



T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



YÜKSEK LİSANS TEZİ

ARAÇ ROTALAMA PROBLEMİNE GENETİK ALGORİTMA
YAKLAŞIMI ve ÖRNEK BİR UYGULAMA

Saniye ÇEYREKOĞLU

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı


DANIŞMAN
Prof. Dr. Şakir ESNAF


Nisan, 2017

İSTANBUL

Bu çalışma 13.04.2017 tarihinde ařağıdaki jüri tarafından Endüstri Mühendisliğı Anabilim Dalı, Endüstri Mühendisliğı Programında Yüksek Lisans tezi olarak kabul edilmiştir.


Tez Jürisi


Prof. Dr. Şakir ESNAF (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Alp BARAY
İstanbul Üniversitesi
Mühendislik Fakültesi


Doç. Dr. Tarık KÜÇÜKDENİZ
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Mehmet Mutlu YENİSEY
İstanbul Üniversitesi
Mühendislik Fakültesi


Yrd. Doç. Dr. Zeynep GERGIN
İstanbul Kültür Üniversitesi
Mühendislik Fakültesi



20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Günümüzde teknolojinin hızla gelişmesi, sektörlerdeki rekabetin artması ve Dünya'nın bir pazar haline gelmesinden dolayı dağıtım sistemleri çok daha etkin ve karmaşık hale gelmiştir. Lojistik ve tedarik zinciri yönetiminde toplam maliyetlerin önemli bir kısmını taşıma maliyetleri oluşturmaktadır. Firmaların müşteri taleplerini zamanında ve az maliyetle karşılanmasında araç filoları için hazırlanacak iyi bir plan, lojistik maliyetlerini önemli ölçüde düşürecek ve firmalara rekabette ciddi avantaj sağlayacaktır.

Bu çalışmada; müşteri taleplerinin zamanında ve en az maliyetle karşılanmasının planlanması için karmaşık bir optimizasyon problemi olan Araç Rotalama Problemi (ARP)'nin çözümünde Genetik Algoritmalar (GA) kullanılarak yeni bir yazılım oluşturulmuştur. Çalışma gerçek bir dünya problemi de olan Türkiye turu problemi üzerinde diğer bir yöntemle karşılaştırmalı olarak çözülmüştür.

Bu çalışmada, yoğun çalışmaları arasında zamanını ayırarak bana yol gösteren ve yardımcı olan tez danışmanım Prof. Dr. Şakir ESNAF'a, manevi desteklerinden dolayı eşim Osman ÇEYREKOĞLU'na ve aileme teşekkürlerimi sunmayı bir borç bilirim.

Nisan 2017

Saniye ÇEYREKOĞLU

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	xi
SİMGE VE KISALTIMA LİSTESİ	xii
ÖZET.....	xiii
SUMMARY	xv
1. GİRİŞ.....	1
1. GENEL KISIMLAR	3
2.1. ARAÇ ROTALAMA PROBLEMLERİ	3
2.1.1. Araç Rotalama Prensipleri	8
2.1.3. Araç Rotalama Problemi Çeşitleri	9
2.1.3.1. Kısıtlarına Göre Araç Rotalama Problemleri (KARP)	11
2.1.3.2. Yolların Durumuna Göre Araç Rotalama Problemleri	19
2.1.3.3. Rotalama Durumlarına Göre Araç Rotalama Problemleri	19
2.1.4.4. Çevre Durumu Bakımından Araç Rotalama Problemleri	20
2.1.4. Araç Rotalama Problemleri İçin Çözüm Yöntemleri	21
2.1.4.1. Kesin Yöntemler.....	22
2.1.4.2. Sezgisel Yöntemler.....	24
2.1.5. ARP' nin Uygulama Alanları	28
2.2. GENETİK ALGORİTMALAR.....	28
2.2.1. Genetik Algoritmalar ile İlgili Temel Kavramlar	31
2.2.1.1. Gen	31
2.2.1.2. Kromozom.....	31
2.2.1.3. Popülasyon (Yığın, Kütle)	31
2.2.1.4. Kodlama (Dizi gösterimi).....	32
2.2.2. Genetik Algoritmaların Aşamaları	33
2.2.3. Başlangıç Popülasyonu	34

2.2.4. Seçim (Yeniden Üretim İşlemi).....	35
2.2.4.1. Rulet Tekerı.....	35
2.2.4.2. Rank Seçim Yöntemi.....	36
2.2.4.3. Turnuva Seçim Mekanizması.....	37
2.2.4.4. Sıralı Seçim Mekanizması.....	37
2.2.4.5. Global Elitizm ve Yerel Elitizm.....	38
2.2.4.6. Kalabalıklaştırma.....	38
2.2.5. Genetik Algorİtmalarda Kullanılan Operatörler.....	38
2.2.5.1. Çaprazlama Operatörü.....	38
2.2.5.2. Mutasyon Operatörü (Rakam Değişimi).....	44
2.2.6. Genetik Algorİtmaların Çalışma Prensibi.....	45
2.2.7. Genetik Algorİtmaların Performansını Etkileyen Etmenler.....	48
2.2.8. Genetik Algorİtmaların Uygulamaları.....	49
2.3. ARAÇ ROTALAMA PROBLEMLERİNDE GENETİK ALGORİTMALAR KULLANIMININ LİTERATÜR ARAŞTIRMASI.....	51
2. MALZEME VE YÖNTEM	55
3.1. GELİŞTİRİLEN PROGRAMIN TANITIMI.....	55
3.1.1. Programın Çalışma Prensibi.....	55
3.1.2. Amaç Fonksiyonu.....	55
3.1.3. Programın Akış Şeması.....	55
3.1.4. Programın Temel Özellikleri.....	56
3.1.5. Çalışma Adımları.....	58
3.2. EXCEL SOLVER (EVOLUTIONARY) GENETİK ALGORİTMALAR PROGRAMI.....	60
3.3 EN YAKIN KOMŞU ALGORİTMASI.....	61
3.4 UYGULAMA.....	61
3.3.1. Problemin Tanımı.....	61
3.3.2. Problemin En Yakın Komşu Algorİtması ile Çözümü.....	62
3.3.3. Problemin Excel – Solver (Evolutionary) Genetik Algorİtmalar Programı ile Çözümü.....	68
3.3.3. Problemin Geliştirilen Algorİtma Programı ile Çözümü.....	78
3. BULGULAR	109
4.1. KARŞILAŞTIRMA.....	109
4. TARTIŞMA VE SONUÇ	113
KAYNAKLAR.....	115

EKLER.....	121
EK 1. Algoritma Programının Kodları.....	121
ÖZGEÇMİŞ	133



ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1: ARP Örnek Gösterimi (Şen ve diğ., 2015).....	3
Şekil 1.2: Araç Rotalama Problemi Çeşitleri (Koç Ç, 2012).	10
Şekil 1.3: Araç Rotalama Çeşitlerinin Birbiriyle İlişkisi (Toth ve Vigo, 2002a).	11
Şekil 1.4: Zaman Aralıklı Araç Rotalama Problemi Örneği (Demircioğlu, M., 2009).	15
Şekil 1.5: Açık ve Kapalı Uçlu Araç Rotalama Problem Şekilleri (Erol, 2006).	20
Şekil 1.6: Araç Rotalama Problemi Çözüm Yöntemleri (Demircioğlu, M., 2009).	22
Şekil 1.7: Kod ve Çözüm Uzayı (Cheng vd., 1996).	32
Şekil 1.8: Binary Kodlama Gösterimi (Şeker, 2007).....	33
Şekil 1.9: Permütasyon Kodlama (Şeker, 2007).	33
Şekil 1.10: Rulet Tekerı Seçimi (Erdal, 2007).....	36
Şekil 1.11: Rank Seçim Yöntemi 1 (Akoğlu, 2006).....	36
Şekil 1.12: Rank Seçim Yöntemi 2 (Akoğlu, 2006).....	37
Şekil 1.13: Çaprazlama operasyonu (Erdal, 2007).	40
Şekil 1.14: Genetik Algoritma Akış Süreci (Preux ve Talbi, 1999).	48
Şekil 2.1: Programın Akış Şeması.....	56
Şekil 2.2: Uygulama Parametre Ekranı.	58
Şekil 2.3: Uygulama Menüsü.....	59
Şekil 2.4: Uygulama Rota Seçim Ekranı.	59
Şekil 2.5: Uygulama Harita Oluşturma Ekranı.	60
Şekil 2.6: 10 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.	63
Şekil 2.7: 20 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.	64
Şekil 2.8: 30 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.	65
Şekil 2.9: 40 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.	66

Şekil 2.10: 60 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.	67
Şekil 2.11: 81 ilin En Yakın Komşu Algoritması ile Çözümü.	68
Şekil 2.12: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#1).	69
Şekil 2.13: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#2).	70
Şekil 2.14: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#3).	71
Şekil 2.15: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#4).	72
Şekil 2.16: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#5).	73
Şekil 2.17: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#1).	74
Şekil 2.18: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#2).	75
Şekil 2.19: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#3).	76
Şekil 2.20: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#4).	77
Şekil 2.21: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#5).	78
Şekil 2.22: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).	79
Şekil 2.23: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).	80
Şekil 2.24: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).	81
Şekil 2.25: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).	82
Şekil 2.26: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).	83
Şekil 2.27: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).	84
Şekil 2.28: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).	85
Şekil 2.29: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).	86
Şekil 2.30: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).	87
Şekil 2.31: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).	88
Şekil 2.32: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).	89
Şekil 2.33: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).	90
Şekil 2.34: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).	91
Şekil 2.35: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).	92
Şekil 2.36: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).	93
Şekil 2.37: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).	94

Şekil 2.38: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).....	95
Şekil 2.39: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).....	96
Şekil 2.40: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).....	97
Şekil 2.41: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).....	98
Şekil 2.42: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).....	99
Şekil 2.43: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).....	100
Şekil 2.44: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).....	101
Şekil 2.45: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).....	102
Şekil 2.46: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).....	103
Şekil 2.47: 81 ilin Uygulama (GA) ile Çözümü (#1).	104
Şekil 2.48: 81 ilin Uygulama (GA) ile Çözümü (#2).	105
Şekil 2.49: 81 ilin Uygulama (GA) ile Çözümü (#3).	106
Şekil 2.50: 81 ilin Uygulama (GA) ile Çözümü (#4).	107
Şekil 2.51: 81 ilin Uygulama (GA) ile Çözümü (#5).	108

TABLO LİSTESİ

Sayfa No

Tablo 4.1: Karşılaştırmalı Çözüm Sonuçları. 111



SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
C	: Araç kapasite değeri
d_{ij}	: i. ve j. noktalar arası mesafe
L	: Araç mesafe kısıtı
N	: Genetik Algoritmalarda popülasyon büyüklüğü
q_i	: i nolu müşterinin talep miktarı
s_i	: Araçların servis süresi

Kısaltmalar	Açıklama
ARP	: Araç Rotalama Problemi
AUARP	: Açık Uçlu Araç Rotalama Problemi
ÇARP	: Çok Depolu Araç Rotalama Problemi
GA	: Genetik Algoritmalar
GSP	: Gezgin Satıcı Problemi
DARP	: Dinamik Araç Rotalama Problemi
KARP	: Kısıtlarına Göre Araç Rotalama Problemleri
KKARP	: Kapasite Kısıtlı Araç Rotalama Problemi
KUARP	: Kapalı Uçlu Araç Rotalama Problemi
MKARP	: Mesafe Kısıtlı Araç Rotalama Problemi
NP	: Non Polinomyal Problemler
PARP	: Periyodik Araç Rotalama Problemi
PDARP	: Parça Dağıtım Araç Rotalama Problemi
SARP	: Stokastik Araç Rotalama Problemi
TDARP	: Topla-Dağıt Araç Rotalama Problemi
ZARP	: Zaman Pencere Araç Rotalama Problemi
ZBARP	: Zaman Bağımlı Araç Rotalama Problemi

ÖZET

YÜKSEK LİSANS TEZİ

ARAÇ ROTALAMA PROBLEMİNE GENETİK ALGORİTMA YAKLAŞIMI ve ÖRNEK BİR UYGULAMA

Saniye ÇEYREKOĞLU

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Şakir ESNAF

Günümüzün rekabetçi ortamında var olan ve gelecekte de var olmak isteyen işletmeler her geçen gün kendilerinden ve rakiplerinden daha iyi olmaya çalışmaktadırlar. İşletmelerin dağıtım ağı süreçlerini en iyilemek ve etkin bir şekilde yönetmek amacıyla özellikle ulaştırma, taşıma ve dağıtım konularında çeşitli kararlar alması gerekmektedir. Bu kararlardan birisi de tesislerden müşterilere gerçekleştirilecek olan rotalama kararlarıdır. Dağıtım, planlama, lojistik gibi alanlar başta olmak üzere birçok sektörde geniş uygulama alanlarına sahip olan araç rotalama problemi, optimizasyon alanında araştırmacı ve akademisyenler tarafından üzerinde uzun yıllardır yoğun olarak çalışılan popüler, çözümü zor (NP-hard) bir problemdir. Ayrıca, gerçek hayatta, sistemler için modellemede kullanılan önemli bir dağıtım problemidir.

Kombinatoriyal optimizasyon problemlerine uygulanabilen genetik algoritmalar (GA), doğal genetik seçim ve rastsal arama tekniklerini kullanarak çözüm bulmaya çalışan bir yöntem olarak eldeki çözümlerden hareketle daha iyi çözümler üreterek optimum veya optimuma yakın çözümlere ulaşmaya çalışır. Genetik algoritmalar kendi arama proseslerini oluşturması bakımından pek çok avantaja sahiptir. GA, en çok sıralama, çözümlenme, atama, dağıtım, planlama problemlerinin çözümünde kullanılmaktadırlar.

Bu alıřmada; meta sezgisel bir yntem olan genetik algoritmalar yardımı ile ara rotalama problemine zm aranmıř ve geliřtirilen algoritma gerek bir problem olan Trkiye illeri zerinde uygulanmaktadır. Kullanılabilecek oklu rotalar zorluk derecesi ařamalı olarak arttırılarak ve rasgele seilerek oluřturulmaktadır. Bu rotalar, geliřtirilen algoritma ile optimize edilerek ıkan sonular sayısal ve grsel olarak verilmektedir. Geliřtirilen algoritmanın etkinlięini ve performansını lmek iin aynı rotalar Excel - Solver (Evolutionary) Genetik Algoritmalar Programı ve En Yakın Komřu Algoritması ile de optimize edilmektedir. Alınan sonular, mesafe ve sre bakımından karřılařtırmalı olarak sunulmaktadır. Bu sonulara gre, geliřtirilen algoritma sonuları dięer yntemlere gre daha bařarılıdır.

Nisan 2017, 149 sayfa.

Anahtar kelimeler: Ara Rotalama Problemleri, Genetik Algoritmalar



SUMMARY

M.Sc. THESIS

GENETIC ALGORITHM APPROACH TO VEHICLE ROUTING PROBLEM and CASE STUDY

Saniye ÇEYREKOĞLU

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Industrial Engineering

Supervisor : Prof. Dr. Şakir ESNAF

The corporations, which are existing in the competitive world of our time and which aim to be existing also in future, are trying to be better than themselves and their rivals day by day. The corporations should give various decisions especially in transportation, transfer and distribution in order to make their distribution net the most superior and manage it effectively. One of these decisions is the rotation decisions that will take place between the facilities and the customers. Vehicle rotation problem, which has a large application area like distribution, planning and logistics, is a popular and difficult to solve (NP-hard) problem that has been by studied for years by research specialists in optimization and academists. In addition, it is an important distribution problem that can be used to model many real-life systems.

Genetic algorithms (GA), which can be applied to combinatorial optimization problems with random search techniques and natural genetic selection, aims to reach optimum or almost optimum solutions by creating better solutions considering the existing results. Genetic algorithms has many advantages about setting up its search processes. GA is mostly used in solving the problems of sorting, scheduling, assingment, distribution and planning

In this study, a solution has been searched for vehicle rotation problems with the help of GA that is a metaheuristic method and the developed algorithm has been applied on cities in Turkey that constitutes a real problem. The multi routes that can be used have been formed by its level of difficulty being increased progressively and by being chosen randomly. Once these routes have been optimized with the developed algorithm, the results have been submitted numerically and visually. The same routes are optimized with Excel-Solver (Evolutionary) genetics algorithm program and Nearest Neighborhood Algorithm to measure the effectiveness and performance of the developed algorithm. The results have been presented comparatively with a view to distance and time. According to these results, the results of the developed algorithm are more successful than the other methods.

April 2017, 149 pages.

Keywords: Vehicle Routing Problem, Genetic Algorithms



1. GİRİŞ

Bir işletmenin başlıca lojistik amacı, hedeflere minimum maliyetle ulaşılabilen bir sistem kurmaktır. Lojistik sürecinin en önemli bileşenlerinden biri de ürünlerin hareketlerini başlangıç noktasından satış noktasına kadar takip etmektir. Dağıtım faaliyetleri, ürünlerin hareketlerinin ve bu kapsamda yapılan faaliyetler çerçevesinde ulaşım metodunun belirlenmesidir. Doğru yolun seçimi için, birçok yerel, bölgesel ve uluslararası yerli ve yabancı hizmetlerden haberdar olmak gereklidir. Lojistik süreç içerisinde dağıtım, genelde tek başına yüksek maliyetli bir işlemdir. Bu nedenle önemli bir bileşen olup etkin olarak yönetilmesi gereklidir.

İşletmelerin toplam lojistik maliyetlerinin %30- %60' ı dağıtım maliyetlerinden kaynaklanmaktadır. Bu nedenle dağıtım ekipman ve personelinin etkili ve verimli bir şekilde kullanılması işletme yöneticileri açısından önemli bir ilgi alanıdır. Bu bakımdan dağıtım maliyetlerini azaltmak ve müşterilere sunulan servisin kalitesini arttırmak için en kısa zamanı ve mesafeyi verecek olan, bir aracın şebeke içerisinde izleyeceği en uygun rotayı bulmak günümüzde en çok tartışılan konu haline gelmiştir.

En küçük maliyetli rotaların, eldeki araç filosu ile belirlenmesi olarak tanımlanabilecek olan araç rotalama problemi (ARP) ile ilgili birçok çalışma yapılmıştır. Bu problem genel anlamıyla herhangi bir işletmenin müşterilerine (konumları belirli olan) hizmet ya da ürün ulaştırabilmesi için operasyonel ve fiziki kısıtları göz önünde bulundurarak minimum maliyetli rotayı elde edebilme problemidir. Hizmet ve ürünün ulaştırılmasında işletmelerin önüne çıkan bu problem işletmelerde maliyetlerin ciddi oranda yükselmesine neden olabilmektedir. Bu nedenle araç rotalama probleminin verimli olarak çözümü tasarruf sağlanması konusunda büyük öneme sahiptir.

Genetik algoritmalar doğayı taklit ederek en iyi sonuca ulaşmaya çalışırlar. Bu algoritmaların amacı doğadaki birey ve sistemlerin uyum özelliğini kullanarak yapay sistemler oluşturmaktır. Genetik algoritmalar rastlantısal deneme teknikleri ile sonuç bulmak için uğraşan parametre kodlama esaslı arama tekniklerindedir. Genetik algoritmalar çoğu problem çeşidi için doğru parametreler kullanarak çalışıldığında en

iyiye yakın sonuçlar üretebilen yöntemlerdendir. Genetik algoritmalar, teoride optimum sonucu garanti edememesine karşın uygulamada optimuma yakın sonuçları üretebilmekte ve çoğu problemde optimum sonuca ulaşılabilir. Buna göre genetik algoritmalar çoğu kombinatoriyal optimizasyon problemine gayet doğru ve başarı ile uygulanabilmektedir.

Sezgisel yöntemlerin bir sınıfına giren ve meta-sezgisel bir yöntem olan Genetik Algoritmalar (GA), doğadaki evrim sürecini bilgisayar ortamında taklit ederek bu tür problemler için en iyi çözümü ya da yakın çözümü elde edebilmektedir. Literatürde GA' ın kullanıldığı ve ARP gibi diğer kombinatoriyal 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, ARP çözümünde kullanımı için yeni algoritma geliştirilmiştir.

Bu tez çalışmasında ilk olarak araç rotalama problemine değinilmekte, literatürde yer alan temel araç rotalama problemi çeşitleri, kullanılan çözüm yöntemleri ve uygulama alanları açıklanmaktadır.

İkinci bölümde Genetik Algoritmalar detaylı olarak açıklanmaktadır.

Üçüncü bölümde Araç Rotalama Problemlerinde Genetik algoritmaların kullanımı ile ilgili literatür çalışması yapılmaktadır.

Dördüncü bölümde geliştirilen algoritmanın programı detaylı olarak anlatılmakta, ele alınan problem tanıtılmakta; kullanabilecek çoklu rotalar zorluk derecesi aşamalı olarak artırılarak ve rasgele seçilerek oluşturulmakta ve bu rotalar geliştirilen algoritma ile optimize edilerek çıkan sonuçlar sayısal ve görsel olarak verilmektedir.

Beşinci bölümde de bu tez çalışmasının özeti oluşturularak sonuç ve değerlendirmeler yapılmaktadır.

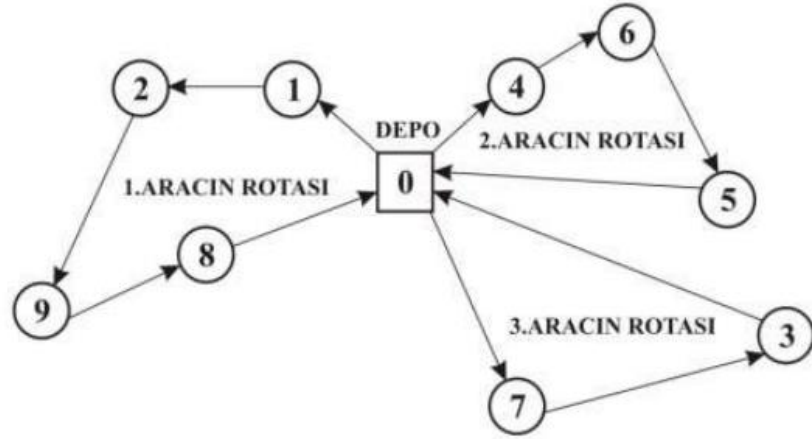
1. GENEL KISIMLAR

2.1. ARAÇ ROTALAMA PROBLEMLERİ

Araç Rotalama Problemi, en basit şekli ile bir merkezi depoda üstlenmiş bulunan ve her biri aynı kapasiteye sahip olan homojen araçlar filosunun her biri farklı bir yerleşime ve bilinen bölünemeyen talebi olan müşteriler setine, toplam seyahat süresini ve mesafesini minimize edecek şekilde hizmet sunarak depoya geri dönmesi için gerekli rotanın ve her bir rotadaki yolların belirlenmesi problemidir (Ayan, 1999).

Araç Rotalama Problemi (ARP) taşıma maliyetlerini azaltmak ve müşteri hizmetlerini arttırmak için bir aracın takip etmesi gereken en iyi rotayı bulmaktır (Alkan, 2003).

ARP'nin genel yapısı Şekil 1.1' de gösterilmiştir. Şekil 1.1' de gösterilen 1'den 9'a kadar numaralanmış düğümler sipariş noktalarını, 0 ise ana depoyu ifade etmektedir.



Şekil 1.1: ARP Örnek Gösterimi (Şen ve diğ., 2015).

Araç Rotalama Problemi (ARP) Dantzing ve Ramser tarafından 1959' da ilk olarak ele alınmış ve literatüre kazandırılmıştır. Bu çalışmada üzerinde durulan konu benzin istasyonlarına benzin dağıtım problemidir. Bu problemde en iyi sonuç için yazarlar ilk matematiksel programlama modelini oluşturmuşlardır. Bu çalışmadan sonra Clarke ve

Wright 1964' te probleme yeni bir sezgisel çözüm önermiştir. Bu öneriden sonra araç rotalama problemine olan ilgi artmıştır. Üzerinde en çok metot üretilen optimizasyon problemlerinden biri araç rotalama problemidir (Toth ve Vigo, 2002).

Malakandri ve Daskin (1992) trafik sıkışıklığı temelli, zaman bağımlı ARP algoritmaları üzerinde çalışmışlardır. Bunun sonucunda trafik sıkışıklığının olduğu durumlarda sabit bir rotanın olmayacağını, bunun seyahat edilecek zamanın günün hangi saatleri olacağına bağlı olduğunu ileri sürmüşlerdir (Alkan, 2003).

Firmaların dağıtım ve lojistik maliyetlerinin tamamı üzerinde, ciddi bir oran, taşıma maliyetlerinden oluşmaktadır. Bu nedenle taşıma maliyetlerinin azaltılması, çalışan ve ekipmanların verimli ve etkin olarak kullanılması firma yönetici için önemli bir hale gelmiştir. Bu giderleri azaltmak ve sunulan hizmet kalitesini artırmak için minimum mesafe ve zaman içinde aracın müşteri noktalarına en uygun rotayı kullanarak ulaşması günümüzün üzerinde en çok durulan konularındandır. Klasik bir ARP, farklı lokasyonlarda yer alan müşteri noktalarındaki taleplerin karşılanması amacıyla depolardan araçların sevk edilmesidir. Bu problem ele alınırken hedef, etkin ve verimli olarak müşteri taleplerinin olabildiğince minimum zaman, mesafe ve maliyetle karşılayan rotayı üretmektir. Aşağıdaki maddelere, firmalar müşterilerine hizmet ve ürün ulaştırmada rota oluştururken dikkat etmelidir (Karahana, 2003).

- Farklı konumlarda bulunan tüm müşterilerin talepleri karşılanmalıdır.
- Her müşteriye talebinin karşılanması için sadece bir araç bir defa uğramalıdır.
- Rotaya depodan başlayan bir araç mutlaka depoya geri dönmelidir.
- Uğruncak müşterilerin talep toplamları araç kapasitesini geçmemelidir.
- Her bir rota için sadece bir araç planlanmalıdır.
- Araç rotalamada temel amaç, her bir aracın katedeceği mesafenin en aza indirilmesi olmalıdır.

Bir birleşti optimizasyon (combinatorial optimization) problemi olan araç rotalama problemi, NP-zor problem sınıfına aittir. Başka bir ifade ile araç rotalama problemlerinde değişken sayısı arttıkça çözüme ulaşma zamanı minimum üstsel olarak artış gösterir (Taşkın, 2003).

Araç rotalama probleminde, müşteriye sunulan hizmetin kalitesi arttırmak istendiğinde bazı operasyonel kısıtlamalar karşımıza çıkabilmektedir. Bu kısıtlamalardan bazıları aşağıda verilmiştir:

- Belirlenen rota için seçilen aracın taşıma ve/veya katedebileceği mesafe kapasitesi
- Aracın müşteriye hizmet verme zamanı
- Müşterinin hizmet veya ürünü, sadece belirli bir zaman aralığında alabilmesi (zaman penceresi kısıtı)
- Araç sürücüsü mesai saatlerinin rotalama esnasında göz önünde bulundurulması
- Filodaki araç sayısının kısıtlı olması

Klasik bir araç rotalama problemi için uygulanabilecek, belirlenmiş amaç fonksiyonları için örnekler aşağıdadır (Toth ve Vigo, 2002).

- Taşıma giderlerinin azaltılması için araçların katedeceği toplam mesafe ve seyahat süresinin azaltılmaya çalışılması.
- En az araç sayısı ile bütün müşterilerin taleplerini yerine getirmek.
- Oluşturulan tüm rotaların yük ve seyahat süresi bakımından dengeli olmasının sağlanması.

Araç Rotalama Problemi (ARP)' nin ana yapıtaşları; taleplerin yapısı, sevkiyatı yapılacak ürünün tipi, depo ve müşteri konumları ve araç filosundan oluşur.

1. **Taleplerin Yapısı:** Araç rotalama problemleri (ARP) 'nde talepler sabit veya değişken olabilir. Sabit olması durumunda önceden talep bilinir. Ancak değişken olması durumunda bazı taleplerin bilinmesine karşın bazılarında da araç seyahat halinde iken belirlenmektedir.
2. **Sevkiyatı Yapılacak Ürünün Tipi:** Araçlar ile farklı tiplerde ürünler taşınmaktadır. Bunlar gıda ürünleri, tehlikeli olan maddeler, çöp toplama, gazete dağıtımı gibi paketler problem için extra bir zorluk oluşturmazlar. Ancak öğrenci servisleri, eşitlik, verimlilik ve güvenlik gibi extra ihtiyaçlardan dolayı daha zorlu olabilirler.
3. **Depo ve Müşteri Konumları:** Aracın rotası depoda başlar ve biter. Problem, çok depolu veya tek depolu olarak depo sayısına göre belirlenir. Problemden,

depolar birden çok ise her deponun sahip olduğu araçlar bulunur. Bu nedenle de problem, her deponun kendine ait bir araç rotalama problemine dönüşür. Ancak herhangi bir depodan rotaya başlayan bir araç başka bir depoya dönerse problem bir bütün olarak incelenmelidir.

Müşteri konumları, belli ise hangi müşteriye hangi aracın gideceği belirlenmelidir. Aksi durumda müşteri konumları olası konumlar arasından belirleneceği için extra bir yerleştirme kararına ihtiyaç duyulacaktır.

Depo ve müşteri konumları, bazı araç rotalama problemlerinde birbirleriyle aynıdır. Örnek olarak öğrenci servisleri verilebilir. Öğrenci servislerinde, öğrenci durakları depo iken okul müşteri konumu; öğrencilerin okul dönüşlerinde ise okul depo, öğrenci durakları da müşteri konumuna dönüşmektedir.

4. **Araç Filosu:** Tüm araç rotalama problemlerinde araç kapasitelerinin eşit olduğu varsayılır. Bazı durumlarda filodaki araç kapasiteleri farklı olabilir. Böyle durumlarda, hangi araç kapasitesine sahip aracın hangi rotaya atanacağı extra bir kısıt getirir. Filodaki araçların diğer niteliklerinden olan yakıt sarfiyatı, hız ve sevkiyatı yapılacak ürüne uygunluğunun rotaların belirlenmesi üzerinde direkt etkileri bulunmamaktadır (Eryavuz ve Gencer, 2001).

Standart bir tek depoya sahip araç rotalama problemi formüle edilmek istenirse aşağıdaki doğrusal model kullanılabilir:

M = Araç adedi

N = Müşteri adedi

d_{ij} = i ve j konumları arasındaki mesafe

q_i = i müşterisinin talep miktarı

C = Araç kapasitesi

Değişken;

$$X_{ijk} = \begin{cases} 1, & \text{eğer } k \text{ nolu araç } i \text{ noktasından } j \text{ noktasına hareket ederse} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$\text{Amaç Fonksiyonu: } \min Z = \sum_{i=0}^N \sum_{j=0, j \neq i}^N \sum_{k=1}^M d_{ij} X_{ijk} \quad (2.1)$$

Şu kısıtlara göre:

$$i = 0 \text{ için } \sum_{k=1}^M \sum_{j=1}^N X_{ijk} = M \quad (2.2)$$

$$i \in \{1, \dots, N\} \text{ için } \sum_{k=1}^M \sum_{j=0, j \neq i}^N X_{ijk} = 1 \quad (2.3)$$

$$j \in \{1, \dots, N\} \text{ için } \sum_{k=1}^M \sum_{i=0, i \neq j}^N X_{ijk} = 1 \quad (2.4)$$

$$k \in \{1, \dots, M\} \sum_{i=1}^N X_{i0k} \leq 1 \quad (2.5)$$

$$k \in \{1, \dots, M\} \text{ için } \sum_{i=1}^N q_i \sum_{j=0, j \neq i}^N X_{ijk} \leq C \quad (2.6)$$

(2.1) nolu amaç fonksiyonu minimum maliyetin (mesafe) elde edilmesi gerekliliğini anlatmaktadır. (2.2) nolu denklem depodan yola çıkacak araç sayısının M adet olması gerektiğini, (2.3) ve (2.4) nolu denklemler müşterinin sadece bir araç tarafından ziyaret edileceğini (2.5) nolu denklem bir aracın sadece bir defa depodan çıkıp rotada bir defa kullanılacağını ve (2.6) nolu denklem ise yüklemelerin, araç kapasitesi C' yi aşmaması gerektiğini ifade etmektedir. Araç adedi maksimum M adet olması gerektiği durumlarda (2.2) nolu denklemde eşittir (=) yerine küçük eşit (\leq) işareti kullanılmalıdır. Oluşturulan bu model içinde temel kısıt olan (2.3) ve (2.4) nolu denklemler rotaların sürekliliği açısından önem kazanmaktadır. Standart bir araç rotalama problemi çözümü için genelde şu bilgilere ihtiyaç duyulmaktadır:

- Müşteri konumları arasındaki mesafe veya seyahat süresi
- Depo ile müşteri konumları arasındaki mesafe veya seyahat süresi
- Müşterilere ait talep miktarları
- Araç adedi ve araçların kapasitesi

- En iyilenmesi hedeflenen değer veya değerler (amaç fonksiyonu)

Literatürde, araç rotalama problemi için belirlenmiş birçok optimizasyon kriteri bulunmaktadır. Bunlardan en çok kullanılanları, toplam rota mesafesi, seyahat süresi, rota adedi, müşteri memnuniyeti ve ürün adedi dengeleme olarak belirtilebilir. En yaygın olarak kullanılanlar toplam rota mesafesi ve rota adedi ARP amaç fonksiyonlarıdır. ARP çözümlerinde optimum metotlar veya sezgisel yöntemler kullanılır.

Coğrafi alan ve hizmet ulaştırılan müşteri adedi bakımından araç rotalama problemi zor bir problemdir. Hizmet sunulan müşteri adedi arttığında olası rota adedi artacak ve hesaplama işlemi zorlaşacaktır. Dolayısıyla, sezgisel algoritmalar araç rotalama problemlerinin çözümünde daha avantajlıdır (Eryavuz ve Gencer, 2001).

2.1.1. Araç Rotalama Prensipleri

Literatüre bakıldığında çeşitli araç rotalama problemleri ve her problem için oluşturulmuş çok sayıda algoritma görülecektir. Fakat bu algoritmalara bakıldığında gerçek hayatta firmaların optimum çözümlerine ulaşamadıkları görülmektedir. Bu nedenle halen araç rotalama problemlerinin sağlıklı çözümler verebilecek algoritmaları üretmek adına araştırmacılar çalışmalara devam etmektedir.

Araştırmacılar ve uygulayıcılar, araştırmalar ve uygulamalar esnasında daha başarılı ve uygulanabilir rotaların oluşturulması için çalışmalarda aşağıdaki 8 prensibin göz önünde bulundurulması konusunda hem fikir olmuşlardır (Ballou, 1999):

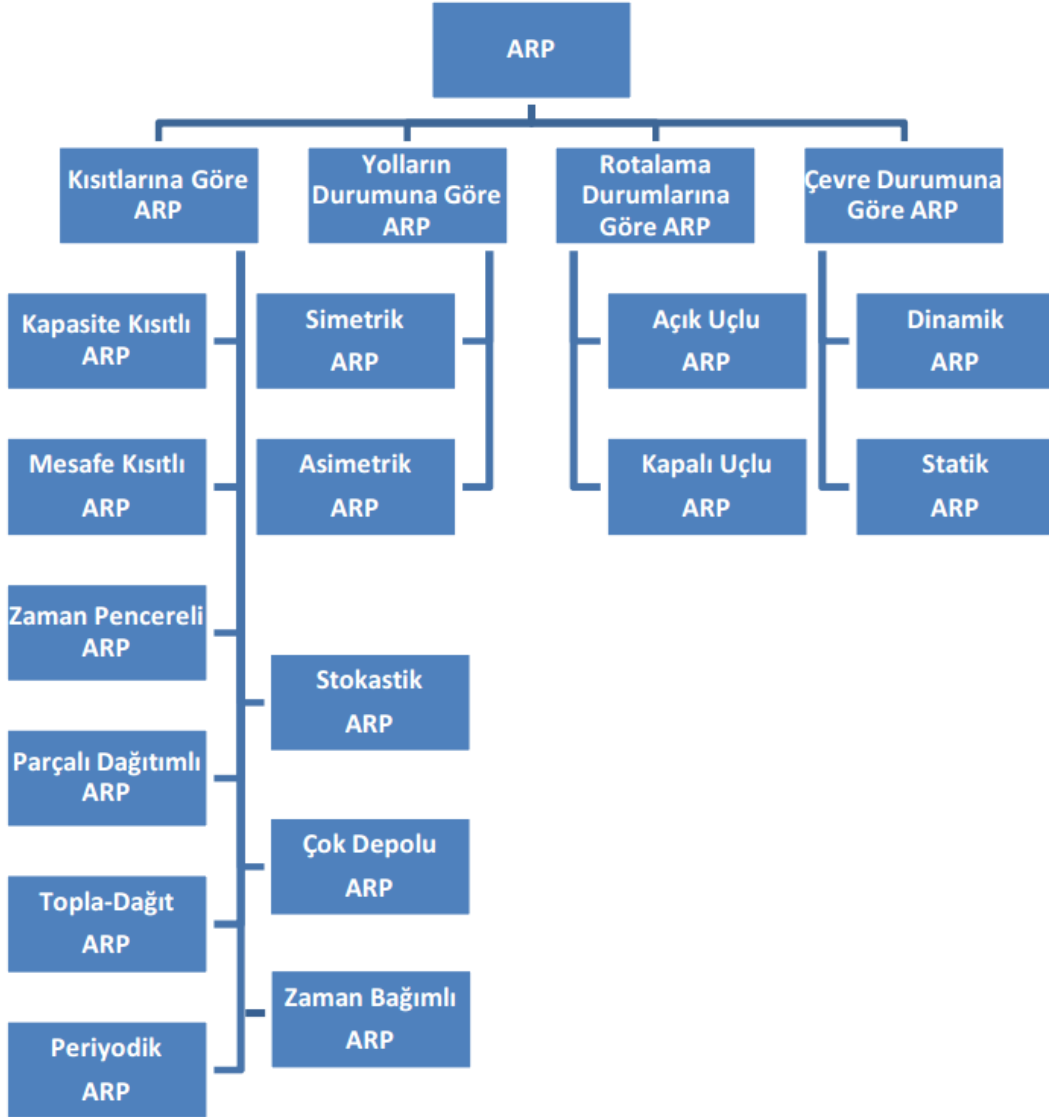
1. Noktalar arasında en yakın olanlar seçilmelidir. Bu sayede toplam gidilen yolun kısalması sağlanır.
2. Mümkün olduğunca farklı günlerde yapılacak müşteri teslimatları birleştirilmelidir. Bu sayede aynı rotalardaki müşterilere yakın tarihlerde teslimat yapılması engellenir.
3. Rota başlangıçları, depoya en uzak nokta ile yapılmalıdır.
4. Oluşturulacak rotanın görüntüsü damla şeklinde olmalıdır. Böylece uzak noktalara ulaşımında kazanç elde edilecektir.
5. Rotalara atanacak olan araçların yüksek kapasiteli olması tercih edilmelidir. Böylece toplam gider düşecek ve kazanç sağlanacaktır.

6. Rota dışında kalan konumlara ulaşmak için küçük araçlar tercih edilmelidir.
7. Mümkün ise dağıtım ve tedarik araçları aynı olmalıdır. Böylece toplam gider ve ihtiyaç duyulan süre düşecektir.
8. Dağıtım ve tedarik zamanları gerektiğinde yeniden kararlaştırılarak vakit tasarrufu sağlanmalıdır.

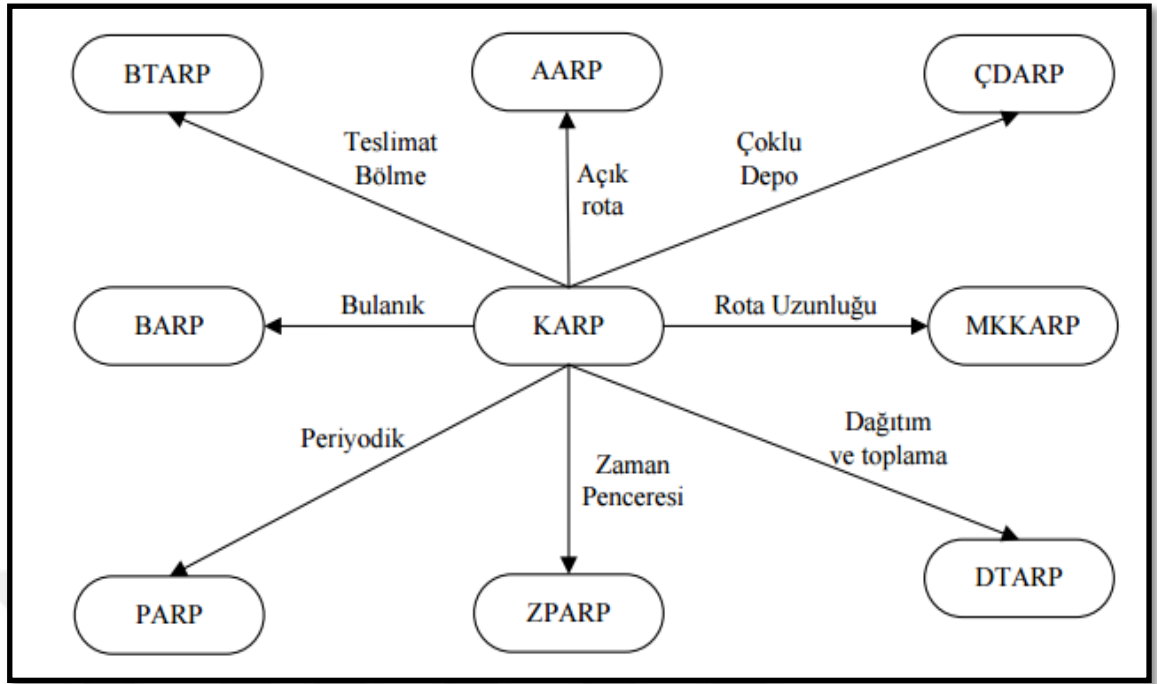
2.1.3. Araç Rotalama Problemi Çeşitleri

Her geçen gün araç rotalama problemi için doğru çözümlerin oluşturulması zorlaşmaktadır. Bunun sebebi, rekabet şartlarının artması ve çevre koşullarının değişmesi nedeni ile rotalama problemleri için kısıtlarının gittikçe artmasıdır. Araç rotalama problemlerinde karşımıza çıkan kısıtlar, farklı kapasitelerdeki araçlar, oluşturulan rotadaki müsaade edilen seyahat süresi, zaman aralıkları, konumlar arası farklı hız sınırlamaları, araç sürücülerinin dinlenme zamanları olarak belirtilebilir. Araç rotalama probleminde kısıtların artması problemin daha da karmaşık hale gelmesine neden olmaktadır (Erel, 1995).

