

T.C.
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**AKILLI ŞEHİRLERDE ÇÖP TOPLAMA SİSTEMİNİN GENETİK ALGORİTMA
İLE ENİYİLEMESİ**

YÜKSEK LİSANS TEZİ

Mehmet ÖZMEN
1800005525

Anabilim Dalı: Bilgisayar Mühendisliği

Program: Bilgisayar Mühendisliği

Tez Danışmanı: Prof. Dr. Özgür Koray ŞAHİNGÖZ

ŞUBAT 2021

T.C.
İSTANBUL KÜLTÜR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**AKILLI ŞEHİRLERDE ÇÖP TOPLAMA SİSTEMİNİN GENETİK ALGORİTMA
İLE ENİYİLEMESİ**

YÜKSEK LİSANS TEZİ
Mehmet ÖZMEN
1800005525

Anabilim Dalı: Bilgisayar Mühendisliği
Program: Mimarlık – Bilgisayar Mühendisliği

Tez Danışmanı: Prof. Dr. Özgür Koray ŞAHİNGÖZ

Jüri Üyeleri: Doç. Dr. Akhan AKBULUT

Dr. Öğretim Üyesi Hakan AYDIN

ŞUBAT 2021

ÖNSÖZ

“AKILLI ŞEHİRLERDE ÇÖP TOPLAMA SİSTEMİNİN GENETİK ALGORİTMA İLE ENİYİLEMESİ” adlı yüksek lisans tez çalışmam süresince bilgi ve deneyimi ile çalışmalarımı yönlendiren ve desteğini esirgemeyen değerli tez danışmanım Prof. Dr. Özgür Koray Şahingöz’e, her durumda şartsız ve koşulsuz desteklerini ve sevgilerini benden esirgemeyen aileme, katkıda bulunan tüm hocalarım ve arkadaşlarıma teşekkürlerimi sunarım.



İÇİNDEKİLER

ÖNSÖZ	iii
İÇİNDEKİLER.....	i
ŞEKİL LİSTESİ.....	iii
TABLO LİSTESİ.....	v
KISALTMALAR	vi
ÖZET	vii
ABSTRACT.....	ix
1. GİRİŞ	1
1.1. Aşırı Kentleşme ve Kentlerde Atık Yönetimi Problemi	2
1.2. Atık Yönetiminde Akıllı Şehirlerin Rolü.....	3
1.3. Çöp Kamyonu Rotasının Planlanması	5
1.4. Proje Hedefleri	7
2. İLGİLİ ÇALIŞMALAR	9
2.1. Gezgin Satıcı Problemi	9
2.2. Araç Rotalama Problemi.....	10
2.3. Akıllı Şehirlerde Katı Atık Yönetimi.....	12
3. GENETİK ALGORİTMA.....	18
3.1. Kromozom ve Popülasyon.....	19
3.2. Uygunluk Fonksiyonu.....	20
3.3. Ebeveyn Seçimi	20
3.3.1. Rulet Çarkı Seçimi.....	20
3.3.2. Derece Seçimi	20
3.3.3. Kararlı Durum Seçimi.....	20
3.3.4. Turnuva Seçimi.....	21

3.3.5.	Elitizm Seçimi	21
3.3.6.	Boltzmann Seçimi.....	21
3.4.	Çaprazlama	21
3.4.1.	Partially-Mapped Crossover (PMX).....	21
3.4.2.	Order Based Crossover (OBX).....	22
3.4.3.	Modified Order Crossover (MOC)	22
3.4.4.	Order Crossover Operator (OX1)	23
3.4.5.	Edge Recombination Crossover (ERX).....	23
3.5.	Mutasyon (Mutation)	24
3.6.	2-Opt	25
3.7.	Sonlandırma	26
4.	ÖNERİLEN MODEL	27
4.1.	Google Maps API	28
4.2.	Mesafe Matrisi	29
4.3.	Genetik Algoritma	30
4.3.1.	Kromozom	30
4.3.2.	Uygunluk Fonksiyonu	31
4.3.3.	Başlatma	31
4.3.4.	Ebeveyn Seçimi	32
4.3.5.	Çaprazlama	33
4.3.6.	Mutasyon	34
4.3.7.	2-Opt Optimizasyonu.....	34
4.3.8.	Sonlandırma (Termination).....	35
5.	DENEYSEL SONUÇLAR.....	36
6.	SONUÇLAR	43
7.	KAYNAKLAR.....	44

ŞEKİL LİSTESİ

Şekil 1.1 - Akıllı Şehir, Örnek Görsel	2
Şekil 1.2 - Akıllı Şehirlerde Akıllı Atık Yönetimi	3
Şekil 1.3 - Örnek Atık Toplama Noktası, Çöp Konteynırı.....	5
Şekil 1.4 – Çöp Toplama Rotası İçin Örnek Bölge, Türkiye, İstanbul, Küçükçekmece	8
Şekil 2.1 - Bir Satıcının A, B, C ve D Şehirlerini Ziyaret Etmek İsteddiği TSP Örneği9	
Şekil 2.2 - Depo, Araç, Müşteri için Rota Örneği.....	11
Şekil 2.3 - Katı Atık Yönetimi için Kablosuz Sensör Ağ Mimarisi.....	12
Şekil 2.4 - Chaudhari, Bhole, IoT ile Katı Atık Yönetim Sistemi Tasarımı	14
Şekil 2.5 – Gutierrez ve Arkadaşları Tarafından Tasarlanan Sistem	16
Şekil 3.1 - Genetik Algoritma Örnek Akış Diyagramı.....	19
Şekil 3.2 - Gen, Kromozom ve Popülasyon Örneği	19
Şekil 3.3 - Partially-Mapped Crossover (PMX) Örneği.....	21
Şekil 3.4 - Order Based Crossover (OBX) Örneği.....	22
Şekil 3.5 - Modified Order Crossover (MOC) Örneği	22
Şekil 3.6 - Edge Recombination Crossover (ERX) Örneği.....	23
Şekil 3.7 - 2-Opt Uygulama Örneği	25
Şekil 4.1 - Küçükçekmece Lokasyonuna Rastgele Yerleştirilen Çöp Konteynırı Konumları	27
Şekil 4.2 - Google Maps Distance Matrix API ile Veri İsteme Kodu Örneği.....	28
Şekil 4.3 - Google Distance Matrix API ile Oluşturulan Mesafe Matrisi Örneği	29
Şekil 4.4 - 25 Adet Koordinat Noktasından Oluşan Örnek Başlangıç Kromozomu ..	30
Şekil 4.5 - Çalışma Kapsamında Tanımlanan Kromozomun Kod ile Gösterilmesi... 31	
Şekil 4.6 - Çalışma Kapsamında Tanımlanan Popülasyonun Kod Örneği.....	32
Şekil 4.7 - Toplam Mesafenin En Kısa Olduğunun Seçildiği Tournament Selection Örneği.....	32
Şekil 4.8 - OX1 (Order Crossover Operator) Örnek Uygulama.....	33
Şekil 4.9 - Mutasyon Operasyonu Örnek Uygulama.....	34
Şekil 4.10 - 2-Opt Algoritmasının Kod Görünümü.....	34
Şekil 5.1 - Berlin 52 Veritabanı İçin İstatistiksel ve Grafikselleştirilmiş Sonuçlar.....	36

Şekil 5.2 – İstanbul, Küçükçekmece Dolaşı Rotası, Harita Gösterimi.....	37
Şekil 5.3 - İstanbul Küçükçekmece Dolaşı Rotası, İstatistiksel Sonuçlar.....	37
Şekil 5.4 – İstanbul/Küçükçekmece Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği	38
Şekil 5.5 - İstanbul/Bağcılar Dolaşı Rotası, Harita Gösterimi	40
Şekil 5.6 - İstanbul/Bağcılar Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği	41
Şekil 5.7 - İstanbul/Bağcılar Dolaşı Rotası, Harita Gösterimi	41
Şekil 5.8 - İstanbul/Bağcılar Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği	42



TABLO LİSTESİ

Tablo 5.1 - Çalışmanın Test Edildiği Sistem Özellikleri	36
Tablo 5.2 - İstanbul/Küçükçekmece Dolaşı Rotası için Parametreler.....	38
Tablo 5.3 - Farklı Parametreler ile Yapılan Test Sonuçları	39



KISALTMALAR

ARP	: Ark Rotalama Problemi
GA	: Genetik Algoritma
CARP	: Kapasiteli Ark Rotalama Problemi
TSP	: Gezgin Satıcı Problemi
MTSP	: Gezgin Satıcılar Problemi
IoT	: Nesnelerin İnterneti
ICT	: Bilgi ve İletişim Teknolojileri
GIS	: Coğrafi Bilgi Sistemleri
VRP	: Araç Rotalama Problemi
NP-Hard	: Deterministik Olmayan Polinomal Zaman
WSN	: Kablosuz Sensör Ağı
DTN	: Veri Aktarım Düğümleri
DSS	: Karar Destek Sistemi
TCP/IP	: Ağ Protokolü
GPRS	: 2G Şebekesinden Veri İletimi Sağlayan Teknoloji
Net2Plan	: Açık Kaynak Kodlu Ağ Planlayıcı

Üniversite	:	T.C. İstanbul Kültür Üniversitesi
Enstitüsü	:	Lisansüstü Eğitim Enstitüsü
Anabilim Dalı	:	Bilgisayar Mühendisliği
Program	:	Bilgisayar Mühendisliği
Tez Danışmanı	:	Prof. Dr. Özgür Koray ŞAHİNGÖZ
Tez Türü ve Tarihi	:	Yüksek Lisans – Şubat 2021

ÖZET

AKILLI ŞEHİRLERDE ÇÖP TOPLAMA SİSTEMİNİN GENETİK ALGORİTMA İLE ENİYİLEMESİ

Şehirlerdeki demografik büyüme hızı, nüfus yoğunluğunda ve buna bağlı olarak günlük atık miktarlarında büyük bir artışa sebep olmuştur. Bu artış tüm dünyada yönetilmesi gereken en önemli konulardan biri haline gelmiştir çünkü düzgün yönetilmeyen atık toplama sistemleri çevreye/doğaya ciddi zararlar vermektedir. Bu nedenle, problem tamamen ortadan kaldırılamasa bile, bu problemi mümkün olan en iyi seviyede yönetebilen sistemlerin geliştirilmesine ihtiyaç duyulmaktadır.

Bu problemin yönetimi son yıllarda tüm dünyada hükümetlerin odak noktası olan, global adı ile “Smart City” olarak adlandırılan Akıllı Şehir kurma projelerine konu olmuştur. Akıllı Şehir projelerinde katı atıkların toplanması sensörler, ağlar, yazılımlar ve diğer akıllı şehir sistemlerine konu teknolojik gelişmeler ışığında “Atık Toplama Rota Optimizasyonu” problemi olarak tanımlanıp çözülmeye çalışılmaktadır.

Atık Toplama Rotası Optimizasyonu problemi, atık konteynirlerinin doluluğunun tespit edilerek toplama araçlarının gereksiz dolaşımını engelleyerek araç dolaşımında oluşan trafiği azaltmak, yakıt tasarrufu sağlayarak gider azaltmak ve doğaya daha az zararlı gaz salınımı yapmayı sağlayabilme problemi olarak tanımlanmaktadır.

Bu çalışmada, atık konteynirlerinin doluluklarının tespit edilerek toplama listesine alınması ve bu listedeki çöplerin en kısa yoldan toplanması amaçlanmıştır. Önerilen sistemde, atık konteynirlerinin doluluklarının tespiti, tabanlarına yerleştirilen

ve konteynır içindeki atıkların ağırlığını ölçemeye yarayan sensörlerle yapılmaktadır. Sensörlerden gelen veriye göre doluluk oranının toplanmak için yeterli olduğuna karar verilen konteynırlar toplama listesine dâhil edilmektedir. Sonrasında ise listede bulunan konteynırların konumlarına göre Genetik Algoritmalar ile mesafe hesaplamaları yapılarak toplama aracının rotası belirlenmektedir.

Anahtar Kelimeler: Atık Toplama, Akıllı Şehir, Dinamik Rota Optimizasyonu



University : T.C. İstanbul Kültür University
Institute : Institute of Graduate Studies
Department : Computer Engineering
Program : Computer Engineering
Thesis Advisor : Prof. Dr. Özgür Koray ŞAHİNGÖZ
Degree Awarded And Date : MA – February 2021

ABSTRACT

OPTIMIZATION OF GARBAGE COLLECTION SYSTEM IN SMART CITIES WITH GENETIC ALGORITHM

The demographic growth rate in the cities has led to a large increase in population density and consequently daily waste amounts. This increase has become one of the most important issues to be managed all over the world because improperly managed waste collection systems cause serious damage to the environment / nature. Therefore, even if the problem cannot be completely eliminated, there is a need to develop systems that can manage this problem at the best possible level.

The management of this problem has been the subject of Smart City building projects, called "Smart City" with its global name, which is the focal point of governments all over the world. The collection of solid wastes in Smart City projects is defined and tried to be solved as a "Waste Collection Route Optimization" problem in the light of technological developments in sensors, networks, software and other smart city systems.

Waste Collection Route Optimization problem, by determining the fullness of waste containers and preventing unnecessary circulation of collection vehicles, it is defined as the problem of reducing the traffic in vehicle circulation, reducing expenses by saving fuel and ensuring less harmful gas emission to the environment.

In this study, it is aimed to determine the fullness of waste containers, to put them in the collection list and to collect the garbage in this list in the shortest way. In the proposed system, the determination of the fullness of the waste containers is made

with sensors placed on their bottom and used to measure the weight of the waste in the container. According to the data from the sensors, the containers that are determined to be sufficient to collect are included in the collection list. Afterwards, distance calculations are made with Genetic Algorithms according to the locations of the containers in the list, and the route of the collection vehicle is determined.

Keywords: Waste Collection, Smart City, Dynamic Route Optimization



1. GİRİŞ

İnsanođlu tarih boyunca sahip olduđu imkânları hayatını kolaylařtırmaya yarayacak řekilde kullanmayı amaçlamıřtır. Dođasında bulunan bu özelliđi sebebiyle teknolojiyi her zaman bir ũste tařımayı zorlamıř ve bunu bařarmıřtır, bařarmaktadır. Birçok alanda verilen bu uđrařların ortak noktası en az efor ile en fazla iř yapmaktır. En az efor ile en fazla iř yapabilmek iin insanođlu kendisinden ve diđerlerinden daha hızlı olma yarıřına girmiřtir.

Tũm bu yarıřın iinde en fazla paya sahip konulardan biri ulařımdır. Ulařım dũnya ũzerinde sađlık, ticaret, kũltũr alıřveriři, bilgi alıřveriři, savař ve yařamın iindeki diđer tũm konuların merkezindedir.

