w$ ,  $c_1$  ve  $c_2$  katsayıları bu hızın hesaplanmasındaki ağırlıklardır. Örnek olarak yüksek bir  $c_2$  değeri parçacıkların hızlı bir şekilde *algoritma eniyi* etrafında toplanmasına sebep olurken, yüksek bir  $c_1$  değeri parçacıkların birbirinden kopuk olarak arama yapması anlamına gelmektedir. Benzer şekilde  $w$  değerinin yüksek seçilmesi parçacıkların yerel aramayı (local search) daha yüzeysel yapıp hızlıca eniyiye yakınsamasına, düşük seçilmesi ise yerel aramanın hassas yapıp parçacıkların yavaş hareket etmesine sebep olmaktadır.

(4.2)'de ise  $i$ . parçacığın  $(t+1)$ . artırma için olması gereken hızı ile  $t$  anındaki konumu toplanarak, ilgili parçacığın  $(t+1)$  anındaki konumu elde edilmektedir.

PSO'da kullanılan parametre sayısının fazla olması algoritmanın başarısı açısından, önemli bir soruna işaret etmektedir. Bu nedenle, çeşitli denemeler sonrasında bu parametrelerin genellikle belirli aralıklarda kabul edilmesi, literatürde yaygınlık kazanmıştır. Örnek olarak  $w \in [0, 1.2]$ ,  $c_1 \in [0,2]$ ,  $c_2 \in [0,2]$  olarak kabul edilmektedir. Ayrıca Clerc ve Kennedy'ye (2002) göre  $c_1+c_2=4.1$  olarak kabul edilmiştir. Tüm bu varsayımlar altında, iyi sonuçlar türetebilmek için, parametrelerin olması gereken seviyelerinin belirlenmesi kolaylaşabilmektedir.

Hız vektörünün hesaplanmasında rassal bir parçacığın da etkisinin katıldığı çalışmalar mevcuttur. Aşağıdaki denklem (4.3), (4.1) denklemine  $c_3\Omega_3[X_{rnd} - X_{i(t)}]$  ifadesinin eklenmesi ile elde edilmiştir.  $X_{rnd}$  sürüdeki rassal bir parçacığın konum vektörüdür.  $c_3\Omega_3$  ise (4.1)'deki gibi kullanılmaktadır.

$$V_{i(t+1)} = wV_{i(t)} + c_1\Omega_1[X_{iind(t)}^* - X_{i(t)}] + c_2\Omega_2[X_{gl(t)}^* - X_{i(t)}] + c_3\Omega_3[X_{rnd} - X_{i(t)}] \quad (4.3)$$

Kesikli bir çözüm uzayı düşünüldüğünde, sürekli doğrusal olmayan eniyileme için önerilmiş PSO'nun uygulanması mümkün olamamaktadır.

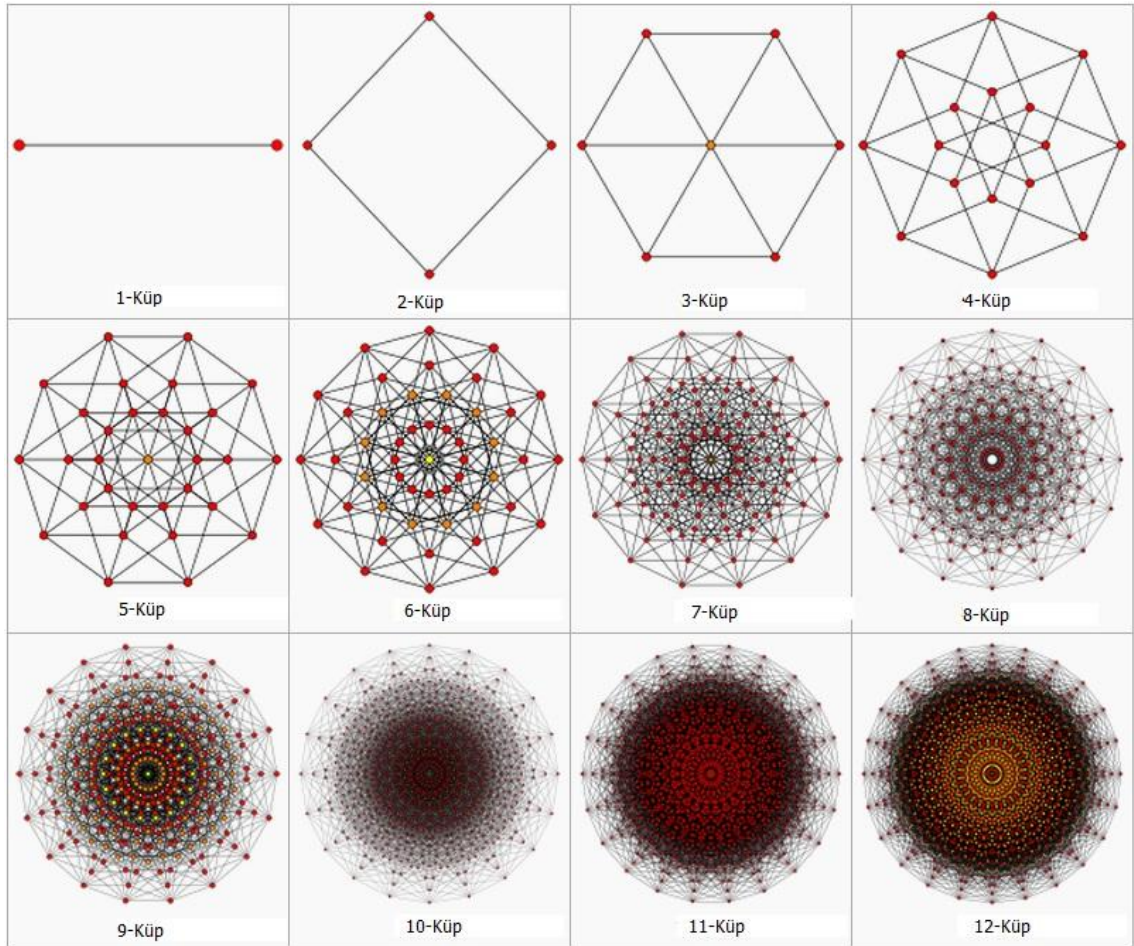
Örnek olarak, problemin karar değişkenleri  $x_i \in (0,1)$  ve  $i = 1,2,3$  iken, 4 parçacıktan oluşan sürü aşağıdaki gibi verilmiş olsun.

**Çizelge 4.1.** 0-1 tamsayılı değişkenlerden oluşan bir sürü örneği

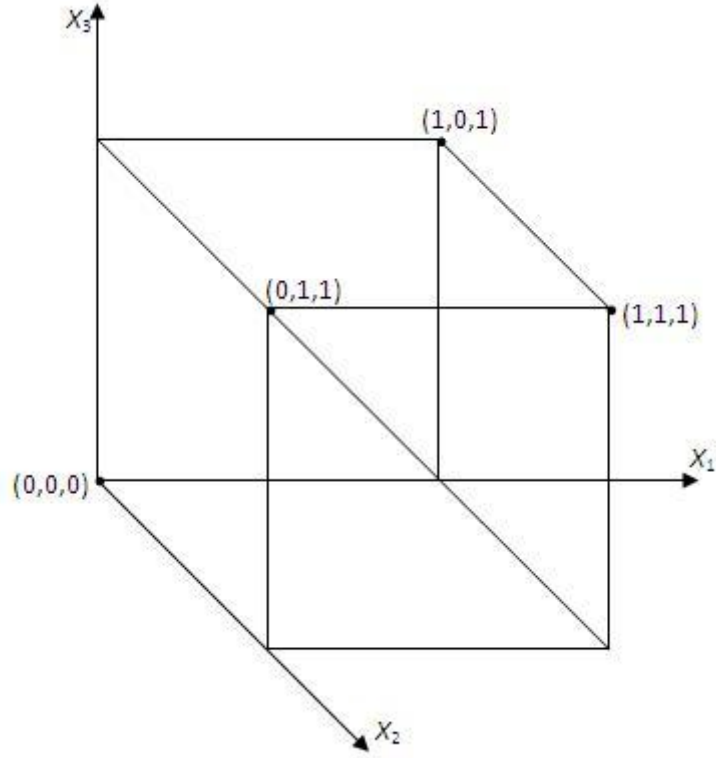
	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	0	1	1
<b>2</b>	0	0	0
<b>3</b>	1	1	1
<b>4</b>	1	0	1



0-1 tamsayı problemlerin çözüm uzayları hiperküp (hypercube) oluşturmaktadır.  $n$  değişken sayısı iken çözüm uzayında  $2^n$  tane uç nokta oluşmaktadır. Uç nokta sayısına göre isimlendirilmiş bazı hiperküp Şekil 4.8’de verilmiştir



Şekil 4.8. Çeşitli hiperküp (http://en.wikipedia.org/wiki/Hypercube).



**Şekil 4.9.** Örnek problemdeki noktaların yerleri

Sürekli doğrusal olmayan problemler için önerilmiş (4.1) ve (4.2) numaralı PSO denklemleri böyle bir çözüm uzayında kullanılamamaktadır. Örnek olarak  $[X_{gl(t)}^* - X_{i(t)}]$  ifadesi sırasıyla 2. ve 3. parçacıkların konumlarının farkına karşılık geliyor iken, hız vektörüne bu bileşenden gelecek katkı,  $(0,0,0)-(1,1,1)=(-1,-1,-1)$  olmaktadır. Bu vektörün diğer bileşenlerden gelebilecek 0'dan küçük elemanlara sahip vektör ile toplanması, çeşitli katsayılarla çarpılması ve (4.2)'de kullanılması, parçacıkları çözüm uzayı dışına itecektir.

Bu nedenle farklı formülasyonlara ve uygulamalara ihtiyaç vardır. Bu konudaki ilk çalışma Kennedy ve Eberhart (1997) tarafından yapılmıştır.

Bu çalışmada hız vektörünün görevini gören bir lojistik transformasyon (logistic transformation) önermiş ve değişkenlerin alabilecekleri değerlerin bir sigmoid fonksiyonu yardımıyla belirlenebileceğini göstermişlerdir.

Bu yönteme göre  $i$ . parçacık için hız vektörü yine (4.1)'e göre hesaplanmakta fakat ardından denklem (4.4)'deki seçim uygulanmaktadır. Buna göre,

$x_{id}$   $i$ . parçacığın  $d$ . boyutuna karşı gelen 0-1 değişken,  $v_{id}$   $i$ . parçacığın  $d$ . boyutuna karşı gelen hız vektörü,  $rnd \in [0.0, 1.0]$  olmak üzere sürekli düzgün bir rassal sayı,  $S$  sigmoid fonksiyonu olsun.

$$x_{id} = \begin{cases} 1, & rnd \leq S(v_{id}) \\ 0, & dd \end{cases} \quad (4.4)$$

0-1 tamsayı modeller için önerilmiş yukarıdaki PSO'nun başarılı sonuçlar elde edebileceği gösterilmiş olsa da uygulama kolaylığı, gerekli çözüm zamanı ve elde edilen çözümün değerinin kolay hesaplanabilir olmaması açısından sıklıkla tercih edilen bir yöntem olarak günümüze ulaşamamıştır. Ayrıca problemdeki kısıtlar da göz önünde bulundurulduğunda yöntemin uygulanabilirliği son derece azalmaktadır.

Bu nedenle sürekli olmayan çözüm uzayları için Kennedy ve Eberhart'ın (1997) 0-1 PSO'sundan farklı yaklaşımlar geliştirilmiştir. Bu yaklaşımların çoğunda GA çaprazlama operatörü kullanılmış ve çalışmalar çoğunlukla kombinatoriyel problemler arasında iş-makine çizelgeleme problemleri üzerinde yoğunlaşmıştır. Bu alandaki eserler şu şekildedir: Anghinolfi ve Paolucci (2009), geliştirdikleri operatör yardımıyla iş sıralamalarını düzenleyen bir PSO önermişlerdir. Pan vd. (2008), Lian vd. (2006), Tseng ve Liao (2008), Niu vd. (2008), çeşitli GA çaprazlama ve mutasyon operatörlerini kullanan PSO geliştirmişlerdir.

Rotalamada ise kesikli uzayda PSO'nun uygulanması konusundaki yayınlar şu şekildedir: Shi vd. (2007) ve Fan (2010) GSP üzerinde yine çaprazlama operatörü kullanan bir PSO tabanlı yaklaşım geliştirmiştir. Marinakis vd. (2010) çeşitli komşuluk arama teknikleriyle PSO'yu melezlemişler ve bu yaklaşımın performansını KARP üzerinde incelemişlerdir. Marinakis ve Marinaki (2010) ise aynı problem üzerinde PSO'yu genellikle YAA'da kullanılan uyarlamalı hafıza (adaptive memory) ile birleştirip bu hafızadan hangi parçacıkların çekileceğini bir rassal sayı yardımıyla belirlemişlerdir. Hafızanın güncellenmesi ise belirli koşullara bağlanmıştır. Özellikle bu

çalışmadaki yaklaşım tez kapsamında PSO'ya uygulanan çaprazlama ölçğine kaynak olmuştur.

Yukarıdaki çalışmalar haricinde, literatürdeki farklı problemler üzerinde çaprazlama operatörü kullanan PSO ile ilgili çalışmalar (Kaur and Bhatt, 2011; Pant et al., 2007; Pant et al., 2009; Premalatha and Natarajan, 2009; Park et al., 2007; Su et al., 2009; Fan, 2010; Zhang et al., 2009) ile karşılaşmak da mümkündür.

Bu konu çok güncel ve araştırmaya açıktır. Son olarak Qin vd. (in press) PSO'nun kesikli uzaylara uygulanabilmesi amacıyla algoritmik bir yaklaşım geliştirmişlerdir.

## 5. BAARP İÇİN GELİŞTİRİLEN ÇÖZÜM YAKLAŞIMLARI

Çalışmanın temel amacı, BAARP için kesikli PSO'nun ve yanı sıra daha önce sıklıkla denenmiş yöntemlerden GA ve YAA'nın performanslarının araştırılmasıdır. Bu amaçla söz konusu ileri sezgiseller kullanılarak probleme uygun algoritmalar tanımlanmış, detayları izleyen kesimlerde verilmiştir.

### 5.1. Başlangıç Çözüm Bulma Yöntemi

BAARP için çözüm yöntemleri üzerinde araştırmalar yapılırken, başlangıç çözüm önce rassal olarak alınmıştır. Ancak daha sonra, amaç fonksiyonu değeri bakımından daha iyi bir noktadan yola çıkmanın daha iyi olup olmayacağı da araştırılmış ve bunun için bir sezgisel algoritma geliştirilmiştir. Sonuç olarak ilgilenilen çözüm yöntemleri (YAA, PSO ve GA), rassal bir başlangıç çözüm yerine, sezgisel algoritma ile elde edilen bir başlangıç çözümden başlatıldığında, daha iyi sonuçlar elde edilebildiği gözlenmiştir. Ek 4'de verilen örnek problem için rassal bir çözüm oluşturulmuş ve çözüm değeri 233,14 br olurken, geliştirilen bu algoritma ile başlangıç çözüm 160,96 br olarak belirlenmiş ve iyileştirilmiştir. Özellikle BAARP'nin bazı özelliklerinden faydalanılarak geliştirilen bu sezgisel algoritma, bölümün devamında ayrıntılı olarak anlatılmış ve çalışma kapsamında incelenen test problemlerinin tamamında başlangıç çözüm bulma tekniği olarak kullanılmıştır.

BAARP'nin amaç fonksiyonu, her bir araç için, düğümleri gezerken oluşan birikimli artan sürenin toplamı şeklindedir. Bu amaç fonksiyonun enküçüklenmesi, Tek Tezgâh Akış Tipi Çizelgeleme Problemi'nde (Single Machine, Flow Shop), toplam akış süresinin (flow time) enküçüklenmesi ile büyük benzerlik göstermektedir.

Bu problemde Enküçük İşlem Süresi Yöntemi (Shortest Processing Time), toplam akış zamanını enküçükleme (Pinedo, 2008). Bu yöntemle göre işlem süresi diğer işlere göre daha az olan iş önce yapılmaktadır.

Öte yandan işlerin sıra bağımlı (sequence dependent) olduğu tek Tezgâh Akış Tipi Çizelgeleme Problemi, aslında GSP ile aynı problemi tariflemektedir. Çizelgelenmesi gereken işlerin hangi sırada yapılması gerektiği sorusu, GSP’de aracın, düğümleri hangi sırada gezmesi gerektiği ile eşdeğerdir.

Buradan hareketle BAARP’de araçların kapasite kısıtları gözetenmeden ve tüm düğümlerin tek bir araçla gezilebildiği varsayımı ve problemin amaç fonksiyonunun birikimli özellik göstermesi sebebi ile, SPT ile benzerlik gösteren iki aşamalı bir sezgisel algoritma geliştirilmiştir. Bu algoritmanın birinci aşaması sıralama, ikinci aşaması bölüntülemedir. Geliştirilen algoritmanın detayları aşağıda verilmiştir.

$G=(V,A)$  seriminde,  $n$  toplam düğüm sayısı,  $0$  depo düğümü iken, düğümler kümesi  $V=\{0,1,2,\dots,n\}$ , ayırtlar kümesi ise  $A= \{ (i,j): i, j \in V, i \neq j \}$  olarak gösterilsin.  $d_{i,j}$ ,  $i$  ve  $j$  düğümleri arasındaki uzaklık,  $0 \notin \Phi$  ve  $0 \notin \Phi'$  iken  $\Phi$  tura atanan,  $\Phi'$  ise tura atanmamış düğümler kümesini,  $\bar{T}_{1xn}$ , tur vektörünü oluştursun.

**Adım 1:**  $\bar{T}_{1xn} = \bar{0}$ ,  $i=0$ ,  $\Phi=\emptyset$ ,  $\Phi'=\{1,2,3,\dots,n\}$  olarak ata.

**Adım 2:**  $i=i+1$ . Eğer  $i=1$  ise  $enk\{d_{0,k}\}$ ,  $\forall k \in \Phi'$  bul ve  $\bar{T}_{1xi} = k$  olarak ata.  $\Phi \cup \{k\}$  ve  $\Phi' \setminus \{k\}$  değişikliğini yap ve Adım 3’e geç. Aksi halde  $enk\{d_{\bar{T}_{(i-1),k}}\}$ ,  $\forall k \in \Phi'$  bul ve  $\bar{T}_{1xi} = k$  olarak ata.  $\Phi \cup \{k\}$  ve  $\Phi' \setminus \{k\}$  değişikliğini yap ve adım 3’e geç.

**Adım 3:**  $s(\Phi)=n$  ise dur. Diğer durumlarda Adım 2’ye dön.

Bu aşama depo düğümünden başlanarak sırayla birbirine en yakın olan düğümleri sıralar. Sıralama işleminden sonra,  $\bar{T}$  vektörü, depo düğüm haricindeki tüm farklı düğümlerden oluşan büyük turu ifade edecektir. Bu tur dikkat edilirse araç kapasiteleri dikkate alınmadan elde edilmiştir.

Sezgisel algoritmanın 2. aşaması olan bölüntüleme işlemi araçlara, kapasiteleri doğrultusunda rotalarını tayin etmekte ve bunun sonucu olarak turun uygunluğunu test etmektedir. Bölüntüleme işlemi izleyen paragrafta anlatılmıştır.

$m$  araç sayısı,  $n$ =depo hariç düğüm sayısı,  $Q_k$   $k$ . aracın kalan kapasitesi,  $q_i$   $i$ . düğümün talebi iken,

**Adım 1:**  $k=1, i=0$

**Adım 2:**  $i=n$  ise dur. Aksi halde Adım 3'e geç.

**Adım 3:**  $i=i+1$ . Eğer  $Q_k \leq q_{\bar{T}_i}$  ise Adım 4'e geç. Aksi halde Adım 5'e geç.

**Adım 4:**  $Q_k=Q_k - q_{\bar{T}_i}$  yap. Adım 2'ye dön.

**Adım 5:**  $k=k+1$  yap. Adım 2'ye dön.

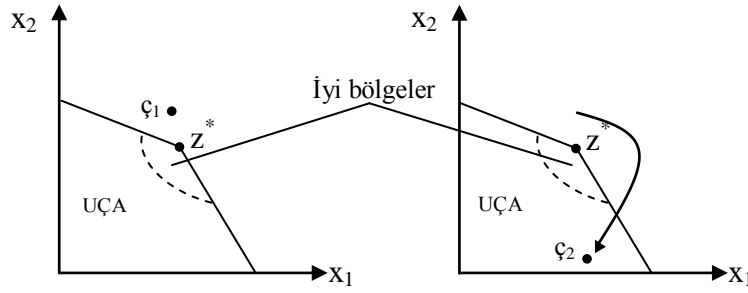
2. aşamada ise  $\bar{T}$  turundaki elemanlar sırayla, araç kapasitesi doluncaya kadar ilgili aracın rotasına atanmaktadır. Kapasitenin yetersiz kaldığı durumda ise izleyen aracın rotası oluşturulmaya başlanır. Bölüntüleme algoritması çalıştırıldığında elde edilen  $k$  değerine göre  $\bar{T}$  turunun uygunluğu test edilmiş olmaktadır. Problemden  $m$  adet aracın tamamının kullanılması isteniyorsa,  $k=m$  iken  $\bar{T}$  uygundur aksi halde uygun değildir. Ancak problem en fazla  $m$  adet araç kullanılması üzerine kurulu ise  $k \leq m$  iken  $\bar{T}$  uygundur.