Şekil 1.2' de ARP çeşitleri sınıflandırılmaları ve Şekil 1.3' te Araç Rotalama çeşitlerinin birbirleriyle ilişkileri gösterilmiştir.



Şekil 1.2: Araç Rotalama Problemi Çeşitleri (Koç Ç, 2012).



Şekil 1.3: Araç Rotalama Çeşitlerinin Birbiriyle İlişkisi (Toth ve Vigo, 2002a).

2.1.3.1. Kısıtlarına Göre Araç Rotalama Problemleri (KARP)

Tüm kısıtları modelin içine dahil ederek en iyi çözüme ulaşmak zor olduğundan, gerçek hayatta oluşturulan lojistik ve dağıtım sistemleri için bu kısıtlardan en önemli olanları seçilip kalanı modele dahil edilmeyerek en iyi sonuca ulaşılmaya çalışılmaktadır. Seçilen bu önemli kısıtlar, yükleme boşaltma durumu, mesafe kısıtı, araç kapasitesi ve zaman pencereleridir. Bu nedenle seçilen kısıtlar Araç Rotalama Problemlerinin isimlerini belirlemektedir, bu ARP türleri aşağıdaki gibi olabilir.

2.1.3.1.1. Kapasite Kısıtlı Araç Rotalama Problemleri (KKARP)

Müşteri taleplerinin önceden bilindiği ve araçların belirli bir kapasitesinin olduğu Dağıtım ağı'nın en iyilenmesi noktasında önemli problemlerden olan Kapasite Kısıtlı ARP en yaygın türdür. En basit hali ile problemde tüm araçlar eşit kapasiteye sahiptir ve bir depodan rotaya başlar, yine dönüşleri bu depoya olur. Müşteri talepleri birkaç defa da değil tek bir defada teslim edilmektedir.

Tüm müşterilerin birbirinden farklı ve sabit bir talep miktarı olan problemler, literatürde en çok yer verilen ve üzerinde çalışılan problemlerdir. Müşterilere bir araç ile ve yalnızca bir defa uğranılabilir. Depo ve müşteriler arasındaki mesafelerin simetrik

olduğu kabul edilerek, seyahat sürelerinin mesafeler ile doğru orantılı alınması sağlanmaktadır. Çoğu problemde de olduğu gibi ele alınan bu problemde de asıl amaç, araçlar ile en kısa mesafenin kat edilmesini sağlamaktır (Lin, 2009).

2.1.3.1.1.1. Eşit Talepli Kapasiteli Araç Rotalama Problemleri

Kapasiteli araç rotalama problemlerinde taleplerin eşit olduğunun kabul edildiği bir senaryo kuralım: Herhangi bir müşterinin w kadar bir ürün talep ettiğini varsayalım. Bu talep birden çok araç ile ulaştırılabilirse, o halde bu talebin aynı konumdaki w kadar müşteriye 1'er birim ürün ulaştırmak olarak düşünebiliriz. Bu şartlarda yeni problemde kapasite sınırı, bir aracın ziyaret edeceği maksimum müşteri sayısı olarak ifade edilebilir. Burada kapasite $C \geq 1$ ' dir. Bu halde, eğer talepleri bölmeye izin varsa ki bu durum genelde istenmez, tüm müşterilerin 1 birim kadar talebi olduğu durumda tek araç en çok C kadar müşteriye talebini ulaştırabilir. Bu nedenle modelin ismi; Eşit Talepli Kapasite Kısıtlı Araç Rotalama Problemi ya da Ayrılabilir Kapasite Kısıtlı Araç Rotalama Problemi olarak değişiklik gösterebilmektedir.

1985'te Haimovic ve Rinnooy Kan tarafından ortaya atılmış daha sonra 1999'da Altinkemer ve Gavish tarafından KKARP için geliştirilmiş basit bir sezgisel, bir "Gezgin Satıcı Turunu" bölümlere ayırmak, her bölüme bir araç göndermek ve her bölümde en fazla C adet durak noktası oluşturmaktır. Bu sezgisel "Tekrarlı Tur Bölümlendirme" adını alır ve depo ve tüm müşteriler arasında dolaşan bir Gezgin Satıcı Turu ile başlar. Depodan başlayarak tüm müşterileri sırasıyla ziyaret ederken, her bölümde en fazla C adet müşteri olacak şekilde bir bölümlendirme yapılır ve her bölümün ucu tekrar depoya bağlanır. Her bölüm C adet veya daha az müşteri içerebilir. Bu durum, bu problem için fizibil bir çözüm arz edebilir (Glynn ve Robinson,1997).

2.1.3.1.1.2. Eşit Olmayan Taleplere Sahip Kapasiteli Araç Rotalama Problemleri

Kapasiteli araç rotalama problemlerinin bu türünde ise, w_i müşterinin talep miktarını ve kapasite sınırını, i ise her müşteriye, bir aracın taşıyabileceği toplam yükün C 'den fazla olamayacağını ifade eder. Amaç fonksiyonu çoğu problemdeki gibi araçların en kısa mesafe ile taşıma rotasını tamamlamasının sağlanmasıdır.

Bu tipteki araç rotalama problemlerinde, müşterilerin yaptığı talepler birden çok araca dağılamaz ve her müşteri için sadece tek araç ataması yapılacağı varsayılır. Kapasiteli araç rotalama problemlerinin daha genel bir versiyonu olan bu tip problemlere bazen

“ayrılmaz talepli” KKARP de denir. Bir müşterinin talebini parçalara bölmek çoğunlukla firmalar için uygulanabilecek bir çözüm değildir ve bu durum hem müşteri hizmeti veya finansal yönetim güçlükleri yaratabilir, hem de fiziksel olarak imkansız olabilir.

Bu tipteki araç rotalama modelleri, pratikte de en fazla karşılaşılan problemlerdir. Çünkü gerçek hayatta pek çok kısıt araç rotalama problemlerine dahil edilir ve problemin çözülmesi zorlaşır. Bu amaçla akademik alanda bu konuda pek çok araştırma yapılarak, bu problemlerin çözülmesi için çeşitli algoritmalar geliştirilmiştir. Bu tip problemlerin çözülmesinde sıklıkla kullanılan metodlar, GSP (Gezgin Satıcı Problemi), Savings (Tasarruf) Tekniği, Sweep Metodu vs... dir (Glynn ve Robinson, 1997).

2.1.3.1.2. Mesafe Kısıtlı Araç Rotalama Problemleri (MKARP)

Mesafe Kısıtlı Araç Rotalama Problemlerinde (MKARP) rotalarda seyahat edecek her araç için kısıt olarak maksimum mesafe bulunmaktadır. Önceden tanımlanan modele mesafe kısıtı aşağıda verilen (2.7) nolu denklem ile ilave edilerek yeni model oluşturulmaktadır. Denklem içinde araçların kat edebileceği maksimum mesafe L değeri ile temsil edilmektedir.

$$k \in \{1, \dots, M\} \text{ için } \sum_{i=0}^N \sum_{j=0}^N d_{ij} \sum_{j=0, j \neq i}^N X_{ijk} \leq L \quad (2.7)$$

Kapasite Kısıtlı Araç Rotalama Problemi’nde (KKARP) olduğu gibi Mesafe Kısıtlı Araç Rotalama Problemine (MKARP) ait çeşitli versiyonlar literatürde yer almaktadır. Mesela araçların farklı tipleri için farklı mesafe kısıtları ($L_k, k=1, \dots, M$) mevcut olabilir. Aynı zamanda mesafe kısıtı değil de seyir süresi sınırlaması (mesafeye orantılı) probleme dâhil edilirse, araçlar bir müşteriden diğer müşteriye gitmeden önce s_i süresi kadar beklemek zorunda kalacaktır. Eğer bu sınırlama probleme eklenirse (2.7) nolu kısıt denklemi aşağıdaki şekilde (2.8) nolu denklemler olarak değişecektir. (Toth ve Vigo, 2002b):

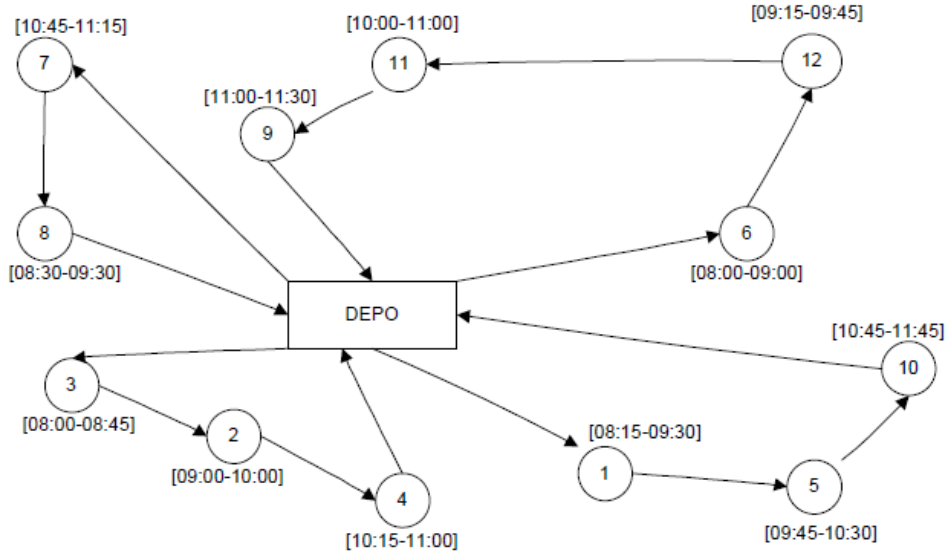
$$k \in \{1, \dots, M\} \text{ için } \sum_{i=0}^N \sum_{j=0}^N d_{ij} \sum_{j=0, j \neq i}^N pX_{ijk} + \sum_{i=0}^N \sum_{j=0}^N \frac{s_i + s_j}{2} \sum_{j=0, j \neq i}^N X_{ijk} \leq T \quad (2.8)$$

Aracın seyahat süresini aracın yol aldığı toplam mesafe üzerinden bulmak için yapılan hesaplama için p parametresi kullanılmaktadır. Müşteriye hizmet süresinin (s_i) ikiye bölünmesinin nedeni ise araç rotası üzerindeki yolları sıralarken, rota içindeki her müşterinin hem yolun başında hem de diğer yolun sonunda bulunması ve bundan dolayı denklemdeki toplam ibaresinde aynı müşteriye iki defa yer verilmesidir.

Hem araç sayısının hem de seyahat süresinin sınırlı olduğu durumlarda, problem çözüm sonucu tüm noktaları kapsamama gibi bir sorunu ortaya çıkarabilir. Şayet tüm müşteri konumlarına kesinlikle gidilmesi gerekiyorsa ve araç sayısında artış imkansız ise, bu durumda seyahat süresi sınırının ihlal edilmesi gerekebilir. Böyle bir durum ile karşılaşıldığında ihlal edilen her birim süre karşılığında ceza maliyeti eklenerek, toplam ceza maliyeti daha da azaltılabilir (Erel, 1995).

2.1.3.1.3. Zaman Pencereci Araç Rotalama Problemleri (ZARP)

Pek çok dağıtım sisteminde her müşteri, talebi karşılayacak ürün miktarının yanında bu dağıtımın tamamlanacağı bir zaman dilimi belirleyebilir. Bu zaman dilimine "Zaman Penceresi" denir (Glynn ve Robinson, 1997). Bu tip bir probleme her araç için depoda başlayan, farklı müşterilere uğrayan ve tekrar depoya dönen bir rota belirlenerek çözüm oluşturulmaktadır. Araçlar, rotalarında bulunan müşterilere önceden tanımlanmış zaman aralıkları içerisinde teslimata başlayabilmelidirler. Her müşteri için belirlenmiş olan en erken ve en geç teslimat başlangıç zamanları, zaman aralığı kısıtının sınırlarını oluşturmaktadır. Araç müşteriye en erken teslimat anından önce varmış ise en erken teslimat anına kadar bekleme yapmak durumundadır. Aracın müşteriye en geç teslimat anından sonra varış yapması durumunda müşteri teslimatı kabul etmeyeceğinden uygun olmayan bir çözüm oluşturulmuş olacaktır.



Şekil 1.4: Zaman Aralıklı Araç Rotalama Problemi Örneği (Demircioğlu, M., 2009).

Araç Rotalama Problemi türlerinden zaman pencereci araç rotalama problemi için bir örnek Şekil 1.4 ' te verilmiştir. Problemin tanımı sevkiyatın gereksinimlerini yerine getirecek şekilde rotanın oluşturulması olarak yapılabilir. Müşteri taleplerinin tamamı sevkiyatı yapılacak ürünün boyutunu, depo ve müşteri konumları ürünlerin nereden alınacağı ya da nereye dağıtılacağını, belirlenen zaman penceresi kısıtı da dağıtım ya da toplama için zaman aralıklarını belirlemiş olur. Zaman Sınırlı Araç Rotalama probleminde seyahat için katedilen mesafe yerine seyahat süresi de kullanılır. Bu durumda araç müşteri konumuna zamanından erken ulaştığı için, bu konumda bir bekleme yapması gereklidir (Demircioğlu, M., 2009).

Yüksek düzeyde uygulanabilir ve ekonomik öneminden dolayı bu problem pek çok çeşidiyle literatürde yer bulmuştur. Bu konu ile ilgili çalışma için Solomon ve Desrosiers (1998) incelenebilir. Pek çok çalışma problemi empirik açıdan ele alınırken, pek az çalışmada da problemin analitik yönü incelenmiştir. Bu konu üzerinde çalışmalar yapan diğer akademisyenler, Federgruen ve von Ryzin (1992), Brameil ve Simchi – Levi (1996) dır (Karahana, 2003).

2.1.3.1.4. Parça Dağıtım Araç Rotalama Problemleri (PDARP)

Bu problemde toplam maliyetin azalmasını sağlayacak ise, aynı müşteriye farklı araçların uğraması kabul edilebilir. Bu noktada, müşteri siparişinin büyüklüğü araç kapasitesine eşit ise, kritik bir konu olarak klasik ARP probleminin genişletilmesi ele alınmalıdır.

PDARP' de, ARP'de ulaşılan optimum çözümlere ulaşmak çok daha zor olabilir.

Amacı: PDARP'de amaç müşteri taleplerinin karşılanması ve araçların akışı için ihtiyaç duyulan toplam süreyi en iyi hale getirmektir.

Uygunluk: ARP kısıtlarının tamamı (aynı müşteriye birden çok aracın hizmet vermesi dışında) sağlanabiliyorsa çözüm "uygulanabilir" olarak kabul edilir.

Formülasyon: ARP'yi PDARP'ye dönüştürmek için, her müşterinin siparişini daha küçük, bölünemeyen siparişlere ayırmak ve dağıtımların parçalanmasını sağlamak kolay bir yoldur. Problem çözümü için oluşturulacak bütün rota maliyetleri düşürülür (Şeker, 2007).

2.1.3.1.5. Topla-Dağıt Araç Rotalama Problemleri (TDARP)

Eş Zamanlı Topla-Dağıt ARP'de, 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. Araçların, müşteri ziyaretleri sırasında dağıtım ve toplama aynı anda yaptıkları için araç kapasitelerinin korunması gibi zor bir durum bulunmaktadır. Tüm müşterilere teslimat yapılacak ürün miktarı, ya da toplama yapılacak miktar önceden bilinmektedir. Bianchessi ve Righin (2007) '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. 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.

2.1.3.1.6. Periyodik Araç Rotalama Problemleri (PARP)

Periyodik Araç Rotalama Probleminde (PARP)' de bir dönemin planlaması önceden yapılır ve müşteriler bu dönemde birden çok defa ziyaret edilmektedir. Müşteri ziyaret adetleri müşterinin talep miktarına ve müşterinin ürünleri stoklayacağı alanın kapasitesine bağlı olarak değişiklik göstermektedir. Müşterinin talebi çok ise talebi az olan müşteriye göre daha çok, stoklama yapacağı alan kapasitesi küçük ise de stoklama kapasitesi daha yüksek olan bir müşteriye göre daha çok ziyaret edilecektir. Bu tür problemler alkolsüz içecek dağıtımı, bakkaliye ve atık toplama gibi sektörlerde meydana gelmektedir (Hemmelmayr, 2007).

2.1.3.1.7. Stokastik Araç Rotalama Problemleri (SARP)

Stokastik Araç Rotalama Problemi (SARP), kısıtlardan bir ya da fazlasının olası olduğu koşulların klasik araç rotalama problemine uygulandığı bir problem türüdür.

SARP' nin 3 kadar çeşidi vardır (Koç, 2012).

- Stokastik Müşterili: p_i olasılığına göre i kadar müşteri vardır. $1-p_i$ olasılığına göre ise yoktur.
- Stokastik Talepli: Müşteri taleplerinin her biri q_i olası bir değişken olacaktır.
- Stokastik Zamanlı: Dağıtım Zamanları s_i ve Seyahet Süreleri t_{ij} rassal değişkenler olacaktır.

SARP' de çözüme ulaşabilmek için geliştirilen yöntem iki aşamadan oluşmaktadır. İlk aşamada olası değişken değerlerinin bilinmediği bir çözüm oluşturulur. Bu aşamanın amacı değerleri belirlenmiş olan olası değerdeki müşteri taleplerini karşılayabilmek için gereken rota zamanı ve sevkiyat süresini en aza indirmektir. Bazı değerlerin olası olduğu bu problemlerde, olası tüm değerlerdeki kısıtların yerine getirilmesi beklenemez. Bu sebeple problem çözümü için karar verici ya sadece bazı kısıtların yerine getirilmesini kabul edebilir ya da kısıtların herhangi biri yerine getirilemediğinde sorunun düzeltilmesi için probleme ek aksiyonlar ilave edebilir (Koç, 2012).

2.1.3.1.8. Çok Depolu Araç Rotalama Problemleri (ÇPARP)

Önceden ele alınan problemlerde araçlar tek depo üzerinden rotalarına başlamaktadır, ancak bu problem türünün adından da anlaşılacağı gibi araçlar birden çok depodan rotalarına başlayabilmektedir. Bu problemde müşteri ve depo lokasyonları belirlenmiş

ve tüm depo kapasiteleri müşterilerin bütün taleplerine cevap verebilecek kapasitededir. Probleme araçlar depolara kayıtlı olduğundan, araç ancak çıkış yaptığı depoya geri dönebilecektir. Eğer tek depodan daha fazla depoya sahip bir firmanın rota düzenlemesi yapılıyor ise birden fazla depo denklemini hazırlanan modele eklemek gerekir. Bu türdeki ARP' de en iyi çözüme ulaşabilecek herhangi verimli bir metod yoktur ve bu nedenle problem NP-zor sınıfına girmektedir (Ho, 2008).

2.1.3.1.9. Zaman Bağımlı Araç Rotalama Problemleri (ZBARP)

İlk olarak Dantzig ve Ramser ile 1959 yılında gündeme gelen ARP, bugüne dek birçok alt dala kavuşarak hayatımızdaki problemlere uyarlanabilecek düzeye getirilmeye çalışılmıştır. Hayatımızın önemli problemlerine ARP'nin çözüm bulmak için uğraşan türlerinden birisi de ZBARP' dir.

Bu alandaki denemelerin çoğunda konumlar (depo ve müşteri konumları) arası seyahat süresi sabit olarak kabul edilmektedir. Fakat şehir içinde yaşadığımız, günün belirli zamanlarındaki olası trafik yoğunluğu nedeniyle araçların hız ve seyahat süreleri, kullanacakları yol ve bu yolun kullanım zamanına göre farklılık göstermektedir. Bu tür kısıtların da bulunduğu problemler literatürde ZBARP olarak anılmaktadır.

Günden güne önemi daha da artan taşımacılık sistemlerinin temel bileşenlerinden biri olan araç rotalama, ilk gündeme geldiğinden bu yana çeşitli çalışmalarla zenginleştirilen bir problemi türü haline gelmiştir. Zaman Bağımlı Araç Rotalama Problemi'nde depo ve müşteri konumları arasındaki seyahat sürelerinin günün her saatinde farklılık gösterdiği ve her müşteri için önceden belirlenmiş bir hizmet zamanı olduğu zaman penceresi kısıtı altında problem ele alınmaktadır. Bu nedenle, planlama yapılan gün ya da dönem aralıklara ayrılarak ve belirlenen bu aralıklarda her yola sabit bir hız tanımlanmaktadır. Herhangi bir müşteriden çıkacak araç için çıkış zamanına göre ve kullanılacak yola göre belli bir hız ile hareket etmektedir. Bir sonraki müşteriye seyahat sırasında belirlenen zaman aralıklarından sonrakine geçiş oluyorsa, aracın kalan seyahat hızı bu aralık için tanımlanmış olan hız üzerinden hesaplanacaktır. Bu durum gerçek hayatta karşılaşılan problemlere oldukça benzemektedir. Örnek olarak iş yerlerinin fazla olduğu yerlerde sabah saatleri, öğle araları ve iş çıkışı saatleri yoğun trafik, bunun dışındaki zamanlarda ise yoğun olmayan trafik oluşması muhtemeldir.

Yoğun trafik yaşanan bir şehirde, iki konum arasında geçen yolculuk süresi aracın hızı sabit kabul edilemeyeceğinden fonksiyon olarak alınmaz. Trafik yoğunluğunun gün içinde değişmesi aracın hızını da değiştirecek buna bağlı olarak da toplam seyahat süresi de değişecektir. Bu değişikliklere neden olan olayların birinci bölümü; hava durumu, trafik kazaları veya olası değişikliklerdir. Diğer bölümü ise trafiğin en yoğun olduğu zamanlardaki dönemsel ve mevsimsel yoğunluklardır (Malandraki ve Daskin, 1992).

2.1.3.2. Yolların Durumuna Göre Araç Rotalama Problemleri

Araç rotalama problemi, müşteri konumları veya depo konumu arasındaki mesafenin gidiş ve geliş mesafesi bakımından eşit olup olmaması diye ikiye ayrılır (Erol, 2006).

2.1.3.2.1. Simetrik Araç Rotalama Problemleri

Oluşturulan birden çok rotanın toplam seyahat maliyetinin en aza indirilmesi problemine simetrik araç rotalama problemi denilmektedir. Bu rotalama probleminde rotalar depodan başlayıp depoda biter, her müşteri konumu sadece bir rota üzerinde bulunur, bir rota üzerindeki müşterilerin talep toplamları aracın kapasitesini aşmaz ve her rota için belli bir mesafe sınırı vardır (Barnhart, Laporte, 2007).

2.1.3.2.2. Asimetrik Araç Rotalama Problemleri

Araç rotalama problemlerinde bazen müşteri ve depo konumları arasındaki mesafelerin birbirine eşit olmadığı durumlarla karşılaşılabilir. Eğer araç rotalama problemi bu türde ve kapalı uçlu ise depodan sonraki ilk müşteri konumu önemlidir. Bu nedenle rotanın dönüş yönünü belirleyebilmek için her iki yönün mesafesinin karşılaştırılması gerekmektedir. Bu tarz kısıtları içeren problemlere asimetrik yollu araç rotalama problemleri denilmektedir (Erol, 2006).

2.1.3.3. Rotalama Durumlarına Göre Araç Rotalama Problemleri

Araç rotalama problemleri, araçların depodan çıkıp depoya dönmesi veya aracın rotasının depodan bağımsız olarak rota üzerindeki en son müşteri konumunda sonlanması durumlarına bağlı olarak açık ya da kapalı uçlu olarak iki şekilde incelenir (Erol, 2006).

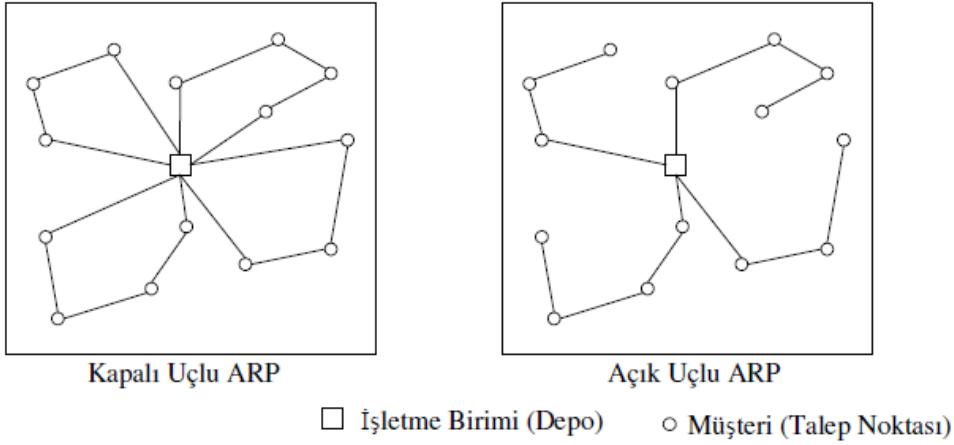
2.1.3.3.1. Açık Uçlu Araç Rotalama Problemleri (AUARP)

Açık uçlu araç rotalama problemlerine (AUARP), onlarca yıldır araştırmacılar tarafından etkin bir çözüm yolu sunmadığı için fazla ilgi gösterilmemiştir. Talebi ve

konumu belli olan bir dizi müşteri için kullanılan araç filolarının rotalamasında açık uçlu araç rotalama problemi kullanılabilir. Bu problemde rota bir dizi müşteri konumundan oluşur, rotalar depoda başlar ve en son müşteride biter (Wang ve diğ., 2006). Bu problem türü genelde araç kiralamalarında uygulanır ve aracın sadece gidiş rota planlaması yapılır (Tüfekçier, 2008).

2.1.3.3.1. Kapalı Uçlu Araç Rotalama Problemleri (KUARAP)

Rotaların depodan çıkıp yine aynı depoya döndüğü problemlere kapalı uçlu araç rotalama problemleri (KUARAP) denilmektedir. Genelde literatürde yapılan çalışmalar bu rotalama problem türlerindedir. Geliştirilen yöntemler çoğunlukla kapalı uçlu araç rotalama problemi ele alınarak çözüldüğünden kıyaslandığı yöntem de yine bu tür bir problemidir (Erol, 2006).



Şekil 1.5: Açık ve Kapalı Uçlu Araç Rotalama Problem Şekilleri (Erol, 2006).

2.1.4.4. Çevre Durumu Bakımından Araç Rotalama Problemleri

Olaylara sabit ya da değişken başka bir deyişle verilerin belirli ya da belirsiz olma durumlarına göre yaklaşan ve en çok kullanılan problemlerden biridir. Bu kriter doğrultusunda bu tür araç rotalama problemleri Dinamik ARP ve Statik ARP olmak üzere iki çeşitte incelenebilir:

2.1.4.4.1. Dinamik Araç Rotalama Problemleri (DARP)

Aracın rotasında ilerlediği sırada yeni bir müşteri noktası çıkması, müşteriye ait talebin değişmesi, rota üzerindeki yollardan bazılarının kullanılamaz hale gelmesi, trafik

yoğunluğu gibi nedenlerle seyahat süresi ya da mesafenin artması durumlarında sorunun çözümü için hızlı kararlar alınması gerekmektedir. Bu araç rotalama probleminin değişken olan bileşenleri, müşterinin talebinin değişmesi ya da belirsiz değişkenlere bağlı yolculuk süreleri olarak öne çıkmaktadır. Dinamik araç rotalama problemi pratikte genel olarak müşterisine saha satış ya da kurye vasıtasıyla hizmet veren firmalarda önem kazanmaktadır.

2.1.4.4.1. Statik Araç Rotalama Problemleri

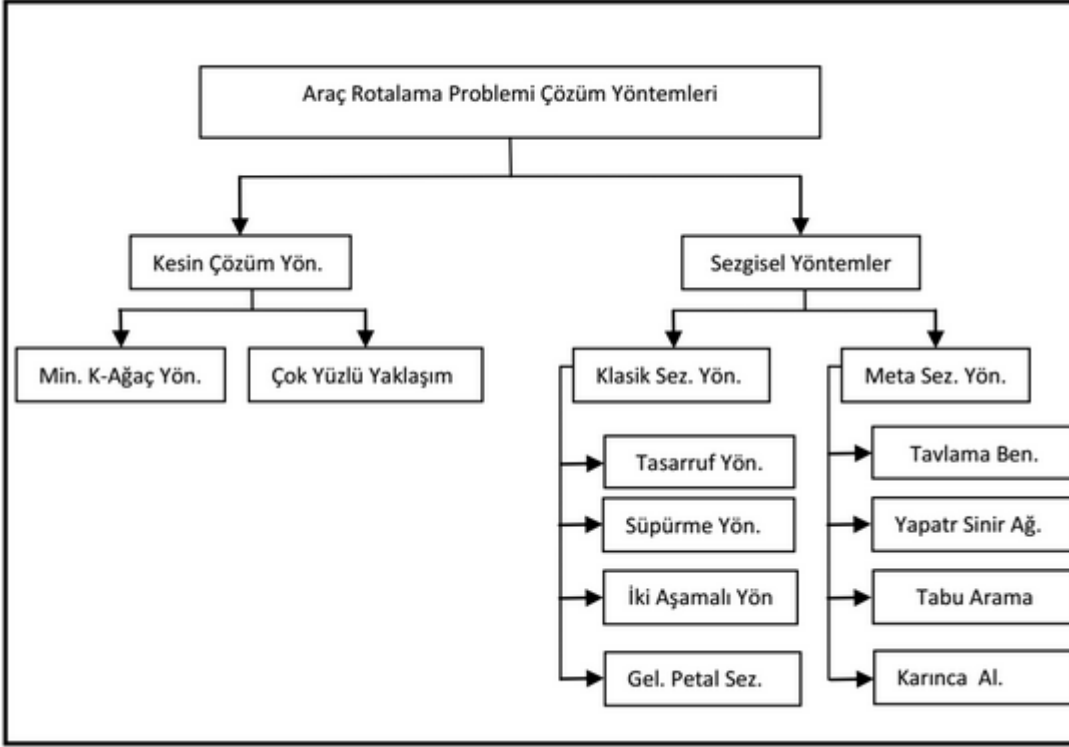
Bu problem türünde problemin çözümüne başlanmadan gereken bütün veriler (araç kapasiteleri, maliyetler, kısıtlar ve talepler) bellidir ve bu veriler problem çözümlenirken değişmez, kesindir. Genellikle statik araç rotalama problemi literatürde, üzerinde çalışılan bir problem olup belirli ARP olarak da yer alabilmektedir. Statik ARP çözümleri talebin miktarı ve zamanı önceden bilinen rotaların oluşturulmasında ve genel olarak servis sistemleri değerlendirilirken kullanılır (Potvin, 2004).

2.1.4. Araç Rotalama Problemleri İçin Çözüm Yöntemleri

Araç Rotalama Problemleri içerisinde alt bir problem olarak gezgin satıcı problemi içerdiğinden NP-zor bir problemdir. Gezgin satıcı problemine göre çözümü oldukça zor bir problemdir çünkü içerisinde iki alt problem barındırır. Birinci problem müşterileri önceden bilinmeyen sayıda kutuya (rotaya) doldurmayı (kümelenlendirmeyi) gerektiren kutu doldurma (binpacking) problemidir. Daha sonra her rota için içerdikleri tüm müşterileri ziyaret eden bir gezgin satıcı problemi çözümlenmelidir (Kılıç, 2008).

Araç rotalama problemleri 1950'li yıllardan beri önemli bir araştırma konusu olup, çeşitli çözüm yöntemleri geliştirilmiştir. Bu çözüm yöntemleri temel olarak 2 sınıfa ayrılır. Bunlar kesin (exact) ve sezgisel (bulgusal, heuristic) yöntemler olarak adlandırılırlar. Kesin yöntemler ile optimum çözümler bulunur. Fakat büyük boyutlu problemlerde, problemin çözüme kavuşturulması oldukça vakit alabilmektedir. Sezgisel yöntemlerde ise hedef, optimuma yakın kabul edilebilir bir çözüme polinomyal zaman içerisinde ulaşmaktadır. 1960' lı ve 1970' li yıllarda, araç rotalama problemlerinin çözümü için rota oluşturma, rota iyileştirme ve iki aşamalı sezgisel yöntemler üzerinde yoğunlaşmıştır. 1980' li yıllarda, matematiksel programlama tabanlı yöntemler geliştirilmiştir. Bu yöntemler daha fazla hesaplama gücüne gereksinim duymakla beraber, yüksek kalitede çözümler üretmişlerdir. Bu 30 yıllık dönemin sonunda,

yaklaşık 50 müşteriye kadar olan bazı problemler optimal olarak çözülebilmektedir. 1990'lı yıllarda, araştırmaların odağı modern sezgisel yöntemlere kaymıştır. Modern sezgisel yöntemler literatürde metahöristikler olarak da geçmektedir. Bu yöntemler arasında yasaklı arama, tavlama benzetim, karınca kolonileri ve genetik algoritmalar sayılabilir (Crainic ve Laporte, 1998). ARP çözüm yöntemleri Şekil 1.6' da gösterilmiştir.



Şekil 1.6: Araç Rotalama Problemi Çözüm Yöntemleri (Demircioğlu, M., 2009).

2.1.4.1. Kesin Yöntemler

ARP için kesin çözüm yöntemleri, gezgin satıcı problemi yönteminin iletilmesi ile elde edilmiştir (Düzakın ve Demircioğlu, 2009). En iyi çözümler, kesin çözüm yöntemleri ile bulunur. Ama problemin büyüklüğüne bağlı olarak problemin çözüm süresi üstel bir şekilde artar. Genel olarak küçük ve orta büyüklükteki problemler pratik bir şekilde çözümlenir. Araç rotalama problemlerinin hepsini başarılı bir şekilde çözen kesin yöntem bulunmamaktadır (Ropke, 2005; Kumar, 2012).

Kesme Düzlemi:

Bu algoritmada ana fikir, doğrusal programlama ile elde edilen optimum çözüm alanındaki bazı parçaların kesip çıkartılarak, optimum tamsayılı çözümü simpleks yöntemle oluşturmaktır (Pan, 2015). Farklı bir ifade ile tam sayıya ulaşabilmek için kısıtlar ilave edilir. Kısıtlar ilave edildiğinde, ulaşılan yeni çözüm tam sayılı ise optimal çözüm elde edilmiştir. Fakat halen küsüratlı ise tam sayılı çözümü buluncaya kadar kısıt ilave edilmeye ve çözüme ulaşılması sürdürülür (Toth ve Vigo, 2002).

Dal ve Sınır Algoritması

Bu algoritmada (Balas ve Toth, 1983), gezgin satıcı problemi, alt tur (depoda başlayıp bitmeyen turlar) gibi engelleyici kısıtlar ortadan kaldırılarak problem atama işlemine dönüştürülür ve ilk olarak Macar Yöntemi ile çözülmeye çalışılır. Eğer alt tur problemi ortaya çıkarsa; en kısa döngü engellenecek şekilde kısıtlar dallara ayrılır. İstenmeyen rotaların düzenlenmemesi için ceza katsayısı yerine büyük M sayısı yerleştirildikten sonra maliyet matrisi yeniden çözümlenerek aday çözümler tüm dallar için oluşturulur. Aynı işlemler tüm dallar için tekrarlanarak en iyi çözüm seçilir.

Dal ve Kesme Algoritması

Dal-Kesme yöntemi, dal-sınır ve kesme düzlemi yöntemlerinin bir birleşimidir (Araque ve diğ., 1994). İlk önce problemin çözümü için doğrusal programlama kullanılır. Model oluşturabilmek için bu aşamada kısıtlar ve amaç fonksiyonu belirlenerek yazılır. Çözüm sırasında oluşacak alt turların sıfıra (0) eşitlenmesi ile problem dallara ayrılmış olur. Daha sonra karşılaşılabilecek alt dallarda da araç adedi gibi kısıtları karşılayabilmek için model alt tur engelleme kısıtlayıcısı ile desteklenerek en iyi sonuç bulunmaya çabalanır (Başkaya ve Öztürk, 2005).

Dinamik Programlama

Bu programlama tipinde, problem birbiri ile ilgisiz alt problemlere ayrılmaktadır. Araç rotalama probleminin çözümüne ulaşmak için daha önce oluşturulan alt problem çözümleri kaydedilir ve daha sonra ihtiyaç durumunda bu çözümler kullanılır (Chauhan ve diğ., 2012).

2.1.4.2. Sezgisel Yöntemler

Gayet kısıtlı bir alanda arama yapmasına karşın gayet kısa bir zaman içerisinde iyiye yakın çözüm üretmekte sezgisel yöntemler öne çıkmaktadır. Aynı zamanda bu yöntemler çoğunlukla uygun ilk çözümleri hızlıca bulurlar ve bu çözümleri daha iyi hale getirmeye uğraşırlar. Araç rotalama problemi için üretilen sezgisel yöntemlerden çoğu simetrik yollu kapasite kısıtlı araç rotalama problemi için geliştirilmişlerdir

Algoritmanın anlamı, bir problemin çözülmesi için gerekli olan davranışların tanımlanmış ya da daha sonra tamamlanacak veri modeli temel alınarak basamak basamak oluşturulması ve bunun herhangi bir programlama dili ile bilgisayar ortamında kodlanmasıdır.

Problem çözümüne ulaşmak için sırayla uygulanacak basamakların ve kuralların kesin olarak tanımlanması algoritma ile sağlanır.

Buradan da anlaşılacağı gibi algoritma genel olarak tek bir işin kotarılması üzerine yoğunlaşmıştır. Örneğin bir kümenin elemanlarını sıralama, bir graf üzerinde en kısa yolun bulunması, bir matrisin determinantının alınması gibi algoritmalar tek bir amaca yöneliktir.

Algoritma mekanik davranan kişiye ve makineye bir takım verilerden yola çıkarak ve sonlu sayıda aşamalardan geçerek belli bir problemi çözüme imkanı veren, çok kesin komutlar bütününde oluşmaktadır. Bir algoritmanın çalışmasındaki mutlak zorunluluk, her türlü belirsizlikten arınmış olmasıdır. Bir algoritmanın yürütülmesi, her biri komutla belirlenen bir etkiler dizisi oluşturur ve bir dizi önceki komutun yürütülmesinin sona ermesiyle birlikte yürütülmeye başlar.

ARP tipi problemlerin çözümünde genellikle sezgisel yöntemler kullanılır. Ancak bir problem için geçersiz olan sezgisel yaklaşım diğeri için başarılı sonuçlar verebilir. Sezgisel yaklaşımın temel adımları aşağıdaki gibidir:

- Mümkün olabilecek durumların içinde herhangi birinin seçilmesi
- Durumun değiştirilmesi için seçilen duruma uygun gidişlerin uyarlanması
- Durumun gözden geçirilmesi
- İhtiyaç duyulmayan durumların silinmesi

- Çözümüne ulaşılan durumlarda işlemlerin durdurulması, henüz çözümün bulunamadığı durumlarda ise değerlerin yenilenerek işlemlerin tekrar edilmesi.

Sonuca ulaşmak için veya belirlenmiş bir hedefi gerçekleştirmek için birçok işlem içinden etkili olanı seçebilmek için belirlenmiş bilgisayar metotlarına ya da tanımlanmış kriterlere sezgisel algoritmalar denilmektedir. Bu tür algoritmalar en iyi sonucu garanti etmezler, ancak en iyi çözüme en yakın çözümü garanti edebilirler, buradan bu algoritmaların yakınsama özelliğine sahip olduğu anlaşılır (Erol, 2006).

Sezgisel Yöntemlerin Sınıflandırılması

Araç rotalama problemi sezgisel algoritmaları genellikle üç sınıfa ayrılır (Eryavuz ve Gencer, 2001):

1. Tur Kurucu Sezgisel Yöntemler
2. Tur Geliştirici Sezgisel Yöntemler
3. İki Aşamalı Metotlar

2.1.4.2.1. Tur Kurucu Sezgisel Yöntemler

Bu yöntemde, ilk çözüm uygun olmayan sıralamalarla başlatılır, her adımda iki nokta arasında yeni bir dal oluşturularak uygun çözüme ulaşılmaya uğraşılır. Dal oluşturulurken bazı maliyet tasarrufları kullanılır ve araç kapasite kısıtı dikkate alınır. Bu yöntemler Clarke ve Wright' ın, Dantzig ve Ramser' den esinlenerek ürettikleri Tasarruf (Saving) algoritması en çok tercih edilendir. Tasarruf (Saving) Algoritması yıllar içinde ana teması bozulmadan birçok algoritma olarak karşımıza çıkmıştır (Eryavuz ve Gencer, 2001). Bunun dışında yerleştirme (insertion), en yakın komşu (nearest neighbour) yöntemleri de bu kategoriye girer (Emel ve Taşkın, 2005).

2.1.4.2.2. Tur Geliştirici Sezgisel Yöntemler

Bu yöntemde, ilk çözüm uygun olan sıralamalarla başlatılır ve her adımda bu çözüm daha da geliştirilir. Her işlemde dal sıralamaları değiştirilerek bu sıralamanın maliyeti azaltıp azaltmadığı ve en uygun çözüme ulaşıp ulaşımadığı kontrol edilir (Eryavuz ve Gencer, 2001).