20. yũzyılın bařlarında karayolu ulařımında motorlu tařıtlar kullanılmaya bařlanmasıyla birlikte ulařımda yeni bir dũnemi bařlamıřtır. İlerleyen dũnemlerden gũnũmũze kadar olan sũrete en fazla kullanılan tařımacılık bu motorlu araların karayolu tařımacılıđında olmuřtur. řehirlerarası ulařımın ođunda, řehir ii ulařımın neredeyse tamamında, alıřveriřte, iř yolculuklarında, tařımacılıkta ve diđer tũm ulařım alanlarında karayolu aralarının kullanımı zirvededir. Dijitalleřen dũnyanın son geldiđi noktadaki hedefi olan akıllı řehir sistemlerinin merkezinde ulařım vardır.

Gũnũmũzde tũm dũnyada, ũlke yŕneticileri ve yerel yŕnetimler "Smart City" olarak adlandırılan akıllı řehirler inřa etmeyi ve mevcut řehirleri akıllı řehirlere dŕnũřtũrmeyi planlamaktalar. Aslında IoT projesi ile ortak ilerleyen Akıllı řehir projeleri; bir ev, ofis veya herhangi bir ortamdaki elektronik eřyaların birbiri ile haberleřmesi, sonrasında ađ ũzerinden uzak lokasyonlardaki nesnelere ve ortamlar ile haberleřmesi, sonrasında ise tũm nesnelere birbiri ile haberleřmesi ve bu haberleřme ile akıllı ortamlara dâhil edilmesi projesidir. Akıllı řehir, giderek artan nũfus ve hızlı kentleřmenin tetiklediđi engelleri ortadan kaldırmaya yŕnelik fũtũristik bir yaklařımdır ve bu da hũkũmetlerin yanı sıra kitlelere de fayda sađlayacaktır [1]. řekil 1.1'de Akıllı řehir konsepti tasarımına dair rnek bir grũnũm sunulmaktadır.

Akıllı řehir, planlama, yŕnetim, inřaat ve akıllı hizmetleri kolaylařtıracak bũyũk veri, bulut biliřim, IoT ve GIS gibi zamanın gereksinimleri iin uygun bilgi ve iletiřim teknolojilerinin gsterdiđi bir kavramdır [2].



Şekil 1.1 - Akıllı Şehir, Örnek Görsel

Ulaşımın yukarıda bahsi geçen serüveni ve kendi içindeki sorunları bir tarafa ulaşım ile doğrudan ilişkili, aşırı kentleşme ile gelen çevresel ve sosyo-ekonomik zorluklardan en önemlilerinden biri de atık yönetimidir [3]. Atık toplama; küresel ısınma, hava kirliliği ve biyolojik çeşitlilikteki azalma gibi güncel ve basitçe gözlemlenebilecek problemlerden dolayı düzgün yönetilmesi ve iyileştirilmesi gereken bir süreçtir.

1.1. Aşırı Kentleşme ve Kentlerde Atık Yönetimi Problemi

Kent toplumlarında her geçen gün atık miktarı artmaktadır. 2025 yılına gelindiğinde, özellikle düşük gelirli ülkelerde 375,5 milyar dolarlık yönetim maliyetiyle 2,2 milyar ton artırması beklenmektedir [4]. Bununla birlikte, atık üretimi konusunda gelişmekte olan ülkelere sahip bölgelerin liderlik edeceği yerlerde kayma eğilimi görülmektedir.

Düzenli toplanamayan atıklar çevreye zararlı olabilir ve bunun sonucunda topluma çeşitli sağlık sorunları getirebilir. Ayrıca, kötü atık yönetiminin şehirler üzerinde ekonomik etkileri de vardır, çünkü sonradan getireceği maliyetler sorunu uygun şekilde ele almaktan daha yüksek olabilir. Evsel atıkların toplanmasını yönetmek, şehir yönetimlerinde dünya genelinde karşılaşılan çözümü zor bir problemdir.

Artarak devam eden kentleşme, verimli atık toplama işleminin önemini göstermektedir. Şehirler atık toplama çözümünü en üst düzeye çıkarmanın yollarını bulmalıdır. Atık toplama; konutlardan, ticaretten, sanayiden ve diğer tüm katı atık üretenlerden bu atıkların toplanması, taşınması ve bertaraf edilmesi ile ilgilidir. Toplama işlemi kendi kendine teslim edilen topluluk kutuları aracılığıyla evden eve

(veya kapıdan kapıya) yapılabilir. Atık toplama, toplanmayı etkileyen birçok tarafın farkında olması gereken zor bir sorundur. Toplama işlemini verimli hale getirmek zordur, çünkü bu tür problemlerin uygulanma anında kesin bir çözümü yoktur.

1.2. Atık Yönetiminde Akıllı Şehirlerin Rolü

Şehirlerin temel problemi, halk sağlığına ve yaşam kalitesine öncelik verirken biyolojik çeşitliliği ve doğal kaynakları tüketen ekosistem hizmetleri bağımlılığını yönetebilmektir [5]. Şehirlere olan sürekli göç ilgisi son zamanlarda genişlemeye sebep oldu. Bu genişleme nedeniyle yönetimi zorlaşan bu şehirleri kontrol etmek için akıllı sistemlere ihtiyaç duyulmaktadır. Akıllı trafik, akıllı sağlık hizmetleri ve akıllı atık toplama, akıllı eğitim, akıllı yaşam, akıllı enerji gibi Akıllı Şehir çalışmaları kapsamında hizmetler sunarak sürdürülebilir bir yaşam sunulması gerekiyor. Yakın zamanda çevreci önlemler alınmaz ise katı atık miktarı artacak, böylece çevre ve havayı kirletecek ve sağlık sorunları yaratacaktır. Bu nedenle, toplama sürecini optimize etmek ve atıkları daha akıllı ve daha verimli yönetmek çok önemlidir. Bir yandan bu tür zorlukları yönetip bir yandan da ortamların yaşanabilirliğin korunması, bir Akıllı Şehir yönetiminin ne kadar önemli olduğunun daha derin bir şekilde anlaşılmasını ve ortaya çıkan sorunların üstesinden gelmek için dünyadaki şehirlerin Akıllı Şehir konseptinde yönetimlerin desteklenmesi gerektiğini ortaya koyuyor.

Akıllı Şehirler; yardımcı veriler, ICT bileşenleri ve bunları yeni teknolojiler ile kombine ederek atık yönetimi sorunuyla başa çıkmak için akıllı çözümler ortaya koyar. Şekil 1.2’de örneği gösterilen akıllı atık yönetimi sistemi, geri dönüşüm kutularındaki sensörlerin ve diğer ICT bileşenlerinin kullanılmasıyla oluşturulan ortam ile atık toplama aşamasında harcanan zamandan ve paradan tasarruf sağlar [6].



Şekil 1.2 - Akıllı Şehirlerde Akıllı Atık Yönetimi

Akıllı şehir kavramı, büyük ölçüde şehirlerin çevresel yönüne ve halk ile hükümetin çevre faaliyetlerine katılmasına dayanmaktadır. Doğal kaynakların ve altyapının korunması konusunda özel bir motivasyon vardır ve daha önce tartışıldığı gibi, atık yönetimi şehirdeki sosyo-ekonomik etkiye sahip en önemli sorunlardan biridir. Gerçekten de akıllı şehirler, kentsel alanların bugün karşılaştığı ve muhtemelen yakın gelecekte karşılaşacakları zorluklarla karşı karşıyadır.

Tüm yeni kavramlar gibi, Akıllı Şehir kavramı da, henüz standart bir çerçeve veya şablon tanımına sahip değildir. Geçtiğimiz yıllarda politika alanında bu kavram büyük oranda alıntılanmıştır. Bu alana yaklaşılacak odak, Bilgi ve İletişim Teknolojileri kısaltması olan ICT'nin rolü ile ilgili gibi görünüyor. AB'deki birçok ülkede Akıllı Şehir projeleri ICT temelinde yürütülmektedir [7].

Açıkçası, ICT ekonomisi, sosyal ve çevre gibi birçok kentsel alanın içine doğrudan dâhil oldu. Ancak sadece teknoloji ve iletişimde yer almak tüm şehre fayda sağlamaz. Bazı durumlarda, akıllı şehirler, sosyal kutuplaşma gibi toplum üzerinde büyük sosyal bölünmeler yaratan bir davranış oluşturan bir sorunla uğraşmak zorunda kalabilir. Teknolojik politikaları doğrudan kullananlar olarak çoğunlukla orta sınıftan oluşan eğitilmiş ve teknolojiye ulaşabilen topluluklar, kentin geleneksel ve daha fakir sakinlerinin dahil olamadığı soyut ve elit mahalleler üretebilir [8].

Akıllı şehirlerdeki atık yönetimi sorunuyla yeni teknikler, veriler, ICT bileşenleri veya bu kavramların bir kombinasyonunu kullanarak daha akıllı çözümler ortaya çıkmaktadır. Önerilen çoklu çözümler, geri dönüşüm kutularındaki sensörler yardımıyla ICT'yi yoğun bir şekilde kullanmaktadır ve atık yönetimine atık toplama adımında harcanan zaman ve paradan tasarruf sağlayan büyük faydalar sağlamaktadır. Bazı ülkelerde bu teknolojiler halihazırda kullanılmaktadır. Yeraltı konteynerleri, yük seviyelerinin verilerini gerçek zamanlı olarak gösteren sensörler ile donatılmıştır. Bu yaklaşım, kamyonların çöpleri toplamak için kullandığı yolculukların sayısını azalttı ve sonuç olarak, süreçte elde edilen ekonomik tasarruf, karbon emisyonları ve kat edilen mesafeler de azaldı [9].

Atık yönetiminde ICT kullanımı, atıkların, çöp kamyonunun istediği zaman toplayabileceği sokaklar boyunca konulan konteynirlara atıldığı yerleşimlerde verimli çalışır. Kapıdan veya sokaklarda belirli köşelerden, belirli gün ve saatlerde çöp

kamyonlarınca toplanmak üzere bırakılan atıkların toplama işleminde ICT temelli bir yaklaşım faydalı olmayacaktır. Bu tip durumlar için farklı öneriler getirilebilir.

1.3. Çöp Kamyonu Rotasının Planlanması

Akıllı toplama yönetimi, maliyet azaltımını sağlamak, atık toplama sürecinin kapsamını ve verimliliğini artırmak için çok önemlidir. ICT bileşenlerine odaklanırken, toplama işleminin verimli bir şekilde yapılmasını sağlamak için birçok teknik kullanılabilir, ancak daha önce de belirtildiği gibi, kapıdan atık toplama işlemlerini kolaylaştırmak için çözüm aramak yerine sokaklarda statik bidonlara yerleştirilen çipler ile yöntem geliştirmek daha faydalıdır. Bu nedenle, ICT kullanılarak her kapıdan atık toplanması bu çalışma kapsamı dışındadır. Bunun yerine, bu projenin amacı, bilinen her bir kamyon için daha iyi bir rota sağlamak ve aynı zamanda bilinen sınırlamalara uymak için toplama kamyonlarının yönlendirmesini optimize etmektir. Şekil 1.3'te çöp konteynirlerinin bulunduğu örnek bir atık toplama noktası görüntüsü gösterilmektedir.



Şekil 1.3 - Örnek Atık Toplama Noktası, Çöp Konteyniri

Toplama süreci, zayıf rota planlama ve mevcut araç sayısı gibi çeşitli faktörlerden olumsuz etkilenmektedir. Kamyonların bir şehrin sakinlerine hizmet etmek için kattıkları yolların güzergâhları optimize etmek hem insanlar hem de hükümet için faydalı olacaktır. Daha iyi rotaların planlanması zor olduğu bilinen bir problemdir ve bu sorun için iyi çözümler sunmak amacıyla bazı algoritmalar oluşturulmuştur.

Rota oluřturma probleminin czm iin kullanılan en yaygın algoritmalar, Ara Rotalama Problemi (ARP) [10] ve Kapasitif Ark Rotalama Problemi (CARP) [11] varyasyonlarıdır. CARP, ARP iindeki kilit sorundur ve Ara Rotalama Problemi (VRP – Vehicle Routing Problem)'nin karřılıđıdır [12]. Her iki problem de zor kombinatoriyal optimizasyon problemleri olarak kabul edilir. Bu algoritmalar, atık toplama durumunda sokakları ve toplama noktalarını temsil eden grafikler zerinde alıřır. Bu algoritmalar arasındaki fark; ikisi arasında en ok alıřılan rotalama problemi olan VRP'nin grafikteki dđmlerin taleplerini iřlemekten ibaret olmasıdır, CARP ise dđmler yerine kenarlara hizmet etmeye odaklanır. İki algoritmanın atık ynetimi sorunuyla iliřkilendirilmesi; VRP'de, her blmenin grafikteki bir dđm olduđu ve kenarların aralarındaki sokakları temsil ettiđi konteynırlar ile bařa ıkmak iin uygulanabileceđi, CARP'da ise kenarlar sokakları ve dđmler ise sokaklar arasındaki kavřakları gsterdiđi ile ifade edilebilir. p kamyonu cpleri belirli bir cp kutusu yerine bir sokaktan toplamak zorunda olduđundan CARP yaklařımı kapıdan toplama iin daha uygundur, nk dđmler yerine kenarlara hizmet vermek bu soruna daha iyi uyuyor.

Ara tahsisi ve rota belirleme gibi toplama sorunlarıyla ilgilenen yntemler, dođrusal yntemler gibi geleneksel matematiksel yntemler kullanılarak uygulanabilir, ancak orta lekli rneklerde bile hesaplama kaynakları yetersiz kalabilir. Bu tip problemler NP-Hard problemleri olarak bilinir ve gnmze kadar karmařıklıkları nedeniyle sadece ok kk rnekler iin kesin czm yntemleri olarak kullanılabilir [13]. Bu nedenle, sezgisel ve meta-sezgisel yntemler yaklařık czmler olarak kullanmak daha dođrudur. Bu yaklařımlar sorunlara en uygun czmleri garanti etmese de genellikle gerek yařam uygulamalarında kullanılabilen iyi czmler sunar.

Birok arařtırmacı son yıllarda meta-sezgisel yntemleri kullanmıřtır ve birok durumda sezgisel taramadan daha mit verici sonular elde edilmiřtir [14]. Genetik algoritmalar literatr zerinde geniř apta incelenmiřtir ve czm alanı zerinde daha kapsamlı bir arařtırmaya izin verdiđi grlmektedir. Bu yaklařım, optimum czmlerden uzak alanlara sıradıđında iyi czmleri izleyebilir, ancak aynı zamanda rekombinasyon ve rastgele mutasyonlar da dahil olmak zere bazı tekniklerle yerel ve kresel olmayan optimum czmlerden uzaklařmaya da izin verir. Bazı arařtırmacılar, VRP ve ARP'yi bařarıyla czmek iin genetik algoritmalar uygulamaktalar.

1.4. Proje Hedefleri

Akıllı Şehir fikir ve çalışmalarının yükselişiyile, operasyonunun her yönü yeniden düşünülmüştür. Atık yönetimi, şehrin çevresi ve vatandaşların yaşamı üzerinde büyük etkisi olan bu unsurlardan biridir. Şehirlerde atıkların ekonomik ve çevresel etkilerini azaltmak için bu sorunlarla başa çıkmanın daha iyi yolları bulunmalıdır. Aslında, birçok araştırmacı akademik ve kurumsal alanda bu konularla ilgilenmektedir.

