

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ZAMAN PENCERELİ ARAÇ ROTALAMA PROBLEMİ'NİN GENETİK
ALGORİTMA İLE MODELLENMESİ**

**YÜKSEK LİSANS TEZİ
Pınar DURSUN**

Anabilim Dalı : Endüstri Mühendisliği

Programı : Endüstri Mühendisliği

TEMMUZ 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ZAMAN PENCERELİ ARAÇ ROTALAMA PROBLEMİ'NİN GENETİK
ALGORİTMA İLE MODELLENMESİ**

**YÜKSEK LİSANS TEZİ
Pınar DURSUN
(507061121)**

**Tezin Enstitüye Verildiği Tarih : 27 Temmuz 2009
Tezin Savunulduğu Tarih : 30 Temmuz 2009**

**Tez Danışmanı : Doç. Dr. Y. İlker TOPÇU (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. M. Nahit SERARSLAN (İTÜ)
Prof. Dr. Demet BAYRAKTAR (İTÜ)**

TEMMUZ 2009

Anneme,

ÖNSÖZ

Öncelikle, bu çalışma sırasında yardımlarını ve desteklerini benden esirgemeyen tüm sevdiklerime çok teşekkür ediyorum. Danışmanım ve sevgili Hocam Y. İlker Topçu'ya, konu ile ilgili çok değerli tavsiyeleri ve yönlendirmeleri için teşekkürü bir borç bilirim. Yardımlarını ve çalışmamı titizlikle inceleyip çok kıymetli fikirlerini benden esirgemediği için sevgili Hocam C. Erhan Bozdağ'a ve tüm Hocalarıma çok teşekkür ediyorum. Bu zorlu süreçte her anlamda yanımda olan çok değerli meslektaşlarım İTÜ Endüstri Mühendisliği Araştırma Görevlileri'ne ve tüm dostlarıma çok teşekkür ediyorum.

Portekiz'de bulunduğum sürede bana destek olan danışmanlarımdan öncelikle José António V Oliveira ve Guilherme Pereira'ya, ve tüm Minho Üniversitesi Üretim ve Sistem Mühendisliği Departmanı çalışanlarına teşekkür ederim.

Ve son olarak tüm eğitimim süresince yanımda olan aileme teşekkür ederim.

Temmuz 2009

Pınar Dursun

Araştırma Görevlisi

İÇİNDEKİLER

ÖNSÖZ.....	v
İÇİNDEKİLER.....	vii
KISALTMALAR.....	ix
ÇİZELGE LİSTESİ	xi
ŞEKİL LİSTESİ	xiii
ÖZET.....	xv
SUMMARY.....	xvii
1. GİRİŞ.....	1
2. ARAÇ ROTALAMA PROBLEMİ.....	3
2.1 Araç Rotalama Problemi'nin Tanımı	3
2.2 Araç Rotalama Problemi Çeşitleri.....	4
Kapasite Kısıtlı Araç Rotalama Problemi (Kapasite Kısıtlı ARP).....	5
2.2.2 Mesafe Kısıtlı Araç Rotalama Problemi (Mesafe Kısıtlı ARP)	7
2.2.3 Önce Dağıt Sonra Topla Araç Rotalama Problemi (Önce Dağıt Sonra Topla ARP)	8
2.2.4 Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi (Eş Zamanlı Topla-Dağıt ARP).....	8
2.2.5 Bölünmüş Dağıtımli Araç Rotalama Problemi (Bölünmüş Dağıtımli ARP)	8
2.2.6 Çok Depolu Araç Rotalama Problemi (Çok Depolu ARP).....	9
2.2.7 Periyodik Araç Rotalama Problemi (Periyodik ARP).....	9
2.2.8 Zaman Pencerele Araç Rotalama Problemi (Zaman Pencerele ARP).....	9
2.3 Araç Rotalama Problemleri İçin Çözüm Yöntemleri	10
2.3.1 Küme Örtüleme Yaklaşımı (Set-Covering Approach).....	10
2.3.2 Araç Akışına Dayanan Formülasyonlar (Vehicle Flow Based Formulation).....	12
2.3.2 Ürün Akışına Dayanan Formülasyonlar (Commodity Flow Based Formulation).....	13
3. SEZGİSEL YÖNTEMLER.....	15
3.1 Sezgisel Yöntemlere Giriş	15
3.2 Klasik Sezgisel Yöntemler	16
3.2.1 Yapısal Sezgisel Yöntemler	16
3.2.2 İki-Aşamalı Sezgisel Yöntemler	17
3.2.3 Geliştirici Sezgisel Yöntemler	17
3.3 Meta-Sezgisel Yöntemler	17
3.3.1 Tavlama Benzetimi (TB).....	18
3.3.2 Tabu Arama (TA)	20
3.3.3 Genetik Algoritmalar (GA)	21
4. GENETİK ALGORİTMA.....	29
4.1 Genetik Algoritma'nın Tarihçesi.....	29
4.2 Genetik Algoritma'daki Kavramlar.....	30
4.3 Genetik Algoritma Aşamaları.....	31
4.4 Genetik Algoritmalar'ın Kullanıldığı Problemler	40
4.5 Genetik Algoritma ve ARP Arasındaki İlişki.....	42

4.6 Rassal Sayı Kodlamalı Genetik Algoritma ve Kombinatorik En İyileme Problemleri ile Arasındaki İlişki	47
5. PROBLEM TANIMI	49
6. MODEL.....	51
6.1 Genetik Algoritma	51
6.2 Yardımcı Algoritma.....	55
6.3 Sayısal Sonuçlar	60
7. SONUÇLAR VE ÖNERİLER.....	63
KAYNAKLAR.....	65
EKLER	71

KISALTMALAR

ARP	: Araç Rotalama Problemi
GSP	: Gezgin Satıcı Problemi
KÖP	: Küme Örtüleme Problemleri
SA	: Süpürme Algoritması
TB	: Tavlama Benzetimi
TA	: Tabu Arama
GA	: Genetik Algoritma

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 : Meta-sezgisel yöntemlerin kullanıldığı çalışmalar.	27
Çizelge 4.1 : Standart ikili kodlama ve gri ikili kodlama.	32
Çizelge 4.2 : Genetik Algoritma ile yapılmış ARP çalışmaları.	46
Çizelge 6.1 : Yardımcı algoritma örnek verileri.	58
Çizelge 6.2 : Yardımcı algoritma örnek konum uzaklıkları.	58
Çizelge A.1 : c101 problem verisi.	727

ŞEKİL LİSTESİ

Sayfa

Şekil 4.1 : Allel-gen-kromozom ilişkisi.....	31
Şekil 6.1 : Genetik algoritma model akışı.....	51
Şekil 6.2 : Yardımcı algoritma akışı.....	56
Şekil 6.3 : Yardımcı algoritma örnek rotaları.....	59
Şekil 6.4 : 5 farklı koşumun iterasyon sayısı-toplam mesafe grafiği.....	60
Şekil 6.5 : Modelden elde edilen en iyi çözümün (deneme1) iterasyon sayısı-toplam mesafe grafiği.....	61

ZAMAN PENCERELİ ARAÇ ROTALAMA PROBLEMİ'NİN GENETİK ALGORİTMA İLE MODELLENMESİ

ÖZET

Küreselleşmenin rekabeti hızla arttırdığı son yıllarda müşteri memnuniyeti bu rekabetin en belirleyici faktörlerinden biri olmuştur. Tedarik zincirinde, ürünün ya da hizmetin müşteriye ulaştığı son aşama olan lojistik ve dağıtım şirketlerin üzerinde daha dikkatli durduğu bir alan haline gelmiştir. Ne kadar hızlı ve çok sayıda müşteriye ulaşılsa, o kadar müşteri memnuniyeti artacaktır. Fakat bunun yanısıra şirketler, kendi maliyetlerini de azaltmaya çalışmaktadır. Söz konusu rekabetin içerisindeki büyük lojistik ve dağıtım şirketlerinin en iyilemeye çalıştıkları problemler de büyük ve karmaşık olacaktır. Aynı anda hem müşterilerine hızlı ve kaliteli hizmet sağlamak hem de maliyetleri en aza indirmek bu şirketlerin hedefidir. Bu çalışmada, bu şirketlerin problemlerinden biri olan araç rotalama probleminin özel bir hali ele alınmıştır. Müşterilere belirli bir zaman aralığında hizmet verilebilen zaman pencereli araç rotalama probleminin çözümünde hızlı ve maliyetleri en aza indiren bir rotalama ortaya konulmaya çalışılır. Belirli bir dönem için bu rotaların çıkarılması gerektiği ve problem boyutunun büyük olması nedeniyle kesin en iyileme yöntemleri yavaş çözüm vereceklerinden, söz konusu yöntemleri kullanmak verimli olmayacaktır. Dolayısıyla en iyi çözümü garanti etmeyen fakat en iyi çözüme makul yakınlıkta çözüm elde edilebilen sezgisel yöntemler devreye girmektedir. Bu çalışmada meta-sezgisel bir yöntem olan genetik algoritma ile söz konusu problem modellenerek, genel bir uygulama verisi üzerinde test edilmiştir. Genetik algoritmada yeni bir birey kodlama yaklaşımı olan ve bu çalışmada araç rotalama problemi için kullanılan rassal sayılı kodlama, permütasyon kodlama gibi onarıcı işlemlere gerek duymadığından modelin daha etkin ve hızlı olmasını sağlayacaktır, bu nedenle birey kodlamada bu yöntemden faydalanılmıştır. Bu problemi ele alan diğer modellerle karşılaştırılabilir bir model ortaya koymak, bu çalışmanın amacıdır.

MODELING VEHICLE ROUTING PROBLEM WITH TIME WINDOWS WITH GENETIC ALGORITHM

SUMMARY

In recent years, globalization increases competition between companies rapidly, customer satisfaction has become one of the most significant factors of the competition. In supply chain, logistics and distribution, which is the last process that product or service reaches to customers, has become an area that companies pay more attention. Faster delivery and more customer reachability cause more customer satisfaction. But, companies should also consider and decrease their operational costs. The problems that large logistics and distribution companies in the competition try to overcome will be also large and complicated. The aim of these companies is to provide quality and speed service and minimizing the costs as well. In this study, one of the special forms of vehicle routing problem is taken into consideration. At the solution of the vehicle routing problem with time windows, which the customers can be served between a specific time windows only, a routing that minimizes the time and the costs is tried to be reached. Because of being necessary of getting the routes between a specific period and the size of problem is large, classical optimization methods are not enough because of being slow. Therefore, the heuristic methods that do not guarantee the optimal solution but can obtain a good solution, which is approximate enough to optimal solution, are used. In this study, the problem was modeled with genetic algorithm, which is one of the meta-heuristic methods and tested over a benchmark data. Random key representation is a new approach for individual coding in genetic algorithm and does not need repairment operations as permutational coding. Because of being effective and fast this method was used for coding. The aim of the study is to get a comparative model with other studies.

1. GİRİŞ

Küreselleşme ile birlikte gelişen rekabet ortamı, şirketleri sürekli daha iyi olmaya zorlamaktadır. Büyük şirketlerde karşılaşılan problemlerin boyutu da büyük olmakta ve problemin çözülme süreci daha karmaşık bir hal almaktadır. Bu noktada klasik en iyileme yöntemleri yetersiz kalacaktır, çünkü bu yöntemlerle sözkonusu problemlerin çözülmesi çok uzun süre gerektirecektir, halbuki rekabet etmenin gerekliliklerinden biri de hızlı olmaktır.

Bitmiş ürünün depolardan müşterilere dağıtımı lojistik yönetiminin önemli bir parçasıdır. Günümüzde hızla önem kazanmakta olan dağıtım sektöründe, lojistik şirketleri, kuryeler ve büyük çaplı şirketlerin kendi dağıtım departmanları, müşterilerine hızlı ve kaliteli hizmet vermek durumundadırlar. Daha iyi rotalama ve çizelgeleme kararları, kısa zamanda daha çok müşteriye hizmet edebilmeyi beraberinde getireceğinden daha fazla müşteri memnuniyeti sağlayacaktır. Fakat şirketler bir yandan da kendi performanslarını ve ekonomik çıkarlarını gözönünde bulundurmaları gerekmektedir. Literatürde sıkça yer alan Araç Rotalama Probleminin (ARP) çözülmesi ile sözkonusu şirketler gerçek hayatta dağıtım ağlarını en iyileme yoluna gideceklerdir.

Kombinatorik bir problem olan ve en iyi çözümün zor elde edilebildiği ARP'nin, sahip olduğu kısıtlara göre değişkenlik gösteren çok fazla çeşidi bulunmaktadır. Zaman Pencere ARB bunlardan biridir ve bu problemde müşteriler belirli zaman aralıklarında hizmet kabul edebilmektedir, ayrıca deponun açık kalabileceği bir son tarih mevcuttur. Klasik ARP gibi NP-zor olan bu problemin de büyük boyutlarının çözümü kesin yöntemlerle çok uzun sürede gerçekleşebilmektedir. Uzun sürede en iyi çözümü bulmak yerine, çok daha kısa bir sürede en iyi çözüme yakın ya da en iyi çözümü veren sezgisel yöntemlerden faydalanmak bu problem için uygun olacaktır. Böylece klasik en iyileme yöntemleri için gerekli olan süre gerekmeyecek, kabul edilebilir bir sapma ile en iyi çözüme yakın bir çözüm elde edilebilecektir. Sezgisel yöntemlerin bir sınıfına giren ve meta-sezgisel bir yöntem olan Genetik Algoritma (GA), doğadaki evrim sürecini bilgisayar ortamında taklit ederek bu tür problemler için en iyi çözümü ya da yakın çözümü elde edebilmektedir. Literatürde GA'nın

kullanıldığı ve ARP gibi diğer kombinatorik problemlerin çözümü için geliştirilmiş çok sayıda modelin olduğu çalışma bulunmaktadır. Bu çalışmada da, çözüm uzayının birden fazla noktasından arama yapmaya başlayan, hızlı ve etkili bir yöntem olan GA Zaman Pencereli ARP çözümü için kullanılmıştır.

GA'da bir çözümü temsil eden bireyin nasıl kodlandığı modelin etkinliği açısından önemlidir. Kombinatorik problemler için çoğunlukla permütasyon kodlama kullanılmaktadır, fakat sözkonusu kodlamada genetik operatörlerden biri olan çaprazlama sırasında sorun çıkabilmekte ve bunun giderilebilmesi için onarıcı işlemlere gerek duyulmaktadır. Yeni bir kodlama yaklaşımı olan rassal sayılı kodlamada bu tür onarıcı işlemlere gerek yoktur. Literatürden elde edilen kaynaklar arasında, herhangi bir ARP çözümü için yapılmış ve bu kodlamanın kullanıldığı bir çalışmaya rastlanamamıştır. Bu çalışmada klasik ARP'den daha zor bir problem olan Zaman Pencereli ARP'nin büyük boyutlarının çözümü için, rassal sayı kodlamanın sözkonusu olduğu GA kullanılarak, hızlı ve verimli bir model kurulmaya çalışılmıştır. Sözkonusu problem için geliştirilen model ile literatürde var olan diğer modeller ile rekabet edebilecek bir çalışma ortaya koyabilmek çalışmanın amacıdır.

Tezin şu an anlatılan giriş bölümünden sonra ikinci bölümünde ARP hakkında bilgi verilecek, çeşitleri ve en iyi çözüm veren kesin çözüm yöntemleri kısaca ele alınacaktır. Üçüncü bölümde Sezgisel Yöntemler, Klasik Sezgisel Yöntemler ve Meta-sezgisel Yöntemler olmak üzere iki alt başlıkta ele alınacak ve literatürde bu yöntemlerin ARP çözümünde kullanan çalışmalardan bahsedilecektir. GA'nın detaylı bir şekilde ele alınacağı dördüncü bölümde, GA ile yapılmış çalışmalar ele alınacaktır. Ayrıca bu çalışmada kullanılacak olan Rassal Sayı Kodlamalı GA ile ARP arasındaki ilişkiden kısaca bahsedilecektir. Daha sonra beşinci bölümde modeli kurulacak problemin tanımı yapılacak ve takip eden altıncı bölümde ise model anlatılacaktır. Son bölüm olan yedinci bölümde sonuçlar tartışılarak, öneriler ve konuyla ilgili gelecekte yapılabilecek çalışmalardan bahsedilecektir.

2. ARAÇ ROTALAMA PROBLEMİ

2.1 Araç Rotalama Problemi'nin Tanımı

Araç Rotalama Problemi (ARP), ilk olarak Dantzig ve Ramser tarafından 1959 yılında yapılan bir çalışma ile literatürde yerini almıştır [1]. Dantzig ve Ramser çalışmalarında, bir gerçek hayat uygulaması olan servis istasyonlarına benzin dağıtım problemini ele almışlar ve çözüm için bir matematiksel programlama modeli ve algoritmik yaklaşım ortaya koymuşlardır. Daha sonra 1964 yılında, Clarke ve Wright, Dantzig-Ramser yaklaşımını geliştiren etkili bir açgözlü sezgisel algoritma önermiştir. Başlarda küçük bir grup matematikçinin ilgisini çeken ARP, şu an bir çok disiplindeki araştırmacının üzerinde çalıştığı alan haline gelmiştir [1, 2, 3].

Ayrıca, ARP literatürde sıkça yer alan Gezgin Satıcı Problemi (GSP) ile benzerlik göstermektedir. GSP, ARP'nin en temel ve üzerinde en çok çalışılan versiyonudur. Aralarındaki fark GSP'de gezgin satıcının (aracın) kapasitesi yoktur dolayısıyla bir tek satıcı (araç) tüm müşterilere hizmet verebilmektedir. Dolayısıyla ARP için ortaya konacak etkili algoritmalar GSP'yi büyük oranda etkileyecektir [4, 5].

Kombinatorik en iyilemenin en önemli ve çok çalışılan problemlerinden biri olan ARP'de, belirli bir müşteri kümesine hizmet edecek olan bir araç filosunun izleyeceği rotaların en iyisinin belirlenmesine çalışılır. Klasik ARP problemlerinin çözümü, her rotanın depodan başlayıp depo ile bittiği ve her bir müşteriye bir kez uğranması kısıtının sağlandığı rotalar kümesidir. Bunun yanı sıra problemin türüne göre bazı yan kısıtların da sağlanması gerekebilmektedir. En yaygın olan yan kısıtlar; kapasite kısıtı, bir rotada olabilecek en fazla talep noktası kısıtı, bir rotada aracın toplam süre kısıtı, talep noktalarına hizmetin başlanabileceği zaman penceresi kısıtı, bir talep noktasının başka bir talep noktasından önce ziyaret edilmesinin gerektiği öncelik kısıtıdır [6].

Toplam taşıma maliyetlerinin en küçüklenmesi bu problemin amacıdır. Maliyeti yaratan etkenler gidilen mesafe, araç ve diğer elemanların kullanım süreleri, taşıma maliyetleri vs.. olabilmektedir. Dolayısıyla toplam gidilen mesafe, dağıtım süreleri ya da araç sayısı azaltılarak amaç fonksiyonu azaltılabilmektedir. Yani ARP'de, bir

depo, m tane araç ve n tane müşteriden bulunmaktadır ve başlangıçta depoda yer alan araçlar, toplam maliyet en az olacak şekilde sözkonusu müşterilerin ayrık miktardaki (kesikli) taleplerini karşılamak durumundadırlar [2,3].

ARP’de birden fazla ve genelde birbiriyle çelişen amaçlar dikkate alınabilmektedir. En bilinen amaçlar şunlardır:

- Araçların toplam katettiği mesafeye (ya da toplam seyahat süresine) ve kullanılan araçların sabit maliyetlerine (ve ilgili sürücüye) bağlı olan toplam taşıma maliyetlerini en küçüklemek
- Tüm müşterilere hizmet etmek için gerekli olan toplam araç sayısını (ya da sürücü sayısını) en küçüklemek
- Rotaları seyahat süresi ve araç yükü açısından dengelemek
- Müşterilere parçalı dağıtım yapılmasından kaynaklanan cezaları en küçüklemek

Yukarıda sayılan amaçların çeşitli kombinasyonları da problemin amaç fonksiyonunu oluşturabilmektedir [3].

Araç rotalama probleminde her araç bir rota olarak düşünülmekte olup, hangi müşterinin hangi araç tarafından hizmet göreceğine karar verilmesini rotalama, sözkonusu müşterinin atandığı rotada hangi sırada hizmet göreceğine karar verilmesini ise çizelgeleme olarak tanımlayan çalışmalar da bulunmaktadır. Dolayısıyla ARP’nin en temel hali olan GSP’de sadece çizelgeleme sözkonusu olup hem rotalama hem de çizelgelemenin olduğu ARP’den daha kolaydır [4].

2.2 Araç Rotalama Problemi Çeşitleri

Lojistikte önemli bir yer teşkil eden dağıtım tasarımı sorunlarından biri olan ARP, gerçek hayat koşullarından kaynaklanabilecek kısıtlar nedeniyle farklılaşabilmektedir. Zaman kısıtları; araçların aktif olabileceği zaman kısıtı, araç kullanıcısından kaynaklanabilecek zaman kısıtı, deponun açık kalabileceği ya da müşterilerin hizmet kabul edebileceği bir zaman aralığından kaynaklanan kısıt, mesafe kısıtları; araçların kat edebileceği mesafe kısıtı, kullanım kısıtları; kullanılabilecek toplam araç sayısı ya da araç kullanıcısı sınırlaması gibi kısıtlar nedeniyle bu farklılaşmalar ortaya çıkabilmektedir. Klasik ARP tanımında,

maliyetler, müşteri talepleri, araç seyahat süreleri gibi probleme ilişkin tüm parametreler belirlidir.

ARP çeşitlerinde öncelikle iki sınıflandırma yapabiliriz; problemin çözümünden önce tüm bilgilerin (kapasiteler, talep miktarları, mesafeler, talep noktası konumları, zaman pencereleri) bilindiği ve çözüm süresince değişmediği, statik ARP ve bunların değişken olabildiği dinamik ARP. Gerçek hayatın dinamik koşullarında, verimlilik ve hizmet seviyesindeki ihtiyaçların artması, taşımacılık sistemlerinin geliştirilmesi ve iyileştirilmesi gerekliliği gibi kaçınılmaz gerçeklerin olduğu düşünüldüğünde statik ARP modelleri gerçek uygulamaların basite indirgenmiş halidir [3]. Stokastik ARP problemlerinde ise yukarıda bahsedilen tüm parametreler rassaldır. Gündelik hayatta karşılaşılabilecek olası aksamaların yansıtılarak çözüme ulaşılmanın hedeflendiği Stokastik ARP modelleri çok daha zordur [7]. Dolayısıyla sıkça çalışılan bir konu olan ARP'nin literatürde çok sayıda çeşidi bulunmaktadır. En temel ARP problemi, araçların kapasite kısıtının da olduğu Kapasite Kısıtlı ARP ile literatürde çoğu zaman eş tutulmaktadır.

Aşağıda anlatılacak olan ARP çeşitleri birbirlerinden problemin sahip olduğu kısıtlar ile farklılık göstermektedir ve tüm parametreler statiktir. Müşterilere yapılacak dağıtım parçalanarak yapılabilir, ya da her müşterinin hizmet kabul edebileceği belirli bir zaman aralığı olabilir. Bu gibi farklılıklar sözkonusu çeşitliliğe sebep olmaktadır.

Kapasite Kısıtlı Araç Rotalama Problemi (Kapasite Kısıtlı ARP)

Dağıtım ağı en iyilemenin en önemli problemlerden biri olan [8] ve ARP'nin en yaygın türü olan Kapasite Kısıtlı ARP'de her aracın belirli bir kapasitesi vardır ve müşterilerin talepleri önceden bilinmektedir. En basit kapasiteli araç rotalama probleminde her aracın kapasitesi eşittir ve araçlar bir depodan harekete başlarlar ve en son yine bu depoya geri dönerler. Müşterilerin talepleri tek seferde teslim edilmektedir, parçalama sözkonusu değildir.

Literatürde üzerinde çok fazla çalışma olan bu problem türünde, her müşterinin belirli talep miktarı vardır ve her müşteri yalnızca bir araç tarafından ve sadece bir kez ziyaret edilebilir. Depo ve müşterilerin birbirlerine olan uzaklıkları simetriktir ve araçların seyahat süresi bu mesafeyle doğru orantılı alınabilmektedir. Bu problemde de sıklıkla amaç, araçların kat ettiği toplam mesafeyi en küçükmektir [8].