2.1.4.2.3. İki Aşamalı Metotlar

Bu yöntemin ilk basamağında, araçların kapasiteyi aşılmadan müşteri konumları rotaya eklenir. Sonraki basamakta ise bu araçlar için oluşturulan rotalar sezgisel yöntemler ile tekrar düzenlenir. İki aşamalı metotlar için önce grupta sonra rotala türündeki algoritmalar örnek gösterilebilir. İki aşamalı metotlara örnekler aşağıdaki gibidir:

- 1974 yılında Gillet ve Miller tarafından geliştirilen Süpürme (Sweep) algoritması,
- 1981 yılında Fisher ve Jaikumar tarafından geliştirilen algoritma,
- 1980 yılında Christofides, Mingozzi ve Toth tarafından geliştirilen algoritma (Eryavuz ve Gencer, 2001).

İki aşamalı yöntemler önce grupta sonra rotala (cluster first – route second), önce rotala sonra grupta (route first - cluster second) olmak üzere ikiye ayrılmaktadır (Emel ve Taşkın, 2005).

2.1.4.3. Metasezgisel Yöntemler

Metasezgisel yöntemler, çözüm uzayında etkin ve verimli olarak arama yapılabilmesi için, farklı yapılarda olan sezgisel algoritmaların birleştirilmesi ile meydana getirilmiş tekrarlayan problem çözme yöntemleridir. Metasezgisel yöntemler, her işlemde tek çözüm üzerinden ya da çözüm dizisi üzerinden farklı çözümler üretirler. Bu yaklaşımların çoğu olası çözümler içinde stokastik ancak bilinçli olarak tarama yaparlar (Blum ve Roli, 2003).

Bu algoritmaların amacı, olası çözümler içinde en iyi çözüme yakın aramalar yaparak yerel çözümlerden kurtulup optimum çözüme daha da yaklaşmaktır. Bu algoritmalar, bellek yapılarından ve karmaşık komşuluk kurallarından kombinasyonlar oluşturarak yeni çözümler üretmek için uğraşırlar. Bu algoritmaların hesaplama işlemleri, klasik algoritmalara nazaran daha uzun sürede ortaya çıkan çözümler diğer sezgisel algoritmalara kıyasla daha iyidir. Bu yöntemlerde, parametreler ele alınan problemin özelliklerine göre değişiklik gösterebilmektedir. Yöntemlerin bu özelliği her probleme uygulanabilmesi noktasında zorluklara yol açmaktadır. Klasik sezgisel algoritmaların doğadan esinlenerek geliştirilmiş hali metasezgisel algoritmaları oluşturur (Laporte vd., 2000).

Aşağıdaki genellemeler Blum ve Roli tarafından metasezgisel yöntemler için verilmiştir (Blum ve Roli, 2003):

- Tarama işlemi sırasında metasezgisel yöntemler klavuz stratejilerdir.
- Yöntemin amacı, olası çözümler içinde etkin bir araştırma yaparak en iyi ya da en iyiye yakın sonuçları bulmaktır.
- Bu yöntemlerin tarama teknikleri, yerel tarama kadar basitten karmaşık öğrenme tekniklerine varıncaya dek yaygınlaşır.
- Genelde bu algoritmalar, kesin değil yaklaşık çözümler vermektedir.
- Olası sonuçlar içinde yerel optimum noktalara takılmadan ilerlemeyi sağlayacak mekanizmalara sahiptirler.
- Metasezgisel yöntemler probleme özel olarak geliştirilmezler. Genellikle geliştirilen yöntem tüm zor problemlere uygulanırlar.
- Gelişmiş metasezgisel algoritmalar, günümüzde tarama esnasında yol göstermesi açısından hafıza bazlı işlemler barındırmaktadır.

Bu algoritmaların tanımı, zor olan optimizasyon çözümlerinde daha iyi sonuçları elde etmek için üst seviye işlemler içeren sezgisel prosedürler olarak yapılabilir. Metasezgisel yöntemlerin esnek olabilmesi için tek parametreden daha fazla parametreye ihtiyacı vardır. Ancak bu parametre fazlalığı her probleme göre farklılık arz etmektedir. Neredeyse tüm metasezgisel algoritmalar aynı anda birden çok işlemi yürütebilmek için uygundur. Metasezgisel algoritmalar aşağıdaki gibi sınıflandırılabilir (Tarantilis vd., 2004):

- Popülasyon bazlı algoritmalar – Tek noktalı (yerel taramalı) algoritmalar
- Tabiattan ilham alınarak geliştirilenler – Tabiattan ilham alınmadan geliştirilenler
- Dinamik amaç fonksiyonlu – Statik amaç fonksiyonlu algoritmalar
- Hafızaya sahip– Hafızaya sahip olmayan algoritmalar
- Tek komşu yapılı – Çok komşu yapılı algoritmalar

İlk olarak metasezgisel algoritmalar, yerel taramalı ve popülasyon bazlı olmak üzere ikiye ayrılır. Yerel tarama yönteminde, her işlemde tek çözüm seçilir ve uygun komşu çözümler incelenerek olası çözümler içinde belirlenen tek noktada yoğun olarak arama

gerçekleştirilir. Tabu Arama Algoritması ve Tavlama Benzetimi bu yönteme örnek olarak verilebilir. Popülasyon bazlı tarama yönteminde her işlemde daha önceden oluşturulmuş çözümler içinde rasgele çözümler seçilir ve bu çözümler kendi içinde birleştirilir ve yeni çözümler ortaya çıkar. Bu yöntemlere örnek olarak, iki çözümü rasgele seçerek bu çözümlerden seçtiği parçaları birleştirerek yeni çözüm üreten genetik algoritmalar verilebilir (Cordeau vd., 2002).

2.1.5. ARP' nin Uygulama Alanları

Genelde, araç rotalama problemlerinin ilgi alanına, belirli konumlar içinde hizmet ve ürün sevkiyatı yapan sektörler girmektedir. Karşılaşılan örnek problemler,

- Tek depodan ya da birden fazla depodan hizmet ve ürünlerin birçok müşteri konumuna dağıtım yapılması,
- Üretilen mamul ve yarı mamullerin, hammadde ve üretim planlamasının fabrikalar arasındaki sevkiyatı,
- Ürünlerin mağazalara taşınması ve stok planlaması,
- Ürün ve yolcuların havayolu ile taşınması,
- Para dağıtımı,
- Bar ve lokantalara içecek dağıtımı,
- Süt dağıtımı ve toplanması,
- Akaryakıt dağıtımı,
- İnternet alışverişlerindeki teslimatlar,
- Çöp toplama ve taşıma işlemleri,
- Posta hizmetleri,
- Depodan mağazalara ürün dağıtılması,

günlük hayattan verilebilir (Demircioğlu, 2009).

2.2. GENETİK ALGORİTMALAR

Genetik algoritma (Genetic Algorithm, GA) tabanlı çözümlerin temeli ilk olarak 1975' te John Holland (Holland, 1975) tarafından ortaya çıkarılmıştır. Genetik algoritmalar yöntemi, genetik biliminden faydalanarak problemi modellemektedir. Problemlerde yer alan değişkenler ve çözümler, genler ve kromozomlar olarak ifade edilirler ve genetikte

olduğu gibi her bir nesilde çaprazlamalarla çözümler daha da iyileştirilerek devam edilir.

Genetik algoritmalar yönteminin temeli evrim teorisinde Darwin tarafından ortaya çıkarılmış olan “adaptasyon” ve “doğal seçim” kavramlarına dayanmaktadır. Adaptasyon, canlıların içinde buldukları çevre koşullarına göre ayakta kalabilmesini sağlayan genetik değişikliklerdir. Darwin’in adaptasyon için yaptığı tanımlamada; değişen çevre koşullarına ayak uydurabilen canlılar yaşar, uyduramayanlar ise ölürlür. Bu ifadeleri NP sınıfı problemler için yorumlayacak olursak; her bir iterasyonda uyumu en iyi olan çözümler popülasyonda kalırken, uyumu düşük olan çözümler ise popülasyondan elenmektedir. Burada gerçekleşen bu doğal seleksiyon ve çevre koşullarına olan adaptasyonlar ile her bir iterasyonda daha da iyileşen popülasyon bir süre sonra en mükemmel hale gelerek istenilen çözüm değerine götürebilmektedir.

Genetik algoritmalar, büyük ve lineer olmayan arama uzaylarında geleneksel hesaplama yöntemlerinin işe yaramadığı durumlarda kullanılabilen doğal seleksiyon ve doğal genetiğin yöntemlerinin kullanan bir arama algoritmasıdır (Louis, Li, 2000).

Genetik algoritmalarda temel prensip olarak doğadaki canlıların geçirdiği süreç örnek olarak alınır; iyi olan nesillerin yaşamlarını korumaları ve kötü olan nesillerin yok olmasına dayanır. Kesin çözümlere ulaşılamayan veya matematiksel olarak modellenmesi yapılamayan problemlerde genetik algoritmalarından faydalanılır. Genetik algoritmalar, bir önceki nesil olan anne birey ve baba bireyden doğan yeni çocuk bireylerin mevcut olan şartlara uyum sağlaması, yaşamlarını sürdürmeleri temeline dayanır. Yeni çocuk bireyler, bünyelerinde anne ve baba bireylerden aldıkları iyi olan genleri barındırarak kötü olan genleri de almış olma ihtimali bulunmaktadır. Böyle durumlarda kötü olan genleri olan çocuk bireyler varlıklarını devam ettiremezler. Genetik algoritmalar, doğada en iyinin hayatta kalması ilkesini benimsediğinden bunu sağlamak için uygunluk (fitness) fonksiyonu, çaprazlama, mutasyon ve kopyalama gibi operatörleri kullanarak yeni çözümleri üretir. Bu algoritmaların bir çözüm havuzu içerisinde arama yaparak çok sayıda çözüm içinden en iyiyi seçmesi önemli özelliklerinden biridir (Biroğul, 2005).

Genetik algoritmalar, tabii seçim metodunun benzetme yapılarak bilgisayarlara uyarlanması sayesinde geliştirilen bir tarama metodudur. Darwin’ in evrim teorisi baz

alınarak ortaya çıkan genetik algoritmalar evrimsel programlamanın bir parçasıdır. Evrimsel programlama ise Rechenberg' in "Evrimsel Stratejileri" adıyla 1960' lı yıllarda gündeme gelen bir çalışmadır. Genetik algoritmaların kombinatoriyal problemler üzerindeki başarısı 1989'da Goldberg tarafından yazılan bir kitap ile ortaya çıkmış, ancak ilk olarak 1975' te Holland tarafından bugünkü biçimi belirtilmiştir. Standart bir genetik algoritmada çözümler vektörel ve eşit boyutlu olarak ifade edilmektedir. İlk olarak bu vektörler arasından rasgele bir grup seçilir ve önceden belirlenmiş büyüklükte bir popülasyon üretilir. Bu vektörlere, kromozom adı verilir ve bu kromozomlardan nesiller boyunca değişikliğe uğrayacak yeni popülasyonlar oluşturulur. N boyutundaki bir vektörün karşılığı kromozom üzerindeki genlerdir. Üretilen her kromozoma amaç fonksiyonu uygulanarak sonucun iyiliği her yeni nesildeki kromozomlar için ölçülür. Bazı kromozomlar çaprazlama ve mutasyon operatörleri kullanılarak yeniden üretilir ve bu kromozomlardan bir sonraki nesil oluşturulur. Çaprazlama ve mutasyonlar için özel tip operatörler geliştirilmiştir (Nabiyev, 2003).

Genetik Algoritmaların geleneksel yöntemlerden farkı aşağıdaki gibidir.

1. Genetik algoritmalar parametrelerin koduyla uğraşır. Parametrelerin kendileriyle direkt ilgilenmez.
2. Genetik algoritmalar bir tek yerden değil, bir grup çözüm içinden arama yapar.
3. Genetik algoritmalar ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir. Bu nedenle bir kör arama (blind search) metodudur.
4. Genetik algoritmalar olasılık kurallarına göre çalışır. Programın ne kadar iyi çalışacağı önceden kesin olarak belirlenemez. Ama olasılıklarla hesaplanabilir (Goldberg, 1989).

Genetik algoritmaların avantajları şu şekilde sıralanabilir (Şeker, 2008).

1. Sürekli ve ayrık parametreleri optimize eder.
2. Türevsel bilgiler gerekmez.
3. Maliyet fonksiyonunu geniş bir spektrumda araştırır.
4. Çok sayıda parametrelerle çalışma imkanı vardır.
5. Paralel PC'ler kullanılarak çalıştırılabilir.
6. Karmaşık maliyet fonksiyon parametrelerini, yerel minimuma veya maksimuma takılmadan optimize edebilir.

7. Sadece tek çözüm değil, birden fazla parametrelerin optimum çözümlerini elde edebilir.

2.2.1. Genetik Algoritmalar ile İlgili Temel Kavramlar

2.2.1.1. Gen

Kromozom yapısında kendi başına birer genetik bilgi taşıyan en ufak yapı birimine gen denir. Kısmi bilgiler taşıyan bu ufak yapıların bir araya gelmesiyle bütün bir çözüm kümesini oluşturan kromozom (dizi) meydana gelir. GA' nın kullandığı programlama yapısında bu gen yapıları programcının tanımlamasına bağlıdır. Bir genin içerdiği bilgi sadece ikili tabandaki sayıları içerebileceği gibi onluk taban ve onaltılık tabandaki sayı değerlerini de içerebilir. Dolayısıyla yazılan programa göre gen içeriği çok önem kazanmaktadır.

2.2.1.2. Kromozom

Çözüme ait bütün veriyi barındıran ve bir ya da daha çok genin yan yana gelmesi ile oluşan dizilere kromozom adı verilir. Popülasyon (yığın) ise birden çok kromozomdan oluşur. Kromozom içindeki her birey gen olarak, popülasyon içindeki her birey ise kromozom olarak adlandırılır. Ele alınan problemin verilerini kromozomlar barındırır, genetik algoritmalarda en önemli bileşen kromozomlar olduğundan bilgisayar ortamında ifade edilirken önem verilmesi gerekmektedir.

Kromozomdaki bölümlerin ne anlam ifade edeceği ya da ne gibi veri barındıracağı kullanıcının bakış açısını etkiler.

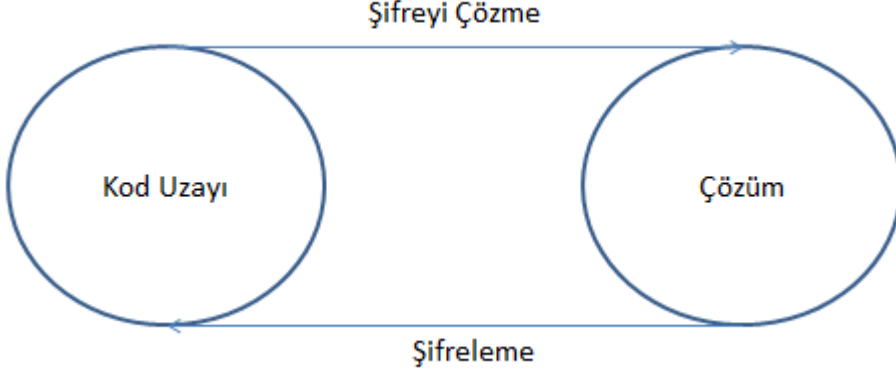
2.2.1.3. Popülasyon (Yığın, Kütle)

Olası çözüm gruplarının oluşturduğu ve çözüm verilerini barındıran kromozomlar bir araya gelerek popülasyonu oluştururlar. Üzerinde durulan problemin zorluğuna ve özelliğine bağlı olarak popülasyonu oluşturan kromozom sayısı programlayıcı tarafından belirlenir. Popülasyon üye sayısının korunması için genetik algoritmaların işleyişi sırasında yok olan kromozomlar yerine yeni kromozomlar eklenmektedir.

Popülasyonun büyüklüğü problemin çözüm süresi ile doğru orantılıdır. Büyük popülasyonlar problemin daha geç çözülmesine neden olurken, küçük popülasyonlar ise henüz en iyi çözüme ulaşılmadan işlemlerin sonlanmasına ya da olası çözümler içinde takılıp en iyi çözüme ulaşılamamasına neden olmaktadır (Duman, 2007).

2.2.1.4. Kodlama (Dizi gösterimi)

Çözüm ve kod uzayında çalışmaları genetik algoritmaların en basit özelliğidir. Kod ve çözüm uzayı arasındaki etkileşim aşağıda, Şekil 1.7 'de gösterilmiştir.



Şekil 1.7: Kod ve Çözüm Uzayı (Cheng vd., 1996).

Çözümün kromozomlara nasıl kodlanacağı genetik algoritmalarda anahtar niteliği taşır. Son on yıl boyunca değişik kodlama teknikleri geliştirilmiştir. Kromozomların kodlanması aşamasında dikkat edilmesi gereken üç önemli nokta vardır (Cheng vd., 1996). Bunlar;

- Kromozomların uygunluğu (Feasibility) kontrol edilir.
- Kromozomun belirtilen şartları (Constraints) sağlayıp sağlamadığı (Legality) kontrol edilir.
- Kodlama haritasında, kromozomun tek olduğu kontrol edilir.

2.2.1.4.1. İkili Kodlama (Binary Encoding)

İkili düzende kodlamanın önemi, ikili dizilere genetik operatörler uygulandığında uygun çözümün garantisi olmasıdır. Uygun olmayan çözümlerle ilgilenmekte iki yol vardır. Birinci yol, uyumun görünüşünü değiştirmeden uygun olmayan çözümlerin uyumunu cezalandırmak için ceza fonksiyonu oluşturmaktır. İkinci yol ise, uygun olmayan çözümü, uygun çözüme dönüştüren sezgisel operatörler tasarlamaktır. Genelde ceza fonksiyonunu belirlemek zor bir iş olduğu için ikinci yol tercih edilmektedir.

En yaygın kullanılan kodlama yöntemidir. Sayıların ikili sistemde gösterimine dayanır. Bu yöntemde kromozomlar sıfır ve birlerden oluşan dizilerdir. Her parametre için n-bit kullanılır, n, her parametre için farklı olabilir.

İkili düzende kodlama, çok sık kullanılan bir kodlama tipi olmasına rağmen bazı sakıncaları vardır. Örneğin, çok değişkenli fonksiyon en iyilemesi için değişkenlerin alt ve üst sınırlarına bağlı olarak elde edilen dizi çok uzun olabilir. Aynı zamanda gezgin satıcı, çizelgeleme, kareli atama gibi kombinatoryal en iyileme problemlerinde ikili düzende kodlama, araştırma uzayını tam olarak temsil edememektedir. Bu nedenle literatürde, permütasyonlu kodlama (sayısal kodlama) daha sık kullanılmaktadır (Duman, 2007).

Kromozom A: 101100101100101011100101

Kromozom B: 111111100000110000011111

Şekil 1.8: Binary Kodlama Gösterimi (Şeker, 2007).

2.2.1.4.2. Permütasyonlu Kodlama (*Permutation Encoding*)

Sıralama problemlerinde permütasyonlu kodlama kullanılır. Bu kodlamada, her kromozom, dizideki bir sırayı temsil eden sayılardan oluşur.

Örneğin gezgin satıcı probleminde (travelling salesman problem) “verilen şehirler arasındaki uzaklıklar bilinirken, bütün şehirleri birleştiren minimum yol uzunluğu ne olmalıdır?” sorusuna cevap aranmaktadır. Burada kromozomlar, şehirlerin hangi sırada ziyaret edilebileceğini gösteren permütasyonlar şeklinde kurulurlar (Duman, 2007).

Kromozom A: 1 5 3 2 6 4 7 9 8

Kromozom B: 8 5 6 7 2 3 1 4 9

Şekil 1.9: Permütasyon Kodlama (Şeker, 2007).

2.2.2. Genetik Algoritmaların Aşamaları

Başlangıç: Uygun bir çözüm için N kromozomlu rasgele popülasyonun oluşturulması.

Tekrar Üretim: İki ebeveynin oluşturulan popülasyondan seçilmesi. Bu işlem stokastiktir ve uyumluluk fonksiyonunun yüksek olduğu bireylerin seçilme olasılıkları yüksektir.

Yeni Popülasyon: Yeni popülasyonun oluşumuna kadar aşağıda yer alan basamakların tekrar edilmesi.

Seçim: Uyumluluğuna göre iki ebeveyn kromozomun seçimi.

Çaprazlama: Seçilen ebeveynlerin yeni fert oluşturmak için bir çaprazlama olasılığına göre çaprazlanması. Çaprazlama olmazsa yeni fert ebeveynlerin aynısı olacaktır.

Mutasyon: Kromozom içindeki konum (lokus) değiştirilerek yeni ferdin oluşturulması. Mutasyon olasılığının küçük seçilmesi tavsiye edilir.

Ekleme: Bu operatörler vasıtasıyla oluşturulan bireyin mevcut popülasyona eklenmesi.

Değiştirme: Yukarıdaki işlemler sonucunda elde edilen yeni popülasyonun algoritmanın bundan sonraki çalıştırılmasında kullanılması.

Test: Eğer sonuç tatmin ediyorsa algoritmanın sona erdirilmesi ve son popülasyonun çözüm olarak sunulması.

Döngü: Eğer sonuç tatmin etmiyorsa 2. adıma geri dönülmesi (Potvin, Duhamel ve Guertin, 1996).

2.2.3. Başlangıç Popülasyonu

Popülasyonun birey sayısı genetik algoritmaların önemli bileşenlerindedir. Bu bileşen bir işlem için kaç adet kromozomun kullanılacağını ifade eder. Popülasyon içindeki kromozom adedi sabit tutulmalıdır. Fakat bununla ilgili genel bir kural bulunmamaktadır.

Popülasyona ait birey sayısı, genetik algoritmaların performans ve verimliliğini ile doğru orantılıdır. Genetik algoritmalarda az üye sayısına sahip popülasyonlar kullanıldığında iyi sonuçlara ulaşılamamaktadır. Kromozom sayısı az ise genetik algoritma, olası çözümlerin içinde yalnızca bir bölümüne ulaşabilmekte ve çaprazlama yapacağı fazla kromozom bulamamaktadır. Popülasyon sayısı arttıkça daha çok sayıdaki

alanlardan temsiller içermesi olasılığı artmaktadır. Popülasyonların büyüklüğü yüksek tutulursa uygun olmayan çözümlere takılıp kalma sorunu yaşanmayacaktır. Öte yandan fazla kromozoma sahip popülasyonlar her nesilde daha fazla işlem yapmayı gerektirecektir. Böyle bir durum, genetik algoritmaların sonuca ulaşmada daha fazla bir süreye ihtiyaç duymasına neden olacaktır (Grefenstette, 1992).

2.2.4. Seçim (Yeniden Üretim İşlemi)

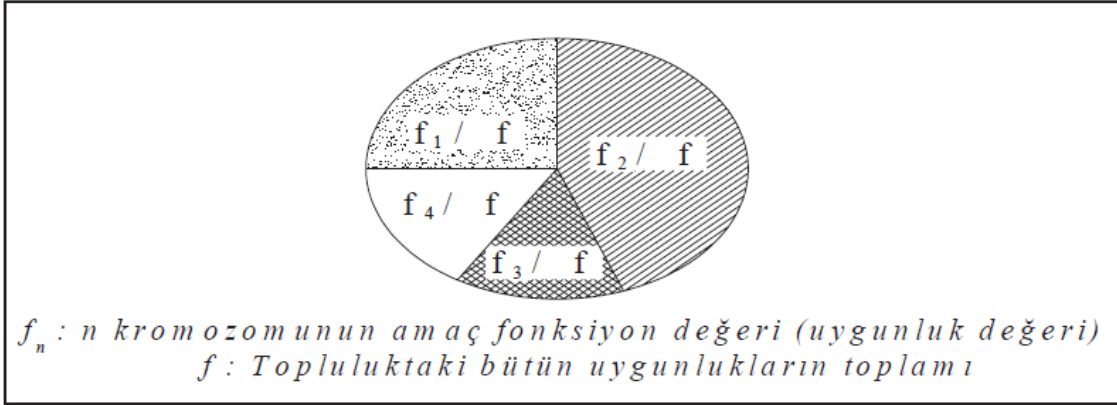
Genetik algoritmaların hesap başlangıcında oluşturulan rasgele çözüm grubundan başarılı çözümler seçilerek başarısız nesiller atılır. Yeniden üretim (seçim) işlemi nesildeki bireylerin üreme havuzuna atılmasıdır. Yeniden üretim işlemi, bir sonraki kuşak için yavru üretmek amacıyla hangi ailelerin yer alması gerektiğine karar verir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımaktır (Erdal, 2007).

Seçim işlemi yapılırken birkaç farklı yöntem kullanılmaktadır. Bu yöntemlerden bazıları;

2.2.4.1. Rulet Tekerı

Bu yöntemde seçilme işlemi bireylerin uygunluk değerlerine göre yapılmaktadır. Fakat uygunluk değeri en büyük olanın seçileceği garanti edilemez, yalnız seçilme şansı daha fazla olacaktır. Bu yöntemde tüm bireylerin uygunluk değerleri bir tabloya yazılır ve toplanır. Sonra uygunluk değerleri toplam uygunluk değerine bölünerek bireylerin (0,1) aralığında seçilme olasılıkları belirlenir. Sayıların hepsi bir tabloda tutulur ve sayılar birbirine rasgele eklenerek rasgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu çözüm seçilmiş olur. Bu yöntem rulet tekerı ismi, bir daireyi, çözümlerin uygunluklarına göre dilimleyip çevirdiğimizde olacaklara benzediğinden verilmiştir.

Rulet tekerı seçiminde çözümlerin uygunluk değerleri pozitif olması gerekmektedir. Çünkü olasılıkların negatif olması, çözümün seçilme şansının olmadığını göstermektedir. Çoğunluğunun uygunluk değeri negatif olan nesiller bir noktaya takılıp kalabilir (Erdal, 2007).

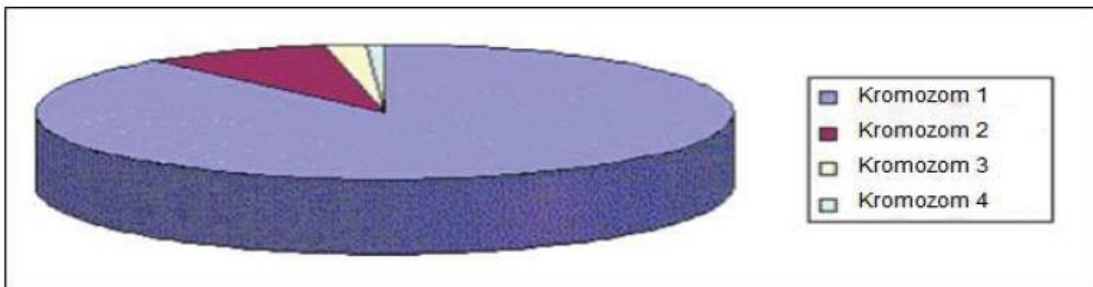


Şekil 1.10: Rulet Tekerleri Seçimi (Erdal, 2007).

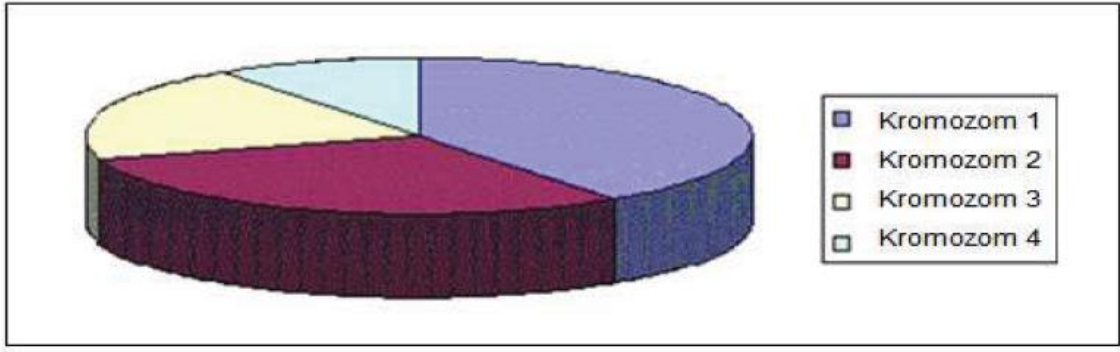
2.2.4.2. Rank Seçim Yöntemi

Rulet tekeri seçim yöntemi eğer uygunluklar çok fazla değişiyorsa bazı sorunlara yol açacaktır. Örneğin en iyi kromozom uygunluğu, tüm rulet tekerleğinin %90' ı ise diğer kromozomların seçilme şansları çok az olacaktır. Rank seçim yöntemi önce popülasyonu sıralar ve daha sonra her kromozom uygunluğu bu sıralamadan sonra alır. En kötüsü 1 uygunluğu alacak, ikinci en kötü 2 ve en iyisi N uygunluk değerini alacaktır.

N , popülasyondaki kromozom sayısıdır. Sayıları düzenlemek için uygunlukları değiştirdikten sonra durumun nasıl değiştiğini şekil yardımı ile görebiliriz.



Şekil 1.11: Rank Seçim Yöntemi 1 (Akoğlu, 2006).



Şekil 1.12: Rank Seçim Yöntemi 2 (Akoğlu, 2006).

Bu yöntemde her kromozomun seçilme hakkı olacaktır. Ama bu yöntem daha yavaş çalışır, çünkü en iyi kromozomlar diğerlerinden fazla değişiklik göstermez (Akoğlu, 2006).

2.2.4.3. Turnuva Seçim Mekanizması

Turnuva seçim yöntemi, seçim baskısı yönünden sıralama seçimlerine benzer, ancak işlemsel olarak daha verimli olup, paralel uygulamaya daha yatkındır. Turnuva seçiminde, popülasyondan yerine koyarak ya da yerine koymadan rasgele t birey seçilir, t genişliğine turnuva adı verilir. Bu gruptaki en iyi birey, yeni popülasyona kopyalanır. Bu işlem N kez tekrarlanır. Büyük t değeri, yöntemin seçicilik baskısını attırır. Turnuvalarda çoğunlukla $t=2$ alınır ve ikili turnuva olarak adlandırılır. U yöntemde zamansız yakınsama, durağanlaşma ve açık uyuma gereksinim yoktur.

Stokastik turnuva seçimi, Wetzel tarafından önerilmiştir (Çetin, 2002). Bu yöntemde seçim olasılıkları hesaplanır ve rulet çemberi yardımıyla ardı ardına birey çiftleri seçilir. Bir çift belirlendikten sonra, yüksek uyuma sahip olanı alınarak yeni popülasyona dahil edilir ve diğer çift seçilir. Süreç popülasyon tamamlanana kadar sürer.

2.2.4.4. Sıralı Seçim Mekanizması

Sıralama seçimi, ilk olarak Baker tarafından uyuma orantılı seçimin dezavantajlarını yok etmek amacıyla önerilmiştir. Sıralama seçiminde bireyler uyum değerlerine göre sıralanırlar. Her bireyin beklenen değeri, sahip olduğu sıraya bağlıdır. Bu sıra dikkate alınarak seçme yapılır. Bu seçme işleminde tüm kromozomların seçilme ihtimali mevcuttur. Fakat bu yöntem çok farklı olmayan bireylerin seçilmesine daha çok imkan tanıyacağından istenen tarama bölgesini çabuk taramayabilir (Duman, 2007).

2.2.4.5. Global Elitizm ve Yerel Elitizm

Bu yöntem seçme algoritması olmaktan çok en iyi çözümü (kromozomu) korumaya yönelik olan bir yöntemdir. İkiye ayrılmasının sebebi, global olarak bir bireyin korunmasını sağlamak veya yerel olarak bireylerin korunmasını sağlamak amacıyla (Karcı, 2002).

2.2.4.6. Kalabalıklaştırma

Nesiller arasındaki dağılım farkını minimuma indirerek prematüre çözümlerin oluşmasını engelleme yöntemine kalabalıklaştırma denir. Bu yöntemde popülasyonun belli bir orana göre bir kısmı seçilir ve bir sonraki nesil oluşturulur. Bu yöntemin temel prensibi var olan popülasyondan evlatlar üretilir ve sonuç evlatları popülasyondaki bireylerin yerlerini alırken, her evlat için popülasyonda gelişigüzel olarak belli sayıda birey seçilir ve en çok benzeyenin yerine evlat konulur. Benzerlik metriği tanım kümesinden bağımsız olabileceği gibi tanım kümesine bağımlı da olabilir. Yerine yeni birey seçilecek olan, en çok benzeyen birey seçilecek bir nesilden sonraki nesle geçilirken popülasyon dağılımları arasındaki farkın minimum olmasına çalışılır. Bu durum genetik algoritmanın tek çözüme meyilli olarak davranmasını engeller (Karcı, 2002).

2.2.5. Genetik Algoritmalarda Kullanılan Operatörler

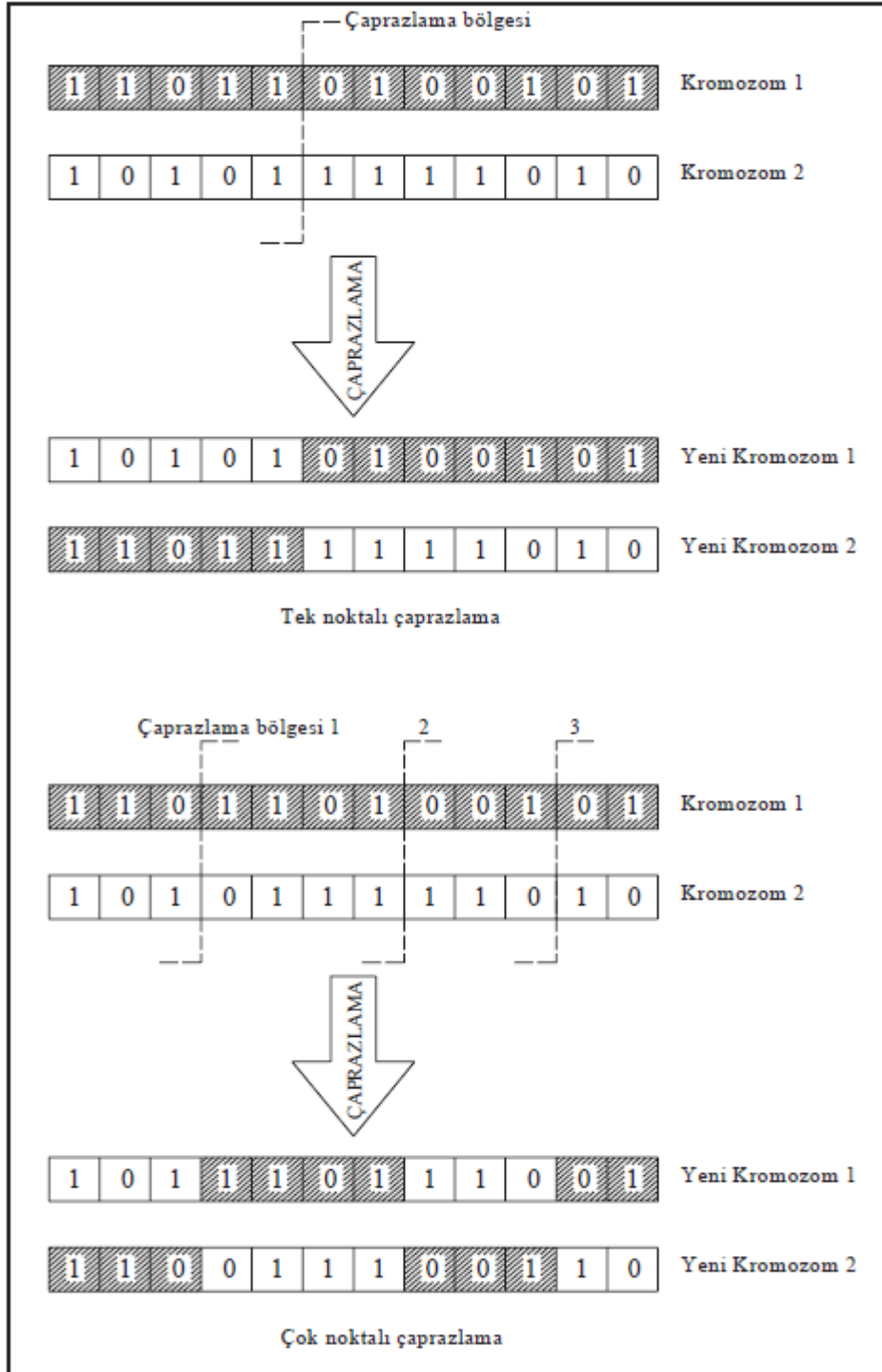
2.2.5.1. Çaprazlama Operatörü

Ebeveynlere ait kromozomların birleşmesi ile yeni kromozomların elde edilmesine Genetik Algoritmalarda Çaprazlama aşaması denilmektedir. Genetik Algoritmalarda çözüm aranırken daha önce oluşturulmuş çözümler içinden seçilecek iki farklı çözüm dizilimi arasında yapılan çaprazlama ile önceki çözüm dizilimlerin daha iyi ve en iyi çözümü daha yakın alan yeni iki çözüm dizilimi elde edilir. Bu yeni yöntemler seçilen daha önceki çözüm dizilimlerinden benzerlikler içerir. Bu sayede oluşturulan yeni daha iyiye yakın çözüm dizilimleri iyiye doğru evrimleşerek yeni bir toplum oluştururlar. Bu algoritmalarda iki çözüm diziliminin sayısal karşılıklarının bazı bölümleri kendi arasında çaprazlama yaparak yeni çözüm dizilimleri oluştururlar. Bu işlem yaşayan organizmalardaki DNA çaprazlamasına benzetilerek elde edilmiştir (Akoğlu, 2006).

Çaprazlama işleminde, kromozomlara ait genler birbirleriyle yer değiştirirler. İlk olarak çaprazlama işleminin uygulanacağı kromozom çiftleri rasgele seçilmektedir. Sonrasında

ise seçilen bu kromozomlara çaprazlama için kesilecek olan genlerin hangi genden başlayıp hangi gende bitirileceği yine rasgele seçilir. Kesilmiş olan bu gen sırlaması diğer eş gen arasında değiştirilir. Bunun amacı mevcut nesilden farklı yeni nesiller oluşturmaktır. Çaprazlama işlemi farklı şekillerde uygulanabilir.



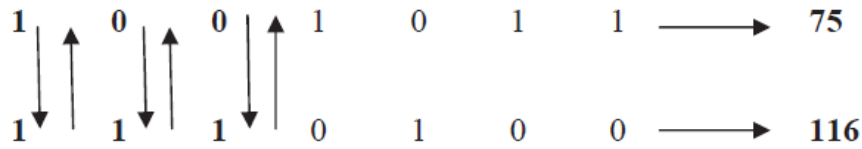


Şekil 1.13: Çaprazlama operasyonu (Erdal, 2007).

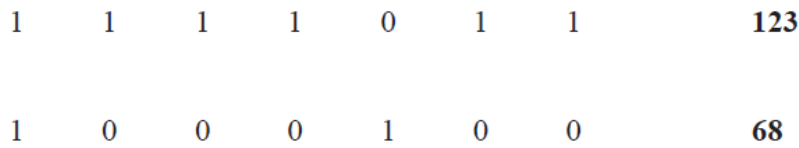
2.2.5.1.1. Tek Kesimli Çaprazlama

Burada eşleşen iki kromozom ikiz doğurarak kromozom sayısının aynı kalmasını sağlar. Öncelikle iki kromozomun ilk ve son rakamları arasında bir yerde rasgele olarak çapraz geçiş noktası belirlenir. Bir kromozomun çapraz geçiş noktasının solunda bulunan tüm haneleri olduğu gibi diğer kromozom ile yer değiştirir. Önce ilk iki kromozomdan

birindeki çapraz noktanın solundaki tüm haneler yeni çocuğun kromozomuna geçer. Benzer şekilde diğer kromozomun çapraz geçiş noktasının solundakilerle diğer yeni doğacak kromozoma geçer. Bundan sonra birinci esas kromozom çapraz geçiş noktasının sağındaki haneler ikinci yeni doğanın önceki hanelerinin sağına geçer. Böylece, ilk yeni doğanın kromozomu tamamlanmıştır. Benzer şekilde ikinci esas kromozomun çapraz geçiş noktasının sağındaki hanelerde birinci yeni doğan çocuğun hanelerinin sağına yerleşir. Böylece yeni doğan iki kromozom öncekilerinin hanelerini ihtiva eder. Esas kromozomlar toplam olarak $N_{kötü}$ tane yeni doğuşa sebep olmalıdır ki toplum büyüklüğü sabit kalsın ($N_{top} = \text{sabit}$). Buna basit veya tek çapraz geçiş noktalı doğumlar adı verilir. Bu işlemde kromozomlar rasgele bir yerden kesilir ve sonra ilgili genler ile yer değiştirilir (Şen, 2004).



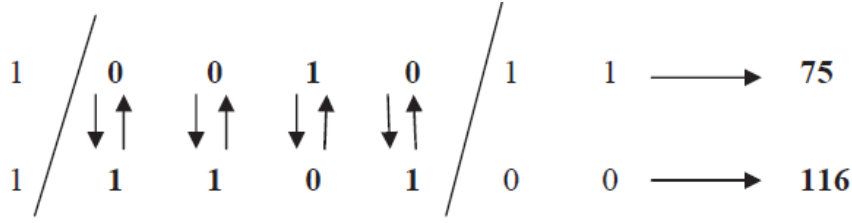
Yukarıdaki kromozom ikilisi 3. haneden hemen sonra kesilmiş ve koyu genler yer değiştirilerek aşağıdaki yeni kromozomlar elde edilmiştir. Yukarıdaki anlatımdan anlaşılacağı gibi her bir kromozoma bir sayı karşı gelir ve kromozomlardaki genlerden bir tanesinin bile değişmesi ile bu sayılar da değişir (Şen, 2004).