Literatürde uygun olmayan çözümlerin değerlendirilmesi genellikle dört şekilde olmaktadır. Bunlar *tamir* (repairing), *red* (rejecting), *kabul* (accepting) ve *cezalandırma* (penalizing) olarak isimlendirilmektedir (Gen and Cheng, 1997).

*Tamir* yönteminde uygun olmayan her bir çözüm uygun hale getirilmektedir. Ancak bu yöntemin zayıf noktası her problem için probleme özgü bir tamir prosedürünün geliştirilmesi zorunluluğu ve bunun problemi çözmekten daha zor olabilmesidir (Gen and Cheng, 1997).

*Red* yönteminde uygun olmayan tüm çözümler reddedilmektedir. Özellikle dışbükey çözüm uzaylarında uygulanması kolay bir yöntemdir. Ancak red yönteminin zayıf noktası, çözüm uzayındaki amaç fonksiyonu değeri bakımından eniyi değere yakın ancak uygun olmayan çözümleri de reddedebilmesidir.

*Tamir ve red yöntemlerinin* Gen ve Cheng'e (1997) göre çeşitli zayıf noktaları vardır. Bunlar Şekil 5.1'de iki değişkenden oluşan dışbükey bir çözüm uzayında özetlenmeye çalışılmıştır.  $z^*$  uzayın eniyi çözümünü,  $\zeta_1$  uygun olmayan ancak eniyi çözüm değerine yakın bir çözümü,  $\zeta_2$  ise uygun olan ancak eniyi çözümden uzak bir çözümü tarif etmektedir.  $\zeta_1$  tamir edilerek  $\zeta_2$  elde edilmiştir ancak son durumda  $|z^* - \zeta_2| > |z^* - \zeta_1|$  olmuştur. Benzer şekilde  $\zeta_1$  red yöntemine göre reddedilir,  $\zeta_2$  ise kabul edilir. Bunun sonucu olarak iyi bir çözüm kaçırılabilir.



**Şekil 5.1.** Tamir ve red yöntemlerinin muhtemel zayıflıkları

*Kabul yönteminde* ise uygun olmayan çözümlerin uygun hale getirilmesi için herhangi özel bir işlem yapılmaz. Popülasyondaki ilgili kromozomun ya da sürüdeki parçacığın ardıştırmalar ilerledikçe uygun çözüm uzayına girmesi beklenir.

*Cezalandırma yönteminde* ise, problemin amaç fonksiyonuna, uygun olmayan bir çözüm için amaç fonksiyonunu olumsuz yönde etkileyen bir ifade eklenir. Bu eklenen ifadeye penaltı fonksiyonu denilir. Bu ifade çok hassas olmalıdır ve bunu tayin etmek bu yöntemin temel zorluğudur (Gen and Cheng, 1997).

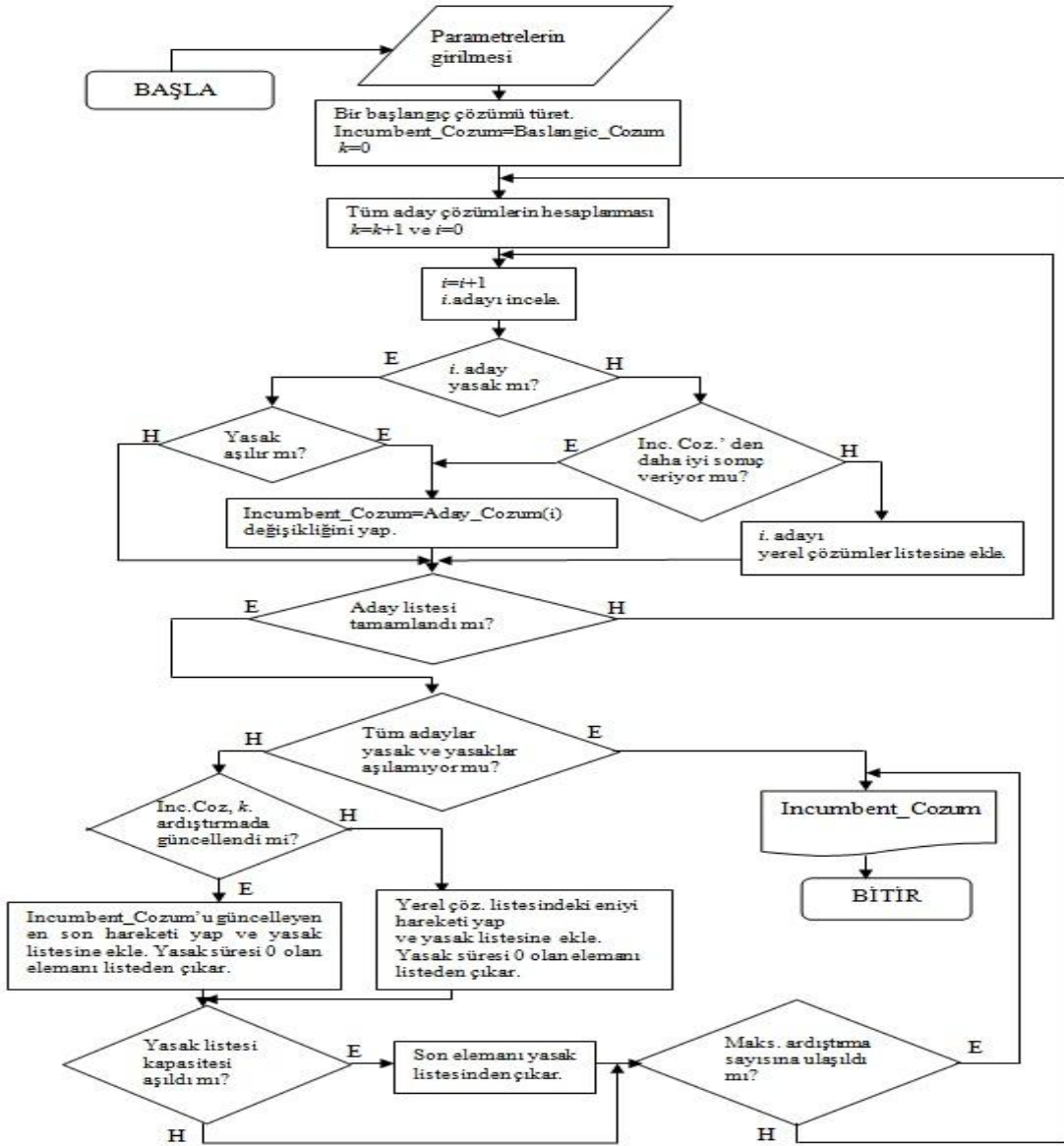
Bunların dışında sadece uygun çözüm uzayında çözüm üreten, problemin yapısına özel olarak *ayarlanmış genetik gösterimler ve operatörler* (modifying genetic operators) ile de uygun olmayan çözümlerle mücadele edilebilir (Gen and Cheng, 1997).

Bu çalışmada algoritmaların işlem sürelerinin azaltılabilmesi hedeflenmiş ve ek işlem süresi gerektirmemesi için, uygun olmayan çözümlerin değerlendirilmesinde *kabul yöntemi* kullanılmıştır.



## 5.2 BAARP İçin Geliştirilen Yasaklı Arama Yaklaşımı

BAARP için geliştirilen YAA'da komşuluk araması için ikili yer değiştirme tekniği kullanılmıştır. Farklı parametre değerleri ile çeşitli denemelerin ardından YAA parametreleri şu şekilde seçilmiştir: Aday çözüm sayısı 250, ardıştırma sayısı 5000,  $n$  problemdeki düğüm sayısı iken, yasak listesi kapasitesi ise  $int(n/3)$  değerlerine sabitlenmiştir. Geliştirilen YAA'ya ait genel bir program akış diyagramı Şekil 5.2'de verilmiştir.



Şekil 5.2. Geliştirilen Yasaklı Arama Algoritması yaklaşımının akış şeması

İlgili parametre değerlerinin belirlenmesinde Varyans Analizi (Analysis of Variance) gibi bir yaklaşım kullanılmamış bunun yerine denemeler yapılarak çözüm süresinin katlanılabilir olduğu parametre değerlerinin belirlenmesine özen gösterilmiştir. Bu düşünceyle aday çözüm sayısı, 80, 150, 200, 250, 300, 350, artırma sayısı 2000, 5000, 7000, 10000, yasak listesi kapasitesi 10, 20, 30, 70 ve son olarak  $int(n/3)$  olduğu değerler seçilerek denemeler yapılmış ve YAA'nın başarılı olduğu parametre değerleri kullanılmıştır.

### 5.3 BAARP İçin Geliştirilen Genetik Algoritma Yaklaşımı

BAARP için geliştirilen genetik algoritma yaklaşımında, test problemlerinde başlangıç çözümün %10'u önerilen sezgisel algoritma ile %90'ı ise rassal çözümlerden oluşturulmuştur. YAA'da 250 aday çözüm kullanılması sebebiyle GA'nın da 250 farklı nokta ile hareket etmesi hedeflenmiştir. Bu nedenle popülasyon büyüklüğü 250 olarak seçilmiştir.

Ardıştırma sayısı ise YAA, GA ve PSO'nun çözüm sürelerinin birbirlerine yakınlığına göre belirlenmeye çalışılmıştır. Denemelerin çoğunluğunda YAA'nın bir ardıştırması için gerekli çözüm süresinin GA'nın bir ardıştırması için gerekli süresinin yaklaşık olarak üç katı olduğu gözlemlenmiştir. Bu nedenle GA'nın ardıştırma sayısı YAA'da kullanılan ardıştırma sayısının yaklaşık olarak üç katı olarak seçilmiş ve 13000'e sabitlenmiştir.

Çaprazlama tekniği olarak tek noktalı sıra tabanlı çaprazlama tekniği kullanılmıştır (Gen and Cheng, 1997). Bu teknik aşağıda anlatılmıştır.

Çaprazlamaya girecek olan iki birey  $P_1$  ve  $P_2$ , çaprazlama sonucu oluşacak birey  $O_1$ ,  $P_1$  ve  $P_2$ 'nin turları aşağıdaki gibi olsun.

$P_1$	5	6	9	1	8	3	4	2	7
$P_2$	2	8	4	5	1	6	7	9	3

Şekil 5.3. Çaprazlamaya girecek bireyler.

İki bireyin arasında rassal bir seçim olur ve seçilen bireyden bir segment yine rassal olarak seçilir.  $P_1$  ve aşağıdaki şekilde belirtilen segment rassal olarak seçilmiş olsun.

$P_1$	5	6	9	1	8	3	4	2	7
-------	---	---	---	---	---	---	---	---	---

**Şekil 5.4.** Rassal seçilen birey ve segment.

Rassal seçilen bu segment aynı boyuttaki genleri boş olan çocuk kromozoma kopyalanır, kopyalanan genlerin değerleri, çaprazlamaya giren diğer birey  $P_2$ 'den silinir ve aşağıdaki şekil elde edilir.

$P_2$	2	8	4	5	1	6	7	9	3
$O_1$			9	1	8	3			

**Şekil 5.5.** Ara birey.

$P_2$ 'den silinmemiş değerler  $O_1$ 'deki boş yerlere aynı sırayla eklenir ve çaprazlama tamamlanarak aşağıdaki şekildeki tur elde edilir.

$O_1$	2	4	9	1	8	3	5	6	7
-------	---	---	---	---	---	---	---	---	---

**Şekil 5.6.** Çaprazlama sonucu oluşan birey.

Çeşitli denemelerin ardından GA'da kullanılan çaprazlama oranı %70'lerde iken başarılı ve çözüm süresi olarak daha katlanılabilir sonuçlar elde edilebildiği görülmüştür. Bu nedenle çaprazlama oranı %70 olarak belirlenmiştir.

Mutasyon operatörü olarak ikili yer deęiřtirme teknięi kullanılmıř ve mutasyon oranı çeřitli denemelerin ardından %30'a yakın deęerlerde iken daha bařarılı sonuřlar elde edilebildięi, yerel eniyilerden daha kolay kařılabildięi gözlemlenmiřtir. Bu nedenle mutasyon oranı %30 olarak seřilmiřtir.

GA'da seřim iřlemi olarak, turnuva büyüklüęü 2 olmak üzere turnuva seřim teknięi kullanılmıřtır. Bu teknięe göre aprazlama ve mutasyon iřlemlerinden sonra yeni uyum deęerlerine göre 2 farklı birey rassal olarak seřilmektedir ve uyum deęeri daha küçük olan birey bir sonraki ardıřtırma için yeni popülasyona kopyalanmaktadır (Gen and Cheng, 1997). Bu iřlemin sözde kodları Őekil 5.7'de verilmiřtir.

```

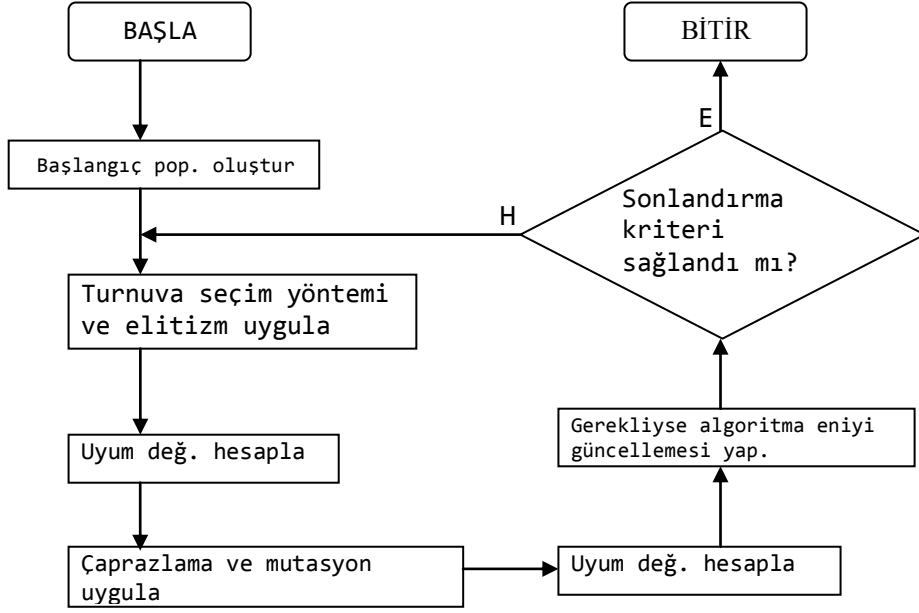
i=0
Do until i = PB
  i=i +1
1:   q = int(Rnd()*PB+1)
     t = int(Rnd()*PB+1)
     If q = t then
       Goto 1
     Else
       If FV(q)<FV(t) then
         q indisli kromozomu sonraki popülasyona kopyala
       Else
         t indisli kromozomu sonraki popülasyona kopyala
       End if
     End if
Loop

PB: Popülasyon büyüklüęü
q,t:q,t ∈ ℤ , 1≤q≤PB, 1≤t≤PB ve q≠t
FV: Uyum deęeri (Fitness value).

```

**Őekil 5.7.** Turnuva seřim yöntemi sözde kodları.

Probleme özğü geliřtirilen GA'nın akıř diyagramı Őekil 5.8'de verilmiřtir.



Şekil 5.8. Geliştirilen genetik algoritma yaklaşımının akış diyagramı.

#### 5.4 BAARP İçin Geliştirilen Parçacık Sürüsü Optimizasyonu Yaklaşımı

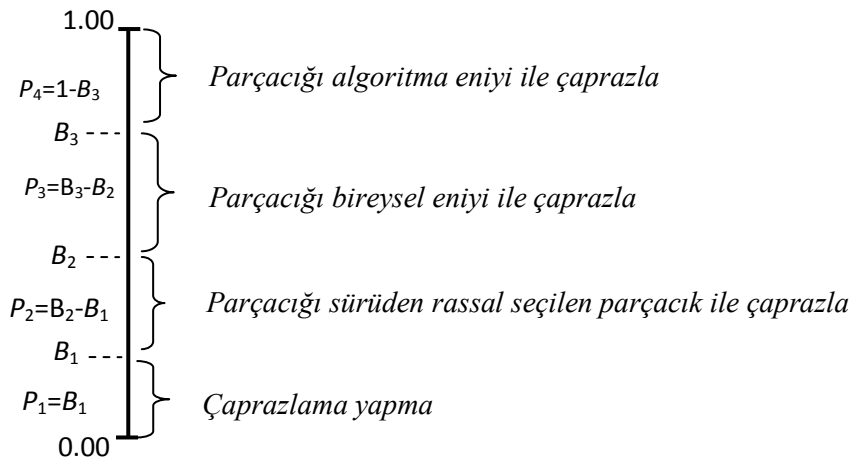
BAARP için geliştirilen PSO tekniğinde parçacıkların birbirleri arasında bilgi aktarımı, kendi sahip oldukları ve sürüdeki eniyi çözüm bilgilerinin kullanımı çeşitli oranlarda gerçekleşmektedir. Ancak çözüm uzayının kesikli olması sebebiyle, bu işlemler, sürekli PSO'da kullanılan (4.1) ve (4.2) numaralı denklemlerdeki gibi ifade edilememektedir. Bu nedenle bu çalışmada, (4.1) ve (4.2) denklemlerinin kullanılmadığı ve sürekli PSO'dan farklı bir yaklaşım izlenmiş, GA operatörlerini PSO işleyişinde kullanan bir algoritma uygulanmıştır.

Literatür araştırmasında daha önce de bahsedildiği gibi, GA operatörleri ile PSO işleyişini birleştiren yaklaşımlar geliştirilmiştir (Anghinolfi and Paolucci, 2009; Kaur and Bhatt, 2011; Niu et al., 2008; Lian et al., 2006; Shi et al., 2007; Pant et al., 2007; Premalatha and Natarajan, 2009; Park et al., 2007; Qin et al., in press; Pan et al., 2008; Tseng and Liao, 2008; Pant et al., 2009; Fan, 2010; Su et al., 2009; Marinakis et al., 2010; Marinakis and Marinaki, 2010; Zhang et al., 2009). Buradan hareketle bu çalışma

kapsamında da bu yaklaşımlara benzer bir yaklaşım geliştirilmiş ve bunlardan farklı olarak bir çaprazlama skalası önerilmiştir.

Bu skala ile, parçacıkların birbirleri arasındaki bilgi aktarımı, kendi sahip oldukları ve sürüdeki eniyi çözüm bilgilerinin kullanımının, sürekli PSO'da olduğu gibi çeşitli oranlara dağıtılması ve sürekli PSO işleyiş mantığının, kesikli PSO'ya mümkün olduğu kadar yansıtılması hedeflenmiştir. Bu yaklaşımda klasik GA'dan farklı olarak sürüdeki bireylerin her biri için bireysel eniyi çözüm dizisi tutulmaktadır. Bu çözüm dizisi her ardışırma sonunda gerekliyse (bireysel olarak daha iyi bir çözüme erişildiyse) güncellenmektedir.

Skalanın kullanılabilmesi için,  $B_t \in [0,1]$ ,  $t=1,2,3$  ve  $B_1 < B_2 < B_3$  olmak üzere, üç farklı yeni parametreye ihtiyaç vardır. Bu parametreler  $[0,1]$  aralığını aşağıdaki şekildeki gibi çaprazlama yapma, rassal birey çaprazlaması, bireysel eniyi çaprazlaması ve algoritma eniyi çaprazlaması olmak üzere dört farklı bölgeye bölmektedir. Sürüdeki her birey için çekilen  $r_i \in [0,1]$  rassal sayısının hangi bölgede yer alacağı ile hangi tip çaprazlamanın yapılacağı bilgisi elde edilmektedir. Çaprazlama skalası Şekil 5.9'da verilmiştir.



Şekil 5.9. PSO yaklaşımında kullanılan çaprazlama skalası.

$P_i$ , çekilen rassal sayının  $i$ . bölgeye düşme olasılığı iken,  $P_i$  olasılık ile  $i$ . tip çaprazlama yapılmaktadır.

Örnek olarak sürüdeki bir parçacık için çekilen rassal sayı  $[B_2, B_3]$  aralığında çıkarsa, ilgili parçacık, o ardıştırmaya kadar erişebildiği *bireysel eniyi çözüm* ile çaprazlanır.