Müşterilere sadece bir kez uğranması gereken Kapasite Kısıtlı ARP’de amaç, talepleri bilinen müşterilerin toplam hizmet maliyetlerini en küçükleme. NP-zor sınıfına giren ARP için polinom zamanda en iyi çözümü bulabilen etkili bir algoritma yoktur. Matematiksel yöntemlerin kullanılması en iyi çözümü verecektir, fakat bu kesin yöntemler çözüm uzayının tamamını arayacağından hesaplama karmaşıklığına sahip olan ARP için problemin büyüklüğü arttıkça çözüm için gereken süre de üssel olarak artacaktır [3].

ARP için geliştirilmiş çok sayıda matematiksel model bulunmaktadır, literatürde yer alan Christofides ve diğ. 1981 yılında yaptığı çalışmasındaki tamsayılı modelde M araç sayısını ve N müşteri sayısını göstermektedir [9]. x_{ijk} değişkeni, i müşterisinden j müşterisine k aracı ile gidilip gidilmediğini belirtsin.

$$x_{ijk} = \left\{ \begin{array}{l} 1, \text{ eğer } k \text{ nolu araç } i \text{ müşterisinden } j \text{ müşterisine gidiyorsa} \\ 0, \text{ aksi takdirde} \end{array} \right\}$$

c_{ij} = i müşterisi ile j müşterisi arasındaki mesafe (maliyet)

Q = Bir araca yüklenebilecek toplam ürün miktarı

q_i = i müşterisinin talep miktarı

y_i = Alt turları engellemek için kullanılan rastgele değişken

Model şu şekildedir;

Amaç fonksiyonu:

$$\text{Min } z = \sum_{i=0}^N \sum_{j=0}^N (c_{ij} \sum_{k=1}^M x_{ijk}) \quad (2.1)$$

Öyle ki:

$$\sum_{i=0}^N \sum_{k=1}^M x_{ijk} = 1, \quad j = 1, \dots, N \quad (2.2)$$

$$\sum_{i=0}^N x_{ipk} - \sum_{j=0}^N x_{pjk} = 0, \quad k = 1, \dots, M, \quad p = 0, \dots, N \quad (2.3)$$

$$\sum_{i=1}^N (q_i \sum_{j=0}^N x_{ijk}) \leq Q, \quad k = 1, \dots, M \quad (2.4)$$

$$\sum_{j=1}^N x_{0jk} = 1, \quad k = 1, \dots, M \quad (2.5)$$

$$y_i - y_j + N \sum_{k=1}^M x_{ijk} \leq N - 1, \quad i \neq j = 1, \dots, N \quad (2.6)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j, k \quad (2.7)$$

$$y_i \text{ iřareti serbest} \quad (2.8)$$

Amaç fonksiyonu toplam yol maliyetini (mesafesini) en küçüklemeektir. (2.2) numaralı kısıta göre her müşteri yalnızca bir kez ziyaret edilecektir. (2.3) numaralı kısıt ise eğer bir araç bir müşteri ziyaret ediyorsa aynı zamanda o müşteriden hareket etmek zorunda olduğunu belirtir. Her aracın taşıyabileceğı toplam ürün miktarının bir sınırı vardır ve bununla ilgili kapasite kısıtı (2.4) numaralı ifade ile belirtilmiştir. (2.5) numaralı kısıta göre her araç yalnızca bir kez kullanılmalıdır, yani bir rotaya atanmalıdır. Problemin çözümünde alt turların engellenmesi için (2.6) numaralı kısıt eklenmiştir. Bu kısıt her rotanın depodan geçmesini sağlamaktadır. Christofides ve diğ. çalışmasında bu kısıtlara ek olarak her aracın toplam maliyet kısıtı da bulunmaktadır, bu kısıt aşağıda anlatılacak olan mesafe kısıtının da olduğu ARP çeşidinde yer almaktadır [9].

2.2.2 Mesafe Kısıtlı Araç Rotalama Problemi (Mesafe Kısıtlı ARP)

Mesafe Kısıtlı ARP'yi Kapasite Kısıtlı ARP'den ayıran kısıt, rotalara atanmış her aracın katedebileceğı belirli bir toplam mesafe olmasıdır. Bu durum gerçek bir dağıtım probleminde taşınan ürünün cinsinden, araç veya sürücü kısıtlarından dolayı sözkonusu olabilir. Eğer taşınan ürünün uzun süre taşıma nedeniyle bozulabilmesi sözkonusuysa, ya da araç kullanıcısının sürekli olarak belirli bir süreden daha fazla yolculuk yapamaması sözkonusu ise bu kısıt eklenmelidir. Dolayısıyla yukarıda ifade edilen modele bir kısıt daha eklenecektir:

T : Her bir aracın katedebileceğı toplam mesafe

$$\sum_{i=0}^N \sum_{j=0}^N c_{ij} x_{ijk} \leq T \quad (2.9)$$

2.2.3 Önce Dağıt Sonra Topla Araç Rotalama Problemi (Önce Dağıt Sonra Topla ARP)

ARP'nin bu versiyonunda müşteriler, ürün teslim edilecek ve tam tersi ürün teslim alınacak müşteriler olmak üzere iki çeşittir. Bir rotaya her iki müşteri tipinden de atanabilmektedir, fakat önce ürün dağıtımı yapılacak olan müşterilere uğranacaktır, daha sonra toplama işlemi yapılacaktır [3]. Dağıtımın ve toplamanın rastgele yapılması durumunda aracın yükleme yapılan tarafının yeniden düzenlenmesi ekonomik ve olurlu değildir.

Teslim edilecek ve toplanılacak miktarlar daha önceden bilinmektedir ve tüm araçlar aynı özelliklere sahiptir, yani aynı miktarda taşıma kapasitesine sahiptir. Burada dikkat edilmesi gereken diğer bir nokta da araçların hem dağıtım yapacağı miktar, yani depodan hareket etmeden önce araca yüklenecek ürün miktarı hem de teslim alınacak ürünlerin toplam miktarı söz konusu araç kapasitesini geçemez [10].

2.2.4 Eş Zamanlı Topla-Dağıt Araç Rotalama Problemi (Eş Zamanlı Topla-Dağıt ARP)

Eş Zamanlı Topla-Dağıt ARP'de ise, isminde de belirtildiği gibi, eş zamanlı olarak bir müşteriye hem ürün teslimatı yapılabilir ya da müşteriden ürün teslim alınabilir. Bu problem türünde müşteriler iki ayrı gruba ayrılmazlar. Yani önce dağıtım yapılacak ve sonra toplama yapılacak gibi bir kısıt yoktur. Araçlar rotaları boyunca dağıtım ve toplamayı bir arada yapabildiklerinden dolayı araç kapasitesinin her an korunması gibi zor bir durumla söz konusudur. Her müşteriye teslimat yapılacak ürün miktarı, ya da toplama yapılacak miktar önceden bilinmektedir. Bianchessi ve Righin'in yaptıkları çalışmalarındaki tanıma göre, Eş Zamanlı Topla-Dağıt ARP, yapılacak işlemler arasında bir öncelik ilişkisi olmadan aynı anda ürün dağıtımının ve atık toplamanın uygun şekilde birleştirilmesidir [11]. ARP'nin bu türü, hem ürün dağıtımının yapıldığı ve kullanılmış ürünlerin geri dönüşüm tesislerinde tekrar işlenmesi için geri toplanması ile ilgilenen ters lojistik uygulamaları için uygundur.

2.2.5 Bölünmüş Dağıtımlı Araç Rotalama Problemi (Bölünmüş Dağıtımlı ARP)

Bu problem tipinde bir müşterinin talepleri birden fazla araç tarafından temin edilebilmektedir, yani bir müşteriye farklı araçlar tarafından birden fazla uğranabilmektedir. Bu dağıtım şeklini, ortalama müşteri talebi çok büyük olduğunda

kullanmak anlamlıdır. Bölünmüş Dağıtımli ARP araçların gittiği toplam mesafeyi en küçüklerken, klasik ARP probleminden sadece sağladığı şu kısıt ile farklılaşmaktadır: Bir müşterinin talebi bir veya daha fazla araç tarafından temin edilir. Bir diğer farklılığı da yukarda belirtildiği gibi ortalama müşteri talebi klasik ARP'dekinden çok fazladır [12].

2.2.6 Çok Depolu Araç Rotalama Problemi (Çok Depolu ARP)

Daha önce bahsedilen ARP problemlerinde tek deponun olduğu kabul edilmiştir, bu problem türünde ise isminden de anlaşıldığı üzere araçların harekete başlayabileceği birden fazla depo bulunmaktadır. Depoların ve müşterilerin konumları önceden bilinmektedir ve her depo tüm müşterilerin toplam taleplerini karşılayabilecek kapasiteye sahiptir. Bu problemde her araç hareket ettiği depoya geri dönmek durumundadır. Birden fazla deposu olan bir dağıtım şirketinin araç rotalaması yapılmakta ise çok depolu olma durumunu yapılan modele ilave etmek gerekecektir. Bu ARP türü de NP-zor bir problemdir ve en iyi çözümün elde edilebileceği verimli bir yöntem bulunmamaktadır [4].

2.2.7 Periyodik Araç Rotalama Problemi (Periyodik ARP)

Periyodik ARP'de belirli bir dönemin planı en başta yapılmaktadır ve müşteriler bu süreçte birden fazla hizmet görmektedir. Müşterilere yapılacak servis sayısı müşterilerin talep miktarlarına, stok alanlarına göre değişmektedir. Eğer bir müşterinin talep miktarı çok fazla ise az miktarda talebi olan müşteriye göre ya da stoklama alanı küçük ise büyük olan müşteriye göre daha fazla ziyaret edilecektir. Bu problem sınıfı bakkaliye, alkolsüz içki endüstrisi, atık toplama gibi alanlarda ortaya çıkmaktadır [13].

2.2.8 Zaman Pencere Araç Rotalama Problemi (Zaman Pencere ARP)

Zaman Pencere ARP bu tez çalışmasında ele alınan ARP çeşididir. Bu problemde Kapasite Kısıtlı ARP'de olduğu gibi belirli sayıda ve aynı kapasiteye ve özelliklere sahip özdeş araçlar vardır, talep miktarları ve konumları bilinen müşteriler ve yine konumu bilinen bir tane depo bulunmaktadır. Bu problem türünü diğerlerinden ayıran ve problemi daha da güçleştiren kısıt, her müşteriye servise başlanabilecek belirli bir (a_i, b_i) zaman aralığının olmasıdır. Bu zaman aralığı içerisinde sözkonusu müşteriye hizmete başlamak gerekmektedir. Her müşteri için dağıtım ya da ürün

toplamak için belirli bir s_i servis süresi vardır ve araç müşteriye uğradıktan sonra bu s_i hizmet süresi kadar kalacaktır, daha sonra bir sonraki müşteriye ya da depoya hareket edecektir. Gerçek hayatta müşteriler ürün teslimatını belirli zaman aralığında kabul edebilirler, bu durumlarda bu problem türü için geliştirilen yöntemleri kullanmak uygundur.

Zaman Pencereli ARP iki alt sınıfa ayrılmaktadır:

Sıkı Zaman Pencereli ARP (Hard Time Windows VRP)

Eğer bir araç müşterinin sözkonusu zaman aralığının başlangıcından önce gitmişse beklemek durumundadır, eğer son tarihinden sonra gitmişse de hizmet verememektedir.

Esnek Zaman Pencereli ARP (Soft Time Windows VRP)

Esnek Zaman Pencereli ARP’de ise müşterilere ilgili zaman pencerelerinin dışında hizmet verilebilmektedir, fakat bu durumda bir ceza maliyeti sözkonusudur. Deponun zaman penceresi sağlanmak zorundadır [14].

Bu tez çalışmasında sıkı zaman penceresi kısıtlı sözkonusudur.

2.3 Araç Rotalama Problemleri İçin Çözüm Yöntemleri

NP_zor bir problem olan ARP çözümü için literatürde çok sayıda yöntem bulunmaktadır. Bu yöntemler en iyi çözümü veren kesin yöntemler ve sezgisel yöntemler olarak ikiye ayrılmaktadır. ARP için kullanılan kesin yöntemler 3 kategoriye ayrılabilir: Tamsayılı Doğrusal Programlama, Dinamik Programlama, Doğrudan Ağaç Arama Yöntemleri. Oldukça geniş bir kategori olan ve 3 temel yaklaşımın detaylandırılmış hali olan Tamsayılı Doğrusal Programlama yöntemleri birbiriyle örtüşmekte ve ilişkilidir. Sözkonusu 3 temel yaklaşım şunlardır: Küme Örtüleme Yaklaşımı, Araç Akışına Dayanan Formülasyonlar, Ürün Akışına Dayanan Formülasyonlar. Bu yöntemlerden özellikle Araç Akışına Dayanan Formülasyon sıklıkla kullanılmaktadır [5, 15].

2.3.1 Küme Örtüleme Yaklaşımı (Set-Covering Approach)

Tamsayılı Programlama’nın bir sınıfı olan Küme Örtüleme Problemleri’nde (KÖP) verilen bir kümenin (küme 1 olsun) her bir elemanı, başka bir kümenin (küme 2 olsun) uygun bir elemanı tarafından kapsanması gerekmektedir. Bu problemde amaç

küme 2’de en az sayıda eleman olacak şekilde küme 1’deki tüm elemanların kapsanmasıdır.

Model formülasyonu şu şekildedir;

x_j ikili (binary) karar değişkeni j araç kümesinin (vehicle cluster) kullanılıp kullanılmayacağını gösterebilir. Eğer bu değişken 1 ise j araç kümesi kullanılıyor, 0 ise kullanılmıyordur. a_{ij} karar değişkeni ise i talep noktasının j araç kümesine atanıp atanmadığını gösteren ikili katsayılar olsun. Bu durumda [5],

Amaç fonksiyonu:

$$\text{Min } z = \sum_{j=1}^J c_j x_j \quad (2.10)$$

Öyle ki:

$$\sum_{j=1}^J a_{ij} x_j \geq 1 \quad (2.11)$$

$$x_j = 0 \text{ veya } 1 \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, J \quad (2.12)$$

Modeldeki kısıt, bir talep noktasının uygun olan araç kümelerinden sadece bir tanesine atanmasını sağlamaktadır. c_j katsayısı ise j kümesine atanan talep noktalarının araç tarafından en az maliyetle uğranabileceği sıralamanın maliyetini göstermektedir. Bu katsayının hesaplanması GSP çözümü ile aynıdır. Bu model başka bir yan kısıta sahip olan ARP için uygulanacaksa ilgili kısıtlar eklenecektir. Örneğin her aracın bir sabit K kapasitesi varsa (Kapasite Kısıtlı ARP), ve d_i i talep noktasının talep miktarını belirtmek üzere şu kısıt eklenecektir.

$$\sum_{i=1}^n a_{ij} d_i \leq K \quad (2.13)$$

KÖP’nin özel bir hali olan Küme Bölümleme Problemleri’nde ise küme 1’in her bir elemanı, küme 2’nin uygun olan ve sadece bir elemanı ile eşleşmelidir. Havayolu mürettebat çizelgeleme, uçuş çizelgeleme, araç rotalama gibi çok sayıda gerçek hayat uygulamaları bu problem ile formüle edilebilir [16]. Bu yaklaşımın ARP’de kullanımında önce araç kümelemeleri gerçekleştirilmektedir, bu araç kümelerinde bazı talep noktaları söz konusu araca atanmıştır. Yani ilk aşamada bazı talep noktalarının ilgili araca atandığı kümeler söz konusudur. Bu kümelerin sayısı en az

olacak şekilde ve tüm talep noktalarını kapsayacak (her talep noktası bir araca atanacak) şekilde kümeler seçilmektedir. Maliyetlerin en azlanmaya çalışıldığı modelde bir aracın dağıtım maliyeti o araca atanmış noktaların en az maliyetle gezilebileceği sıralamanın maliyetidir. Yani bu aşama çizelgeleme aşamasıdır ve GSP ile birebir aynıdır. Bu formülasyonda talep noktalarının tüm araç kümelerine atandığı tüm olası durumları saymak (enumeration) gerekmektedir [17] ve dolayısıyla büyük boyutlu problemlerde çok da etkin olmayacaktır. Sayma işlemi şu şekilde açıklanabilir; Tamsayılı Programlamada değişkenler tamsayıdır ve her biri sonlu sayıda ayrık değerler alabilir. Değişkenlerin alabileceği tüm olası durumları ele alarak problemin amaç fonksiyonu hesapladıktan sonra en iyi ve olurlu olanını seçme işlemine sayma adı verilir [18].

2.3.2 Araç Akışına Dayanan Formülasyonlar (Vehicle Flow Based Formulation)

Araç akışının formüle edildiği ve temel ARP problemleri için sık kullanılan bu modellerde de ikili karar değişkeni kullanılır ve bu değişken bir k aracının i müşterisinden j müşterisine gidip gitmeyeceğini belirtir [3, 5].

x_{ij} i talep noktasından j talep noktasına olan yolun üzerinden geçilip geçilmeyeceğini ifade eden ikili karar değişkeni ve c_{ij} bu yolun maliyetini (mesafe, zaman, para) belirten katsayılarıdır. V talep noktalarının kümesidir ve 0 numaralı nokta depoyu göstermektedir.

Öyle ise model şu şekildedir [3];

Amaç fonksiyonu:

$$\text{Min } z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.14)$$

Öyle ki:

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (2.15)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (2.16)$$

$$\sum_{i \in V} x_{i0} = K \quad (2.17)$$

$$\sum_{j \in V} x_{0j} = K \quad (2.18)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (2.19)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \quad (2.20)$$

Amaç fonksiyonu toplam maliyeti en küçüklemeektir. (2.15) ve (2.16) numaralı kısıt sırasıyla bir talep noktasına sadece bir aracın gitmesini ve aynı noktadan bir aracın hareket etmesini sağlamaktadır. Bu kısıt depo haricindeki noktalar için geçerlidir, çünkü depodan toplam kullanılan araç sayısı kadar çıkış yaşanacaktır ve aynı sayıda araç depoya geri dönecektir ve bunu ifade eden kısıtlar ise (2.17) ve (2.18) numaralı kısıtlardır. $r(S)$, S alt kümesindeki talep noktalarına servis yapabilecek en az araç sayısını göstermek üzere, (2.19) numaralı kısıt çözümün bağlantılı olmasını ve kapasite kısıtını sağlamaktadır.

Daha önce ifade edilen ve bölüm 2.2.1’de verilmiş olan matematiksel model de araç akışına dayanmaktadır.

2.3.2 Ürün Akışına Dayanan Formülasyonlar (Commodity Flow Based Formulation)

Bu modellerde ise iki tür karar değişkeni mevcuttur; i talep noktasından j talep noktasına herhangi bir araç ile gidilip gidilmeyeceğini ifade eden ikili karar değişkeni ve ürün akışını temsil eden ve k noktasının talep miktarının ne kadarının i talep noktası ile j talep noktası arasında taşınacağını belirten karar değişkeni.

ARP için yapılan ilk çalışmalar arasında olan ve 1963 yılında Balinski ve Quandt’ın dağıtım problemi için geliştirdikleri tamsayılı programlamada yazarlar Genelleştirilmiş Örtüleme Problemi’nin çözümü için etkili bir algoritma ortaya koymuşlardır. Önce hangi araca hangi talep noktalarının atanacağını belirlediği kümeleme aşamasını gerçekleştiren ve daha sonra araçlara atanmış talep noktalarını çizelgeleyen Küme Örtüleme Yaklaşımı’nda, talep noktalarının araç kümelerine atanabileceği olası tüm durumların incelenmesinden kaynaklanan zorluğun, birbirine baskın olabilen rotaların kaldırması ile elde edilecek alt kümeyi çözerek giderilebileceğini ifade etmişlerdir. Elde ettikleri alt problemi Kesme Düzlemi

Algoritması'nı kullanarak çözmüşler ve yöntemlerini 15 dağıtım noktası olan bir problem için uygulamışlardır [15, 17].

Christofides ve diğ. (1981) yaptıkları çalışmalarında ARP'yi çözebilmek için bir dizi dal sınır algoritması ortaya koymuşlardır. Bu algoritmalar En Kısa Kapsayan Ağaç (Shortest Spanning Tree) ve En Az q-Rota (Minimum q-Routes) yöntemlerinden elde edilen sınırları temel almaktadır. Langrange ceza prosedürleri bu sınırları hesaplamak için kullanılmıştır. Elde ettikleri yöntemleri müşteri sayısı 10 ile 25 arasında değişen 10 farklı veri seti üzerinde uygulamışlardır. En Az q-Rota yaklaşımını kullandıkları yöntem daha iyi sonuç vermiştir ve 25 müşteriye kadar olan problemlerde en iyi sonucu elde edebilmişlerdir [9].

Magnanti'nin (1981), Laporte ve Nobert'in (1987) ve Laporte'nin (1991) yaptıkları çalışmalarda o güne kadar literatürde ARP için ortaya konmuş yöntemlerin üzerinden geçmiştir. Dal ve sınır algoritması, dinamik programlama, küme ayrıştırma ve sütun oluşturma, araç akışı gibi yöntemlerin formülasyonlarının kullanılarak oluşturulduğu yöntemler detaylı bir şekilde anlatılmıştır [5, 6, 15]. Ralphs ve diğ. 2003 yılında Kapasite Kısıtlı ARP için bir dal ve kesme algoritması geliştirmişlerdir [19].

ARP'nin yapısından kaynaklı olarak literatürde var olan kesin yöntemler (exact-optimal algorithm) sadece küçük boyutlu problemleri etkili bir şekilde çözebilmektedirler, dolayısıyla çok fazla müşterinin ve kısıtın olduğu gerçek hayat problemlerinde verimli bir şekilde kullanılamazlar. Sezgisel yaklaşımlar ARP için kabul edilebilir sürelerde ve en iyi çözüme çok yakın sonuçlar veren güvenilir yöntemlerdir. 1970'lerden bu yana, ARP için 1965 yılında sunulan ünlü Clarke ve Wright algoritmasından son zamanlarda kullanılan Tavlama Benzetimi, Tabu Arama, Değişken Komşu Arama ve benzeri yaklaşımlara kadar uzanan çok sayıda klasik sezgisel yöntemler önerilmiştir [20]. Söz konusu sezgisel yöntemler 1960 ile 1990 yılları arasında gelişen klasik sezgisel yöntemler ve daha yeni olan meta-sezgisel yöntemler olmak üzere iki sınıfa ayrılmaktadır [3].

3. SEZGİSEL YÖNTEMLER

3.1 Sezgisel Yöntemlere Giriş

Gündelik hayatta da karşılaşılabileceğimiz çok karmaşık problemleri çözmek yorucu ve uzun sürecektir. Eğer en iyi olanı elde etmeye çalışırsak, belirli bir süre sonra problemin içinden çıkamayacak ve o zamana kadar ki zamanımızı boşa harcamış olacağız. Genelde, karmaşık problemlere hızlı çözüm bulmak, en iyi çözüm bulmaktan daha kritiktir. Sürekli rekabet halinde olan şirketler, karşılaştıkları karmaşık problemler için en iyiye yakın çözümleri hızlı bir şekilde elde etmek zorundadırlar, aksi takdirde hızın çok önemli olduğu rekabet şartlarına direnemezler. Gerçek hayatta karşılaşılan karmaşık problemler, sayılabilir ve sonlu sayıda çözüme sahip olan kombinatorik problemlerle formüle edilebilirler. Kombinatorik problemlerin bir örneği için geliştirilecek naif bir yaklaşımda tüm olurlu çözümler listelenecek ve amaç fonksiyon değerlerine göre en iyisi seçilecektir. Bu kesin yöntem yerel en iyi noktalara takılmayacaklardır, fakat genel en iyiyi bulmak için çok fazla uzun zaman gerekmektedir. Diğer bir eksik noktası ise en iyi çözümün ıskalanabileceğidir, dolayısıyla kesin yöntemler küçük boyutta problemler için etkilidir [21]. Üzerinde çok fazla çalışılan, kolay ifade edilebilen fakat zor çözülebilen GSP problemini düşünecek olursak, eğer problem boyutu büyükse kesin yöntemlerle bu problemin çözümü çok zor olacaktır [22]. Yıllardır, kombinatorik en iyileme teorisinde yaklaşık çözüm veren algoritmalar yani klasik algoritmalar konusunda çok fazla çalışma mevcuttur [23].

Sezgisel yaklaşımlar iki gruba ayrılabilirler; klasik sezgisel (özel amaçlı) yaklaşımlar ve meta-sezgisel (genel amaçlı) yaklaşımlar. Klasik sezgisel yaklaşımlar belirli bir problem için olurlu çözümü hızlıca bulabilmektedir, fakat bulunan bu çözüm en iyi çözümden çok farklı olabilir. Meta-sezgisel yaklaşımlar en iyi çözüme yakın çözümler, hatta en iyi çözümleri bulabilmektedir. Dolayısıyla, Tavlama Benzetimi, Tabu Arama, Genetik Algoritma gibi meta-sezgisel yaklaşımlar en iyi çözümü bulmak için kullanılmaktadır [8].