Atık yönetimi tek bir faaliyetin yönetimi değildir; daha önce de belirtildiği gibi, gerçekleştirilmesi gereken çoklu faaliyetleri içerir. Tüm bu faaliyetler arasında, bu çalışmanın konusu olan araştırma, evsel atık toplamada kamyonların güzergahlarını ve kullanımını optimize etmeyi amaçlamaktadır.

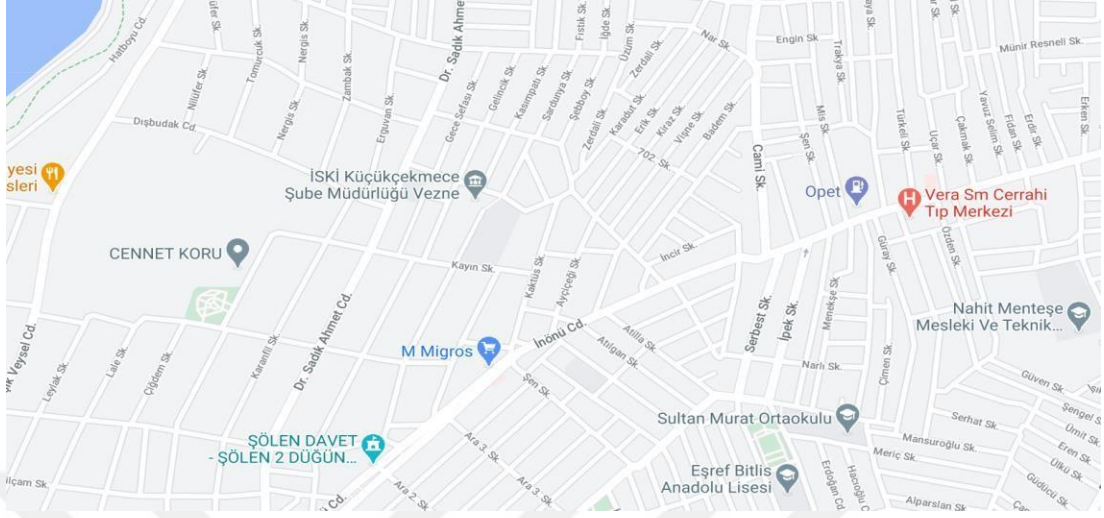
Rotalama problemi zor olarak sınıflandırılır ve NP-Hard problem kümesinde konumlandırılır. Bugüne kadar, bu problemle başa çıkmak için uygulanan kesin bir algoritma oluşturulmamıştır. Bu problemle başa çıkmak için sezgisel ve meta-sezgisel yöntemler geliştirmenin akıllı bir tercih olduğu söylenebilir.

Proje, sokak ve caddelerde belirli yerlere konumlandırılmış atık konteynırlarının altına yerleştirilen ağırlık sensörlerini kullanarak çöplerin doluluklarını tespit etmeyi ve toplanması gereken çöpleri toplayacak kamyonlara uygun rota belirlemeyi amaçlamaktadır. Yol uzunluğu, sokak yönü ve bağlantıları gibi bazı önlemleri hesaplamak için sokak bilgisine ihtiyaç vardır.

Bu projenin gerçekleştirilmesi için, konteynırların doluluk oranlarının tespiti için çöp konteynırlarına yerleştirilecek sensörlere ihtiyaç duyulmaktadır. Fakat proje kapsamında gerçek bir sensör kullanmak yerine rastgele sensör verileri oluşturulacak ve bu verilere göre çalışma gerçekleştirilecektir. Sistem gerçekte uygulanmak istendiğinde sensörlerden alınan verilerle çalışmasında herhangi bir fark olmayacaktır.

Sokak bilgilerine erişmek için Google Maps Api kullanılacaktır. Çöp kamyonunun gidebileceği her rota parçasının bir veri yapısında haritalanması ve depolanması gerekir. Bu bilgiler, çöp miktarları tespiti yapıldıktan sonra toplanması gereken çöplerin toplanma güzergahı oluşturulurken mesafe konteynırların bulunduğu noktaların mesafe hesaplamasında ve sonrasında rota belirlemede kullanılacaktır. Çalışma kapsamında Şekil 1.4'te örnek görüntüsü bulunan Türkiye, İstanbul,

Küçükçekmece (Tepeüstü civarları) bölgesinde tek araç ile toplama yapılan atık konteynırları kullanılacaktır.



Şekil 1.4 – Çöp Toplama Rotası İçin Örnek Bölge, Türkiye, İstanbul, Küçükçekmece

Konteynır doluluk oranları ve atık toplanacak alanın sokaklarını temsil eden bir veri yapısı ile birden fazla noktadan oluşan çöp kamyonunun izleyeceği rotayı hesaplamak için genetik algoritma kullanmak uygun olacaktır. Algoritmaya uygulanacak doğru operatörleri seçmek genetik algoritma geliştirme'nin zor bir adımıdır. Bu algoritma ile daha önce yapılan çalışmalar sonucunda uygulanan Araç Rotalama Problemi algoritmaları bu proje kapsamında incelendi ve bu projenin genetik operasyonlarında kullanılacaktır.

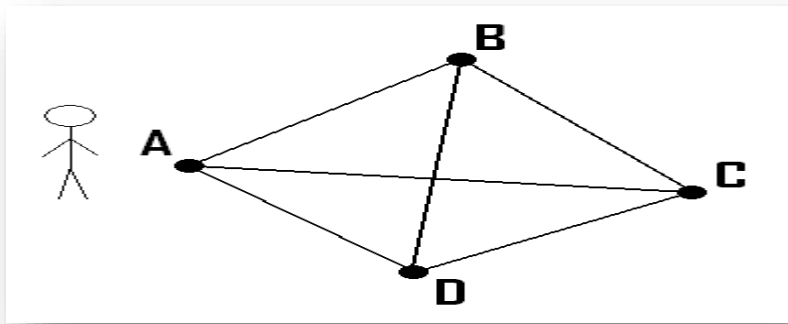
2. İLGİLİ ÇALIŞMALAR

Araç Rotalama Problemi, literatürde çoğunlukla Gezgin Satıcı Problemi (Travelling Salesman Problem – TSP) adı ile bahsedilen; bir aracın çöp toplama, ürün dağıtım, öğrenciler veya çalışanlar için servis hizmeti verilmesi vb. amaçlarla birden fazla noktaya uğrayarak tamamlayacağı bir görevde aracın minimum maliyet (uzunluk) ile yolunu belirlemeye çalışır. Çalışmamızın konusu olan çöp toplama araçlarında güzergâh, tek bir seferde toplanacak çöplerin yalnızca doluluk şartını sağlayanların göreve çıkış noktasından başlayacak ve aynı noktaya tekrar dönecek şekilde belirlenmiştir. Amaç, bir bölgede toplanması gereken bütün çöplerin tek bir dolaşım ile en kısa yoldan toplanarak aracın kat ettiği toplam mesafeyi en aza indirmektir.

Bu amaç üzerine optimizasyon, yapay zekâ teknikleri, sezgisel ve metasezgisel yöntemler, gezgin satıcı problemi, araç rotalama problemleri, akıllı şehirler, katı atık yönetimi konularında daha önce yapılan çalışmalara dayanarak edinilen bilgiler bu başlık altında açıklanmıştır.

2.1. Gezgin Satıcı Problemi

Gezgin Satıcı Problemi (TSP) bir satıcının güzergâh olarak belirlediği noktaları dolaşabilmesinin en iyi yolu (en ucuz ve en kısa) olarak tanımlanmıştır. Şekil 2.1’de örneği görünen bir gezgin satıcı A, B, C ve D şehirlerini ziyaret etmektedir. Bu problem sonucunda ortaya konulan çözümlerin kolay uygulanabilir olması ve bu problemin çok sayıda uygulama alanına hitap ediyor olması bu alanda nitelik ve nicelik olarak çok miktarda çalışma oluşmasına neden olmuştur.



Şekil 2.1 - Bir Satıcının A, B, C ve D Şehirlerini Ziyaret Etmek İstedığı TSP Örneği

Gezgin Satıcı Problemi ile ilgili daha önce Euler tarafından tasarlanan Königsberg köprüleri ile ilgili bir problemin çözümü için TSP oluşturulmuştur. Bu problem, Königsberg kentinin halkının Pregel nehri üzerindeki yedi köprüden geçiş yapmasına dair bir oyundur [15].

Karl Menger çözüm için uygun olan tüm yolları deneyip optimum sonuca ulaşma stratejisini garantilemediğini, bazen en yakın komşuyu seçmenin en kötü sonuca götürebileceğini söylemiştir [16].

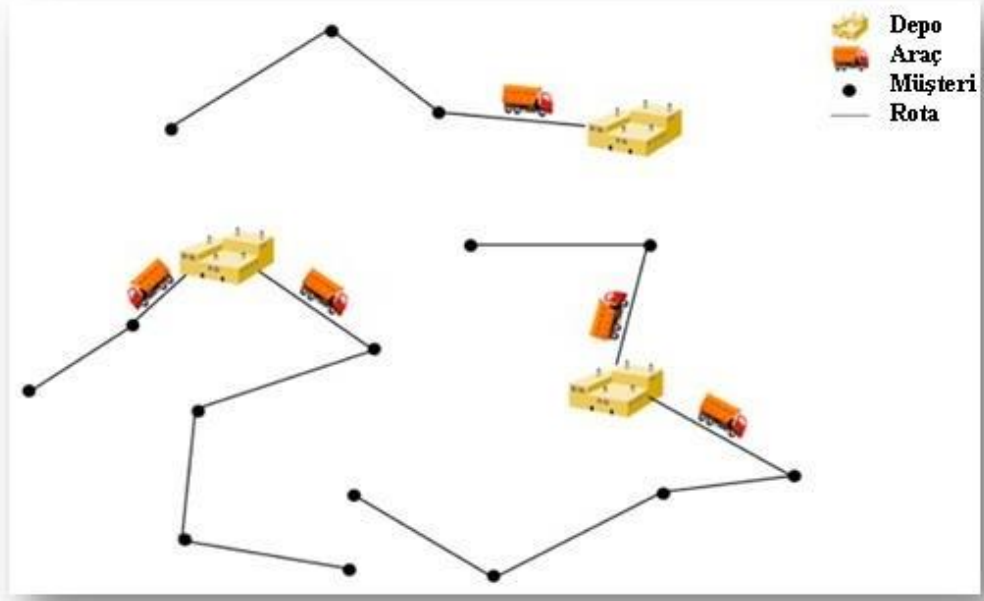
Christofides TSP için yeni bir alt sınır önermiştir. Bu alt sınır en iyi çözüm için daha önce önerilen minimum örten ağaç ve atama problemine dayalı alt sınır hesaplamalarına daıdır. 14 test problemi için yapılan çalışmada yeni alt sınır algoritması uygulandıktan sonra simetrik durumda en iyi değerin %4,7 asimetrik durumda ise en iyi değerin sadece %3,8 altında alt sınır değerleri olduğu görülmüştür [17].

Karp, şehirleri küçük gruplara bölüp her grup için en iyi turu bulup birleştiren bir bölümlenme algoritması oluşturmuştur [18]. Crowder ve Padberg kesme düzlemi algoritmasını geliştirmişlerdir. Bu çalışmada kesme düzlemi algoritması ile 318 şehirli bir problemi ve daha sonra dal-kesme algoritması ile 532 şehirli bir problemi çözmüşlerdir [19].

Alaykırın ve Engin tarafından metasezgisel algoritmalarından karınca kolonisi algoritması kullanılarak TSP üzerine bir uygulama gerçekleştirilmiştir. Karınca kolonileri metasezgiseli için uygun parametre değerleri belirlenip TSP çözümü gerçekleştirilmiştir [20].

2.2. Araç Rotalama Problemi

Araç Rotalama Problemi, belirlenen durak noktalarında ürünleri veya yolcuları almak ya da teslim etmek için tanımlanan araç filolarının etkin kullanılması problemidir [21]. Rota hesaplamalarında minimizasyon işlemleri için izlenen bir yöntemdir. Farklı problemlerde, farklı şekillerde araç rotalama problemleri tanımlanmış olup Şekil 2.2'de Depot (depo), Vehicle (araç) ve Customer (müşteri) durumlarının görüldüğü rota örneği gösterilmektedir..



Şekil 2.2 - Depo, Araç, Müşteri için Rota Örneği

Araç Rotalama Problemi ilk Dantzig ve Ramser [22] tarafından 1959 yılında tanımlanıp çalışılmıştır. 1964 yılında Clarke ve Wright [23] tarafından klasik tasarruf metodu önerilmiştir. ARP'nin değişik çeşitlerine çözüm bulmak için farklı model ve algoritmalar önerilmiştir. Uygulama alanının çokluğu ve problemin farklı olması nedeniyle ARP fazla sayıda araştırmacının ilgisini çekmiştir.

Eryavuz ve Gencer [24], 2001 yılında, Balıkesir Ordudonatım Okulu için personel servis araçlarının taşımacılık hizmeti yaparken izledikleri rotalara tasarruf algoritmalarını uygulama çalışması yapmış olup mevcut rota durumlarına göre %28'lik bir iyileştirme sağlandığı sonucuna varmışlardır.

Eren ve arkadaşları [25], Kırıkkale ilinin tıbbi atık toplama işlemleri sırasında kullanılan rotaları iyileştirmek için rota optimizasyonu çalışması yapmışlardır. Ulaşılan durum anlık rotalara göre yıllık bazda %20,63 iyileştirilebileceği sonucu ortaya çıkmıştır.

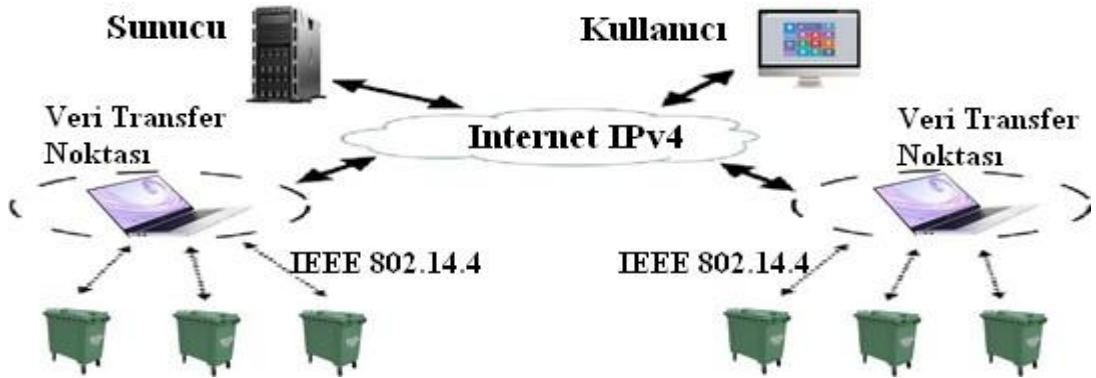
Çalışkan [26], Araç Rotalama Problemine sezgisel yöntemler ve kümeleme teknikleri ile çözüm sağlamayı amaçlamıştır. Deposu birden fazla olan dağıtım firmasının depoları k-ortalamlar tekniği ile kümelmiştir.