Geliştirilen PSO yaklaşımında uygulanan çaprazlamanın sözde kodları ise Şekil 5.10'da verilmiştir.

```

i=0
Do until i = SB
i=i +1
ri =Rnd()
If ri ≤ B1 then
    parçacık(i)'yi çaprazlama
End if
If ri ≤ B2 and ri > B1 then
    parçacık(i)'yi sürüden seçilen rassal bir parçacıkla çaprazla.
End if
If ri ≤ B3 and ri > B2 then
    parçacık(i)'yi bireysel eniyi(i) ile çaprazla.
End if
If ri > B3 then
    parçacık(i)'yi algoritma eniyi ile çaprazla.
End if
Loop

SB: Sürü büyüklüğü
Bt : Bt∈[0,1], t=1,2,3 ve B1<B2<B3 olmak üzere, çaprazlama parametreleri.
ri : ri∈[0,1] olmak üzere, i. parçacık için çekilen rassal sayı.

```

**Şekil 5.10.** Geliştirilen PSO yaklaşımındaki çaprazlama işleminin sözde kodları.

Popülasyon tabanlı bir teknik olmasından ve GA ile bu açıdan kıyaslanabilmesi sebebiyle, GA'da kullanılan parametrelerin bir çoğu PSO'da da uygulanmıştır.

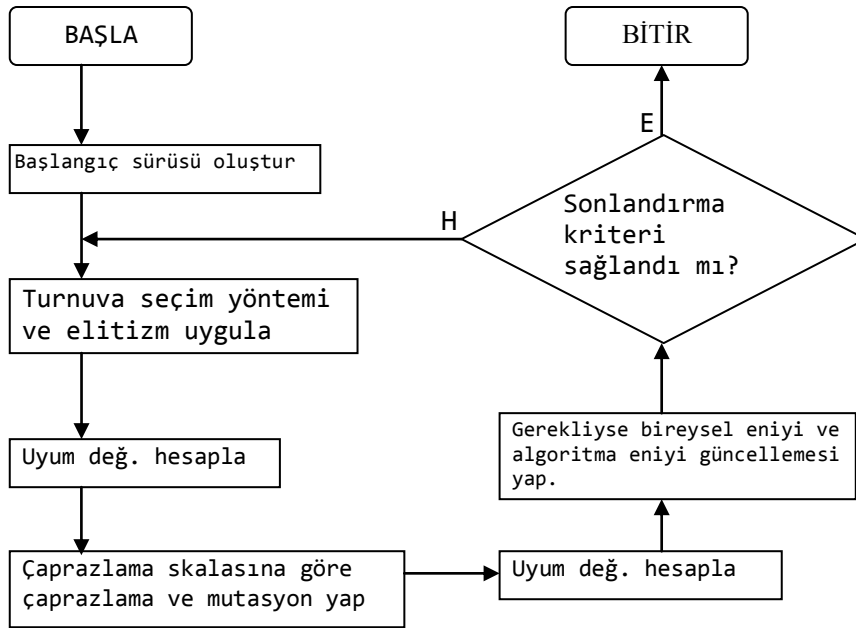
PSO'nun test problemlerinde de başlangıç çözümün %10'u sezgisel algoritma ile elde edilen çözümden %90'ı ise rassal çözümlerden oluşturulmuştur. GA ve YAA arasındaki aynı komşuluk sayısı sebebiyle PSO'da da 250 farklı nokta ile hareket edilmesi hedeflenmiştir. Bu nedenle sürü büyüklüğü 250 olarak seçilmiştir.

PSO'daki çaprazlama parametreleri belirlenirken  $B_1=0.05$ ,  $B_2=0.85$   $B_3=0.95$ ,  $B_1=0.30$ ,  $B_2=0.90$   $B_3=0.95$  ve  $B_1=0.1$ ,  $B_2=0.8$   $B_3=0.9$  gibi üç farklı parametre seti, farklı

boyutlardaki test problemleri üzerinde denenmiştir. GA'da çaprazlama oranının 0.7 seçilmesi (PSO'daki karşılığı  $B_2-B_1=0.7$ ) ve özellikle,  $B_1=0.1$ ,  $B_2=0.8$   $B_3=0.9$  değerlerinde daha başarılı sonuçlar türetebilmesi sebebiyle, çaprazlama parametreleri  $B_1=0.1$ ,  $B_2=0.8$   $B_3=0.9$  değerlerine sabitlenip tüm test problemlerinde aynı değerler kullanılmıştır.

PSO'da, GA'daki mutasyon operatörü olarak kullanılan ikili yer değiştirme tekniği kullanılmıştır. Mutasyon oranı ise PSO'nun GA ile kıyaslanabilir olması açısından GA'daki mutasyon oranı olan %30'a sabitlenmiştir. Aynı sebepten dolayı elitizm tekniği de GA'da uygulandığı gibi uygulanmış, bir sürüdeki eniyi parçacık izleyen sürüye doğrudan kopyalanmıştır.

Geliştirilen PSO yaklaşımının akış diyagramı Şekil 5.11'de verilmiştir.



**Şekil 5.11.** Geliştirilen PSO yaklaşımının akış diyagramı.

Geliştirilen yöntemlerin test problemleri üzerindeki performansları ise izleyen bölümde verilmiştir.



## 6. TEST PROBLEMLERİ VE ELDE EDİLEN SONUÇLARI

Literatürde erişilebildiği kadarıyla BAARP için önerilmiş test problemleri yoktur. Bu nedenle PSO, GA ve YAA'nın performansları, <http://www.branchandcut.org/> sitesinde asıl olarak KARP için yayınlanmış test problemlerinin verileri kullanılarak, Core2 Quad E8400 işlemci, 4GB DDR3 RAM özellikteki bilgisayarlarda incelenmiştir. Geliştirilen algoritmalar Visual Basic 6.0 ortamında kodlanmıştır.

### 6.1. Test Problemlerinin Özellikleri

Test problemlerinin çözdürülmesi aşamasından önce matematiksel modelin ve bu çalışmada kullanılan algoritmaların doğrulanması amacıyla, yine bu çalışmada kullanılmış enküçük boyutlu test probleminin eniyi çözümünün ve çözüm değerinin elde edilmesi hedeflenmiştir. Bu nedenle, diğer problemlere göre daha az sıklıkta sayılabilecek (%88) ve aynı zamanda en az sayıda düğümden oluşan P-n16-k8 problemi ele alınmıştır. Problemin çözümü, 2.0.28.0 versiyonlu GAMS IDE yazılımı CPLEX MIP (mixed integer programming) çözücüsü ile araştırılmış, 27276,92 sn. (yaklaşık olarak 7,5 saatten biraz fazla) sonunda ağaç düğüm dosya boyutu (tree node file size) 2328,80 MB (sıkıştırılmış boyutu ise 615,94 MB) değerine ulaşmış ancak buna rağmen problemin kesin çözümü belirlenememiştir. Bu süre içinde elde edilen eniyi olurlu (feasible) çözüm değeri ise 397,2114 olarak belirlenmiştir. Aynı değer, GA tarafından 177 sn., PSO tarafından ise 188 sn., içersisinde elde edilmiş durumdadır. YAA tarafından ise aynı P-n16-k8 probleminde elde edilen eniyi çözüm değeri 408,99 olmuştur. Bu tablo, BAARP için, ileri sezgisel çözüm yöntemlerinin gerekliliğini özetlemektedir.

Çalışmada kullanılan enküçük boyutlu problemin kesin çözümü bulunamadığı için çalışmada kullanılan ve daha büyük boyutlu olan test problemlerinin çözümünde model kullanılmamış, geliştirilen ileri sezgisel yaklaşımlarla çözüm aranmıştır. Öte yandan daha küçük boyutlu bir problem rassal olarak oluşturularak modelin performansı için bir araştırma daha yapılmıştır. Bu problem depo düğüm dâhil olmak üzere 10

düğümünden oluşmaktadır. Düğümlerin koordinatları, [1,20], talepler ise [10,20] aralığındaki rassal tamsayılardan seçilmiştir. Problemden 75 br kapasiteli 2 araç kullanılmış ve sıklık % 89 olarak belirlenmiştir. Problemden parametrelere ait bilgiler ve işlem penceresi (process window) çözüm raporu çıktısı Ek 4’de verilmiştir. Buna göre problemin eniyi çözüm değeri 2131,75 sn. sonunda 113,2407 olarak bulunmuş ve bu çözüme göre birinci aracın rotası 0-5-2-9-8, doluluk oranı  $70/75=0,93$ , 2. aracın rotası 0-10-6-3-4-7, doluluk oranı  $73/75=0,97$  olarak belirlenmiştir. Aynı çözüm değerine GA, 19 sn sonunda 912., YAA ise 29 sn sonunda 819. ardıştırmada erişebilmiştir. PSO ise eniyi çözüm olarak 115,61 değerini elde etmiştir.

Bu çalışmada, geliştirilen ileri sezgisel algoritmaların performanslarını büyük boyutlu problemler üzerinde göstermek için literatürde günümüze kadar yayınlanmış tüm farklı test problemleri gruplarından örnekler alınarak farklı özelliklerdeki test problemleri üzerinde denemeler yapılmaya gayret edilmiştir. Ancak bazı test problemi gruplarındaki veri formatlarının anlaşılabilmesi nedeniyle bazılarının üzerinde çalışılamamıştır. Kalan gruplardan ise tüm problemler seçilmemiş, grubu temsil edebilecek sayıda boyut artışı ve % sıklık değerlerine göre problemler hedeflenmiştir. Böylece tüm test problemi gruplarından, verisi okunabilen toplam 199 adet test probleminden seçilen 39 adet problem üzerinde PSO, GA ve YAA beşer defa çalıştırılmıştır.

Literatürde KARP için yayınlanmış test problemlerinde 1 tane merkez depo, çok sayıda talep noktası vardır ve araç turları kapalıdır. Ele alınan BAARP’de ise bu yapıya ek olarak araçların açık tur yapmasına izin verilmektedir. Ayrıca amaç fonksiyonu da farklıdır. Bu nedenle test problemlerindeki veriler aynen kullanılabilir ancak amaç fonksiyonu değeri ve elde edilecek turlar farklı olacaktır. Sonuç olarak elde edilecek değerlerin literatürdeki sonuçlarla karşılaştırılması mümkün değildir.

Seçilen test problemlerinin özellikleri Çizelge 6.1’de verilmiştir.

**Çizelge 6.1.** Literatürden alınmış test problemlerinin özellikleri

#	<i>Ad</i>	<i>n</i>	<i>k</i>	% <i>Sıklık</i>
1	A-n32-k5	32	5	0,82
2	A-n33-k5	33	5	0,89
3	A-n37-k5	37	5	0,81
4	A-n39-k6	39	6	0,88
5	A-n48-k7	48	7	0,89
6	A-n53-k7	53	7	0,95
7	A-n62-k8	62	8	0,92
8	A-n63-k9	63	9	0,97
9	A-n69-k9	69	9	0,94
10	A-n80-k10	80	10	0,94
11	B-n31-k5	31	5	0,82
12	B-n34-k5	34	5	0,91
13	B-n43-k6	43	6	0,87
14	B-n44-k7	44	7	0,92
15	B-n50-k8	50	8	0,92
16	B-n56-k7	56	7	0,88
17	B-n57-k9	57	9	0,89
18	B-n68-k9	68	9	0,93
19	B-n78-k10	78	10	0,94
20	E-n22-k4	22	4	0,94
21	E-n30-k3	30	3	0,94
22	E-n76-k7	76	7	0,89
23	E-n101-k14	101	14	0,93
24	F-n135-k7	135	7	0,95
25	M-n101-k10	101	10	0,91
26	M-n151-k12	151	12	0,93
27	M-n200-k17	200	17	0,94
28	P-n16-k8	16	8	0,88
29	P-n20-k2	20	2	0,97
30	P-n45-k5	45	5	0,92
31	P-n50-k10	50	10	0,95
32	P-n55-k7	55	7	0,88
33	P-n60-k10	60	10	0,95
34	P-n65-k10	65	10	0,94
35	P-n70-k10	70	10	0,97
36	P-n76-k5	76	5	0,97
37	P-n101-k4	101	4	0,91
38	bays-n29-k5	29	5	0,93
39	dantzig-n42-k4	42	4	0,93

Çizelge 6.1'deki sütunlar sırasıyla test problemlerinin numaralarını (#), adlarını (*Ad*), boyutlarını (*n*) (problemdeki depo dâhil olmak üzere toplam düğüm sayısı), problemde kullanılması gereken araç sayılarını (*k*) ve problemin sıklığını (% *sıklık*) göstermektedir. Bir problemin sıklığı, taşınacak yükün toplam araç kapasitesi içindeki değerini göstermektedir. Bir diğer deyişle araçların ortalama olarak ne kadar dolu olacağını gösterir. Doğal olarak bir aracın kapasitesi ne kadar çok kullanılıyorsa talep

noktalarının talepleri kamyonlara o kadar iyi yerleştiriliyor demektir. Bu da olası pek çok kombinasyonun araştırılması anlamına geleceği için problemin çözümü zorlaşır. Öte yandan problemin sıklık değeri ne kadar az ise uygun çözüm bulmak o kadar kolay olacaktır. Test problemlerinin sıklıkları yayınladıkları internet sitesinde verilmiş olup aşağıdaki formülasyona göre hesaplanmışlardır.

$n$  toplam düğüm sayısı,  $m$  toplam araç sayısı,  $i=1,2,\dots,n$ ,  $k=1,2,\dots,m$ ,  $q_i$   $i$ . düğümün talebi,  $C_k$   $k$ . aracın kapasitesi iken,

$$\% \text{ sıklılık} = \frac{\sum_{i=1}^n q_i}{\sum_{k=1}^m C_k} \quad (6.1)$$

Çizelge 6.1'in ad sütununda A, B ve P kodu ile başlayan problemler Augerat vd.'nin (1995) yayınladığı A B ve P grubu olarak da isimlendirilen test problemleridir. Diğer E F M P kodları ile başlayan problemler sırasıyla ise sırasıyla Christofides ve Eilon'nun (1969), Fisher'ın (1994), Christofides vd.'nin (1979) yayınlamış oldukları test problemleridir. Bu problemlere, <http://www.branchandcut.org/> bağlantısından on-line olarak erişim mümkündür. Son iki satırdaki test problemleri ise (38 ve 39 numaralı olanlar) TSPLIB'da asıl olarak GSP için yayınlanmış fakat düğümlere talep atanarak ve uygun kapasitede araçlar kullanılarak KARP için de kullanılabilir hale dönüştürülmüş problemlerdir. İlgili değerlerin atanması çalışma kapsamında yapılmamış olup yine <http://www.branchandcut.org/> sitesinden alınmış veriler aynen kullanılmıştır.

$n$  toplam düğüm sayısı,  $i=1,2,\dots,n$ ,  $q_i$   $i$ . düğümün talebi iken 38 ve 39 numaralı test problemleri haricindeki tüm test problemlerindeki düğümlerin talepleri,  $q_i \in \mathbb{Z}^+$  olmaktadır. 38 ve 39 numaralı test problemlerinde ise düğümler birim talebe sahiptir ve  $q_i=1 \forall i$  olarak problemler çözülmüştür.

Tüm test problemlerine ait serimlerde yer alan düğümlerin koordinatları iki boyutlu Öklid düzlemi üzerinde yer almaktadır.  $x_i$   $i$ . düğümün apsisi  $y_i$   $i$ . düğümün ordinatı olmak üzere, 24 numaralı test problemi haricindeki test problemlerinde  $x_i, y_i \in \mathbb{Z}^+$ , 24 numaralı test probleminde ise  $x_i, y_i \in \mathbb{R}$ 'dir.

## 6.2 Amaç Fonksiyonu Değerleri ve Çözüm Sürelerine İlişkin Hesapsal Sonuçlar

Her test problemi PSO, GA ve YAA algoritmaları için beşer defa koşturulmuş, elde edilen amaç fonksiyonu değerleri sırasıyla Çizelge 6.2-6.4'de, çözüm süreleri sırasıyla Çizelge 6.2-6.4'de verilmiştir.

**Çizelge 6.2.** PSO ile elde edilen amaç fonksiyonu değerleri

#	Ad	% Sıklık	PSO <sub>1</sub>	PSO <sub>2</sub>	PSO <sub>3</sub>	PSO <sub>4</sub>	PSO <sub>5</sub>
1	A-n32-k5	0,82	2366,25	2305,50	<b>2289,35</b>	2421,89	2344,27
2	A-n33-k5	0,89	<b>1771,17</b>	1994,73	1816,18	<b>1771,17</b>	1788,12
3	A-n37-k5	0,81	2333,05	2192,59	2548,90	<b>2173,50</b>	2223,66
4	A-n39-k6	0,88	2418,72	2571,82	<b>2317,29</b>	2428,84	2355,59
5	A-n48-k7	0,89	<b>3283,04</b>	3584,51	3572,26	3494,82	3493,73
6	A-n53-k7	0,95	3845,04	3886,57	3678,61	3660,48	<b>3535,07</b>
7	A-n62-k8	0,92	4547,38	<b>4327,24</b>	4536,73	4411,54	4574,98
8	A-n63-k9	0,97	5506,12	5403,65	<b>5241,61</b>	6227,24	5241,01
9	A-n69-k9	0,94	4358,99	<b>3781,63</b>	4037,97	4125,91	3928,80
10	A-n80-k10	0,94	6793,25	<b>6715,70</b>	6806,58	6766,15	6836,06
11	B-n31-k5	0,82	1877,77	1923,02	1880,66	<b>1865,77</b>	<b>1865,77</b>
12	B-n34-k5	0,91	<b>2369,58</b>	2370,43	2484,81	2379,29	2391,22
13	B-n43-k6	0,87	2264,15	2286,49	2291,36	<b>2254,87</b>	2257,33
14	B-n44-k7	0,92	2492,68	2861,71	<b>2398,02</b>	2415,57	2415,98
15	B-n50-k8	0,92	3272,50	3242,85	<b>3137,79</b>	3224,07	3292,70
16	B-n56-k7	0,88	<b>2567,45</b>	2622,13	2637,15	2570,19	2595,30
17	B-n57-k9	0,89	<b>4618,60</b>	4660,37	4633,40	4639,71	4674,72
18	B-n68-k9	0,93	4647,60	<b>4307,80</b>	4480,00	4447,88	4381,21
19	B-n78-k10	0,94	<b>4155,75</b>	4248,56	4414,64	4290,31	4329,10
20	E-n22-k4	0,94	1019,18	869,63	<b>855,23</b>	899,65	933,99
21	E-n30-k3	0,94	2137,89	<b>2026,62</b>	2066,18	2049,36	2090,67
22	E-n76-k7	0,89	4163,85	3995,09	<b>3863,95</b>	4037,13	4116,51
23	E-n101-k14	0,93	<b>3618,70</b>	3701,11	3766,25	3725,01	4484,47
24	F-n135-k7	0,95	<b>8113,12</b>	8390,28	8374,89	8311,17	8453,88
25	M-n101-k10	0,91	3978,88	4180,38	4098,39	4002,11	<b>3929,71</b>
26	M-n151-k12	0,93	6735,58	6726,10	6728,21	6927,92	<b>6621,34</b>
27	M-n200-k17	0,94	7855,83	8525,11	7896,11	7714,12	<b>7604,23</b>
28	P-n16-k8	0,88	<b>397,21</b>	<b>397,21</b>	403,16	416,91	405,13
29	P-n20-k2	0,97	<b>989,07</b>	1021,75	1139,70	1161,52	1036,12
30	P-n45-k5	0,92	2177,90	2449,49	2209,22	<b>2157,16</b>	2270,25
31	P-n50-k10	0,95	1628,20	1584,63	1619,31	<b>1576,94</b>	1716,37
32	P-n55-k7	0,88	2298,16	2385,26	2179,75	<b>2121,20</b>	2359,13
33	P-n60-k10	0,95	<b>1716,37</b>	2154,47	2149,45	2095,94	2049,82
34	P-n65-k10	0,94	2465,81	2617,75	2467,26	<b>2323,18</b>	2494,07
35	P-n70-k10	0,97	2913,82	4028,71	<b>2635,35</b>	2816,15	3043,81
36	P-n76-k5	0,97	5112,08	5362,50	5344,01	<b>4967,68</b>	5054,92
37	P-n101-k4	0,91	9029,58	8859,72	<b>8470,78</b>	8887,14	8636,86
38	bays-n29-k5	0,93	<b>29886,93</b>	<b>29886,93</b>	30571,36	30840,49	31449,25
39	dantzig-n42-k4	0,93	4460,04	<b>4264,12</b>	4606,12	4694,01	4590,11