3.2 Klasik Sezgisel Yöntemler

Bir önceki bölümde anlatılan ve ARP'nin çözümü için geliştirilen tamsayılı programlama modelleri küçük boyutlu problemler için en iyi çözümü vermektedirler. Gerçek hayata uygulanmak istendiklerinde problem boyutu büyüyeceğinden bu algoritmaların en iyi çözümü bulma süreleri de büyük oranda artacaktır. En iyi çözüm olmasa da en iyi çözüme çok yakın çözümleri hızlı bir şekilde elde etmek günlük hayatta daha etken bir yoldur. Bu nedenlerle ARP için de sezgisel yöntemler geliştirilmiştir ve 3 başlık altında incelenebilmektedirler: Yapısal (Constructive) Sezgisel Yöntemler, İki-Aşamalı Sezgisel Yöntemler ve Geliştirici Sezgisel Yöntemler [3].

3.2.1 Yapısal Sezgisel Yöntemler

ARP için geliştirilmiş Yapısal Sezgisel Yöntemler genel olarak şu şekilde çalışır: Önce maliyetin en azlanması kriterine göre müşteriler seçilir ve daha sonra kapasite ve zaman kısıtları dikkate alınarak rotalama gerçekleştirilir [2]. Literatürde sıkça bahsedilen ve Kapasite Kısıtlı ARP için ortaya konulmuş Clarke ve Wright Tasarruf Algoritması (1964) Yapısal Sezgisel Yöntemler'den en çok bilinenidir ve ARP için geliştirilmiş ilk sezgisel yöntemlerdendir. Bu algorithmada başlangıç koşulu olarak bir aracın bir müşteriye hizmet ettiği durum alınır. Eğer N tane talep noktası (müşteri) varsa başlangıçta depodan her bir talep noktasına giden ve tekrar depoya dönen N tane rota mevcuttur. Daha sonra sözkonusu rotaların birleştirilmesi ile tasarruf maliyetleri hesaplanır ve en yüksek tasarrufu sağlayacak şekilde rotalar daha fazla iyileştirme yapılamayınca kadar ve araçların kapasite kısıtı sağlanıncaya kadar birleştirilir [2]. Diyelim ki herhangi bir aşamadaki herhangi iki rota $(0, \dots, i, 0)$ ve $(0, j, \dots, 0)$ olsun (0 depoyu, i ve j talep noktalarını göstermektedir ve $i \neq j$). s_{ij} depodan önceki son talep noktası i olan rota ile depodan hemen sonraki ilk talep noktası j olan iki rotanın birleşmesinden elde edilecek tasarruf maliyetini, c_{ij} i talep noktasından j talep noktasına gitmenin maliyetini belirtsin. Öyle ise bu iki rota birleştirildiğinde $(0, \dots, i, j, \dots, 0)$ rotası elde edilecektir ve birleştirmeden elde edilecek tasarruf maliyeti şu şekilde hesaplanır: $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ [3]. Bu algoritmanın rotaların ardışık ve paralel yapılandırıldığı iki versiyonu bulunmaktadır, paralel olan daha iyi sonuç vermektedir [2]. Diğer bir yapısal sezgisel yöntem En Yakın Komşu Sezgiseli'dir. Bu yöntemde de bir rotaya eklenmiş en son müşterinin ardına bu

müşteriye en yakın olan başka bir müşteriye eklemektir. Bir rotaya atanacak ilk müşteri rassal olarak ya da keyfi seçilebilir. Ekleme işlemi araç kapasitesi sağlanana kadar yapılmaktadır [2]. Bu yöntem dışında Eşleme Temelli Tasarruf Algoritmaları (Matching-Based Savings Algorithms), Sıralı Ekleme Sezgisel Yöntemleri (Sequential Insertion Heuristic Methods) mevcuttur [3].

3.2.2 İki-Aşamalı Sezgisel Yöntemler

Bu sezgisel yaklaşımlarda önce kümeleme sonra rotalama ya da önce rotalama sonra kümeleme gibi iki aşamalı yöntemler mevcuttur. Süpürme Algoritması (SA) önce kümelemenin yapıldığı ve iki-aşamalı yöntemlerin en çok bilinenidir. İlk aşamada Kapasiteli ARP problemini, müşterileri kümelere ayırarak m-GSP problemine çevirir. Kümeleme işlemi iki kriter altında gerçekleştirilir: müşteriler ve deponun pozisyonları, depo orijin noktasında olacak şekilde, polar koordinatlara taşınır. İlk kriter müşterilerin birbirlerine olan açıdır ve en az açığa göre kümeleme gerçekleştirilir. İkinci kriter ise kümelenen müşterilerin taleplerinin araç kapasitesini geçmemesidir. İkinci aşamada ise her küme GSP problemi gibi çözülür [2]. Bu yöntemlerin yanı sıra Genelleştirilmiş Atama Temelli Algoritma (Generalized-Assignment-Based Algorithm) ve Konum Temelli Sezgisel (Location Based Heuristic) sayılabilir [3].

3.2.3 Geliştirici Sezgisel Yöntemler

ARP için ortaya konulmuş geliştirici sezgisel yöntemler bir veya birden fazla rota üzerinde işlem yapmaktadırlar. Bir rota üzerinde iyileştirme sözkonusu ise GSP’de kullanılan geliştirici yöntemler kullanılabilir. Bu yöntemlere Tek-Rota İyileştirme ve Çok-Rota İyileştirme sezgiselleri örnek olarak gösterilebilir [3].

3.3 Meta-Sezgisel Yöntemler

1980 yılının başlarında ortaya çıkan ve bugüne kadar teorisinde ve uygulama alanlarında etkileyici gelişmeler olan meta-sezgiseller yöntemler, çözümü zor olan kombinatorik en iyileme problemlerinde yaygın olarak ve başarılı bir şekilde kullanılmaktadır [23, 24]. Yinelemeli ileri çözüm süreçleri olan bu yöntemler iyi kalitede çözümler üretir ve alt kademedeki sezgisel yöntemlerin etkili bir şekilde birleşiminden oluşur. Her iterasyonda bir çözümden ya da çözümler topluluğundan

yola çıkarak yeni çözümler üretirler. Meta-sezgisel yaklaşımlar performanslarını arttırmak için klasik sezgiseller, yapay zeka, biyoloji, matematik, doğal ve fizik bilimlerinden türeyen kavramları kullanarak sadece yerel arama yapan ya da yapısal yöntemler olan alt kademedeki sezgisel yöntemlere üstünlük sağlamaktadırlar [23, 24]. Bilgisayar teknolojisindeki güçlü gelişmelere ve bilgisayar bilimi ile yönetim bilimi arasındaki büyük etkileşime rağmen işletme, mühendislik, ekonomi alanlarındaki önemli uygulamalarda kesin çözümler üretmek hala zordur. Meta-sezgisel yöntemler, sözkonusu uygulamalarda karar vericilere makul sürelerde yüksek kaliteli çözüm üretebilmeleri için sağlam araçlardır [24]. Bunun yanı sıra meta-sezgisel yöntemlerin teorisinde bazı varsayımlar bulunmaktadır ve çoğu uygulamada bu varsayımlar sağlanamaz. Dolayısıyla en iyi çözüme ulaşmak yerine yaklaşık çözümler elde edilir. Fakat yine de, meta-sezgisel yöntemlerin bu kusuruna rağmen çoğu en iyileme probleminde, bu problemlerin alt sezgisellerine göre en iyi ya da yaklaşık çözüm bulmada daha etkilidir [23].

Daha önce de ifade edildiği gibi kombinatorik en iyileme problemlerinden birisi olan ARP'nin çözümü için son yıllarda çok sayıda meta-sezgisel yöntem önerilmiştir. Bu yöntemler, çoğunlukla bazı standart rota yapılarını ve iyileştirici sezgisel yöntemleri içinde barındırarak iyi çözümlere ulaşmak için çözüm uzayını araştıran genel çözüm süreçleridir. Bu yöntemler, klasik yöntemlerden farklı olarak, arama sürecinde bozunuma (deteriorating) hatta bazen olursuz ara çözümlere de izin vermektedir. Klasik sezgisellerden daha iyi yerel sonuçlar elde etmektedirler, fakat bu yöntemlerin iyi çözüm bulması için de bazen uzun süre gerekmektedir [3]. Yaygın olarak kullanılan ve en çok bilinen meta-sezgisel yöntemler; Tavlama Benzetimi (TB), Tabu Arama (TA) ve Genetik Algoritmalar'dır (GA).

3.3.1 Tavlama Benzetimi (TB)

Tavlama Benzetimi'nin kesikli en iyileme için bir yöntem olarak kullanılışı 1980'li yılların başlarına dayanmaktadır. Yaygın olarak ve basit bir şekilde uygulanabilir olması nedeniyle, literatürde TB'nin kullanıldığı çok fazla çalışmaya rastlanır. TB'nin temelini oluşturan fikrin ortaya atıldığı ilk çalışma Metropolis ve diğ. 1953 yılında sıcak banyo içindeki malzemenin soğutulması olan tavlama işlemini simüle eden algoritması ile atılmıştır. Eğer katı cisim erime noktasına kadar ısıtılırsa ve

tekrar katı haline soğutulursa, soğutulmuş cismin yapısal özellikleri soğuma hızına bağlı olacaktır [25].

Yoğun madde fiziğinde tavlama, sıcak banyoda tüm molekülleri rassal bir şekilde olana kadar ısıtılmış ve akışkan hale getirilmiş katı (kristal) cisme uygulanan bir işlemdir. Kristal yapıdaki cisim erimiş hale geldikten sonra sıcaklık kristalin yapısı donmuş hale gelene kadar azaltılır. Eğer soğutma işlemi sırasında dış sıcaklık çok hızlı bir şekilde sıfıra düşürülürse ve kristalin ısıl dengeye ulaşması engellenirse, kristalin yapısında düzensizlikler ve kusurlar ortaya çıkacaktır. Bunun yaşanmaması için dış sıcaklık derece derece ve yavaş bir şekilde azaltılmalıdır. Her derecede kristal ısıl dengeye gelene kadar bekletilir. Bunun sebebi cismin donma noktasına gelirken enerjinin de düşmesini sağlamaktır ve kristalin yapısında orataya çıkacak bozukluğu engellemektir [23].

Tavlama işleminin simülasyonu, malzemenin parçacıklar sistemi olarak düşünülmesiyle gerçekleştirilebilir. Aslında, Metropolis'in algoritması malzemenin donma haline gelene kadarki soğutma işlemi sırasında sistemin enerjisinin değişimini simüle etmektedir. 30 yıl sonra istatistiksel mekanik ile kombinatorik en iyileme arasındaki ilişkiyi vurgulayan Kirkpatrick ve diğ., bu yaklaşımın en iyi çözüme ulaşmak için olurlu çözümlerin aranmasında kullanılabileceğini söylemişlerdir [25, 26]. TB her iterasyonunda küçük olasılıklarla kötü çözümleri de kabul eden yapısı ile en iyi çözüme takılmayı engelleyen, yerel arama temelli bir sezgisel yöntemdir [27]. Fakat yerel arama yönteminde mevcut çözüm sürekli iyileştirilmektedir, dolayısıyla genel en iyi yerine yerel en iyi çözüme takılmak daha olasıdır [23].

Fizikteki tavlama ile ilişkisi şu şekilde düşünülebilir; soğutma sırasında sistemin enerji seviyesi yavaşça düşürülmeye çalışılır ve donma noktasına ulaşıldığında en düşük seviyeye ulaşılır. En iyileme probleminin çözümünde azaltılmaya çalışılan amaç fonksiyonu, tavlamadaki sistemin enerjisi olarak düşünülebilir. Donma noktası da ulaşılmaya çalışılan yaklaşık çözümdür.

Algoritmada akış şu şekildedir: Bir olurlu çözümle ve bir T sıcaklık değeriyle başlanır. Bir sonraki adımda bu olurlu çözümün bir komşu çözümü değerlendirilir. Eğer yeni çözüm daha iyi bir amaç fonksiyon değerine sahip ise algoritma bu çözümle devam eder. Fakat daha kötü bir sonuç elde ediliyorsa bu çözümün kabul edilip edilmeyeceği başta belirlenen sıcaklığa ve $(0,1)$ aralığında üretilen bir R rassal

sayısının, amaç fonksiyon farkının ve mevcut sıcaklığın olduğu bir denkleme bağlıdır. Eğer t . adım ile $(t+1)$. adımıdaki amaç fonksiyon değerleri arasındaki fark Δ ise, $\Delta < 0$ ise yani yeni çözüm daha iyi sonuç veriyorsa kabul edilir. Aksi halde yani yeni çözüm daha kötüyse, $R \leq \exp^{-\Delta/T}$ eşitsizliği sağlanıyorsa bu kötü çözüm ile devam edilir, aksi halde algoritma mevcut çözümle devam eder. Bunun anlamı, mevcut çözümden çok kötü çözümlerin kabul edilme olasılığı, daha az kötü olana göre az olacaktır [28]. Söz konusu T sıcaklığı başta büyük alınır, böylece kötü çözümlerin kabul edilme olasılığı fazladır, fakat ilerledikçe bu sıcaklık değeri de $(0,1)$ aralığında bir soğutma oranına bağlı olarak azalır, böylece kötü çözümler daha az olasılıkla kabul edilir [23].

3.3.2 Tabu Arama (TA)

Bir başka meta-sezgisel yaklaşım olan TA, 1986 yılında Glover tarafından literatüre kazandırılmış olup, en iyileme problemlerinin çözümünde kullanılan yinelemeli bir arama yöntemidir [23] ve bir çok kombinatorik problem çözümünde de kullanılmıştır [29].

TB gibi komşu arama temelli ve yerel en iyi çözümden kaçan bir yöntem olan TA, insan belleği sürecini modellemeye çalışır [22]. Daha önce elde edilen çözümlerin basit fakat etkili bir şekilde kaydedildiği veri yapısına sahip olan yöntem, zeki problem çözüme ilkelerine dayanır [22, 23].

TA, Yerel Arama Yöntemi gibi komşu çözümlerde arama yapar, fakat bir yerel çözüme takıldığında iyileştirici olmayan harekete izin verir ve daha önce elde edilen ve hafızaya kaydedilmiş (tabu listesi) çözümlere tekrar dönmeyi engelleyerek çevrimin önüne geçer [29]. Geçmiş aramalardan oluşturulan tabu listesindeki çözümler belirli bir adım kadar yasaklanmıştır, bu çevrimin önüne geçtiği gibi aramada çeşitlilik de sağlamaktır [22]. Tabular kısa dönemli hafızada tutulmaktadır ve genelde sınırlı ve sabitlenmiş sayıda çözüm tutulmaktadır. Genelde mevcut çözüm üzerinde uygulanmış olan son bir kaç adım ve yasaklanmış ters dönüşümler (transformations) söz konusu hafızada tutulmaktadır [29].

TA yöntemi temel olarak ve özetle şu şekilde çalışır:

Adım1. İlk olarak Yerel Arama Yönteminde olduğu gibi olurlu çözümler kümesinden bir x çözümü x^{mevcut} olarak atanır. Tabu olan çözümleri tutan H tabu kümesi tanımlanır ve başta boştur.

Adım2. Belirlenen komşu arama yöntemi ile x^{mevcut} çözümünün tüm komşu çözümleri bulunur ve $N(H, x^{\text{mevcut}})$ kümesine atanır. $N(H, x^{\text{mevcut}})$ kümesi mevcut çözümün tüm komşu çözümlerini ve bu komşu çözümlerden belirli bir adım kadar dönülmeyecek olan tabu çözümlerin belirtildiği kümedir. $N(H, x^{\text{mevcut}})$ kümesindeki tabu çözümleri içermeyen aday çözümlerden oluşan bir alt kümesi olan $\text{aday_N}(x^{\text{mevcut}})$ belirlenir. Belirlenen bitirme koşulu sağlanıyorsa algoritma sonlanır ve yaklaşık en iyi çözüm elde edilmiş olur.

Adım 3. Eğer algoritma sonlanmıyorsa $N(H, x^{\text{mevcut}})$ kümesi güncellenir, yani H kümesine tabu çözümler eklenir ve $x^{\text{birsonraki}}$ çözümü yeni adımdaki x^{mevcut} çözümü olur. Mevcut çözüm için Adım.1'e geri dönülür [22].

Algoritmanın 2. adımındaki bitirme koşulu genelde; belirli bir adım sayısı kadar algoritmanın koşması, bir süre sonra belirli sayıdaki adımda herhangi bir iyileşmenin gerçekleşmemesi ya da çözümün başta belirlenmiş bir eşik değerinin altına düşmesidir. Ayrıca, TA sezgiselinin tasarımında arama uzayının ve komşu çözümler bulurken kullanılacak olan komşuluk yapısının belirlenmesi önemli bir adımdır. Bu nedenle model tasarlanmadan önce çözülecek olan problemin yapısı iyi anlaşılmalı ve bilinmelidir [29].

3.3.3 Genetik Algoritmalar (GA)

Evrimsel algoritmaların yaygın olarak bilinen bir çeşidi olan Genetik Algoritmalar (GA), doğadaki olayları taklit eden meta-sezgisel en iyileme yöntemleridir. Darwin'in evrim kuramına göre farklı özellik ve kabiliyetlerde olan ve üreme içgüdüsüne sahip bireylerden oluşan bir çevrede, sınırlı sayıda birey yaşayabiliyorsa, bireyler arasında seçim kaçınılmaz olacaktır. Uygun olanın hayatta kalması olgusu olarak da bilinen bu durum, doğanın var olan bireylere benzer özellikte bireyler üretmesi ve şartlara daha iyi uyum sağlayanları yani kaynaklar için daha iyi mücadele eden bireyleri desteklemesi ile açıklanabilmektedir. Dolayısıyla bu tanıma uyan bireyler, doğa tarafından çoğalmak üzere daha çok seçilecektirler. Yaşamın temel ilkelerini taklit eden ve genetik ve birçok en iyileme problemine rahatlıkla

uygulanabilen evrimsel algoritmalar, dönüşüm (mutasyon), çaprazlama ve seçim gibi genetik operatörleri bir allel dizisine uygularlar [30, 31].

GA'nın temel fikri seçici baskılar altında evrim geçiren aday çözümlerden oluşan bir popülasyonu korumaktır. Dolayısıyla GA, yukarıda anlatılan diğer meta-sezgisel yaklaşımlardan (TB, TA) farklı olarak, yerel aramanın bir tek çözüm üzerinde değil bir çözüm kümesi üzerindeki özellikleri değiştirerek yeni çözümler üreten sınıfına giren bir yöntemlerdir. Son yıllarda GA, GSP ve kuadratik atama problemi gibi zor en iyileme problemlerinin çok fazla sınıfında ve başarılı bir şekilde uygulanmaktadır. Bu yöntemin başarısı basit oluşuna, kolay uygulanabilir oluşuna ve esnek oluşuna bağlıdır [4].

GA özetle şu şekilde çalışır: Algoritma rassal çözümler kümesinden oluşan ve popülasyon adı verilen bir küme ile başlar. Popülasyondaki her çözüm arama uzayındaki bir noktayı temsil eder ve kromozom olarak adlandırılır. Her adımda sözkonusu kromozomlar evrim geçirir ve bu adıma üreme denir. Her üremede kromozomlar bazı uygunluk ölçüleri ile değerlendirilir. Kromozomlar ne kadar uygunsa üreme için gerekli olan çaprazlama ve mutasyon gibi genetik işlemlere tabi tutulmak için o kadar büyük olasılıkla seçilecektir. Çaprazlama aşamasında iki kromozom arasında bazı bölgeler yer değiştirerek yeni kromozomlar elde edilir ve ebeveynlerden oluşmuş yeni çözümlere çocuk denir. Bu işlem daha iyi çözümler elde etmek için uygulanır. Mutasyon aşamasında ise yerel en iyi çözümlere takılmanın önüne geçilmesi için popülasyon içerisindeki çeşitlilik korunmaktadır. Her adımda popülasyon büyüklüğü sabit kalacaktır, dolayısıyla üreme için uygunluklarına göre seçilecek ebeveynler ya da çocuklar haricinde kalan çözümler popülasyondan atılacaktır. Önceden belirlenmiş sayıda üreme sonrasında algoritma en iyi kromozoma yakınsayacaktır, belki en iyi çözümü ya da en iyiye yakın çözümü geri döndürecek [4].

Bu çalışmada ortaya konulan modelde GA kullanıldığı için bir sonraki bölümde daha detaylı şekilde ele alınacaktır.

3.4 ARP Çözümünde Meta-sezgisel Yöntemlerin Kullanımı

ARP problemi kombinatorik bir en iyileme problemidir, bu sınıfa giren en iyileme problemleri gibi ARP için de çok fazla meta-sezgisel yöntemlerin kullanıldığı çalışmalar yapılmıştır. Literatürde yer alan çalışmalara bakacak olursak; ARP'nin zor

bir çeşidi olan Zaman Pencereli ARP için çok sayıda çalışmaya rastlanmaktadır. Chiang ve Russell 1996 yılında yaptıkları çalışmalarında söz konusu problemi çözmek için TB sezgiselinden faydalandıkları çalışmalarında 3 farklı algoritma sunmuşlardır. Solomon'un veri setini kullanarak değerlendirdikleri TB algoritmaları 2 farklı komşuluk yapısından (Osman'ın λ -takas (swap) mekanizması, Christofides'in k-düğüm takas işlemi), kısa dönem hafızalı yasak listesinden ve rota oluşturma sırasında çağırılan geliştirici prosedürlerden oluşmaktadır [32]. Bir yıl sonra, Badeau ve diğ. geliştirdikleri modelde aynı problem için TA sezgiselinden faydalanmışlardır. Aynı anda birden fazla işlemci tarafından paralel olarak çalıştırılan modelde bir ana işlemci ve bu işlemciye bağlı ve iletişim içinde olan yan işlemciler vardır. Tüm işlemciler farklı başlangıç noktalarından başlayarak aynı anda çözüm uzayının farklı yerlerini ararlar. TA yöntemi içerisinde başlangıç çözümleri elde etmek için Rassal Ekleme Sezgisel Yöntemi, komşu çözümler aramak için ise Çapraz (CROSS) değiştirme yöntemi kullanılmaktadır [14]. Czech ve Czarnas ise 2001 yılında yaptıkları çalışmalarında, TB yönteminin paralel olarak kullanıldığı bir model geliştirmişlerdir. Yine aynı anda birden fazla işlemcinin aynı problem için TB çalıştırdığı modelde, belirli bir adım sayısından sonra başta belirlenen bir işlemciye (0. işlemci) geri kalan işlemciler mevcut iyi çözümlerini göndermekte ve bu işlemci diğerlerinin en iyi çözümünü kendi çözümüyle karşılaştırmaktadır. Eğer daha iyi bir çözüm ise yeni adımda bu çözümle devam etmektedir [33]. Yine Solomon'un veri setinden faydalanılan ve aynı problem çeşidi için yapılan bir başka çalışmada, TB için gerekli olan başlangıç çözüm İlerletici Ekleme Sezgisel Yöntemi (Push-Forward Insertion Heuristic) ile elde edilir. Her adımda 5 farklı komşu çözüm bulan yöntemlerden rassal olarak birini kullanan algorithmada, tavlama sıcaklığı diğer çalışmalardan farklı olarak sürekli azalmaz. Eğer 1000 adımda iyi çözümde bir değişiklik olmazsa sıcaklık derecesi artabilir. 30000 adım sonrasında TB algoritması sonlanmaktadır, daha sonra elde edilen çözüme Tepe Tırmanma Stratejisi (Hill Climbing Strategy) uygulanarak melez bir çözüm elde edilmektedir. Bu yöntemin kullanılma sebebi probleme TB uygulamasının başında sıcaklık çok yüksektir, yani kötü çözümlerin kabul edilme olasılığı yüksektir, bu nedenle mevcut çözüme komşu bir çözüm TB'nin sonuçta verdiği çözümünden daha iyi olabilir ve sıcaklığın yüksek olması nedeniyle bu çözümünden uzaklaşılabilir. TB sırasında her 1000 adım sonrasındaki çözüme bu strateji uygulanarak, arama yapılan bölge farklılaştırılır. Bu stratejinin uygulanma sebebi TB'nin sonlandığında elde edilecek iyi bir çözümünden

daha iyi bir çözüm ya da eşit bir çözüm elde etmektir [28]. Lin ve diğ. de TB sezgisel yöntemini ve Solomon'un veri setlerini algoritmanın etkinliğini ölçmek için kullandıkları çalışmalarında, TB için gerekli olan ilk çözümün Sıralı Ekleme Sezgisel'i ile elde etmişler ve amaç fonksiyon değeri olarak, araç sayısının, zaman penceresi kısıtının sağlanamaması durumunda eklenecek ceza maliyetinin ve uzaklığın, ağırlıkları azalan şekilde olacak şekilde, bir kombinasyonu olarak almışlardır [34].