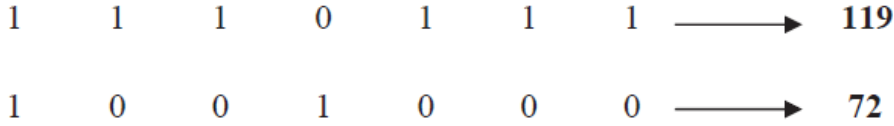
Tek kesimli çaprazlama ile birbirinden farklı sadece iki kromozom elde edilir.

2.2.5.1.2. Çift Kesimli Çaprazlama

Tek kesimli çaprazlamadan farklı olarak bu yöntemde kesim işlemi iki noktada yapılmaktadır.



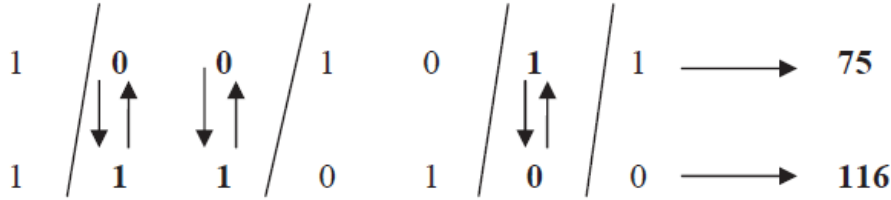
Bu örnekte kromozom çiftleri iki farklı noktadan kesilerek çaprazlama uygulanmıştır.



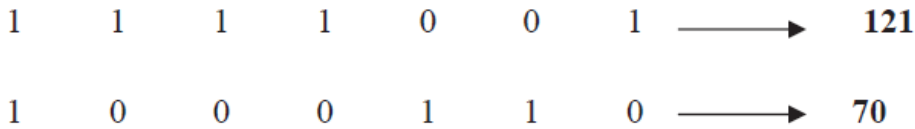
Tek kesimli çaprazlamada olduğu gibi bu yöntemde de yeni nesil eski ne hanelerini ihtiva etmektedir. İki kesimli çaprazlamada her kromozom üç parçaya bölünür. Bu parçalardan karşılıklı her ikisinin yer değiştirmesi ile farklı adet yeni kromozom elde edilir.

2.2.5.1.3. Çok Kesimli Çaprazlama

Kromozomların çapraz olarak ikiden fazla yerden rasgele kesilerek genlerin yer değiştirmesi ile sağlanır.



Bu çaprazlama sonucu oluşan yeni kromozom ve sayılar aşağıdaki gibidir.



Burada da karşılıklı parçalara arasında çaprazlama ile çok sayıda yeni kromozom elde edilir (Şen, 2004).

2.2.5.1.4. Tekdüze (Üniform) Çaprazlama

Bu işlemin aslı rasgele hanelerin iki kromozom arasında yer değiştirmesi ile olur.

$$\begin{array}{ccc|ccc}
 1 & 0 & 0 & 1 & 0 & 1 & 1 & \longrightarrow & 75 \\
 1 & 1 & 1 & 0 & 1 & 0 & 0 & \longrightarrow & 116
 \end{array}$$

Kromozomların karşılıklı çaprazlaması sonucunda

$$\begin{array}{ccccccc}
 0 & 1 & 0 & 1 & 0 & 1 & 1 & \longrightarrow & 43 \\
 1 & 0 & 1 & 0 & 1 & 0 & 0 & \longrightarrow & 84
 \end{array}$$

elde edilir.

2.2.5.2. Mutasyon Operatörü (Rakam Değişimi)

Genetik Algoritmelerde uygulanan diğer bir işlem ise bir kromozom üstünde yapılan mutasyon işlemidir. Mutasyon işlemi vasıtasıyla Genetik Algoritmelerin en iyi çözümü arayışında daha iyi ya da daha kötü bir çözüm oluşturularak, çeşitlilik sağlanmış olur. Kötüye doğru gidilmesinde bile bir sonraki adımda yapılacak bir rakam değişimi ile algoritma kendisini toparlayarak en iyiye doğru hızla yaklaşabilir. Böyle bir fırsat klasik yöntemlerde yoktur, çünkü hepsi önceden belirli bir sistem dahilinde karar uzayında bir sonraki adımı atar. Bu tür atılımlar her zaman başarılı olmasa da algoritma kendisini rasgele yenilediği için fazla zaman kaybı olmayabilir. Rakam değişikliği (mutasyon) kromozomun bir hanesindeki 0 değerini 1 veya 1 değerini 0 yapma işlemidir. Bu işlem kromozomun bir geninde yapılabileceği gibi birden çok gende de uygulanabilir. Bu sayede sadece bir değer değişmesi bile ortaya çok farklı bir sayı çıkmasını sağlar.

Bu süreç aynı kromozomun bir veya birkaç hanesindeki rakamın karşıt rakam türü ile değiştirilmesiyle ortaya çıkar. Rakam değiştirilmeleri ile GA' lar ikinci türden hedef fonksiyonu yüzeyini araştırır. Böylece başlangıçta bulunmayan yeni kromozom türleri elde edilerek en iyileme işleminde GA' nın çözüme süratle yaklaşmasını da önler. Eğer bir kromozoma birli bir rakam değişimi uygulanıyorsa, bu takdirde "1" sayısı "0" sayısına veya aksine dönüştürülür. Rakam değişiminin yapılacağı hane konumu $N_{top} * N_{hane} * N_{gen}$ toplam hane toplumundan rastgele olarak seçilir. Rakam değişim nokta sayısının artırılması genetik algoritmelerin doğal olarak taradığı çözüm uzayının,

klasik GA yaklaşımı ile dokunulmayacak yerlerinin de incelenmesini sağlar. Genelde, her iterasyonda haneleri %1' i ile %0,1'i miktarında rakam değişimi yapılır. En son iterasyonda artık rakam değişimi yapılmaz. Genel olarak, en iyi çözümlerlerde de rakam değişikliğine müsaade edilmez. En iyi kromozomların dışındakiler arasından mesela %5 yani 0,05 miktarında rakamların değiştirilmesi yapılır.

Rakamların değişimi çaprazlamadan farklı olarak iki kromozomun yerine bir tek kromozomun üzerinde rastgele belirlenen hanelerin karşıt rakamlarla yer değiştirilmesi ile yapılan işlemdir. Burada hedef verilen belli bir rakam değişimi değeriyle 0 olan rakamları 1, 1 olan rakamları 0 yaparak yerel en iyilerin dışında mümkün olabilecek en iyileri de araştırmaktır. Rakam değişimi değeri yani rakam değişimine tabi tutulacak hanelerin sayısı küçük tutulur, çünkü olaya rastgeleliğin hakim olması arzu edilmez.

Bir kromozomun içindeki hanelerin birisinin konumu rastgele seçildikten sonra, o hanedeki rakam karşıt rakama dönüştürülürse yeni kromozom elde edilir. Yeni kromozom çözüm uzayının başka bir yerine rastgele olarak sıçramıştır. Genetik algoritma işlemleri sırasında rakam değişimi işlemi nadir olarak yapıldığından düşük ihtimaller göz önünde tutularak uygulanır. Pratik çalışmalarda bu ihtimal değerinin 0,01 ile 0,001 arasında değişmesi sonucunda karar verilmiştir (Şen, 2004).

2.2.6. Genetik Algoritmaların Çalışma Prensipleri

Klasik bir Genetik Algoritmanın çalışması aşağıdadır (Nabiyev, 2003):

1. Bir çözüm grubu oluşturmak için olası çözümler kodlanır. (Biyolojide ki benzerliğinden dolayı çözüm grubuna popülasyon, çözüm kodlarına da kromozom denilebilir.) Birey adedi noktasında popülasyon için belirlenmiş bir standart ya da sınır bulunmaz. Ancak genelde bu rakam 100 ila 300 arasında önerilir. Popülasyon birey sayısının seçimi sırasında, yapılacak arama derinliği ve işlem karmaşıklığı önem kazanır. Bu aşamadan sonra popülasyon klasik sezgisel yöntemler kullanılarak ya da rasgele üretilir. Popülasyon büyüklüğüne karar verildikten sonra üzerinde çalışılacak probleme göre kromozomların kodlar ile ifade edilmesi gerekebilir.
2. Çözüm grubu (Popülasyon) içindeki tüm kromozomların ne kadar iyi olduğu hesaplanır. Bu hesap için kullanılacak fonksiyon "Uygunluk Fonksiyonu" olarak

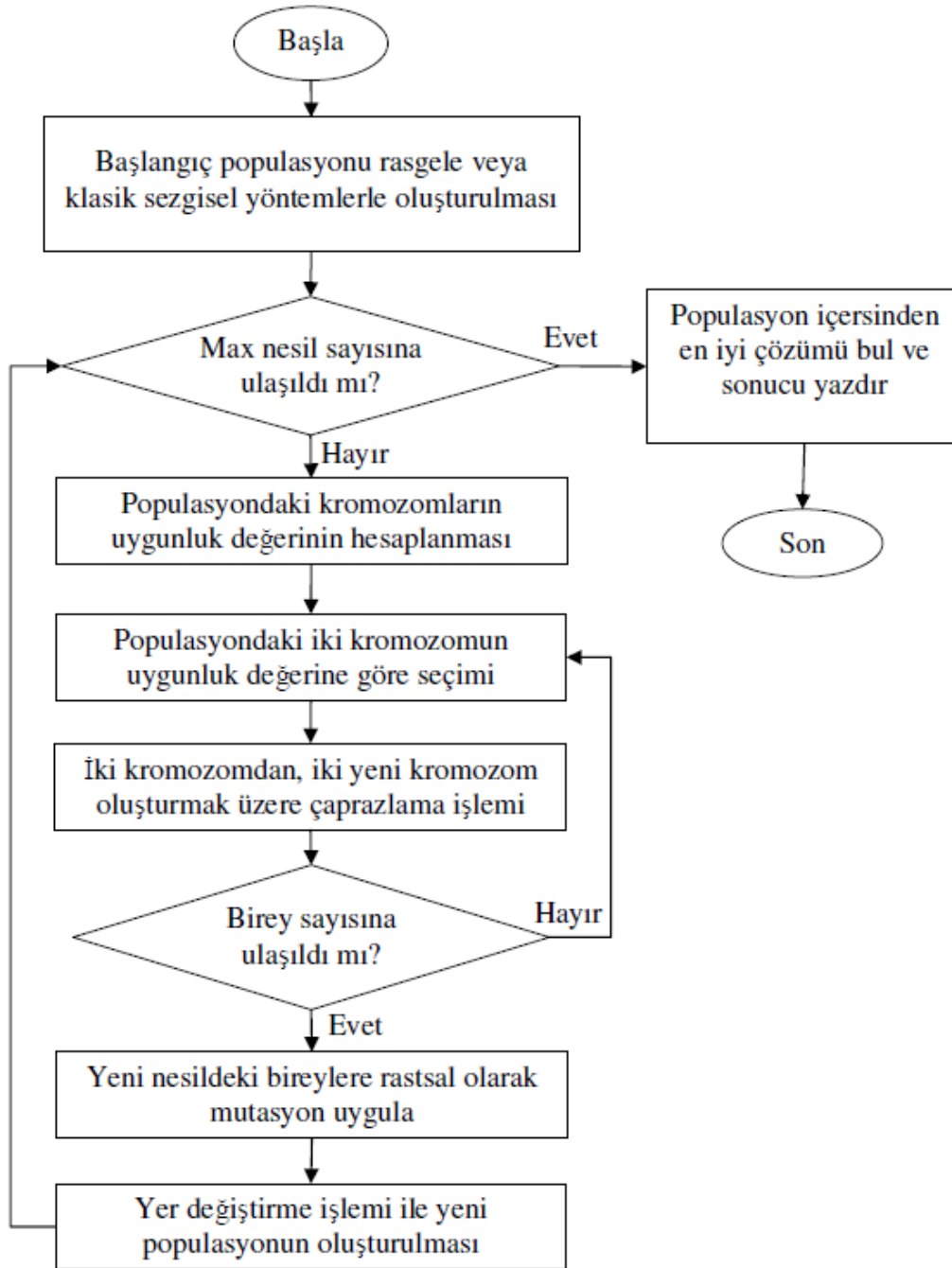
adlandırılır. Bu kromozomların uygunluk fonksiyonu sayesinde uygunluk oranlarının tespit edilmesine de ‐Evrimleşme‐ ismi verilir. Genetik Algoritmanın beynini uygunluk fonksiyonu oluşturur. Probleme özel olarak çalışacak tek bileşen, Genetik Algoritmelerde bu fonksiyondur. Bu fonksiyon kromozomların probleme ait parametrelere dönüştürülmesini sağlayarak başka bir deyişle onların şifrelerini çözmektedir. Bu nedenle uygunluk fonksiyonunun hassas ve verimli oluşturulması çoğunlukla Genetik Algoritmanın başarısını belirler.

3. Uygunluk değerine göre popülasyon içindeki kromozomlar (Çözümler) olasılıksal olarak seçilerek eşleştirilir ve çaprazlama, mutasyon operatörleri uygulanır. Bunun sonucunda farklı bir toplum oluşturulmuş olur. Seçime karar vermek için bazı yöntemler kullanılabilir. Bunlar, turnuva seçimi ve rulet tekeri seçimi gibi metodlardır. Çaprazlama işlemi kromozomların içinden sıralı olarak seçilen bazı genlerin diğer kromozoma aktarılmasıdır. Böylece toplumda çeşitlilik sağlanır. Yerel optimum noktalara takılıp kalmayı engellemek için sadece tek kromozom üzerinde gen değişimi yapan Mutasyon işlemi kullanılır.
4. Eski kromozomlar popülasyon içinden silinerek yerlerine yeni oluşturulan kromozomlar yerleştirilir ve bu sayede popülasyon birey sayısı sabit tutulur.
5. Yeni oluşturulan bu nesildeki tüm kromozomlara ait uygunluk oranları yeniden hesaplanır ve neslin başarısı elde edilir.
6. Bu işlemler ard arda tekrar edilerek nesil sayısı süresince yeni popülasyonlar oluşturulur ve bu popülasyonların daha iyiye ulaşması sağlanır.
7. Sonuç olarak popülasyon içindeki kromozomların uygunluk oranlarının hesaplanması aşamasında en iyi bireylere ulaşıldığında çözüm de bulunmuş olur.

Genetik Algoritmalar, popülasyonlar kullanarak, yapılan araştırmalarda daha optimal hale getirilebilecek sonuçlar için liderlik etmektedir. Problem içindeki değişkenlerin belli bir düzen içinde sıralanması ile kromozom oluşturulmaktadır. Bir kez oluşturulan rasgele başlangıç popülasyonundan sonra seçim, çaprazlama ve mutasyon aşamalarından meydana gelen evrim süreci başlamış olur. Bu süreç Şekil 1.14’de gösterilmiştir. Birçok deney ile Genetik Algoritmaların problem çözme başarıları kanıtlanmıştır. Bu çalışmaların çoğunda özel durumlar ele alınmıştır. Probleme göre özelleştirilen yöntemler içerdiği ve Genetik Algoritma uygulamalarının çok çeşitli

olması nedeniyle incelemeler daha zor hale gelmektedir. Genetik Algoritma ile çözülmeye en uygun problemler klasik metodlar ile çözümlenmesi imkansız olan veya çözüme ulaşma süresi problemin büyüklüğü ile orantılı olduğu durumdakilerdir.

Genetik Algoritmalar, popülasyona bağımlı bir tabana sahip olduğundan hayatımızda çözüm beklenen problemlere uygulanmasında dezavantaja sahiptir. Her algoritmanın problemi çözüme ulaştırma işlemi belli bir süre içinde tamamlanmalıdır. Yerel tarama algoritmaları ile popülasyon bazlı algoritmalar kıyaslandığında işlem süresi popülasyon bazlı algoritmalar için daha uzun sürmektedir. İşlem sürelerini azaltmak için daha az bireye sahip popülasyon ile çalışabilecek ve daha iyi bir verime sahip Genetik Algoritmaların geliştirilmesi önemli bir hale gelmektedir. Başka bir dezavantajda Genetik Algoritmanın bölgesel tarama hızının yeterince iyi olmamasıdır. Bu dezavantajı ortadan kaldırmak için klasik Genetik Algoritmalar ile klasik komşuluk arama algoritmalarını birleştiren araştırmacılar vardır. Bu tür birleşik algoritmalar Literatürde Melez Genetik Algoritma olarak anılmaktadır (Karaboğa, 2004).



Şekil 1.14: Genetik Algoritma Akış Süreci (Preux ve Talbi, 1999).

2.2.7. Genetik Algoritmaların Performansını Etkileyen Etmenler

Genetik algoritmaların performansını etkileyen etmenler kısaca şöyle özetlenebilir (Kulluk, 2003):

Kromozom Sayısı: Kromozomların sayısının, azaltılması kromozom çeşitliliğini yok ederken attırmak ise çalışmanın zamanını da attırmaktadır.

Mutasyon oranı: Mutasyon işlemi, kromozomların birbirlerine benzemeye başladığı anda, bulunan çözüm hala optimumun çok uzağında ise genetik algoritmaların çözüme yaklaştırmanın tek çaresidir. Fakat mutasyon oranını gereğinden fazla yüksek tutmak genetik algoritmayı kararlı noktaya erişmekten uzak tutacaktır.

Çaprazlama nokta sayısı: Çaprazlama işlemi genellikle tek noktada gerçekleştirilmektir fakat yapılan bazı çalışmalar bazı problemlerde çok noktalı olarak yapılan çaprazlamanın daha faydalı olduğunu göstermektedir. Çaprazlama sonucunda meydana gelen yeni bireylerin nasıl değerlendirilmesi gerektiği, oluşan iki bireyin aynı anda kullanılması bazen çok önemli sonuçlar doğurmaktadır.

Ayrıık Nesillerin olup olmaması: Her nesil bütünüyle önceki nesile bağlı olarak meydana gelir. Fakat bazen yeni nesilleri, eski nesil ile beraber; yeni olan neslin o aşamaya gelinceye kadar meydana gelen bireyleri ile yaratmak yararlı olabilecektir.

Parametre kodlaması: Parametre kodlamasının, yapılaş biçimi önemli konulardan birini oluşturmaktadır. Örneğin genetik algoritma performansı, parametrelerin logaritmik veya doğrusal kodlanması nedeni ile önemli farklar oluşabilir.

Kodlamanın gösterim biçimi: Kodlamanın gösterim biçimi de GA' nın performansını etkileyen önemli bir etmendir. En yaygın kullanılan yöntemler arasında gray kodu ile yapılan gösterim, ikili düzen ve kayan nokta aritmetiği sayılabilir.

Başarı değerlendirmesi yapılaş biçimi: Değerlendirme işlevinin seçimi, sonuca ulaşılma zamanı üzerinde çok etkilidir hatta makul seçilememesi hiç bir zaman çözüme ulaşılmamasını sağlayabilir.

2.2.8. Genetik Algoritmaların Uygulamaları

Genetik algoritmaların karmaşık uzaylarda, sağlam araştırmalara imkan tanıdığı teorik olarak ispatlanmıştır. Problemlerin ihtiyaç duyduğu verimli ve etkili araştırmalarda geçerli bir yaklaşım olarak yerleşen genetik algoritmalar; mühendislik, iş/ticaret ve bilimsel çalışmalarda daha geniş uygulama imkanları bulmaktadır. Genetik algoritmaların uygulama alanlarının genişlemesinin sebebi, hesaplamalardaki kolaylığı ve arama uzayı ile ilgili varsayımlarda önemli sınırlayıcılarla kısıtlandırılmamasıdır.

GA, sezgisel bir metot olduğundan dolayı verilen bir problem için optimum sonucu bulamayabilir, ancak; geleneksel metotlarla çözülemeyen veya çözüm zamanı problemin büyüklüğü ile üstel olarak artan problemlerde optimale çok yakın sonuçlar vermektedir. Başlangıçta doğrusal olmayan (non-linear) optimizasyon problemlerine uygulanan GA, sonraları gezgin satıcı, kareli atama, yerleşim, atölye çizelgeleme, ders/sınav programı hazırlanması gibi kombinatoriyal optimizasyon problemlerine de başarı ile uygulanmıştır (Liepins ve diğ., 1990). Genetik algoritmaların kombinatoriyal problemlerin çözümünde oldukça başarılı olduğu ispatlanmıştır (Drezner, 2003).

GA, çeşitli alanlarda kullanılmıştır (Balakrishnan ve Jacob, 1996): Goldberg (1989) genetik algoritmaları boru hattı sistemlerinin optimizasyonunda; Cleveland ve Smith (1989) seri üretim çizelgelemesinde, Liepins ve Potter (1991) çoklu hata tanımında; Holsapple ve arkadaşları esnek imalat sistemlerinin çizelgelemesinde ve uyumlu karar destek sistemlerinde kullanmışlardır.

GA' lar problem çözümede alternatif metotlar sağlamakla kalmaz, aynı zamanda problemlerin çoğunda diğer geleneksel yöntemleri de tutarlıca gerçekleştirir. Örneğin, optimal parametreler bulmada yer alan gerçek dünya problemleri geleneksel metotlar için zor olabilir; fakat GA' lar için idealdir. GA' lar, optimizasyondaki etkili performanslarından dolayı yanlış bir kanıyla işlem en iyileyici olarak görülmektedirler. Aslında genetik algoritmalara çok fazla bakış açısı vardır. Birçok kullanıcı; GA' ya problem çözücü olarak bakmaktadır. Fakat bu kısıtlayıcı bir bakış açısıdır. GA' lar,

- Araştırma alanı geniş, kompleks ve anlaşılması zayıfsa,
- Konudaki bilgi azsa ya da eldeki uzman bilgi, araştırma alanını daraltmada zorlanıyorsa,
- Matematiksel analiz elde edilemiyorsa,
- Geleneksel araştırma metotları başarısız olmuşsa faydalı ve etkilidirler (Çetin, 2002).

GA yaklaşımının avantajı, zorluklarla ve hedeflerle başa çıkmadaki rahatlığıdır.

2.3. ARAÇ ROTALAMA PROBLEMLERİNDE GENETİK ALGORİTMALAR KULLANIMININ LİTERATÜR ARAŞTIRMASI

Araç rotalama problemi, kombinatoryal en iyileme problemidir, bu sınıfta yer alan iyileme problemleri gibi araç rotalama problemleri için de fazla metasezgisel yöntemlerin kullanıldığı çalışmalar yapılmıştır. Araç rotalama probleminin Genetik algoritmalar metasezgisel yöntemi ile çözüldüğü literatürde yer alan çalışmalara bakacak olursak;

Goldberg ve Lingle (1985), gezgin satıcı problemlerine sonuç bulmak için GA' yı kullanmıştır ve sonraki araştırmacılar bu önemli çalışmaya başvurmuşlardır (Bayzan, 2005).

Chatterjee ve diğ. (1996), GA yaklaşımını ile gezgin satıcı problemlerine benzer çalışmalara göre daha iyi sonuçlara ulaşmışlardır (Bayzan, 2005).

Chu R.C. ve Beasley J.E. (1997), ARP ile yakın özellikler gösteren probleme değişik bir yaklaşım getirmişlerdir. Bu çalışma ile, kaynak kısıtlamalarının değerlendirildiği ve her bir işin en az maliyetle acentelere verilmesi problemi olarak da tanımlanan genel görev problemini için GA dan farklı her kromozom için ikili olmayan bir ondalık sayı dizisini kullanılmışlardır (Bayzan, 2005).

Thangiah ve Petrovic (1998), ZPARP ve Çok Depolu karmaşık kısıtlamaları olan Araç Rotalama Probleminin çözümü için GA' dan faydalanmışlardır (Bayzan, 2005).

Golden ve diğ. (1998), ARP' nin çözümü için içinde genetik algoritmalarında yer aldığı metasezgisel çalışmalar yapmışlardır. Sezgisel algoritma performansını klasik ARP' de meydana gelen çözümlerle karşılaştırmışlar ve tabu arama tekniğinin diğer metasezgisel yöntemlere göre ulaşılan sonuçlardan daha başarılı sonuçlara ulaştığını kanıtlamışlardır. Ama bu yöntem diğer yöntemlere kıyasla daha fazla parametre kullanıldığını da ortaya koymuşlardır (Bayzan, 2005).

Ichoua ve diğ. (2000), ZPARP için ilk giren ilk çıkar prensip şartları ile çözüm yöntemi geliştirilmiştir (Bayzan, 2005).

Jung ve diğ. (2001), Araç Rotalama Problemi için GA yönteminden faydalanmışlardır (Bayzan, 2005).

Tan ve diğ. (2001), melez Genetik Algoritmalar yaklaşımını ZPARP çözümünde kullanmışlardır (Bayzan, 2005).

Berger ve diğ. (2003), ZPARP çözümünde melez GA yaklaşımıyla çözmeye çalışmışlardır (Bayzan, 2005).

Baker ve Ayechev (2003), tek depodan müşterilerin siparişlerinin karşılandığı temel ARP çözümü için komşu arama algoritmasında kullanılmış olduğu melez GA kullanmışlardır (Şahin ve Eroğlu, 2014).

Jaszkiewicz ve Kominek (2003), çözüm kalitesini artırmak için çözüm özellikleri için küresel konveks testlerinde yer aldığı bir GA geliştirmişlerdir (Şahin ve Eroğlu, 2014).

Berger ve Barkaoui (2004), ZPARP için iki melez GA ile çalışan 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 (Dursun, 2009).

Hanghani ve Jung (2005), zaman bazlı olarak seyahat süre formülasyonunu DARP için uygulamışlardır. Bu çalışma ile sürekli seyahat zamanları bilgileri ARP' ye dahil edilmiş ve ilk giren ilk çıkar yöntemi için bekleme zamanını kaldırılmışlardır. Problemin çözümü için Genetik Algoritmalar kullanılmıştır ve diğer bazı çözüm metotları ile karşılaştırılarak çözümün etkinliğini ölçmüşlerdir (Bayzan, 2005).

Alba ve Dorronsoro (2006), literatürde yer alan en iyi çözümlerin elde edilmesi için hücresele GA tabanlı yöntem önermiştir (Şahin ve Eroğlu, 2014).

Mester ve Braysy (2007), KKARP için yönlendirilmiş yerel tabanlı arama ve GA' dan oluşan iki aşamalı yöntem geliştirmişlerdir (Şahin ve Eroğlu, 2014).

Alvarenga ve diğ. (2007), genetik algoritmalar ve küme bölme yaklaşımını bir arada kullandıkları çalışmalarında, yine ZPARP için toplam mesafenin en küçüklenmesini temel amaç alan melez sezgisel model geliştirmişlerdir. Problemin, yerel en iyi

çözümündeki 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 algoritmalar kullanarak birbirinden bağımsız koşullar 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 (Dursun, 2009).

Wang ve diğ. (2008), 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 yer alan iki nokta belirlenir, arada kalmış olan alt kromozomlar ile yer değiştirilir. 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 (Dursun, 2009).

Wang ve Lu (2009), başlangıç çözümün süpürme (sweep) ve en yakın ekleme yöntemlerinin kombinasyonundan oluşan yöntem ile birleştirilmiş bir genetik algoritmalar uygulamıştır (Şahin ve Eroğlu, 2014).

Jaszkiewicz ve diğ. (2012), melez parçacık sürü optimizasyon algoritmasını genetik algoritmalar ile birlikte bulanık talepli KARP' nin çözümü için kullanmıştır (Şahin ve Eroğlu, 2014).

Nazif ve Lee (2012), KARP' nin çözümüne yönelik olarak için tam yönsüz ikili grafik kullanılarak oluşturulan optimize edilmiş bir çaprazlama operatörünün kullanıldığı genetik algoritmalar esaslı bir yöntem geliştirmiştir (Şahin ve Eroğlu, 2014).

Razali (2015), bir araç filosu ile birçok müşteriye araç rotalama yapılmasında diğer çözüm yöntemleri ile kıyaslama yapılarak geliştirilen bir model ile genetik algoritmalar çalışması yapılmıştır.

Pierre ve Zakaria (2016), stokastik parça optimizasyonlu aprazlama operatörü ile geliştirilen genetik algoritmalar yöntemini zaman pencereli araç rotalama problemi çözümünde kullanmışlardır.



2. MALZEME VE YÖNTEM

Bu bölümde ilk önce araç rotalama problemini çözmek için metasezgisel bir yöntem olan “Genetik Algoritmalar” kullanılarak geliştirilen algoritma programı tanıtılacaktır. Ardından çözümü yapılacak olan problem tanımı yapılacaktır. Problem, geliştirilen genetik algoritmalar programı, Excel - Solver (Evolutionary) genetik algoritmalar programı ve En Yakın Komşu Algoritması ile çözümleri yapılarak sonuçlar karşılaştırılacaktır. Geliştirilen genetik algoritmalar programına neden ihtiyaç duyulduğu izah edilecektir.

3.1. GELİŞTİRİLEN PROGRAMIN TANITIMI

Araç Rotalama Probleminin Genetik Algoritmalar ile çözümü için geliştirilen genetik algoritmalar programı; Microsoft Excel ve Visual Basic kullanılarak hazırlanmıştır.

3.1.1. Programın Çalışma Prensibi

Programın çalışma prensibi, çalışma parametresi olarak belirlenen popülasyon büyüklüğüne göre rassal şekilde rotalar oluşturulmaktadır. Probleme uygun olarak kullanıcı, Genetik Algoritmalar operatör yüzdelerini tanımlar ve bunun sonucunda programın çalıştırılması ile en kısa yol hesaplanır.

3.1.2. Amaç Fonksiyonu

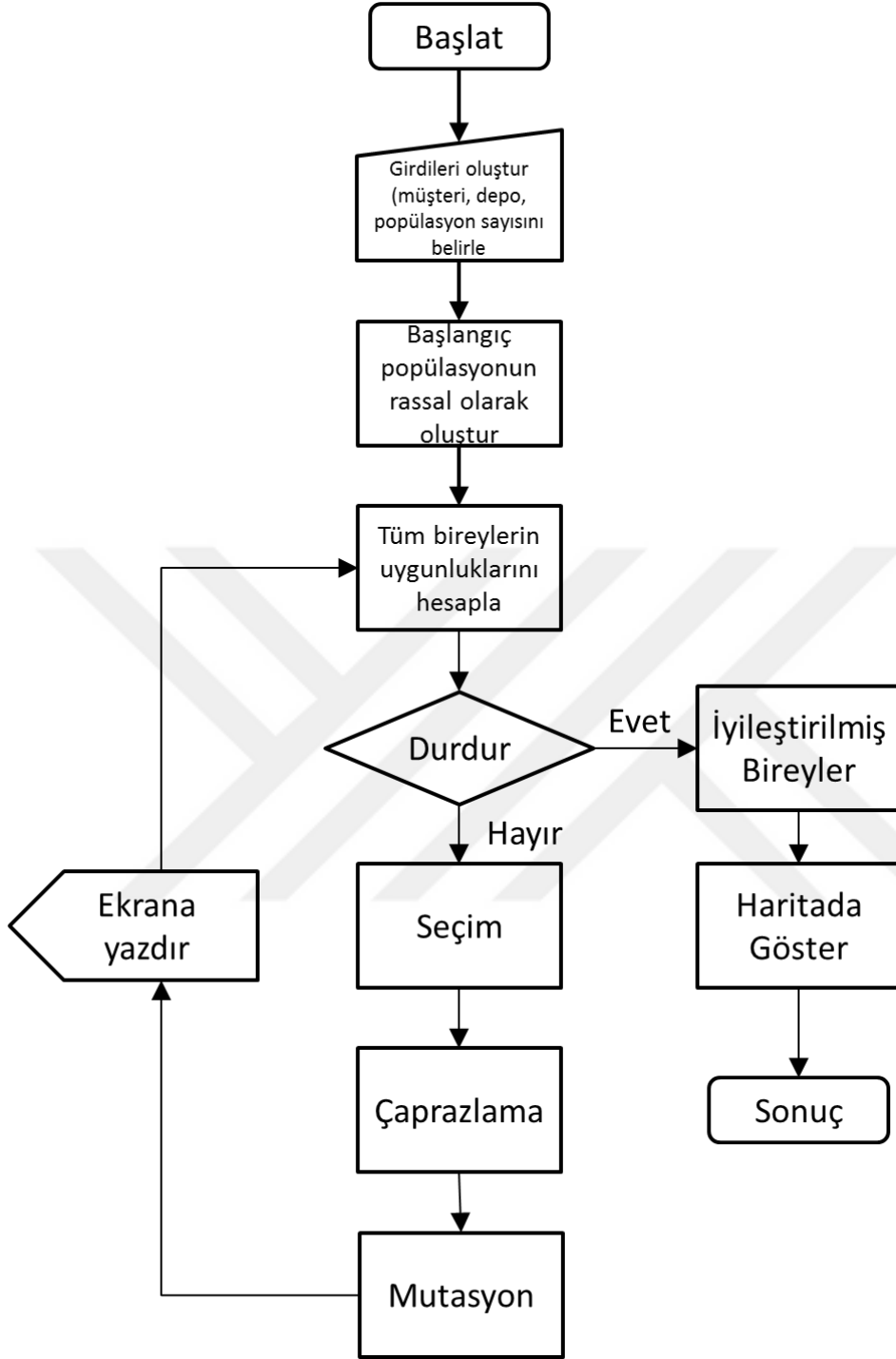
Problemin amaç fonksiyonu, müşteri taleplerinin;

- Araçların en kısa yolu kullanmasını esas almaktadır.

Rotalar tespit edilirken her müşterinin, sadece bir araç tarafından servis görebileceği araçların rotasının depoda başlayıp depoda biteceği şartları mevcuttur.

3.1.3. Programın Akış Şeması

Programın kodlanmasında aşağıdaki algoritma adımları izlenmiştir. Şekil 2.1’ de problem çözülürken izlenen akış şeması şematik olarak gösterilmiştir.



Şekil 2.1: Programın Akış Şeması.

3.1.4. Programın Temel Özellikleri

- Program basit Genetik Algoritmalar kullanılarak hazırlanmıştır.
- Müşteriler için Türkiye'nin 81 ilini kapsayan İller Arası Mesafe Cetveli kullanılmıştır.
- Her il plaka kodu ile ifade edilmektedir.

- İller arası mesafe X ve Y koordinat düzlemindeki mesafeler alınarak hesaplanmaktadır.
- Problemden tek depo olduğu düşünülerek simülasyon çalıştırılmıştır.
- Rotalar depoda başlayıp depoda sonlanmaktadır.
- Her araç, müşterilere hizmet verdikten sonra ayrıldığı depoya geri dönmelidir.
- Depo ve müşteriler problem oluşturulurken manuel olarak seçilir.
- Programda, Tekdüze (Uniform) Çaprazlama Tekniği kullanılmıştır.
- Programda, 3 çeşit mutasyon yöntemi kullanılarak çeşitlilik artırılmıştır. Kullanılan teknikler yer değiştirme, kaydırma ve ters çevirme yöntemleridir.
- Yeniden üretim (seçim) işleminde Rank Seçim Yöntemi tercih edilmiştir.

Programda bulunan uygulama parametreleri Şekil 2.2' de paylaşılmıştır.

Buna göre terimler ve açıklamaları;

ŞEHİR SAYISI: Tanımlamalarda seçilen toplam şehir sayısı

POPÜLASYON SİZE: İlk oluşturulacak rota sayısı

TOPLAM MESAFE: Her rotanın ayrı ayrı uygunluk değeri

MİNİMUM MESAFE: En uygun rotanın uygunluk değeri

UYGUN POPÜLASYON: En uygun rotanın dizilişi

DEPO: Depo olarak seçilen şehir plakası

DENEME SAYISI: Çaprazlama ve mutasyon işlemlerinin toplam deneme sayısı

UYGUNLUK HESABI: Uygunluk hesabının hangi metot ile yapılacağı (karayolu, vektörel) Bu uygulamada karayolu kullanılacaktır.

ÇAPRAZLAMA YÜZDESİ: İşlem sayısı içerisinde çaprazlama için ayrılan işlem yüzdesidir.

MUTASYON YÜZDESİ: İşlem sayısı içerisinde mutasyon için ayrılan işlem yüzdesidir.

İŞLEM YAP: Seçilen parametrelerin baz alınarak programın çalıştırılmasını sağlar.

İLK POPÜLASYON: İlk popülasyon grubunu oluşturur.

ÇAPRAZLAMA: Yukarıda seçilen parametreler ile programın çalıştırılması fakat en kısa mesafenin hesaplanamaması durumunda ilave çaprazlama işleminin yapılmasını sağlar.

MUTASYON: Yukarıda seçilen parametreler ile programın çalıştırılması fakat en kısa mesafenin hesaplanamaması durumunda ilave mutasyon işleminin yapılmasını sağlar.

HARİTA OLUŞTUR: Program tarafından hesaplanan rotaların görsel olarak harita üzerinde hazırlanmasını sağlar.

The screenshot shows a software interface for setting parameters. On the left, there are several text boxes with labels and arrows pointing to specific fields in the table below. On the right, there are four buttons: 'İlk Popülasyon', 'Çaprazlama', 'Mutasyon', and 'Harita Oluştur'. A large red button labeled 'İşlem Yap' is positioned below these buttons.

	M	N	O	P
Şehir Sayısı		10		
Popülasyon Size		15		
Toplam Mesafe		6170		
Minimum Mesafe		4853	1	
Uygun Popülasyon	35,48,7,31,30,53,61,14,16,17,35			
Başlangıç Şehri		7		
Deneme Sayısı		15		
Uygunluk Hesabı		Karayolu		
CrossOver Yüzdesi			70	11
Mutation Yüzdesi			30	5

Labels on the left side of the table:

- Tanımlamalarda seçilen şehirlerin sayısı
- İlk Oluşturulacak Popülasyon sayısı
- Her Kromozomun ayrı ayrı uygunluk değeri
- Kromozomların Başlangıç değeri (Başlangıç Şehri)
- Çaprazlama ve Mutasyon işlemlerinin toplam deneme sayısı (Crossover Sayısı * Mutation Sayısı)
- Uygunluk hesabının hangi metod ile yapılacağını belirler. (Karayolu, Kuş uçuşu)
- Crossover Yüzde Değeri
- Mutasyon Yüzde Değeri

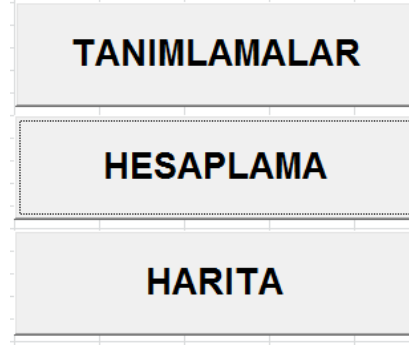
Labels on the right side of the table:

- En uygun Kromozomun uygunluk değeri
- En uygun Kromozomun Dizilişi

Şekil 2.2: Uygulama Parametre Ekranı.

3.1.5. Çalışma Adımları

Programda yer alan Uygulama Menüsü Şekil 2.3 ile gösterilmiş ve açıklanmıştır.



Şekil 2.3: Uygulama Menüsü.

Adım 1: Tanımlamalar

Problemden önce müşteriler (Türkiye illeri) belirlenir. Müşterileri belirlemek için “Tanımlar” kısmından müşterilerin yanına X işareti koyularak seçilir. Sonra aracın başlama ve bitiş yeri olan ve başlangıç şehri olan ana depo belirlenir. Böylece rota oluşturulmuş olur. Uygulama rota seçim ekranı Şekil 2.4’te gösterilmiştir.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Plaka	Şehir Adı	X	Y	Route	Başlangıç		Plaka	Şehir Adı	SıraNo			Plaka	Sıra		İşaretlenen Şehir Sayısı
2	1	ADANA	50	14			X	3	AFYON	1						10
3	2	ADYAMAN	65	21				7	ANTALYA	2						
4	3	AFYON	25	29	X	X		11	BİLECİK	3						
5	4	AGRI	89	40				15	BURDUR	4						
6	5	AMASYA	51	46				19	ÇORUM	5						
7	6	ANKARA	35	39				26	ESKİŞEHİR	6						
8	7	ANTALYA	24	12	X			33	MERSİN	7						
9	8	ARTVIN	82	53				35	İZMİR	8						
10	9	AYDIN	10	21				38	KAYSERİ	9						
11	10	BALIKESİR	10	39				40	KIRŞEHİR	10						
12	11	BİLECİK	22	42	X		X	3	AFYON	11						
13	12	BİNGÖL	77	33												
14	13	BITLİS	85	29												
15	14	BOLU	31	47												
16	15	BURDUR	21	17	X											
17	16	BURSA	16	43												
18	17	ÇANAKKALE	3	42												
19	18	ÇANKIRI	39	47												
20	19	ÇORUM	46	46	X											
21	20	DENİZLİ	17	21												
22	21	DIYARBAKIR	76	23												
23	22	EDİRNE	4	53												
24	23	ELAZIĞ	71	28												
25	24	ERZİNCAN	69	38												
26	25	ERZURUM	81	44												
27	26	ESKİŞEHİR	27	37	X											
28	27	GAZİANTEP	61	14												
29	28	GİRESUN	65	47												
30	29	GÜMÜŞHANE	69	45												
31	30	HAKKARİ	95	22												
32	31	HATAY	54	7												
33	32	İSPARTA	25	22												
34	33	MERSİN	41	8	X											
35	34	İSTANBUL	16	51												
36	35	İZMİR	5	26	X											
37	36	KARS	90	47												
38	37	KASTAMONU	41	54												
39	38	KAYSERİ	52	28	X											
40	39	KIRKLARELİ	8	57												
41	40	KIRŞEHİR	43	34	X											

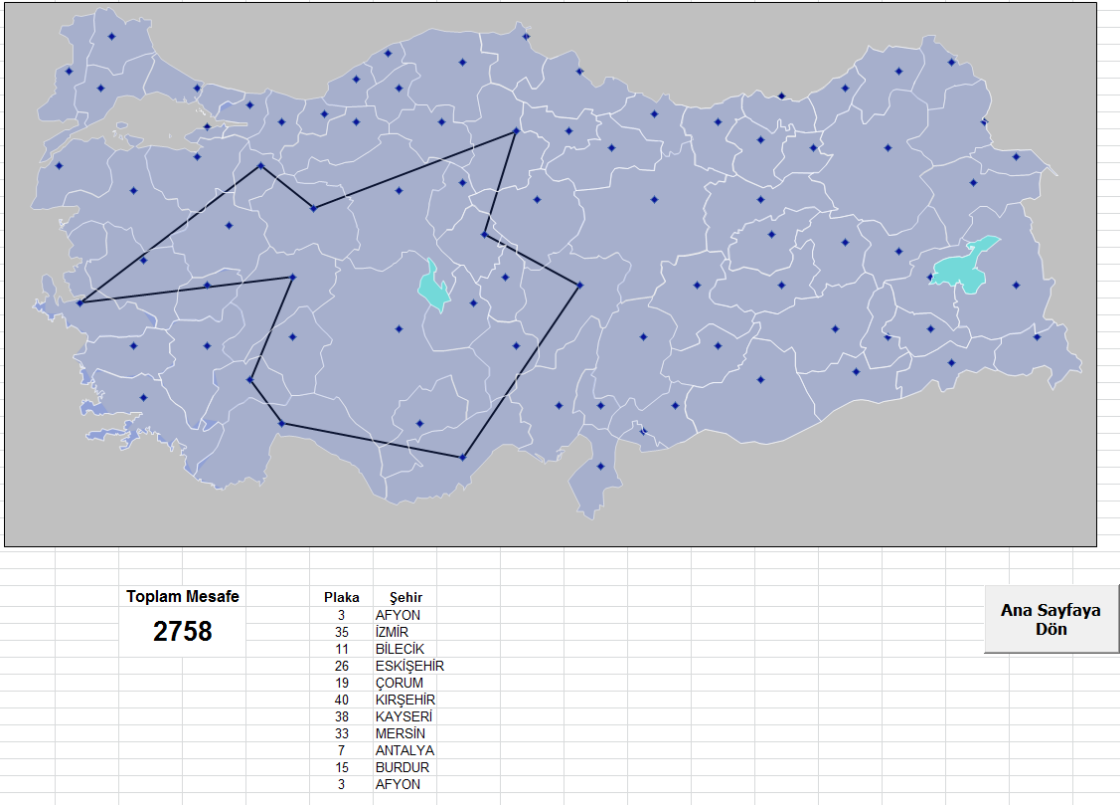
Şekil 2.4: Uygulama Rota Seçim Ekranı.