Bir akaryakıt dağıtım sistemi için dağıtım maliyetlerinin optimizasyonu ile ilgili bir çalışma yapan Çetin [27], problemin çözümü için başlangıç rotalarını tasarruf algoritması kullanarak oluşturduktan sonra bu rotaların optimizasyonu için genetik algoritma temelli Tabu Arama algoritması geliştirmiştir. Geliştirilen algoritmayı test verileri üzerinde uyguladığında yöntemin makul seviyede kabul edilebilir sonuçlar verdiği sonucuna ulaşmıştır.

2.3. Akıllı Şehirlerde Katı Atık Yönetimi

Katı atık yönetimi, üretme, toplama ve izleme için farklı yollar üzerinde birçok çalışma yapılmıştır. Birçok araştırmacı gelişmekte olan ülkelerde katı atık yönetimi ile ilgili farklı ekonomik, teknolojik ve yönetsel zorluklar konusunda özel rehberlik vermiştir, farklı sektörlerden üretilen katı atıklar için araştırmalar yapmıştır.

Longhi ve arkadaşları [28] Şekil 2.3'te devre tasarımı gösterilen, katı atık yönetimi için kablosuz sensör ağ mimarisi yaklaşımı ile bir uygulama geliştirmişlerdir. Ev, endüstri, çevre ve sağlık gibi çok sayıda uygulama alanında, yönetim sorunlarını ele almak için çeşitli Kablosuz Sensör Ağı (WSN) uygulamaları oluşturulmuştur. Atık yönetimi de bu metodolojinin bağlanabileceği bir alandır. Bu çalışmada atık yönetimi sürecinde takibin yapılması ve verinin taşınması için amacıyla bir yapı tasarımı önerilmiştir. Çerçeve yapısal planlama, TelosB sensör hub'larına bağlıdır ve uzak bir sunucuya atık konteynirlerinin doluluk tahminleri bilgisini vermeyi amaçlayan Veri Aktarım Düğümlerinden (DTN) yararlanır. Ayrıca, bir web programı kullanarak istemcinin çerçeveye ilişkilendirme olasılığını veren bir uzaktan kontrol düzenlemesi gerçekleştirilmiştir. Güçlü atık yönetimine bağlı varlık bütünlüğü sorunlarını çözmek için bir Karar Destek Sistemi (DSS) alıştırması yapılmıştır.



Şekil 2.3 - Katı Atık Yönetimi için Kablosuz Sensör Ağ Mimarisi

Önerilen mimari üç bölümden oluşmaktadır:

1. WSN Mimarisi: Esnek, güvenilir ve düşük maliyetli DTN'ler sağlamak için Quectel M10 GSM/GPRS modüllerinden başlayarak uzun menzilli iletişim panoları geliştirilmiştir. Bu modüller yerleşik bir ARM işlemci içerir ve dahili OpenCPU kullanılarak programlanabilir.
2. Sunucu Katmanı: Sunucu katmanı, kullanıcılar ve WSN'ler arasında arabulucu rolünü uygular. Uzun erimli iletişim modülleri ile etkileşim titizlikle ve disiplinle tasarlanmalıdır. Bu bağlamda, SEA proje mimarisinde, modem ve sunucu arasında veri aktarımına izin vermek için iki çözüm uygulanmıştır: biri TCP/IP soketi yaklaşımına dayalı ve diğeri SMS üzerine. Birincisi, alınan verilerin tutarlılığını sağlayan bir ön işlem gerçekleştiren bir arka plan programı kullanır. İkincisi GPRS bağlantısı eksik olduğunda veri toplamak için uygulanmıştır.
3. Uzun Menzilli İletişim Modülü: Sistem, kullanıcının onunla etkileşime girmesine izin vermek için iki yol sağlar: özel bir yazılım istemcisi ve bir web uygulaması. Birincisi, kullanıcı bilgisayarında kurulu bir istemciden oluşur ve merkezi veritabanına erişimi vardır. Bu çözüm biraz eski moda ve esneklikten yoksun. Diğer yandan, ikincisi modern bulut bilişimi kullanır ve bir web uygulaması üzerinden erişim sağlar. Projeye bakıldığında, en önemli gereksinimlerden biri, çöp kutularının doluluğunun uzaktan izlenmesidir. Bundan başlayarak, Gezgin Satıcı Problemi algoritması için en iyi yolu sağlar.

Chaudhari ve Bhole [29], 2018 yılında, akıllı şehirler için IoT çözümü ile katı atık toplama sistemini gerçekleştirmiştir. Nesnelerin İnterneti teknolojileri, akıllı şehirlerde hizmetleri etkin bir şekilde ele alabilir. Bu çalışmada, akıllı bir şehirlerde çöp kutusu izleme, etkin planlama ve çöp toplama kamyonlarının iletimini sağlayan IoT tabanlı katı atık yönetim sistemi teklif edilmiştir. Şekil 2.4'te tasarımı gösterilmiş olan teklif edilen sistemin ucuz fiyatlı gömülü aygıtla çevrelenmiş çöp kutuları şehrin farklı yerlerinde bulunmaktadır. Çöp kutusu konumu ile çöp seviyesinin gerçek zamanlı durumu buluta gönderilir. Toplama sürücülerini için bulut tabanlı mobil bir sistem kuruldu. Mobil uygulama, atık toplama sürücülerinin etkin ve kısa yolu kullanarak çöp kutularına ulaşmasını kolaylaştırır.



Şekil 2.4 - Chaudhari, Bhole, IoT ile Katı Atık Yönetim Sistemi Tasarımı

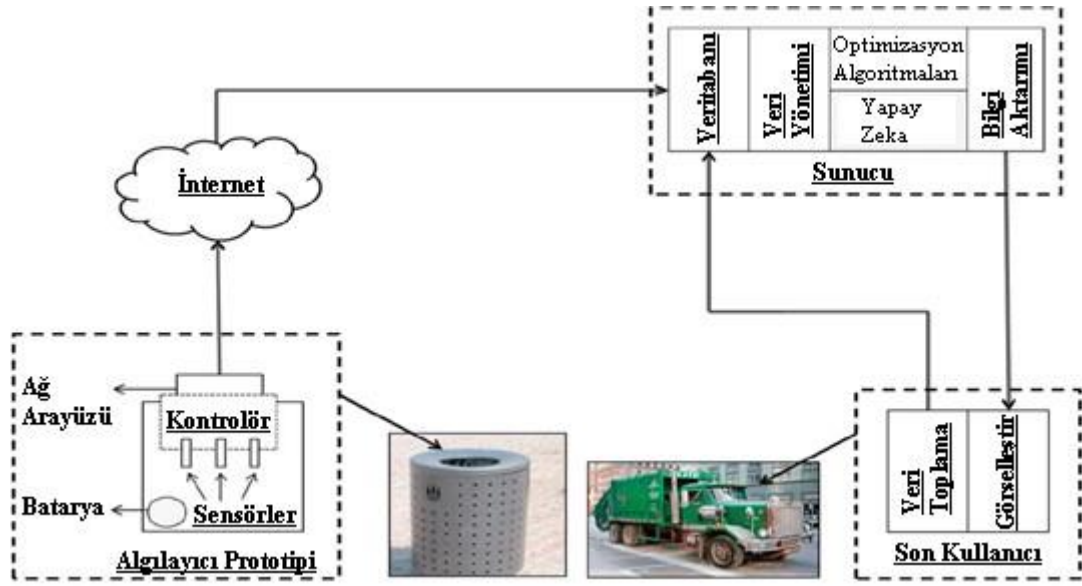
Chaudhari ve Bhole tarafından önerilen sistem, belirli bir coğrafi bölgedeki farklı yerlere yerleştirilen ıslak ve kuru çöp bidonlarından katı atıkların izlenmesini ve toplanmasını içerir ve aşağıdaki aşamalar dikkate alınarak uygulanır:

1. Çöp kutularındaki çöp seviyesini gösteren donanım prototipinin tasarımı: Her yere, çöp kutusundaki çöp seviyesini gösteren ultrasonik sensörler içeren iki çöp kutusu (ıslak ve kuru çöp için) yerleştirilir. Çöp kutusu durumunu merkezi bulut sunucusuna göndermek için mikro denetleyici ve RFID tabanlı devre kullanılır.
2. Merkezi bulut sunucusunda yazılım ve donanım modülünün geliştirilmesi: Bu modül, tüm çöp kutularından gerçek zamanlı durum güncellemeleri alacak ve sürekli olarak web uygulamasında görüntüleyecek ve ayrıca müşteri tarafında (Belediye Şirketi, Çöp toplayıcı kamyon şoförleri vb.) Mobil uygulamasında bildirimleri iletacaktır.
3. Bölgesel alan haritası üzerinde çöp kutularının mevcut konumunu ve durumunu göstermek için Mobil uygulama geliştirilmesi: Mobil uygulama, çöp toplama kamyonu sürücüleri tarafından kullanılan çöp kutularının gerçek zamanlı durumunu gösterecektir.

4. Çöp toplayıcı kamyonun tamamen dolu çöp kutularına kadar dinamik en kısa yolun üretilmesi ve gösterilmesi: Bu modül aynı zamanda çöp toplayıcı kamyonundan çöp kutularına en kısa yolu hesaplamaktan da sorumludur. Güzergâh optimizasyonu merkezi bulut sunucusu tarafından gerçekleştirilir.

Gutierrez ve arkadaşları [30], 2015 yılında, konum anlamaya dayalı akıllı atık toplama sistemi çalışmasıyla, internet üzerinden çöp hacmi verilerini okuyabilen, toplayabilen ve iletebilen sensörlerle gömülü bir IoT prototipi kullanarak istihbarat sağlamaya dayalı bir atık toplama çözümü sunmuşlardır. Uzamsal-zamansal bir bağlama yerleştirilen ve grafik teorisi optimizasyon algoritmaları ile işlenen bu veriler, atık toplama stratejilerini dinamik ve verimli bir şekilde yönetmek için kullanılabilir. Ekonomik faktörler de dahil olmak üzere geleneksel sektörel atık toplama yaklaşımlarına kıyasla böyle bir sistemin faydalarını araştırmak için deneyler yapılır. Kopenhag şehrinde Açık Veri kullanılarak, üçüncü tarafların Akıllı şehir çözümlerine katkıda bulunmaları ve geliştirmeleri için bu tür girişimlerin yarattığı fırsatları vurgulayan gerçekçi bir senaryo oluşturulmuştur.

Şekil 2.5'te gösterilen, önerilen atık toplama sistemi, bir metropoliten alandaki çöp kutularından gelen atık seviyesi verilerine dayanmaktadır. Sensörler tarafından toplanan veriler Internet üzerinden depolandığı ve işlendiği bir sunucuya gönderilir. Toplanan veriler daha sonra, toplanacak olan çöp kutularının günlük seçimini izlemek ve optimize etmek için kullanılır, bu şekilde güzergahlar hesaplanır. Her gün, işçiler yeni hesaplanan rotaları navigasyon cihazlarından alırlar. Bu sistemin temel özelliği, deneyimden öğrenmek ve sadece günlük atık seviyesi durumu hakkında değil, aynı zamanda gelecekteki durum tahmini, trafik sıkışıklığı, dengeli maliyet-verimlilik fonksiyonları ve öncelikli olan diğer faktörler hakkında karar vermek için tasarlanmasıdır. Çöp kutularının doldurulma hızı, geçmiş verilere ve gerçekleşmeden önce öngörülen taşmaya göre analiz edilebilir. Toplanacak optimize edilmiş çöp tenekeleri seçiminin, önceden tanımlanmış ekonomik gerekliliklere bağlı olarak maliyetleri düşürmesi ve toplama verimliliğini artırması beklenmektedir.



Şekil 2.5 – Gutierrez ve Arkadaşları Tarafından Tasarlanan Sistem

Delgado ve arkadaşları [31], 2019 yılında, seçmeye dayalı atık toplama için en iyi yol planlaması kapsamında bir çalışma gerçekleştirdiler. Çalışma kapsamında geliştirilen algoritma, tipik olarak iletişim ağlarını modellemek ve planlamak için kullanılan açık kaynaklı bir planlama aracı olan Net2Plan'da yürütülür. Net2Plan, akıllı kutuların konumlarını da içerebilen Net2Plan-GIS kütüphanesini kullanarak coğrafi bilgi sistemi (GIS) veritabanlarından otomatik olarak içe aktararak şehir düzeni giriş bilgilerinin algoritmaya girmesini kolaylaştırır. Algoritma, Net2Plan aracı ve uzantısı açık kaynaklıdır ve halka açık bir ortamda bulunmaktadır. Plastik atık toplama için en uygun yol planlamasının ele alındığı Cartagena (İspanya) şehrinde pratik bir vaka sunulmuştur. Bu çalışma, akıllı şehirlerin kentsel hareketlilik planlarına katkıda bulunur ve optimum yol planlama talepleri ile diğer akıllı şehirlerin senaryolarına genişletilebilir.

Sahar Idwan arkadaşları ile birlikte [32] IoT cihazları ile akıllı şehirlerde katı atık yönetimine dair çalışmasını 2019 yılında gerçekleştirmişlerdir. Bu çalışmada, atık toplama kamyonlarının programını ve yollarını belirlemek için IoT teknolojisi kullanılmıştır. Tekli kamyon yönlendirme algoritması ile ilgili önceki çalışmalar tartışılmış ve akıllı damperler ve ajan tabanlı modeller kullanarak yönetim atık filosu için ideal rotayı keşfetmek için iki aşamalı sezgisel algoritma çoklu kamyon rotalama algoritması (MITRA - multiple trucks routing algorithm) geliştirilmiştir. Akıllı damperler, atık seviyelerini ölçen sensörler ve kablosuz ağ kullanarak merkezi

yönetim sistemine güncelleme göndermek için bir kontrolör ile donatılmıştır. Hedef, yoldaki tıkanıklığı, harcanan hizmet süresini ve toplam seyahat uzunluğunu azaltarak atık toplama sürecini iyileştirmektir. Kapasitif araç rotalama problemine zamanlama kısıtlaması uygulayarak MITRA algoritmasını geliştirdik. MITRA, metropol alanını her biri bir dizi çöplük içeren bir dizi sektöre ayırır. Sunulacak sektörlerin sırasını seçerek atık toplamadaki yönlendirmeye rehberlik edilip en uygun rotayı hesaplamak için genetik algoritma uygulanmıştır.