1997 yılında Brandão ve Mercer, ARP'nin farklı bir çeşidi olan, bir aracın gün içerisinde birden fazla sefer yapabildiği Çok Seferli Araç Rotalama ve Çizelgeleme problemi için TA yöntemini kullanmışlardır [35]. Gendreau ve diğ. (1999) ise TA sezgisel yöntemini her bir aracın kapasitesinin, sabit maliyetinin ve değişken maliyetinin farklı olduğu heterojen filolu ARP çözümü için kullanmışlardır [36].

Nanry ve Barnes aynı yöntemin ileri bir hali olan, algoritma parametrelerinin her adımda aramanın mevcut durumuna ve kalitesine bağlı olarak ayarlandığı Reaktif TA'yı, Zaman Pencereli Ürün Toplama ve Dağıtım Problemi için kullanmışlardır. Söz konusu problemde her bir müşterinin sisteme ilettiği isteği, belirli bir yerden ürün toplanması ve toplanan ürünlerin başka bir yere teslim edilmesidir. Her lokasyonun belirli bir zaman penceresi vardır, toplama ve dağıtım yapılacak ikili yerlere aynı araç tarafından hizmet verilir. Başlangıç çözümü için Öncül-Ardıl Çift Ekleme Algoritması'nı, komşu çözümler için ise bir tek çiftin araya sokulduğu yaklaşım, rotalar arası çiftleri değiştirme ve rota içinde çiftlerin yerlerini değiştirme gibi yaklaşımlar kullanılmıştır. Yöntem Solomon'un Zaman Pencereli ARP veri setinden uyarlanmış veri seti için kullanılmıştır [37]. 2004 yılında yapılmış bir başka çalışmada ise TA yöntemi, hem her bir konumun (talep noktası ve depo) zaman penceresinin olduğu hem de taleplerin bölünerek karşılanabildiği ARP için kullanılmıştır. Başlangıç olurlu çözüm bulunurken rotalanmamış en yakın komşu olgusuna dayanan bir yaklaşım kullanılmıştır. Komşu çözümler için ise yeniden yerleştirme, takas etme ve 2-opt işlemlerinden faydalanılmıştır. Bu modelin etkinliği de Solomon'un veri seti ile test edilmiştir [38]. Moghaddam ve diğ. 2005 yılında yaptıkları çalışmalarında ise TB yöntemini Bölünmüş Dağıtımlı ARP için kullanmışlardır. Araç sayısı başta belirlidir, fakat araç kapasiteleri ve maliyetleri değişkendir [39]. Tavakkoli-Moghaddam ve diğ. (2006) yılında yaptıkları çalışmalarında Kapasite Kısıtlı ARP'nin heterojen araçlı hali için geliştirdikleri

modelde toplam maliyeti en azlarken araç kullanım oranını en çoklamaya çalışmışlardır. Maliyetin mesafeden bağımsız olduğu kabul edilen çalışmada, TB yöntemi için En Yakın Komşu Sezgiseli'ni temel alan bir yöntemle başlangıç çözümü elde etmektedirler. Küçük boyutlu problemler için en iyi çözümü veren bu yöntem, büyük boyutlu problemler için iyi bir başlangıç çözümü vermektedir. TB ile elde ettikleri başlangıç çözümünü iyileştiren araştırmacılar, komşu çözüm bulmak için 1-opt ve 2-opt yöntemlerini kullanmışlardır [40]. Lin ve dig. ise Kapasite Kısıtlı ARP için melez bir meta sezgisel yöntem geliştirmişlerdir. Çalışmada TB ve TA yöntemlerinden faydalanılmıştır ve sözkonusu yöntemler yerel arama ile birleştirilmiştir. Literatürde yer alan ve diğer araştırmacılar tarafından da kullanılmış olan 14 denektaş problemi üzerinde, hem küçük hem de büyük boyutlu problemler için denenen yöntem, diğer yöntemlerle rekabet edebilecek kadar iyidir [8].

2006 yılında Çok Depolu ARP probleminin her depo için kullanılabilecek araç sayısının sabit olduğu çeşidi için, Lim ve Zhu tarafından TB'nin kullanıldığı bir çalışma mevcuttur. Çalışmada ilk önce Rassal En İyi Yerleştirme (Randomized Best Insertion) yöntemi ile araç sayısı en azlanarak TB için başlangıç çözümü elde edilmektedir. Daha sonra genelleştirilmiş n-op işlemlerine dayanan komşuluk işlemlerini kullanan TB ile, elde edilen başlangıç çözümü toplam mesafeyi azaltacak şekilde iyileştirilir. Son olarak da TB ile elde edilen çözümdeki her rotadaki alt turlar GSP için kullanılan Dinamik Programlama Algoritması en iyi hale getirilir [41]. Diğer bir çalışmada ise Periyodik ARP için bir sezgisel yöntem geliştirilmiştir. Değişken Komşuluk Arama (Variable Neighborhood) sezgiseline dayanan bu yerel arama yönteminde komşuluk belirlenmesi TB yönteminden etkilenmiştir. Yöntem periyodik araç rotalama problemi ve periyodik gezgin satıcı problemi için kullanılmıştır [13].

Lin, Yu ve Chou ARP'nin farklı bir versiyonu olan, bir römorkun bir kamyon tarafından çekildiği kamyon-römork birleşiminden oluşan bir aracın da servis için kullanılabildiği, kamyon ve römork rotalama probleminin çözümü için TB'den faydalanmışlardır. Bu problemde iki tür müşteri vardır; sadece kamyon ile servis yapılabilecek müşteriler ve kamyon-römork aracı ile servis yapılabilecekler. Birinci tip müşteri eğer kamyon-römork aracına atanmışsa, bu araç belirli bir noktada durup römorkunu park ederek sadece kamyon bu müşteriye hizmet vermekte ve römorkunu almak üzere tekrar park yerine geri dönmektedir. Buna alt tur denir, yani sadece

kamyon belirli noktaları dolaşır ve asıl ana turu römorku ile birlikte devam ettirir. Herhangi bir çözüm müşteri sırasından (bu müşterilere kukla depolar veya park yerleri de ilave edilmiştir) ve bu müşterilerin hangi tür müşteri olduğu bilgisini veren (sadece kamyon-römork aracı müşterileri için) bir gösterimden oluşmaktadır. TB uygulamasında bir çözümün komşu çözümü; ekleme, takas ve servis edecek aracın çeşidi değiştirilerek elde edilir. Araştırmacıların literatürde yer alan denektaşlarından faydalanarak elde ettikleri sonuçlar TB'nin bu problemde başarıyla uygulanabileceğini göstermiştir [27].

Brandão ARP'nin araçların dağıtım sonrasında tekrar depoya dönmediği, eğer döneceklerse dağıtım sırasında uğradıkları müşterilere ürün toplamak için ters sırada tekrar uğramalarının gerektiği hali için TA yöntemini kullanmışlardır. Başlangıç çözüm için en yakın komşu sezgiseli ve sözde (pseudo) alt sınır temelli başka bir sezgisel kullanılmıştır, komşu çözümler ekleme ve takas etme yaklaşımları kullanılmıştır [42]. Aynı araştırmacının 2006 yılında yaptığı çalışmada ise Önce Dağıt Sonra Topla ARP için TA yönteminden faydalanmıştır [43].

Bu çalışmaların hangi ARP çeşidi için yapıldığını, hangi yöntemi kullandığını, başlangıç çözüm ve komşu çözümler için hangi alt sezgisellerden yararlandıklarını ve son olarak eğer kullanıyorlarsa hangi literatürdeki hangi veri setini kullandıklarını kısaca özetleyen bilgiler bir sonraki sayfadaki Çizelge 3.1'de verilmiştir. ARP için diğer son sırada söz edilen meta-sezgisel yaklaşım GA'nın kullanıldığı çalışmalar 4. bölümde detaylı şekilde verilecektir.

Çizelge 3.1 : Meta-sezgisel yöntemlerin kullanıldığı çalışmalar.

Yazar	Yılı	ARP çeşidi	YÖNTEMLER	Başlangıç Çözüm İçin Kullanılan Yöntemler	Komşu Çözüm İçin Kullanılan Yöntemler	Kullanılan Veri Seti
Wen-Chyuan Chiang & Robert A. Russell [32]	1996	Zaman Pencereli ARP	*Tavlama Benzetimi	*paralel yapısal sezgiseli (parallel construction heuristic)	* λ -takas mekanizması *k-düğüm takas yöntemi	*Solomon
Philippe Badeau ve diğ. [14]	1997	Zaman Pencereli ARP	*Tabu Arama (Paralel)	*rassal ekleme sezgiseli	*çapraz takas	*Solomon
Zbigniew J. Czech & Piotr Czamas [33]	2001	Zaman Pencereli ARP	*Tavlama Benzetimi	*rassal	*rotalar arası & rota içi en iyi müşteri takası	*Solomon
Humberto César Brandão de Oliveira ve diğ. [28]	2006	Zaman Pencereli ARP	*Tavlama Benzetimi *Tepe Tırmanma Stratejisi	*ilerletici ekleme sezgisel yöntemi (PFH)	*takas *ekleme *karıştırma *ters çevirme	*Solomon
S.-W. Lin ve diğ. [34]	2006	Zaman Pencereli ARP	*Tavlama Benzetimi	*sıralı ekleme sezgiseli	*takas *ekleme	*Solomon
Jose Brandão & Alan Mercer [35]	1997	Çok Seferli ARP	*Tabu Arama	*en yakın komşu alg. *ekleme alg.	*ekleme *takas	*gerçek uygulama (dağıtım şirketi)
Michel Gendreau ve diğ. [36]	1999	Heterojen filolu ARP	*Tabu Arama	*GENIUS	*yerel arama	*Golden *Taillard
William P. Nanry & J. W. Bames [37]	2000	Zaman Pencereli Topla-Dağıt Problemi	*Tabu Arama	*öncül-ardıl çift ekleme alg.	*çift ekleme *rotalar arası & rota içi takas	*Solomon (türetme)
S.C. Ho & D. Haugland [38]	2004	Zaman Pencereli ve Bölünmüş Dağıtımli ARP	*Tabu Arama	*en yakın rotalanmamış komşu alg.	*yeniden yerleştirme *takas *2-opt	*Solomon
R. Tavakkoli-Moghaddam ve diğ. [39]	2005	Bölünmüş Dağıtımli ve Kapasiteli ARP	*Tavlama Benzetimi	*Kapasiteli ARP temelli bir sezgisel	*1-opt *2-opt	-
R. Tavakkoli-Moghaddam ve diğ. [40]	2006	Kapasiteli ARP	*Tavlama Benzetimi	*en yakın komşu alg.	*1-opt *2-opt	-
Lin, S. -W. ve diğ. [8]	2008	Kapasiteli ARP	MELEZ *Tavlama Benzetimi *Tabu Arama	*en yakın komşu alg.	*takas *ekleme	*Christofides ve diğ.
Andrew Lim & Wenbin Zhu [41]	2006	Çok Depolu ARP	*Tavlama Benzetimi	*rassal en iyi ekleme algoritması	*genelleştirilmiş n-op yöntemi	A. Lim & F. Wang
Shih-Wei Lin ve diğ. [27]	2009	Kamyon & Römork Rotalama	*Tavlama Benzetimi	*rassal	*ekleme *takas *araç çeşidi değiştime	*Chao
Jose Brandão [42]	2004	Açık Uçlu ARP	*Tabu Arama	*en yakın komşu alg. *sahte(pseudo) alt sınır temelli sezg.	*ekleme *takas	*Christofides *Fisher
José Brandão [43]	2006	Önce Dağıt Sonra Topla ARP	*Tabu Arama	*açık başlangıç çözüm *K-ağaç başlangıç çözüm	*ekleme *takas *rota içi takas	*Wassan

4. GENETİK ALGORİTMA

4.1 Genetik Algoritma'nın Tarihçesi

Darwin'in evrim teorisinden esinlenilerek oluşturulan GA, yapay zekanın giderek daha da önem kazanan bir kolu olup evrimsel hesaplama işlemlerine dayanır. Genetik Algoritma kavramı, bir çok biyoloğun biyolojik sistemlerin simülasyonu için bilgisayar kullanmaya başladığı 1950'li yılların başlarına dayanmaktadır [44]. İlk olarak 1960'lı yıllarda J. Rechenberg tarafından "Evrimsel Stratejileri (Evolutionstrategie)" isimli eserde tanımlanan evrensel hesaplama [45], daha sonra geliştirilmiş ve bilgisayar bilimi uzmanı olan John Holland'ın çabaları sonucunda genetik sürecin bilgisayar ortamında ele alınışı ilk kez gerçekleştirilmiştir [46]. 1975'te "Doğal ve Yapay Sistemlerde Adaptasyon (Adaptation in Natural and Artificial Systems)" adındaki kitabında çalışmalarının sonuçlarını toplayan Holland, ortaya çıkardığı bu yöntemin Genetik Algoritma adı ile anılmaya başlanmasını sağlamıştır. Daha sonra, Holland'ın bu buluşlarının sadece teorik yararları olduğuna ve pratikte yararlılığının az olduğuna ilişkin eleştiriler yapılmıştır. Holland'ın öğrencisi olan inşaat mühendisi David E. Goldberg, 1985 yılında bitirdiği doktora çalışması ve 1989'da yayınladığı kitabı ile genetik algoritmanın pratikteki yararlarına dikkat çekmiştir. Genetik algoritmanın uygulama alanlarına değindiği kitabında, 83 uygulamaya yer vermiş ve GA'nın uygulanabildiği bir çok alan olduğunu göstermiştir [46]. GA'nın kullanarak çeşitli problemlere çözüm üretmesini sağlayan programlar geliştirilmesi ise, 1992 yılında gerçekleştirilmiştir. 1992 yılında John Koza tarafından geliştirilen programlama yöntemine Genetik Programlama adı verilmiştir [45].

Bugüne kadar geliştirilen tüm yöntemlerin temel amacı belirli bir problemi çözmektir, yani potansiyel çözümlerden oluşan bir uzayı araştırmaktır. Küçük boyutlu çözüm uzayının sözkonusu olduğu durumlarda tüm uzayı eksiksiz arayan kesin yöntemler yöntemler kullanılabilir; fakat arama uzayının boyutunun büyük olması bazı özel yapay zeka tekniklerini gerektirmektedir. GA böyle bir yöntemdir; arama işlemini genetik kalıtım ve Darwin'in hayatta kalma mücadelesi

gibi bazı doğal fenomenleri kullanarak gerçekleştiren stokastik bir arama algoritmasıdır [44].

4.2 Genetik Algoritma'daki Kavramlar

GA'nın nasıl çalıştığına geçmeden önce algortmada geçen kavramlardan bahsedilecektir.

Genotip ve fenotip:

Genetikteki bir temel gözleme göre, her birey genotip (kalıtsal yapı) özelliklere ve bu kalıtsal (genetik) özellikler sonucu oluşan fenotip (kalıtımla oluşan dış görünüm) özelliklere sahip olan çiftleş bir varlıktır (dual entity). Yani, bir bireyin genotipi onun fenotipini şifreler [30]. Başka bir tanıma göre, genotip bireyin mevcut gen yapısı ve fenotip ise organizmanın gözlenen özellikleridir [47]. Üreme sırasında çocuk ebeveyninin fenotip özelliklerini miras olarak almaz, sadece fenotip özelliklere ait genotip bilgilerini miras alır. Çocuk bu bilgileri hem anneden hem babadan almaktadır. Dolayısıyla, bireylerin gelişimi fenotip seviyesinde gerçekleşirken, genetik operatörler genotip seviyesinde gerçekleşmektedir [31].

Allel:

Alleller bir kromozomdaki en küçük birimlerdir. Doğada, alleller XY gibi iki tane sembolden oluşmaktadır, fakat genetik algoritma uygulamalarında çoğunlukla tek sembol ile ifade edilirler. İlerde bahsedilecek ikili kodlamada, alleller 0 veya 1 değeri alabilmektedirler [31].

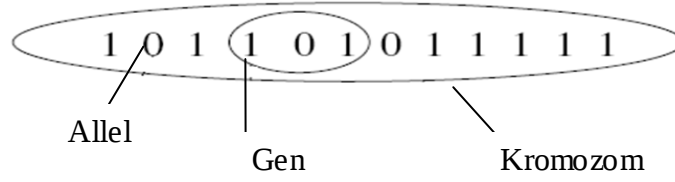
Gen:

Gen, bir kromozom (birey) içerisinde belirli bir fenotip özelliği temsil eden allel ya da alleller dizisidir. Yani, bir bireyin dış fenotipik bir özelliği bir ya da birden fazla allelin bir araya gelmesi ile oluşan genlerle temsil edilmektedir [31].

Kromozom (Birey):

Genetik algortmada bir çözümü temsil eden, belirli uzunluktaki ve bireyin tüm genetik bilgisini içeren kromozom yani birey alleller dizisidir. En iyileme probleminde tüm olası çözümler bir kromozom ile temsil edilmektedir. Doğada birden fazla kromozom kullanılmasına karşılık, çoğu GA uygulamalarında bir kromozom kullanılmaktadır [31].

Şekil 4.1’de yukarda açıklanan allel, gen ve kromozom arasındaki ilişki ifade edilmiştir.



Şekil 4.1 : Allel-gen-kromozom ilişkisi.

Popülasyon (Nüfus):

Popülasyon çözüm uzayındaki belirli sayıda noktanın oluşturduğu çözüm kümesidir, yani belirli sayıda kromozomdan oluşur. GA’nın başında bu çözümler rassal olarak belirlenmektedir ve genellikle bu kümenin büyüklüğü algoritma boyunca sabit tutulmaktadır. Algoritmanın başında belirlenmesi gereken en önemli nokta bu popülasyonun büyüklüğünün ne kadar olacağı ve hangi bireylerin üreme için seçileceğidir. Sözkonusu çözüm kümesinin büyüklüğü algoritmanın etkili ve verimli bir şekilde çalışmasına yönelik olarak belirlenmelidir. Küçük boyutta popülasyon GA’nın aynı anda çözüm uzayının farklı yerlerini araması özelliğini etkin kılmayacak, çok büyük olması ise algoritmanın hızlı çalışmasını engelleyerek verimliliğini azaltacaktır. Bu konuyla ilgili detaylı bilgi için [48]’e bakılabilir.

4.3 Genetik Algoritma Aşamaları

Aşama 1. Genetik Kodlama

Bir bireyin, genlerinin deşifre edilmiş fenotipinin kodlanmış hali bireyin gen yapısını verir. En iyileme probleminde de benzer şekilde, problemin bir çözümü olan bir birey için genetik kodlama, bu bireyin gen yapısını oluşturmaktır.

Problemin aday bir çözümü olan bir bireyin gen yapısının nasıl kodlanacağı, sözkonusu gen yapısı ile deşifre edilmiş yapısı arasındaki eşleştirmenin nasıl yapılacağını belirlemek, herhangi bir evrim algoritmasının ilk aşamasını oluşturmaktadır [30]. Belirlenen kodlama ile en iyileme probleminin olası tüm olurlu çözümlerinin şifrelenebilmesi ve çaprazlama, mutasyon (dönüşüm) gibi genetik operatörlerin, bu kodlama ile yapılmış bireylere uygulanabilir olması gerekmektedir. Genetik kodlamadaki ufak değişiklikler sonucu algoritmanın performansı büyük

ölçüde değişebilmektedir, genelde araştırmacının sezgileri ile belirlenen sözkonusu kodlamanın doğru seçilmesi algoritmanın performansını önemli ölçüde etkilemektedir [31]. Çok sayıda mevcut olan genetik kodlama çeşitleri aşağıda açıklanmaktadır.

İkili Kodlama:

En eski kodlamalardan biri olan ve yaygın olarak kullanılan ikili kodlama, Holland'ın ilk genetik algoritma çalışmalarında kullanılmıştır [30, 46]. İkili kodlamada kromozomdaki bir allel 0 ve 1 değerini alabilmektedir, yani çözüm uzayındaki değerler ikilik sayma sistemine göre kodlanmıştır. Dizi uzunluğunun ne kadar olacağına karar vermek gerekmektedir ve sözkonusu dizinin, genin deşifre edilmiş haline nasıl dönüştürüleceği belirlenmelidir.

Gri Kodlama:

Her problem için klasik ikili kodlamanın en iyi ve tek mümkün kodlama olmadığını öne süren araştırmacılar gri kodlamayı önermişlerdir [47]. İkili kodlamada ortaya çıkan sorun; çözüm uzayında ardışık gelen iki noktanın kodlandığında birbirinden çok farklı bireyler ortaya çıkmasıdır ya da tam tersi ardışık iki genotip birbirinden farklı iki fenotipi temsil edebilmesidir. 7 sayısına denk gelen 0111 ve ardışığı olan 8 sayısına denk gelen 1000 bu duruma örnek olarak gösterilebilir. Gri kodlama ardışık iki çözümün kodlamalarının birbirlerine olan uzaklığı Hamming uzaklığı cinsinden 1 olacağını garanti eder, yani ardışık iki genotip ardışık iki fenotipe karşılık gelecektir [30, 47]. Özetle aralarındaki fark hangi genotipin hangi fenotipi temsil edeceği farklıdır. Çizelge 4.1'de klasik ikili kodlama ve gri kodlama arasındaki ilişki gösterilmiştir.

Çizelge 4.1 : Standart ikili kodlama ve gri ikili kodlama.

Tamsayı	0	1	2	3	4	5	6	7
Standart	000	001	010	011	100	101	110	111
Gri	000	001	011	010	110	111	101	100

Tamsayılı Kodlama:

Kromozomdaki allellerin kardinalitesi $\chi=2$ olan ikili kodlamanın dışında daha yüksek olan dizilerden de oluşabilirler. İkili kodlamada l uzunluğundaki bir

kromozom 2^l farklı bireyi temsil edebileceksen, daha yüksek kardinaliteli kodlamaya sahip olan bir kromozom χ^l farklı bireyi gösterecektir [49].

Gerçek Sayılı Kodlama:

Bu kodlamada tüm gerçek sayılar kromozomun allelleriyle atanabilmektedir. Yani kardinalitesi tüm reel sayılar olan bir küme kullanılacaktır. Öyle ise l uzunluğundaki bir kromozom $\mathfrak{R}^l$ farklı bireyi gösterebilecektir. Bu tür kodlamanın kullanıldığı GA modeli mutasyon bazlıdır, çaprazlama bazlı modellerde daha çok ikili gösterim kullanılmaktadır. Kromozomların ikili olarak gösterilmesi GSP için ve diğer kombinatorik problemler için uygun değildir. Son yıllarda GSP için çok sayıda kodlama ortaya konulmuştur [49].

Permütasyon Kodlama:

Bunlardan biri permütasyon kodlamadır. Talep noktalarının satıcı tarafından ziyaret edilme sırasına göre dizildiği kodlamadır. Küçük bir örnek için gösterecek olursak 9 talep noktası olan bir GSP için uğranılacak müşteriler sırasıyla 3-2-5-4-7-1-6-9-8 ise kodlama da [3 2 5 4 7 1 6 9 8] şeklinde olacaktır [50]. Fakat bu gösterimde dikkat edilmesi gereken husus çaprazlamanın talep noktası tekrarından kaçınacak şekilde yapılmasıdır.

Rassal Sayılı Kodlama:

İlk olarak Bean tarafından çizelgeleme ve en iyileme problemleri için rassal sayı kodlamalı genetik algoritma yaklaşımı literatüre kazandırılmıştır [51]. (0,1) aralığındaki rassal sayıların kullanıldığı bu kodlamada önce her allel için atanan sözkonusu rassal sayılar sıralandıktan sonra permütasyon gösterimi elde edilmektedir. Yukarıda GSP için verilmiş örnek bu kodlama ile şu şekilde yapılır:

[0.23 0.82 0.45 0.74 0.87 0.11 0.56 0.69 0.78]

Her bir hücredeki rassal sayı bir talep noktasını temsil etmektedir ve artan sıraya dizildiklerinde şu dizilim elde edilecektir:

6 – 1 – 3 – 7 – 8 – 4 – 9 – 2 – 5

Bu kodlama ile çaprazlama sonrasında ortaya çıkabilecek olursuz çocukların önüne geçilebilmektedir. Sözkonusu kodlama makina çizelgeleme, kaynak atama, araç rotalama, kuadratik atama problemi gibi çok sayıda çizelgeleme probleminde

kullanılmaktadır [50]. Bu çalışmada da kombinatorik bir problem olan Zaman Pencereli ARP, GA ile modellenmiştir ve rassal sayılı kodlama kullanılmıştır.