Adım 2: Uygulama

Tanımlama kısmı tamamlandıktan sonra müşteri sayısı ve depo belirlenmiş olunur. Tanımlanacak değişkenlerimiz; Popülasyon Size, Deneme Sayısı, Çaprazlama Yüzdesi, Mutasyon Yüzdesidir.

Bu deęişkenler tanımlandıktan sonra programın alıřma mantıęı kısaca; popölasyon sayısı kadar rastgele rota oluřturulur. Belirlenmiř olan aprazlama ve mutasyon yölzdesine göre deneme sayısı kadar, uygunluk deęerini saęlayacak yeni rotalar üretilmeye bařlanır. Her rota için uygunluk deęeri hesaplanır, en yüksek uygunluk deęerine sahip olan rotalar en üste alınır ve bu rotalar aprazlama ve mutasyona uęratılarak yeni rotalar elde edilir. Sonuç olarak Gezgin Satıcı Yöntemine göre en iyi rota bulunarak problem özölür.

En iyi rota bulunduğktan sonra “Harita Oluřtur” ile görsel olarak en iyi rota görsel hale getirilir. Harita Oluřturma Ekranı Őekil 2.5’ te paylařılmıştır.



Őekil 2.5: Uygulama Harita Oluřturma Ekranı.

3.2. EXCEL SOLVER (EVOLUTIONARY) GENETİK ALGORİTMALAR PROGRAMI

Evolutionary Solver programı, genetik ve evrimsel algoritmalar uygulayarak optimizasyon problemlerinin özölümünde kullanılır. Program, birok olasılık oluřturarak bu olasılıkların içinden en iyilerini seer ve bunlara genetik algoritmaları uygulayarak

en iyiyi bulmaya çalışır. Solver içinde bulunan bu metodun doğru çalışabilmesi için verilen amaç fonksiyonunun doğruluğu çok önemlidir. Solver izin verilen alandaki değerleri bahsedilen genetik operatörler kullanarak değiştirir ve her değişim sonunda amaç fonksiyonu sonucuna göre dizilimi değerlendirir. En iyi sonucu bulana dek ya da belirtilen kısıtlara ulaşına dek bu operatörleri uygulamaya devam eder.

3.3 EN YAKIN KOMŞU ALGORİTMASI

En Yakın Komşu Algoritmasının genel mantığı, başlangıç olarak seçilen bir düğüme en yakın olan ve daha önce ziyaret edilmemiş düğüm, sonraki ziyaret edilecek düğüm olarak seçilir. Ardından gidilen bu düğümede daha önce ziyaret edilmemiş en yakın düğüm seçilerek tanımlanan tüm düğümler ziyaret edilene kadar bu seçim işlem adımları devam ettirilir. Algoritmanın işlem adımları aşağıdaki gibi özetlenebilir:

Adım 1 : Seçilen düğüme en yakın mesafedeki düğüme git.

Adım 2 : Bu düğüm daha önce ziyaret edilmiş mi? Evet ise Adım 1' e geri dön ve sonraki en yakın düğüme git. Hayır ise düğümü rotaya ekle ve Adım 3' e git.

Adım 3 : Tüm düğümler ziyaret edildi mi? Evet ise işlemleri bitir, Hayır ise Adım 1'e geri dön.

3.4 UYGULAMA

3.3.1. Problemin Tanımı

Örnek uygulamamızda 6 adet düğüm sayıları farklı araç rotalama problemleri 3 farklı yöntem ile ele alınarak çözülecektir. Çözümler karşılaştırılarak problemlerin çözümünde en iyi yöntem belirlenmiş olacaktır. Bu problemler,

1. 10 ilin rasgele seçimi problemi
2. 20 ilin rasgele seçimi problemi
3. 30 ilin rasgele seçimi problemi
4. 40 ilin rasgele seçimi problemi
5. 60 ilin rasgele seçimi problemi
6. 81 ilin seçimi problemi

Uygulamamızda araç rotalama problemi olarak Türkiye mesafe cetveli kullanılarak müşteriler rasgele olarak Türkiye'nin illerinden seçilecektir. Böylelikle bir gerçek dünya problemi de olan Türkiye turu problemi çözülmeye çalışılacaktır.

Uygulamamızda rasgele seçilen müşterilere yapılacak bir dağıtım rotası belirlenmesi için bir dağıtım ağı ele alınmıştır. Buna göre 10 ilin rasgele seçimi probleminde, bir il depo olarak belirleneceğinden dolayı 9 müşterisi mevcuttur. Deponun müşterilere ve müşterilerin birbirlerine olan uzaklıkları için karayolu Türkiye mesafe cetveli kullanılacaktır.

Uygulamada her şehre yalnızca bir defa uğranacak ve en son olarak başlangıç şehrine dönecektir.

Amaç fonksiyonu, diğer araç rotalama problemlerinde olduğu gibi en düşük maliyetli rotaları tespit etmektir. Bu rotalar tespit edilirken her müşterinin sadece bir araç tarafından servis görebileceği araçların rotasının depoda başlayıp depoda biteceği şartları mevcuttur.

Problemin çözümünde kullanılacak 3 yöntem, geliştirilen genetik algoritmalar programı, Excel – Solver (Evolutionary) genetik algoritmalar ve En Yakın Komşu Algoritmasıdır.

Çözüm için geliştirilen genetik algoritmalar, Excel ve Visual Basic programında kodlanmış ve problem çözümleri 4 Gb hafızada 2,6 Ghz işlemci bir bilgisayarda test edilmiştir.

Diğer programlar için çözümler de 4 Gb hafızada 2,6 Ghz işlemci bir bilgisayarda test edilmiştir.

3.3.2. Problemin En Yakın Komşu Algoritması ile Çözümü

Seçilen 6 adet problem öncelikle 1. yöntem olarak seçilen En Yakın Komşu Algoritması kullanılarak çözülecektir.

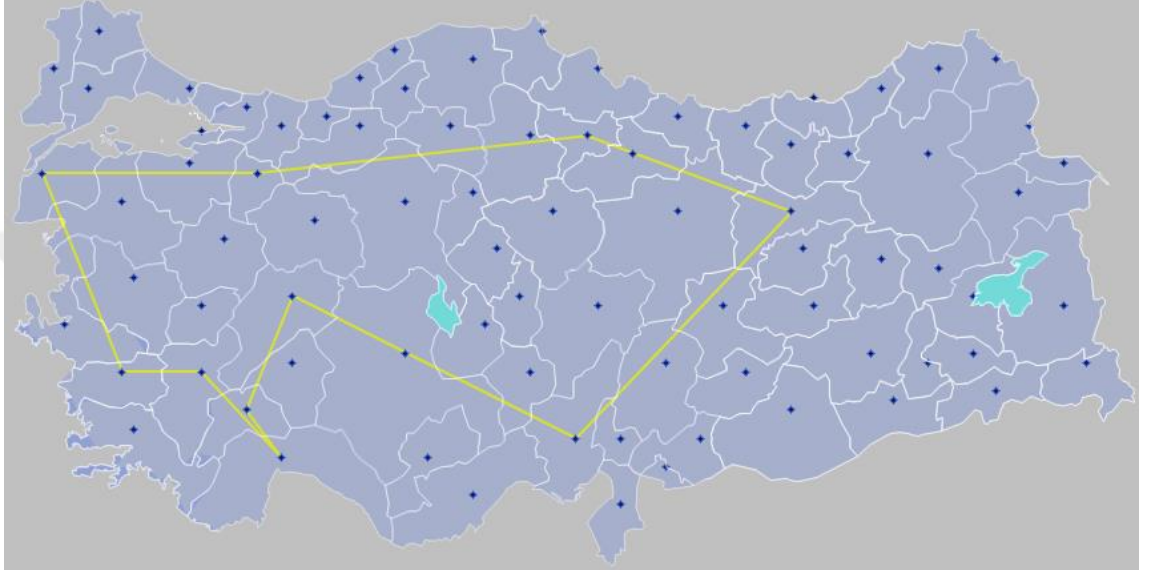
En yakın komşuluk yönteminde depodan çıkan araçlar, öncelikle depoya en yakın müşteriye gitmektedir. Daha sonra bu müşteriden en yakın olan diğer müşteriye veya

depoya gidilerek rota tamamlamaktadır. Bu işlem tüm müşteriler ziyaret edilinceye kadar devam ettirilir.

10 adet Rasgele İlin Seçimi Problemi ve Çözümü:

Çözüm Dizilimi : 3,15,7,20,9,17,11,5,24,1,3

Amaç Fonksiyonu Sonucu : 3698 Km.

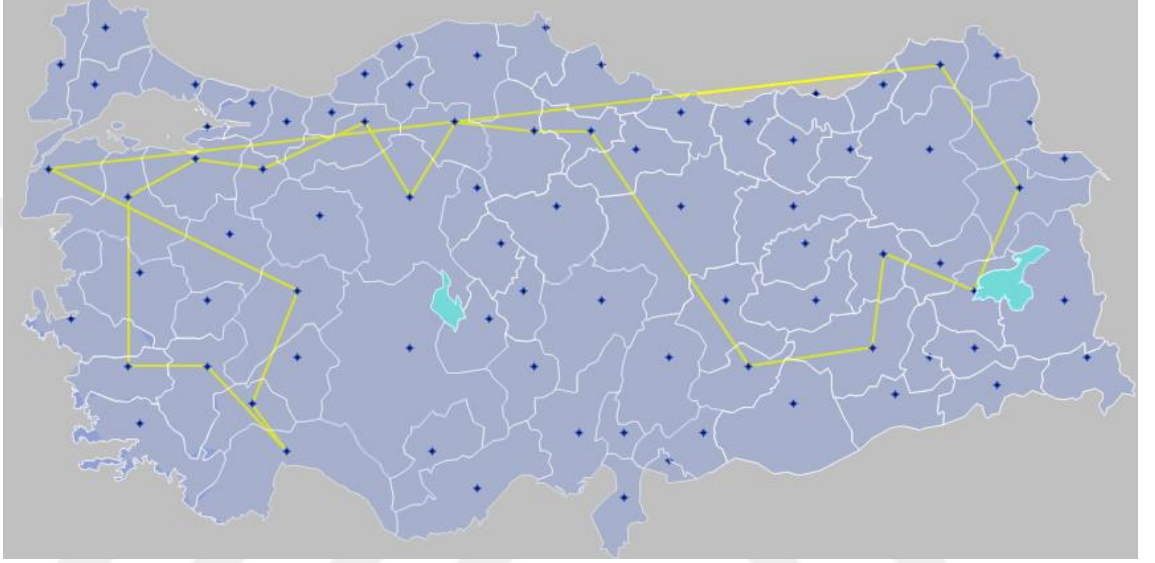


Şekil 2.6: 10 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.

30 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi :**

3,26,11,77,41,81,34,17,9,48,20,15,7,1,80,31,63,44,38,60,5,37,52,28,69,24,75,65,56,72,
3

Amaç Fonksiyonu Sonucu : 7686 Km.

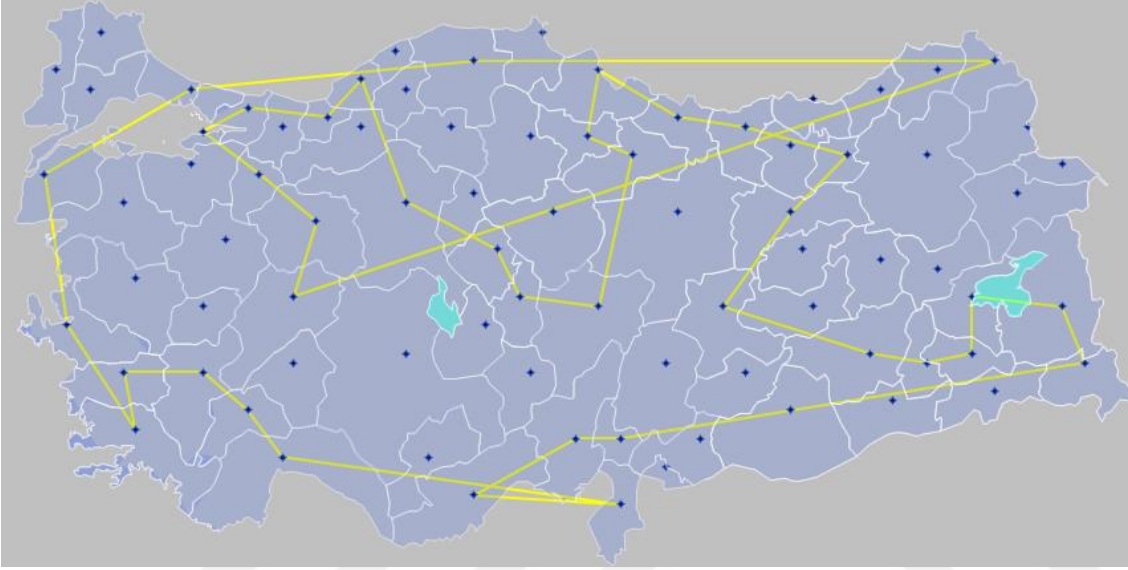


Şekil 2.8: 30 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.

40 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi :**

3,26,11,77,41,81,67,6,40,50,38,60,5,55,52,28,69,24,44,21,72,56,13,65,30,63,80,1,33,31
,7,15,20,9,48,35,17,34,37,75,3

Amaç Fonksiyonu Sonucu : 9826 Km.

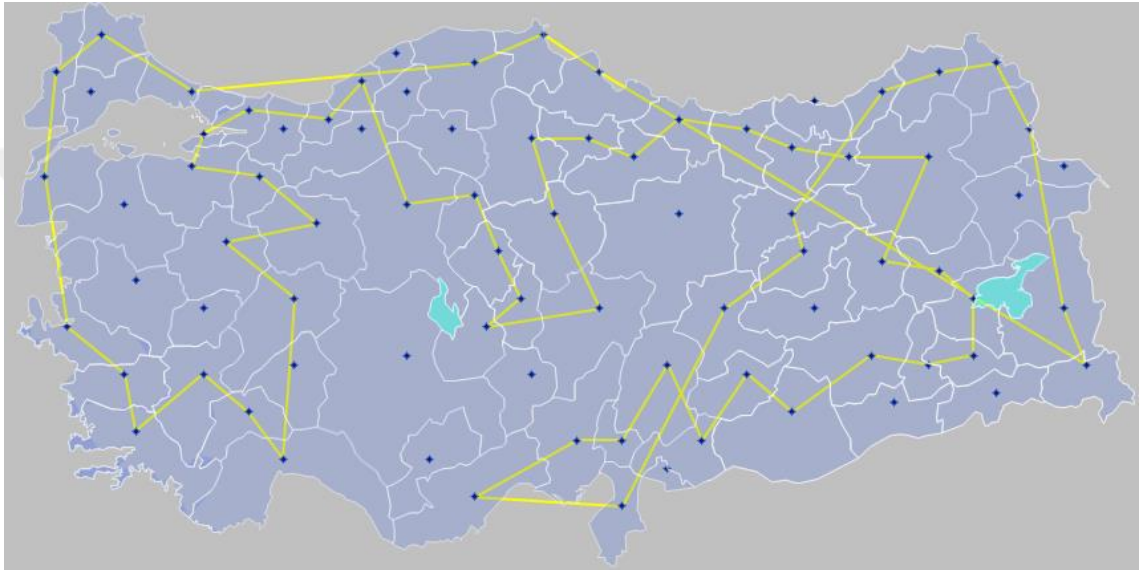


Şekil 2.9: 40 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.

60 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi :**

3,43,26,11,16,77,41,81,67,6,71,40,50,68,38,66,19,5,60,52,28,29,69,25,12,49,13,56,72,2
1,63,2,27,46,80,1,33,31,44,62,24,53,8,75,36,65,30,55,57,37,34,39,22,17,35,9,48,20,15,
7,3

Amaç Fonksiyonu Sonucu : 10060 Km.



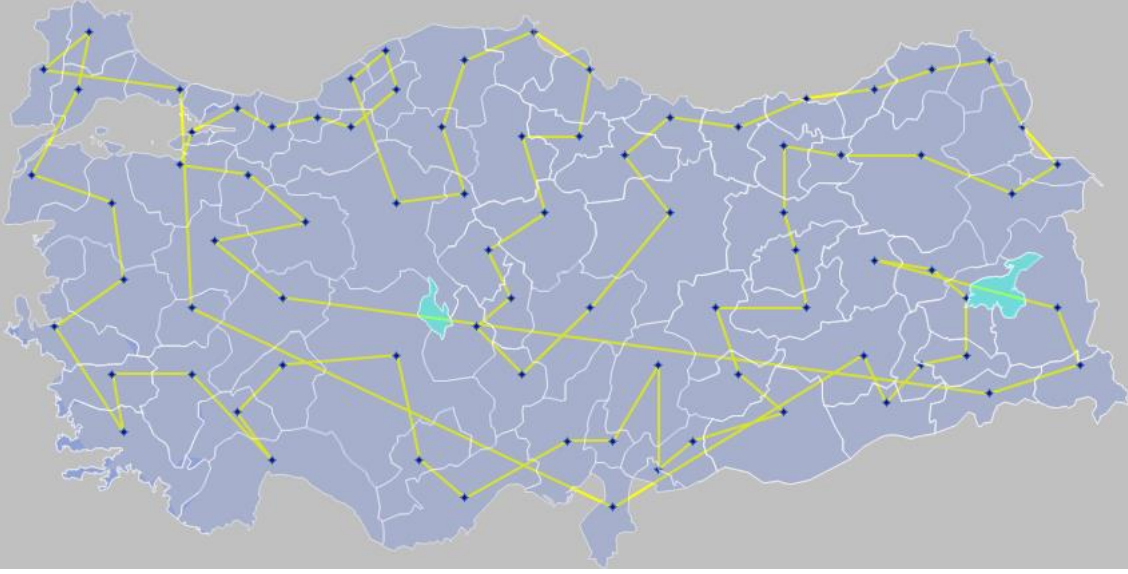
Şekil 2.10: 60 adet Rastgele İlin En Yakın Komşu Algoritması ile Çözümü.

81 İlin Seçimi Problemi ve Çözümü:

Çözüm Dizilimi :

3,43,26,11,16,77,41,54,81,14,78,74,67,6,71,18,37,57,55,5,19,66,40,50,68,51,38,58,60,5
2,28,61,53,8,75,36,76,4,25,69,29,24,62,23,44,2,63,27,79,46,80,1,33,70,42,32,15,7,20,9,
48,35,45,10,17,59,39,22,34,64,31,21,47,72,56,13,49,12,65,30,73,3

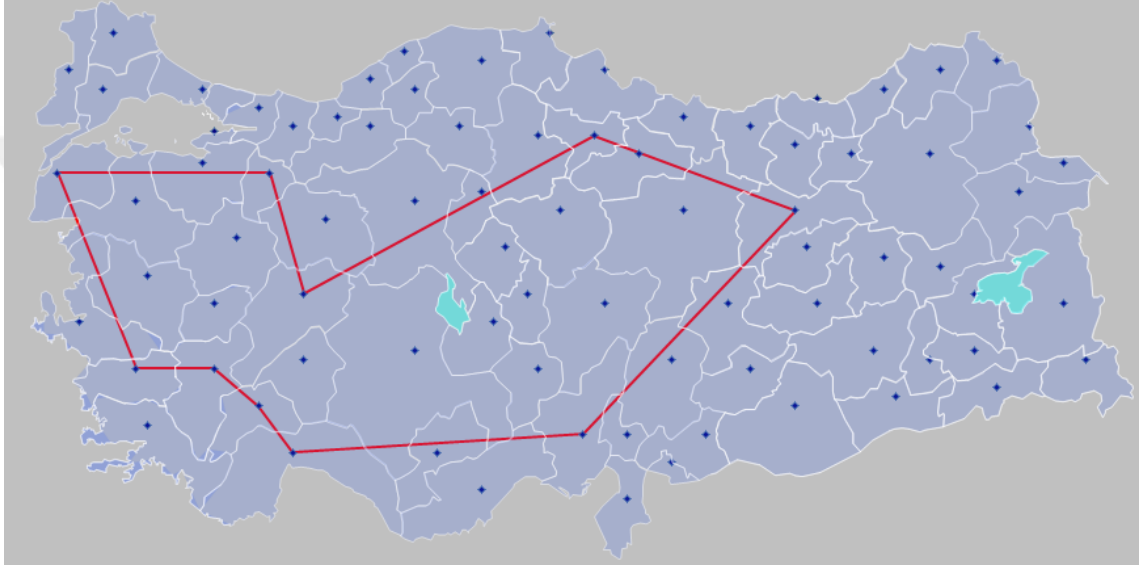
Amaç Fonksiyonu Sonucu : 12894 Km.



Şekil 2.11: 81 ilin En Yakın Komşu Algoritması ile Çözümü.

3.3.3. Problemin Excel – Solver (Evolutionary) Genetik Algoritmalar Programı ile Çözümü

Seçilen 6 adet problemin çözümü için 2. yöntem olarak “Excel – Solver” menüsünde yer alan evrimsel algoritma uygulaması kullanılacaktır. Her problem için program 5 defa çalıştırılarak çözümler alınacak ve nihai sonuç olarak karşılaştırmalarda 5 çözümün ortalaması ve standart sapması kullanılacaktır. Fakat 20’ den fazla il seçiminde Excel – Solver Genetik Algoritmalar Programı kullanılarak çözüm sağlanamadığı gözlemlenmiştir.

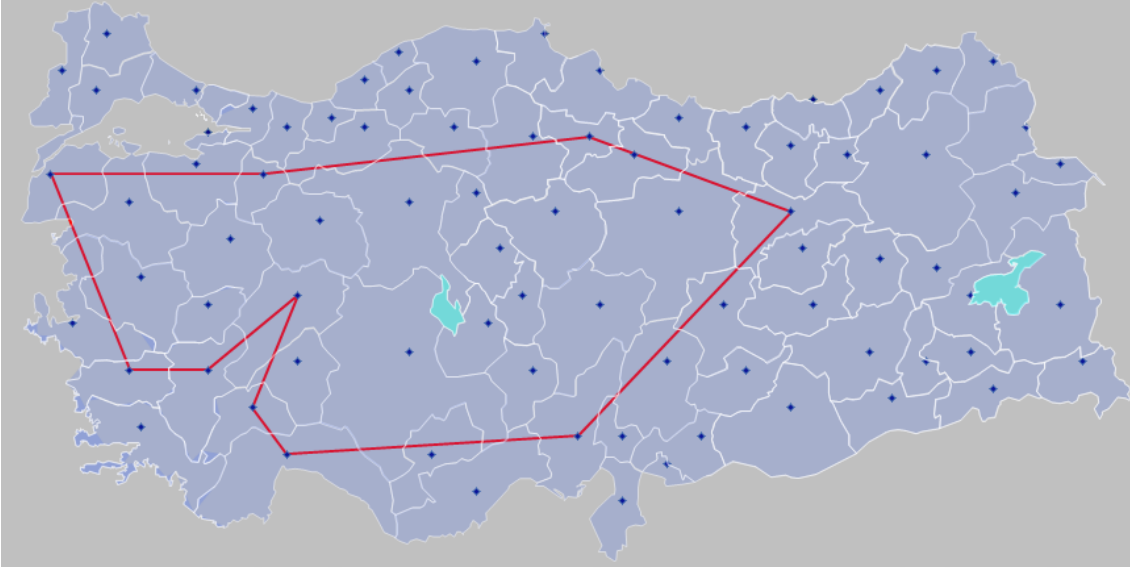
10 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):** 3,11,17,9,20,15,7,1,24,5,3**Amaç Fonksiyonu Sonucu :** 3618 Km.**Amaç Fonksiyonu Ortalaması :** 3659 Km.**Amaç Fonksiyonu Standart Sapması :** 37,2 Km.**Şekil 2.12:** 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#1).

Çözüm Dizilimi (#2): 3,15,7,1,24,5,11,17,9,20,3

Amaç Fonksiyonu Sonucu : 3686 Km.

Amaç Fonksiyonu Ortalaması : 3659 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



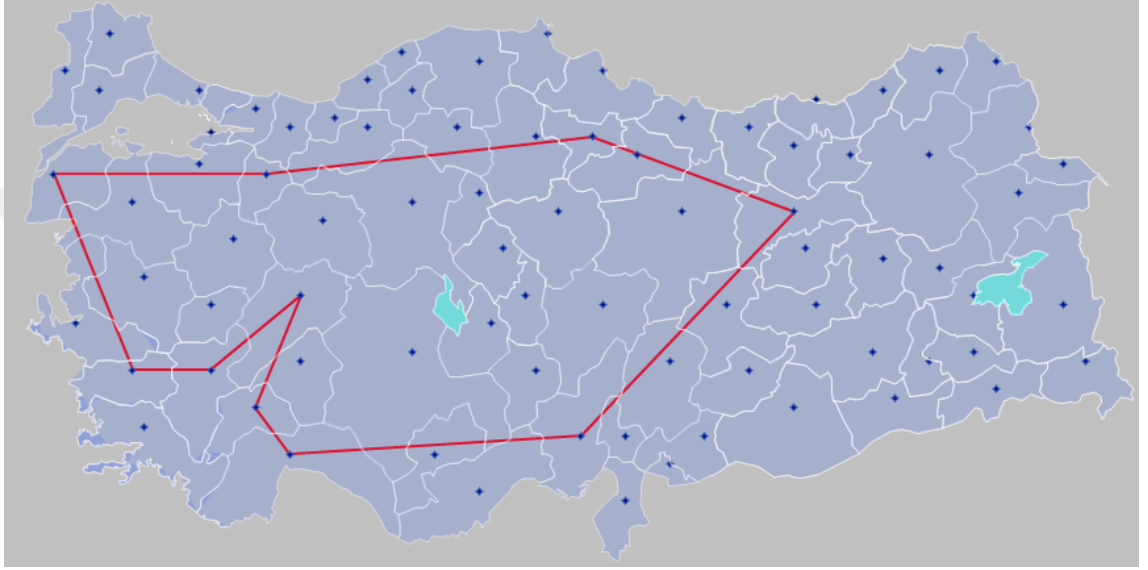
Şekil 2.13: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#2).

Çözüm Dizilimi (#3): 3,15,7,1,24,5,11,17,9,20,3

Amaç Fonksiyonu Sonucu : 3686 Km.

Amaç Fonksiyonu Ortalaması : 3659 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



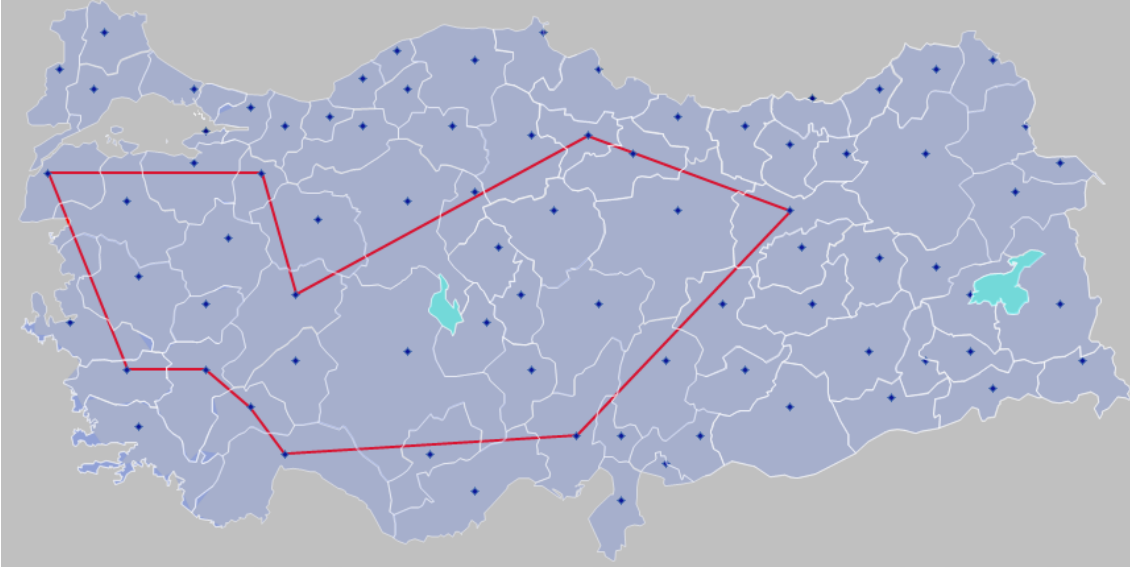
Şekil 2.14: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#3).

Çözüm Dizilimi (#4): 3,5,24,1,7,15,20,9,17,11,3

Amaç Fonksiyonu Sonucu : 3618 Km.

Amaç Fonksiyonu Ortalaması : 3659 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



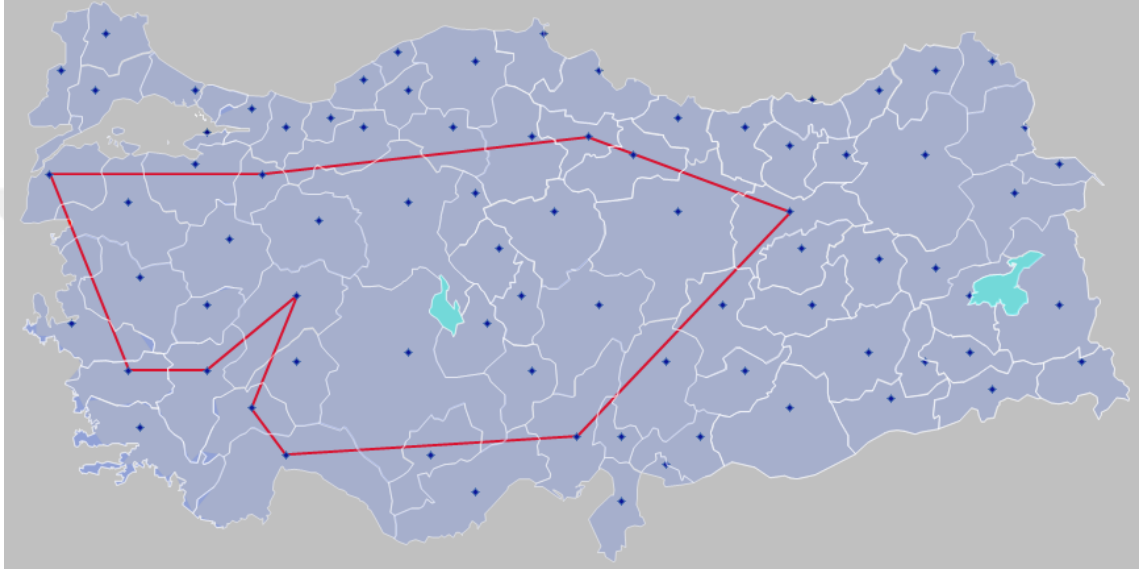
Şekil 2.15: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#4).

Çözüm Dizilimi (#5): 3,15,7,1,24,5,11,17,9,20,3

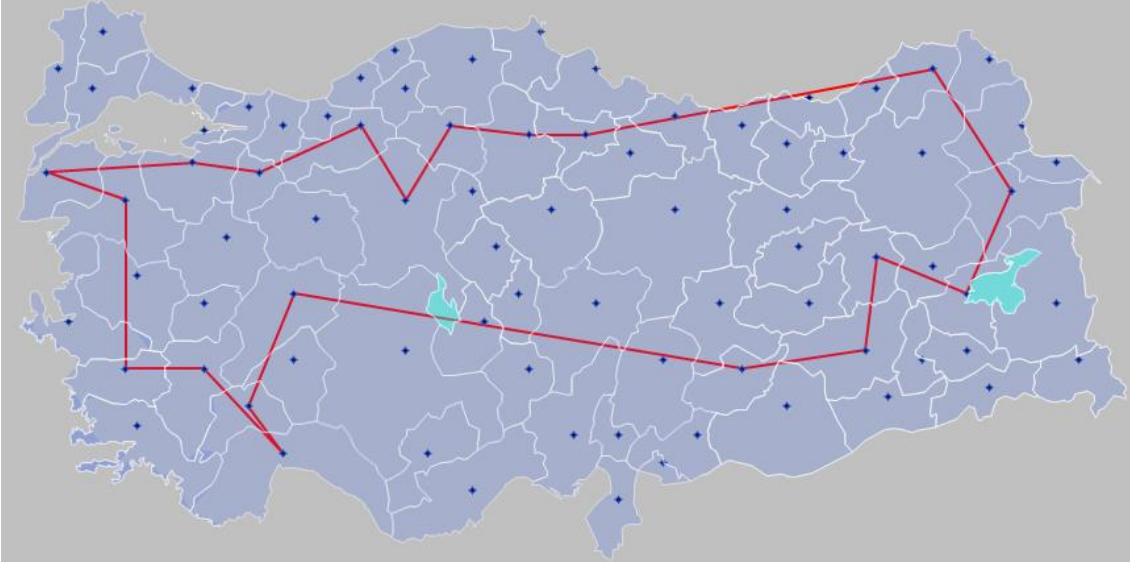
Amaç Fonksiyonu Sonucu : 3686 Km.

Amaç Fonksiyonu Ortalaması : 3659 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



Şekil 2.16: 10 adet Rastgele İlin Excel Solver GA ile Çözümü (#5).

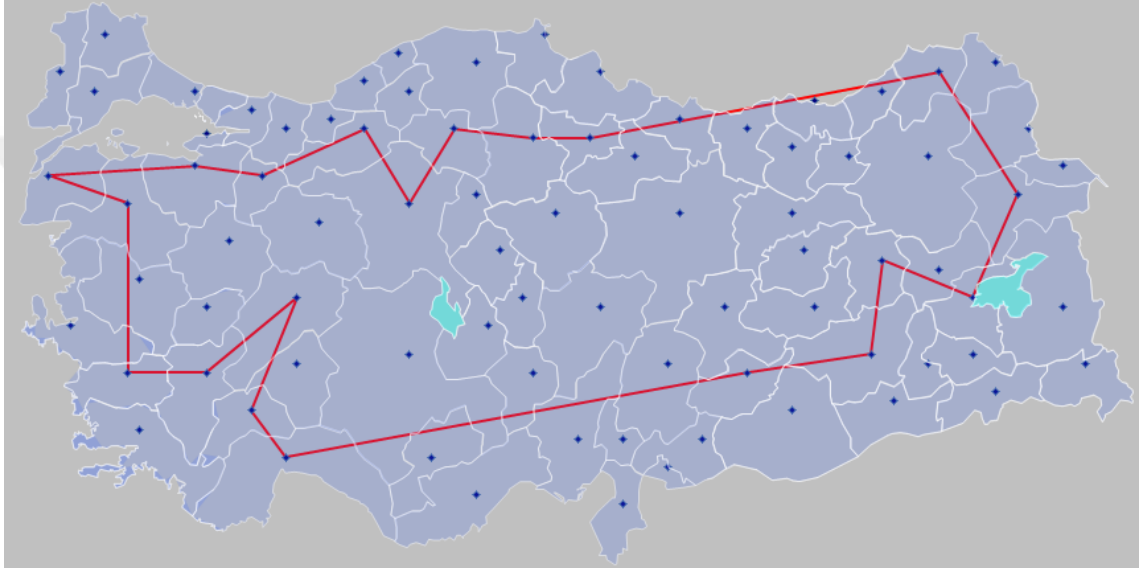
20 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):** 3,2,21,12,13,4,8,5,19,18,6,14,11,16,17,10,9,20,7,15,3**Amaç Fonksiyonu Sonucu :** 5061 Km.**Amaç Fonksiyonu Ortalaması :**5061 Km.**Amaç Fonksiyonu Standart Sapması :** 14,3 Km.**Şekil 2.17:** 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#1).

Çözüm Dizilimi (#2): 3,15,7,2,21,12,13,4,8,5,19,18,6,14,11,16,17,10,9,20,3

Amaç Fonksiyonu Sonucu : 5049 Km.

Amaç Fonksiyonu Ortalaması :5061 Km.

Amaç Fonksiyonu Standart Sapması : 14,3 Km.



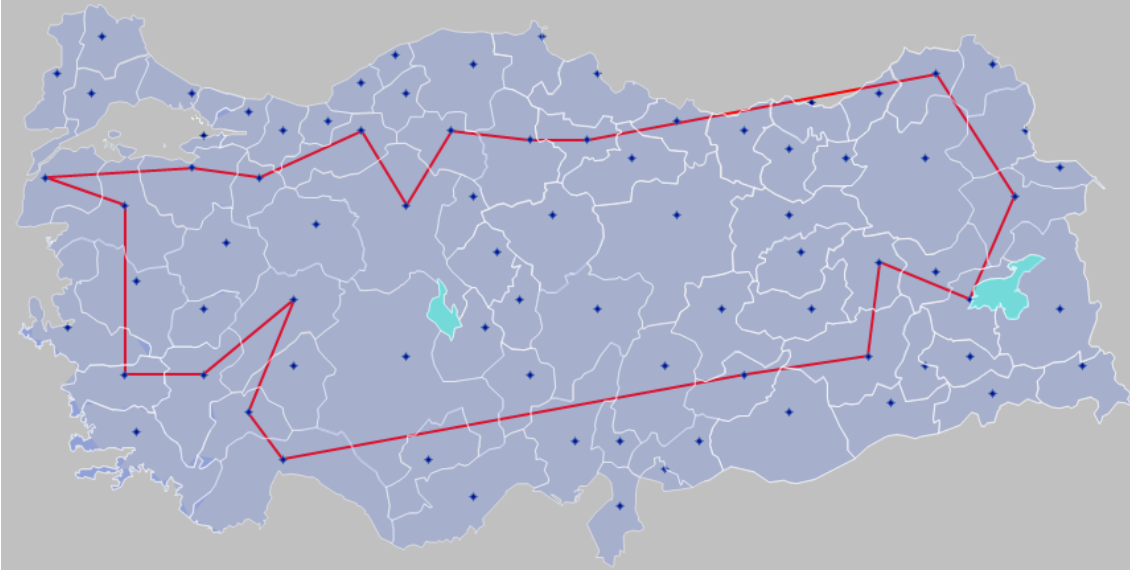
Şekil 2.18: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#2).

Çözüm Dizilimi (#3): 3,15,7,2,21,12,13,4,8,5,19,18,6,14,11,16,17,10,9,20,3

Amaç Fonksiyonu Sonucu : 5049 Km.

Amaç Fonksiyonu Ortalaması : 5061 Km.

Amaç Fonksiyonu Standart Sapması : 14,3 Km.