Çizelge 6.3. GA ile elde edilen amaç fonksiyonu değerleri

#	Ad	% Sıklık	GA <sub>1</sub>	GA <sub>2</sub>	GA <sub>3</sub>	GA <sub>4</sub>	GA <sub>5</sub>
1	A-n32-k5	0,82	2466,27	<b>2405,50</b>	2542,41	2479,97	2421,89
2	A-n33-k5	0,89	1922,04	2131,92	<b>1896,99</b>	1928,32	1922,04
3	A-n37-k5	0,81	<b>2401,37</b>	2933,93	2697,57	2751,34	2542,91
4	A-n39-k6	0,88	2926,84	3483,33	2964,86	<b>2899,89</b>	3055,38
5	A-n48-k7	0,89	4401,65	4027,29	3958,92	<b>3783,90</b>	4038,44
6	A-n53-k7	0,95	4616,52	4419,02	4603,12	<b>4361,36</b>	4459,15
7	A-n62-k8	0,92	<b>5404,43</b>	5465,12	5818,73	5730,29	5539,70
8	A-n63-k9	0,97	<b>7239,45</b>	7807,11	8695,99	8015,74	8502,38
9	A-n69-k9	0,94	4946,68	5079,92	4959,64	5033,76	<b>4740,83</b>
10	A-n80-k10	0,94	7582,44	7812,18	<b>7461,42</b>	7803,98	7729,93
11	B-n31-k5	0,82	1916,11	1957,87	1910,98	1904,84	<b>1904,38</b>
12	B-n34-k5	0,91	<b>2356,20</b>	2401,88	2509,31	2538,51	2406,51
13	B-n43-k6	0,87	<b>2310,32</b>	2413,83	2400,29	2364,08	2427,11
14	B-n44-k7	0,92	2812,50	<b>2539,02</b>	2811,61	2710,63	2674,99
15	B-n50-k8	0,92	3538,80	3656,28	3598,61	<b>3509,84</b>	3592,72
16	B-n56-k7	0,88	2757,81	2858,67	<b>2668,34</b>	2719,70	2738,98
17	B-n57-k9	0,89	5040,16	5040,16	<b>5008,80</b>	5108,34	5091,38
18	B-n68-k9	0,93	4941,04	<b>4724,92</b>	4875,32	4975,60	4736,14
19	B-n78-k10	0,94	5280,46	5381,08	5568,79	5521,98	<b>5154,38</b>
20	E-n22-k4	0,94	<b>923,16</b>	1020,67	971,06	939,19	995,10
21	E-n30-k3	0,94	2365,10	2168,63	2371,29	2245,63	<b>2155,03</b>
22	E-n76-k7	0,89	4487,28	<b>4350,98</b>	4590,52	4408,86	4383,13
23	E-n101-k14	0,93	4444,27	4472,19	4385,86	4467,49	<b>4378,16</b>
24	F-n135-k7	0,95	<b>9093,03</b>	9905,15	9676,93	9682,72	9849,76
25	M-n101-k10	0,91	4686,95	4673,14	4660,79	<b>4637,77</b>	4713,63
26	M-n151-k12	0,93	7659,29	<b>7552,86</b>	7664,91	7712,29	7799,41
27	M-n200-k17	0,94	8947,63	9001,09	<b>8913,16</b>	9170,96	9021,92
28	P-n16-k8	0,88	<b>397,21</b>	<b>397,21</b>	421,13	<b>397,21</b>	<b>397,21</b>
29	P-n20-k2	0,97	<b>973,47</b>	1003,42	998,35	1037,30	1071,06
30	P-n45-k5	0,92	2707,22	2597,49	2689,52	2645,70	<b>2512,71</b>
31	P-n50-k10	0,95	2309,38	2115,63	2185,69	2236,03	<b>1972,45</b>
32	P-n55-k7	0,88	2691,96	2535,91	2545,71	2607,56	<b>2366,30</b>
33	P-n60-k10	0,95	2730,81	2668,77	2578,51	2553,60	<b>2512,26</b>
34	P-n65-k10	0,94	3053,26	2994,62	<b>2967,75</b>	2978,63	3029,06
35	P-n70-k10	0,97	5175,91	4637,19	4516,12	4786,92	<b>3908,06</b>
36	P-n76-k5	0,97	6008,94	5970,89	5984,95	5846,33	<b>5727,55</b>
37	P-n101-k4	0,91	9518,13	9257,82	<b>9008,79</b>	9212,82	9384,62
38	bays-n29-k5	0,93	<b>30626,43</b>	32341,01	37029,12	34674,12	34553,15
39	dantzig-n42-k4	0,93	5743,86	5735,68	5606,93	5573,36	<b>5487,83</b>

Çizelge 6.4. YAA ile elde edilen amaç fonksiyonu değerleri

#	Ad	% Sıklık	YA <sub>1</sub>	YA <sub>2</sub>	YA <sub>3</sub>	YA <sub>4</sub>	YA <sub>5</sub>
1	A-n32-k5	0,82	2291,07	<b>2281,73</b>	2291,07	2291,07	2291,07
2	A-n33-k5	0,89	2292,90	2298,84	2287,70	<b>2282,70</b>	2291,16
3	A-n37-k5	0,81	<b>2078,60</b>	2216,23	2216,23	2541,78	2233,14
4	A-n39-k6	0,88	2331,07	2351,16	2328,00	2355,73	<b>2313,71</b>
5	A-n48-k7	0,89	3517,39	3517,39	3419,07	<b>3239,95</b>	3244,44
6	A-n53-k7	0,95	3847,21	3504,98	3556,77	<b>3461,53</b>	3764,05
7	A-n62-k8	0,92	<b>4162,68</b>	4258,76	4257,42	4324,36	4235,81
8	A-n63-k9	0,97	<b>4854,53</b>	4973,77	4900,84	4900,84	4988,11
9	A-n69-k9	0,94	<b>3798,15</b>	<b>3798,15</b>	4493,79	3837,85	<b>3798,15</b>
10	A-n80-k10	0,94	6529,92	<b>6441,31</b>	6640,88	6457,49	6529,92
11	B-n31-k5	0,82	1848,51	1851,38	1840,93	1847,39	<b>1840,41</b>
12	B-n34-k5	0,91	<b>2425,55</b>	2446,42	2536,62	2532,03	2560,07
13	B-n43-k6	0,87	<b>2194,01</b>	2204,57	2218,71	2205,56	2221,03
14	B-n44-k7	0,92	2397,89	2421,77	2409,81	<b>2391,41</b>	2407,92
15	B-n50-k8	0,92	3075,23	3057,88	<b>3040,99</b>	3064,73	3044,74
16	B-n56-k7	0,88	2509,79	2516,41	2484,45	2541,64	<b>2443,04</b>
17	B-n57-k9	0,89	4703,85	4685,24	<b>4611,29</b>	4642,78	4614,15
18	B-n68-k9	0,93	4271,47	4213,66	<b>4160,95</b>	4366,31	4251,40
19	B-n78-k10	0,94	4313,84	4139,20	4109,39	<b>4049,26</b>	4160,41
20	E-n22-k4	0,94	911,48	911,31	901,75	900,31	<b>897,44</b>
21	E-n30-k3	0,94	<b>2108,09</b>	2139,65	<b>2108,09</b>	2109,46	2255,90
22	E-n76-k7	0,89	3609,72	3505,96	3462,81	3519,09	<b>3282,33</b>
23	E-n101-k14	0,93	3439,37	<b>3368,52</b>	3831,21	3523,05	3517,97
24	F-n135-k7	0,95	8243,79	8019,51	<b>7956,92</b>	8046,48	8637,82
25	M-n101-k10	0,91	<b>3837,28</b>	3873,02	3866,20	4080,87	4032,23
26	M-n151-k12	0,93	6316,04	6205,93	6135,55	<b>6106,88</b>	6600,88
27	M-n200-k17	0,94	7537,06	7489,18	<b>7479,32</b>	7619,76	7479,32
28	P-n16-k8	0,88	<b>408,99</b>	410,65	<b>408,99</b>	<b>408,99</b>	425,81
29	P-n20-k2	0,97	947,67	947,67	998,35	<b>946,34</b>	1063,41
30	P-n45-k5	0,92	2107,94	2197,02	<b>2021,64</b>	2276,06	2063,71
31	P-n50-k10	0,95	1503,22	1557,33	1467,80	<b>1408,28</b>	1500,74
32	P-n55-k7	0,88	1912,15	1946,11	1970,66	1983,28	<b>1878,98</b>
33	P-n60-k10	0,95	1876,05	1902,46	1889,16	1876,05	<b>1863,24</b>
34	P-n65-k10	0,94	2078,58	2107,18	2157,30	<b>1751,92</b>	1915,14
35	P-n70-k10	0,97	2582,36	2513,25	<b>2398,83</b>	2757,26	2748,15
36	P-n76-k5	0,97	5007,61	4887,98	<b>4595,77</b>	4892,02	4788,54
37	P-n101-k4	0,91	8067,14	<b>8054,54</b>	8143,04	8088,20	8066,16
38	bays-n29-k5	0,93	<b>28731,02</b>	29329,86	29329,86	<b>28731,02</b>	<b>28731,02</b>
39	dantzig-n42-k4	0,93	4164,19	4252,37	4317,26	<b>4137,49</b>	4142,67

Çizelge 6.2-6.4'de, ilk üç sütun sırasıyla test probleminin numarası, adı ve % sıklığını sonraki beş sütun ise (PSO<sub>i</sub>, GA<sub>i</sub> ve YAA<sub>i</sub>  $i=1,2,\dots,5$ ) her bir test problemi için sırasıyla PSO, GA ve YAA'nın  $i$ . koşturumunda bulunan eniyi amaç fonksiyonu değerlerini göstermektedir. Her bir problem için elde edilen eniyi değerler koyu olarak işaretlenmiştir.

Benzer şekilde Çizelge 6.5-6.7'de ilk üç sütun sırasıyla test probleminin numarası, adı ve % sıklığını temsil etmekte, sonraki 5 sütun ise (PSO<sub>i</sub>, GA<sub>i</sub> ve YAA<sub>i</sub>

$i=1,2,\dots,5$ ) her bir test problemi için sırasıyla PSO, GA ve YAA'nın  $i$ . koşturumunda elde edilen çözüm sürelerini (saniye olarak) göstermektedir. Koyu olarak işaretlenmiş değerler, elde edilmiş eniyi çözüm süreleridir.

**Çizelge 6.5.** PSO ile elde edilen çözüm süreleri (sn.)

#	Ad	% Sıklık	PSO <sub>1</sub>	PSO <sub>2</sub>	PSO <sub>3</sub>	PSO <sub>4</sub>	PSO <sub>5</sub>
1	A-n32-k5	0,82	301	315	291	<b>278</b>	311
2	A-n33-k5	0,89	313	<b>253</b>	290	460	329
3	A-n37-k5	0,81	521	547	<b>513</b>	541	528
4	A-n39-k6	0,88	535	438	<b>346</b>	379	371
5	A-n48-k7	0,89	650	533	470	<b>457</b>	576
6	A-n53-k7	0,95	<b>354</b>	485	445	527	508
7	A-n62-k8	0,92	876	839	744	834	<b>728</b>
8	A-n63-k9	0,97	492	<b>418</b>	592	530	485
9	A-n69-k9	0,94	843	582	650	<b>460</b>	611
10	A-n80-k10	0,94	1140	972	<b>594</b>	1132	959
11	B-n31-k5	0,82	427	330	274	<b>259</b>	352
12	B-n34-k5	0,91	307	392	<b>282</b>	369	432
13	B-n43-k6	0,87	396	489	<b>355</b>	477	553
14	B-n44-k7	0,92	378	465	405	<b>366</b>	530
15	B-n50-k8	0,92	719	753	709	<b>613</b>	750
16	B-n56-k7	0,88	520	638	462	<b>393</b>	721
17	B-n57-k9	0,89	785	614	526	<b>476</b>	691
18	B-n68-k9	0,93	625	<b>486</b>	581	758	737
19	B-n78-k10	0,94	1092	834	713	<b>647</b>	822
20	E-n22-k4	0,94	297	237	198	<b>181</b>	226
21	E-n30-k3	0,94	336	339	<b>250</b>	400	319
22	E-n76-k7	0,89	1062	812	700	<b>639</b>	780
23	E-n101-k14	0,93	1173	1204	882	1490	<b>867</b>
24	F-n135-k7	0,95	1827	<b>1536</b>	1800	1914	1553
25	M-n101-k10	0,91	1481	<b>1437</b>	1493	1465	1483
26	M-n151-k12	0,93	2218	2323	<b>2135</b>	2242	2306
27	M-n200-k17	0,94	3057	3057	3115	<b>3014</b>	3049
28	P-n16-k8	0,88	197	188	<b>138</b>	232	<b>138</b>
29	P-n20-k2	0,97	266	213	182	<b>164</b>	205
30	P-n45-k5	0,92	613	478	405	<b>370</b>	470
31	P-n50-k10	0,95	582	581	<b>423</b>	686	545
32	P-n55-k7	0,88	<b>517</b>	662	603	616	592
33	P-n60-k10	0,95	<b>545</b>	948	831	678	721
34	P-n65-k10	0,94	<b>700</b>	880	753	764	737
35	P-n70-k10	0,97	792	754	777	<b>698</b>	731
36	P-n76-k5	0,97	<b>840</b>	1057	907	914	893
37	P-n101-k4	0,91	1466	1127	1200	<b>1056</b>	1110
38	bays-n29-k5	0,93	378	358	407	<b>342</b>	418
39	dantzig-n42-k4	0,93	588	522	603	<b>503</b>	611



Çizelge 6.6. GA ile elde edilen çözüm süreleri (sn.)

#	Ad	% Sıklık	GA <sub>1</sub>	GA <sub>2</sub>	GA <sub>3</sub>	GA <sub>4</sub>	GA <sub>5</sub>
1	A-n32-k5	0,82	325	<b>307</b>	312	309	317
2	A-n33-k5	0,89	308	<b>234</b>	292	445	312
3	A-n37-k5	0,81	513	515	<b>499</b>	503	508
4	A-n39-k6	0,88	504	<b>277</b>	344	380	370
5	A-n48-k7	0,89	648	540	<b>529</b>	636	546
6	A-n53-k7	0,95	519	412	465	<b>335</b>	461
7	A-n62-k8	0,92	835	853	<b>687</b>	818	721
8	A-n63-k9	0,97	752	<b>401</b>	558	495	553
9	A-n69-k9	0,94	600	538	800	<b>460</b>	604
10	A-n80-k10	0,94	1087	<b>575</b>	886	1076	929
11	B-n31-k5	0,82	402	316	262	<b>256</b>	369
12	B-n34-k5	0,91	304	368	<b>272</b>	370	433
13	B-n43-k6	0,87	386	463	<b>369</b>	469	547
14	B-n44-k7	0,92	571	450	372	<b>364</b>	526
15	B-n50-k8	0,92	696	707	688	<b>594</b>	730
16	B-n56-k7	0,88	518	609	<b>471</b>	616	603
17	B-n57-k9	0,89	772	611	516	<b>450</b>	683
18	B-n68-k9	0,93	633	764	<b>577</b>	737	712
19	B-n78-k10	0,94	1070	854	716	<b>624</b>	943
20	E-n22-k4	0,94	301	237	197	<b>175</b>	229
21	E-n30-k3	0,94	343	339	<b>239</b>	406	320
22	E-n76-k7	0,89	1041	790	678	<b>609</b>	784
23	E-n101-k14	0,93	1204	1164	<b>824</b>	1406	1103
24	F-n135-k7	0,95	1801	<b>1456</b>	1780	1852	1522
25	M-n101-k10	0,91	1416	1404	<b>1383</b>	1428	1396
26	M-n151-k12	0,93	2215	<b>2114</b>	2259	2244	2170
27	M-n200-k17	0,94	3001	<b>2443</b>	2855	2891	2966
28	P-n16-k8	0,88	185	179	<b>138</b>	221	177
29	P-n20-k2	0,97	271	200	174	<b>165</b>	205
30	P-n45-k5	0,92	461	<b>371</b>	389	448	609
31	P-n50-k10	0,95	565	548	<b>421</b>	677	542
32	P-n55-k7	0,88	742	573	622	598	<b>565</b>
33	P-n60-k10	0,95	<b>645</b>	802	693	907	682
34	P-n65-k10	0,94	<b>707</b>	882	784	777	770
35	P-n70-k10	0,97	963	764	816	<b>730</b>	751
36	P-n76-k5	0,97	<b>819</b>	1032	859	885	868
37	P-n101-k4	0,91	1252	<b>1005</b>	1109	1059	<b>1005</b>
38	bays-n29-k5	0,93	419	455	412	<b>341</b>	430
39	dantzig-n42-k4	0,93	573	500	582	<b>492</b>	605

Çizelge 6.7. YAA ile elde edilen çözüm süreleri (sn.)

#	Ad	% Sıklık	YA <sub>1</sub>	YA <sub>2</sub>	YA <sub>3</sub>	YA <sub>4</sub>	YA <sub>5</sub>
1	A-n32-k5	0,82	536	<b>481</b>	795	571	507
2	A-n33-k5	0,89	1064	989	1216	975	<b>471</b>
3	A-n37-k5	0,81	581	<b>554</b>	574	557	598
4	A-n39-k6	0,88	<b>410</b>	587	824	1095	830
5	A-n48-k7	0,89	1108	920	817	887	<b>789</b>
6	A-n53-k7	0,95	<b>598</b>	608	606	1325	611
7	A-n62-k8	0,92	1352	1579	1471	1367	<b>1289</b>
8	A-n63-k9	0,97	<b>629</b>	662	1235	1359	665
9	A-n69-k9	0,94	1308	1302	<b>697</b>	1376	1311
10	A-n80-k10	0,94	1442	711	1475	721	<b>695</b>
11	B-n31-k5	0,82	<b>528</b>	540	562	542	567
12	B-n34-k5	0,91	556	<b>492</b>	1077	1276	1205
13	B-n43-k6	0,87	596	<b>581</b>	1186	1295	583
14	B-n44-k7	0,92	1025	1044	620	<b>590</b>	1173
15	B-n50-k8	0,92	629	1149	516	<b>454</b>	1096
16	B-n56-k7	0,88	1311	1282	<b>651</b>	1366	660
17	B-n57-k9	0,89	1391	1338	<b>1225</b>	1346	1268
18	B-n68-k9	0,93	1345	1339	1431	1362	<b>694</b>
19	B-n78-k10	0,94	1336	<b>696</b>	733	707	723
20	E-n22-k4	0,94	991	939	627	<b>382</b>	618
21	E-n30-k3	0,94	984	734	851	<b>551</b>	635
22	E-n76-k7	0,89	1126	514	<b>451</b>	914	1097
23	E-n101-k14	0,93	<b>642</b>	1263	1190	773	1160
24	F-n135-k7	0,95	663	988	666	712	<b>458</b>
25	M-n101-k10	0,91	795	<b>747</b>	1052	1341	744
26	M-n151-k12	0,93	960	945	<b>884</b>	906	942
27	M-n200-k17	0,94	<b>1125</b>	1191	1619	1188	1798
28	P-n16-k8	0,88	1222	<b>367</b>	1022	442	851
29	P-n20-k2	0,97	971	451	<b>350</b>	1168	960
30	P-n45-k5	0,92	1226	1297	<b>500</b>	1055	1212
31	P-n50-k10	0,95	1335	1340	1332	1240	<b>1059</b>
32	P-n55-k7	0,88	519	<b>512</b>	1138	522	1311
33	P-n60-k10	0,95	1088	<b>533</b>	752	546	1235
34	P-n65-k10	0,94	628	602	640	<b>538</b>	675
35	P-n70-k10	0,97	698	1117	688	684	<b>590</b>
36	P-n76-k5	0,97	991	695	705	910	<b>686</b>
37	P-n101-k4	0,91	<b>619</b>	743	657	643	657
38	bays-n29-k5	0,93	541	886	527	<b>365</b>	431
39	dantzig-n42-k4	0,93	607	506	505	<b>319</b>	389

Çizelge 6.2-6.7 incelendiğinde eniyi çözümün elde edildiği koşturmanın genellikle en kısa çözüm süresinin elde edildiği durum olmadığı görülmektedir. Kısacası eniyi çözüm değeri ile enkısa çözüm süresi arasında bir ödünleşmenin olduğu söylenebilir. Enkısa çözüm süresinin istenmesi halinde çözüm kalitesinden taviz verileceği anlaşılmaktadır. Öte yandan çözüm süreleri arasında en kötü değer 3115 saniyedir. Bu da yaklaşık 52 dakika anlamına gelmektedir. Kısacası bütün problemler için önerilen yöntemlerle 52 dakikanın altında çözüm bulunmuştur. Çözüm sürelerine genel olarak bakıldığında ise 39 problem için yapılan 5 koşturma sonucunda PSO için

ortalama çözüm süresinin 12,258 dakika, GA için 12,057 dakika ve YAA için 14,542 dakika olduğu hesaplanmıştır. Söz konusu problemlerin NP-zor sınıfında olduğu ve büyük boyutlar için çözüldüğü düşünülürse bu süre değerlerinin katlanılabilir oldukları açıktır. Ayrıca daha iyi donanım özellikleri olan bilgisayar kullanımı ve daha iyi kodlama ile bu süre değerlerinin azalabileceği söylenebilir. Dolayısıyla geliştirilen yaklaşımlarının tümü çözüm süresi açısından tatminkâr görünmektedir ve yöntemleri çözüm süreleri açısından karşılaştırmak anlamlı olmayacaktır. Bunun yerine elde edilen eniyi amaç fonksiyonu değeri ve ortalama amaç fonksiyonu değerleri açısından karşılaştırmak daha yararlı olacaktır.