Bu kodlamaların dışında Düzensiz (Messy) Kodlama, Direk Kodlama [50] ve Harf Kodlama [52] gibi birey kodlamaları da mevcuttur

Aşama 2. Başlangıç popülasyonu oluşturma

Daha önce de belirtildiği gibi problemin yapısına göre belirlenecek olan boyutta, rassal çözümlerden bir başlangıç popülasyonu oluşturulur. Çözüm uzayının farklı bölgelerinden olabilecek bu küme sayesinde, diğer meta-sezgisel yöntemlerden farklı olarak, farklı bölgelerde aynı anda arama yapılabilecektir.

Aşama 3. Uygunluk değerlendirme (evaluation of fitness) fonksiyonu

Uygunluk değerlendirme fonksiyonu, GA'da problem ile algoritma arasında bağlantı sağlayan bir işlemdir. Bu fonksiyon bireylerin popülasyon içerisindeki değerini uygunluk değerlerine göre belirler: daha iyi bireyler hayatta daha büyük şansla kalacaktır ve yeniden üreyecektir. Dolayısıyla uygunluk değerlendirme fonksiyonu tanımlamak çok önemlidir. Olurlu olmayan çözümlerin bu fonksiyon sayesinde dikkate alınmaması gerekmektedir [44]. En iyileme probleminde tanımlanacak amaç fonksiyonu bu görevi görecektir. Kodlanmış birey algoritma sırasında deşifre edilerek belirlenen amaç fonksiyonuna sokulur ve amaç fonksiyon değeri elde edilir.

Aşama 4. Seçim operatörü

Seçim operatörü GA'da yeniden üreme işleminin en önemli kısmını oluşturur. Bu operatör ile algoritmadaki birey çeşitliliği artacak, böylece çözüm uzayında farklı bölgeler aranabilecektir [50]. Çok sayıda seçim yöntemi sunulmuştur, aşağıda kısaca bahsedilecektir.

Rulet Çemberi Seçim Yöntemi:

Uygunlukla Orantılı Seçim yöntemlerinde biri olan bu yöntem en sık kullanılan seçim yöntemidir. Bireyler uygunluk değerlerine göre belirli olasılık değerlerine sahiptirler. Bu olasılıklar, her bir bireyin uygunluk değerinin toplam uygunluk değerine oranlanması ile elde edilir. Bu çalışmada bu seçim yöntemi kullanılacağından adımları anlatılacaktır:

Toplam popülasyon büyüklüğü m olsun ve en büyükleme problemi olsun.

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

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

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

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

Adım 5. Popülasyondaki birey sayısı kadar (0,1) aralığında rassal sayı atılır ve her rassal sayıdan bir sonra gelen birikimli olasılık değerine karşılık gelen birey seçilir [30,50].

Sıralı Seçim Yöntemi:

Uygunlukla Orantılı Seçim'den uyarlanmış bir başka seçim yöntemidir. Popülasyondaki bireyler uygunluk değerlerine göre sıralanır daha sonra bu sıralamaya göre seçim yapılır. Rulet çemberi seçimi eğer uyumluluk çok fazla değişiyorsa sorun çıkartabilir. Örneğin en iyi kromozomun uyumluluğu %90 ise diğer kromozomların seçilme şansı azalacaktır. Bunu önlemek için sıralı seçim kullanılabilir. Sıralama ile seçilme olasılığı arasındaki ilişki keyfi olarak yapılır toplam olasılık 1 olacak şekilde doğrusal ya da üssel olarak azalan şekilde olabilir. [30].

Turnuva Seçimi:

Popülasyon büyüklüğü çok fazla olduğunda yukarıda bahsedilen yöntemler çok fazla zaman gerektirecektir. Bunu engellemek için Turnuva seçimi kullanılır. Bu yöntemde popülasyon içerisinde belirli sayıda (k adet) birey alınır ve bu bireylerden uygunluk değeri en yüksek olan ebeveyn olarak seçilir. Bu işlem popülasyon büyüklüğü kadar tekrarlanır. Bu k değerinin büyük olması yüksek uygunluktaki bireylerin ebeveynleri oluşturmaya, küçük olması ise kötü bireylerin çok fazla yeniden üreme için seçilmesine neden olacaktır [30].

Elitizm:

Elitizm ile en iyi bireylerden belirli miktarının her nesilde kalması sağlanır. Yani, her yeni nesil üretilirken bir önceki nesildeki en iyi birey(ler) hiç değiştirilmeden bir

sonraki nesile aktarılır. Eğer bu yapılmazsa iyi bireyler bir sonraki nesile geçerken çaprazlama için seçilemeyebilir, ya da çaprazlama ve mutasyon sırasında yok olabilirler. Bir çok araştırmacı elitizmin GA'nın performansını arttırdığını ifade etmişlerdir [46].

Aşama 5. Yeniden Üreme (Reproduction)

Seçilen bireylerin (ebeveynlerin) bir sonraki nesli (yeni çözümleri) üretmesi için uygulanan temel genetik işlemler çaprazlama ve mutasyondur [22, 50].

Aşama 5.1 Çaprazlama

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal populasyonlardaki çaprazlamaya karşılık gelmektedir. GA'da en önemli işlemlerden biri ve genetik işlem olan çaprazlama sonrasında elde edilen çocuklar ebeveynlerin genetik özelliklerini taşır. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldığı için farklı çaprazlama yöntemleri kullanılır, tek noktalı çaprazlama, iki noktalı çaprazlama, ve düzenli (üniform) çaprazlama gibi. En basit çaprazlama yöntemi tek noktalı çaprazlama yöntemidir. Tek noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir [53]. Amaç, ebeveyn kromozom genlerinin yerini değiştirerek çocuk kromozomlar üretmek ve böylece varolan uygunluk değeri daha yüksek olan kromozomlar elde etmektedir.

Tek noktalı çaprazlama:

n noktalı çaprazlamanın $n=1$ olduğu bu çaprazlamada çocuk üretmek için seçilen ebeveynler bir tek noktadan kesilirler ve bu parçalar takas edilir.

Ebeveyn 1 **11000110**

Ebeveyn 2 01100111

Eğer kesme noktası soldan 4. hücreden sonra ise çocuklar şu şekilde olacaktır:

Çocuk 1 **11000111**

Çocuk 2 01100**110**

İki noktalı çaprazlama:

Bu çaprazlamada ise belirlenen kesme noktası 2 tanedir. Bir önceki örnekteki ebeveynlerde 2. ve 6. allelden sonra çaprazlama yapılırsa çocuklar şu şekilde olacaktır;

Çocuk 1 **01100110**

Çocuk 2 **01000111**

Düzenli (uniform) çaprazlama:

Bu çaprazlamada ise elde edilcek çocuğun hangi allelini hangi ebeveynden alacağı ebeveynler ile aynı boyutta olan ve hücreleri Bernolli dağılımına göre stokastik atanan ikili bir maskeye göre belirlenir. Eğer maskenin bir hücresinde 1 varsa çocuğun ilgili aleli birinci ebeveynden, 0 ise ikinci ebeveynden gelecektir [48]. Eğer 2 tane çocuk üretilecekse ikinci çocuk için tersi geçerlidir. Yine aynı ebeveynler için uygularsak, eğer maske aşağıdaki gibi üretilmişse, çocuklar şu şekilde olacaktır;

Maske 11001010

Çocuk 1 **11100111**

Çocuk 2 **01000110**

Bu çaprazlama yöntemlerinin yanısıra permütasyon kodlama için kullanılan çok sayıda yöntem vardır. Sözkonusu yöntemlerde, ikili kodlamadaki çaprazlama tekniklerinin uygulanması ile ortaya çıkabilecek olursuz çocukların ortaya çıkmasına engel olacak geliştirmelerde mevcuttur. Örneğin permütasyon kodlamalı iki ebeveyn tek noktalı çaprazlamaya tabi tutulursa, çaprazlama sonrasında çocuklarda aynı noktaların tekrarı durumu ortaya çıkabilir. Yani;

Ebeveyn 1 **1 2 3 4 5 6 7 8 9**

Ebeveyn 2 **3 5 2 8 1 4 7 9 6**

Ve sabit nokta 5. allel ise, ortaya şu çocuklar çıkacaktır:

Çocuk 1 **1 2 3 4 5 4 7 9 6**

Çocuk 2 **3 5 2 8 1 6 7 8 9**

Çocuk 1’de 4 numaralı talep noktasının iki kez yer aldığını, 8 numaralı talep noktasının ise yer almadığını görürüz. Bu durum iki kez tekrar edilen noktaya iki kez uğranacak ve diğerine ise hiç uğranmayacak olan olursuz bir çözüme sebep olacaktır.

Kısmi Planlı Çaprazlama (PMX):

Bu çaprazlama, ikili kodlama için kullanılan iki noktalı çaprazlamanın permütasyon kodlaması için olanıdır. Özel bir onarım aşamasına sahip olan çaprazlama şu şekilde gerçekleştirilir [50] :

Adım 1. İki tane çaprazlama noktası rassal olarak seçilir ve bu noktaların ayırdığı alt dizilere eşleştirme bölgesi denir.

Adım 2. Alt dizilerden her biri bir ebeveyn den olmak üzere ikisi takas edilir ve yarı-çocuklar elde edilir.

Adım 3. Eşleştirme bölgeleri arasındaki eşleştirme ilişkisi belirlenir.

Adım 4. Söz konusu ilişkiye göre çocuklar olurlu hale getirilir.

Küçük bir örnek üzerinde gösterecek olursak:

Ebeveyn 1 1 2 3 4 5 6 7 8 9

Ebeveyn 2 5 4 6 9 2 1 7 8 3

Adım 1. Alt dizilerin seçilmesi: koyu olan bölgeler ve açık olan bölgeler alt dizilerdir.

Ebeveyn 1 1 2 **3 4 5 6** 7 8 9

Ebeveyn 2 5 4 **6 9 2 1** 7 8 3

Adım 2. Alt dizilerin takas edilmesi:

Yarı-çocuk 1 1 2 **6 9 2 1** 7 8 9

Yarı-çocuk 2 5 4 **3 4 5 6** 7 8 3

Adım 3. Eşleştirme ilişkisinin kurulması:

6 9 2 1
↕ ↕ ↕ ↕
3 4 5 6

$1 \leftrightarrow 6 \leftrightarrow 3$, $9 \leftrightarrow 4$, $2 \leftrightarrow 5$

Adım 4. Eşleştirme ilişkisi ile yarı-çocukları olurlu hale getirme:

Çocuk 1 3 5 **6 9 2 1** 7 8 4

Çocuk 2 2 9 3 4 5 6 7 8 1

Bu çaprazlamayı dışında Sıralı Çaprazlama (OX), Pozisyon Bazlı Çaprazlama, Dairesel Çaprazlama (CX) gibi çaprazlamalar mevcuttur, ayrıca sezgisel olarak geliştirilmiş yöntemler de vardır [50].

Aşama 5.2 Mutasyon (dönüşüm) operatörü

GA, diğer yapay zeka sezgiselleri olan TB ve TA'ya benzer şekilde, yerel en iyi çözüme yakalanmaktan genetik işlemlerden biri olan mutasyon sayesinde kurtulabilmektedir [4]. Yaygın olarak kullanılan mutasyon işlemi, arama uzayının çaprazlama işlemi ile ulaşılamayan bölgelerine ulaşmayı sağlamaktadır [22]. Böylece, arama uzayının alt-en iyi (sub-optimal) bölgelerinden kaçmayı sağlayarak, popülasyon çeşitliliğini makul bir seviyede tutulmasını sağlayan bir fonksiyondur. Amaç, varolan bir kromozomun genlerinin bir ya da birkaçının yerlerini değiştirerek yeni kromozom oluşturmaktır. Yeniden ve sürekli yeni nesil üretimi sonucunda belirli bir süre sonra nesildeki kromozomlar birbirini tekrarlama konumuna gelebilir ve bunun sonucunda farklı kromozom üretimi durabilir veya çok azalabilir. İşte bu sebeple nesildeki kromozomların çeşitliliğini arttırmak için kromozomlardan bazıları mutasyona uğratılır [53]. Mutasyonun bu çeşitliği sağlamada ikinci derecede etkili olduğunu savunan araştırmacılar mevcuttur, bunun yanı sıra farklı bir yaklaşım da bu çeşitlilikteki esas rolü mutasyonun oynadığını savunur. Belki en doğrusu, çaprazlama ve mutasyon arasında probleme özgü bir dengenin sağlanması gerektiğini söylemektir [48]. Küçük bir örnek üzerinde mutasyonun ne yaptığına bakacak olursak: 0 1 1 0 0 1 ikili kodlamalı kromozom üzerindeki her bir allel küçük bir olasılığa bağlı olarak mutasyona uğrasın. Eğer 3. ve 5. allelin mutasyona uğratılırsa yeni kromozom 0 1 0 0 1 1 elde edilecektir. Mutasyona uğrayacak allellerdeki değerler değişmektedir [22].

Yukarıda örnek olarak verilen Basit Mutasyon dışında, özellikle permütasyon kodlamalı kromozomlarda kullanılan Ters Çevirme Mutasyonu, Ekleme Mutasyonu, Yerdeğiştirme Mutasyonu, Karşılıklı Değişim Mutasyonu ve Sezgisel Mutasyon mevcuttur [50].

Aşama 6. Bitirme Koşulu

Algoritmada, yeniden üreme aşamasından sonra eğer belirlenen bitirme koşulu sağlanıyorsa elde edilen popülasyondaki en iyi birey algoritma sonucundaki en iyi

özüm olacaktır. Bu bitirme koşulu; belirli bir iterasyon sayısı, belirli bir eşik değeri ya da belirli sayıda iterasyon sonucunda değışme gerçekleşmiyorsa durdurmak olabilir.

4.4 Genetik Algoritmalar'ın Kullanıldığı Problemler

Son yıllarda, genetik algoritmaların karmaşık problemlerin en iyilenmesinde kullanılabilecek etkili teknikler olması nedeniyle önemleri hızla artmakta ve endüstri mühendisliği alanında başarıyla uygulanmaktadır [50]. GA'lar klasik sezgisel yöntemler ve bazı meta-sezgisel yöntemler gibi (TA, TB) tek noktadan arama yapmaz, aksine çok fazla çözümün olduğu bir popülasyondan başlayarak arama yapmaktadır [54]. Çözüm uzayının daha çeşitli bölgelerinde arama yapıyor olması diğer yöntemlerden üstün olan taraflarından biridir. En iyi bilinen uygulama alanları; çizelgeleme ve sıralama, güvenilirlik tasarımı, araç rotalama ve çizelgeleme, grup teknolojisi, tesis tasarımı ve yerleşimi, ulaştırma ve diğerleridir [50].

1954 yılında Johnson tarafından yapılan çalışma sonrasında dikkatleri üzerine çeken ve diğer araştırmacıların da üzerinde durduğu bir konu haline gelen ve çizelgeleme problemlerinin en çok bilinen sınıfı olan Akış Tipi İş Sıralama (Flow-Shop Sequencing) problemi GA'nın da etkili bir şekilde kullanıldığı problemidir. Kısaca bu problemde, m tane makina ve n tane iş vardır, her iş farklı makinalarda gerçekleşmek üzere m farklı işlem içermektedir ve her işin her makinada işlem süresinin farklıdır. Her iş aynı işlem sırasıyla m makinada işlem görecektir. Amaç toplam üretim süresini en azlayacak şekilde bu işleri sıralamaktır [50]. Genetik algoritmalar akış tipi iş sıralama problemlerinin çözümünde başarılı bir şekilde kullanılmaktadır. Reeves yılında ortaya koyduğu çalışmasında bu problem için bir genetik algoritma modeli geliştirmiştir ve TB yöntemi ile GA'nın bu problem çözümünde rekabet edebileceğini belirtmiştir. Ishibuchi, Yamamoto, Murata, ve Tanaka bulanık akış tipi iş çizelgeleme problemi için GA ve En Yakın Komşu Algoritması'nı birlikte kullanmışlardır. Geliştirdikleri melez yöntem sözkonusu problemin çözümünde etkilidir [50]. Son yıllarda bu problem sınıfı için yapılmış olan çalışmalardan bir kaçına bakacak olursak; Iyer ve Saxena tüm işlerin aynı işlem sırasına sahip olduğu permütasyon Akış Tipi İş Sıralama problemi için geliştirdikleri modelde standard GA'dan farklı olarak, çaprazlama operatörünü probleme göre değiştirerek kullanmışlardır. Toplam üretim süresinin en azlanmaya çalışıldığı modelde,

çaprazlama yapılacak iki ebeveynde yer alan ortak iş sıraları korunarak geride kalan işler belirli bir stratejiye göre takas edilir. Ortak iş sırası, birbirlerini en az şekilde bekletecek işlerin oluşturduğu sıradır. Çaprazlama sonrasında elde edilen çocuklardan bazılarına (düşük bir oranla) 2 allelinin yer değiştirmesi şeklindeki mutasyon uygulanmıştır [55]. Ruiz ve diğ. yaptıkları çalışmada yine aynı problem için iki tane GA geliştirmişlerdir. Permütasyon kodlamanın kullanıldığı, yani her kromozomun bir iş sırasını temsil ettiği, algoritmalarda başlangıç popülasyonu rassal olarak değil erken yakınsamanın önüne geçilmesi için bir sezgisel ile üretilmiştir. Sözkonusu sezgisele göre en uzun işlem süresine sahip işler erken çizelgelenir. Melezleştirme için ayrıca yerel aramadan da faydalanmışlardır. Genetik operatörler olarak, sıralama ve turnuva seçim yöntemi, çalışmada önerdikleri 4 yeni çaprazlama yöntemi ve her allel için düzgün dağılımda bir olasılıkla mutasyon gerçekleştirmişlerdir. Sözkonusu yöntemler bu problem türü için etkilidir [56]. 2008 yılında yine aynı problem için geliştirilen çalışmada ise Yapısal GA kullanılmıştır. Klasik GA'dan farklı yanı, popülasyonun dinamik boyutta yapı taslaklarından oluşmasıdır. Yapısal gösterim diğer çalışmada olduğu gibi tüm işlerin olduğu gösterimdir fakat taslak gösterimde bu işlerden bazıları yer alır, boş olan sıralarda ise taslak olduğunu ifade eden işaretler mevcuttur. Geliştirilen yöntem bu popülasyondaki bazı işlerden oluşan taslakları muhafaza eder. Bu çalışmada da bir öncekinde kullanılan sezgiselden ve yerel aramadan faydalanılmıştır [57]. Yukarıda kısaca bahsedilen yöntemlerin ortak noktası kodlamanın permütasyon şeklinde yapılmış olması ve toplam üretim süresinin en azlanmaya çalışılmasıdır.

Makina çizelgeleme problemlerinden en çok bilineni klasik Atölye Tipi Çizelgeleme problemleridir. Bu problemde işler makinalarda farklı sıralarda işlem görebilmektedirler. Bu problemde GA'nın uygulanmasında üzerinde durulması gereken en önemli noktalardan biri çözümün nasıl kodlanacağıdır. Çözüm kodlamasının diğer bir kombinatorik problem olan GSP gibi kolay olmadığı bu problemde, GA kullanan araştırmacılar olursuz çözüm sorununun önüne geçmek için onarma stratejileri kullanmaktadırlar. Direkt ve indirek kodlamanın kullanıldığı bu problemde, direkt kodlama ile çizelgenin kendisi kromozom olarak kodlanmaktadır ve bu kodlama üzerinden iyileştirmeler gerçekleştirilmektedir. İşlem temelli, iş temelli, tamamlanma süresi temelli kodlamalar ve rassal sayılı kodlamalar bu

yaklaşımında kullanılmaktadır. İndirek kodlamalarda ise iş ataması için hareket kurallarının sıralaması kodlanmaktadır [50].

4.5 Genetik Algoritma ve ARP Arasındaki İlişki

Genetik Algoritmalar, ARP'nin belirli çeşitlerini de içeren bir çok kombinatorik en iyileme probleminin çözümünde kullanılmaktadır. Özellikle zaman penceresi kısıtının olduğu ARP için GA ile yapılmış çalışmalar bulunmaktadır [58]. Genelde alt sezgisel yöntemlerle melezleştirilerek kullanılan GA yöntemlerinde, başlangıç popülasyonun rassal değil iyi çözümlerden oluşması için sözkonusu alt sezgisellerden faydalanılmaktadır.

Literatürde yer alan yakın zamanlı çalışmalara kısaca bakacak olursak; Baker ve Ayechev 2003 yılında yaptıkları çalışmalarında, klasik ARP çözümü için GA kullanmışlardır. Başlangıç popülasyonunu oluştururken SA ve Genelleştirilmiş Atama Algoritması'ndan faydalanmışlar, rassal atamayla başlangıç popülasyonu elde ettikleri denemelerinden daha iyi sonuç elde ettiklerini göstermişlerdir. İkili turnuva seçim operatörü ile seçtikleri bireylere iki noktalı çaprazlama yöntemi uygulamışlar ve elde ettikleri çocuklara birey içerisinde rassal seçilmiş iki allelin yer değiştirdiği basit mutasyon uygulamışlardır. Geliştirdikleri GA yöntemini en yakın komşu arama yaklaşımı ile melezleştirdikleri ikinci algorithmada çözüm kalitesi açısından çok anlamlı sonuçlar elde etmemişler fakat işlemci kullanım süresinde kayda değer bir iyileşme elde etmişlerdir [58]. Bir başka çalışmada ise, farklı kapasitelerde ve maliyetlerde (sabit ve değişken maliyet) araçlardan oluşan bir heterojen araç filosunun sözkonusu olduğu ve hangi araçtan ne kadar kullanılması gerektiğine de cevap aranan ARP'nin farklı bir versiyonu için GA'dan yararlanmışlardır. Çözüm gösterimi için permütasyon kodlamadan faydalanılmıştır. Başlangıç popülasyonunun bir kısmını oluşturmak için Kapasiteli ARP için önerilmiş olan Tasarruf algoritmasından ve SA'dan faydalanılmıştır, geri kalan kromozomlar rassal oluşturulmuştur. Genetik operatörler olarak, Turnuva seçim yöntemi, Sıralı çaprazlama (OX) ve çaprazlama için seçilen ebeveynlerin aynı uygunluk değerine sahip olma sıklığı artmaya başladığı yani popülasyonun iyi bir noktaya yakınsamaya başladığı zaman Tek Ebeveynli Çaprazlama kullanılmıştır. Bazı çocuklara uygulanan mutasyon işleminde yerel aramadan faydalanılmıştır [59]. Wang ve Lu Kapasiteli ARP için yapılan bir başka çalışmada da başlangıç popülasyonu oluşturmak için SA

ve En Yakın Ekleme Yöntemi'nin birleşimi kullanılmış, sistematik olarak yapılmış denemelerden faydalanarak Yanıt Yüzey Yöntemi (Response Surface Methodology) ile çaprazlama ve mutasyon oranlarını en iyilenmiştir. Ayrıca mutasyon sırasında yine SA'dan faydalanarak popülasyondaki çeşitlilik korunmaktadır [60].

Berger ve Barkaoui Zaman Pencere ARAP için iki melez GA'nın birlikte çalıştığı bir model geliştirmişlerdir. Aynı anda çalışan GA'lar farklı amaç fonksiyonlarına sahiptir; ilk algoritmada toplam mesafe en azlanmaya çalışılırken ikincisinde ise olurlu çözüm elde etmek için kısıt ihlali en azlanmaya çalışılır [61]. Çok Depolu ARAP için geliştirilen bir başka çalışmada ise 2 tane melez genetik algoritma modeli ortaya konulmuştur. HGA1 ve HGA2 olarak adlandırılan modellerde çözümler permütasyon gösterimden farklı olarak, depo noktasında her rota için yer aldığı rota gösterimi ile belirtilmiştir. Önce talep noktalarının hangi depodan, daha sonra hangi araç tarafından ve hangi sırada servis göreceğinin belirlendiği sırasıyla gruplama, rotalama ve çizelgeleme adımlarıyla başlangıç çözümlerinin elde edildiği bu iki yöntem arasındaki fark HGA1'de tüm adımlarda çözümler rassal olarak belirlenir. HGA2'de ise gruplamada talep noktası arasındaki mesafeye göre gruplama, Clarke ve Wright Tasarruf Algoritması'na göre rotalama, en yakın komşu sezgiseline göre ise çizelgeleme gerçekleştirilir. Yöntemlerin her ikisinde de başlangıç çözümler elde edildikten sonra, genetik algoritma işlemleri öncesinde Yinelemeli Takas Yöntemi (Iterated Swap Procedure) kullanılarak rota içi, depo içi, depolar arası, rotalar arası iyileştirmeler yapılmaktadır. Rulet çemberi seçim yöntemi, sıralı çaprazlama, sezgisel ve ekleme mutasyonları kullanılmıştır. HGA2 toplam teslimat süresi açısından daha iyi sonuç vermektedir [4].