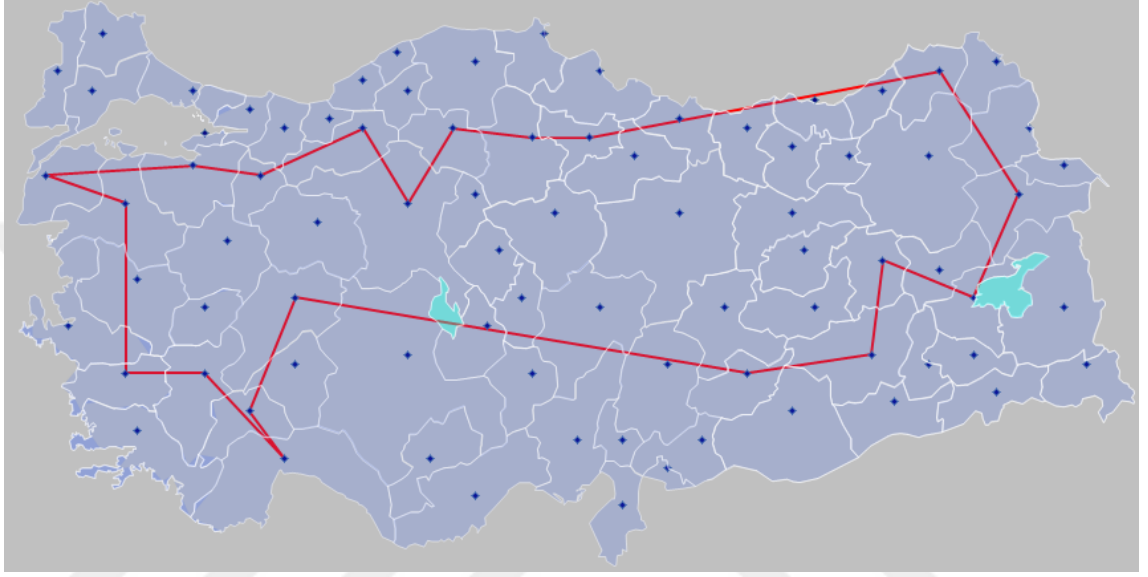
Şekil 2.19: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#3).

Çözüm Dizilimi (#4): 3,15,7,20,9,10,17,16,11,14,6,18,19,5,8,4,13,12,21,2,3

Amaç Fonksiyonu Sonucu : 5061 Km.

Amaç Fonksiyonu Ortalaması :5061 Km.

Amaç Fonksiyonu Standart Sapması : 14,3 Km.



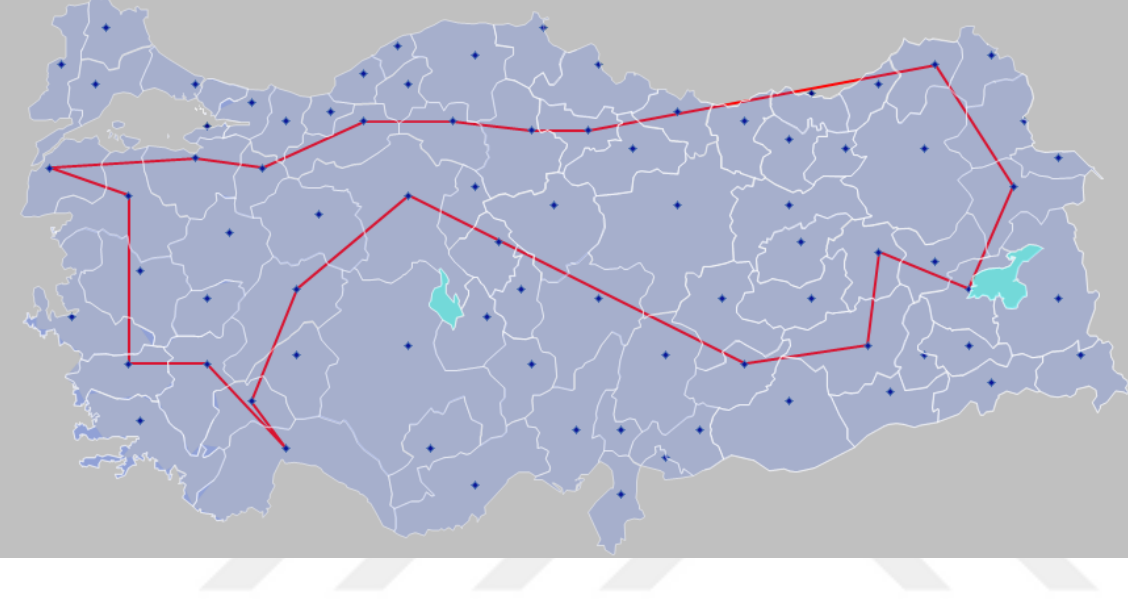
Şekil 2.20: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#4).

Çözüm Dizilimi (#5): 3,6,2,21,12,13,4,8,5,19,18,14,11,16,17,10,9,20,7,15,3

Amaç Fonksiyonu Sonucu : 5084 Km.

Amaç Fonksiyonu Ortalaması : 5061 Km.

Amaç Fonksiyonu Standart Sapması : 14,3 Km.



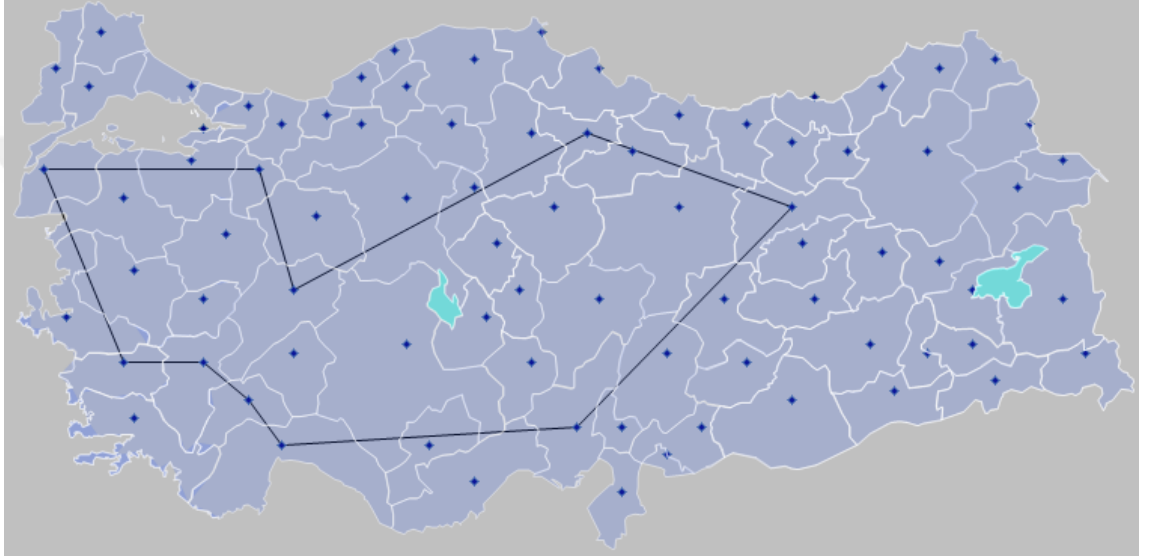
Şekil 2.21: 20 adet Rastgele İlin Excel Solver GA ile Çözümü (#5).

30, 40, 60 adet Rasgele İlin Seçimi ile 81 il Problemi ve Çözümü:

30,40,60 ilin Rasgele Seçimi Problemi ile 81 il problemi Excel – Solver Genetik Algoritmalar Programı kullanılarak çözüm sağlanamamıştır.

3.3.3. Problemin Geliştirilen Algoritma Programı ile Çözümü

Seçilen 6 adet problemin çözümü için 3. yöntem olarak geliştirilen algoritma uygulaması kullanılacaktır. Her bir problem için program 5 defa çalıştırılarak çözümler alınacaktır ve nihai çözüm olarak karşılaştırmalarda 5 çözümün ortalaması ve standart sapması kullanılacaktır.

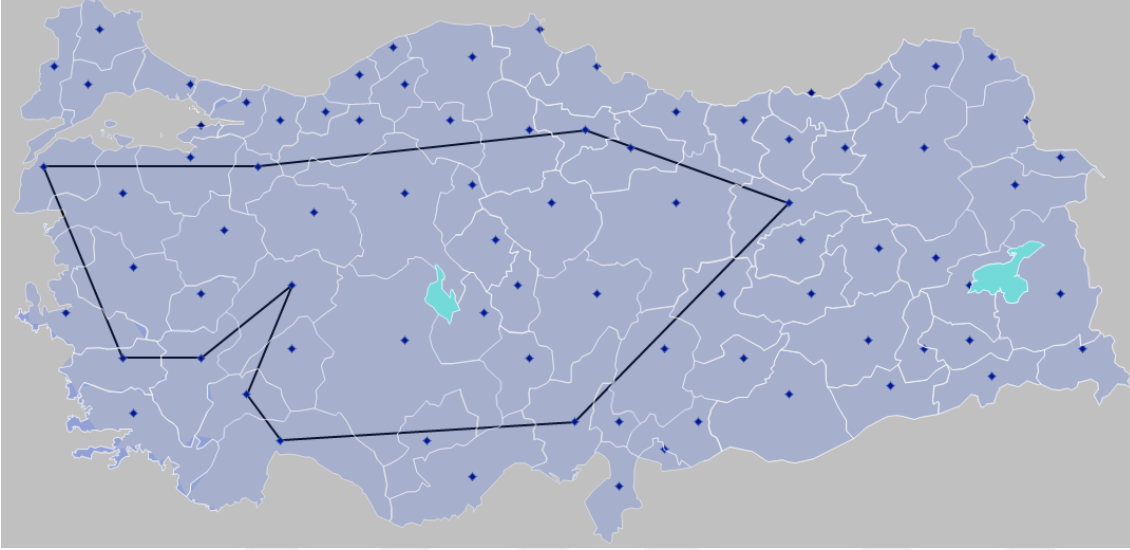
10 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):** 3,5,24,1,7,15,20,9,17,11,3**Amaç Fonksiyonu Sonucu :** 3618 Km.**Amaç Fonksiyonu Ortalaması :** 3645 Km.**Amaç Fonksiyonu Standart Sapması :** 37,2 Km.**Şekil 2.22:** 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).

Çözüm Dizilimi (#2): 3,20,9,17,11,5,24,1,7,15,3

Amaç Fonksiyonu Sonucu : 3686 Km.

Amaç Fonksiyonu Ortalaması : 3645 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



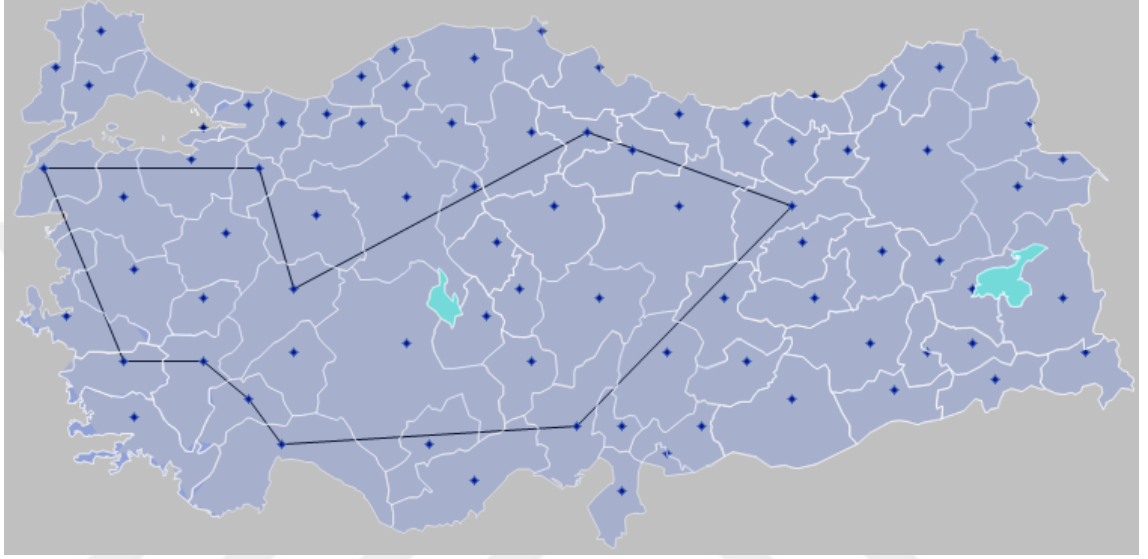
Şekil 2.23: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).

Çözüm Dizilimi (#3): 3,5,24,1,7,15,20,9,17,11,3

Amaç Fonksiyonu Sonucu : 3618 Km.

Amaç Fonksiyonu Ortalaması : 3645 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



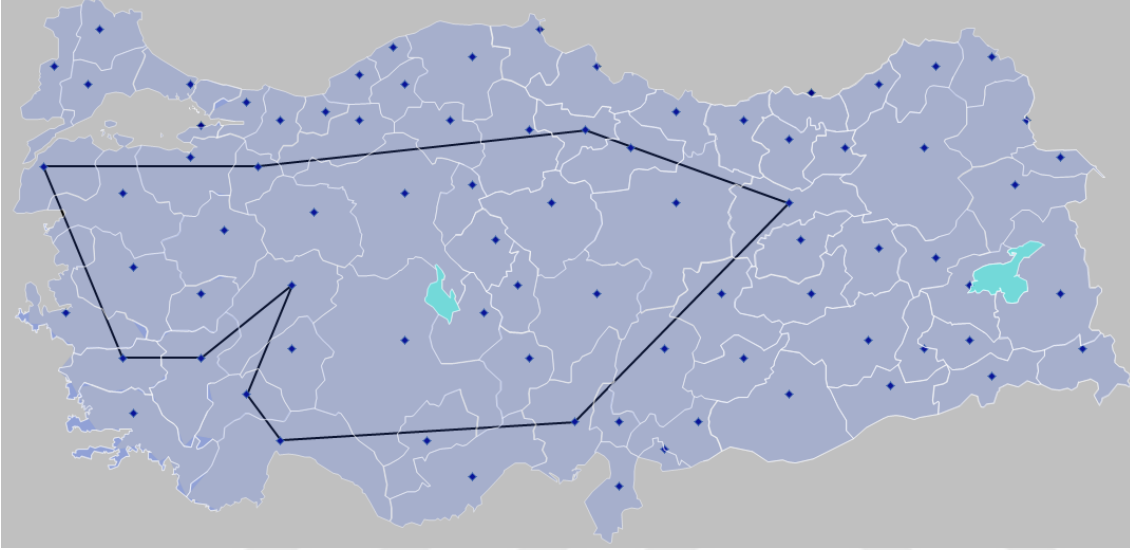
Şekil 2.24: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).

Çözüm Dizilimi (#4): 3,5,24,1,15,7,20,9,17,11,3

Amaç Fonksiyonu Sonucu : 3686 Km.

Amaç Fonksiyonu Ortalaması : 3645 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



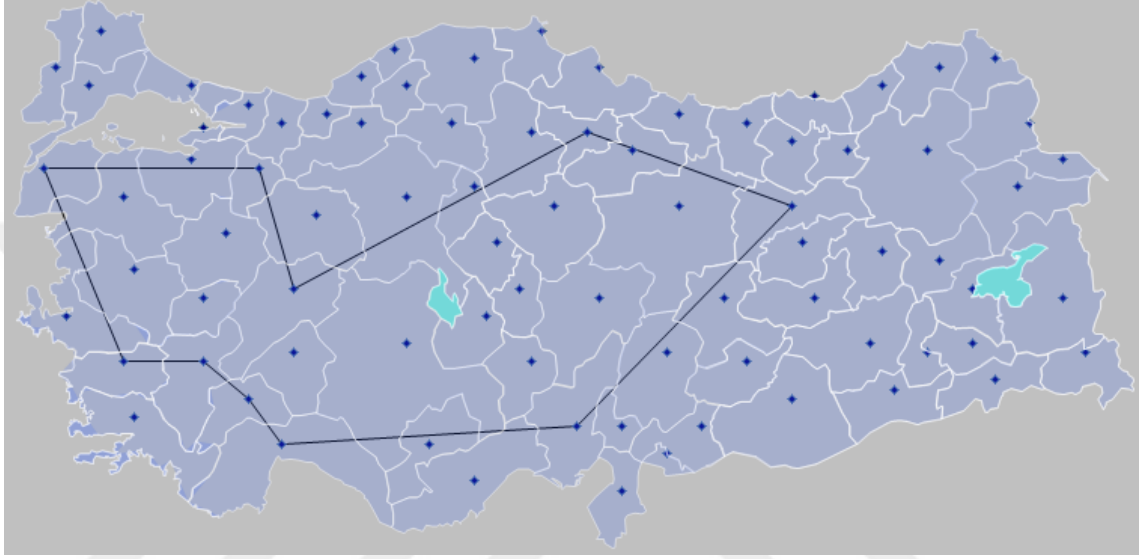
Şekil 2.25: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).

Çözüm Dizilimi (#5): 3,5,24,1,7,15,20,9,17,11,3

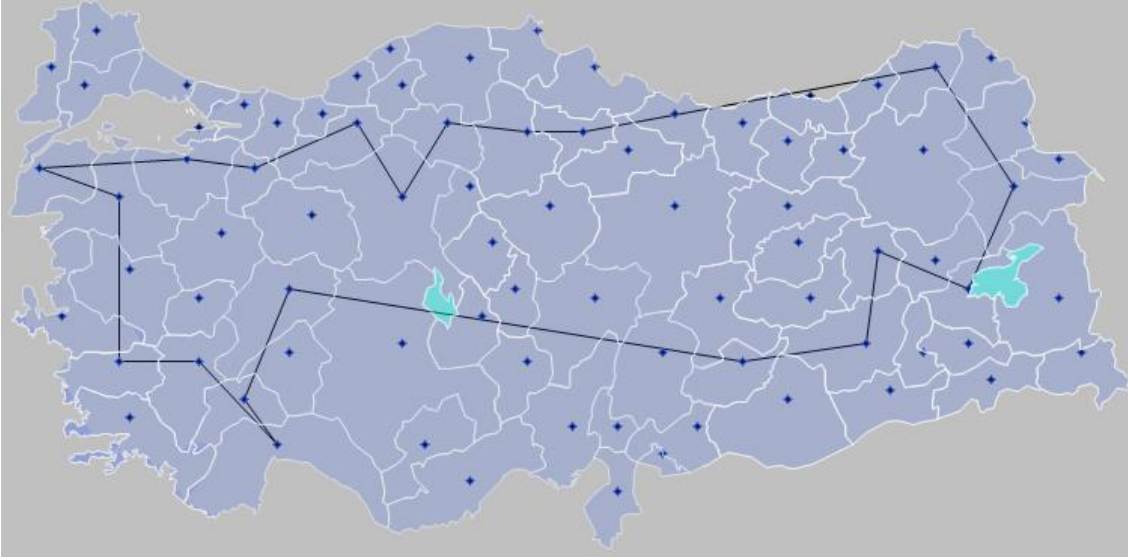
Amaç Fonksiyonu Sonucu : 3618 Km.

Amaç Fonksiyonu Ortalaması : 3645 Km.

Amaç Fonksiyonu Standart Sapması : 37,2 Km.



Şekil 2.26: 10 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).

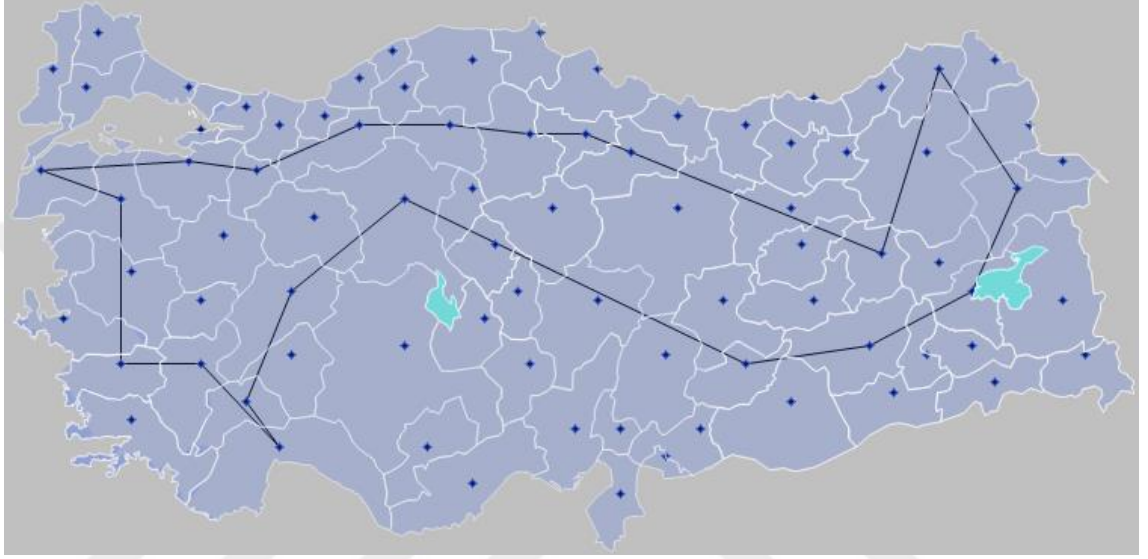
20 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):** 3,15,7,20,9,10,17,16,11,14,6,18,19,5,8,4,13,12,21,2,3**Amaç Fonksiyonu Sonucu :** 5061 Km.**Amaç Fonksiyonu Ortalaması :** 5152 Km.**Amaç Fonksiyonu Standart Sapması :** 105,7 Km.**Şekil 2.27:** 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).

Çözüm Dizilimi (#2): 3,6,2,21,13,4,8,12,5,19,18,14,11,16,17,10,9,20,7,15,3

Amaç Fonksiyonu Sonucu : 5310 Km.

Amaç Fonksiyonu Ortalaması : 5152 Km.

Amaç Fonksiyonu Standart Sapması : 105,7 Km.



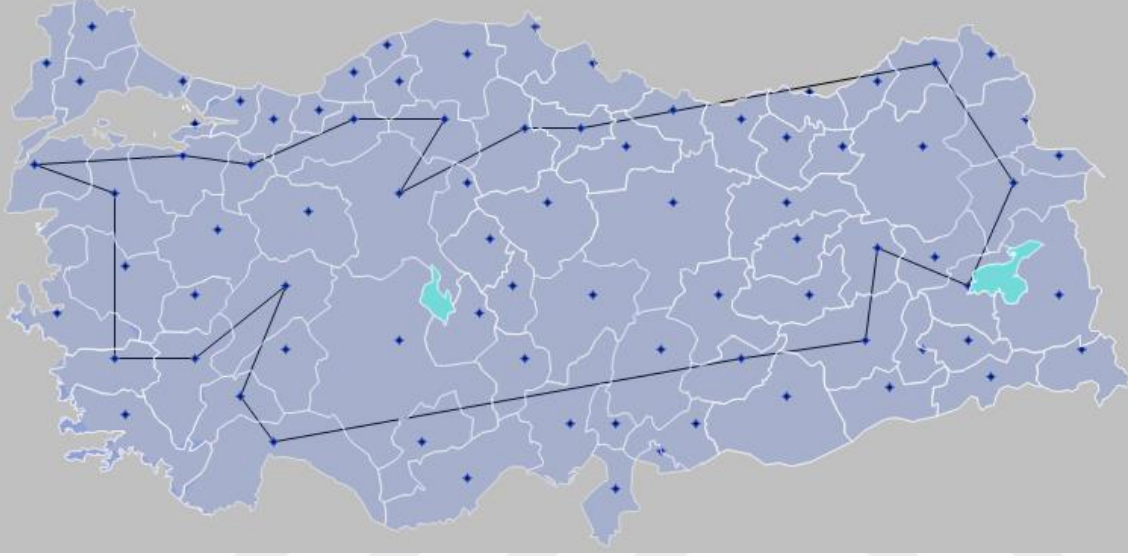
Şekil 2.28: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).

Çözüm Dizilimi (#3): 3,15,7,2,21,12,13,4,8,5,19,6,18,14,11,16,17,10,9,20,3

Amaç Fonksiyonu Sonucu : 5181 Km.

Amaç Fonksiyonu Ortalaması : 5152 Km.

Amaç Fonksiyonu Standart Sapması : 105,7 Km.



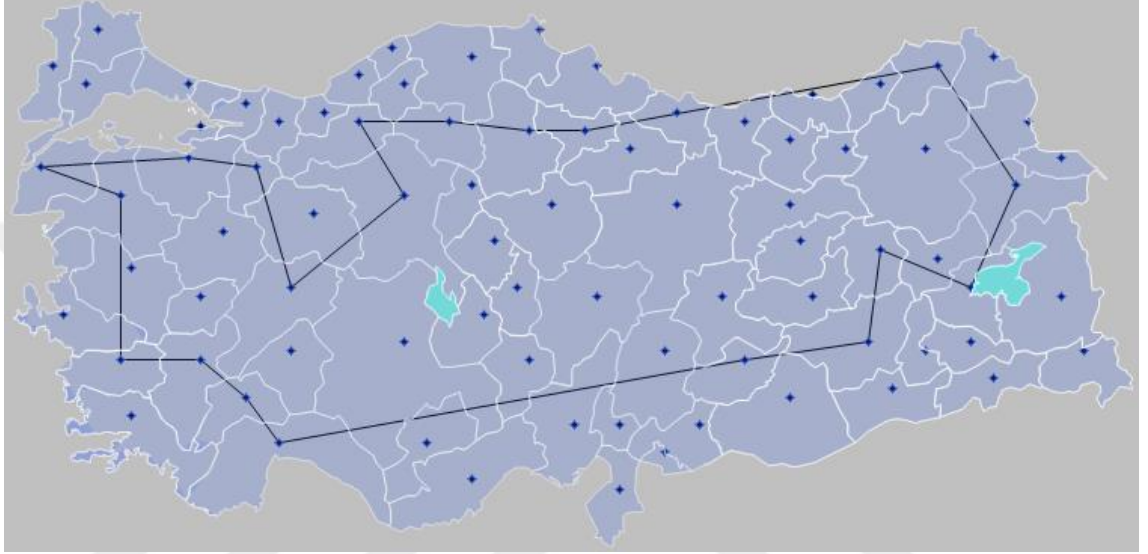
Şekil 2.29: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).

Çözüm Dizilimi (#4): 3,6,14,18,19,5,8,4,13,12,21,2,7,15,20,9,10,17,16,11,3

Amaç Fonksiyonu Sonucu : 5158 Km.

Amaç Fonksiyonu Ortalaması : 5152 Km.

Amaç Fonksiyonu Standart Sapması : 105,7 Km.



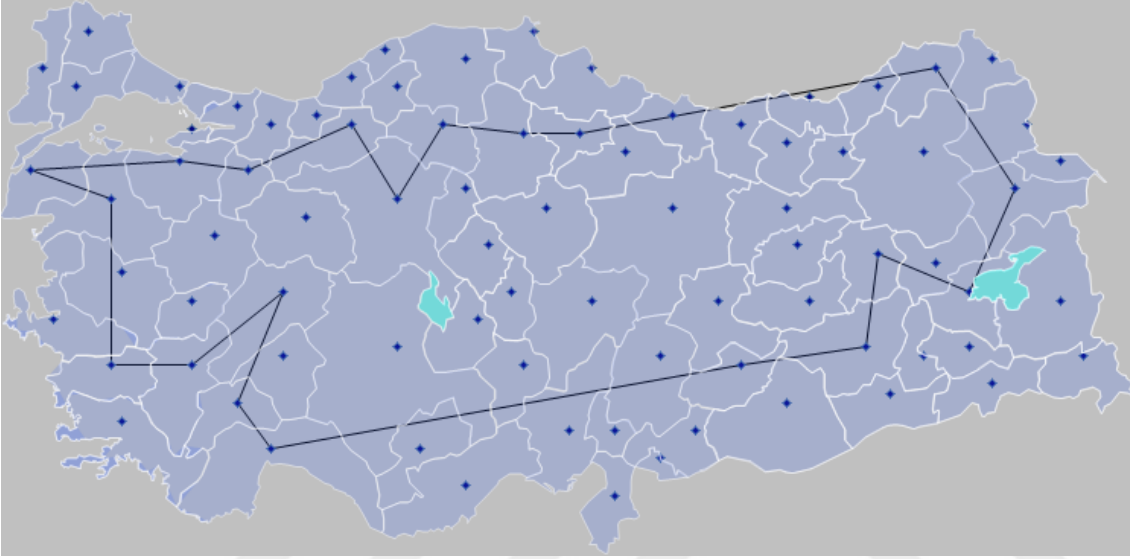
Şekil 2.30: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).

Çözüm Dizilimi (#5): 3,20,9,10,17,16,11,14,6,18,19,5,8,4,13,12,21,2,7,15,3

Amaç Fonksiyonu Sonucu : 5049 Km.

Amaç Fonksiyonu Ortalaması : 5152 Km.

Amaç Fonksiyonu Standart Sapması : 105,7 Km.



Şekil 2.31: 20 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).

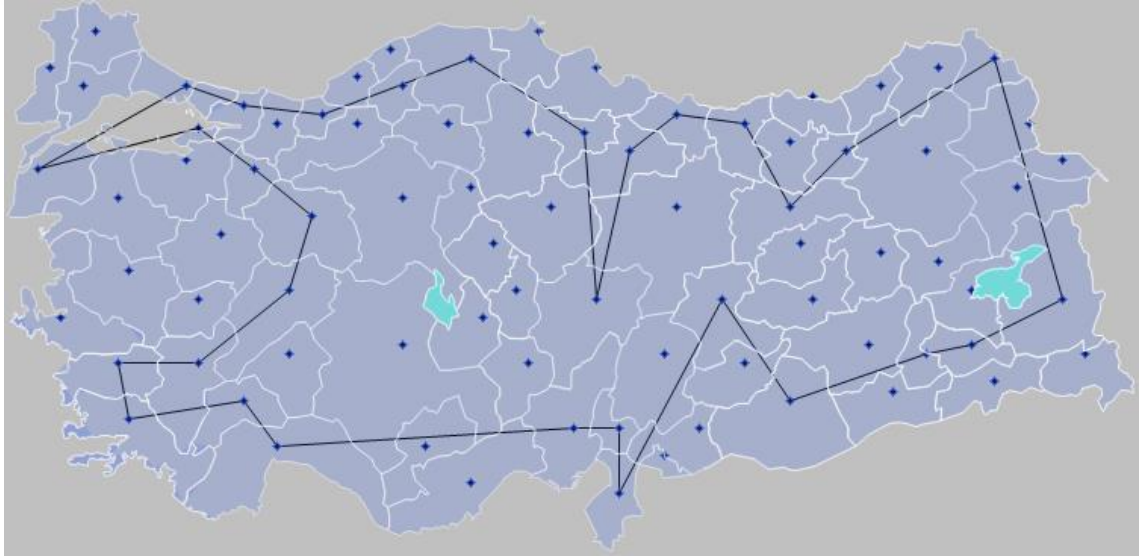
30 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):**

3,20,9,48,15,7,1,80,31,44,63,72,56,65,75,69,24,28,52,60,38,5,37,81,41,34,17,77,11,26,
3

Amaç Fonksiyonu Sonucu : 6754 Km.

Amaç Fonksiyonu Ortalaması : 6743 Km.

Amaç Fonksiyonu Standart Sapması : 56,5 Km.



Şekil 2.32: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).

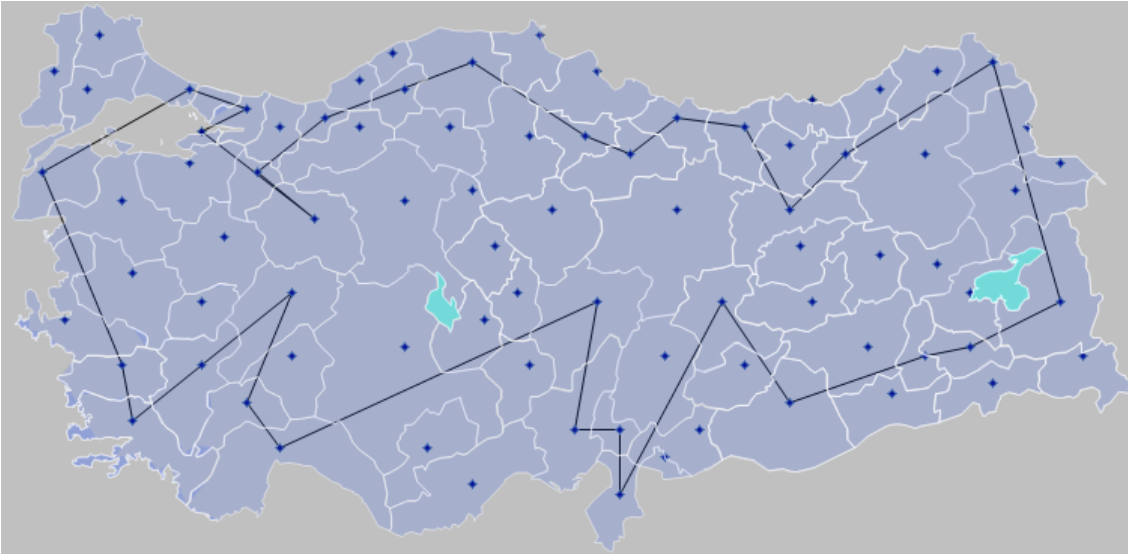
Çözüm Dizilimi (#2):

3,20,48,9,17,34,41,77,26,11,81,37,5,60,52,28,24,69,75,65,56,72,63,44,31,80,1,38,7,15,
3

Amaç Fonksiyonu Sonucu : 6781 Km.

Amaç Fonksiyonu Ortalaması : 6743 Km.

Amaç Fonksiyonu Standart Sapması : 56,5 Km.



Şekil 2.33: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).

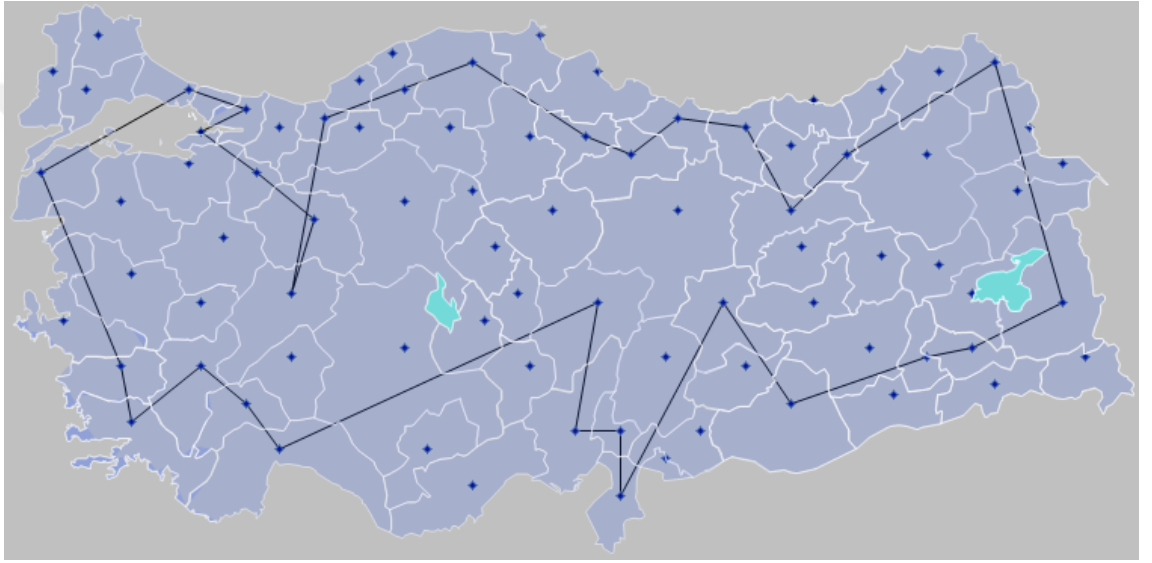
Çözüm Dizilimi (#3):

3,81,37,5,60,52,28,24,69,75,65,56,72,63,44,31,80,1,38,7,15,20,48,9,17,34,41,77,11,26,
3

Amaç Fonksiyonu Sonucu : 6810 Km.

Amaç Fonksiyonu Ortalaması : 6743 Km.

Amaç Fonksiyonu Standart Sapması : 56,5 Km.



Şekil 2.34: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).

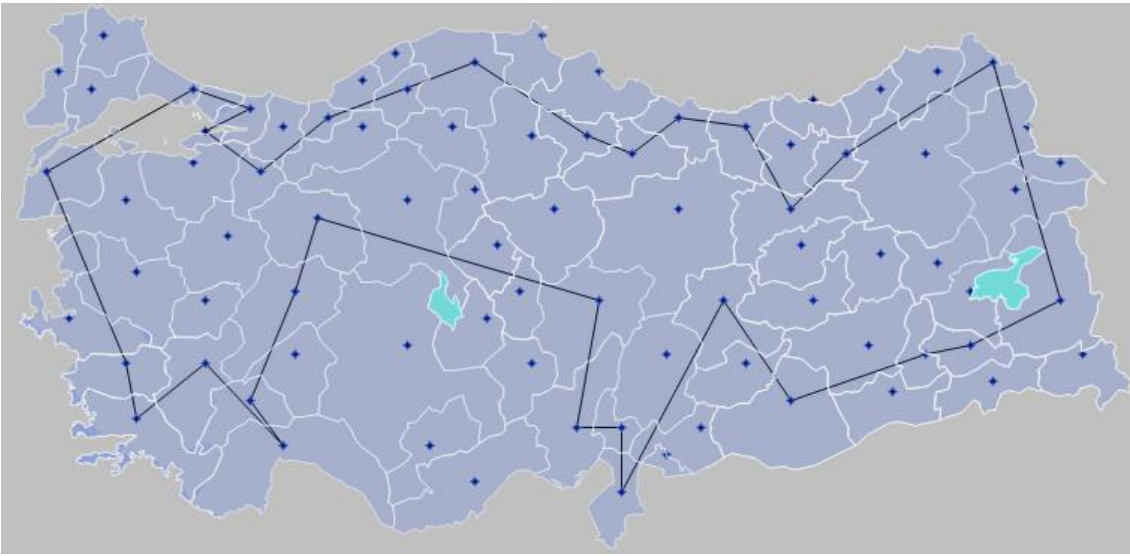
Çözüm Dizilimi (#4):

3,26,38,1,80,31,44,63,72,56,65,75,69,24,28,52,60,5,37,81,11,77,41,34,17,9,48,20,7,15,3

Amaç Fonksiyonu Sonucu : 6685 Km.

Amaç Fonksiyonu Ortalaması : 6743 Km.

Amaç Fonksiyonu Standart Sapması : 56,5 Km.



Şekil 2.35: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).

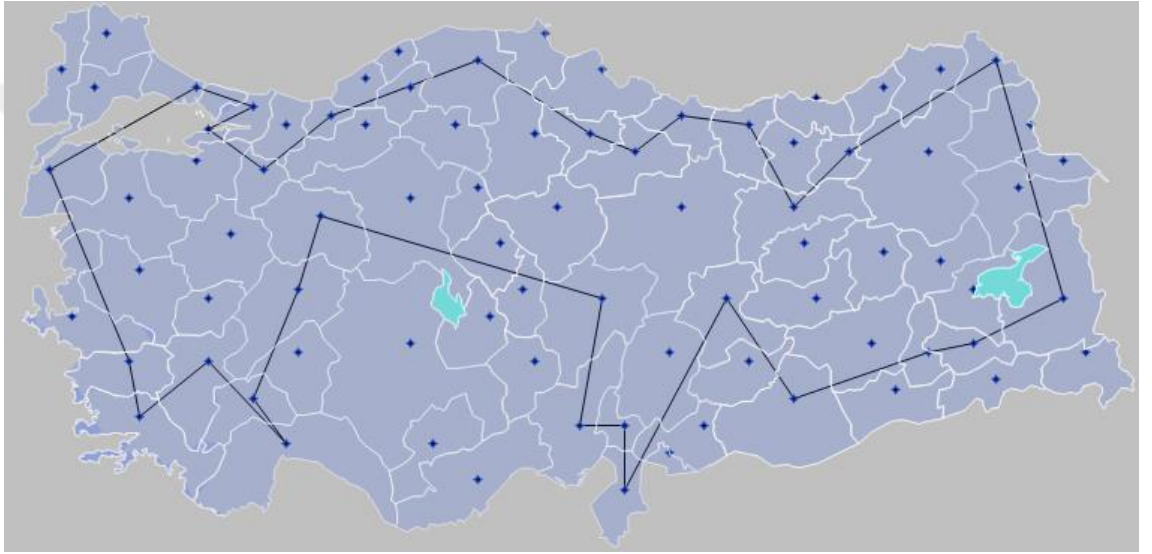
Çözüm Dizilimi (#5):

3,26,38,1,80,31,44,63,72,56,65,75,69,24,28,52,60,5,37,81,11,77,41,34,17,9,48,20,7,15,
3

Amaç Fonksiyonu Sonucu : 6685 Km.

Amaç Fonksiyonu Ortalaması : 6743 Km.

Amaç Fonksiyonu Standart Sapması : 56,5 Km.



Şekil 2.36: 30 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).

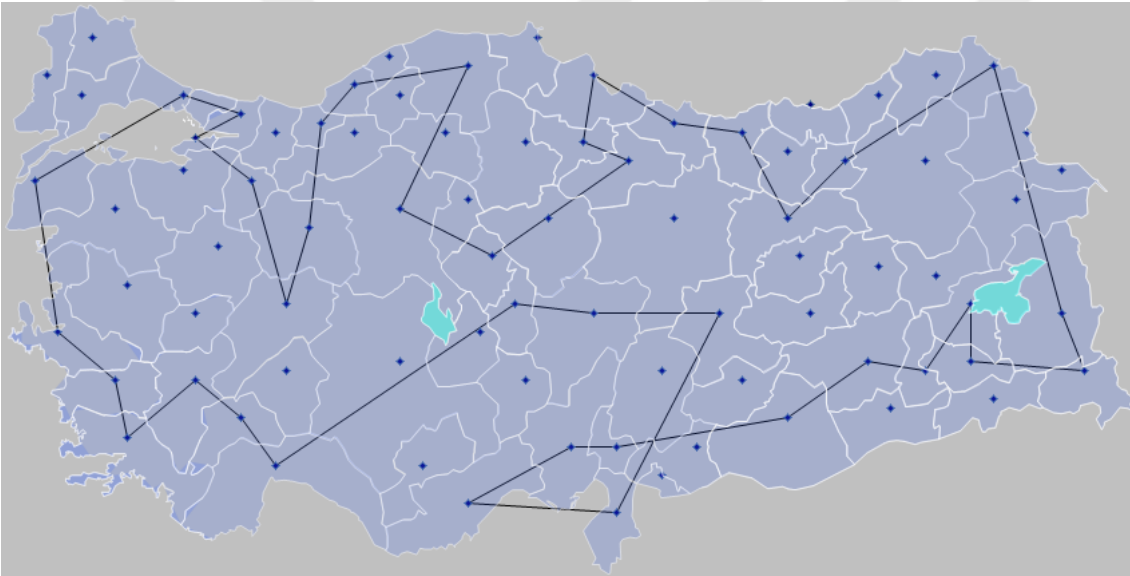
40 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):**

3,11,77,41,34,17,35,9,48,20,15,7,50,38,44,31,33,1,80,63,21,72,13,56,30,65,75,69,24,28
,52,55,5,60,40,6,37,67,81,26,3

Amaç Fonksiyonu Sonucu : 8036 Km.