Test problemlerinin çözümünden elde edilen ve Çizelge 6.2-6.4de verilen amaç fonksiyonu değerlerinden, her bir test probleminin PSO, GA ve YAA amaç fonksiyonu değerleri için eniyi ( $z^*$ ), enkötü ( $z_{enb}$ ) değerleri, değişim aralığı ( $R$ ), ortalamaları ( $z_{ort}$ ) ve standart sapmaları ( $S$ ) hesaplanmış ve Çizelge 6.8-6.10'da verilmiştir. Değişim aralığı değeri eniyi ile enkötü amaç fonksiyonu değerinin farkı olarak hesaplanmıştır.

Çizelge 6.8. PSO ile elde edilen amaç fonksiyonu değerleri için istatistikler

#	Ad	% sıklık	z*	zenb	R	z <sub>ort</sub>	S
1	A-n32-k5	0,82	2289,35	2421,89	132,54	2345,45	52,48
2	A-n33-k5	0,89	1771,17	1994,73	223,56	1828,27	94,85
3	A-n37-k5	0,81	2173,50	2548,90	375,40	2294,34	155,18
4	A-n39-k6	0,88	2317,29	2571,82	254,53	2418,45	97,23
5	A-n48-k7	0,89	3283,04	3584,51	301,47	3485,67	120,90
6	A-n53-k7	0,95	3535,07	3886,57	351,50	3721,15	143,89
7	A-n62-k8	0,92	4327,24	4574,98	247,74	4479,57	105,83
8	A-n63-k9	0,97	5241,01	6227,24	986,23	5523,93	409,01
9	A-n69-k9	0,94	3781,63	4358,99	577,36	4046,66	216,81
10	A-n80-k10	0,94	6715,70	6836,06	120,36	6783,55	45,51
11	B-n31-k5	0,82	1865,77	1923,02	57,25	1882,60	23,59
12	B-n34-k5	0,91	2369,58	2484,81	115,23	2399,07	48,71
13	B-n43-k6	0,87	2254,87	2291,36	36,49	2270,84	16,94
14	B-n44-k7	0,92	2398,02	2861,71	463,69	2516,79	196,25
15	B-n50-k8	0,92	3137,79	3292,70	154,91	3233,98	59,91
16	B-n56-k7	0,88	2567,45	2637,15	69,70	2598,44	30,93
17	B-n57-k9	0,89	4618,60	4674,72	56,12	4645,36	22,23
18	B-n68-k9	0,93	4307,80	4647,60	339,80	4452,90	127,33
19	B-n78-k10	0,94	4155,75	4414,64	258,89	4287,67	95,88
20	E-n22-k4	0,94	855,23	1019,18	163,95	915,54	65,34
21	E-n30-k3	0,94	2026,62	2137,89	111,27	2074,14	42,64
22	E-n76-k7	0,89	3863,95	4163,85	299,90	4035,31	116,29
23	E-n101-k14	0,93	3618,70	4484,47	865,77	3859,11	353,70
24	F-n135-k7	0,95	8113,12	8453,88	340,76	8328,67	130,74
25	M-n101-k10	0,91	3929,71	4180,38	250,67	4037,89	100,53
26	M-n151-k12	0,93	6621,34	6927,92	306,58	6747,83	111,17
27	M-n200-k17	0,94	7604,23	8525,11	920,88	7919,08	358,09
28	P-n16-k8	0,88	397,21	416,91	19,70	403,92	8,07
29	P-n20-k2	0,97	989,07	1161,52	172,45	1069,63	76,25
30	P-n45-k5	0,92	2157,16	2449,49	292,33	2252,80	117,94
31	P-n50-k10	0,95	1576,94	1716,37	139,43	1625,09	55,52
32	P-n55-k7	0,88	2121,20	2385,26	264,06	2268,70	114,34
33	P-n60-k10	0,95	1716,37	2154,47	438,10	2033,21	182,21
34	P-n65-k10	0,94	2323,18	2617,75	294,57	2473,61	104,79
35	P-n70-k10	0,97	2635,35	4028,711393,36		3087,57	546,81
36	P-n76-k5	0,97	4967,68	5362,50	394,82	5168,24	176,67
37	P-n101-k4	0,91	8470,78	9029,58	558,80	8776,82	221,46
38	bays-n29-k5	0,93	29886,9331449,251562,32			30526,99	665,23
39	dantzig-n42-k4	0,93	4264,12	4694,01	429,89	4522,88	167,06

Çizelge 6.9. GA ile elde edilen amaç fonksiyonu değerleri için istatistikler

#	Ad	% sıklık	z*	zenb	R	zort	S
1	A-n32-k5	0,82	2405,50	2542,41	136,91	2463,20	53,85
2	A-n33-k5	0,89	1896,99	2131,92	234,93	1960,26	96,71
3	A-n37-k5	0,81	2401,37	2933,93	532,56	2665,42	203,25
4	A-n39-k6	0,88	2899,89	3483,33	583,44	3066,06	240,55
5	A-n48-k7	0,89	3783,90	4401,65	617,75	4042,04	225,33
6	A-n53-k7	0,95	4361,36	4616,52	255,16	4491,83	113,27
7	A-n62-k8	0,92	5404,43	5818,73	414,30	5591,65	176,46
8	A-n63-k9	0,97	7239,45	8695,99	1456,54	8052,13	578,59
9	A-n69-k9	0,94	4740,83	5079,92	339,09	4952,17	130,12
10	A-n80-k10	0,94	7461,42	7812,18	350,76	7677,99	152,16
11	B-n31-k5	0,82	1904,38	1957,87	53,49	1918,84	22,35
12	B-n34-k5	0,91	2356,20	2538,51	182,31	2442,48	77,58
13	B-n43-k6	0,87	2310,32	2427,11	116,79	2383,13	47,00
14	B-n44-k7	0,92	2539,02	2812,50	273,48	2709,75	113,24
15	B-n50-k8	0,92	3509,84	3656,28	146,44	3579,25	56,89
16	B-n56-k7	0,88	2668,34	2858,67	190,33	2748,70	69,95
17	B-n57-k9	0,89	5008,80	5108,34	99,54	5057,77	40,94
18	B-n68-k9	0,93	4724,92	4975,60	250,68	4850,60	115,45
19	B-n78-k10	0,94	5154,38	5568,79	414,41	5381,34	170,74
20	E-n22-k4	0,94	923,16	1020,67	97,51	969,84	39,82
21	E-n30-k3	0,94	2155,03	2371,29	216,26	2261,14	103,68
22	E-n76-k7	0,89	4350,98	4590,52	239,54	4444,15	96,09
23	E-n101-k14	0,93	4378,16	4472,19	94,03	4429,59	44,79
24	F-n135-k7	0,95	9093,03	9905,15	812,12	9641,52	322,74
25	M-n101-k10	0,91	4637,77	4713,63	75,86	4674,46	28,39
26	M-n151-k12	0,93	7552,86	7799,41	246,55	7677,75	89,58
27	M-n200-k17	0,94	8913,16	9170,96	257,80	9010,95	99,24
28	P-n16-k8	0,88	397,21	421,13	23,92	401,99	10,70
29	P-n20-k2	0,97	973,47	1071,06	97,59	1016,72	37,95
30	P-n45-k5	0,92	2512,71	2707,22	194,51	2630,53	78,37
31	P-n50-k10	0,95	1972,45	2309,38	336,93	2163,84	128,28
32	P-n55-k7	0,88	2366,30	2691,96	325,66	2549,49	119,82
33	P-n60-k10	0,95	2512,26	2730,81	218,55	2608,79	89,12
34	P-n65-k10	0,94	2967,75	3053,26	85,51	3004,66	35,70
35	P-n70-k10	0,97	3908,06	5175,91	1267,85	4604,84	461,98
36	P-n76-k5	0,97	5727,55	6008,94	281,39	5907,73	118,78
37	P-n101-k4	0,91	9008,79	9518,13	509,34	9276,44	191,15
38	bays-n29-k5	0,93	30626,43	37029,12	6402,69	33844,70	2446,87
39	dantzig-n42-k4	0,93	5487,83	5743,86	256,03	5629,53	109,64

Çizelge 6.10. YAA amaç fonksiyonu değerleri için istatistikler

#	Ad	% sıklık	z*	zenb	R	z <sub>ort</sub>	S
1	A-n32-k5	0,82	2281,73	2291,07	9,34	2289,20	4,18
2	A-n33-k5	0,89	2282,70	2298,84	16,14	2290,66	6,01
3	A-n37-k5	0,81	2078,60	2541,78	463,18	2257,20	170,89
4	A-n39-k6	0,88	2313,71	2355,73	42,02	2335,93	17,35
5	A-n48-k7	0,89	3239,95	3517,39	277,44	3387,65	138,72
6	A-n53-k7	0,95	3461,53	3847,21	385,68	3626,91	169,17
7	A-n62-k8	0,92	4162,68	4324,36	161,68	4247,81	58,02
8	A-n63-k9	0,97	4854,53	4988,11	133,58	4923,62	55,87
9	A-n69-k9	0,94	3798,15	4493,79	695,64	3945,22	307,14
10	A-n80-k10	0,94	6441,31	6640,88	199,57	6519,90	78,91
11	B-n31-k5	0,82	1840,41	1851,38	10,97	1845,72	4,84
12	B-n34-k5	0,91	2425,55	2560,07	134,52	2500,14	59,98
13	B-n43-k6	0,87	2194,01	2221,03	27,02	2208,78	11,12
14	B-n44-k7	0,92	2391,41	2421,77	30,36	2405,76	11,68
15	B-n50-k8	0,92	3040,99	3075,23	34,24	3056,71	14,13
16	B-n56-k7	0,88	2443,04	2541,64	98,60	2499,07	37,35
17	B-n57-k9	0,89	4611,29	4703,85	92,56	4651,46	41,73
18	B-n68-k9	0,93	4160,95	4366,31	205,36	4252,76	76,16
19	B-n78-k10	0,94	4049,26	4313,84	264,58	4154,42	98,45
20	E-n22-k4	0,94	897,44	911,48	14,04	904,46	6,52
21	E-n30-k3	0,94	2108,09	2255,90	147,81	2144,24	63,86
22	E-n76-k7	0,89	3282,33	3609,72	327,39	3475,98	120,75
23	E-n101-k14	0,93	3368,52	3831,21	462,69	3536,02	176,80
24	F-n135-k7	0,95	7956,92	8637,82	680,90	8180,90	277,04
25	M-n101-k10	0,91	3837,28	4080,87	243,59	3937,92	110,47
26	M-n151-k12	0,93	6106,88	6600,88	494,00	6273,06	200,22
27	M-n200-k17	0,94	7479,32	7619,76	140,44	7520,93	60,20
28	P-n16-k8	0,88	408,99	425,81	16,82	412,69	7,37
29	P-n20-k2	0,97	946,34	1063,41	117,07	980,69	51,27
30	P-n45-k5	0,92	2021,64	2276,06	254,42	2133,27	102,95
31	P-n50-k10	0,95	1408,28	1557,33	149,05	1487,47	54,69
32	P-n55-k7	0,88	1878,98	1983,28	104,30	1938,24	42,82
33	P-n60-k10	0,95	1863,24	1902,46	39,22	1881,39	14,92
34	P-n65-k10	0,94	1751,92	2157,30	405,38	2002,02	166,68
35	P-n70-k10	0,97	2398,83	2757,26	358,43	2599,97	154,10
36	P-n76-k5	0,97	4595,77	5007,61	411,84	4834,38	154,30
37	P-n101-k4	0,91	8054,54	8143,04	88,50	8083,82	35,26
38	bays-n29-k5	0,93	28731,02	29329,86	598,84	28970,56	328,00
39	dantzig-n42-k4	0,93	4137,49	4317,26	179,77	4202,80	78,95

Çizelge 6.8-6.10 incelendiğinde önerilen her 3 yaklaşım için de problemin boyutu ve % sıklık değeri arttıkça elde edilen çözüm değerleri arasındaki sapmanın ( $R$  ve  $S$  değerlerinin) artış eğiliminde olduğu gözlenmektedir. Aslında bu da beklenen bir sonuçtur.

Bu üç algoritmanın karşılaştırmalı amaç fonksiyonu değerleri ise Çizelge 6.11'de sunulmuştur.

**Çizelge 6.11.** PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerlerinin eniyi ve ortalama değerler açısından karşılaştırması.

#	Ad	% sıklık	PSO		GA		YAA	
			z*	z <sub>ort</sub>	z*	z <sub>ort</sub>	z*	z <sub>ort</sub>
1	A-n32-k5	0,82	2289,35	2345,45	2405,50	2463,21	<b>2281,73</b>	<b>2289,20</b>
2	A-n33-k5	0,89	<b>1771,17</b>	<b>1828,27</b>	1896,99	1960,26	2282,70	2290,66
3	A-n37-k5	0,81	2173,50	2294,34	2401,37	2665,42	<b>2078,60</b>	<b>2257,20</b>
4	A-n39-k6	0,88	2317,29	2418,45	2899,89	3066,06	<b>2313,71</b>	<b>2335,93</b>
5	A-n48-k7	0,89	3283,04	3485,67	3783,90	4042,04	<b>3239,95</b>	<b>3387,65</b>
6	A-n53-k7	0,95	3535,07	3721,15	4361,36	4491,83	<b>3461,53</b>	<b>3626,91</b>
7	A-n62-k8	0,92	4327,24	4479,57	5404,43	5591,65	<b>4162,68</b>	<b>4247,81</b>
8	A-n63-k9	0,97	5241,01	5523,93	7239,45	8052,13	<b>4854,53</b>	<b>4923,62</b>
9	A-n69-k9	0,94	<b>3781,63</b>	4046,66	4740,83	4952,17	3798,15	<b>3945,22</b>
10	A-n80-k10	0,94	6715,70	6783,55	7461,42	7677,99	<b>6441,31</b>	<b>6519,90</b>
11	B-n31-k5	0,82	1865,77	1882,60	1904,38	1918,84	<b>1840,41</b>	<b>1845,72</b>
12	B-n34-k5	0,91	2369,58	<b>2399,07</b>	<b>2356,20</b>	2442,48	2425,55	2500,14
13	B-n43-k6	0,87	2254,87	2270,84	2310,32	2383,13	<b>2194,01</b>	<b>2208,78</b>
14	B-n44-k7	0,92	2398,02	2516,79	2539,02	2709,75	<b>2391,41</b>	<b>2405,76</b>
15	B-n50-k8	0,92	3137,79	3233,98	3509,84	3579,25	<b>3040,99</b>	<b>3056,71</b>
16	B-n56-k7	0,88	2567,45	2598,44	2668,34	2748,70	<b>2443,04</b>	<b>2499,07</b>
17	B-n57-k9	0,89	4618,60	<b>4645,36</b>	5008,80	5057,77	<b>4611,29</b>	4651,46
18	B-n68-k9	0,93	4307,80	4452,90	4724,92	4850,60	<b>4160,95</b>	<b>4252,76</b>
19	B-n78-k10	0,94	4155,75	4287,67	5154,38	5381,34	<b>4049,26</b>	<b>4154,42</b>
20	E-n22-k4	0,94	<b>855,23</b>	915,54	923,16	969,84	897,44	<b>904,46</b>
21	E-n30-k3	0,94	<b>2026,62</b>	<b>2074,14</b>	2155,03	2261,14	2108,09	2144,24
22	E-n76-k7	0,89	3863,95	4035,31	4350,98	4444,15	<b>3282,33</b>	<b>3475,98</b>
23	E-n101-k14	0,93	3618,70	3859,11	4378,16	4429,59	<b>3368,52</b>	<b>3536,02</b>
24	F-n135-k7	0,95	8113,12	8328,67	9093,03	9641,52	<b>7956,92</b>	<b>8180,90</b>
25	M-n101-k10	0,91	3929,71	4037,89	4637,77	4674,46	<b>3837,28</b>	<b>3937,92</b>
26	M-n151-k12	0,93	6621,34	6747,83	7552,86	7677,75	<b>6106,88</b>	<b>6273,06</b>
27	M-n200-k17	0,94	7604,23	7919,08	8913,16	9010,95	<b>7479,32</b>	<b>7520,93</b>
28	P-n16-k8	0,88	<b>397,21</b>	403,92	<b>397,21</b>	<b>401,99</b>	408,99	412,69
29	P-n20-k2	0,97	989,07	1069,63	973,47	1016,72	<b>946,34</b>	<b>980,69</b>
30	P-n45-k5	0,92	2157,16	2252,80	2512,71	2630,53	<b>2021,64</b>	<b>2133,27</b>
31	P-n50-k10	0,95	1576,94	1625,09	1972,45	2163,84	<b>1408,28</b>	<b>1487,47</b>
32	P-n55-k7	0,88	2121,20	2268,70	2366,30	2549,49	<b>1878,98</b>	<b>1938,24</b>
33	P-n60-k10	0,95	<b>1716,37</b>	2033,21	2512,26	2608,79	1863,24	<b>1881,39</b>
34	P-n65-k10	0,94	2323,18	2473,61	2967,75	3004,66	<b>1751,92</b>	<b>2002,02</b>
35	P-n70-k10	0,97	2635,35	3087,57	3908,06	4604,84	<b>2398,83</b>	<b>2599,97</b>
36	P-n76-k5	0,97	4967,68	5168,24	5727,55	5907,73	<b>4595,77</b>	<b>4834,38</b>
37	P-n101-k4	0,91	8470,78	8776,82	9008,79	9276,44	<b>8054,54</b>	<b>8083,82</b>
38	bays-n29-k5	0,93	29886,93	30526,99	30626,43	33844,77	<b>28731,02</b>	<b>28970,56</b>
39	dantzig-n42-k4	0,93	4264,12	4522,88	5487,83	5629,53	<b>4137,49</b>	<b>4202,80</b>

Çizelge 6.11’de her bir test problemi satırında, ortalamalar ve eniyi çözüm değeri açısından elde edilmiş enküçük değerler koyu olarak işaretlenmiştir. Örnek olarak 9 numaralı test probleminde eniyi amaç fonksiyonu değeri, bir başka ifadeyle üç algoritmanın erişebildiği eniyi değer PSO’ya aittir. Aynı test probleminde ortalama amaç fonksiyonu değerlerine bakıldığında YAA’nın diğerlerine göre daha üstün olduğu görülmektedir.

Bütün test problemleri göz önüne alındığında 39 test problemi arasında eniyi çözüm değeri açısından 32 problemde YAA'nın diğerlerinden üstün olduğu, 6 problemde PSO'nun ve sadece 2 problemde de GA'nın diğerlerinden daha iyi sonuç verdiği gözlenmektedir. Üstelik 28 numaralı problemde GA ile elde edilen değer PSO ile de bulunmuştur. Amaç fonksiyonu ortalama değerleri açısından da durum benzerdir. 39 problemin 32'sinde YAA, 4'ünde PSO ve sadece 1 tanesinde GA eniyi değerlere erişmiştir. Bu sonuçlara göre YAA'nın BAARP için bariz bir şekilde önerilen diğer yöntemlerden daha iyi olduğu anlaşılmaktadır.

Test problemlerinin çözümleri elde edilmeden önce PSO'nun genetik algoritmaya göre daha gelişkin özellikler içermesi nedeniyle GA'dan daha başarılı olacağı tahmin edilmiştir. Nitekim 39 test probleminin 36 tanesinde amaç fonksiyonunun eniyi değeri açısından PSO, GA'dan daha iyi sonuç üretmiştir. Sadece 2 problemde GA üstün gelmiş ve 1 problemde de aynı değeri bulmuşlardır. Bu beklenen bir sonuç iken YAA'nın PSO karşısında bariz üstün bir performans göstermesi şaşırtıcı olmuştur.

### 6.3 Sonuçların Tekrar Edilebilirliği Ve Yakınsaklık

Eniyi amaç fonksiyonu değerleriyle birlikte PSO, GA ve YAA ile her bir test probleminde türetilen bu değerlerin tekrar edilebilirliğinin test edilmesi de önem taşımaktadır. Bu nedenle test problemleri A, B, E, P, M ve TSPLIB isimlerinde olmak üzere sınıflara ayrılmış ve ilgili sınıflar bazında değerlendirmeler izleyen iki çizelgede verilmiştir.