Li ve Zhang tersine lojistikte ortaya çıkabilecek planlama problemlerinden birinin araç rotalama olduğunu belirtmişler ve Eş Zamanlı Topla-Dağıt ARAP'nin burada kullanılabileceğinden yola çıkarak sözkonusu problemin çözümü için GA ile bir model geliştirmişlerdir. Permütasyon kodlama kullanarak elde edilen bireylerden rassal ve SA ile başlangıç popülasyonu elde etmişler ve her iki durum için algoritmanın performansını ölçmüşlerdir. Seçim için rulet çemberi ve elitizm kullanılmıştır. Çaprazlama olarak yeni bir yaklaşım ortaya koymuşlar: çaprazlama için seçilen iki ebeveynin her birinde iki tane ayırma noktası belirlenir. Bu noktaların arasında kalan alt kromozomlar, karşılıklı olarak ebeveynlerin baş kısmına atanır. Geri kalan dizide tekrarlama yapan noktalar elenir. Elde edilen çocuklara basit

mutasyon uygulanır [62]. Wang ve diğ. Zaman Pencereli ARP için geliştirdikleri çalışmada, bir önceki çalışmada olduğu gibi GA'nın erken yakınsamasının önüne geçmek için yeni bir çaprazlama yaklaşımı ortaya koymuşlardır: Seçtikleri iki kromozomda iki nokta belirlenir, yine arada kalan alt kromozomlar takas edilir. Burada ise, ikinci kromozomun alt kromozomu ilkinin başına, ilkinin alt kromozomu ise ikincisinin sonuna gelir. Daha sonra ebeveynin orjinal kromozomunda tekrarlanan noktalar elenir. Bu çaprazlama ile eğer iki tane aynı ebeveyn çifleşecekse ortaya farklı çocuklar çıkacaktır, böylece algoritmanın adımlarında birbiriyle aynı ya da çok benzer olan kromozomların olmasının önüne geçilir [63].

Alvarenga ve diğ. genetik algoritma ve küme bölme yaklaşımını bir arada kullandıkları çalışmalarında, yine Zaman Pencereli ARP problemi için toplam mesafenin en küçüklenmesini temel amaç alan melez bir sezgisel model ortaya koymuşlardır. Problemin herhangi bir yerel en iyi çözümünde yer alacak rotaların çoğunluğunun büyük olasılıkla genel en iyi çözümde de olacağını öne süren araştırmacılar, küme bölme probleminde kullanılacak rotalar kümesini birbirinden farklı yerel en iyi çözümlerden faydalanarak elde etmişlerdir. Genetik algoritma kullanarak birbirinden bağımsız koşumlar gerçekleştirilir, elde ettikleri çözüm kümesine evrim adası adı verilir. Daha sonra bu adaların kombinasyonları oluşturularak, küme bölme için en iyi kalitede çözüm elde edilmesi için rota kümesi elde edilir. Dolayısıyla, küme bölme yöntemini uygulamak için daha küçük fakat iyi çözümlerden elde edilen ve iyi rotaları kapsayan rotalar kümesi elde edilmiştir [64]. Araç sayısının sınırlı sayıda olduğu Zaman Pencereli ARP problemi için ise yine GA'nın kullanıldığı bir başka çalışmada, çözüm gösterimi olarak talep noktasının hizmet göreceği araç ve hangi sırada göreceğini belirten sıra numarasının olduğu bir kodlama kullanılmıştır. Sınırlı sayıda araç olması tüm müşterilerin hizmet görmemesine sebep olabileceğinden kodlamada sıra numarası olarak 0 ilave edilerek bu durum da modele ilave edilmiştir. Rulet çemberi ve elitizimin birlikte kullanıldığı seçim yöntemi, tek ve iki noktalı çaprazlama kullanılmıştır. İki bilgiyi birden taşıyan allellere küçük olasılıkla mutasyon uygulanmıştır [65]. Ma ve diğ. Zaman Pencereli ARP için diğer bir biyolojik olgudan faydalanmışlardır. Antikor üreterek antijenlere karşı vücudu koruma mekanizması olan bağışıklık sisteminden esinlenerek geliştirdikleri modelde, bu yaklaşımın GA'ya eklenmesinin popülasyondaki çeşitliliği arttırdığını belirtmişlerdir. Küçük boyutta bir örnek

üzerinde basit GA ile bu modeli karşılaştırmışlardır. Birey gösterimini doğal sayı kullanarak ve başlangıç popülasyonunu da rassal gerçekleştirmişlerdir. Rulet çemberi seçim yöntemi ve elitizim kullanarak seçtikleri ebeveynlerden Sıralı Çaprazlama ile çocuk elde etmişler ve takas yöntemi ile mutasyon gerçekleştirmişlerdir [66].

Yukarıda kısaca bahsedilen çalışmalarda gözlenen ortak noktalar, birey kodlamasının genelde permütasyon kodlama kullanılarak yapılmasıdır. Başlangıç çözümler ya rassal elde edilmiş ya da melez çalışmalarda alt sezgisel yöntemlerden faydalanılarak iyi çözümleri içeren bir başlangıç popülasyonu edilmiştir. SA, tasarruf algoritması, ekleme algoritmaları sık kullanılan alt sezgisel yöntemlerdendir. Çaprazlama sırasında ortaya çıkabilecek olursuz çözümlerin önüne geçmek için ya 4.3'te belirtilen bu kodlama yapısına uygun çaprazlama yöntemleri ya da araştırmacıların kendi geliştirdikleri yöntemler kullanılmıştır. Seçim yöntemi olarak çoğunlukla rulet çemberi seçim yöntemi tercih edilmiş ve çoğu modelde en iyi çözümün bir sonraki çözümde de varlığını koruyan elitizmden faydalanılmıştır. Mutasyon için ise genelde düşük oranlarda basit mutasyon, yani rassal olarak seçilmiş iki allelin yer değiştirmesi, uygulanmıştır. Çizelge 4.2'de bu çalışmaların kullandıkları operatörler çeşitlerini içeren bir özet sunulmuştur. Bu çalışmada da yukarıdaki çalışmalara paralel olan yanlar mevcuttur; başlangıç popülasyonu rassal olarak belirlenmiş, seçim yöntemi olarak rulet çemberi kullanılmış ve elitizmden faydalanılmıştır. Çalışmanın farklılığı birey gösteriminde ortaya çıkmaktadır. Rassal sayı kodlamalı gösterimin kullanıldığı bu çalışma literatürde ARP için yapılmış olan ve GA kullanan diğer çalışmalardan ayrılmaktadır. Bu gösterim nedeniyle çaprazlamada ve mutasyonda diğerlerinden farklı olarak sırasıyla düzenli (uniform) çaprazlama ve rassal olarak seçilmiş bir allelin rassal sayı değeri yerine yeni bir rassal sayının atandığı mutasyon uygulanmıştır.

Çizelge 4.2 : Genetik Algoritma ile yapılmış ARP çalışmaları

Yazarlar	Yıl	ARP çeşidi	Kodlama	Başlangıç Popülasyonu	Seçim	Çaprazlama	Mutasyon
Barrie M. Baker & M.A. Ayechev [58]	2003	ARP	İkili olmayan	*süpürme algoritması *genelleştirilmiş atama algoritması	*ikili turnuva	*iki noktalı çaprazlama	* basit mutasyon (takas)
Shuguang Liu ve diğ. [59]	2009	Heterojen Filolu ARP	Permütasyon	* tasarruf algoritması * süpürme algoritması	* turnuva	*sıralı çaprazlama *tek ebeveynli çaprazlama	* yerel arama
Chung-Ho Wang & Jiu-Zhang Lu [60]	2009	Kapasiteli ARP	Permütasyon	*en yakın ekleme yöntemi * süpürme algoritması	*rulet çemberi *elitizm	* doğrusal sıralı çaprazlama	* basit mutasyon (takas)
Jean Berger & Mohamed Barkaoui [61]	2004	Zaman Pencereli ARP	açık çözüm kodlamalı	* sıralı ekleme sezgiseli *rota-komşu yapısı	*rulet çemberi	* ekleme sezgiseli *geniş komşu arama *rota komşu temelli iki aşamalı meta-sezgisel	6 farklı yöntem
William Ho ve diğ. [4]	2008	Çok Depolu ARP	permütasyon (rotalı)	*HGA1 için: rassal *HGA2 için: *tekrarlı takas *Clarke veWright *en yakın komşu	* rulet çemberi	*sıralı çaprazlama	*ekleme *sezgisel mutasyon
Li Jun & Zhang Jian-yong [62]	2007	Eş Zamanlı Topla-Dağıt ARP	permütasyon	*rassal *süpürme algoritması	*rulet çemberi *elitizm	* yeni bir çaprazlama (çalışmaya özgü)	* basit mutasyon (takas)
Wenfeng Wang ve diğ. [63]	2008	Zaman Pencereli ARP	permütasyon	*rassal	*rulet çemberi *elitizm	* yeni bir çaprazlama (çalışmaya özgü)	* basit mutasyon (takas)
G.B. Alvarenga ve diğ. [64]	2007	Zaman Pencereli ARP	Permütasyon	*ilerletici ekleme sezgisel yöntemi (Push-Forward Insertion Heuristic)	*k-noktalı turnuva	* yeni bir çaprazlama (çalışmaya özgü) *elitizm	8 farklı yöntem
Wang Xu-ping ve diğ. [65]	2008	Sınırlı Sayıda Araç ve Zaman Pencereli ARP	permütasyon (rotalı)	*müşteri kümesi atama algoritması *rota yapısı sezgisel algoritması	*rulet çemberi *elitizm	*tek noktalı *iki noktalı	* basit mutasyon (takas)
Jia Ma ve diğ. [66]	2006	Zaman Pencereli ARP	permütasyon (rotalı)	*rassal	*rulet çemberi *elitizm	*sıralı çaprazlama	* basit mutasyon (takas)

4.6 Rassal Sayı Kodlamalı Genetik Algoritma ve Kombinatorik En İyileme Problemleri ile Arasındaki İlişki

GA'nın bir en iyileme probleminin çözümünde kullanılabilmesi için ilk olarak probleme uygun birey kodlamasının nasıl yapılacağına karar verilmelidir. GA denildiğinde akla ilk gelen kodlama ikili kodlamadır. Fakat kombinatorik en iyileme problemlerinde ikili kodlama kullanmak pek mümkün değildir. Genelde permütasyon kodlamanın yapıldığı bu problemlerde çaprazlama sırasında ortaya çıkabilecek sorunları gidermek için bu aşamada temel çaprazlama yöntemlerine (tek noktalı, iki noktalı) onarıcı işlemlerin eklendiği yöntemler sözkonusudur. Rassal sayı kodlamada ise bahsedilen sorunlarla karşılaşılmaz, böylece onarıcı işlemlere de gerek kalmaz [54].

Daha önce de bahsedildiği gibi, rassal sayı kodlama Bean tarafından 1994 yılında literatüre kazandırılmıştır. Araştırmacı, tek ve çoklu makina çizelgeleme problemlerinde, ARP'de, Genelleştirilmiş GSP'de, kaynak tahsis problemlerinde ve kuadratik atama problemlerinde bu kodlamanın kullanılabileceğini belirtmiştir. Sözkonusu kodlamada sayı kodlamada her kromozomun her alleli (0,1) aralığında tekrarsız rassal sayıdan oluşmaktadır. Uygunluk fonksiyonunda kullanılmak üzere permütasyon hali ise bu rassal sayıların artan (ya da azalan) şekilde sıralanması ile elde edilir. Çaprazlama işlemi kromozomun allellerinin sıralanmadan önceki rassal sayılı haline uygulandığı için olurlu birey elde edileceği garantidir [51]. Hem fazla işlem yükünden kurtaran hem de olursuz çözüm elde etme riskini ortadan kaldıran bu yaklaşımı uygulayan çalışmalara rastlamak mümkündür.

Synder ve Daskin, Genelleştirilmiş GSP için rassal sayı kodlamalı GA kullanmışlar fakat kodlamada, Bean'in çalışmasında bu problem sınıfı için önerdiği gibi, biraz farklılık söz konusudur. Genelleştirilmiş GSP'de düğümler belirli sayıda alt kümelerle ayrılmıştır ve gezgin satıcının bu alt kümelerin her birinden bir tane düğüme uğraması gerekmektedir. Kullanılan kodlama kesirli sayıdan oluşmaktadır; tamsayı kısmı ilgili kümedeki düğüm sırasını, kesirli kısım ise artan sıraya göre sıralanacak kısmı göstermektedir [51, 54]. Samanlıoğlu ve diğ. Simetrik Çok Amaçlı GSP çözümü için yaptıkları çalışmada GA kullanmışlar ve birey gösterimi için rassal sayı kodlamadan faydalanmışlardır. Araştırmacılar (0,1) arasında rassal sayı yerine belirli bir A,B aralığında değişen tamsayılar kullanmış ve kod çözme

sırasındaki sıralamada azalan sıradan faydalanmışlardır. Çalışmalarındaki küçük bir örneği belirtecek olursak; 7 tane düğümün olduğu bir GSP problemi için $A=0$ ve $B=3000$ olmak üzere iki tane birey 2 12 15 29 30 45 52 ve 24 32 13 10 4 22 48 şeklinde gösterilebilir. Bu bireylerin sıralanmış halleri; 1 2 3 4 5 6 7 ve 5 4 3 6 1 2 7 şeklinde olacaktır [69]. Bu tez çalışmasında da aynı yöntem kullanılmıştır.

Gonçalves ve diğ. de Atölye Tipi Çizelgeleme problemi için geliştirdikleri melez GA yönteminde rassal sayılı kodlamayı kullanmışlardır. Bir başka çalışmada Kaynak Kısıtlı Proje Çizelgeleme Problemi için rassal sayı kodlamalı GA kullanan Mendes ve diğ., çaprazlama işleminde seçilen iki ebeveynin allellerini takas edip etmemeyi bir $(0,1)$ aralığında bir oranla belirlemişler. Kromozom büyüklüğü kadar, yani her allel için $(0,1)$ aralığında bir rassal sayı atamışlar ve eğer bu rassal sayı belirledikleri çaprazlama oranından büyükse takas gerçekleşmekte, eğer küçükse değişiklik yapılmamaktadır. Belirlenen oran ebeveynlerin allellerinin yüzde kaçını takas edilecek kısmen bunu verir. Bu çaprazlama ile 2 tane çocuk elde edilmektedir [70].

Literatürde yapılan çalışmalarda rassal sayı kodlamalı GA'nın herhangi bir ARP çözümü için kullanımına rastlanmamıştır. Bu çalışmada yeni bir yaklaşım olarak bu kodlamanın Zaman Pencereli ARP için uygunluğu incelenecektir.

5. PROBLEM TANIMI

Daha önce de ifade edildiği gibi, bu çalışmada Sıkı Zaman Pencereli ARP problemi için Rassal Sayı Kodlamalı GA ile bir model ortaya konulmaktadır. Bu problemin modellenmesinde uygulama verisi olarak Solomon'un 1987 yılında yaptığı bir çalışmada belirttiği [67] ve literatürde sıklıkla kullanılan veriler kullanılmaktadır [14, 28, 32, 33, 34, 37, 38]. Hepsi Zaman Pencereli ARP problemi olan ve çeşitli parametreleri ile birbirlerinden ayrılmış olan toplam 56 farklı uygulama verisi bulunmaktadır. Bu çalışmada bu uygulama verilerinden c101 olarak adlandırılmış olanı kullanılmıştır [68]. Zaman Pencereli ARP barındırdığı kısıtlar nedeniyle GSP ve ARP'den daha zordur [34].

Problemde bir depo ve 100 talep noktası (müşteri) bulunmaktadır. Müşteriler ve depoların konumları (x, y) koordinatları ile belirlenmiştir ve sabittir. Birbirlerine olan uzaklıkları öklid uzaklığı ile hesaplanır ve elde edilen değer mesafe birimdir, süre olarak kullanıldığında da zaman birim olarak alınır. Toplam 25 adet kapasiteli araç bulunmaktadır ve araçların kapasiteleri eşit ve 200'dür. Müşterilerin belirli miktarda talepleri vardır ve her müşterinin hizmete en erken başlanabileceği bir hazır olma zamanı ile en son başlanabileceği son tarihleri vardır. Tüm müşterilere yapılacak teslimatın süresi aynı ve 90 zaman birimidir.

Bu problemin çözümünde, müşterilere teslimatın başlayabileceği zaman aralığı kısıtı ve araçların kapasite kısıtı sağlanacak şekilde en az toplam mesafe ile tüm müşterilere hizmet verilmeye çalışılmaktadır. Her bir rotadaki aracın kat edeceği mesafelerin toplamı bize toplam mesafeyi vermektedir. Elde edilen çözüm ile şu soruların cevabı elde edilmektedir:

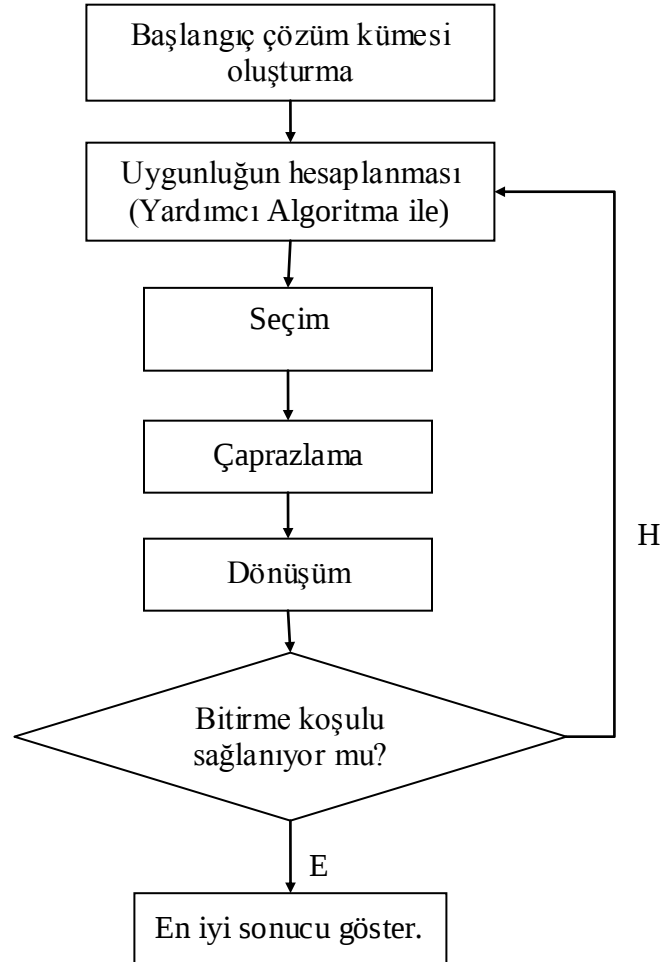
- Kaç adet araç (rota) mevcuttur?
- Hangi müşteriler, hangi araca (rotaya), hangi sırada atanmıştır?

Sözkonusu uygulama verileri Çizelge A1'de bulunmaktadır. Çizelgedeki ilk satırdaki müşteri numarası 0'dır, bu satır depoyu temsil etmektedir. Dolayısıyla deponun koordinat bilgileri ve son tarihi mevcuttur, geri kalan değerleri 0'dır.

6. MODEL

6.1 Genetik Algoritma

Solomon'un Zaman Pencereli ARP problemi için literatüre kattığı veri seti Rassal Sayı Kodlamalı GA ile modellenmiştir. Model iki algorithmadan oluşmaktadır: GA ve GA içerisinde kullanılan Yardımcı Algoritma. İlk olarak 4.3'te anlatılan ve aşağıda verilen Şekil 6.1'daki GA aşamalarına uygun olarak model anlatılacak, daha sonra bireylerin uygunluk değerlerini hesaplarken faydalanan Yardımcı Algoritma anlatılacaktır.



Şekil 6.1 : Genetik algoritma model akışı.

Aşama 1. Rassal Sayılı Kodlama

Daha önce anlatılan rassal sayılı genetik kodlamaya göre her birey 100 elemanlık rassal sayı dizisinden oluşmaktadır. Bu rassal sayılar, algoritma içerisinde azalan şekilde sıralanarak müşteri sırası elde edilir, bu yöntemi kullanan çalışmalar mevcuttur [69]. Kodlama yapılırken dikkat edilmesi gereken hususlar şunlardır: Algoritmanın devamında gerçekleştirilecek çaprazlama işlemi nedeniyle, bireylerde yaşanabilecek rassal sayı tekrarının önlenmesi için, rassal sayılar birbirinden farklı olmalıdır. Başlangıç çözüm kümesi N adet bireyden oluşmaktadır, dolayısıyla algoritmanın başlangıcında toplam $100 \cdot N$ adet birbirinden farklı rassal sayı gerekmektedir. Mutasyon aşamasında da yeni rassal sayılara ihtiyaç olacağından ve iterasyon sayısı büyük olduğundan, söz konusu rassal sayılar yeterince büyük olması nedeniyle 7 ile 2^{29} arasında üretilmektedir (kod içerisinde rassal sayı üreticinde kök değer olarak 7 verilmiştir). Örnek bir birey (kromozom) gösterimi aşağıdaki şekildedir:

1	2	3	...	99	100
100780101	7217008	45	...	254544	350

Bu bireyin yardımcı algoritma içerisinde kullanılacak müşteri sırası rassal sayıların azalan şekilde sıralanması ile aşağıdaki gibi elde edilir. Buna göre sıralamanın ilk hücresinde 97 numaralı müşteri bulunacaktır.

1	2	3	...	99	100
93	67	1	...	54	4

Aşama 2. Başlangıç Popülasyonun Oluşturulması

Genetik algoritma anlatımında bahsedildiği gibi başlangıç çözüm kümesi oluşturulmuştur ve bu kümede yer alacak birey sayısı çeşitli denemeler sonrasında $N=900$ olarak belirlenmiştir. Algoritma boyunca bu büyüklük sabit kalmaktadır. Yani küme 900 tane 100 elemanlık ve yukarıda belirtilen aralıkta değişen rassal sayı dizilerinden oluşmaktadır. Başlangıç çözüm kümesi aşağıdaki gibi olacaktır:

	1	2	3	...	99	100
1	100780101	7217008	45	...	254544	350
2	183	41632114	99762	...	239200123	871221
...						
900	8700018	60309	756929990	...	169	343256

Aşama 3. Uygunluk Değerlerinin Hesaplanması

Algoritmada bireylerin uygunlukları bilinmelidir, bu nedenle bir müşteri sırasını temsil eden her bir bireyin oluşturduğu çözümün amaç fonksiyonu olmalıdır. Bu amaç fonksiyonunun bulunabilmesi için bir sonraki bölümde anlatılacak olan yardımcı algoritma kullanılmıştır. Söz konusu yardımcı algoritma ile herhangi bir müşteri sırası için olurlu bir çözüm elde edilecek, yani rotalar belirlenip toplam mesafe hesaplanacaktır. Algoritmanın başında oluşturulan başlangıç çözüm kümesi büyüklüğü N tane birey için N tane amaç fonksiyonu mevcuttur.

Aşama 4. Seçim Operatörü

Seçim operatörü olarak sıklıkla kullanılan rulet çemberi seçim operatörü kullanılmıştır. Yardımcı algoritma ile hesaplanan amaç fonksiyon değerleri ile bireylerin uygunlukları ele alınacaktır. Seçilen bireylerin hepsi ikiyeşerli olarak çaprazlanmaktadır. Ayrıca en iyi bireyin bir sonraki adımda da olmasını sağlamak için elitizm uygulanmıştır. En iyi birey çaprazlama aşamasında başka bireyle çaprazlanır fakat kendisi de yine bir sonraki nesilde korunur.

Aşama 5. Yeniden Üreme

Aşama 5.1 Çaprazlama Operatörü

Mendes ve dig. çalışmalarında kullandığı çaprazlama yöntemi kullanılmıştır [70]. Çaprazlama için seçilmiş iki bireyin hangi genlerinin çaprazlanacağına karar vermek için bir çaprazlama oranı belirlenir (algoritmada bu parametre değeri 0.65 alınmıştır). Gen sayısı kadar (bu problem için 100 olacaktır) 0 ile 1 arasında rassal sayıdan oluşan bir dizi oluşturulur (maske) ve çaprazlama oranından büyük olan rassal sayının bulunduğu hücreler çaprazlanacaktır.