Amaç Fonksiyonu Ortalaması : 7810 Km.

Amaç Fonksiyonu Standart Sapması : 224 Km.



Şekil 2.37: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).

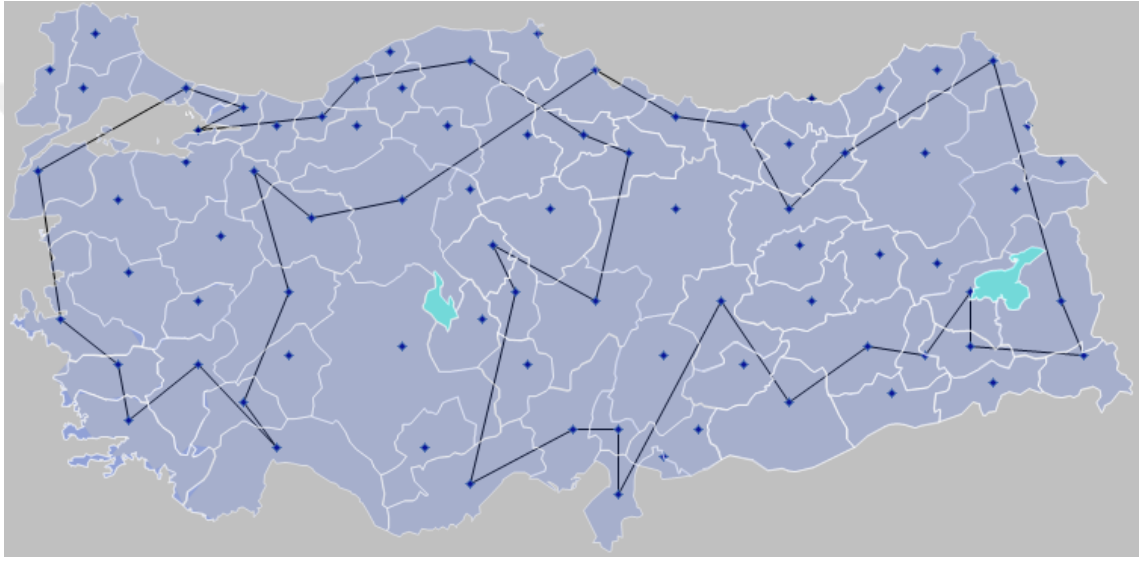
Çözüm Dizilimi (#2):

3,11,26,6,55,52,28,24,69,75,65,30,56,13,72,21,63,44,31,80,1,33,50,40,38,60,5,37,67,81
,77,41,34,17,35,9,48,20,7,15,3

Amaç Fonksiyonu Sonucu : 7722 Km.

Amaç Fonksiyonu Ortalaması : 7810 Km.

Amaç Fonksiyonu Standart Sapması : 224 Km.



Şekil 2.38: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).

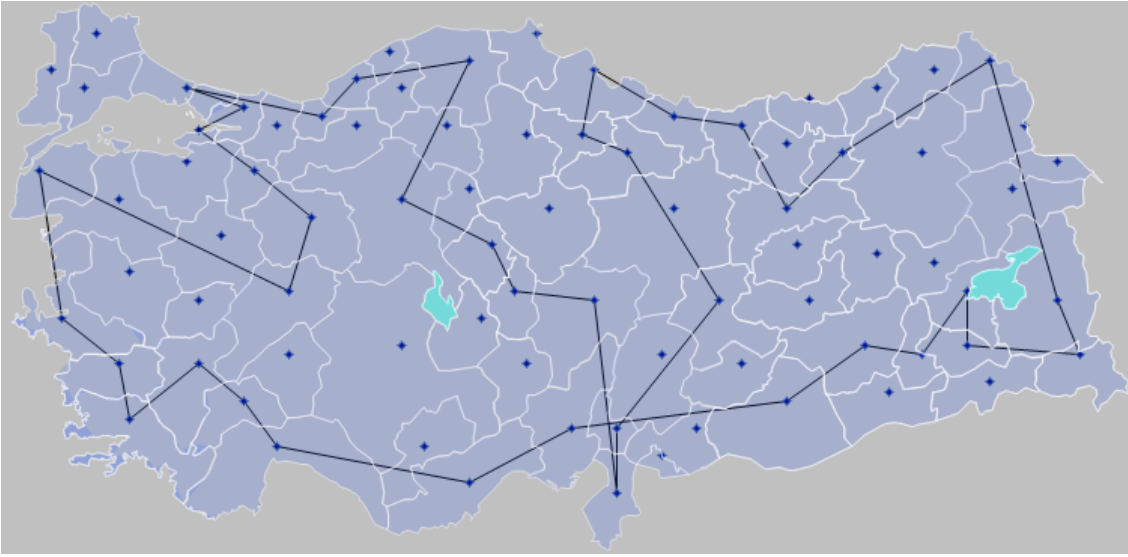
Çözüm Dizilimi (#3):

3,17,35,9,48,20,15,7,33,1,63,21,72,13,56,30,65,75,69,24,28,52,55,5,60,44,80,31,38,50,
40,6,37,67,81,34,41,77,11,26,3

Amaç Fonksiyonu Sonucu : 8058 Km.

Amaç Fonksiyonu Ortalaması : 7810 Km.

Amaç Fonksiyonu Standart Sapması : 224 Km.



Şekil 2.39: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).

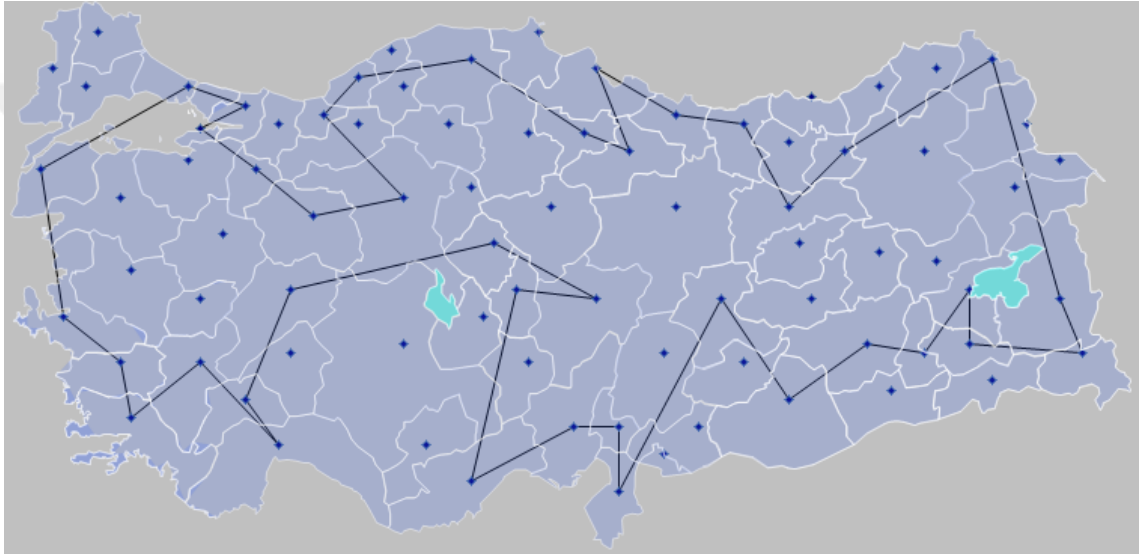
Çözüm Dizilimi (#4):

3,40,38,50,33,1,80,31,44,63,21,72,13,56,30,65,75,69,24,28,52,55,60,5,37,67,81,6,26,11
,77,41,34,17,35,9,48,20,7,15,3

Amaç Fonksiyonu Sonucu : 7670 Km.

Amaç Fonksiyonu Ortalaması : 7810 Km.

Amaç Fonksiyonu Standart Sapması : 224 Km.



Şekil 2.40: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).

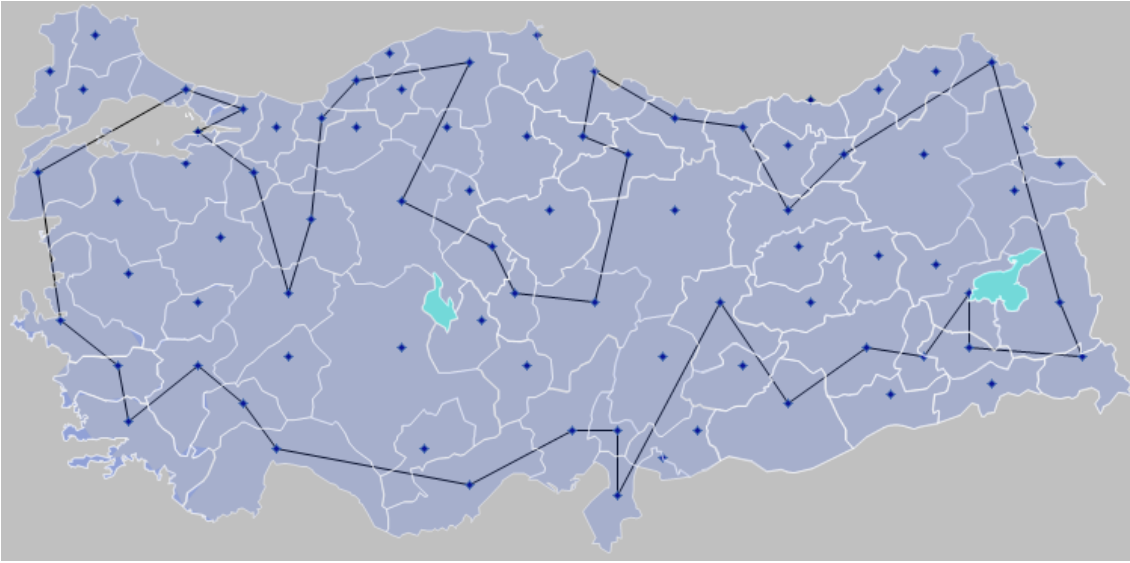
Çözüm Dizilimi (#5):

3,11,77,41,34,17,35,9,48,20,15,7,33,1,80,31,44,63,21,72,13,56,30,65,75,69,24,28,52,55
,5,60,38,50,40,6,37,67,81,26,3

Amaç Fonksiyonu Sonucu : 7563 Km.

Amaç Fonksiyonu Ortalaması : 7810 Km.

Amaç Fonksiyonu Standart Sapması : 224 Km.



Şekil 2.41: 40 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).

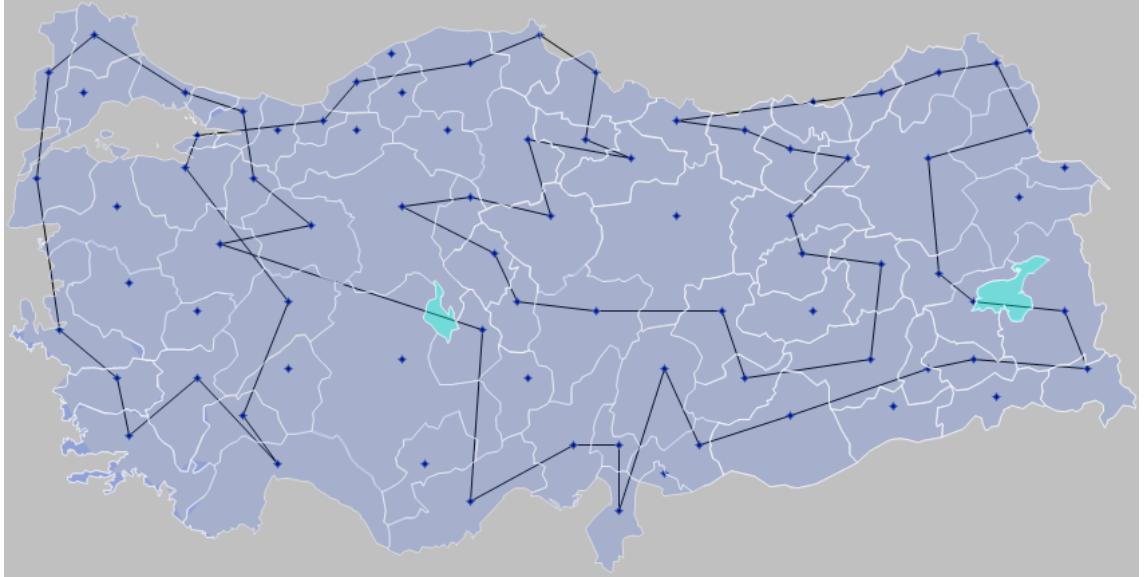
60 adet Rasgele İlin Seçimi Problemi ve Çözümü:**Çözüm Dizilimi (#1):**

3,15,7,20,48,9,35,17,22,39,34,41,11,26,43,68,33,1,80,31,46,27,63,72,56,30,65,13,49,25
,36,75,8,53,52,28,29,69,24,62,12,21,2,44,38,50,40,6,71,66,19,60,5,55,57,37,67,81,77,1
6,3

Amaç Fonksiyonu Sonucu : 9618 Km.

Amaç Fonksiyonu Ortalaması : 10187 Km.

Amaç Fonksiyonu Standart Sapması : 378 Km.



Şekil 2.42: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#1).

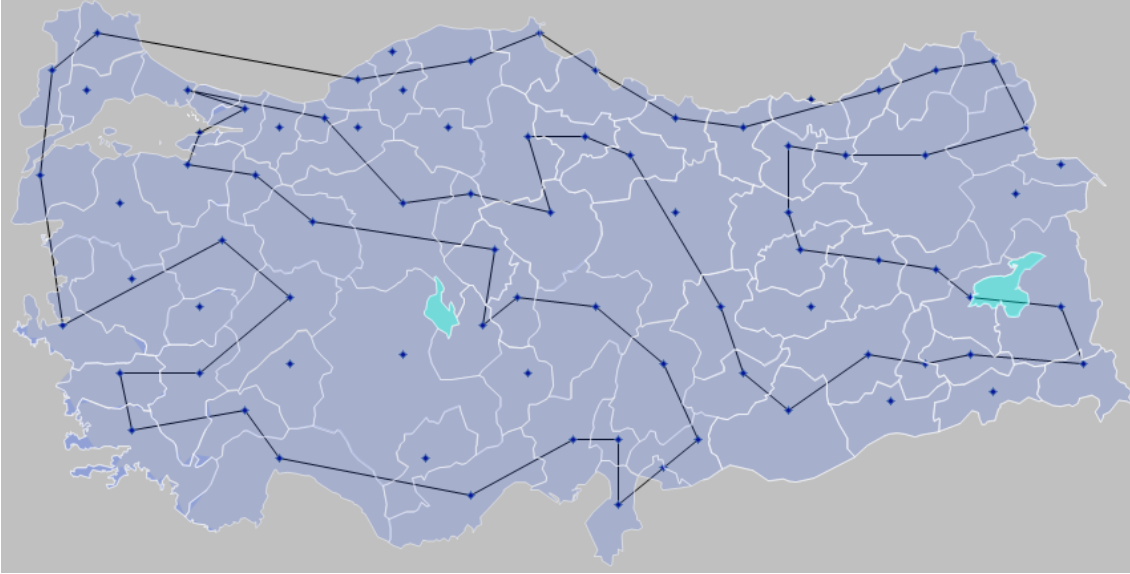
Çözüm Dizilimi (#2):

3,43,35,17,22,39,67,37,57,55,52,28,53,8,75,36,25,69,29,24,62,12,49,13,65,30,56,72,21,
63,2,44,60,5,19,66,71,6,81,34,41,77,16,11,26,40,68,50,38,46,27,31,80,1,33,7,15,48,9,2
0,3

Amaç Fonksiyonu Sonucu : 9864 Km.

Amaç Fonksiyonu Ortalaması : 9636 Km.

Amaç Fonksiyonu Standart Sapması : 193 Km.



Şekil 2.43: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#2).

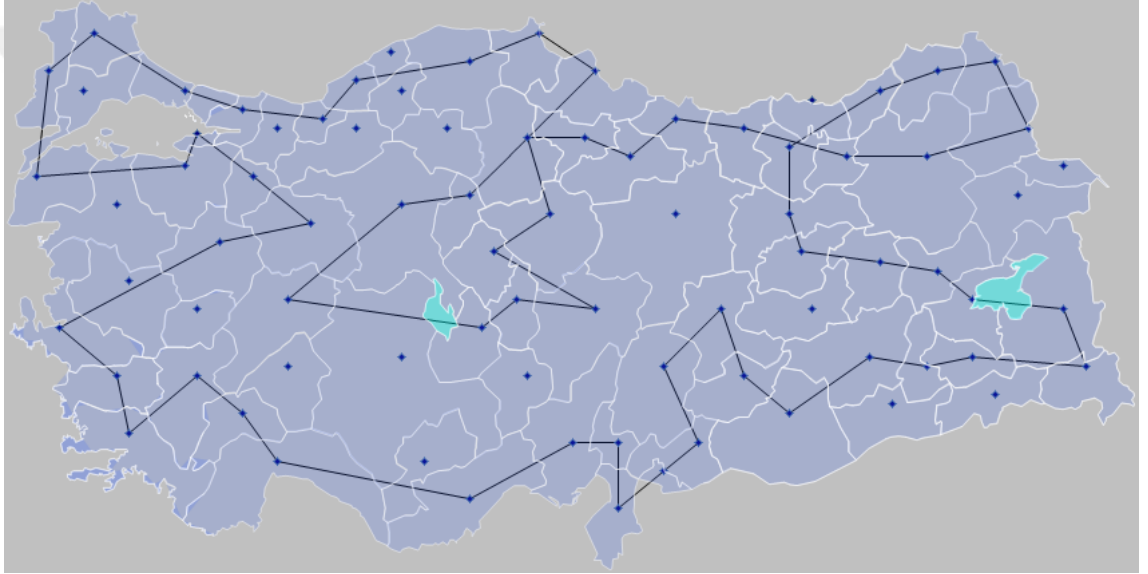
Çözüm Dizilimi (#3):

3,68,50,38,40,66,19,5,60,52,28,69,25,36,75,8,53,29,24,62,12,49,13,65,30,56,72,21,63,2,44,46,27,31,80,1,33,7,15,20,48,9,35,43,26,11,77,16,17,22,39,34,41,81,67,37,57,55,71,6,3

Amaç Fonksiyonu Sonucu : 9357 Km.

Amaç Fonksiyonu Ortalaması : 9636 Km.

Amaç Fonksiyonu Standart Sapması : 193 Km.



Şekil 2.44: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#3).

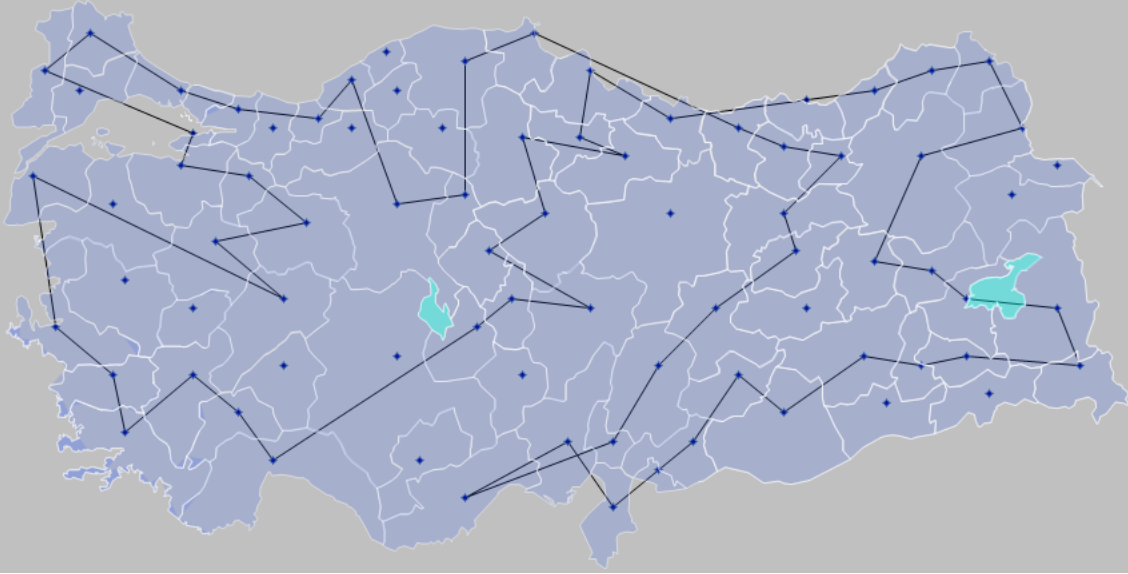
Çözüm Dizilimi (#4):

3,17,35,9,48,20,15,7,68,50,38,40,66,19,60,5,55,52,53,8,75,36,25,12,49,13,65,30,56,72,
21,63,2,27,31,1,33,80,46,44,62,24,69,29,28,57,37,71,6,67,81,41,34,39,22,77,16,11,26,4
3,3

Amaç Fonksiyonu Sonucu : 9759 Km.

Amaç Fonksiyonu Ortalaması : 9636 Km.

Amaç Fonksiyonu Standart Sapması : 193 Km.



Şekil 2.45: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#4).

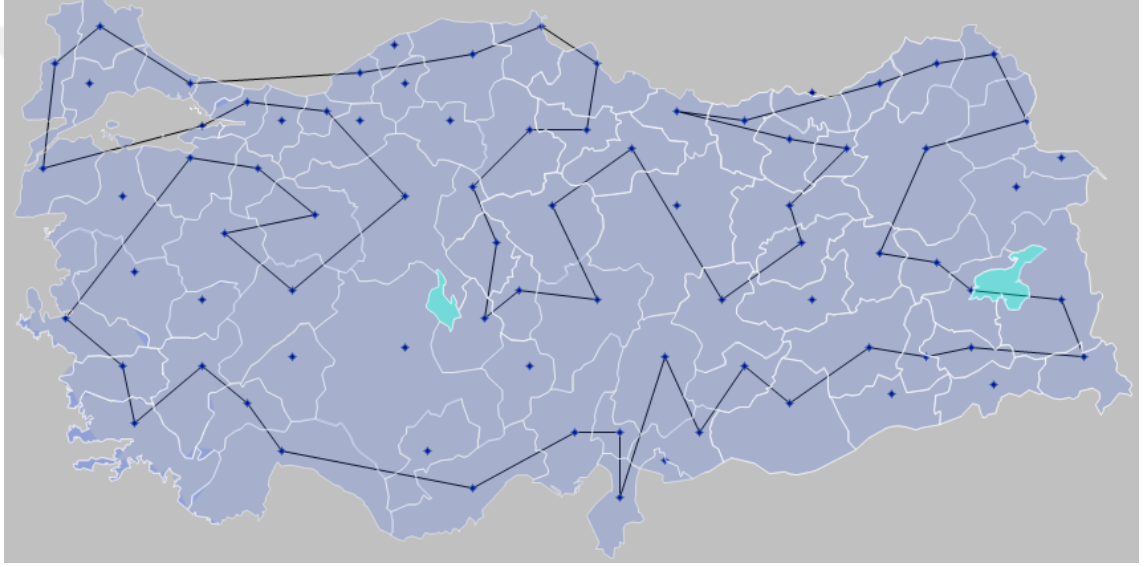
Çözüm Dizilimi (#5):

3,6,81,41,77,17,22,39,34,67,37,57,55,5,19,71,40,68,50,38,66,60,44,62,24,69,29,52,28,5
3,8,75,36,25,12,49,13,65,30,56,72,21,63,2,27,46,31,80,1,33,7,15,20,48,9,35,16,11,26,4
3,3

Amaç Fonksiyonu Sonucu : 9581 Km.

Amaç Fonksiyonu Ortalaması : 9636 Km.

Amaç Fonksiyonu Standart Sapması : 193 Km.



Şekil 2.46: 60 adet Rastgele İlin Uygulama (GA) ile Çözümü (#5).

81 adet İlin Seçimi Problemi ve Çözümü:

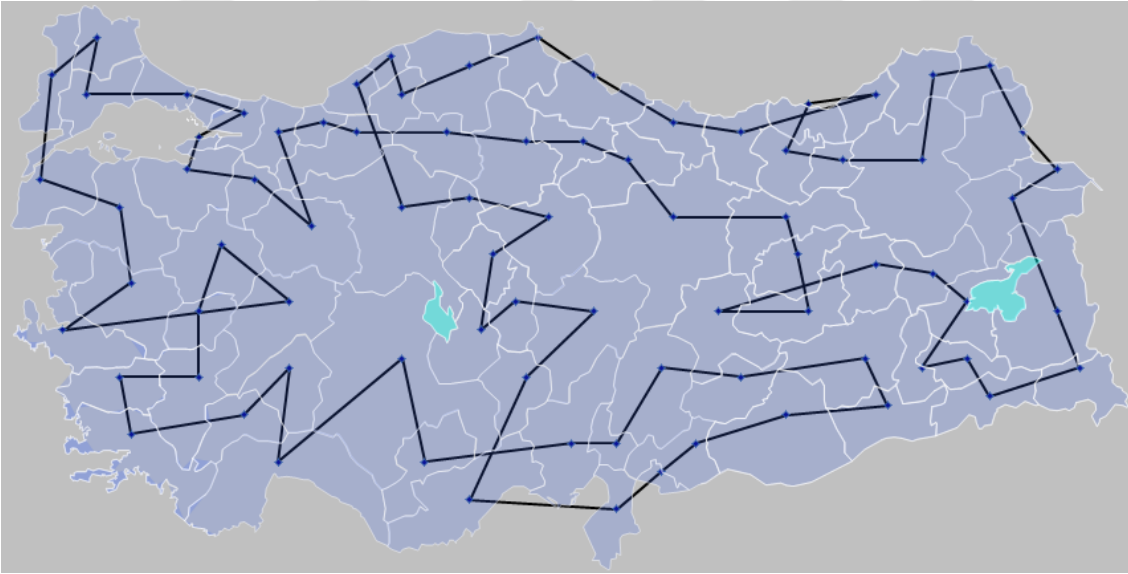
Çözüm Dizilimi (#1):

3,43,64,20,9,48,15,32,7,42,70,1,80,46,2,21,47,63,27,79,31,33,51,38,50,68,40,66,71,6,6
 7,74,78,37,57,55,52,28,53,61,29,69,25,8,75,36,76,4,65,30,73,56,72,13,49,12,44,23,62,2
 4,58,60,5,19,18,14,81,54,26,11,16,77,41,34,59,39,22,17,10,45,35,3

Amaç Fonksiyonu Sonucu : 11213 Km.

Amaç Fonksiyonu Ortalaması : 11251 Km.

Amaç Fonksiyonu Standart Sapması : 299 Km.



Şekil 2.47: 81 ilin Uygulama (GA) ile Çözümü (#1).

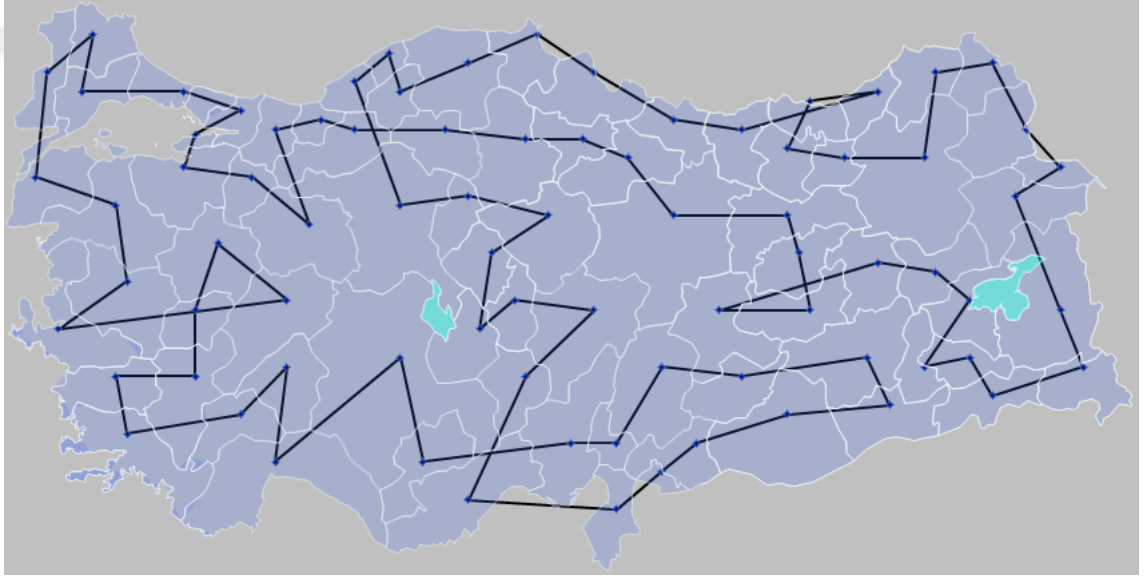
Çözüm Dizilimi (#2):

3,64,45,35,9,48,20,15,32,7,70,43,26,11,77,16,10,17,22,39,59,34,41,54,81,14,67,74,78,3
7,18,6,71,40,38,66,19,5,60,57,55,52,28,29,69,61,53,8,75,36,76,4,49,13,65,30,73,56,72,
21,47,63,27,79,31,1,33,80,46,2,12,25,24,62,23,44,58,50,51,68,42,3

Amaç Fonksiyonu Sonucu : 11412 Km.

Amaç Fonksiyonu Ortalaması : 11251 Km.

Amaç Fonksiyonu Standart Sapması : 299 Km.



Şekil 2.48: 81 ilin Uygulama (GA) ile Çözümü (#2).

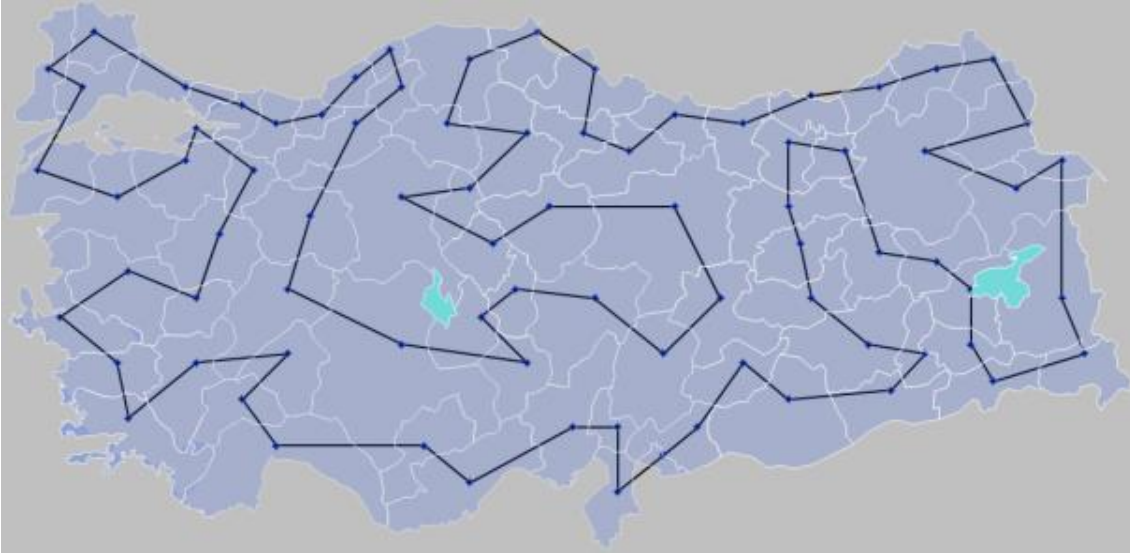
Çözüm Dizilimi (#3):

3,42,51,68,50,38,46,44,58,66,40,6,71,19,18,37,57,55,5,60,52,28,61,53,8,75,36,25,4,76,
65,30,73,56,13,49,12,69,29,24,62,23,21,72,47,63,2,27,79,31,80,1,33,70,7,15,32,20,48,9
,35,45,64,43,11,77,16,10,17,59,22,39,34,41,54,81,67,74,78,14,26,3

Amaç Fonksiyonu Sonucu : 11674 Km.

Amaç Fonksiyonu Ortalaması : 11251 Km.

Amaç Fonksiyonu Standart Sapması : 299 Km.



Şekil 2.49: 81 ilin Uygulama (GA) ile Çözümü (#3).

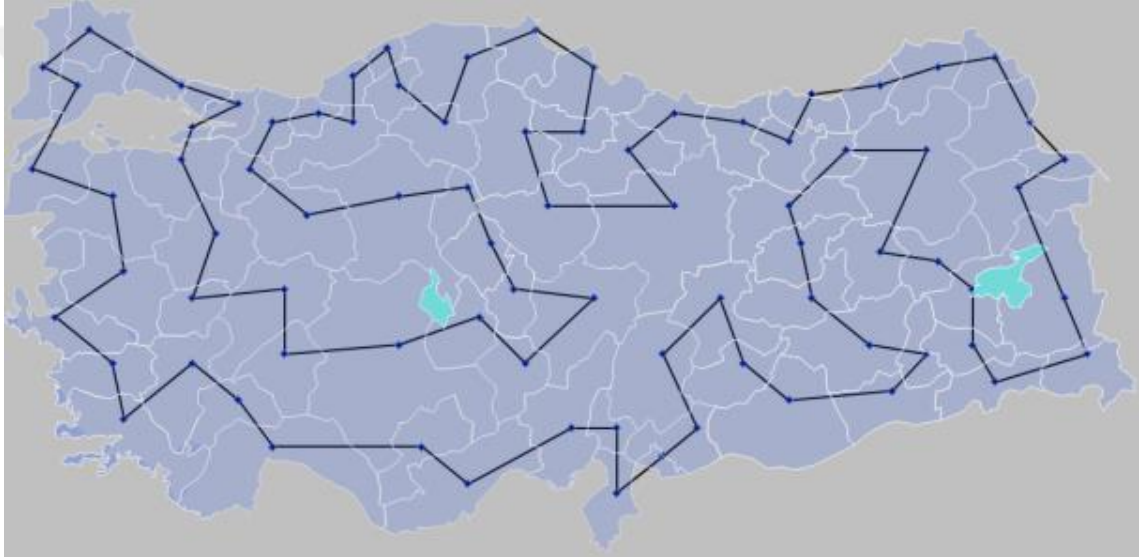
Çözüm Dizilimi (#4):

3,32,42,68,51,38,50,40,71,6,26,11,54,81,14,67,74,78,18,37,57,55,5,19,66,58,60,52,28,2
 9,61,53,8,75,36,76,4,65,30,73,56,13,49,12,25,69,24,62,23,21,72,47,63,2,44,46,27,79,31
 ,80,1,33,70,7,15,20,48,9,35,45,10,17,59,22,39,34,41,77,16,43,64,3

Amaç Fonksiyonu Sonucu : 10962 Km.

Amaç Fonksiyonu Ortalaması : 11251 Km.

Amaç Fonksiyonu Standart Sapması : 299 Km.



Şekil 2.50: 81 ilin Uygulama (GA) ile Çözümü (#4).

3. BULGULAR

Belirlenen 6 adet problemin çözümleri geliştirilen genetik algoritmalar programı, Excel – Solver (Evolutionary) genetik algoritmalar ve En Yakın Komşu Algoritması olmak üzere üç adet yöntem kullanılarak gerçekleştirilmiştir.

Bu çözüm yöntemlerine göre elde edilen bulgular şunlardır:

- Excel – Solver (Evolutionary) genetik algoritmalar programı ile maksimum 20 şehir (20X20 şehir) ile çözüm yapılabilmektedir. Bu değişken sayısı üzerine çıkıldığında program çözümü desteklememektedir.
- En Yakın Komşu Algoritması kullanılarak yapılan çözümde 10, 20, 30, 40, 60, 81 düğümlü problemler çözülmüştür fakat amaç fonksiyonu değerlerinin en iyi değerler olmadığı gözlemlenmiştir. Elde edilen mesafe ve süre değerleri Tablo 1 ' de karşılaştırmalı olarak verilmiştir.
- Genetik algoritmalar ile geliştirilen uygulamamızda, 10, 20, 30, 40, 60, 81 düğümlü problemler çözülmüştür. Her problem 5 defa çalıştırılmış ve sonuçlar Tablo 1 ' de mesafe ve süre olarak karşılaştırmalı olarak verilmiştir. Bu yöntem ile amaç fonksiyonu değerlerinin en iyi sonuçları ile karşılaştırılmıştır.
- Geliştirilen uygulamamızda uygunluk değeri iyileştikçe süre uzamaktadır. Bu da işlem sayısı arttıkça daha iyi sonuçlar alabileceğimizi göstermektedir. Fakat ilk oluşturulan popülasyon kalitesinin süre üzerinde önemli etkilerde bulunduğu gözlemlenmiştir.
- Geliştirilen uygulamamızda çözülen problemlerde yeterli popülasyon sayısından fazla sayıda seçim yapılması problem çözüm süresini uzatmakta ve işlem adımlarını artırdığı gözlemlenmiştir.

4.1. KARŞILAŞTIRMA

Genetik algoritmalar ile geliştirilen uygulama, Excel – Solver (Evolutionary) genetik algoritmalar ve En Yakın Komşu Algoritması olmak üzere 3 yöntem ile belirlenen 6

adet problem; 10, 20, 30, 40 ve 60 ilin rasgele seçilmesi ile 81 il için elde edilen sonuçlar karşılaştırmalı olarak Tablo 4.1' de verilmiştir.

Genetik algoritmalar kullanılarak yapılan çözümler her defasında farklı sonuçlar vereceği için her problem çözümünde, program 5 defa çalıştırılmış ve karşılaştırma için kullanılacak olan ortalama ve standart sapma değerleri tabloya yansıtılmıştır.



Tablo 4.1: Karşılaştırmalı Çözüm Sonuçları.

Problem	En Yakın Komşu		Excel Solver (GA)						GA Uygulama						Fark (GA)		%Fark (GA)			
	Mesafe	Zaman	Mesafe			Zaman			Mesafe			Zaman			Mesafe (km)	Zaman (sn)	Mesafe (km)	Zaman (sn)		
	Kilometre	Saniye	Kilometre	Ortalama	Standart Sapma	Saniye	Ortalama	Standart Sapma	Kilometre	Ortalama	Standart Sapma	Saniye	Ortalama	Standart Sapma						
10 il #1	3.698	0,5	3.618	3.659	37,2	65	64	2,3	3.618	3.645	37,2	4	3	0,8	0	61	0,00%	93,80%		
10 il #2			3.686			64			3.686			2			0	62	0,00%	96,88%		
10 il #3			3.686			65			3.618			3,5			68	62	1,84%	94,64%		
10 il #4			3.618			65			3.686			3			-68	62	-1,88%	95,35%		
10 il #5			3.686			60			3.618			4			68	56	1,84%	93,28%		
20 il #1	5.886	0,5	5.061	5.061	14,3	75	75	3,2	5.061	5.152	105,7	38	39	12,0	0	37	0,00%	49,47%		
20 il #2			5.049			76			5.310			46			-261	30,4	-5,17%	39,79%		
20 il #3			5.049			77			5.181			32			-132	45,2	-2,61%	58,55%		
20 il #4			5.061			69			5.158			23			-97	46,2	-1,92%	66,76%		
20 il #5			5.084			76			5.049			54			35	21,65	0,69%	28,62%		
30 il #1	7.686	0,5	N/A	N/A	N/A	N/A	N/A	N/A	6.754	6.743	56,5	518	383	131,9	N/A	N/A	N/A	N/A		
30 il #2									6.781			240								
30 il #3									6.810			245								
30 il #4									6.685			479								
30 il #5									6.685			435								
40 il #1	9.826	0,8	N/A	N/A	N/A	N/A	N/A	N/A	8.036	7.810	224	601	943	239	N/A	N/A	N/A	N/A		
40 il #2									7.722			924								
40 il #3									8.058			1.079								
40 il #4									7.670			1.240								
40 il #5									7.563			871								
60 il #1	10.060	1	N/A	N/A	N/A	N/A	N/A	N/A	9.618	9.636	193	2.885	2.866	466	N/A	N/A	N/A	N/A		
60 il #2									9.864			3.204								
60 il #3									9.357			2.976								
60 il #4									9.759			3.194								
60 il #5									9.581			2.069								
81 il #1	12.894	1,4	N/A	N/A	N/A	N/A	N/A	N/A	11.213	11.251	299	6.237	5.369	490	N/A	N/A	N/A	N/A		
81 il #2									11.412			5.230								
81 il #3									11.674			5.064								
81 il #4									10.962			5.108								
81 il #5									10.992			5.204								

Bu tabloya göre yapılan karşılaştırmalar;

- I. 10 adet rastgele ilin seçilmesi ile oluşturulan problem sonuçları incelendiğinde; En Yakın Komşu Algoritması kullanılarak yapılan çözüm en kısa sürede elde edilmesine rağmen en iyi amaç fonksiyonun bulunmasında başarılı olamamıştır. Genetik algoritmalar ile geliştirilen uygulamamız ve Excel – Solver (Evolutionary) genetik algoritmalar kullanılarak yapılan çözümler karşılaştırıldığında mesafe ve süre olarak en başarılı sonuçlar genetik algoritmalar ile geliştirilen programımız ile elde edilmektedir. Süre açısından da iki program arasında %97 üzerinde seyreden ciddi bir fark görülmektedir. Standart sapma sonuçları karşılaştırıldığında, her bir çözüm için, geliştirilen algoritma ile ortalamaya daha yakın değerler alınmakta ve daha kaliteli sonuçlar elde edilmektedir.
- II. 20 adet rastgele ilin seçilmesi ile oluşturulan problem sonuçları incelendiğinde; En Yakın Komşu Algoritması, genetik algoritmalar ile geliştirilen uygulamamız ve Excel – Solver (Evolutionary) genetik algoritmalar kullanılarak yapılan çözümler karşılaştırıldığında süre olarak en başarılı sonuç En Yakın Komşu Algoritması ile mesafe olarak en başarılı çözüm Excel – Solver (Evolutionary) genetik algoritmalar elde edilmiştir. En iyi süreyi En Yakın Komşu Algoritması vermesine rağmen en uzun mesafede bu algoritma ile elde edilmiştir. Genetik algoritmalar ile geliştirilen uygulamamız ve Excel – Solver (Evolutionary) genetik algoritmalar kullanılarak yapılan çözümleri süre açısından kıyaslandığımızda geliştirilen algoritma lehindeki fark dikkati çekmektedir.
- III. 30 üzerinde rastgele seçilen il ile oluşturulan problemlerde Excel – Solver (Evolutionary) genetik algoritmalar kullanılarak çözüm elde edilememiştir.
- IV. 30 ve üzeri il ile seçilerek oluşturulan problemlerde en iyi süre En Yakın Komşu Algoritması ile elde edilmesine rağmen en iyi amaç fonksiyonlarını Genetik Algoritmalar ile geliştirilen uygulamamız vermektedir. Bu sonuç genetik algoritmalara duyulan ihtiyacı ve önemi bir kez daha kanıtlamaktadır.