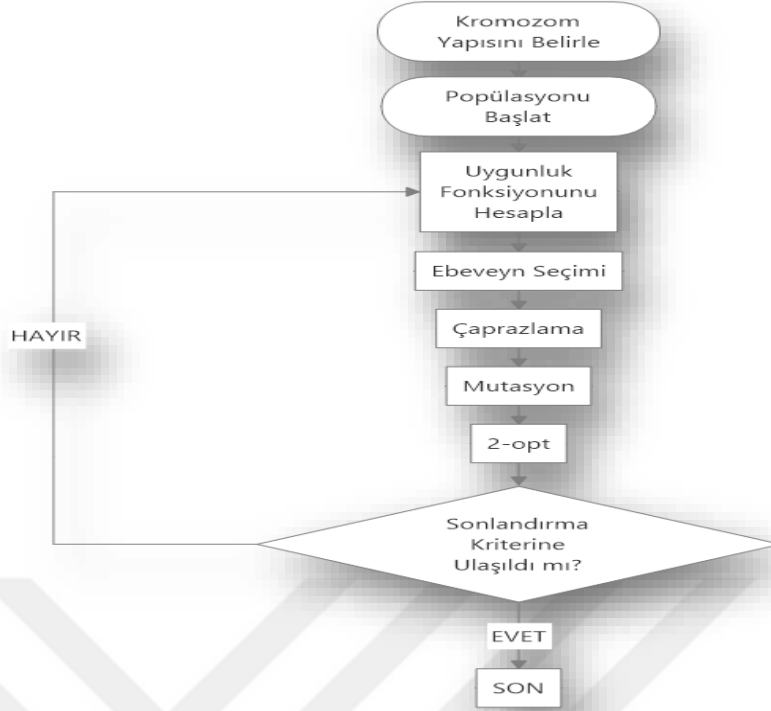
Fujdiak 2016 yılında arkadaşları ile birlikte akıllı şehirlerde gelişmiş atık toplama için genetik algoritma kullanan bir sistem geliştirmişlerdir [3]. Atık toplama lojistik prosedürünü optimize etmek için genetik algoritma uygulaması kullanılmıştır. Sunulan çözüm çöp kamyonu rotalarının hesaplanmasını sağlayıp çıktı olarak, söz konusu alana odaklanmış bir dizi simülasyon sunmaktadır. Tüm algoritmalar, gelecekteki değişiklikler açısından açık kaynak kodlu bir çözüm olarak geliştirilen entegre simülasyon çerçevesi içinde uygulanmıştır.

3. GENETİK ALGORİTMA

Genetik algoritma gerçek, evrimsel algoritmalar sınıfına ait olup 1960'ların başında John Holland tarafından tanımlanmıştır [33] [34]. Gerçek biyolojik üreme sürecini taklit eden metasezgisel bir algoritmadır [35]. Yaygın olarak işlev iyileştirici olarak kullanılır ve bilinen problemler yelpazesinde uygulanmaktadır. Yazılım tasarımının en büyük engellerinden biri olan karmaşık problemlerin yapısını tam olarak anlayıp doğal seçilimi taklit ederek çözebilir, bu tip problemlerin özellikleri ve bunlarla nasıl başa çıkılacağı bu yaklaşım ile çözüme ulaştırılabilir durumdadır.

Genetik algoritmalar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir. Emel ve Taşkın, yaptıkları çalışmada genetik algoritmaların uygulama alanlarını, genel uygulama alanları ve işletmelerdeki yaygın uygulama alanları olmak üzere iki sınıfa ayrılarak alt başlıklarda incelemiştir [36]. Uygulama alanlarını kategorize etmeden hangi problemlerin çözümünde kullanıldığından bahsedecek olursak Genetik algoritmalar; sistem ve parametre tanılama, robotik uygulamalar, kontrol sistemleri, ses ve görüntü tanıma, mühendislik ile ilgili tasarımlar, planlama, yapay zekâ uygulamaları, fonksiyon ve kombinasyonel eniyileme problemleri, ağ tasarım problemleri, yol bulma problemleri, çizelgeleme problemleri, sosyal ve ekonomik planlama problemleri için kullanılmaktadır. Bu proje kapsamında Genetik Algoritma, Gezgin Satıcı Probleminin çözümü için kullanılmıştır.

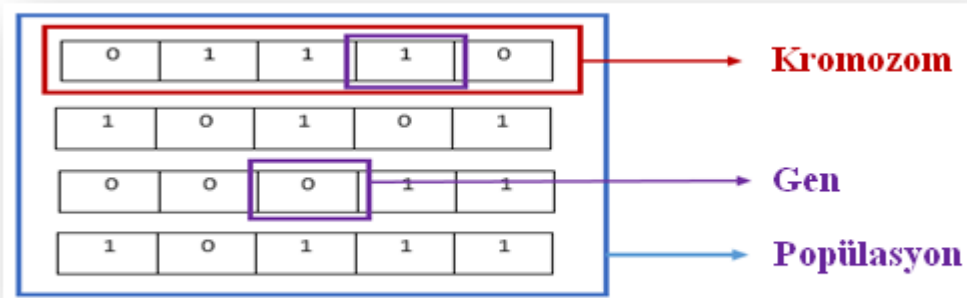
Genetik algoritmalar Gezgin Satıcı Problemi (TSP), Araç Rotalama Problemi gibi karmaşıklığından dolayı tam bir çözüm bulunamayan problemlere çözüm bulmakta başarısı kabul edilen bir algoritmadır. Genetik algoritmalar biyolojik evrime dayandığından, kullanılan terminoloji biyolojide kullanılan terminoloji ile aynıdır, ancak biyolojik muadillerine göre oldukça basit kavramları temsil etmektedir. Genetik Algoritma uygulamalarının çoğu, Şekil 3.1'de akış diyagram ile gösterilen standart adımları paylaşmaktadır. Bu akış diyagramlarının her biri bu başlık altında alt başlık olarak aşağıda incelenmektedir.



Şekil 3.1 - Genetik Algoritma Örnek Akış Diyagramı

3.1. Kromozom ve Popülasyon

Kromozom yapısı genetik algoritmanın temelini oluşturduğu için doğru şekilde oluşturulması çok önemlidir. Her biri birer gen olarak düşünülen koordinatlardan problemimize uygun şekilde kromozom yapısı oluşturulduktan sonra bu kromozomu kullanarak uygun yöntemle başlangıç popülasyonu oluşturulur. Popülasyon oluşturmak için sezgisel, rastgele gibi yöntemler kullanılmaktadır. Örnek kromozom ve popülasyon yapısı Şekil 3.2’de gösterilmektedir.



Şekil 3.2 - Gen, Kromozom ve Popülasyon Örneği

3.2. Uygunluk Fonksiyonu

Uygunluk fonksiyonu, bireyin genetik algoritma uygulamasındaki çözüm uzayına uygun olup olmadığını değerlendiren hedef fonksiyondur. Bu değer bireyin gelecek nesil için seçilme şansını doğrudan etkilediği için çok önemlidir. TSP tipi problemler için uygunluk fonksiyonu noktalar arasındaki her bir mesafenin toplamı olarak değerlendirilebilir.

3.3. Ebeveyn Seçimi

Bu süreç gelecek nesil için yavru birey oluşturmak üzere ebeveynlerin seçim sürecidir. Çeşitlilik ile iyi değere sahip bireyler oluşturabilmek için ebeveyn seçim adımı dikkatli şekilde ele alınmalıdır. Birçok ebeveyn seçimi bulunmakla birlikte bilinen ve sıklıkla kullanan yöntemler aşağıdaki gibidir:

3.3.1. Rulet Çarkı Seçimi

Rulet çarkı seçiminde, gelecek neslin yetiştirilmesi için bir birey seçme olasılığı, uygunluğuyla orantılıdır. Uygunluk ne kadar iyi olursa, o kişinin seçilme şansı o kadar yüksektir. Bireylerin seçilmesi, mevcut nesildeki bireyler kadar kapasitesi olan bir ruleti, olasılıklarına bağlı olarak boyutları ile döndürmek olarak tasvir edilebilir.

3.3.2. Derece Seçimi

Negatif uygunluk değerleriyle çalışır ve çoğunlukla popülasyondaki bireylerin çok yakın uygunluk değerlerine sahip olduğu durumlarda kullanılır (bu genellikle çalışmanın sonunda olur). Bu, her bireyin pastadan neredeyse eşit bir paya sahip olmasına yol açar (uygun orantılı seçim durumunda olduğu gibi) ve bu nedenle, her bireyin birbirine göre ne kadar uygun olursa olsun, ebeveyn olarak seçilme olasılığı yaklaşık olarak aynıdır. Bu da daha fit bireylere yönelik seçim baskısında bir kayba yol açarak, GA'nın bu gibi durumlarda kötü ebeveyn seçimleri yapmasına neden olur.

3.3.3. Kararlı Durum Seçimi

Bu, ebeveynleri seçmenin belirli bir yöntemi değildir. Bu seçimin ana fikri, kromozomların büyük bir kısmının gelecek nesil için hayatta kalması gerektiğidir.

3.3.4. Turnuva Seçimi

Turnuva Seçimi, bireyi bireyler arasından seçmenin bir yöntemidir. Her turnuvanın galibi geçiş yapmak için seçilir.

3.3.5. Elitizm Seçimi

Genellikle daha iyi parametreler elde etmek için kısmi çoğaltmalı stratejiler kullanılır. Bunlardan biri, son nesilden en iyi bireylerin küçük bir kısmının (herhangi bir değişiklik yapılmadan) bir sonrakine taşındığı elitizmdir.

3.3.6. Boltzmann Seçimi

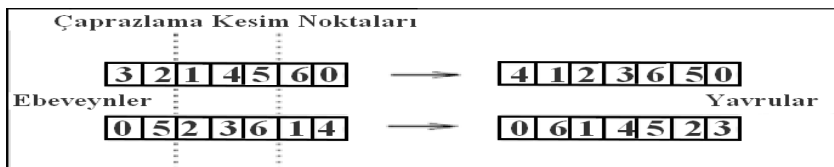
Boltzmann seçiminde, sürekli değişen bir sıcaklık, seçim oranını önceden ayarlanmış bir programa göre kontrol eder. Sıcaklık yüksek başlar, bu da seçim basıncının düşük olduğu anlamına gelir. Sıcaklık kademeli olarak azaltılır, bu da seçim basıncını kademeli olarak artırır, böylece uygun çeşitlilik derecesini korurken GA'nın arama alanının en iyi kısmına daha yakın daralmasına izin verir.

3.4. Çaprazlama

Çaprazlama (Crossover), ebeveynler seçildikten sonra gerçekleştirilen ilk adımdır. Bu adımda bir ebeveynden alınan bazı genlerin diğer ebeveyne yerleştirilmesi ve bu diğer ebeveynde çakışan genlerin çıkarılması şeklinde uygulanır. Bu çalışma kapsamında birden fazla çaprazlama yöntemi denenmiş olup bazıları bu başlık altında kısaca açıklanmıştır. Örneklerde Parent: Ebeveyn, Offspring: Yavru anlamına gelmektedir.

3.4.1. Partially-Mapped Crossover (PMX)

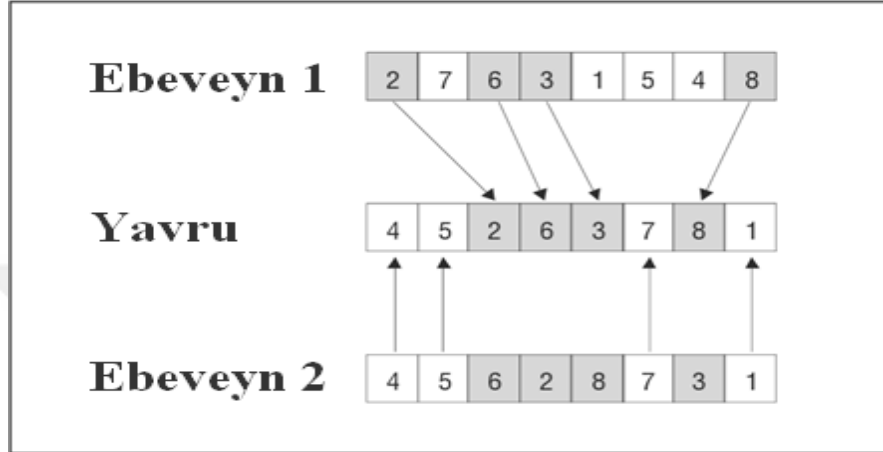
Şekil 3.3'te örneği verilen PMX, ebeveynlerden yavrulara sıralama ve değer bilgilerini iletir. Bir ebeveynin bir bölümü, diğer ana dizinin bir bölümü ile eşlenir ve geri kalan bilgi değiş tokuş edilir [37].



Şekil 3.3 - Partially-Mapped Crossover (PMX) Örneği

3.4.2. Order Based Crossover (OBX)

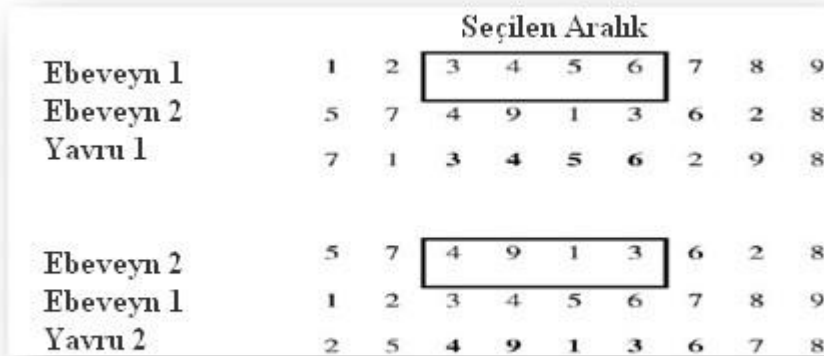
OBX ile çaprazlama yaparken Şekil 3.4'teki gibi, ebeveynlerden rastgele birkaç pozisyon seçilir ve bu ebeveynin seçilen pozisyonlarındaki noktalar sıralı olarak diğer ebeveyne geçirilerek yavru birey oluşturulur. Diğer yavru da aynı yöntemle diğer ebeveyninden türetilir [38].



Şekil 3.4 - Order Based Crossover (OBX) Örneği

3.4.3. Modified Order Crossover (MOC)

Örneği Şekil 3.5'te gösterilen MOC, rastgele seçilen bir nokta, ebeveynleri sol ve sağ alt dizelere böler. Ebeveynlerin sağ alt dizeleri seçilir. Alt dize seçilmesinden sonra süreç, the Order Crossover ile aynıdır. Tek fark, bir ebeveynde rastgele birkaç konum seçmek yerine, rastgele seçilen geçiş noktasının sağındaki tüm konumların seçilmesidir [39].