Çaprazlamanın nasıl yapıldığı aşağıdaki örnek ile açıklanmıştır:

	1	2	3	...	99	100
Ebeveyn 1	100780101	7217008	45	...	254544	350
Ebeveyn 2	183	41632114	99762	...	239200123	871221
MASKE	0.84	0.76	0.23	...	0.93	0.3
Çocuk 1	183	41632114	45	...	239200123	350
Çocuk 2	100780101	7217008	99762	...	254544	871221

Çaprazlama sonunda çözüm kümesi kadar yeni birey (çocuk) elde edilir. Çocukların da amaç fonksiyon değerleri yardımcı algoritma ile hesaplanır. Ebeveyn ile çocuklar yani toplam $2N$ tane birey amaç fonksiyon değerlerine göre sıralanır, ilk N tanesi yeni popülasyon için seçilir. Yeniden üreme operatörüne tekrar geçmeden önce bu kümenin bireylerinin bir kısmı mutasyona uğrar. En iyi birey korunmaktadır.

Aşama 5.2 Dönüşüm (Mutasyon) Operatörü

Popülasyon içerisinde çeşitliliğin sağlanması için bazı bireylerin dönüşüme uğraması gerekmektedir. Ebeveyn ve çocuklardan elde edilen N tane bireyin belirli bir oranına mutasyon işlemi gerçekleştirilecektir. Bu aşamada 2 farklı oran mevcuttur; birincisi ile çaprazlama sonrasında elde edilen N bireyin ne kadarına bu işlem uygulanacak, ikincisi ise ile mutasyon için seçilmiş olan bir bireyin allellerinin ne kadarı değiştirilecek buna karar verilir. Bireylerin bazı genleri yeniden üretilecektir yani yeni rassal sayılar üretilecektir. Çaprazlamaya benzer olarak her birey için bir maske üretilir. Maskedeki genlerden dönüşüm oranından büyük olanları için ilgili bireyin genindeki rassal sayı yerine yenisi üretilir, yeni üretilen rassal sayı daha önce üretilenlerin tekrarı olmayacaktır. Yeni elde edilen popülasyon ile algoritmaya devam edilir.

Aşama 6. Bitirme Koşulu

Algoritma 15000 adım sonrasında sonlandırılacaktır.

Yukarıda anlatılan akış ve içerisinde kullanılan ve bir sonraki bölümde anlatılacak olan yardımcı algoritmadan oluşan tüm algoritma C++ dilinde kodlanmıştır, program kodu cd içerisinde teslim edilen Ek1.'de mevcuttur.

6.2 Yardımcı Algoritma

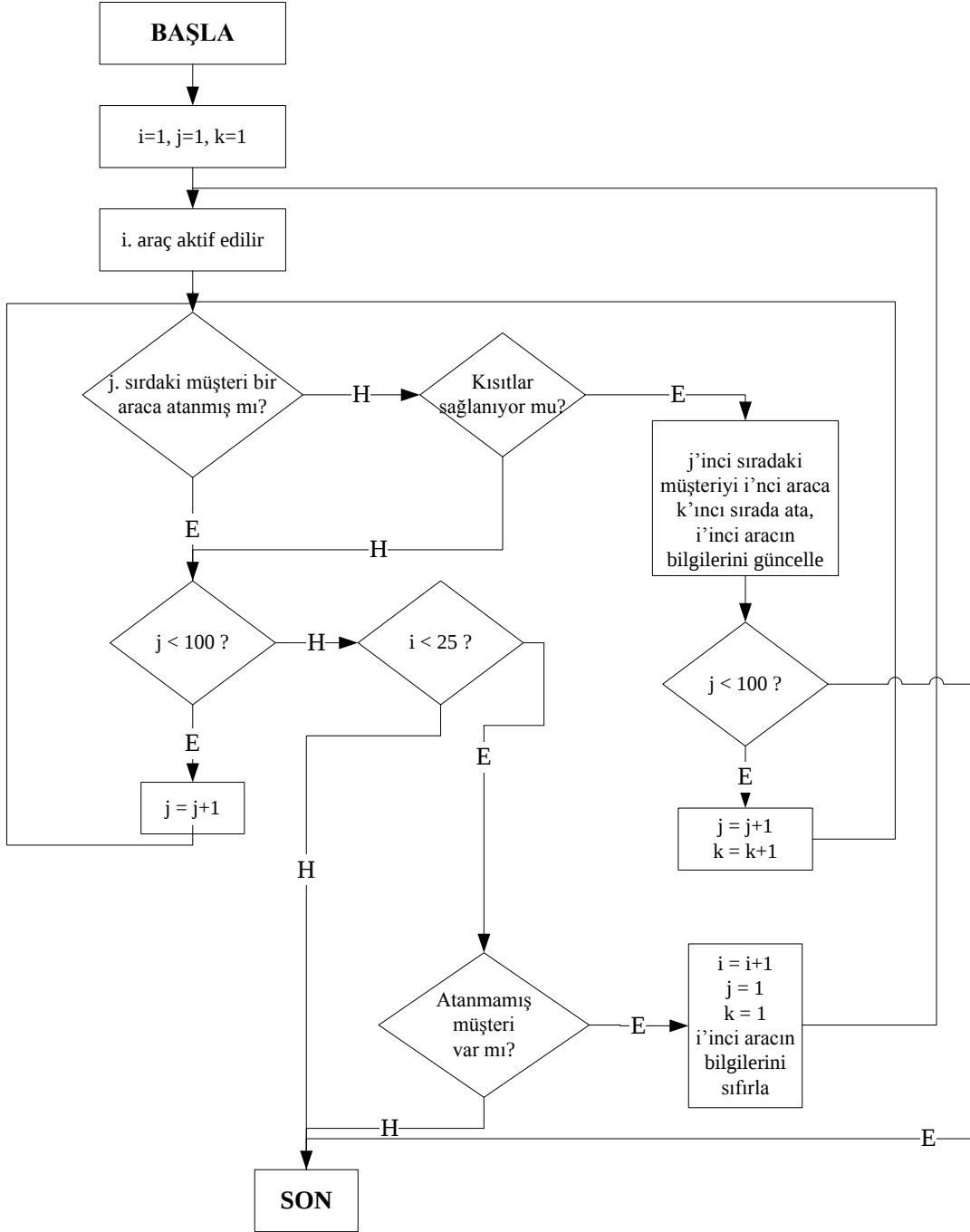
Her bireyin amaç fonksiyon değerinin hesaplanmasını sağlayacak bir algoritma gerekmektedir. Sıralama bazlı olan bir yardımcı algoritma sayesinde, her bir müşteri sırası için bir çözüm ve dolayısıyla toplam mesafe elde edilerek sözkonusu amaç fonksiyon değeri hesaplanır. Müşteriler 1'den 100'e kadar müşteri numaraları hücrelere gelecek şekilde sıralanırlar ve bu sıralama için yardımcı algoritma bir amaç fonksiyon değeri döner, yani algoritmanın girdisi bir müşteri sırasıdır, çıktısı ise toplam mesafe olan bir değerdir.

Şekil 6.2'de akışı gösterilen yardımcı algoritmada aşamalar özetle şu şekildedir: Başlangıçta araç numarasını ifade eden i değeri, müşteri dizisindeki hücre numarasını gösteren j değeri ve sözkonusu aracın rotasındaki mevcut sırayı temsil eden k değeri 1'dir. İlk olarak birinci araç aktif edilir ve sırasıyla müşteriler bu aracın rotasına atanmak üzere sorgulanır. Müşteri dizisinin birinci hücresindeki müşteri herhangi bir araca atanmamışsa üç tane kısıta bakılır;

- Araç kapasite kısıtı: Sıradaki müşterinin talebi, aracın mevcut kullanılan kapasitesine eklendiğinde aracın kapasitesini aşmamalıdır.
- Müşteri son tarih kısıtı: Aracın şimdiki zamanına, mevcut konumu ile müşteri arasındaki mesafe süre cinsinden ilave edildiğinde müşterinin son tarihini geçmemelidir.
- Depo son tarih kısıtı: Mevcut araca sıradaki müşteri atanırsa, teslimat sonrası araç depoya deponun son tarihini geçirmeden dönebilmelidir.

Eğer bu kısıtlar sağlanıyorsa, sıradaki müşteri mevcut aracın rotadaki mevcut sırasına atanır ve aracın bilgileri güncellenir. Yani aracın kullanılan kapasite miktarı müşterinin talep miktarı kadar arttırılır, şimdiki zamanı müşteriye teslimatı bittiği an olarak güncellenir ve rotasının ilgili sırasına müşteri numarası atanır. Artık aracın mevcut konumu da bu müşteri olacaktır. Bir sonraki sıradaki müşteriye bir sonraki sıra için bakılır, yani j değeri ve k değerleri birer arttırılarak mevcut müşteri için aynı sorgulamalar tekrar yapılır. Mevcut araç için tüm müşterilere bakıldıktan sonra eğer herhangi bir rotaya atanmamış müşteri kalmışsa bir başka araç aktif edilir ve rota sırası ve hücre numarası tekrar 1'e indirilerek sırasıyla tüm müşteriler tekrar sorgulanır. Bu işlemler iki bitirme koşulu sağlanana kadar tekrar edilir:

- herhangi bir rotaya atanmamış araç kalmayana kadar ya da
- aktif edilebilecek araç bitene kadar



Şekil 6.2 : Yardımcı algoritma akışı.

Algoritma sonunda kaç tane araç aktif edileceği, hangi müşterilerin hangi araçlara hangi sırada atanacağı elde edilmiş olur. Dolayısıyla tüm rotalar ve tüm aktif yollar belirlenmiş olur, buradan da tüm araçların katedeceği toplam mesafe bizim amaç fonksiyonumuzu verir. Eğer algoritmanın sonlanması araç yetersizliğinden ise ve bir

rotaya atanmamış araç kalmışsa sözkonusu sıralamanın amaç fonksiyon değerine bir ceza değeri eklenir. Çünkü olurlu olmayan bu çözümün amaç fonksiyonu düşük olacaktır ve genetik algoritma sırasında uygunluğu yüksek çıkacaktır ve olurlu olmayan bir çözüm olduğu halde algoritma tarafından iyi çözüm olarak değerlendirilecektir. Bunu engellemek için böyle bir ceza maliyeti ekleme yoluna gidilmiştir. Bu algoritma ile en iyi çözüme ulaşmak istenirse, tüm sıralamalar için algoritma koşturularak en iyi çözüm elde edilmelidir. Yani algoritmanın toplam 100! sıra için koşturulması gerekmektedir, bu da pek mümkün olmayacak kadar uzun süre gerektirmektedir.

Yardımcı algoritmayı 4 müşteri, 5 araçtan oluşan küçük bir örnek üzerinde inceleyelim. Araç kapasitesi 60 adet, servis süresi yine 90 zaman birim ve depo ve müşterilerle ilgili bilgiler Çizelge 6.1'deki gibi olsun. Depo'nun müşteri numarası 0 olan ilk satırla temsil edilmektedir. Sözkonusu müşteriler şu şekilde sıralanmış olsun: 1-3-4-2. Algoritma adımlara geçmeden önce depo ve müşteri konum bilgilerinden, birbirlerine olan uzaklıklarını öklit uzaklığı ile hesaplayacaktır. Sözkonusu uzaklıklar Çizelge 6.2'de bulunmaktadır. Depo ile 1 numaralı müşterinin birbirine olan uzaklığı şu şekilde hesaplanacaktır:

$$\begin{aligned} d(\text{Depo}, \text{Müşteri 1}) &= \sqrt{(15 - 32)^2 + (23 - 21)^2} \\ &= \sqrt{293} \\ &\approx 17.11 \end{aligned}$$

Adım1: Algoritmanın başlangıcında 1'inci araç aktif edilir ve bu aracın bilgileri şu şekildedir: konumu depodur, kullanılan kapasite miktarı ve şimdiki zamanı 0'dır. Rotasının 1'inci sırasına gelecek müşteri için sıralamanın ilk hücresinde olan 1 numaralı müşteriye bakılır, sözkonusu müşteri herhangi bir araca atanmamıştır. Dolayısıyla kısıtların sağlanıp sağlanmadığı sırasıyla kontrol edilir: Eğer müşteri bu aracın rotasına atanırsa, aracın taşınması gereken ürün miktarı toplam 30 adet olacaktır ve bu değer araç kapasitesi olan 60'dan küçüktür, yani ilk kısıt sağlanmıştır. Eğer araç mevcut konumu olan depodan bu müşteriye müşterinin son tarihinden önce ulaşabilmelidir. Depo ve müşteri 1'in arasındaki uzaklık yaklaşık 17.11 uzaklık birimidir, ve süre olarak alındığında yine 17.11 zaman birimi olacaktır. Yani araç müşteriye ulaştığında şimdiki zamanı 17.11 olacaktır, bu da müşterinin son tarihi olan 590 zaman biriminden öncedir, ikinci kısıt da sağlanmaktadır. Fakat araç teslimatın

başlayabileceği müşterinin hazır olma zamanına kadar beklemek zorundadır. Bu durumda teslimat en erken 524 zaman biriminde başlayacaktır. Üçüncü ve son kısıt deponun son tarih kısıtıdır, eğer bu müşteri bu araca atanan son müşteri ise araç teslimat yaptıktan sonra depoya dönecektir ve bu deponun son tarihinden önce gerçekleşmelidir. Araç bu müşteriye 524 zaman biriminde teslimat yapmaya başlayabilecektir, 90 zaman birim teslimat süresinden sonra 614 zaman biriminde depoya geri dönmek üzere yola çıkacaktır. Aralarındaki mesafe simetrik olduğundan dönüşte de 17.11 zaman birimlik mesafe sözkonusudur ve araç depoya dönecekse 631.11 zaman biriminde depoda olacaktır, bu da deponun son tarihi olan 700 zaman biriminden öncedir. Son kısıt da sağlanmaktadır, dolayısıyla birinci aracın rotasındaki birinci sıraya müşteri 1 atanacaktır. Aracın mevcut konumu müşteri 1, mevcut kullanılan kapasitesi 30 adet ve şimdiki zamanı 614 zaman birimi olacaktır (Bir sonraki müşteri sorgulanırken depoya dönüş süresi eklenmeyecektir). Bir sonraki müşterinin aracın rotasındaki ikinci sıraya uygunluğu kontrol edilecektir. Müşteri 1 ise bir araca atanmış olarak işaretlenecektir.

Çizelge 6.1 : Yardımcı algoritma örnek verileri.

Müşteri Numarası	X Koordinatı	Y Koordinatı	Talep Miktarı	Hazır O. Zamanı	Son Tarih	Servis Süresi
0	15	23	0	0	700	0
1	32	21	30	524	590	90
2	5	19	30	226	311	90
3	28	24	10	315	426	90
4	45	7	20	43	150	90

Çizelge 6.2 : Yardımcı algoritma örnek konum uzaklıkları.

UZAKLIK	Depo	M1	M2	M3	M4
Depo	0	17.11	10.77	13.03	34
M1	17.11	0	27.07	5	16.40
M2	10.77	27.07	0	23.53	41.76
M3	13.03	5	23.53	0	24.04
M4	34	16.40	41.76	24.04	0

Adım2. Bir sonraki sırada yani sıralamanın ikinci hücreinde 3 numaralı müşteri bulunmaktadır ve bu müşteri herhangi bir aracın rotasına atanmamıştır. Sırasıyla kısıtlara bakıldığında aracın kapasite kısıtı sağlanmaktadır fakat aracın şimdiki zamanı 624, müşterinin zaman penceresi olan (315,416)'yı geçmiştir, dolayısıyla bu

müşteri aracın rotasında ilgili sıraya atanamaz. Diğer iki müşteri için de zaman penceresi kısıtı sağlanamaz. Sonuç olarak araç 1'in rotasında sadece müşteri 1 olacaktır, araç depodan müşteri 1'e oradan da tekrar depoya dönecektir.

Adım3. 3, 4 ve 2 numaralı müşteriler henüz bir rotaya atanmamış olduklarından algoritma devam eder ve sıradaki araç 2 aktif edilir. Söz konusu aracın başlangıç konumu depodur ve mevcut kullanılan kapasitesi ve şimdiki zamanı 0'dır. Sıralamanın ilk hücresindeki müşteriden başlanarak sorgulama başlar. İlgili hücrede 1 numaralı müşteri vardır ve araç 1'e atanmıştır. Dolayısıyla bir sonraki hücredeki müşteriye yani müşteri 3'e geçilir.

Algoritma herhangi bir araca (rotaya) atanmış müşteri kalmayana kadar ya da aktif edilecek araç bitene kadar devam eder. Bu örnek için algoritma sonucunda toplam 3 araç aktif olmuştur yani 3 rota mevcuttur ve Şekil 6.3'de gösterilmiştir. Elde edilen rotalar ve toplam mesafe şu şekildedir:

Araç 1 (Rota 1): Depo → M1 → Depo

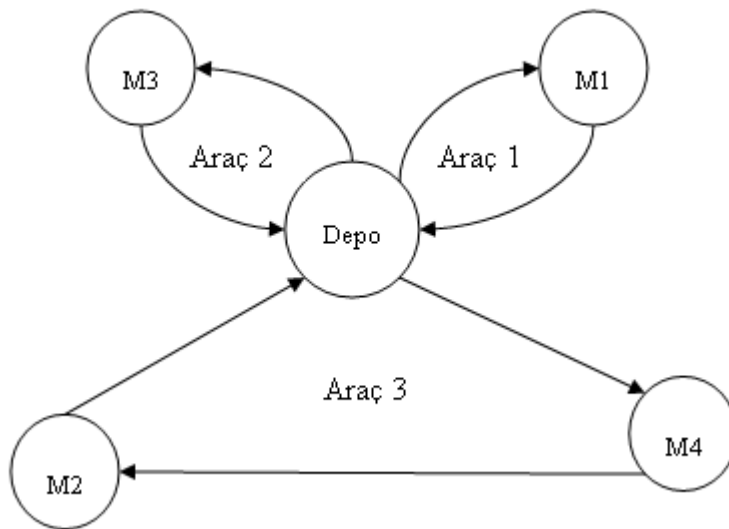
Araç 2 (Rota 2): Depo → M3 → Depo

Araç 3 (Rota 3): Depo → M4 → M2 → Depo

Toplam mesafe = $d(\text{Depo}, M1) + d(M1, \text{Depo}) + d(\text{Depo}, M3) + d(M3, \text{Depo}) +$

$d(\text{Depo}, M4) + d(M4, M2) + d(M2, \text{Depo})$

$= 17.11 + 17.11 + 13.03 + 13.03 + 34 + 41.76 + 10.77 = 146.81$ uzaklık birimi



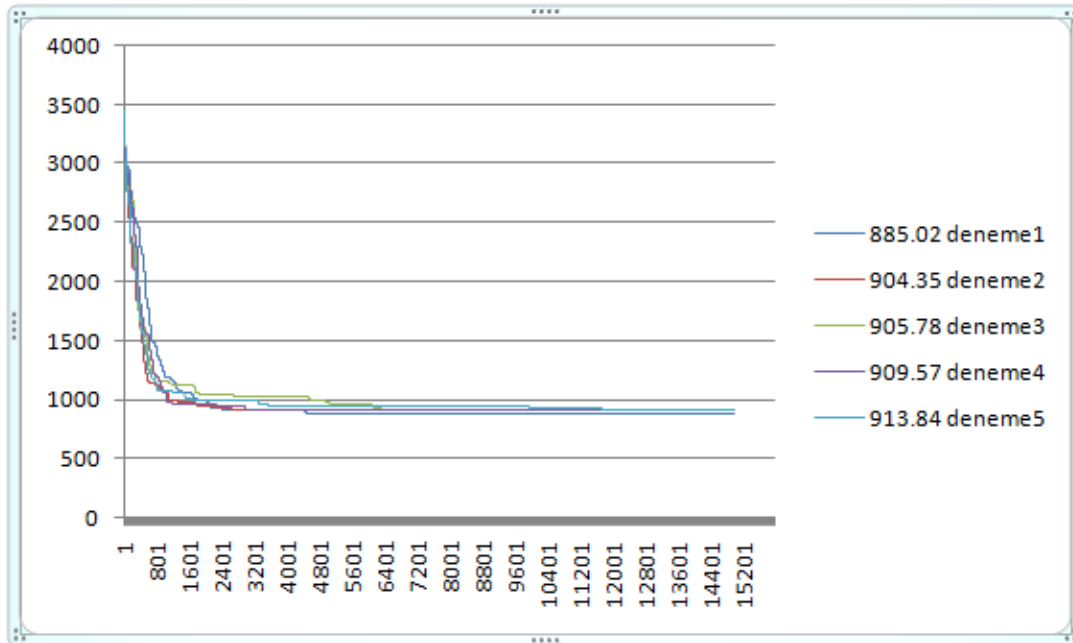
Şekil 6.3 : Yardımcı algoritma örnek rotaları.

6.3 Sayısal Sonuçlar

Model C++ programlama dili ile yazılmıştır ve Intel(R) Core(TM)2 Duo CPU işlemciye ve 2.00 GB RAM özelliğine sahip bir bilgisayarda koşturulmuştur. (Sözkonusu programa cd ile verilmiş olan Ek1'den ulaşılabilir). Farklı parametreler ve popülasyon büyüklükleri için algoritma 15000 iterasyon koşturulacak şekilde çeşitli denemeler yapılmıştır. En iyi sonuçların elde edildiği parametre değerleri şu şekildedir: popülasyon büyüklü 900, çaprazlama oranı 0.65, mutasyon oranları 0.8 ve 0.02'dir. Birbirinden farklı 5 koşumun adım sayısı arttıkça toplam mesafenin nasıl değiştiğini veren grafiği aşağıda Şekil 6.4'de ve özet bilgileri Çizelge 6.3'de verilmiştir (Denemeler uygunluk değerleri azalan sıraya göre olacak şekilde listelenmiştir).

Çizelge 6.3 : Koşum sonuçları.

	Toplam Mesafe	Kullanılan Araç Sayısı	En iyi çözümden uzaklık
Deneme 1	885.02	10	%6.97
Deneme 2	904.35	11	%9.31
Deneme 3	905.78	11	%9.48
Deneme 4	909.57	11	%9.94
Deneme 5	913.84	11	%10.46



Şekil 6.4 : 5 farklı koşumun iterasyon sayısı-toplam mesafe grafiği

Bu sonuçlardan en iyisi, literatürde yer alan problemin en iyi çözümüne yaklaşık %6.97 uzaklıktadır. Elde edilen sonuca göre aktif araç sayısı yani toplam rota sayısı 10'dur. Ve toplam mesafe 885.02 uzaklık birimidir. En iyi (optimum) çözümde ise toplam mesafe 827.3 uzaklık birimidir ve toplam araç sayısı 10'dur. Çalışmada elde edilen ve en iyi çözüme en yakın çözümün grafiği çözümün grafiği de Şekil 6.5'de verilmiştir ve rotalar şu şekildedir:

Rota 1: 43 – 42 – 41 – 40 – 44 – 46 – 45 – 48 – 51 – 50 – 52 – 49 – 47

Rota 2: 20 – 24 – 25 – 27 – 29 – 30 – 28 – 26 – 23 – 22 – 21 – 75

Rota 3: 90 – 87 – 86 – 83 – 82 – 84 – 85 – 88 – 89 – 91

Rota 4: 98 – 96 – 95 – 94 – 92 – 93 – 97 – 100 – 99 – 1

Rota 5: 5 – 3 – 7 – 8 – 10 – 11 – 9 – 6 – 4 – 2

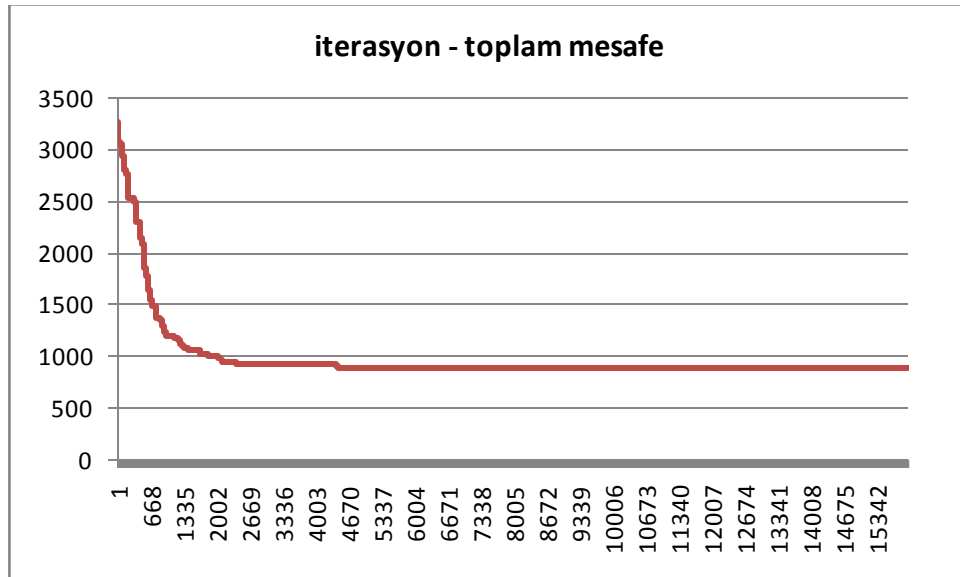
Rota 6: 13 – 17 – 18 – 19 – 15 – 16 – 14 – 12 – 69

Rota 7: 32 – 33 – 31 – 35 – 37 – 38 – 39 – 36 – 34

Rota 8: 81 – 78 – 76 – 71 – 70 – 73 – 77 – 79 – 80

Rota 9: 67 – 65 – 63 – 62 – 74 – 72 – 61 – 64 – 68 – 66

Rota 10: 57 – 55 – 54 – 53 – 56 – 58 – 60 – 59



Şekil 6.5 : Modelden elde edilen en iyi çözümün (deneme1) iterasyon sayısı-toplam mesafe grafiği.