4. TARTIŞMA VE SONUÇ

Firmalar hızla gelişen ve değişen rekabet ortamında ayakta kalabilmek, rakiplerle rekabet edebilmek ve pazar payını genişletebilmek için işletmenin toplam maliyetleri içerisinde önemli bir paya sahip olan lojistik ve dağıtım maliyetlerini minimum seviyeye indirmelidir. Bu da uygun bir dağıtım planının oluşturulmasıyla gerçekleşir. Dağıtım faaliyetlerinin gerçekleştirilmesinde dikkat edilmesi gereken nokta müşteri gereksinimlerini tam olarak karşılayacak optimum dağıtım planının oluşturulmasıdır. Optimal araç yükleme ve rotalama bu planın oluşturulmasında en önemli konulardan biridir. Uygun kapasiteyle yüklenmiş araçların müşteri noktalarına en kısa mesafede ve en kısa zamanda ulaşması hem toplam dağıtım maliyetlerinin azalmasında hem servis kalitesinin artırılmasında hem de firmanın rekabet koşullarında daha avantajlı konuma geçmesine imkan sağlayacaktır.

Son yıllarda optimizasyon kavramı işletmelerde planlama, tasarım, çizelgeleme ve maliyetlendirme faaliyetlerinde oldukça önemli bir konuma gelmiştir, zira optimizasyon işlemi kaynakların verimli bir şekilde kullanılmasını, zaman tasarrufu ve ürün kalitesinin artırılmasını sağlamaktadır. Özellikle büyük boyutlu sistemlerde, etkinliklerin insan emeği ile optimize edilmesi zor olduğundan, bu tür işlemler algoritmalar yolu ile bilgisayar programlarıyla gerçekleştirilebilir. Metasezgisel algoritmalar günümüzde optimizasyon uygulamaları için en etkili sonuç veren yöntemlerden biridir.

Bu çalışmada, araç rotalama problemlerinin çözümünde metasezgisel bir algoritma olan genetik algoritmaların kullanımı büyük boyutlu problemlerde denenmiş, minimum yol amaç fonksiyonunun sağlanmasında karşılaştırılan diğer yöntemlere göre daha başarılı sonuçlar elde edildiği görülmüştür.

Geliştirilen algoritma programının akış şeması, işleme adımları belirlenerek ilgili kodlamalar Excel ve Visual Basic programlama dilinde yapılmıştır. Daha sonra çalışmanın uygulamaları Türkiye illeri üzerinde gerçekleştirilmiştir.

Bu tez çalışmasında, karşılaştırma yapılan diğer yöntemler ile çözülemeyen veya iyi sonuç elde edilemeyen büyük boyutlu gerçek hayat problemlerinde metasezgisel algoritmalarından olan genetik algoritmalara duyulan ihtiyaç kanıtlanmıştır. Ayrıca genetik algoritmaların, arama uzayının büyük ve karmaşık olduğu, mevcut bilgi ile sınırlı arama uzayında çözümün zor olduğu, problemin belirli bir matematiksel modelle ifade edilemediği, geleneksel en iyileme yöntemlerinden istenen sonucun alınmadığı alanlarda etkili ve kullanışlı olduğu da kanıtlanmıştır.



KAYNAKLAR

- Akođlu, K., 2006, *Konteynür Limanının Depolama Sahasının Genetik Algoritma ile Optimizasyonu*, Yüksek Lisans Tezi, Celal Bayar Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği, Manisa.
- Alba, E., Dorronsoro, B., 2005, *The Exploration/Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms.*, Transactions on Evolutionary Computation, 9, IEEE, 26-142.
- 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, 1561-1584.
- Alkan, A., 2003, *Lojistik Yönetiminde Filo Yönetim Sistemleri*, Yıldız Teknik Üniversitesi, İstanbul.
- Araque, J.R., Kudva, G., Morin, T.L., Pekny, J.F., 1994, *A branch-and-cut algorithm for vehicle routing problems*, Annals of Operations Research, 50, 37-59.
- Ballou, R., 1999, *Business Logistics Management, Planning, Organizing and Controlling the Supply Chain*, Prentice Hall, USA.
- Baker B.M, Ayechev M.A., 2003, *A Genetic Algorithm for the Vehicle Routing Problem*, Computers and Operations Research, Vol. 30, 787-800.
- Balakrishnan P. V., Jacob, S.V., 1996 *Genetic Algorithms for Product Design*, Management Science, 42 (8), 1105-1117.
- Balas, E., Toth, P., 1983, *Branch and bound methods for the traveling salesman problem*, Management Science Research Report, MSRR, 488.
- Barnhart, C., Laporte, G., 2007, *Transportation: Handbooks in Operations Research and Management Science*, North-Holland, Amsterdam.
- Barzan, Ş., 2005, *Araç Rotalarının En Kısa Yol Algoritmaları Kullanılarak Belirlenmesi ve .Net Ortamında Simülasyonu*, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli.
- Başkaya Z., Avcı Öztürk, B.,(2005). Tamsayılı programlamada dal kesme yöntemi ve bir ekmek fabrikasında oluşturulan araç rotalama problemine uygulanması. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, Cilt XXIV, Sayı 1, 101-114.
- Berger J., Barkaoui M., Braysy O., 2003, *A Route-Directed Hybrid Genetic Approach For Vehicle Routing Problem With Time Windows*, INFOR 2003, Vol. 41, 179-94.
- 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, 2037–2053.

- 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, 578-594.
- Biroğul, S., 2005, *Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Blum, C., Roli, A., (2003), *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*, ACM Computing Surveys, 35, 3, 268-308.
- Chatterjee S., Carrera C., Lynch L.A., 1996, *Genetic Algorithms and Traveling Salesman Problems*, European Journal of Operational Research, Vol. 93, 490–510.
- Chauhan, C., Gupta, R., Pathak, K., 2012, *Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach*, International Journal of Computer Applications, Volume 52, ISBN: 0975–8887, 4.
- Cheng, R., Gen, M., Tsujimura, Y., 1996, *A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms-I, Representation*, Computers and Industrial Engineering, 4, ISBN: 983-997, 30.
- Chu P.C. ve Beasley J.E. , 1997, *A Genetic Algorithm for The Generalised Assignment Problem*, Science Direct, PII: S0305-0548(96)00032-9, Computers Ops Res. Vol. 24, No. 1, 17-23.
- Cleveland G.A., Smith S.F., 1989, *Using genetic algorithms to schedule flow shop releases*, J.D. Schaffer (Ed.), Proceedings of 3rd International Conference on Genetic Algorithms, Kaufmann, Los Altos, 160–169.
- Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F., 2002, *A Guide To Vehicle Routing Heuristics*, Journal of the Operational Research Society, 53, 512-522.
- Crainic, Gabriel Teodor and Laporte, Gilbert 1998, *Fleet Management and Logistics*, Kluwer, USA.
- Çetin, N., 2002, *Genetik Algoritma*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Demircioğlu, M., 2009, *Araç Rotalama Probleminin Sezgisel Bir Yaklaşım İle Çözülmesi Üzerine Bir Uygulama*, Doktora Tezi, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı.
- Duman, C., 2007, *Genetik Algoritma ile Tesis Yerleşimi Tasarımı ve Bir Uygulama*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Dursun, P., 2009, *Zaman Pencerli Araç Rotalama Probleminin Genetik Algoritma ile Modellenmesi*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Düzakın, E., Demircioğlu, M., 2009, *Araç Rotalama Problemleri ve Çözüm Yöntemleri*, Çukurova Üniversitesi İİBF Dergisi, Cilt:13, Sayı:1, 68-87.
- Drezner, Z., 2003, *A Hybrid Genetic Algorithm for The Quadratic Assignment Problem*, INFORMS, 15.

- Emel, G. G., Taşkın, Ç., 2002, Genetik Algoritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (1):129-152.
- Emel, G. G., Taşkın, Ç., 2005, Araç Rotalama Problemlerinin İki Aşamalı Çözümünde Genetik Algoritma Kullanımı, *Gazi Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi*, 7/1.
- Erdal, M., 2007, *Kısıtlı Kaynak Koşullarında Yapı Projelerinin Genetik Algoritma ile Programlanması*, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yapı Eğitimi Bölümü.
- Erel, R., 1995, *Taşıt Rotalaması ve Çizelgelemesi: Otobüsle Kentlerarası Yolcu Taşımacılığı İçin Bir Model*, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü İnşaat Mühendisliği Bölümü, İstanbul.
- Erol V., 2006, *Araç Rotalama Problemleri İçin Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Sistem Mühendisliği, Sabancı Kütüphanesi, İstanbul.
- Eryavuz, M., Gencer C., 2001, *Araç Rotalama Problemine Ait Bir Uygulama*, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi C.6, S.1.
- Fisher M.L., Jaikumar R., 1981, *A Generalized Assignment Heuristic for Vehicle Routing*, Vehicle Routing Problem, University of Colorado at Boulder, Leeds Scholl of Business, Fall 2004.
- Glynn P., Robinson S., 1997, *Springer Series in Operations Research*, Springer, New York.
- Goldberg D., Lingle R., 1985, Alleles, Loci and The Traveling Salesman Problem, *Proceedings of the 9th International Conference on Genetic Algorithms*.
- Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Publishing Company, Reading, Massachusetts, USA.
- Golden B.L., Wasil E.A., Kelly J.P, Chao I-M., *The Impact Of Metaheuristics On Solving The Vehicle Routing Problem: Algorithms, Problem Sets and Computational Results*. In: Crainic TG, Laporte G, editors., Fleet management and logistics, Boston: Kluwer Academic Publishers, pp. 33-56, 1998.
- Grefenstette, J.J., 1992, *Optimization of control parameters for genetic algorithms*, in Genetic Algorithms, pp.5-11, Eds. Buckles, B.P.& Petry, F.E., IEEE Computer Society Pres, California.
- Haghani A., Jung S., 2005, *A dynamic vehicle routing problem with time-dependent travel times*, Computers & Operations Research, Vol. 32, Issue 11, pp. 2959-2986.
- 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.
- 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.
- Holland, J. H. 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, U Michigan Press.

- Ichoua S., Gendreau M., Potvin J.Y., 2000, *Diversion Issues in Real-Time Vehicle Dispatching*, Transportation Science, Vol. 34, pp. 426-438.
- Jaszkiewicz, A., Kominek, P., 2003, *Genetic local search with distance preserving recombination operator for a vehicle routing problem*, Meta-heuristics in combinatorial optimization, 151 (2): 352-364.
- Jaszkiewicz, A., Ishibuchi, H., Zhang, Q., 2012, *Multiobjective Memetic Algorithms*, (Ed) F. Neri, C. Cotta, P. Moscato, Handbook of Memetic Algorithms, s. 201-217.
- Jung S., Haghani A., 2001, A Genetic Algorithm for the Time Dependent Vehicle Routing Problem, *Journal of Transportation Research Board*, Vol. 1771, pp. 161–171.
- Karahan, A., 2003, *Tedarik Zinciri Yönetiminde Dağıtım Faaliyetlerinin Optimize Edilmesine Yönelik Bir Model Tasarımı*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalı, İstanbul.
- Karacı, A., 2002, *Genetik Algoritmalarda Iraksama ve Yerel Çözümde Kalma Problemlerinin Giderilmesi*, Doktora Tezi, Fırat Üniversitesi, Mimarlık Anabilim Dalı, İzmir.
- Kılıç S., 2008, *Bulanık Karar Ortamında Karınca Kolonisi Optimizasyonu Yöntemiyle Araç Rotalama*, Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Koç Ç., 2012, *Zaman Bağımlı Araç Rotalama Problemi*, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.
- Kulluk, S., 2003, *Tesis Yerleşimi Problemlerinde Genetik Algoritmalar ve Bir Genetik Algoritma Uygulaması*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Kumar, S.N., 2012, Panneerselvam, R., *A Survey on the Vehicle Routing Problem and Its Variants*. *Intelligent Information Management*, 4, 66-74.
- Laporte, G., Gendreau, M., Potvin, J. Y., Semet, F., 2000, *Classical And Modern Heuristics For The Vehicle Routing Problem*, *International Transactions in Operation Research*, 7, 285-300.
- Liepins, G. E., Potter, W. D., Tonn, B. E., Hillard, M. R., Goeltz, R.T. Purucker, S. L., 1990, *Diagnosis, Parsimony and Genetic Algorithms*, IEA/AIE, 1:1-8.
- 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 Application*, Vol. 36, Issue 2, Part 1, pp. 1505-1512.
- Louis, S.J., Li, G. 2000, *Case injected genetic algorithms for travelling salesman problems*, *Information Sciences*, 122,201-225.
- Malandraki, C., 1989, *Formulations, solution algorithms and computations experiments*, Time dependent vehicle routing problem, Northwestern University, USA.
- Malandraki, C., Daskin, M.S., 1992, *Formulations, properties and heuristic algorithms*, Time dependent vehicle routing problems, *Transportation Science*.

- Mester, D., Braysy, O., 2007, *Active-guided evolution strategies for large-scale capacitated vehicle routing problems*, Computers & operations Research, 34 (10): 2964-2975.
- Nabiyev, V. V. ,2003, *Yapay Zeka – Problemler Yöntemler Algoritmalar*, Seçkin Yayınevi, Ankara.
- Nazif, H., Lee. L.S., 2012, *Optimized crossover genetic algorithm for capacitated vehicle routing problem*, Applied Mathematical Modeling, 36, 2110–2117.
- Pan, L., 2015, *Cutting Plane Method*, The Chinese University of Hong Kong, Operations Research and Logistics Jan. 20.
- Pierre D. M., Zakaria N., 2016, *Stochastic partially optimized cyclic shift crossover formulti-objective genetic algorithms for the vehicle routing problemwith time-windows*, High Performance Computing Center, Universiti Teknologi PETRONAS, 32610 Tronoh, Perak, Malaysia.
- Potvin J. Y., Xu, Y., Benyahia, I., 2004, *Vehicle Routing and scheduling with dynamic travel times*, Computers & Operations Research, 6, 557-570.
- Potvin J.Y., Duhamel C., Guertin F., 1996, *A Genetic Algorithm for Vehicle Routing with Backhauling*, Applied Intelligence 6, 345-355, Kluwer Academic Publisher, Manufactured in Netherlands.
- Preux, P., Talbi, E. G., 1999, *Towards Hybrid Evolutionary Algorithms*, International Transactions in Operation Research, 6, 557-570.
- Razali, N, M., 2015, *An Efficient Genetic Algorithm for Large Scale Vehicle Routing Problem Subject to Precedence Constraints*, Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600, Pekan, Pahang, Malaysia.
- Ropke, S., 2005, *Heuristic and exact algorithms for vehicle routing problems*.
- Solomon, M.M. 1987, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research 35:254–265.
- Şahin, Y., Eroğlu, A., 2014, *Kapasite Kısıtlı Araç Problemi için Metasezgisel Yöntemler*, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Y.2014, C.19, S.4, s.337-255.
- Şeker, Ş., 2007, *Araç Rotalama Problemleri ve Zaman Pencere Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Şen T., Cömert, S.E., Yazgan H.R., 2015, *Kapasite kısıtlı araç rotalama probleminin çözümü için yeni bir algoritma geliştirilmesi: bir süpermarket zincirinde uygulanması*, Sakarya Üniversitesi Fen Bilimleri Dergisi, SAÜ Fen Bil Der 19. Cilt, 1. Sayı, s. 83-88, 2015 85.
- Şen, Z., 2004, *Genetik Algoritmalar ve En İyileme Yöntemleri*, Su Vakfı Yayınları, İstanbul.
- Tan K.C., Lee L.H., Ou K., *Hybrid Genetic Algorithm In Solving Vehicle Routing Problems With Time Window Constraints*, Asia-Pacific Journal of Operational Research, Vol.18, pp.121-130, 2001.

- Thangiah S.R., Petrovic P., 1998, *Introduction To Genetic Heuristics And Vehicle Routing Problems With Complex Constraints*, Advanced in computational and stochastic optimization, logic programming, and heuristic search, In: Woodruff D.L., editor, Boston: Kluwer Academic Publishers, pp. 253-286.
- Tarantilis, C. D., Ioannou, G., Prastacos, G., (2004), *Advanced Vehicle Routing Algorithms For Complex Operations Management Problems*, Journal of Food Engineering, Elsevier Ltd.
- Taşkın, Ç., 2003, *Araç Rotalama Problemlerine Genetik Algoritma Yaklaşımı ve Bir Uygulama*, Yüksek Lisans Tezi, Uludağ Üniversitesi Sosyal Bilimler Enstitüsü Yayınlanmamış Bursa.
- Toth P., Vigo D., 2002, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia.
- Toth P., Vigo D., 2002a, *An overview of vehicle routing problems*, The Vehicle Routing Problem, (Ed) Toth, P., Vigo, D., SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, pp. 1–26.
- Toth, P., Vigo, D., 2002b, *The Vehicle Routing Problem*, windows and transshipment, Inform, vol. 44, no. 217-227, ISSN:0315-5986.
- Tüfekçier, H., 2008, *İki Amaçlı Açık Araç Rotalama Problemi İçin Bir Çözüm Yaklaşımı*, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi.
- Wang, C.H. Lu, J.Z., 2009, *A hybrid genetic algorithm that optimizes capacitated vehicle routing problems*, J Expert Syst. Appl., 36 (2): 2921-2936.
- 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.
- Wang, W., Wu, B., Zhao, Y., Feng, D., 2006, *Particle Swarm Optimization for Open Vehicle Routing Problem*, In Computational Intelligence, Vol. MMMCXIV, pp. 999-1007.

EKLER

EK 1. Algoritma Programının Kodları.

```
Option Explicit
Option Private Module
Public Type Sehir
    Plaka As Integer
    X As Integer
    Y As Integer
End Type
```

```
Public Type Kromozom
    Rota() As Sehir
    Mesafe As Long
End Type
```

```
Public Type Populasyon
    Sehirler() As Kromozom
    Sira As Integer
End Type
```

```
Public Depo_Sehir As Sehir
Public SehirSayisi As Integer
Public Populasyon_Sayisi As Integer
Public IslemSayisi As Integer
Public Crossover_Sayisi As Integer
Public Mutasyon_Sayisi As Integer
Public Route As Variant
Public Depo As Integer
Public MaxSameResultIteration As Integer
Public MustMutateFailedCrossovers As Boolean
Public MustDoCrossovers As Boolean
Public mutateDuplicates_ As Boolean
Public Population() As Populasyon
Public EnYiKromozom As Kromozom
Public ArrayItemRemoved As Boolean
Public MaxYenidenCount As Integer
Public Sub Degiskenler()
    SehirSayisi = Int(Sheets("Hesaplama").Range("E1").Text)
    Populasyon_Sayisi = Sheets("Hesaplama").Range("E2").value
    Depo_Sehir.Plaka = Int(Sheets("Tanımlamalar").Cells(SehirSayisi + 2, 8).Text)
```

```

Depo_Sehir.X =
    Application.WorksheetFunction.Index(Sheets("Tanımlamalar").Range("C2:D82")
    , Depo_Sehir.Plaka, 1)
Depo_Sehir.Y =
    Application.WorksheetFunction.Index(Sheets("Tanımlamalar").Range("C2:D82")
    , Depo_Sehir.Plaka, 2)
IslemSayisi = CInt(Sheets("Hesaplama").Range("E7").Text)
Crossover_Sayisi = CInt(Sheets("Hesaplama").Range("G10").Text)
Mutasyon_Sayisi = CInt(Sheets("Hesaplama").Range("G11").Text)
MaxSameResultIteration = Int(IslemSayisi * 0.2)
MaxYenidenCount = 15
MustMutateFailedCrossovers = True
MustDoCrossovers = True
mutateDuplicates_ = True
End Sub

```

```

Public Sub IlkPopulasyon()
Dim Don As Integer
Dim i As Integer
Dim Sutun As Integer
Dim Populasyon() As Populasyon
Dim Yazdim As Boolean
Dim Ziyaret_Edilecek_Sehirler() As Sehir
Degiskenler
    Ziyaret_Edilecek_Sehirler() = Sehirleri_Olustur
    Populasyon() = Rota_Calis(Depo_Sehir, Ziyaret_Edilecek_Sehirler,
        Populasyon_Sayisi)
    Population() = Populasyon()
    Populasyon() = Reproduce(Populasyon())
    Population() = Populasyon()
    Yazdim = Yaz(Population())
    Yazdim = Harita_Isle(0, Population(0).Sehirler(0).Rota(), True)
End Sub

```

```

Public Sub IslemYap()
IlkPopulasyon
Dim t As Long
t = Timer
    Dim Yazdim As Boolean
    Dim PercentComplete As Double
    Dim Don As Integer
    Dim oldBestSolution() As Kromozom
    Dim newBestSolution() As Kromozom
    Dim SameResult As Integer
    Dim BestPop() As Populasyon
    ReDim BestPop(1)
    Sheets("Hesaplama").Range("P4").value = 1
    Sheets("Hesaplama").Range("D29").value = "İşlem Yapılıyor Lütfen Bekleyiniz."
    For Don = 0 To IslemSayisi

```

```

Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual
Application.EnableEvents = False
ActiveSheet.DisplayPageBreaks = False
PercentComplete = ((Don + 1) / IslemSayisi)
oldBestSolution() = Population(0).Sehirler()
Population() = Reproduce(Population())
If mutateDuplicates_ = True Then
    Population() = MutateDuplicates(Population())
End If
newBestSolution = Population(0).Sehirler()
If oldBestSolution(0).Mesafe < newBestSolution(0).Mesafe Then
    Population(0).Sehirler() = oldBestSolution()
End If
If oldBestSolution(0).Mesafe = newBestSolution(0).Mesafe Then
    SameResult = SameResult + 1
Else
    SameResult = 0
End If
BestPop(0) = Population(0)
If SameResult >= MaxSameResultIteration Then
    GoTo Bitir
End If
Yazdim = Yaz(BestPop())
Application.ScreenUpdating = True
Application.StatusBar = Don & "/" & IslemSayisi & " " &
    Format(PercentComplete, "0%") & " Tamamlandı" & " " & SameResult & "-" &
    MaxSameResultIteration & " En Kısa Mesafe: " &
    Population(0).Sehirler(0).Mesafe
DoEvents
Next Don
Bitir:
Application.ScreenUpdating = True
Application.EnableEvents = True
ActiveSheet.DisplayPageBreaks = False
Application.Calculation = xlCalculationAutomatic
Yazdim = Yaz(Population())
Application.StatusBar = "Toplam Süre " & Round(Timer - t, 2) & " saniye"
Sheets("Hesaplama").Range("P4").value = 0
Sheets("Hesaplama").Range("D29").value = ""
Yazdim = Harita_Isle(0, Population(0).Sehirler(0).Rota(), True)
End Sub
Public Function Harita_Isle(Arac_No As Integer, Sehirler() As Sehir, İlk As Boolean)
    As Boolean
Dim Rotalar As Variant
Dim Sil As Integer
Dim Satir As Integer
Rotalar = Array(3, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53)
If İlk Then

```

```

For Sil = 1 To UBound(Rotalar)
    Sheets("Harita").Range(Sheets("Harita").Cells(1, Int(Rotalar(Sil))).Address,
        Sheets("Harita").Cells(100, Int(Rotalar(Sil))).Address).ClearContents
Next Sil
End If
If Arac_No = 0 Then
    Arac_No = 1
End If
    Sheets("Harita").Cells(1, Int(Rotalar(Arac_No))).value = Depo_Sehir.Plaka
For Satir = 2 To UBound(Sehirler())
    Sheets("Harita").Cells(Satir, Int(Rotalar(Arac_No))).value = Sehirler(Satir -
        1).Plaka
Next
    Sheets("Harita").Cells(Satir, Int(Rotalar(Arac_No))).value = Depo_Sehir.Plaka
Sheets("Harita").ChartObjects("Chart 1").Activate
ActiveChart.SeriesCollection(2).ChartType = xlXYScatter
ActiveChart.SeriesCollection(2).MarkerStyle = xlMarkerStyleNone
ActiveChart.SeriesCollection(Arac_No + 2).XValues = "=Harita!" &
    Sheets("Harita").Cells(1, Int(Rotalar(Arac_No)) + 1).Address & ":" &
    Sheets("Harita").Cells(UBound(Sehirler) + 1, Int(Rotalar(Arac_No)) + 1).Address
ActiveChart.SeriesCollection(Arac_No + 2).Values = "=Harita!" &
    Sheets("Harita").Cells(1, Int(Rotalar(Arac_No)) + 2).Address & ":" &
    Sheets("Harita").Cells(UBound(Sehirler) + 1, Int(Rotalar(Arac_No)) + 2).Address
End Function
Public Sub AB()
Dim Yazdim As Boolean
Yazdim = Yaz(Population())
End Sub
Public Function Yaz(Pop() As Populasyon) As Boolean
Dim i As Integer
Dim Kromozom As String
Dim Sutun As Integer
    Sheets("Hesaplama").Range("A:B").ClearContents
For i = 0 To UBound(Pop()) - 1
    Kromozom = Depo_Sehir.Plaka
    For Sutun = 1 To SehirSayisi
        Kromozom = Kromozom & "," & Pop(i).Sehirler(0).Rota(Sutun).Plaka
    Next Sutun
    Sheets("Hesaplama").Cells(i + 1, 1).value = Kromozom
    Sheets("Hesaplama").Cells(i + 1, 2).value = Pop(i).Sehirler(0).Mesafe
Next i
    Yaz = True
End Function

Public Function Sehirleri_Olustur() As Sehir()
Dim SOresult() As Sehir
Dim Satir As Integer
Dim DepoLokasyon As Integer
ReDim SOresult(SehirSayisi)

```

```

For Satir = 2 To SehirSayisi + 1
    If Int(Sheets("Tanımlamalar").Cells(Satir, 8).Text) = Depo_Sehir.Plaka Then
        DepoLokasyon = Satir - 1
    End If
    SOresult(Satir - 1).Plaka = Int(Sheets("Tanımlamalar").Cells(Satir, 8).Text)
    SOresult(Satir - 1).X =
        Application.WorksheetFunction.Index(Sheets("Tanımlamalar").Range("C2:D82")
        , SOresult(Satir - 1).Plaka, 1)
    SOresult(Satir - 1).Y =
        Application.WorksheetFunction.Index(Sheets("Tanımlamalar").Range("C2:D82")
        , SOresult(Satir - 1).Plaka, 2)
Next Satir
If DepoLokasyon <> SehirSayisi Then
    Dim EskiSehir As Sehir
    EskiSehir = SOresult(SehirSayisi)
    SOresult(DepoLokasyon) = EskiSehir
    SOresult(SehirSayisi) = Depo_Sehir
End If
Sehirleri_Olustur = SOresult()
End Function

Public Function Rota_Calis(Depo As Sehir, Sehirler() As Sehir, PopulasyonSayisi As
    Integer) As Populasyon()
    Dim RCresult() As Populasyon
    ReDim RCresult(PopulasyonSayisi)
    Dim i As Integer
    For i = 0 To PopulasyonSayisi - 1
        RCresult(i).Sehirler() = Rastgele_Rota_Olustur(Sehirler())
        RCresult(i).Sira = i
    Next i
    RCresult() = BubbleSort(RCresult)
    EnYiKromozom.Rota() = RCresult(0).Sehirler(0).Rota()
    EnYiKromozom.Mesafe = RCresult(0).Sehirler(0).Mesafe
    Rota_Calis = RCresult()
End Function

Public Function Rastgele_Rota_Olustur(Sehirler() As Sehir) As Kromozom()
    Dim value As Integer
    Dim IslemGorenKromozom() As Sehir
    Dim RROresult() As Kromozom
    Dim i As Integer
    ReDim RROresult(1)
    IslemGorenKromozom() = Sehirler()
    For i = UBound(Sehirler()) - 1 To 1 Step -1
        value = Rastgele_Rakam(i + 1)
        If (value <> i And Sehirler(i).Plaka <> Depo_Sehir.Plaka And
            Sehirler(value).Plaka <> Depo_Sehir.Plaka) Then
            IslemGorenKromozom() = Swap_Sehir(IslemGorenKromozom(), i, value)
        End If
    Next i
End Function

```

```

Next i
RROresult(0).Rota() = IslemGorenKromozom()
RROresult(0).Mesafe = Mesafe_Hesapla(IslemGorenKromozom())
Rastgele_Rota_Olustur = RROresult()
End Function

```

```

Public Function Swap_Sehir(Sehirler() As Sehir, index1 As Integer, index2 As Integer)
    As Sehir()
    Dim Lokasyon1 As Sehir
    Lokasyon1 = Sehirler(index1)
    Dim Lokasyon2 As Sehir
    Lokasyon2 = Sehirler(index2)
    If Lokasyon1.Plaka = Depo_Sehir.Plaka Or Lokasyon2.Plaka = Depo_Sehir.Plaka
        Then
            GoTo Bitir
    End If
    Sehirler(index1) = Lokasyon2
    Sehirler(index2) = Lokasyon1
Bitir:
    Swap_Sehir = Sehirler()
End Function

```

```

Public Function Move_Sehir(Sehirler() As Sehir, fromIndex As Integer, toIndex As
    Integer) As Sehir()
    Dim MSresult() As Sehir
    Dim temp As Sehir
    Dim i As Integer
    temp = Sehirler(fromIndex)
    If (fromIndex < toIndex) Then
        For i = fromIndex + 1 To toIndex Step 1
            Sehirler(i - 1) = Sehirler(i)
        Next i
    Else
        For i = fromIndex To toIndex Step -1
            Sehirler(i) = Sehirler(i - 1)
        Next i
    End If
    Sehirler(toIndex) = temp
Bitir:
    MSresult() = Sehirler()
    Move_Sehir = MSresult()
End Function

```

```

Public Function Reverse_Range(Sehirler() As Sehir, startIndex As Integer, endIndex As
    Integer) As Sehir()
    Dim RRresult() As Sehir
    Dim temp As Integer
    Dim temp1 As Sehir
    If (endIndex < startIndex) Then

```

```

    temp = endIndex
    endIndex = startIndex
    startIndex = temp
End If
Do While startIndex < endIndex
    temp1 = Sehirler(endIndex)
    Sehirler(endIndex) = Sehirler(startIndex)
    Sehirler(startIndex) = temp1
    startIndex = startIndex + 1
    endIndex = endIndex - 1
Loop
RRresult() = Sehirler()
Reverse_Range = RRresult()
End Function

Public Sub ABCD()
Dim i As Integer
Dim S As Integer
For i = 0 To 20
    S = Rastgele_Rakam(3)
Next i
End Sub
Public Function Rastgele_Rakam(Limit As Integer) As Integer
Dim R_Rresult As Integer
Randomize
R_Rresult = Int(Rnd * Limit) + 1

    Rastgele_Rakam = R_Rresult
End Function

Public Function Mesafe_Hesapla(Sehirler() As Sehir) As Long
Dim ToplamMesafe As Long
Dim Mesafe As String
Dim Sehir As Integer
If Sheets("Hesaplama").Range("E8").Text = "Karayolu" Then
    ToplamMesafe =
        WorksheetFunction.Index(Sheets("Mesafeler_XY").Range("C5:CE85"),
            Depo_Sehir.Plaka, Sehirler(1).Plaka)
Else
    ToplamMesafe =
        WorksheetFunction.Index(Sheets("Mesafeler_XY1").Range("C5:CE85"),
            Depo_Sehir.Plaka, Sehirler(1).Plaka)
End If
For Sehir = 1 To SehirSayisi - 1
    If Sheets("Hesaplama").Range("E8").Text = "Karayolu" Then
        Mesafe =
            WorksheetFunction.Index(Sheets("Mesafeler_XY").Range("C5:CE85"),
                Sehirler(Sehir).Plaka, Sehirler(Sehir + 1).Plaka)
    Else

```

```

    Mesafe =
    WorksheetFunction.Index(Sheets("Mesafeler_XY1").Range("C5:CE85"),
    Sehirler(Sehir).Plaka, Sehirler(Sehir + 1).Plaka)
End If
If Mesafe <> "" Then
    ToplamMesafe = ToplamMesafe + CInt(Mesafe)
End If
Next Sehir
Mesafe_Hesapla = ToplamMesafe
End Function

```

```

Public Function BubbleSort(arr() As Populasyon) As Populasyon()
    Dim IlkArr() As Populasyon
    IlkArr() = arr()
    Dim strTemp() As Sehir
    Dim strTempMesafe As Long
    Dim i As Integer
    Dim j As Integer
    Dim lngMin As Integer
    Dim lngMax As Integer
    lngMin = LBound(arr)
    lngMax = UBound(arr)
    For i = lngMin To lngMax - 1
        For j = i + 1 To lngMax - 1
            If arr(i).Sehirler(0).Mesafe > arr(j).Sehirler(0).Mesafe Then
                strTemp() = arr(i).Sehirler(0).Rota()
                strTempMesafe = arr(i).Sehirler(0).Mesafe
                arr(i).Sehirler(0).Rota() = arr(j).Sehirler(0).Rota()
                arr(i).Sehirler(0).Mesafe = arr(j).Sehirler(0).Mesafe
                arr(j).Sehirler(0).Rota() = strTemp
                arr(j).Sehirler(0).Mesafe = strTempMesafe
            End If
        Next j
    Next i
    BubbleSort = arr()
End Function

```

```

Public Function Reproduce_(RP_Sehirler() As Sehir) As Sehir()
    Dim RP_result() As Sehir
    RP_result() = RP_Sehirler()
    If (MustDoCrossovers = False) Then
        RP_result() = RastgeleMutasyon(RP_Sehirler())
        Reproduce_ = RP_result()
    Exit Function
    End If
    Dim otherIndex As Integer
    otherIndex = Rastgele_Rakam(Int(Populasyon_Sayisi / 2))
    Dim other() As Sehir

```



```

other() = Population(otherIndex).Sehirler(0).Rota()
RP_result() = CrossOver_(RP_result(), other(), MustMutateFailedCrossovers)
Reproduce_ = RP_result()
If MustMutateFailedCrossovers = False Then
    If Rastgele_Rakam(10) = 0 Then
        RP_result() = RastgeleMutasyon(RP_result())
    End If
End If
Reproduce_ = RP_result()
End Function

```

```

Public Function RastgeleMutasyon(Sehirler() As Sehir) As Sehir()
    Dim RMresult() As Sehir
    Dim MutasyonCount As Integer
    Dim MutasyonIndex As Integer
    MutasyonCount = Rastgele_Rakam(Int(UBound(Sehirler()) / 10)) + 1
    For MutasyonIndex = 0 To MutasyonCount - 1
        Dim index1 As Integer
        Dim index2 As Integer
Yeniden:
        index1 = Rastgele_Rakam(UBound(Sehirler()))
        index2 = Rastgele_Rakam(UBound(Sehirler()) - 1)
        If (index2 >= index1) Then
            index2 = index2 + 1
        End If
        If Sehirler(index1).Plaka = Depo_Sehir.Plaka Or Sehirler(index2).Plaka =
            Depo_Sehir.Plaka Then
            GoTo Yeniden
        End If
        Select Case Rastgele_Rakam(3)
            Case 1
                Sehirler() = Swap_Sehir(Sehirler(), index1, index2)
            Case 2
                Sehirler() = Move_Sehir(Sehirler(), index1, index2)
            Case 3
                Sehirler() = Reverse_Range(Sehirler(), index1, index2)
        End Select
    Next MutasyonIndex
    RMresult() = Sehirler()
    RastgeleMutasyon = RMresult()
End Function

```

```

Public Function Sehir_Kopyala(Sehirler_Kaynak() As Sehir, fromIndex As Integer,
    Sehirler_Hedef() As Sehir, toIndex As Integer, Length As Integer) As Sehir()
    Dim SKresult() As Sehir
    Dim i As Integer
    For i = 1 To Length
        Sehirler_Hedef((toIndex - 1) + i).Plaka = Sehirler_Kaynak((fromIndex - 1) +
            i).Plaka
    Next i
End Function

```

```

        Sehirler_Hedef((toIndex - 1) + i).X = Sehirler_Kaynak((fromIndex - 1) + i).X
        Sehirler_Hedef((toIndex - 1) + i).Y = Sehirler_Kaynak((fromIndex - 1) + i).Y
    Next i
    SKresult() = Sehirler_Hedef()
    Sehir_Kopyala = SKresult()
End Function

Public Function ArrayRemoveItem(ItemArray() As Sehir, ByVal Plaka As Integer) As
    Sehir()
    Dim lCtr As Long
    Dim lTop As Long
    Dim lBottom As Long
    Dim i As Long
    Dim ItemElement As Long
    Dim sonuc As Boolean

    sonuc = False

    lTop = UBound(ItemArray)
    lBottom = LBound(ItemArray)

    For i = 1 To lTop
        If ItemArray(i).Plaka = Plaka Then
            ItemElement = i
            sonuc = True
            Exit For
        End If
    Next i

    If sonuc = True Then
        If ItemElement < lBottom Or ItemElement > lTop Then
            Err.Raise 9, , "Subscript out of Range"
            Exit Function
        End If

        For lCtr = ItemElement To lTop - 1
            ItemArray(lCtr) = ItemArray(lCtr + 1)
        Next
        On Error GoTo ErrorHandler:

        ReDim Preserve ItemArray(lBottom To lTop - 1)

        GoTo Devam
    ErrorHandler:
        Err.Raise Err.Number, , _
            "You must pass a resizable array to this function"
    End If
    Devam:
    ArrayItemRemoved = sonuc

```

```
ArrayRemoveItem = ItemArray()
End Function
```

```
Public Function MutateDuplicates(Pop() As Populasyon) As Populasyon()
    Dim i As Integer
    Dim SortAgain As Boolean
    SortAgain = False
    Dim CountDuplicates As Integer
    CountDuplicates = 0
    Dim previous() As Sehir
    previous() = Pop(0).Sehirler(0).Rota()

    For i = 1 To UBound(Pop()) - 1
        Dim current() As Sehir
        current() = Pop(i).Sehirler(0).Rota()
        If (CompareArrays(current(), previous()) = False) Then
            previous() = current()
        Else
            CountDuplicates = CountDuplicates + 1
            SortAgain = True
            current() = RastgeleMutasyon(current())
            Pop(i).Sehirler(0).Rota() = current()
            Pop(i).Sehirler(0).Mesafe = Mesafe_Hesapla(current())
        End If
    Next i
    If SortAgain = True Then
        Pop() = BubbleSort(Pop())
    End If
    MutateDuplicates = Pop()
End Function
```

```
Public Function CompareArrays(Sehirler1() As Sehir, Sehirler2() As Sehir) As Boolean
    Dim X As Boolean
    Dim i As Integer
    X = True
    For i = 1 To UBound(Sehirler1())
        If Sehirler1(i).Plaka <> Sehirler2(i).Plaka Then
            X = False
            Exit For
        End If
    Next i
    CompareArrays = X
End Function
```

```
Public Sub IlkP()
    IlkPopulasyon
End Sub
```

```
Public Sub Cross()
```

```

Dim Yazdim As Boolean
MustDoCrossovers = True
Population() = Reproduce(Population())
Yazdim = Yaz(Population())
End Sub

```

```

Public Sub Mutate()
    Dim Yazdim As Boolean
    MustDoCrossovers = False
    Population() = Reproduce(Population())
    Yazdim = Yaz(Population())
End Sub

```

```

Public Sub Islem()
    IslemYap
End Sub

```

```

Public Sub Harita()
    Dim Yazdim As Boolean
    Dim Kromozom() As Sehir
    ReDim Kromozom(SehirSayisi + 1)
    Dim i As Integer
    Dim EnIyiRota() As Integer
    EnIyiRota() = Split(Sheets("Hesaplama").Cells(1, 1).Text, ",")
    For i = 0 To UBound(EnIyiRota)
        Kromozom(i + 1).Plaka = EnIyiRota(i)
    Next i
    Yazdim = Harita_Isle(0, Kromozom(), True)
End Sub

```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Saniye ÇEYREKOĞLU
Doğum Yeri	Eminönü, İSTANBUL
Doğum Tarihi	05.03.1981
Uyruğu	<input checked="" type="checkbox"/> T.C. <input type="checkbox"/> Diğer:
Telefon	0212 674 84 57
E-Posta Adresi	saniyek@thy.com
Web Adresi	



Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Endüstri Mühendisliği Bölümü
Mezuniyet Yılı	2003

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Endüstri Mühendisliği Anabilim Dalı
Programı	Endüstri Mühendisliği Programı
Mezuniyet Tarihi	Devam