Çizelge 6.12'de PSO, GA, YAA ile elde edilen amaç fonksiyonu değerlerinin aralıkları test problemi sınıfı bazında incelenmiş, üç yöntemin amaç fonksiyonu değerlerine yönelik enküçük aralık ( $R_{enk}$ ), enbüyük aralık ( $R_{enb}$ ) ve ortalama aralık ( $R_{ort}$ ) değerleri araştırılmış ve ortalama aralık değeri sütununda enküçük ortalama değeri koyu olarak işaretlenmiştir. Beklenen  $R$  değerinin küçük olması istendiği için de her grup için elde edilen enküçük  $R_{ort}$  değerleri koyu olarak işaretlenmiştir.



**Çizelge 6.12.** PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerleri için aralık araştırması

	<i>Renk</i>			<i>Renb</i>			<i>R<sub>ort</sub></i>		
	PSO	GA	YAA	PSO	GA	YAA	PSO	GA	YAA
<b>A</b>	120,36	136,91	9,34	986,23	1456,54	695,64	357,07	492,14	<b>238,43</b>
<b>B</b>	36,49	53,49	10,97	463,69	414,41	264,58	172,45	191,94	<b>99,80</b>
<b>E</b>	111,27	94,03	14,04	865,77	239,54	462,69	360,22	<b>161,84</b>	237,98
<b>M</b>	250,67	75,86	140,44	920,88	257,80	494,00	492,71	<b>193,40</b>	292,68
<b>P</b>	19,70	23,92	16,82	1393,36	1267,85	411,84	396,76	334,13	<b>194,50</b>
<b>TSPLIB</b>	429,89	256,03	179,77	1562,32	6402,69	598,84	996,11	3329,36	<b>389,31</b>

Çizelgedeki değerleri açıklamak gerekirse, örneğin  $R_{ort}$  PSO sütunu incelendiğinde A sınıfından 10 adet test probleminin her birinin 5 koşurumu sonrasında elde edilen amaç fonksiyonlarının aralık değerlerinin ortalamasının 357,07 olduğu anlaşılmaktadır. Bu değer A sınıfı problemlerde PSO'nun ortalama amaç fonksiyonu aralık değeridir ve aynı zamanda Çizelge 6.8'deki  $R$  sütununun ilk 10 satırı ortalamasıdır. Benzer şekilde A grubu için YAA ile elde edilen değer ise 238,43'tür ve bu, A grubu problemlerde önerilen 3 yaklaşım içindeki  $R_{ort}$  eniyi değeridir.

Çizelge 6.12'ye göre amaç fonksiyonu ortalama aralık değerleri bakımından, A, B, P ve TSPLIB test problemleri sınıflarında YAA; E ve M test problemleri sınıfında ise GA enküçük  $R_{ort}$  değerlerine erişmiştir. Bu da yine YAA'nın diğerlerine göre daha güvenilir sonuçlar verdiğini göstermektedir. PSO'nun bu değerlendirmede bir varlık gösterememesi şaşırtıcıdır ve yöntemin biraz daha iyileştirmeye ihtiyacı olduğunu düşündürmektedir.

Aynı araştırma, aynı sınıflandırma yaklaşımıyla, üç algoritmanın elde ettiği amaç fonksiyonu değerlerinin standart sapmalarının incelenmesi amacıyla yapılmış ve sonuçlar Çizelge 6.13'de verilmiştir.  $SS_{ort}$  sütunundaki enküçük ortalama standart sapma değerleri koyu olarak işaretlenmiştir.

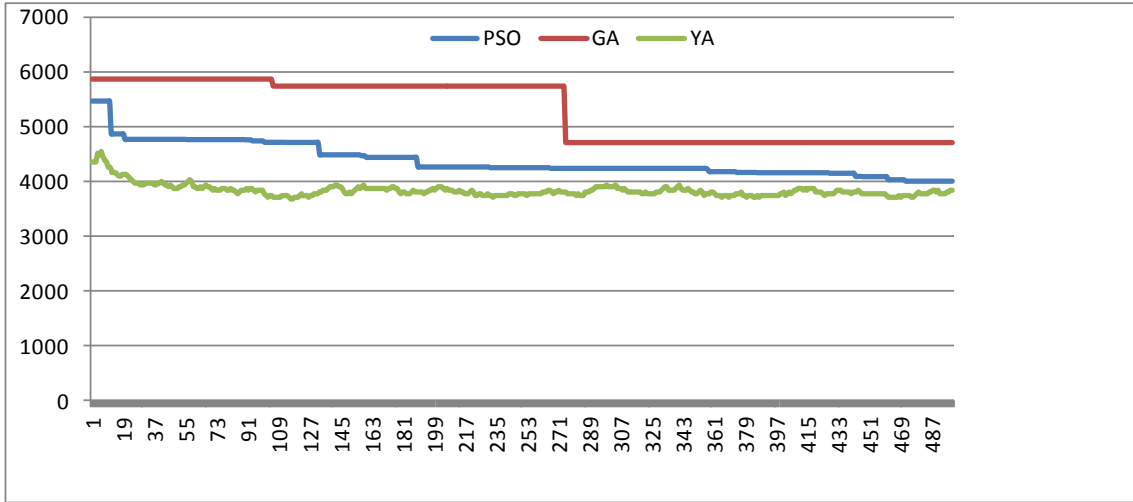
**Çizelge 6.13.** PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerleri için standart sapma araştırması

	$SS_{enk}$			$SS_{enb}$			$SS_{ort}$		
	PSO	GA	YAA	PSO	GA	YAA	PSO	GA	YAA
<b>A</b>	45,52	53,85	4,18	409,02	578,59	307,14	144,17	197,03	<b>100,63</b>
<b>B</b>	16,94	22,35	4,84	196,26	170,74	98,45	69,09	79,35	<b>39,49</b>
<b>E</b>	42,64	39,82	6,52	353,71	103,68	176,80	144,50	<b>71,09</b>	91,98
<b>M</b>	100,53	28,39	60,20	358,10	99,24	200,22	189,94	<b>72,40</b>	123,63
<b>P</b>	8,08	10,70	7,37	546,81	461,98	166,68	160,41	127,18	<b>78,44</b>
<b>TSPLIB</b>	167,06	109,64	78,95	665,24	2446,87	328,00	416,15	1278,26	<b>203,47</b>

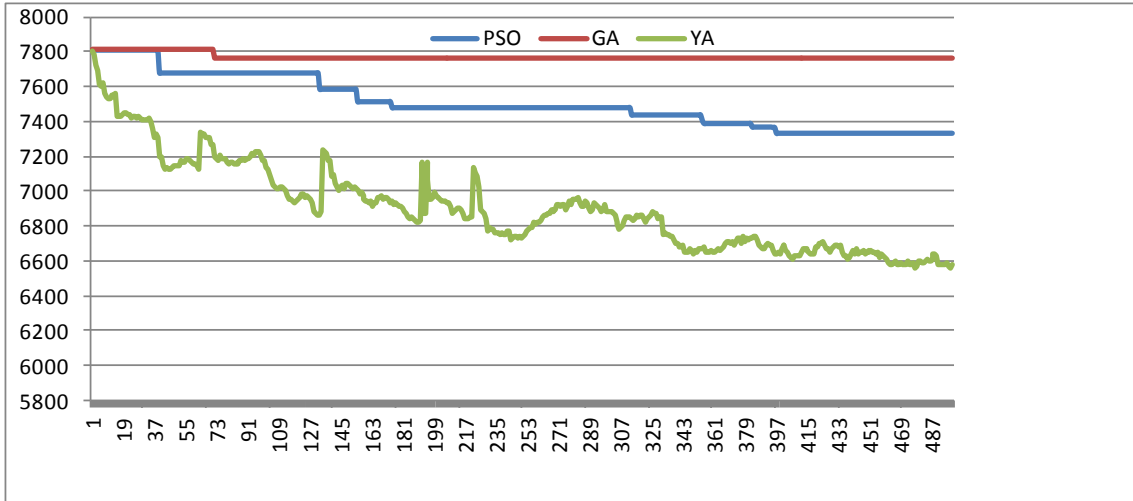
Çizelge 6.13’de  $SS_{enk}$  sütunu ilgili test problemi sınıfının PSO, GA ve YAA ile elde edilen amaç fonksiyonu değerlerinin standart sapmasının enküçüğünü,  $SS_{enb}$  sütunu enbüyüğünü,  $SS_{ort}$  sütunu ise standart sapmaların ortalamalarını temsil etmektedir.

Çizelge 6.13’den amaç fonksiyonu ortalama standart sapma değerleri bakımından, A, B, P ve TSPLIB test problemleri sınıflarında beklendiği gibi YAA’nın; E ve M test problemleri sınıfında ise GA’nın enküçük  $SS_{ort}$  değerlerine ulaştığı gözlenmektedir. Bu da amaç fonksiyonu değerlerinin tekrar edilebilirliği bakımından YAA diğer algoritmalara göre daha başarılı olduğunu göstermektedir.

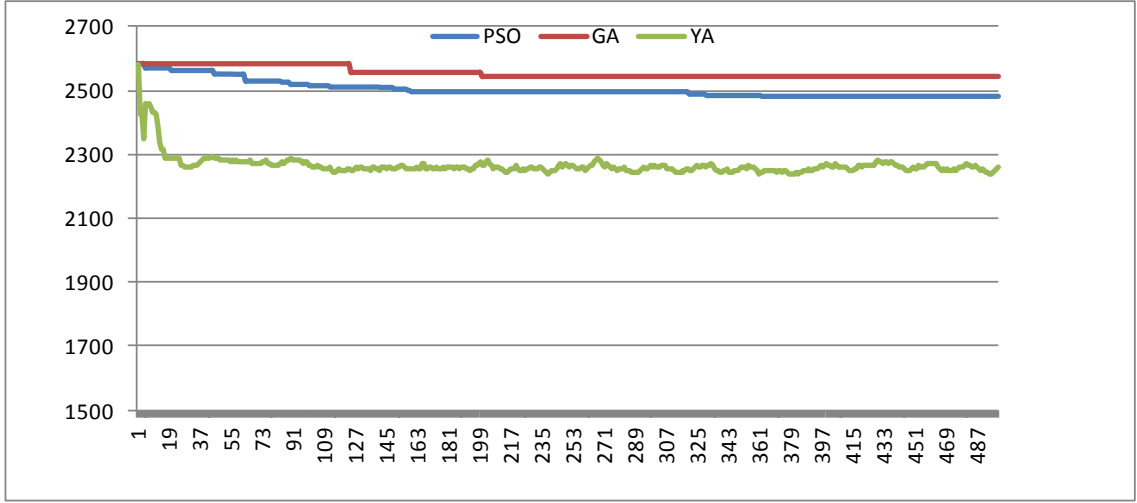
Algoritmaların performanslarının yakınsaklık testlerinde de incelenmesinin faydalı olacağı düşünülmüştür. Bu nedenle her üç algoritma için ilk 500 ardıştırmayı içeren yakınsama grafikleri bazı test problemleri üzerinde Şekil 6.1-6.4’de verilmiştir. Bu şekillerde apsis ardıştırma sayılarına, ordinat ise amaç fonksiyonu değerlerine karşılık gelmektedir.



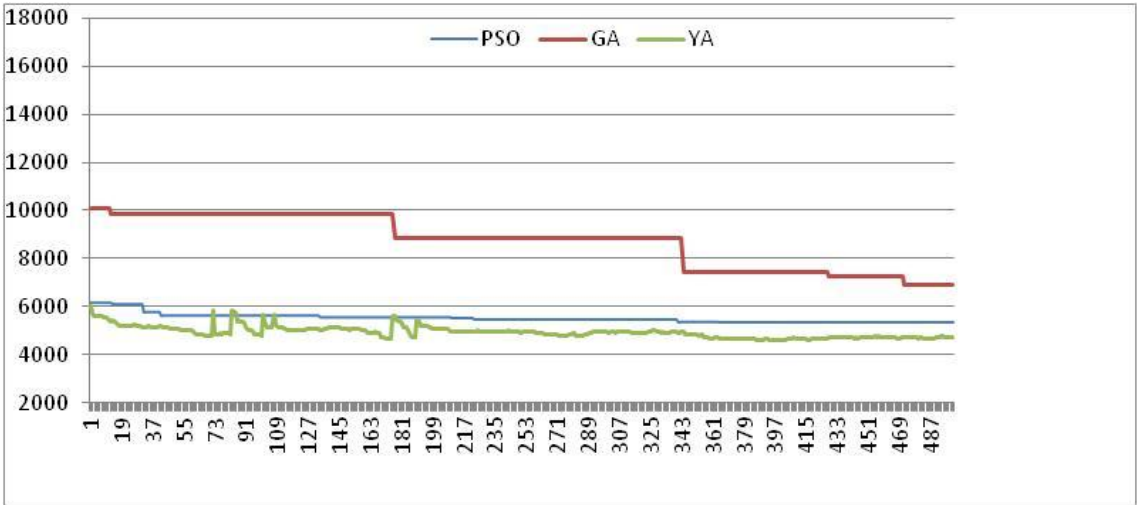
Şekil 6.1. A-n53-k7 için yöntemlerin yakınsama grafikleri



Şekil 6.2. A-n80-k10 için yöntemlerin yakınsama grafikleri



Şekil 6.3. B-n34-k5 için yöntemlerin yakınsama grafikleri



Şekil 6.4. P-n76-k5 için yöntemlerin yakınsama grafikleri.

Şekil 6.1-6.4'ten ilgili test problemlerinde YAA'nın diğerlerinden daha kararlı bir yakınsama sergilediği görülmektedir. GA ve PSO'nun yakınsama sürecinin ise beklendiği gibi adım adım olduğu görülmektedir. 4 örnek için de YAA'nın diğer iki yöntemden daha başarılı bir yakınsama sergilediği açıktır.

Son olarak yöntemlerin erişebildiği eniyi amaç fonksiyonu değerleri aşağıda Çizelge 6.14’de verilmiştir.

**Çizelge 6.14.** Yüzde uzaklıklar çizelgesi

#	Ad	% sıklık	PSO	GA	YAA	% $U_1$	% $U_2$
1	A-n32-k5	0,82	2289,35	2405,50	<b>2281,73</b>	0,33	5,42
2	A-n33-k5	0,89	<b>1771,17</b>	1896,99	2282,70	7,10	28,88
3	A-n37-k5	0,81	2173,50	2401,37	<b>2078,60</b>	4,57	15,53
4	A-n39-k6	0,88	2317,29	2899,89	<b>2313,71</b>	0,15	25,34
5	A-n48-k7	0,89	3283,04	3783,90	<b>3239,95</b>	1,33	16,79
6	A-n53-k7	0,95	3535,07	4361,36	<b>3461,53</b>	2,12	26,00
7	A-n62-k8	0,92	4327,24	5404,43	<b>4162,68</b>	3,95	29,83
8	A-n63-k9	0,97	5241,01	7239,45	<b>4854,53</b>	7,96	49,13
9	A-n69-k9	0,94	<b>3781,63</b>	4740,83	3798,15	0,44	25,36
10	A-n80-k10	0,94	6715,70	7461,42	<b>6441,31</b>	4,26	15,84
11	B-n31-k5	0,82	1865,77	1904,38	<b>1840,41</b>	1,38	3,48
12	B-n34-k5	0,91	2369,58	<b>2356,20</b>	2425,55	0,57	2,94
13	B-n43-k6	0,87	2254,87	2310,32	<b>2194,01</b>	2,77	5,30
14	B-n44-k7	0,92	2398,02	2539,02	<b>2391,41</b>	0,28	6,17
15	B-n50-k8	0,92	3137,79	3509,84	<b>3040,99</b>	3,18	15,42
16	B-n56-k7	0,88	2567,45	2668,34	<b>2443,04</b>	5,09	9,22
17	B-n57-k9	0,89	4618,60	5008,80	<b>4611,29</b>	0,16	8,62
18	B-n68-k9	0,93	4307,80	4724,92	<b>4160,95</b>	3,53	13,55
19	B-n78-k10	0,94	4155,75	5154,38	<b>4049,26</b>	2,63	27,29
20	E-n22-k4	0,94	<b>855,23</b>	923,16	897,44	4,94	7,94
21	E-n30-k3	0,94	<b>2026,62</b>	2155,03	2108,09	4,02	6,34
22	E-n76-k7	0,89	3863,95	4350,98	<b>3282,33</b>	17,72	32,56
23	E-n101-k14	0,93	3618,70	4378,16	<b>3368,52</b>	7,43	29,97
24	F-n135-k7	0,95	8113,12	9093,03	<b>7956,92</b>	1,96	14,28
25	M-n101-k10	0,91	3929,71	4637,77	<b>3837,28</b>	2,41	20,86
26	M-n151-k12	0,93	6621,34	7552,86	<b>6106,88</b>	8,42	23,68
27	M-n200-k17	0,94	7604,23	8913,16	<b>7479,32</b>	1,67	19,17
28	P-n16-k8	0,88	<b>397,21</b>	<b>397,21</b>	408,99	0,00	2,97
29	P-n20-k2	0,97	989,07	973,47	<b>946,34</b>	2,87	4,52
30	P-n45-k5	0,92	2157,16	2512,71	<b>2021,64</b>	6,70	24,29
31	P-n50-k10	0,95	1576,94	1972,45	<b>1408,28</b>	11,98	40,06
32	P-n55-k7	0,88	2121,20	2366,30	<b>1878,98</b>	12,89	25,94
33	P-n60-k10	0,95	<b>1716,37</b>	2512,26	1863,24	8,56	46,37
34	P-n65-k10	0,94	2323,18	2967,75	<b>1751,92</b>	32,61	69,40
35	P-n70-k10	0,97	2635,35	3908,06	<b>2398,83</b>	9,86	62,92
36	P-n76-k5	0,97	4967,68	5727,55	<b>4595,77</b>	8,09	24,63
37	P-n101-k4	0,91	8470,78	9008,79	<b>8054,54</b>	5,17	11,85
38	bays-n29-k5	0,93	29886,93	30626,43	<b>28731,02</b>	4,02	6,60
39	dantzig-n42-k4	0,93	4264,12	5487,83	<b>4137,49</b>	3,06	32,64

Çizelge 6.14’deki ilk üç sütun ilgili test problemi ile ilgili bilgileri sonraki üç sütun sırasıyla PSO, GA ve YAA’nın erişebildiği eniyi amaç fonksiyonu değerlerini vermektedir. %  $U_1$  sütunu, ilgili test probleminde eniyi amaç fonksiyonu değeri bakımından eniyi değere sahip algoritmanın, kendini izleyen algoritmadan yüzde

uzaklığına, % $U_2$  sütunu ise eniyi amaç fonksiyonu değeri ile enkötü amaç fonksiyonu değerleri arasındaki yüzde uzaklığına karşılık gelmektedir. Bu sütunlardaki değerler aşağıdaki formülasyona göre hesaplanmıştır.

$z_i$ :  $i$ . algoritmanın amaç fonksiyonu değeri,  $i=1,2,3$  ve örnek olarak  $z_1 < z_2 < z_3$  iken,

$$U_1 = \frac{100(z_2 - z_1)}{z_1} \quad (6.2)$$

$$U_2 = \frac{100(z_3 - z_1)}{z_1} \quad (6.3)$$

Çizelge 6.14’de, algoritmalar arasında elde edilen eniyi amaç fonksiyonu değeri ise koyu olarak işaretlenmiştir. Örnek olarak 1. test probleminde YAA’nın elde edilen amaç fonksiyonu eniyi değeri açısından PSO’dan %0,33, GA’dan ise %5,42 daha iyi sonuç ürettiği anlaşılmaktadır.

Tüm problemler incelendiğinde enbüyük farklılaşmanın 34 numaralı test probleminde olduğu görülmektedir. Bu problemde YAA’nın, PSO’ya göre %32,61 ve GA’ya göre %69,40 daha iyi sonuçlar elde ettiği görülmektedir. Çizelge 6.14 geneline bakıldığında ise YAA’nın kendisine yakın rakibinden ortalama olarak %10,25 kadar diğer rakibinden ise ortalama olarak %22,38 kadar daha iyi sonuçlar elde ettiği gözlenmektedir.

Çizelge 6.15’de test problemleri % sıklıklarına göre gruplandırılmış ve her gruptaki problemler boyut artışına göre sıralanmıştır. Grupları oluştururken 0,90’dan küçükler *az sıkı* grubuna, 0,90 ve 0,93 kapalı aralığında olanlar *orta sıklıktaki* gruba ve 0,94’den büyük eşit olanlar ise *sıkı* gruba dâhil edilmiş ve bu gruplar sırasıyla çerçeve içerisine alınmıştır.