7. SONUÇLAR VE ÖNERİLER

Günümüzde şirketler arasındaki rekabet, çok detaylı stratejik kararlardan etkilenebilecek hassasiyettedir. Bunun sebebi bilgi ve teknolojinin çok daha hızlı yayılabilmesi ve şirketlerin kendi bünyelerindeki işlemlere hızlı bir şekilde entegre edebildikleridir. Özellikle dağıtım sektörünü ilgilendiren ve lojistiğin önemli bir bileşeni olan araç rotalama, sözkonusu rekabetin öne çıktığı bir alandır. Şirketlerin, müşterilere hızlı ve kaliteli hizmet vererek müşteri memnuniyetini arttırmanın yanı sıra, şirketin kendi iç verimliliğini de sağlaması ve her iki açıdan da en iyi olacak kararı vermesi gerekmektedir.

Gerçek hayatta sıklıkla karşılaşılan ve dağıtım yapılacak müşterilerin belirli zaman aralığında hizmeti kabul edebileceği Zaman Pencereli ARP problemi literatürde sıklıkla incelenen ve çözümü için model geliştirilen bir problemdir. Banka teslimatları, posta teslimatı, endüstriyel atık toplama, okul servis aracı rotalama ve çizelgeleme gibi belirli durumları yansıtan Zaman Pencereli ARP için [67], literatürde diğer meta-sezgisel yöntemler olan TA ve TB'nin kullanıldığı etkin modeller bulunmaktadır, fakat Rassal Sayı Kodlamalı GA ile yapılmış bir çalışmaya rastlanılmamıştır. Daha önce yapılan çalışmalarda Solomon'un veri setinden faydalanan araştırmacıların geliştirdikleri modeller ile c101 veri seti için en iyi çözümü bulmaktadır, fakat bu modellerde ya TA ve TB kullanılmıştır, ya da GA başka bir yöntemle melezleştirilmiştir. GA etkin bir meta-sezgisel yöntemdir ve kombinatorik problemlerin çözümü için elverişlidir. Bireylerin rassal sayı olarak kodlanması ise, bu problem türü için sıklıkla kullanılan permütasyon kodlamadan onarım işlemlerini gerektirmemesi açısından üstünlüğe sahiptir.

Bu çalışmada Zaman Pencereli ARP için, Rassal Sayı Kodlamadan faydalanan ve melez olmayan bir GA modeli ortaya konulmuştur. Başlangıç popülasyonundaki çözümler rassal olarak üretilmektedir ve uygunluk değerlendirme fonksiyonunda da herhangi bir yöntemden faydalanılmamaktadır. Yapılan koşumlar sonucunda elde edilen sonuçlara göre en iyi çözüme %6.5-%10.5 arası uzaklıkta çözümler 120 dk'da elde edilmektedir. Sözkonusu model en iyi çözümü elde edememektedir fakat hızlı bir modeldir.

Klasik sezgisel yöntemler ya da ARP için kullanılmış olan kesin yöntemlerden faydalanılmadan geliştirilen bu model ile Rassal Sayı Kodlamalı GA'nın etkinliği incelenmiştir. İncelenen çalışmalarda bu yeni kodlamanın kullanıldığı bir çalışmaya rastlanmaması bu çalışmanın motivasyonlarından biridir. En iyi çözümün elde edilemediği çalışmada genetik operatörlerin farklı kombinasyonları kullanılarak, ya da göç (immigration) gibi operatörlerin ilave edilmesi modelin performansını etkileyebilir. Ayrıca modelde başlangıç popülasyonu elde edilirken En Yakın Komşu Sezgiseli, Ekleme Algoritmaları gibi klasik sezgisellerden faydalanılabilir, ayrıca uygunluk değerlendirme fonksiyonu da değiştirilerek yine yukarıda bahsedilen yöntemlerden faydalanılarak geliştirilebilir. İyi çözümlerden oluşan başlangıç popülasyonu ile algoritmaya başlamak modelin etkinliğini artıracaktır, bireylerin numaralarına göre sıralanmış halini girdi olarak alan bir algoritma olan uygunluk değerlendirme fonksiyonunun geliştirilmesi ile de daha hassas değerlendirme yapılarak daha iyi rotalama yapılacaktır. Bu yöntemlerden faydalanılması program kodunun daha yavaş çalışmasına neden olacaktır fakat daha az iterasyon ile en iyi çözüme ulaşılabilir. Bu çalışmadaki GA modeli üzerinde uygun geliştirmeler ile Rassal Sayı Kodlamanın kullanıldığı etkin ve diğer yöntemlerle rekabet edebilecek bir model ortaya koymak hedeflenmektedir.

KAYNAKLAR

- [1] **Dantzig G. B. and Ramser, J. H.**, 1959: The Truck Dispatching Problem, *Management Science*, Vol. **6**, no. 1, pp.80-91.
- [2] **Garic, T. and Gold, H.**, 2008: Vehicle Routing Problem, In-Teh, Croatia.
- [3] **Toth, P. and Vigo, D.**, 2002: The Vehicle Routing Problem, Society for Industrial and Applied Mathematics.
- [4] **Ho, W., Ho, G. T. S., Ji, P. and Lau, H. C. W.**, 2008: A hybrid genetic algorithm for the multi-depot vehicle routing problem, *Engineering Applications of Artificial Intelligence*, Vol. **21**, Issue 4, pp. 548-557.
- [5] **Magnanti, T. L.**, 1981: Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects, Massachusetts Institute of Technology, Operations Research Center, Working Paper.
- [6] **Laporte, G.**, 1992: The Vehicle Routing Problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, Vol. **59**, pp.345-358.
- [7] **Gendreau, M., Laporte, G. and Séguin, R.**, 1996: Stochastic vehicle routing, *European Journal of Operational Research*, Vol. **88**, Issue 1, pp.3-12.
- [8] **Lin, S-W., Lee., Z-J., Ying, K-C. and Lee, C-Y.**, 2009: Applying hybrid meta-heuristic for capacitated vehicle routing problem, *Expert Systems with Applications*, Vol. **36**, Issue 2, Part 1, pp.1505-1512.
- [9] **Christofides N., Mingozzi A. and Toth P.**, 1981: Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations, *Mathematical Programming*, North-Holland Publishing Company, Vol. **20**, no. 1, pp. 255-282.
- [10] **Jacobs-Blecha, C. and Goetschalckx, M.**, 1993: The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms, Technical Report MHRC-TR-88-13, Georgia Institute of Technology, Atlanta, Georgia.
- [11] **Bianchessi, N. and Righini, G.**, 2007: Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, *Computers & Operations Research*, Vol. **34**, Issue 2, pp. 578-594.
- [12] **Jin, M., Liu, K. and Eksioglu, B.**, 2008: A column generation approach for the split delivery vehicle routing problem, *Operations Research Letters*, Vol. **36**, Issue 2, pp. 265-270.
- [13] **Hemmelmayr, V. C., Doerner, K. F. and Hartl R. F.**, 2007: A variable neighborhood search heuristic for periodic routing problems, *European Journal of Operational Research*, Vol **195**, Issue 3, pp. 791-802.

- [14] **Badeau, P., Guertin, F., Gendreau, M., Potvin J. and Taillard, E.,** 1997: A Parallel Tabu Search Heuristic for The Vehicle Routing Problem With Time Windows, *Transpn Res.-C*, Vol. 5, no. 2, pp.109-122.
- [15] **Laporte, G. and Nobert, Y.,** 1987: Exact Algorithms for the Vehicle Routing Problem in Surveys in Combinatorial Optimization, Martello, S., Laporte, G., Minoux, M. and Ribeiro, C. (editors), *Annals of Discrete Mathematics*, pp.147-184.
- [16] **Winston, W. Y.,** 2003: Operations Research Applications and Algorithms, Fourth Edition, Duxbury Press.
- [17] **Balinski M. L. and Quandt R. E.,** 1963: On An Integer Program for a Delivery Problem, *Operations Research*, Vol. 12, no. 2, pp.300-304.
- [18] **Topçu, Y. İ,** 2008: Operations Research II Lecture Notes.
- [19] **Ralphs, T. K., Kopman, L., Pulleyblank, W. R. and Trotter, L. E.,** 2003: On the Capacitated Vehicle Routing Problem, *Mathematical Programming*, Vol. 94, no.2-3, pp. 343-359.
- [20] **Pereira, F. B. and Tavares, J.,** 2009: Bio-inspired Algorithms for the Vehicle Routing Problem, *Studies in Computational Intelligence*, Vol. 161 Springer.
- [21] **Haupt, R. L. and Haupt, S. E.,** 2004: Practical Genetic Algorithms, Second Edition, A John Wiley & Sons, Inc.
- [22] **Smith, V. J. R., Osman, I. H., Reeves, C. R. and Smith, G. D.,** 1996: Modern Heuristic Search Methods, John Wiley and Sons Ltd.
- [23] **Osman I. H., Kelly, J. P.,** 1996: Meta-Heuristics: Theory & Applications, Kluwer Academic Publisher.
- [24] **Vob, S., Martello, S., Osman, I. H. and Roucairol C.,** 1999: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publisher.
- [25] **Reeves C. R.,** 1995: Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill International (UK) Limited.
- [26] **Kikpatrick, S., Gelatt, C. D. and Vecchi, M. P.,** 1983: Optimization by Simulated Annealing, *Science*, Vol. 220, no. 4598, pp. 671-680.
- [27] **Lin, S.-W., Yu, V. F. and Chou, S-Y,** 2009: Solving the truck and trailer routing problem based on a simulated annealing heuristic, *Computers & Operations Research*, Vol. 36, Issue 5, pp. 1683-1692.
- [28] **Oliveira, H. C. B., Vasconcelos, G. C. and Bastos, G.,** 2006: A Multi-Start Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows, *Proceedings of the Ninth Brazilian Symposium on Neural Networks (SBRN'06)*.
- [29] **Gendreau, M.,** 2003: An Introduction to Tabu Search in Handbook of Metaheuristics, Glover, F., Kochenberger, G. A., (editors), pp.37-55.
- [30] **Eiben, A. E. and Smith, J. E.,** 2003: Introduction to Evolutionary Computing, Springer.

- [31] **Rothlauf, F.**, 2002: Representations for Genetic and Evolutionary Algorithms, A Springer-Verlag Company.
- [32] **Chiang, W.-C. and Russell R. A.**, 1996: Simulated annealing metaheuristics for the vehicle routing problem with time windows, *Annals of Operations Research*, Vol. **63**, no. 1, pp.3-27.
- [33] **Czech Z. J. and Czarnas, P.**, 2002: Parallel simulated annealing for the vehicle routing problem with time windows, *Proc. 10th Euromicro Workshop Parallel Distributed Network-Based Processing*, Canary Islands, Spain, pp.376-383.
- [34] **Lin, S.-W., Ying, K.-C., Lee, Z.-J. and Chen, H.-S.**, 2006: Vehicle Routing Problems with Time Windows Using Simulated Annealing, *International Conference on Systems, Man, and Cybernetics*, October 8-11, 2006, Taipei, Taiwan, pp.645-650.
- [35] **Brandão, J. and Mercer, A.**, 1997: A tabu search algorithm for the multi-trip vehicle routing and scheduling problem, *European Journal of Operational Research*, Vol. **100**, Issue 1, pp.180-191.
- [36] **Gendreau, M., Laporte, G., Musaraganyi, C. and Taillard, E. D.**, 1999: A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, Vol. **26**, Issue 12, pp.1153-1173.
- [37] **Nanry, W. P. and Barnes, J. W.**, 2000: Solving the pickup and delivery problem with time windows using reactive tabu search, *Transportation Research Part B*, Vol. **34**, Issue 2, pp.107-121.
- [38] **Ho, S.C. and Haugland, D.**, 2004: A tabu search heuristic for the vehicle routing problem with time windows and split deliveries, *Computers & Operations Research*, Vol. **31**, Issue 12, pp.1947-1964.
- [39] **Tavakkoli-Moghaddam, R., Safaei, N., Kah, M. M. O. and Rabbani, M.**, 2007: A New Capacitated Vehicle Routing Problem with Split Service for Minimizing Fleet Cost by Simulated Annealing, *Journal of the Franklin Institute*, Vol. **334**, Issue 5, pp.406-425.
- [40] **Tavakkoli-Moghaddam, R., Safaei, N. and Gholipour, Y.**, 2006: A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length, *Applied Mathematics and Computation*, Vol. **176**, Issue 2, pp.445-454.
- [41] **Lim, A. and Zhu, W.**, 2006: A Fast and Effective Insertion Algorithm for Multi-depot Vehicle Routing Problem with Fixed Distribution of Vehicles and a New Simulated Annealing Approach, *IEA/AIE 2006*, Vol. **4031**, pp.282-291.
- [42] **Brandão, J.**, 2004: A tabu search algorithm for the open vehicle routing problem, *European Journal of Operational Research*, Vol. 157, Issue 3, pp.552-564.
- [43] **Brandão, J.**, 2006: A new tabu search algorithm for the vehicle routing problem with backhauls, *European Journal of Operational Research*, Vol. **173**, Issue 2, pp.540-555.

- [44] **Michalewicz, Z.**, 1998: Genetic Algorithms + Data Structures = Evolution Programs, Third Edition, Springer.
- [45] **Kurt, M. ve Semetay, C.**, 2001: Genetik Algoritma ve Uygulama Alanları, *Mühendis Makine Dergisi*, Ekim Sayısı.
- [46] **Mitchell, M.**, 1999: An Introduction to Genetic Algorithms, The MIT Press.
- [47] **Reeves, C. R. and Rowe J. E.**, 2003: Genetic Algorithms-Principles and Perspectives A Guide to GA Theory, Kluwer Academic Publishers.
- [48] **Reeves, C.**, 2003: Genetic Algorithms in Handbook of Metaheuristics, Glover, F., Kochenberger, G. A., (editors), pp.55-82.
- [49] **Rothlauf, F.**, 2002: Representations for Genetic and Evolutionary Algorithms, Physica-Verlag.
- [50] **Gen, M. and Cheng, R.**, 1997: Genetic Algorithms and Engineering Design, John Wiley and Sons, Inc.
- [51] **Bean J. C.**, 1994: Genetic Algorithms and Random Keys for Sequencing and Optimization, *ORSA Journal on Computing*, Vol. **6**, no.2, pp.154-160, Spring 1994.
- [52] **Goldberg, D. E.**, 1989: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., USA.
- [53] **Burke, E. K. and Kendall, G.**, 2005: Search Methodologies, Springer, USA.
- [54] **Snyder L. V. and Daskin, M. S.**, 2006: A random-key genetic algorithm for the generalized traveling salesman problem, *European Journal of Operational Research*, Vol. **174**, no.1, pp.38–53.
- [55] **Iyer, S.K. and Saxena, B.**, 2004: Improved genetic algorithm for the permutation flowshop scheduling problem, *Computers & Operations Research*, Vol. **31**, Issue 4, pp.593-606.
- [56] **Ruiz, R., Maroto, C. and Alcaraz, J.**, 2006: Two new robust genetic algorithms for the flowshop scheduling problem, *Omega*, Vol. **34**, Issue 5, pp.461-476.
- [57] **Nagano, M. S., Ruiz, R. and Lorena, L. A. N.**, 2008: A Constructive Genetic Algorithm for permutation flowshop scheduling, *Computers & Industrial Engineering*, Vol. **55**, Issue 1, pp. 195-207.
- [58] **Baker, B.M. and Ayeche, M.A.**, 2003: A genetic algorithm for the vehicle routing problem, *Computers & Operations Research*, Vol. **30**, Issue 5, pp.787-800.
- [59] **Liu, S., Huang, W. and Ma, H.**, 2009: An effective genetic algorithm for the fleet size and mix vehicle routing problems, *Transportation Research Part E*, Vol. **45**, Issue 3, pp.434–445.
- [60] **Wang, C.-H. and Lu, J.-Z.**, 2009: A hybrid genetic algorithm that optimizes capacitated vehicle routing problems, *Expert Systems with Applications*, Vol. **36**, Issue 2, pp. 2921–2936.

- [61] **Berger, J. and Barkaoui, M.**, 2004: A parallel hybrid genetic algorithm for the vehicle routing problem with time windows, *Computers & Operations Research*, Vol. **31**, Issue 12, pp.2037–2053.
- [62] **Li, J. and Zhang, J.**, 2007: A genetic algorithm to vehicle routing problem in reverse logistics, *14th International Conference on Management Science & Engineering*, August 20-22, pp.573-578.
- [63] **Wang W., Wang, Z. and Qiao, F.**, 2008: An Improved Genetic Algorithm for Vehicle Routing Problem with Time-window, *International Symposium on Computer Science and Computational Technology*.
- [64] **Alvarenga, G.B., Mateus, G.R. and Tomi, G.**, 2007: A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, *Computers & Operations Research*, Vol. **34**, pp.1561-1584.
- [65] **Wang, X.-P., Xu, C.-L. and Hu, X.-P.**, 2008: Genetic Algorithm for Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles, *15th International Conference on Management Science & Engineering*, 10-12 September 2008, pp.128-133.
- [66] **Ma, J., Zou, H., Gao, L.-Q and Li, D.**, 2006: Immune Genetic Algorithm for Vehicle Routing Problem with Time Windows, *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, Dalian, 13-16 August 2006, pp.3465-3469.
- [67] **Solomon, M. M.**, 1987: Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, Vol. **35**, no.2, pp.254-265.
- [68] **Url-1** <<http://www.idsia.ch/~luca/macsvrptw/problems/welcome.htm>>, alındığı tarih 03.05.2009.
- [69] **Samanlioglu, F., Ferrell Jr., W., G. and Kurz, M. E.**, 2008: A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem, *Computers & Industrial Engineering*, Vol. **55**, Issue 2, pp.439-449.
- [70] **Mendes, J. J. M., Gonçalves, J. F. and Resende, M. G. C.**, 2009: A random key based genetic algorithm for the resource constrained project scheduling problem, *Computers & Operations Research*, Vol.**36**, Issue 1, pp.92 -109.

EKLER

EK A.1 c101 problem verisi.

EK A.1**Çizelge A.1 : c101 problem verisi.**

Müşteri Numarası	X Koordinatı	Y Koordinatı	Talep Miktarı	Hazır O. Zamanı	Son Tarih	Servis Süresi
0	40	50	0	0	1236	0
1	45	68	10	912	967	90
2	45	70	30	825	870	90
3	42	66	10	65	146	90
4	42	68	10	727	782	90
5	42	65	10	15	67	90
6	40	69	20	621	702	90
7	40	66	20	170	225	90
8	38	68	20	255	324	90
9	38	70	10	534	605	90
10	35	66	10	357	410	90
11	35	69	10	448	505	90
12	25	85	20	652	721	90
13	22	75	30	30	92	90
14	22	85	10	567	620	90
15	20	80	40	384	429	90
16	20	85	40	475	528	90
17	18	75	20	99	148	90
18	15	75	20	179	254	90
19	15	80	10	278	345	90
20	30	50	10	10	73	90
21	30	52	20	914	965	90
22	28	52	20	812	883	90
23	28	55	10	732	777	90
24	25	50	10	65	144	90
25	25	52	40	169	224	90
26	25	55	10	622	701	90
27	23	52	10	261	316	90
28	23	55	20	546	593	90
29	20	50	10	358	405	90
30	20	55	10	449	504	90
31	10	35	20	200	237	90
32	10	40	30	31	100	90
33	8	40	40	87	158	90
34	8	45	20	751	816	90
35	5	35	10	283	344	90
36	5	45	10	665	716	90
37	2	40	20	383	434	90
38	0	40	30	479	522	90
39	0	45	20	567	624	90
40	35	30	10	264	321	90
41	35	32	10	166	235	90
42	33	32	20	68	149	90
43	33	35	10	16	80	90
44	32	30	10	359	412	90
45	30	30	10	541	600	90
46	30	32	30	448	509	90
47	30	35	10	1054	1127	90
48	28	30	10	632	693	90
49	28	35	10	1001	1066	90
50	26	32	10	815	880	90
51	25	30	10	725	786	90
52	25	35	10	912	969	90
53	44	5	20	286	347	90
54	42	10	40	186	257	90
55	42	15	10	95	158	90
56	40	5	30	385	436	90
57	40	15	40	35	87	90
58	38	5	30	471	534	90

Müşteri Numarası	X Koordinatı	Y Koordinatı	Talep Miktarı	Hazır O. Zamanı	Son Tarih	Servis Süresi
59	38	15	10	651	740	90
60	35	5	20	562	629	90
61	50	30	10	531	610	90
62	50	35	20	262	317	90
63	50	40	50	171	218	90
64	48	30	10	632	693	90
65	48	40	10	76	129	90
66	47	35	10	826	875	90
67	47	40	10	12	77	90
68	45	30	10	734	777	90
69	45	35	10	916	969	90
70	95	30	30	387	456	90
71	95	35	20	293	360	90
72	53	30	10	450	505	90
73	92	30	10	478	551	90
74	53	35	50	353	412	90
75	45	65	20	997	1068	90
76	90	35	10	203	260	90
77	88	30	10	574	643	90
78	88	35	20	109	170	90
79	87	30	10	668	731	90
80	85	25	10	769	820	90
81	85	35	30	47	124	90
82	75	55	20	369	420	90
83	72	55	10	265	338	90
84	70	58	20	458	523	90
85	68	60	30	555	612	90
86	66	55	10	173	238	90
87	65	55	20	85	144	90
88	65	60	30	645	708	90
89	63	58	10	737	802	90
90	60	55	10	20	84	90
91	60	60	10	836	889	90
92	67	85	20	368	441	90
93	65	85	40	475	518	90
94	65	82	10	285	336	90
95	62	80	30	196	239	90
96	60	80	10	95	156	90
97	60	85	30	561	622	90
98	58	75	20	30	84	90
99	55	80	10	743	820	90
100	55	85	20	647	726	90

ÖZGEÇMİŞ

Ad Soyad: Pınar Dursun

Doğum Yeri ve Tarihi: Eminönü, 20.10.1983

Lisans Üniversitesi: İstanbul Teknik Üniversitesi Matematik Mühendisliği

Yayın Listesi:

- Baysan S., Çınar D., Çelebi D., **Dursun P.** and Özdemir D., 2009: Modeling People Flow: A Real Life Case Study in ITU Science Center. *International Simulation Multi-conference (ISMc'09)*, July 13-16, 2009 Istanbul, Turkey (submitted).
- Topçu İ. and **Dursun P.**, 2008: A simulation based selection model of distribution mode in a supply chain. *OR50 Annual Conference*, September 9-11, 2008 York, England (abstract).
- **Dursun, P.**, Temur, G. T., Tanyaş, M., 2007: İşletmelerde Dağıtım Ağı Tasarımı Ve Optimizasyonu: Bir Uygulama. 4. *Kobiler ve Verimlilik Kongresi*, Aralık 7-8, 2007 İstanbul, Türkiye.
- Ostaysi, B., Baysan, S. and **Dursun, P.**, 2007: A Novel Approach For Economic Justification of RFID Technology in Courier Sector: A Real-Life Case Study, *Annual RFID Eurasia Conference*, September 5-6, 2007 Istanbul, TURKEY.