Şekil 3.5 - Modified Order Crossover (MOC) Örneği

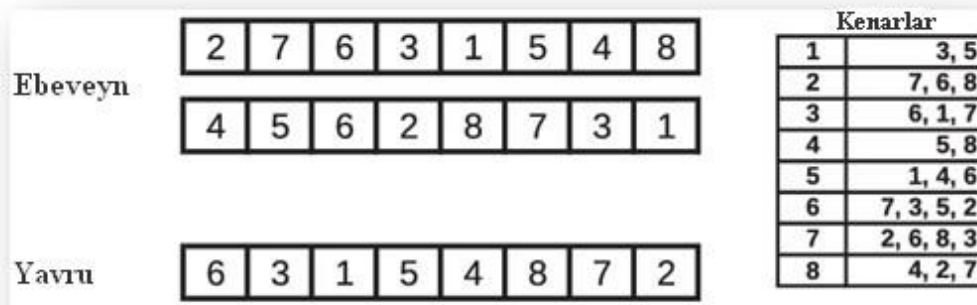
3.4.4. Order Crossover Operator (OX1)

Bu proje kapsamında denenmiş olan diğer crossover operatörleri arasından tercih edilmiş olan OX1, ebeveynlerden birinden rastgele alınan bir aralığın diğer ebeveyne yerleştirilmesi ve söz konusu diğer ebeveynde kalan diğer elemanların -yerleştirilen parçaların çıkarılması şartıyla- yerleştirilen son elemandan sonraki indeksten başlayarak dizi bittiğinde başa dönerek yazılması ile yeni yavrunun oluşturulması işlemidir. Bu çaprazlama biçimi çalışma detayında ayrıca örneklendirilecektir [40].

3.4.5. Edge Recombination Crossover (ERX)

Çoğu zaman, alternatif edge operatörü, tur genişletme seçenekleri sınırlı olduğunda yavruda birçok rastgele kenar ortaya çıkarır. Yavrunun ebeveynlerden mümkün olduğunca çok sayıda kenar alması gerektiğinden, rastgele kenarların girişi en aza indirilmelidir. Edge Recombination Operator, "kenar haritası" adı verilen özel bir veri yapısı ile alternatif kenar yaklaşımının yakınsama davranışını azaltır.

Temel olarak, ebeveyn turlarında her şehre rastlayan ve henüz yavrulara dahil edilmemiş noktalara götüren kenarların listesini tutar. Bu nedenle, bu kenarlar turu uzatmak için hala kullanılabilir ve aktif oldukları söylenebilir. Burada dikkat edilmesi gereken, minimum sayıda aktif kenara sahip şehre giden kenarı seçerek turu uzatmaktır. İki veya daha fazla nokta arasında eşitlik olması durumunda bu şehirlerden biri rastgele seçilir. Bu strateji ile, yaklaşımın bir "çıkılmazda", yani rastgele bir kenarın seçilmesini gerektiren hiçbir aktif kenarı olmayan bir şehirde sıkışıp kalma olasılığı daha düşüktür [41][42]. Şekil 3.6'da ERX uygulama örneği gösterilmektedir.



Şekil 3.6 - Edge Recombination Crossover (ERX) Örneği

3.5. Mutasyon (Mutation)

Mutasyon, genetik algoritma kromozomlarından oluşan popülasyonun bir jenerasyonundan diğerine genetik çeşitliliği sürdürmek için kullanılan genetik bir operatördür. Biyolojik mutasyona benzer. Mutasyon, bir kromozomdaki bir veya daha fazla gen değerini başlangıç durumuna göre değiştirir. Mutasyonda, çözüm önceki çözümden tamamen değişebilir. Dolayısıyla GA, mutasyon kullanarak daha iyi bir çözüme ulaşabilir. Mutasyon, kullanıcı tarafından tanımlanabilen bir mutasyon olasılığına göre evrim sırasında meydana gelir. Bu olasılık düşük ayarlanmalıdır. Çok yükseğe ayarlanırsa, arama ilkel bir rastgele aramaya dönüşecektir.

Bir mutasyon operatörünün klasik örneği, bir genetik dizideki rastgele bir bitin orijinal durumundan çevrilmesi olasılığını içerir. Mutasyon operatörünü uygulamanın yaygın bir yöntemi, bir dizideki her bit için bir rastgele değişken oluşturmayı içerir. Bu rastgele değişken, belirli bir bitin çevrilip çevrilmeyeceğini söyler. Biyolojik nokta mutasyonuna dayanan bu mutasyon prosedürüne tek nokta mutasyonu denir. Diğer türler ters çevirme ve kayan nokta mutasyonudur. Gen kodlaması permütasyon problemlerinde olduğu gibi kısıtlayıcı olduğunda, mutasyonlar değiş tokuş, tersine dönme ve karıştırma.

Farklı kromozom tipleri için farklı mutasyon çeşitleri aşağıda özetlenmiştir:

Bit String: Bit dizilerinin mutasyonu, rastgele pozisyonlarda bit çevirmeleriyle elde edilir.

Flip Bit: Bu mutasyon operatörü, seçilen genomu alır ve bitleri tersine çevirir (Örneğin, genom biti 1 ise 0'a değiştirilir ve bunun tersi de geçerlidir). **Gaussian:** Bu operatör, seçilen gene bir birim Gauss dağıtılmış rasgele değer ekler. Söz konusu gen için kullanıcı tarafından belirlenen alt veya üst sınırların dışına çıkarsa, yeni gen değeri kırılır. Bu mutasyon operatörü yalnızca tam sayı ve kayan genler için kullanılabilir.

Non-Uniform: Bir sonraki nesilde mutasyon miktarının 0'a gitme olasılığı, tek tip olmayan mutasyon operatörü kullanılarak artırılır. Evrimin ilk aşamalarında nüfusun durmasını engeller. Çözümü evrimin sonraki aşamalarında ayarlar. Bu mutasyon operatörü yalnızca tam sayı ve kayan genler için kullanılabilir.

Uniform: Bu operatör, seçilen genin değerini, o gen için kullanıcı tarafından belirlenen üst ve alt sınırlar arasında seçilen tek tip rasgele bir değerle değiştirir. Bu mutasyon operatörü yalnızca tam sayı ve kayan genler için kullanılabilir.

Shrink: Bu operatör, giriş ana vektörünü karakterize eden her bir karar değişkeninin orijinal değerine eşit ortalama ile bir Gauss dağılımından alınan rastgele bir sayı ekler.

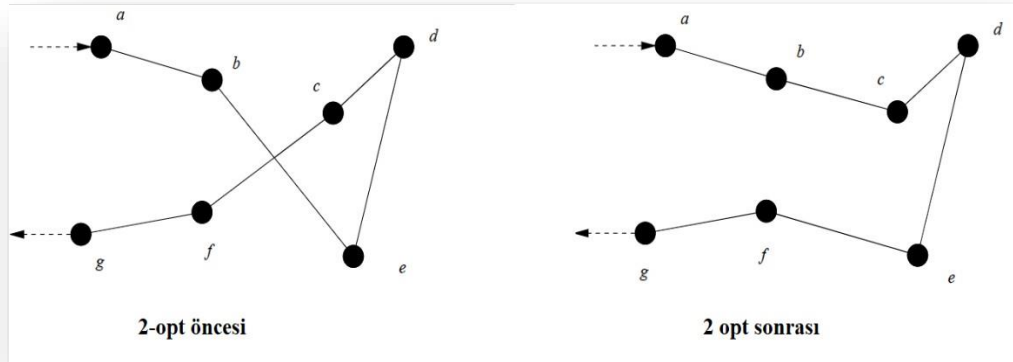
Boundary: Bu mutasyon operatörü, genomu rasgele alt veya üst sınırla değiştirir. Bu, tam sayı ve kayan genler için kullanılabilir.

Swap: Bu mutasyon tipinde 2 kromozom arasında seçilen genler arasında değiş tokuş yapılmakla birlikte, bu yöntem ileride detaylandırılacaktır.

3.6. 2-Opt

2-opt optimizasyonunda, seyahat eden satıcı problemini çözmek için basit bir yerel arama algoritmasıdır. 2-opt algoritması ilk olarak 1958'de Croes tarafından önerildi [43], ancak temel hareket zaten Flood tarafından önerilmişti [44]. Bunun arkasındaki ana fikir, kendi üzerinden geçen bir rotayı alıp, olmaması için yeniden sıralamaktır.

2-opt araması, çakışan rotaların takasına dair olası tüm kombinasyonun deneyecektir. Bu teknik, seyyar satıcı probleminin yanı sıra birçok ilgili soruna da uygulanabilir. Şekil 3.7 bir rotanın 2-opt uygulanmadan önceki ve uygulandıktan sonraki durumunun örneğini göstermektedir.



Şekil 3.7 - 2-Opt Uygulama Örneği

3.7. Sonlandırma

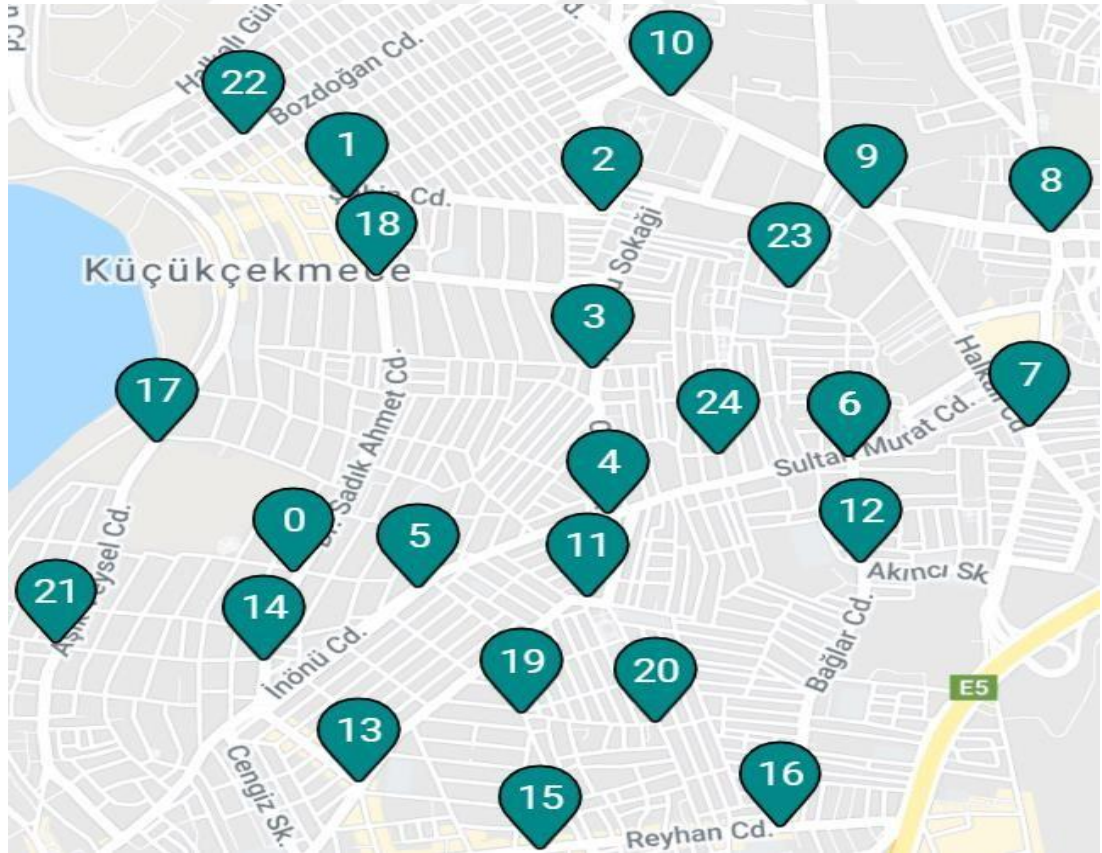
Genetik Algoritma akış diyagramını sonlandırmadan önceki kontrol kriteri olan bu adımda algoritmayı çalıştırmaya devam edip edilmeyeceğine karar verilir. Bu karar için birden fazla kriter konulabilmektedir. Uygulamamız kapsamında değerlendirilen sonlandırma kriterleri aşağıda iletilmekte olup uygulamanın kullandığı sonlandırma kriterinden çalışma detayında ayrıca bahsedilecektir:

- Maksimum iterasyon sayısına ulaşıldıysa
- Uygunluk değeri son 100 iterasyonda değişmediyse
- Uygunluk değeri ...% miktarın altında değişmeye başladıysa
- Uygunluk değeri maksimum iterasyon sayısının ...% miktarı kadar döngüde değişmiyorsa
- Uygunluk değeri gerçekleştirilen toplam iterasyon sayısının ...% miktarı kadar döngüde değişmiyorsa
- Minimum iterasyon sayısına ulaşılmadıysa (döngü devam eder)

4. ÖNERİLEN MODEL

Önerilen model, akıllı şehirlerdeki çöp konteynırlarının toplanması için çöp kamyonlarına kısa ve verimli rotalar planlama sorununa genetik algoritma ile bir çözüm geliştirmeyi amaçlamaktadır. Projede çöp konteynırlarına yerleştirilen sensörler aracılığı ile doluluk bilgisinin alındığı varsayılp koordinasyonu verilen konteynırların bir çöp kamyonu ile en kısa yoldan dolaşabilmesi için bir algoritma üzerinde çalışılmıştır. Bu bölüm, projenin tüm gelişimine, karşılaşılan problemlere, bu problemleri çözmek için uygulanan çözümlere ve problemlere nasıl yaklaşıldığına dair açıklama ve gerçekleştirilen uygulamalara yer vermektedir.

Çalışma için test amaçlı kullanılan koordinat dizilerinden Küçükçekmece lokasyonunda bulunduğu varsayılan çöp konteynırlarının konumunu gösteren harita Şekil 4.1'de gösterilmektedir. Çalışma kapsamında Şekil 4.1'de gösterilen ve buna benzer lokasyonların dolaşmalarını yapmak üzere Genetik Algoritma ile en iyi rotanın hesaplandığı bir sistem önerilmektedir.



Şekil 4.1 - Küçükçekmece Lokasyonuna Rastgele Yerleştirilen Çöp Konteynırı Konumları

4.1. Google Maps API

Google Maps API, özel bir harita, aranabilir harita, check-in işlevleri oluşturmak, konumla eşzamanlı canlı verileri görüntülemek, rotaları planlamak vb. işler için güçlü araçlar sunan bir platformdur. Bu çalışmada Google Maps Distance Matrix API kullanılarak noktalar arası mesafeler canlı olarak alınmıştır. Ayrıca rota çizdirmek için de Google Maps Directions API kullanılmıştır.

Google Maps Distance Matrix API ile mesafe ölçümleri 4 farklı seyahat modu şeklinde alınabilmektedir: Bisiklet, Araba, Toplu Taşıma ve Yaya. Ayrıca trafik yoğunluğunu dikkate alma hususunda da seçenek sunmaktadır. Bunların dışında da özellikler sunmakta olup projemizde önemli olarak görülen nokta trafik yoğunluğunun anlık olarak dikkate alınıp rotanın bu doğrultuda çizdirilmesidir.