Çizelge 6.15. % Sıklık ve Boyut Artışına Bağlı Analiz Sonuçları

#	Ad	% sıklık	% $U_1$ % $U_2$		PSO			GA			YAA		
			$R$	$S$	$t(sn)$	$R$	$S$	$t(sn)$	$R$	$S$	$t(sn)$		
28	P-n16-k8	0,88	0,00	2,97	19,70	8,08	178,60	23,92	10,70	180,00	16,82	7,37	780,80
11	B-n31-k5	0,82	1,38	3,48	57,25	23,60	328,40	53,49	22,35	321,00	10,97	4,84	547,80
1	A-n32-k5	0,82	0,33	5,42	132,54	52,49	299,20	136,91	53,85	314,00	9,34	4,18	578,00
2	A-n33-k5	0,89	7,10	28,88	223,56	94,85	329,00	234,93	96,71	318,20	16,14	6,01	943,00
3	A-n37-k5	0,81	4,57	15,53	375,40	155,12	530,00	532,56	203,25	507,60	463,18	170,89	572,80
4	A-n39-k6	0,88	0,15	25,34	254,53	97,23	413,80	583,44	240,55	375,00	42,02	17,35	749,20
13	B-n43-k6	0,87	2,77	5,30	36,49	16,94	454,00	116,79	47,00	446,80	27,02	11,12	848,20
5	A-n48-k7	0,89	1,33	16,79	301,47	120,91	537,20	617,75	225,33	579,80	277,44	138,72	904,20
32	P-n55-k7	0,88	12,89	25,94	264,06	114,35	598,00	325,66	119,82	620,00	104,30	42,82	800,40
16	B-n56-k7	0,88	5,09	9,22	69,70	30,94	546,80	190,33	69,95	563,40	98,60	37,35	1054,00
17	B-n57-k9	0,89	0,16	8,62	56,12	22,24	618,40	99,54	40,94	606,40	92,56	41,73	1313,60
22	E-n76-k7	0,89	17,72	32,56	299,90	116,30	798,60	239,54	96,09	780,40	327,39	120,75	820,40
38	bays-n29-k5	0,93	4,02	6,60	1562,32	665,24	380,60	6402,69	2446,87	411,40	598,84	328,00	550,00
12	B-n34-k5	0,91	0,57	2,94	115,23	48,72	356,40	182,31	77,58	349,40	134,52	59,98	921,20
39	dantzig-n42-k4	0,93	3,06	32,64	429,89	167,06	565,40	256,03	109,64	550,40	179,77	78,95	465,20
14	B-n44-k7	0,92	0,28	6,17	463,69	196,26	428,80	273,48	113,24	456,60	30,36	11,68	890,40
30	P-n45-k5	0,92	6,70	24,29	292,33	117,95	467,20	194,51	78,37	455,60	254,42	102,95	1058,00
15	B-n50-k8	0,92	3,18	15,42	154,91	59,92	708,80	146,44	56,89	683,00	34,24	14,13	768,80
7	A-n62-k8	0,92	3,95	29,83	247,74	105,83	804,20	414,30	176,46	782,80	161,68	58,02	1411,60
18	B-n68-k9	0,93	3,53	13,55	339,80	127,34	637,40	250,68	115,45	684,60	205,36	76,16	1234,20
37	P-n101-k4	0,91	5,17	11,85	558,80	221,46	1191,80	509,34	191,15	1086,00	88,50	35,26	663,80
25	M-n101-k10	0,91	2,41	20,86	250,67	100,53	1471,80	75,86	28,39	1405,40	243,59	110,47	935,80
26	M-n151-k12	0,93	8,42	23,68	306,58	111,17	2244,80	246,55	89,58	2200,40	494,00	200,22	927,40
23	E-n101-k14	0,93	7,43	29,97	865,77	353,71	1123,20	94,03	44,79	1140,20	462,69	176,80	1005,60
29	P-n20-k2	0,97	2,87	4,52	172,45	76,25	206,00	97,59	37,95	203,00	117,07	51,27	780,00
20	E-n22-k4	0,94	4,94	7,94	163,95	65,34	227,80	97,51	39,82	227,80	14,04	6,52	711,40
21	E-n30-k3	0,94	4,02	6,34	111,27	42,64	328,80	216,26	103,68	329,40	147,81	63,86	751,00
31	P-n50-k10	0,95	11,98	40,06	139,43	55,52	563,40	336,93	128,28	550,60	149,05	54,69	1261,20
6	A-n53-k7	0,95	2,12	26,00	351,50	143,90	463,80	255,16	113,27	438,40	385,68	169,17	749,60
33	P-n60-k10	0,95	8,56	46,37	438,10	182,22	744,60	218,55	89,12	745,80	39,22	14,92	830,80
8	A-n63-k9	0,97	7,96	49,13	986,23	409,02	503,40	1456,54	578,59	551,80	133,58	55,87	910,00
34	P-n65-k10	0,94	32,61	69,40	294,57	104,79	766,80	85,51	35,70	784,00	405,38	166,68	616,60
9	A-n69-k9	0,94	0,44	25,36	577,36	216,82	629,20	339,09	130,12	600,40	695,64	307,14	1198,80
35	P-n70-k10	0,97	9,86	62,92	1393,36	546,81	750,40	1267,85	461,98	804,80	358,43	154,10	755,40
36	P-n76-k5	0,97	8,09	24,63	394,82	176,67	922,20	281,39	118,78	892,60	411,84	154,30	797,40
19	B-n78-k10	0,94	2,63	27,29	258,89	95,89	821,60	414,41	170,74	841,40	264,58	98,45	839,00
10	A-n80-k10	0,94	4,26	15,84	120,36	45,52	959,40	350,76	152,16	910,60	199,57	78,91	1008,80
24	F-n135-k7	0,95	1,96	14,28	340,76	130,75	1726,00	812,12	322,74	1682,20	680,90	277,04	697,40
27	M-n200-k17	0,94	1,67	19,17	920,88	358,10	3058,40	257,80	99,24	2831,20	140,44	60,20	1384,20

Çizelge 6.15'deki sütunlar sırasıyla problem numarasını (#), problem ismini (Ad), % sıklık değerini (% sıklık), %  $U_1$  ve %  $U_2$  değerlerini (%  $U_1$ , %  $U_2$ ) ve PSO,

GA ve YAA sırasında amaç fonksiyonuna ilişkin aralık ( $R$ ), standart sapma ( $S$ ) ve ortalama çözüm sürelerini ( $t(sn)$ ) göstermektedir.

Çizelgeye bakıldığında çözüm süreleri açısından anlamlı bir yorumun yapılabileceği görülmektedir. Her üç grup için de problem boyutları arttıkça PSO ve GA ile elde edilen çözüm sürelerinin arttığı gözlenmektedir. Öte yandan YAA ile elde edilen sonuçlarda problem boyutu ile çözüm süresi arasında bu tür bir ilişki görülememektedir. Benzer şekilde  $R$  ve  $S$  değerleri ile % sıklık ve problem boyutu arasında da bir ilişki kurulamayacağı görülmektedir. Bu sonuçlarla PSO ve GA ile elde edilecek çözüm sürelerinin % sıklık ve problem boyutuyla aralarında doğrusal yönde ilişki olduğu söylenebilir. YAA için böyle bir gözlem yapılamamaktadır.

Sonuç olarak 39 farklı test problemi üzerinde yapılan deneyler ve elde edilen değerlerin karşılaştırılmasından sonra BAARP için YAA'nın, PSO ve GA'ya göre bariz bir şekilde daha iyi sonuçlar verdiği görülmüştür. Bu nedenle YAA'nın BAARP için kullanılması gereken yaklaşım olduğu sonucu ortaya çıkmaktadır. Fakat YAA'nın eniyi çözüme ne kadar yakın sonuçlar verdiği bilinmemektedir. Bunun nedeni kullanılan test problemleri için BAARP'a ait eniyi çözümlerin bilinmiyor olması ve üstelik problemin NP-zor olması nedeniyle matematiksel model kullanılarak da eniyi çözümün kolayca belirlenemeyecek olmasıdır. Bir başka çalışmada yüksek kapasiteli donanım özellikleri olan bilgisayar ve/veya paralel hesaplama (paralel computing) gibi teknikler kullanılarak matematiksel model yardımıyla problemlerin eniyi çözümleri belirlenebilir. Bu sayede YAA'nın da problem için ne kadar kaliteli çözümler ürettiği de gözlenebilir. Bunun dışında yapılabilecek bir başka şey ise özellikle YAA için değişik parametreleri denemek ve her bir problem için deney sayısını arttırmak olabilir. Özellikle deney sayısının yeterli miktarda arttırılmasıyla elde edilecek amaç fonksiyonu değeri için güven aralığı hesaplanabilir. Son olarak problemler için güvenilir alt sınır üreten bir yöntemin geliştirilmesi elde edilen sonucun kalitesini belirlemek açısından yararlı olabilir. Ancak böyle bir yöntemin geliştirilmesi de son derece zordur ve tamamen başka bir çalışmanın konusudur.



## 7. SONUÇ VE ÖNERİLER

Bu çalışmada BAARP için PSO, GA ve YAA tabanlı yaklaşımlar geliştirilmiş ve yöntemlerin performansları, literatürden alınmış 39 test problemi üzerinde karşılaştırılmıştır. Ele alınan BAARP için eniyi çözüm değerlerinin literatürde de bilinmiyor olması sebebiyle karşılaştırmalar, bu algoritmalarından elde edilen eniyi ve ortalama amaç fonksiyonu değerleri, bu değerlerin aralığı ve standart sapmaları ile algoritmaların yakınsama eğilimleri bakımından yapılmıştır.

PSO için kesikli çözüm uzaylarında kullanılabilecek bir yaklaşım literatürdeki benzer çalışmalardan esinlenilerek geliştirilmiş ve uygulanmıştır. Bu yaklaşıma göre GA operatörlerinden çaprazlama operatörü bir skalaya göre belirli oranlarda PSO işleyişinde kullanılmıştır. Aynı parametreler altında çalıştırılan PSO'nun ve GA'nın ürettiği sonuçlar karşılaştırıldığında ise, PSO işleyişinde adapte edilen bu çaprazlama skalasının, amaç fonksiyonu değerlerine olumlu yönde katkısı olduğu, çözümde PSO'nun GA'dan bariz bir şekilde daha iyi sonuçlar ürettiği ve elde edilen çözüme çok daha kararlı bir şekilde yakınsadığı görülmüştür.

Bununla birlikte BAARP'de, YAA'nın hem GA hem de PSO'ya göre daha üstün olduğu gözlenmiştir. Ancak aynı sayıda ardıştırma yapıldığında çözüm süresinin YAA aleyhine büyük artış gösterebileceği de göz ardı edilmemelidir. Bu nedenle çözüm kalitesinden bir miktar ödün vermeyi göze alarak, YAA kadar iyi çözümler üretebilen, PSO'nun da BAARP'in çözümü için kullanılabilecek bir yaklaşım olduğu söylenebilir.

Her üç algoritma da farklı sayılarda parametrelerden oluşmaktadır ve bu parametrelere atanacak değerler algoritmaların üreteceği sonuçlar ile yakından ilgilidir. Bu parametrelerin uygun değerlerinin araştırılması bir başka çalışmanın konusu olabilir. Ayrıca önerilen YAA'nın bu problem için ne düzeyde iyi çözümler bulabilen bir yöntem olduğu da araştırılabilir. Bunun için ya BAARP'a ait gerçekten kaliteli alt sınırlar üretecek bir yöntem geliştirilmeli ya da test problemlerinin eniyi çözümlerinin elde edilmesini sağlayacak bir çözüm araştırılmalıdır. Bunların her ikisi de ayrı birer çalışma yapmayı gerektirir. Kaliteli alt sınır üretecek bir yöntem geliştirmek kolay

olamayabilir. Problemin eniyi çözümünü bulmak ise NP-zor yapı nedeniyle yüksek kapasiteli bilgisayar kullanmayı ya da paralel hesaplama gibi yöntemlerin tasarlanarak kullanılmasını gerektirmektedir. Bunların dışında yapılabilecek bir çalışma ise YAA'daki parametre değişikliklerinin çözüm üzerindeki etkilerini araştırmak olabilir. Son olarak yeterli sayıda deney yapılarak elde edilen çözüm sonuçlarının istatistiksel açıdan değerlendirilmesi ve amaç fonksiyonu için bir güven aralığının belirlenmesi de ayrı bir çalışma olarak yapılabilir.

## KAYNAKLAR DİZİNİ

- Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D. and Rinaldi, G., 1995, Computational results with a branch and cut code for the capacitated vehicle routing problem, Research report 949-M. Grenoble, France: Universite Joseph Fourier.
- Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., 1993, Network flows: theory and algorithms and applications, Printice Hall, New Jersey, 864p.
- Akçelik, R., 1982, Progress in fuel consumption modelling for urban traffic management, Australian Road Research Board Report 124, Canberra.
- Angeline, P., 1998, Using selection to improve particle swarm optimization, IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA, May 4-9.
- Anghinolfi, D. and Paolucci, M., 2009, A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times, European Journal of Operational Research, 193, 73–85.
- Arkin E.M, Hassin, R. and Levin, A., 2006, Approximations for minimum and min-max vehicle routing problems, Journal of Algorithms, 59, 1, 1–18.
- Barth, M., An, F., Younglove, T., Scora, G., Levine, C., Ross, M. and Ventel, T., 2000, Comprehensive modal emissin (CMEM), Version 2.0 User Guide. Technical Report.
- Barth, M. and Boriboonsomsin, K., 2008, Real world CO<sub>2</sub> impacts of traffic congestion, Transportation Research Record 2058, 63-71.
- Bektaş T. and Laporte, G., 2011, Pollution routing problem, Transportation Research Part B, 45, 8, 1232-1250.
- Bowyer, D.P., Akçelik, R. and Biggs, D.C., 1985, Guide to fuel consumption analysis for urban traffic management, Australian Road Research Board Transport Research 32, 21, 6, 485p.
- Campbell, AM, Vandenbussche, D. and Hermann, W., 2008, Routing for relief efforts, Transportation Science, 42, 2, 127–45.
- Christofides, N. and Eilon, S., 1969, An algorithm for the vehicle dispatching problem, Operational Research Quarterly, 20, 3, 309-318.

**KAYNAKLAR DİZİNİ (devam)**

- Christofides, N., Mingozzi, A. and Toth, P., 1979, The vehicle routing problem, Combinatorial optimization, Wiley, New York, pp. 425p.
- Clarke, G. and Wright, J.W., 1964, Scheduling of vehicles from a central depot to a number of delivery points, Operations Research, 12, 568-581.
- Clerc, M. and Kennedy, J., 2002. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, IEEE Transactions on Evolutionary Computations 6, 58-73.
- Dantzig, G.B. and Ramser, J.H., 1959, The truck dispatching problem, Management Science, 6, 1, 80-91.
- Demir, E., Bektaş, T. and Laporte, G., 2011, A comparative analysis of several vehicle emission models for road transportation, Transportation Research Part D, 16, 347-357.
- Dijkstra, E.W., 1959, A note on two problems in connection with graphs, Numerische Mathematik, 1, 269-271.
- Dorigo, M., Maniezzo, V. and Coloni, A., 1991, Positive feedback as a search strategy, Dipartimento di Elettronica, Politecnico di Milano, Italy, Technical Report.
- Dorigo, M., Maniezzo, V. and Coloni, A., 1996, The ant systems: optimization by a colony of cooperative agents, IEEE Trans Man Mach Cybernet-Part B, 26, 1-13.
- Eberhart, R.C. and Kennedy, J., 1995, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43. Piscataway, NJ: IEEE Service Center.
- Eberhart, R.C., Shi, Y. and Kennedy, J., 2001, Swarm Intelligence, San Francisco: Morgan Kaufmann Publishers, 512p.
- Eberhart, R.C., Simpson, P. K. and Dobbins, R. W., 1996, Computational Intelligence PC Tools. Boston, MA: Academic Press Professional.
- Edmonds, J. and Johnson, E.L., 1973, Matching, euler tours and the chinese postman, mathematical programming, 5, 1, 88-124.

**KAYNAKLAR DİZİNİ (devam)**

- Eiselt, H.A., 2000, Arc routing: theory, solutions and applications, Boston, Kluwer Academic Publishers, 512p.
- Eiselt, H.A., Gendreau, M. and Laporte, G., 1995, Arc routing problems, part I: the chinese postman problem, *Operations Research*, 43, 2, 231-242 (a).
- Eiselt, H.A., Gendreau, M. and Laporte, G., 1995, Arc routing problems, part II: the rural postman problem, *Operations Research*, 43, 3, 399-414 (b).
- Euler, L., 1736, *Solutio problematis ad geometriam situs pertinentis*, *Comment Acad Sci Imperialis Petropolit*, 8, 128–140.
- Fang, L., Chen, P. and Liu S., 2007, Particle swarm optimization with simulated annealing for TSP, *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, Greece, February 16-19.
- Fischetti, M., Laporte, G. and Martello, S., 1993, The delivery man problem and cumulative matroids, *Operations Research*, 41, 1055–64.
- Fisher, M.L., 1994, Optimal solution of vehicle routing problems using minimum k-trees, *Operations Research*, 42, 626-642.
- Gen, M. and Cheng, R., 1997, *Genetic algorithms and engineering design*, A Wiley-Interscience Publication John Wiley & Sons, Inc, 432p.
- Glover, F., 1989, Tabu search part I, *ORSA Journal on Computing*, 1, 190–206.
- Glover, F., 1990a. Tabu search part II, *ORSA Journal on Computing*, 2, 4–32.
- Glover, F., 1990b, Tabu search: a tutorial, *Programming Integer Algorithms, Heuristics*, 20, 4, 74-94.
- Goldberg, D.E., 1989, *Genetic Algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Massachusetts, 432p.
- Gribkovskaia, I., Halskau, Sr., Q. and Laporte, G., 2007, The bridges of Königsberg-A historical perspective, *Networks*, 49, 3, 1707-1783.
- Gross, J.L. and Yellen, J., 2004, *Handbook of graph theory*, CRC Press, Boca Raton, 1115p.

### **KAYNAKLAR DİZİNİ (devam)**

- Gutin, G. and Punnen, A.P., (eds.), 2004, The travelling salesman problem and its variations, Kluwer Academic Publishers, Dordrecht, 38p.
- Hao, X., 2010, A fuel consumption objective of VRP and the genetic algorithm, IEEE, 1-4.
- Hickman, A.J., 1999, Methodology for calculating transport emissions and energy consumption, METT Project.
- Holland, J.H., 1975, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, Michigan, 211p.
- Huilian, F., 2010, Discrete particle swarm optimization for TSP based on neighborhood Journal of Computational Information Systems, 6, 10, 3407-3414.
- Kara I., Kara, B.,Y. and Yetiş, M.K., 2007, Energy minimizing vehicle routing problem, Combinatorial Optimization and Applications, Proceedings 1<sup>st</sup> International Conference on Combinatorial Optimization and Applications, Xian, PEOPLES R CHINA.
- Kara I., Kara, B.,Y. and Yetiş, M.K., 2008, Vehicle routing problem, Croatia, 125p.
- Karaboğa, D., 2005, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Computer Engineering Department.
- Karaboğa, D., 2010, Artificial bee colony algorithm, Scholarpedia, 5, 6915p.
- Karaboğa, D. and Akay, B., 2009, A comparative study of artificial bee colony algorithm, Applied Mathematics and computation, 214, 108-132.
- Karaboğa, D. and Baştürk, B., 2007, A powerful and efficient algorithm for numerical function optimization, Journal of Global Optimization, 39, 459-471 (a).
- Karaboğa, D. and Baştürk, B., 2007, Artificial bee colony (ABC) optimization algorithm for solving constraint optimization problem, Springer-Verlag Berlin (b).
- Karaboğa, D. and Baştürk, B., 2007, On the performance of artificial bee colony (ABC) algorithm, Elvister B.V. (c).
- Kaur, A. and Bhatt, D., 2011, Particle swarm optimization with cross-over operator for prioritization in regression testing, International Journal of Computer Applications, 27, 10, 27-34.

### KAYNAKLAR DİZİNİ (devam)

- Kennedy, J., 1997, The particle swarm: social adaptation of knowledge, Proc. Intl. Conf. on Evolutionary Computation, Indianapolis, IN, 303-308. Piscataway, NJ: IEEE Service Center.
- Kennedy, J. and Eberhart, R.C., 1995, Particle swarm optimization. Proc. IEEE Int'l. Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center.
- Kennedy, J. and Eberhart, R.C., 1997, A discrete binary version of the particle swarm algorithm. Proc. 1997 Conf. on Systems, Man, and Cybernetics, 4104–4109. Piscataway, NJ: IEEE Service Center.
- Kirkpatrick, S., Gelatt, C.D. and Jr., Vecchi, M.P., 1983, Optimization by simulated annealing, Science, 220, 671-680.
- Kuo, Y., 2010, Using simulated annealing to minimize fuel consumption for time-dependent vehicle routing problem, Computers & Industrial Engineering, 59, 157-165.
- Larsen, J. and Clausen, J., 2009, Supplementary notes to networks and integer programming, Kongens Lyngby, 193p.
- Lenstra, J.K. and Rinnooy, Kan, A.H.G., 1976, On general routing problems, Networks, 6, 3, 273-280.
- Lian, Z., Gu, X. and Jiao, B., 2006, A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan, Applied Mathematics and Computation, 175, 73–785.
- Lucena, A., 1990, Time-dependent traveling salesman problem: the deliveryman case, Networks, 20, 753–63.
- Marinakis, Y. and Marinaki, M., 2010, A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem, Expert Systems with Applications, 37, 1446–1455.
- Marinakis, Y., Marinaki, M. and Dounias, G., 2010, A hybrid particle swarm optimization algorithm for the vehicle routing problem, Engineering Applications of Artificial Intelligence, 23, 463–472.
- Niu, Q., Jiao, B. and Gu, X., 2008, Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time, Applied Mathematics and Computation, 205, 148–158.

**KAYNAKLAR DİZİNİ (devam)**

- Ngueveu, S.U., Prins, C. and Calvo, R.W., 2010, An effective algorithm for the cumulative vehicle routing problem, *Computers & Operations Research*, 37, 11, 1877-1885.
- Ntziachristos, L. and Samaras, Z., 2000, COPERT III Computer Programme to Calculate Emissions from Road Transport: Methodology and Emission Factors (Version 2.1). Technical Report. European Environment Agency, Copenhag, Denmark.
- Orloff, C.S., 1974, A fundamental problem in vehicle routing problem, *Networks*, 4, 35-64.
- Pan, Q.K., Tasgetiren, M.F. and Liang, Y.C., 2008, A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, *Computers & Operations Research*, 35, 2807 – 2839.
- Pant, M., Thangaraj, R. and Abraham, A., 2007, A new PSO algorithm with crossover operator for global optimization problems, *Innovations in Hybrid Intelligent Systems*, Springer-Verlag Berlin Heidelberg, ASC 44, 215–222.
- Pant, M., Thangaraj, R. and Singh, V.P., 2009, Particle swarm optimization with crossover operator and its engineering applications, *IAENG International Journal of Computer Science*, 36,2.
- Park, J.B., Jeong, Y.W., Shin, J.R., Lee, K.Y. and Kim, J.H., 2007, A hybrid particle swarm optimization employing crossover operation for economic dispatch problems with valve-point effects, November 4 - 8, Kaohsiung, Taiwan.
- Papadimitriou, C.H., 1976, On the complexity of edge traversing, *Journal of the Association for Computing Machinery*, 23, 3, 544-554.
- Pearl, J., 1984, *Heuristics: intelligent search strategies for computer problem solving*, Addison-Wesley Pub., 399p.
- Pham, D.T., Koç, E., Ghanbarzadeh, A., Otri, S., Rahim, S. and Zaidi, M., 2006, The bees algorithm – a novel tool for complex optimisation problems, in: *proceedings of the Second International Virtual Conference on intelligent production machines and systems*, 454–461.
- Pinedo, M.L., 2008, *Scheduling theory, algorithms, and systems* third edition, Springer, Prentice Hall, 671p.
- Premalatha, K. and Natarajan, A.M., 2009, Discrete PSO with GA operators for document clustering, *International Journal of Recent Trends in Engineering*, 1, 1, 20-24.



### KAYNAKLAR DİZİNİ (devam)

- Prins, C., 2004, A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers and Operations Research*, 31, 12, 1985-2002.
- Qin, J., Li, X. and Yin, Y., An algorithmic framework of discrete particle swarm optimization, *Applied Soft Computing*, in press.
- Rees, W.G., 2004, Least-cost paths in mountainous terrain, *Computers & Geosciences*, 30, 203-209.
- Ribeiro, G.M. and Laporte, G., 2012, An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem, *Computers & Operations Research*, 39, 728-735.
- Salehipour, A., Sörensen, K., Goos, P. and Braysy, O., 2008, An efficient GRASP + VND metaheuristic for the traveling repairman problem, Working paper, University of Antwerp, Faculty of Applied Economics.
- Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C. and Wang, Q.X., 2007, Particle swarm optimization-based algorithms for TSP and generalized TSP, *Information Processing Letters*, 103, 169–176.
- Su, D., Xu, W. and Sun, J., 2009, Quantum-behaved particle swarm optimization with crossover operator, *International Conference on Wireless Networks and Information Systems*, DOI 10.1109/WNIS.2009.74, 399-402.
- Tang, J., Zahng, J. and Pan, Z., 2010, A scatter search algorithm for solving vehicle routing problem with loading cost, *Expert Systems With Applications*, 37, 4073-4083.
- Tavares, G., Zsigraiova, Z., Semiao, V. and Carvalho, M.G., 2009, Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling, *Waste Management*, 29, 1176-1185.
- Toth, P. and Vigo, D. (eds.), 2001, *The vehicle routing problem (Monographs on Discrete Mathematics and Applications)*, SIAM, 367p.
- Tsang, K.S., Hung, W.T. and Cheung, C.S., 2011, Emissions and fuel consumption of a Euro 4 car operating along different routes in Hong Kong, *Transportation Research Part D*, 16, 415-422.
- Tseng, C.T. and Liao, C.J., 2008, A discrete particle swarm optimization for lot-streaming flowshop scheduling problem, *European Journal of Operational Research*, 191, 360–373.
- Yong, P. and Xiaofeng, W., 2009, Research on a vehicle routing schedule to reduce fuel consumption, *International Conference on Measuring Technology and Mechatronics Automation, China*, 825-827.

### KAYNAKLAR DİZİNİ (devam)

Zhang, G., Dou, M. and Wang, S., 2009, Hybrid genetic algorithm with particle swarm optimization technique, International Conference on Computational Intelligence and Security, DOI 10.1109/CIS.2009.236, 103-106.

### İNTERNET KAYNAKLARI

**Şekil 2.4:** Düzgün on iki yüzlü ve Karşı gelen uygun bir Hamilton turu, [http://en.wikipedia.org/wiki/Icosian\\_game](http://en.wikipedia.org/wiki/Icosian_game) ve [http://en.wikipedia.org/wiki/Icosian\\_Calculus](http://en.wikipedia.org/wiki/Icosian_Calculus), Son erişim tarihi: 30.11.2011

**Şekil 2.5:** Köningsberg Köprüleri-1736, Köningsberg Köprüleri'nin serim gösterimi, <http://www.transtutors.com/homework-help/Discrete+Mathematics/Graph+Theory/konisberg-multigraph-bridge.aspx>, Son erişim tarihi: 30.11.2011

**Şekil 4.4:** Yasaklı Arama Algoritmasındaki hafıza tipleri, <https://www-edc.eng.cam.ac.uk/projects/tabusearch/>, Son erişim tarihi: 30.11.2011

**Şekil 4.6:** Bir karınca kolonisinin besine erişimi ile ilgili bir gösterim, <http://www.ankurkhetrapal.com/research/proteus.htm>, Son erişim tarihi: 30.11.2011

**Şekil 4.7:** Arı kolonisinin ile ilgili bir gösterim, [http://www.emo.org.tr/ekler/5a2d64e6d533a97\\_ek.pdf](http://www.emo.org.tr/ekler/5a2d64e6d533a97_ek.pdf), Son erişim tarihi: 30.11.2011

**Şekil 4.8:** Çeşitli hiperküpler, <http://en.wikipedia.org/wiki/Hypercube>, Son erişim tarihi: 30.11.2011.

## **EKLER DİZİNİ**

Ek. 1. Parçacık Sürüsü Optimizasyonu İle Elde Edilen Çözümler

Ek. 2. Genetik Algoritma İle Elde Edilen Çözümler

Ek. 3. Yasaklı Arama Algoritması İle Elde Edilen Çözümler

Ek. 4. Rassal Türetilen Probleme Ait Bilgiler

















## Ek. 2. Genetik Algoritma İle Elde Edilen Çözümler

<i>Problem İsmi</i>	<i>Araçlar</i>	<i>Tur</i>										
A-n37-k5	1. araç:	0	22	8	5	34	6	14	23			
	2. araç:	0	2	18	15	24	21	20	3	7		
	3. araç:	0	35	37	33	29	32	27	31	30	11	
	4. araç:	0	17	16	4	36	19	26	9			
	5. araç:	0	13	10	25	12	28					
A-n39-k6	1. araç:	0	16	14	36							
	2. araç:	0	31	33	19	23	35	11	17	15		
	3. araç:	0	21	20	34	3	9	8	5			
	4. araç:	0	25	39	32	38	13	29	26			
	5. araç:	0	4	27	6	30	10	12				
	6. araç:	0	7	2	37	18	24	22	28			
A-n48-k7	1. araç:	0	15	18	19	13	7					
	2. araç:	0	45	24	44	32	6	2	23	33		
	3. araç:	0	37	8	41	46	35	10	25	31	22	30
	4. araç:	0	29	11	3	48	42					
	5. araç:	0	17	34	47	5	43	12	14			
	6. araç:	0	28	16	9	40	27	21				
	7. araç:	0	36	39	26	20	38	4				
A-n53-k7	1. araç:	0	2	5	29	23	31	45	47	9	28	
	2. araç:	0	36	7	39	34						
	3. araç:	0	19	41	27	11	30	50	37	3	38	
	4. araç:	0	46	49	43	24	44	51	20	16	42	
	5. araç:	0	12	53	14	33	17	13				
	6. araç:	0	8	18	25	35	22	15				
	7. araç:	0	4	6	40	26	10	48	52	32	21	
A-n62-k8	1. araç:	0	29	16	8	52	11	62	30	9	35	
	2. araç:	0	44	33	7	41	58	32				
	3. araç:	0	60	56	27	34	59	18	51	39		
	4. araç:	0	21	13	54	57	36	50	12	26		
	5. araç:	0	49	15	19	31	61	6	20	14	2	
	6. araç:	0	55	10	43	4	25	48				
	7. araç:	0	45	46	47	3	23	38				
	8. araç:	0	17	37	53	40	28	24	42	22	5	



## Ek. 2. (devam) Genetik Algoritma İle Elde Edilen Çözümler

<u>Problem İsmi</u>	<u>Araçlar</u>	<u>Tur</u>	<u>Tur</u>									
B-n34-k5	1. araç:	0	6	7	26	18	32	16	29	14		
	2. araç:	0	19	10	15	28						
	3. araç:	0	2	20	17	25	30	4	27	5		
	4. araç:	0	8	22	3	23	33	11	24	9	21	13
	5. araç:	0	12	31	34							
B-n43-k6	1. araç:	0	5	36	42	28	33	30	26	41	21	15
	2. araç:	0	34	22	6	19	20	29	38			
	3. araç:	0	9	3	24	2	39	11	8			
	4. araç:	0	18	40	35	17	16	32	13	37		
	5. araç:	0	10	27	43	7	25					
	6. araç:	0	31	12	4	14	23					
B-n44-k7	1. araç:	0	38	24								
	2. araç:	0	8	17	19	30	33	20	26			
	3. araç:	0	7	14	42	35	10	12	11	39		
	4. araç:	0	22	5	43	31	23	9				
	5. araç:	0	27	15	37	3	40	25	36			
	6. araç:	0	28	32	44	34	4					
	7. araç:	0	6	18	41	2	13	29	21	16		
B-n56-k7	1. araç:	0	28	17	53	2	38	40	36			
	2. araç:	0	49	37	14	50	34	4	33			
	3. araç:	0	26	54	7	25	42	16	20	29		
	4. araç:	0	46	30	43	48	27	22	5	8		
	5. araç:	0	19	47	11	41	15	45	9	18		
	6. araç:	0	44	52	6	13	24	23	51	39	10	
	7. araç:	0	3	31	35	55	56	32	21	12		
B-n57-k9	1. araç:	0	7	33	41	4	47	43				
	2. araç:	0	23	57	11	18	32	56	51			
	3. araç:	0	10	29	28	19	25	2				
	4. araç:	0	6	31	53	46	44	55				
	5. araç:	0	49	39	50	45	3	12				
	6. araç:	0	5	36	34	8	21	15	14			
	7. araç:	0	17	24	27	40	20					
	8. araç:	0	42	22	37	13	38	35				
	9. araç:	0	48	30	54	52	26	16	9			













### Ek. 3. Yasaklı Arama Algoritması İle Elde Edilen Çözümler

<i>Problem İsmi</i>	<i>Araçlar</i>	<i>Tur</i>
A-n37-k5	1. araç:	0 22 17 8 23 14 6 34 11
	2. araç:	0 31 27 19 36 26 9
	3. araç:	0 2 13 3 20 21 28
	4. araç:	0 35 37 33 29 32 30 5 7
	5. araç:	0 16 4 18 15 24 25 10 12
A-n48-k7	1. araç:	0 19 33 39 20 26 7
	2. araç:	0 24 44 32 13 6 2 23
	3. araç:	0 45 36 41 46 35 10 25 5 43 12
	4. araç:	0 15 42 3 11 34
	5. araç:	0 48 17 29 30 22 31 14 47
	6. araç:	0 18 28 16 9 27 40
	7. araç:	0 37 8 38 4 21
A-n53-k7	1. araç:	0 2 5 48 8 13 17 49 38
	2. araç:	0 32 7 21 11 27
	3. araç:	0 28 36 39 19 41 30 3 37 51
	4. araç:	0 29 23 46 43 44 24 20 33 16
	5. araç:	0 10 18 42 25 12 35 53
	6. araç:	0 4 6 40 26 15 22 14
	7. araç:	0 52 9 47 45 31 50 34
A-n62-k8	1. araç:	0 8 21 62 7 33 58 51 59
	2. araç:	0 52 11 9 44 30 41
	3. araç:	0 57 32 35 56 60 27 34 18
	4. araç:	0 16 54 13 15 6 61 26 12 50 36 5 22
	5. araç:	0 29 39 20 31 2 10 25
	6. araç:	0 49 14 19 43 40 28
	7. araç:	0 17 38 23 45 46 48
	8. araç:	0 37 53 3 47 55 4 24 42
A-n63-k9	1. araç:	0 27 20 26 14 42 44 47 2
	2. araç:	0 21 35 50 9 60 25
	3. araç:	0 55 30 31 3 6 46 28
	4. araç:	0 17 62 43 10 39 32 19
	5. araç:	0 13 29 18 38 36 53
	6. araç:	0 57 16 11 40 24 63 48 5
	7. araç:	0 12 34 22 56 33 41 15
	8. araç:	0 45 54 23 51 52 58
	9. araç:	0 8 61 37 49 4 7 59





### Ek. 3. (devam) Yasaklı Arama Algoritması İle Elde Edilen Çözümler

<i>Problem İsmi</i>	<i>Araçlar</i>	<i>Tur</i>													
B-n78-k10	1. araç:	0	18	73	59	22	44	68	39	70	67	61	42	37	
	2. araç:	0	28	48	64	6	71								
	3. araç:	0	3	50	31	38	58	33	4	55					
	4. araç:	0	66	76	30	8	77	74	21						
	5. araç:	0	9	72	23	16	13	35	52	32					
	6. araç:	0	2	25	53	57	46	47							
	7. araç:	0	43	17	19	26	11	69	60						
	8. araç:	0	78	24	45	27	12	65	41	54	15				
	9. araç:	0	7	40	49	5	14	36							
	10. araç:	0	63	51	62	56	34	29	75	20	10				
E-n22-k4	1. araç:	0	17	15	16	19	21								
	2. araç:	0	13	11	9	7	4	5	12						
	3. araç:	0	10	8	6	3	2								
	4. araç:	0	14	20	22	18									
E-n30-k3	1. araç:	0	19	24	11	13	9	15	10	18	8	14	17	16	22
	2. araç:	0	20	27	28	30	26	25	2	7	12				
	3. araç:	0	21	23	4	5	6	3	29						
E-n76-k7	1. araç:	0	76	69	7	52	4	45	33	51	26	56	67		
	2. araç:	0	47	35	9	8	20	36	54	12	60	15			
	3. araç:	0	5	31	75	29	22	48	49	16	58	14	55		
	4. araç:	0	27	13	41	10	40	73	59	39	11	32	66		
	5. araç:	0	68	53	28	46	30	6	38	21	71	61	72	37	
	6. araç:	0	3	63	74	34	2	44	42	43	65	23	62	70	
	7. araç:	0	18	17	64	24	57	50	25	19					

### Ek. 3. (devam) Yasaklı Arama Algoritması İle Elde Edilen Çözümler

<i>Problem İsmi</i>	<i>Araçlar</i>	<i>Tur</i>															
E-n101-k14	1. araç:	0	54	59	41	22	74	73	68								
	2. araç:	0	75	42	23	76	24	40									
	3. araç:	0	27	13	69	81	55	56	26								
	4. araç:	0	29	51	34	82	10	67									
	5. araç:	0	53	89	63	12	20	50	65								
	6. araç:	0	19	83	48	49	8										
	7. araç:	0	32	11	91	33	64	21	35								
	8. araç:	0	90	61	85	6	84	46	9	47	37						
	9. araç:	0	95	96	98	88	38	45									
	10. araç:	0	97	60	101	99	92	15	44	39							
	11. araç:	0	7	100	94	93	86	62	18								
	12. araç:	0	14	3	58	16	43	17	87								
	13. araç:	0	77	78	4	80	79	30	25	5	57						
	14. araç:	0	28	70	2	71	31	52	72	66	36						
M-n101-k10	1. araç:	0	21	22	23	24	27	29	31	28	25	34	36				
	2. araç:	0	26	30	35	37	40	39	38	32	33						
	3. araç:	0	44	48	50	53	51	52	49	46	47	45	61	59	57		
	4. araç:	0	43	42	41	60	58	56	55	54							
	5. araç:	0	64	66	68	67	70	69	65	63	75	73	62				
	6. araç:	0	91	90	88	87	89	86	85	84	83	71					
	7. araç:	0	76	99	97	96	95	94	98	93	101						
	8. araç:	0	6	4	5	8	11	12	10	9	7	3	2	92	100		
	9. araç:	0	14	18	19	20	16	17	13	15							
	10. araç:	0	82	79	77	78	80	81	74	72							
M-n151-k12	1. araç:	0	54	106	27	150	13	77	2	133	70	123	52	104	31		
	2. araç:	0	112	51	103	34	82	10	121	79	80	122	30	25			
	3. araç:	0	29	117	78	4	130	35	136	36	137	72	129	21	67	66	
	4. araç:	0	28	102	71	11	109	91	12	108	20	48	125	50	37		
	5. araç:	0	113	147	53	128	32	89	149	63	127	64	33	132	65		
	6. araç:	0	95	96	118	98	38	101	45	141	39	87	94	62			
	7. araç:	0	7	97	105	100	60	93	99	86	92	17	142	120	143	16	44
	8. araç:	0	14	138	88	145	58	146	42	23	134	75	76	135			
	9. araç:	0	59	41	22	73	57	140	26	56	5	111	40	68	24		
	10. araç:	0	139	110	151	69	81	55	131	74	116	3	43	15			
	11. araç:	0	119	61	6	85	84	115	126	46	18	114	9	47			
	12. araç:	0	90	148	19	107	83	8	124	49	144						









#### Ek. 4. Rassal Türetilen Probleme Ait Bilgiler

Düğüm No	x	y	Talepler
1	17	11	0
2	12	16	16
3	10	7	17
4	7	8	16
5	15	15	12
6	13	5	11
7	2	7	16
8	13	1	13
9	11	14	19
10	17	5	13

**Toplam 133**

**Araç Sayısı 2**

**Araç Kapasitesi 75**

**Toplam Kapasite 150**

**% Sıkılık  $133/150=0.89$**

```
IDE No active process
cum_vrp2
2036700 119 infeasible 113.2407 113.1095 18488136 0.12
2036800 28 infeasible 113.2407 113.2097 18488231 0.03

GUB cover cuts applied: 1
Flow cuts applied: 19
Gomory fractional cuts applied: 3
Fixing integer variables, and solving final LP...
Tried aggregator 1 time.
LP Presolve eliminated 98 rows and 245 columns.
Reduced LP has 180 rows, 20 columns, and 360 nonzeros.
Presolve time = 0.00 sec.
Using devex.

Proven optimal solution.

MIP Solution: 113.240688 (18488246 iterations, 2036832 nodes)
Final Solve: 113.240688 (0 iterations)

Best possible: 113.240688
Absolute gap: 0.000000
Relative gap: 0.000000

--- Restarting execution
--- CUM_VRP2.gms(182) 0 Mb
--- Reading solution for model burcin
*** Status: Normal completion
```