Çalışmada Distance Matrix ile elde edilen verilerin işlenerek Genetik Algoritma ile rota oluşturulduktan sonra bu rotanın çizdirilmesi Google Maps Directions API ile yapılmaktadır. Rota poligon olarak çizdirilmeyip gerçek güzergâh olarak çizdirilmiştir. Böylece rotanın oluşturulup çizilmesine kadar anlık rota çıkarılması sağlanmıştır. Distance Matrix API örnek sorgusu Şekil 4.2’de gösterilmektedir.

```
{ origins: [{lat: 55.93, lng: -3.118}, 'Greenwich, England'],
  destinations: ['Stockholm, Sweden', {lat: 50.087, lng: 14.421}],
  travelMode: 'DRIVING',
  drivingOptions: {
    departureTime: new Date(Date.now() + N),
    trafficModel: 'optimistic' } }
```

Şekil 4.2 - Google Maps Distance Matrix API ile Veri İsteme Kodu Örneği

Google Maps Distance Matrix API kütüphanesini projeye entegre etmek için .Net Core ile yazılmış açık kaynak kodlu vivet/GoogleApi [45] kütüphanesi kullanılmıştır. Kütüphane C# kodu kullanarak Google API kütüphanesinden kolayca veri çekebilmeye imkân sağlamaktadır.

4.2. Mesafe Matrisi

Üretilen her kromozom için toplam mesafeyi hesaplamak üzere koordinatlar arası mesafelere ihtiyaç duymaktayız. Bu mesafeleri her defasında sorgulamamak için bir mesafe matrisi oluşturulur ve Google API ile alınan mesafeler bu matris içinde tutulur. Başlangıç/hedef noktası ile birlikte N tane koordinattan oluşan bir sistemde N x N boyutlarında bir matris oluşturulur. Matris uygulamanın her çalışmasında yeniden oluşturulur ve statik olarak tutulur.

Şekil 4.3’de Distance Matrix API kullanarak alınmış 15 koordinatlık bir rotaya ait mesafe matrisi görülmektedir. Bu mesafe matrisinde dikkat edilmesi gereken bir nokta, matristeki mesafe verilerinin simetrik olmamasıdır. Normal şartlarda A-B yönlü bir mesafe ölçümü ile B-A yönlü bir mesafe ölçümü arasındaki uzaklıkların eşit olması beklenirken burada eşit olmayıp matrisin simetrik görünmediği fark edilmektedir. Bunun sebebi, bazı yolların tek yönlü olması, bazı yolların ise dönüş yönünde trafik olduğu için farklı yol tercih edilmesidir.

0	1252	1775	1211	1219	421	1847	2949	3119	2479	2365	1501	2178	1182	302
1252	0	626	1012	1502	1452	2001	2877	2151	1511	1001	1784	2332	2411	1531
1824	626	0	613	1204	1486	1497	2251	1525	885	770	1456	1828	2321	2104
1211	1012	613	0	610	873	989	2634	1908	1268	1154	862	1320	1802	1490
999	1502	1067	473	0	522	628	1729	1959	1722	1607	282	958	1121	1000
477	1706	1772	1159	944	0	1571	2673	3068	2428	2313	1226	1902	1270	478
1626	2020	1335	1008	628	1150	0	1029	1352	1177	1875	879	331	1718	1628
2513	2675	2048	2432	1514	2036	1167	0	622	1798	2171	1765	1410	3460	2514
3272	2304	1678	2061	2244	2934	1679	1283	0	2269	1801	2496	2218	3335	3552
2645	1677	1050	1434	1594	2307	1028	1366	640	0	1173	1845	1567	2684	2924
2337	1369	743	1126	1717	1999	1635	1972	1246	606	0	1968	2174	2834	2616
986	1709	1548	954	433	510	933	2035	2264	1931	2089	0	1044	839	988
1957	2351	1666	1339	958	1481	331	1030	1683	1508	2206	1210	0	1853	1959
788	2018	2356	1739	1384	952	2011	3113	3343	3011	2896	1666	2342	0	495
302	1531	2055	1490	1499	700	2126	3228	3399	2759	2644	1781	2457	923	0

Şekil 4.3 - Google Distance Matrix API ile Oluşturulan Mesafe Matrisi Örneği

4.3. Genetik Algoritma

Projede nihai hedefe ulaşmak için projenin motivasyonu, sorun ve hedefler açıkça tanımlanıp sürecin her bir parçasının nerede bulunabileceği ile ilgili olarak, uygulanan GA metodolojisinin adımlarından yukarıda ayrıntılı olarak bahsedilmiştir. Bu bölümde ise doğrudan proje kullanılan teknikler anlatılmıştır.

Bu proje, bir bölgedeki bir çöp kamyonuna çöp toplama sırasında verimli rotalar planlamayı genetik algoritma kullanarak yapmayı amaçlamaktadır. Çöp kamyonunun rotası, yola çıkmadan önce çöp konteynirlerindeki ağırlık sensörlerinden alınan doluluk verilerine göre dinamik olarak oluşturulur. Bu sensörler, tüm işlemleri tetiklemek için önemlidir. Proje, sensörlerden alınan verilere göre rotaların sisteme aktarılacağı varsayımı ile çalışmaktadır.

4.3.1. Kromozom

Bu problemde kromozom, ziyaret edilecek noktaların koordinat numaralarını temsil eden numaraların bir dizi içinde tutulmasından oluşur. Her bir çöp konteynirinin adresi X ve Y koordinat değerleri olarak alınır ve bu değerlere numaralar atanır. Bu liste daha sonra kullanılmak üzere düğüm adı verilen bir sınıf listesinde tutulur ve yalnızca numara bilgileri diziye atanarak başlangıç kromozomu oluşturulur. Numaraları tekrar etmemek için 0'dan başlayıp düğüm sayısı eksi 1'e kadar olan numaralar ardışık olarak koordinatlara atanır ve düğümler için numaralar verilir. Şekil 4.4'te 25 adet noktadan oluşan bir koordinat dizisinin örnek kromozom görüntüsü yer almaktadır.

1	2	3	4	5	21	22	23	24	25
---	---	---	---	---	-------	----	----	----	----	----

Şekil 4.4 - 25 Adet Koordinat Noktasından Oluşan Örnek Başlangıç Kromozomu

Ayrıca her biri gen olarak belirlenen bu koordinat numaraları önerilen sistemde integer bir değer olarak tutulmaktadır ve sayı dizisi olarak tutulduğu kodun sunumu Şekil 4.5'tedir. Burada "Route" isimli sınıf rotayı göstermekte olup "NodeNumber" olarak tanımlanan özellik integer veri tipindeki gen dizisinden oluşan kromozomu göstermektedir. "TotalCost" isimli özellik ise bir sonraki adımda anlatılmıştır.

```
public class Route
{
    public int[] NodeNumber { get; set; }
    public double TotalCost { get; set; }
}
```

Şekil 4.5 - Çalışma Kapsamında Tanımlanan Kromozomun Kod ile Gösterilmesi

4.3.2. Uygunluk Fonksiyonu

Bu problem için uygunluk fonksiyonu, popülasyondaki kromozomun toplam mesafesinin minimum değeridir. Mesafeyi hesaplamak için aşağıdaki adımlar atılır:

1. Madde 3.2’de anlatılan mesafe matrisi kullanılır.
2. Kromozomdaki ardışık her 2 düğüm arasındaki mesafe değerinin toplamını alarak kromozoma karşılık gelen rotanın toplam mesafesini hesaplamak için madde 3.2’de bahsedilen mesafe matrisi kullanılır.
3. Dolaşımı tamamlamak üzere ilk ve son nokta arasındaki mesafe de toplam mesafeye ayrıca eklenir.

Şekil 4.5’te görünen Route isimli sınıfta tutulan “TotalCost” özelliği uygunluk fonksiyonu işlemi ile ulaşılan toplam yol miktarını göstermektedir.

4.3.3. Başlatma

Kromozom oluşturulduktan sonra, genetik algoritma sürecini başlatmak için popülasyon oluşturulmalıdır. Nüfus oluşturmak için boyut ve yöntem, problemlerin üstesinden gelinmesi gereken konulardır. Optimal popülasyon büyüklüğüne deneme ile karar verilir. Az sayıda popülasyon, çiftleşme havuzundaki çeşitliliği olumsuz etkilerken, çok fazla popülasyon yavaş uygulamaya neden olabilir.

Popülasyonu başlatmak için bir yöntem olarak sezgisel veya rastgele iki önemli yoldur. Sezgisel başlatma, çözüm bulmak için daha hızlıdır, ancak çeşitliliği engellediği için muhtemelen en iyi çözümü bulamaz. Bu nedenle, bu projede rastgele karıştırma düğümleri olarak rastgele başlatma yöntemi kullanılmıştır.

Önerilen modelde popülasyon 100 adet kromozomdan oluşmakta olup popülasyon sınıfını gösteren kod örneği Şekil 4.6’da gösterilmektedir. Burada “Routes” isimli özellik kromozom listesini, “MinimumCost” ise kromozomların sahip oldukları toplam mesafeler arasında en düşük olanı tutmaktadır.

```
public class Population
{
    public Population(List<Route> routes)
    {
        Routes = routes;
        MinimumCost = routes.Min(selector: x:Route => x.TotalCost);
    }
    public List<Route> Routes { get; set; }
    public double MinimumCost { get; set; }
}
```

Şekil 4.6 - Çalışma Kapsamında Tanımlanan Popülasyonun Kod Örneği

4.3.4. Ebeveyn Seçimi

Ebeveyn seçimi tekniklerinden daha önce bahsedilmiş olup bu çalışma için Tournament Selection [30] metodu kullanılmıştır. Tournament Selection yönteminde özetle aşağıdaki adımlar uygulanmıştır:

1. Popülasyon içinden rastgele 2 veya daha fazla sayıda birey seç.
2. Seçilen bu bireylerden uygunluk fonksiyonu en iyi olanı 1. ebeveyn olarak seç.
3. Aynı adımları diğer ebeveyn için de uygula.

Bu çalışmada Tournament Selection için popülasyon içinden rastgele seçilecek birey sayısı 4 olarak belirlenmiş olup Şekil 4.7’de en küçük toplam mesafesi olanın seçildiği bir örnek gösterilmektedir.

											Toplam Mesafe
Aday Ebeveyn 1	6	8	4	10	9	5	7	3	2	1	12500
Aday Ebeveyn 2	7	2	5	3	6	4	9	1	8	10	13320
Aday Ebeveyn 3	6	8	4	10	9	5	7	3	2	1	15621
Aday Ebeveyn 4	7	2	5	3	6	4	9	1	8	10	11458 ✓

Şekil 4.7 - Toplam Mesafenin En Kısa Olduğunun Seçildiği Tournament Selection Örneği

Ayrıca, Tournament Selection yöntemi dışında, bu çalışmada, Elitizm belirlenmiş olup bu sayı bir popülasyon içinde en iyi kaç tane bireyin her koşulda bir sonraki nesile aktarılacağını göstermektedir. Örneğin bu çalışmada Elitizm 1 olarak belirlenmiştir ve bu durum, popülasyon içindeki en iyi uygunluk değerine sahip olan 1 adet bireyin yeni nesile aktarılacağını göstermektedir.

4.3.5. Çarpazlama

Ebeveyn seçiminde sonrasında çarpazlama işlemi uygulanmaktadır. Önceki bölümlerde bahsedildiği gibi birden fazla çarpazlama yöntemi bulunmaktadır. Bu projede OX1 (Order Crossover Operator) ve ERX (Edge Recombination Crossover) yöntemleri uygulanmıştır. Her 2 yöntem ile de problem çözüme ulaşmış olmakla birlikte ERX yöntemi belirgin bir şekilde daha uzun sürdüğü için OX1 yöntemi tercih edilmiştir.

OX1 yöntemi aşağıdaki gibi uygulanmış olup Şekil 4.8’de örneklendirilmiştir:

1. Kromozom uzunluk aralığında 2 adet rastgele nokta oluşturularak indeks aralığı olarak belirlenir.
2. Ebeveynlerin 1 tanesinden bu belirlenen noktaların indeks aralığındaki noktaları al ikinci ebeveynde aynı pozisyona yerleştir.
3. İkinci ebeveynin kalan elemanlarını, birinci ebeveynden gelen noktaların son indeksinden başlamayıp başa dönerek al ve yine son indeksten başlayarak yerleştir. Yerleştirirken birinci ebeveynden gelen noktaları atla.

Ebeveyn 1	6	8	4	10	9	5	7	3	2	1
Ebeveyn 2	7	2	5	3	6	4	9	1	8	10
Rastgele Aralık	4-8									
Ebeveyn 1'den Al				10	9	5	7	3		
Yavru	6	4	1	10	9	5	7	3	8	2

Şekil 4.8 - OX1 (Order Crossover Operator) Örnek Uygulama

4.3.6. Mutasyon

Mutasyon fazında üst tarafta bahsedilen yöntemlerden swap (takas) yöntemi kullanılmış olup bu yöntemde kromozom uzunluğu aralığında, birbirinden farklı rastgele 2 indeks oluşturulup bu konumdaki numaraların yerleri değiştirilir, örnek uygulama Şekil 4.9’da gösterilmektedir. Mutasyon işlemi çaprazlama işlemi gibi her defasında yapılmayıp bir oran belirlenerek rastgele üretilen sayıların bu oranı yakalaması sonucunda uygulanan bir adımdır.

Eski Kromozom	6	4	1	10	9	5	7	3	8	2
Yeni Kromozom	6	4	7	10	9	5	1	3	8	2

Şekil 4.9 - Mutasyon Operasyonu Örnek Uygulama

4.3.7. 2-Opt Optimizasyonu

2-opt, yerel arama yaparak verilen nokta dizisinden oluşan rota üzerinde birbiri üzerinden geçen noktaların kesişmelerini ortadan kaldıran basit bir optimizasyon işlemidir. 2-opt işlemi de mutasyon işlemi gibi her defasında değil rastgele oluşturulan sayılarla belirlenen oran yakalandığında rastlantısal uygulanan bir adımdır.

2-opt optimizasyonu ile ilgili diğer detaylar bir önceki Genetik Algoritma bahsinde anlatılmış olup bu çalışmada uygulandığını gösteren kod örneği Şekil 4.10’da gösterilmektedir. “NodeCount” toplam node sayısını göstermekte olup “Swap” isimli fonksiyon gönderilen “NodeNumber” sayı dizisinde i ve k pozisyonundaki sayıların yerlerini değiştiren basit bir takas metodudur.

```
for (int i = 0; i < NodeCount - 1; i++)
{
    for (int k = i + 1; k < NodeCount; k++)
    {
        Dizi içindeki i ve k pozisyonlarındaki noktaların yerini değiştir ve yeni rota oluştur.
        Route newRoute = new Route( Swap(route.NodeNumber, i, k) );
        Yeni rotanın toplam mesafesi eski rotanın toplam mesafesinden küçük ise swap geçerli.
        if (newRoute.TotalCost < route.TotalCost)
        {
            route = newRoute;
        }
    }
}
```

Şekil 4.10 - 2-Opt Algoritmasının Kod Görünümü

4.3.8. Sonlandırma (Termination)

Uygulamanın sonlandırılması ařađıdaki 2 kořula gre yapılmaktadır:

1. Bařlangıçta belirlenen maksimum iterasyon sayısına ulařıldıysa sonlandır.
2. Bařlangıçta belirlenen minimum iterasyon sayısına ulařılmadıysa devam et.
3. Gerçekleřtirilen iterasyon sayısının bařlangıçta belirlenen % deđeri oranında dng boyunca hibir iyileřtirme gerekleřmemesi durumunda.



5. DENEYSEL SONUÇLAR

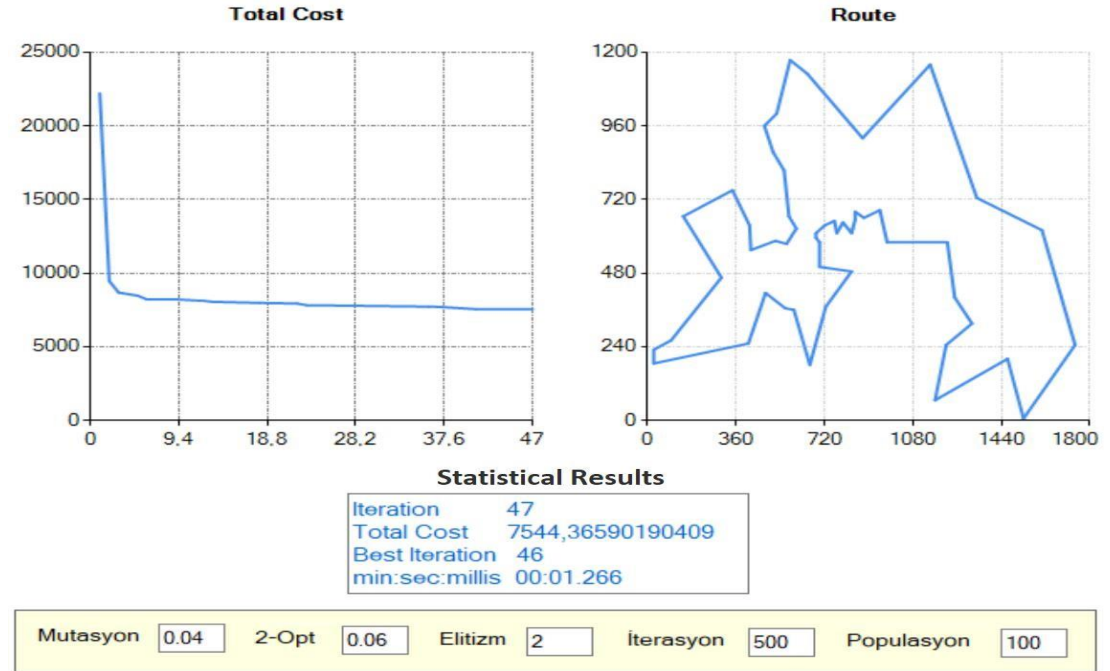
Bu çalışma kapsamında önerilen sistem, Tablo 5.1’de gösterilen özelliklerde bir bilgisayar üzerinde çalıştırılarak test edilmiştir. Sonuçların karşılaştırılabilmesi için bu bilgilerin not edilmesi önem taşımaktadır.

Tablo 5.1 - Çalışmanın Test Edildiği Sistem Özellikleri

Sistem	Özellik
Bilgisayar	Microsoft Surface Pro 3
İşletim Sistemi	Microsoft Windows 10 Pro
CPU	Intel® Core™ i5-4300U Central Processing Unit @ 1.9 GHz
RAM	8 GB
IDE	mSATA 6Gbps Solid State

Geliştirilen sistemin özellikleri, sistemi ölçümlemek için önemli olmakla birlikte, Genetik Algoritma için kullanılan parametreler performansı doğrudan etkilemektedir. Sistem birden fazla nokta seti ve birden fazla parametreyle test edilmiştir.

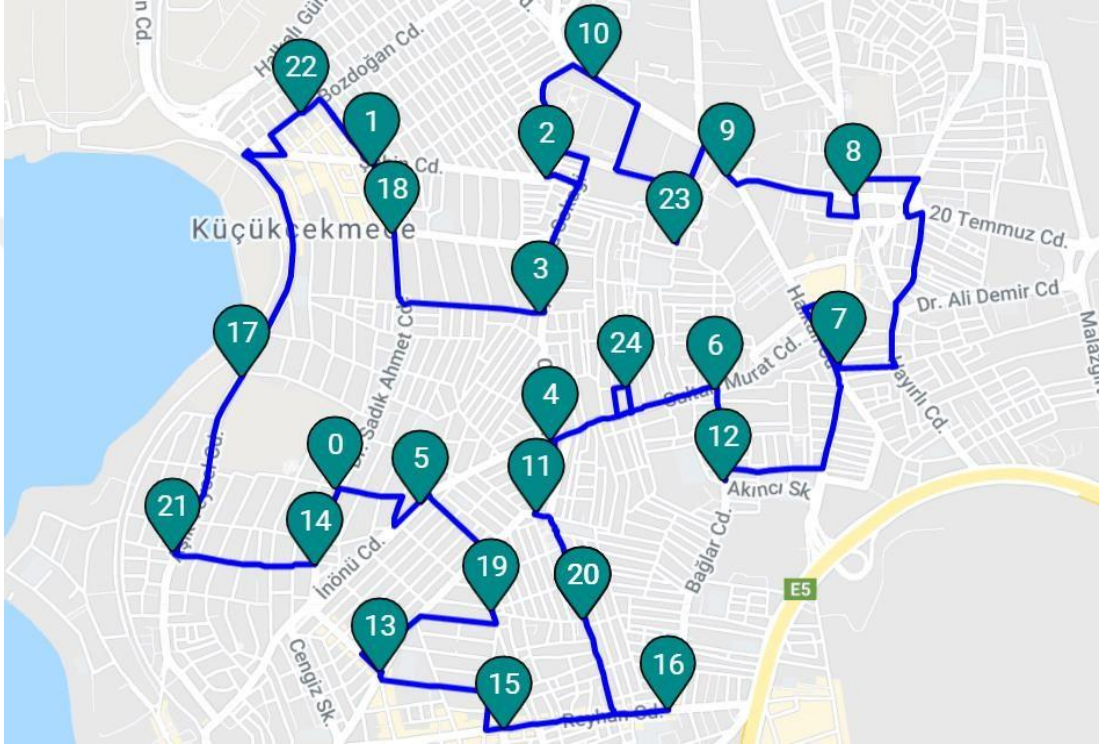
Sistem önce C# masaüstü uygulaması olarak geliştirilmiş ve 52 koordinatlı Berlin.tsp [46] veritabanı ile yapılan test sonucu Şekil 5.1’de paylaşılmıştır.



Şekil 5.1 - Berlin 52 Veritabanı İçin İstatistiksel ve Grafikselsel Sonuçlar

Çalışmanın ikinci fazında, doğrudan ve canlı olarak Google API'leri ile mesafelerin alınıp Genetik Algoritma ile iyileştirilerek elde edilen sonucun web ortamında harita üzerinde gösterilmesi uygulamasına dair detaylı test sonuçları aşağıda gösterilmektedir.

Şekil 5.2 İstanbul/Küçükçekmece bölgesinde rastgele belirlenen çöp konteynırı noktalarının eniyileştirilmiş dolaşı sonucunu harita üzerinde göstermektedir.



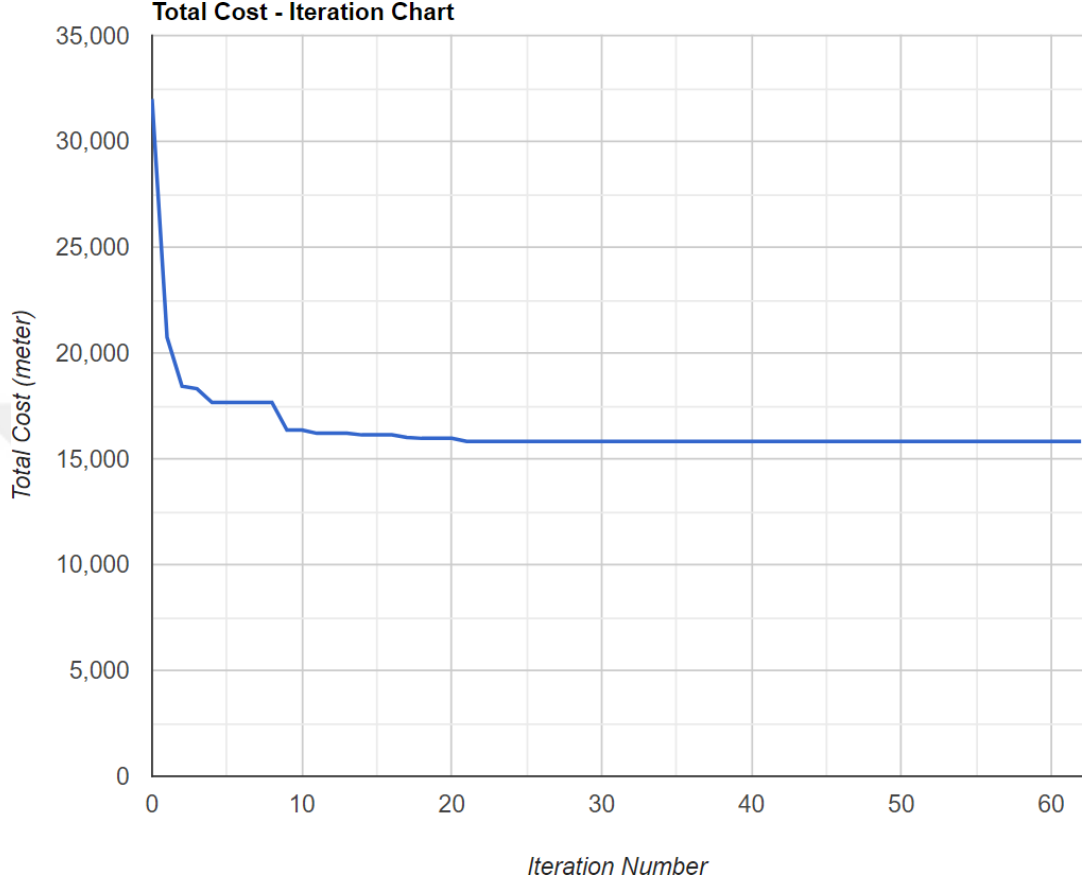
Şekil 5.2 – İstanbul, Küçükçekmece Dolaşı Rotası, Harita Gösterimi

İstanbul/Küçükçekmece bölgesinin eniyileştirilmiş bu dolaşıma ulaşmak için yapılan çalışmada, istatistiksel sonuçların detayları Şekil 5.3'te verilmiştir. Şekil 5.3'te görüldüğü gibi Total Iteration (Toplam İterasyon): 64, Last Best Iteration (En İyi İterasyon): 21, Total Cost (Toplam Mesafe): 15817, Execution Time (Çalışma Zamanı): 169 millis sonuçları elde edilmiştir.

Total Iteration	Last Best Iteration	Total Cost	Execution Time
64	21	15817	min:sec:millis 00:00.169

Şekil 5.3 - İstanbul Küçükçekmece Dolaşı Rotası, İstatistiksel Sonuçlar

İterasyon sayısına göre toplam mesafede yaşanan değişimi gösteren grafik Şekil 5.4'te gösterilmektedir. Grafikte “Total Cost” toplam mesafeyi, “Iteration Number” iterasyon sayısını göstermektedir.



Şekil 5.4 – İstanbul/Küçükçekmece Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği

İstanbul/Küçükçekmece bölgesi için yukarıdaki sonuçları veren değişkenler aşağıda Tablo 5.2’de gösterilmekte olup bu bölge için birçok farklı parametre ile alınan sonuçlar Tablo 5.3’te paylaşılmıştır.

Tablo 5.2 - İstanbul/Küçükçekmece Dolaşı Rotası için Parametreler

Mutasyon Oranı	0,03
2-Opt Oranı	0,06
Elitizm Sayısı	1
Popülasyon Sayısı	100
İterasyon Sayısı	300
İterasyon Sonlandırma Yüzdesi	200

Tablo 5.3 - Farklı Parametreler ile Yapılan Test Sonuçları

#	Popülasyon	Koordinat Sayısı	Mutasyon Oranı (%)	2-Opt Oranı (%)	Elitizm	Toplam İterasyon	En İyi İterasyon	Çalışma Süresi (milisaniye)	Sonuç (metre)
1	25	25	4	4	2	125	109	77	15817 ✓
2	25	25	4	4	1	125	92	43	15817 ✓
3	25	25	4	6	2	125	109	53	16092 —
4	25	25	4	6	1	125	98	53	15965 —
5	25	25	3	4	2	121	40	55	15965 —
6	25	25	3	4	1	125	58	80	15817 ✓
7	25	25	3	6	2	125	49	52	15817 ✓
8	25	25	3	6	1	82	27	64	16092 —
9	50	25	4	4	2	250	220	186	15817 ✓
10	50	25	4	4	1	106	35	119	16500 —
11	50	25	4	6	2	250	129	323	15817 ✓
12	50	25	4	6	1	250	148	218	15817 ✓
13	50	25	3	4	2	250	180	171	15817 ✓
14	50	25	3	4	1	250	161	174	15817 ✓
15	50	25	3	6	2	181	60	153	15817 ✓
16	50	25	3	6	1	244	81	229	15817 ✓
17	100	25	4	4	2	91	30	147	15817 ✓
18	100	25	4	4	1	300	113	484	15817 ✓
19	100	25	4	6	2	193	64	342	15817 ✓
20	100	25	4	6	1	120	40	207	15817 ✓
21	100	25	3	4	2	88	29	136	15817 ✓
22	100	25	3	4	1	76	25	156	15817 ✓
23	100	25	3	6	2	82	27	205	15817 ✓
24	100	25	3	6	1	64	21	169	15817 ✓

Tablo 5.3'teki sütunları aşağıdaki özellikleri temsil etmektedir:

- 1. sütun, sıra numarası,
- 2. sütun, popülasyon sayısı,
- 3. sütun, dolaşılacak toplam koordinat noktası sayısı,
- 4. sütun, mutasyonun gerçekleştirilme oranı,
- 5. sütun, 2-opt işleminin gerçekleştirilme oranı,

- 6. sütun, elitizm özelliği ile üst nesile geçirilecek birey sayısını,
- 7. sütun, gerçekleşen toplam iterasyon sayısını,
- 8. sütun, en son iyileştirme yapılan iterasyon sayısını,
- 9. sütun, toplam çalışma süresini,
- 10. sütun, metre cinsinden bulunan en kısa yolu.

Tablo 5.3'teki verilere göre gerçekleştirilen toplam iterasyon sayısı, sonuca ulaşılan iterasyon sayısı veya en kısa çalışma zamanı ile ilgili farklı satırlarda en iyi sonuçların elde edildiği görülmektedir. Popülasyon sayısı 25 iken en iyi sonuca en kısa sürede ulaşıldığı görülmektedir. Popülasyon 50 iken çalışma zamanı ve iterasyon sayılarının çok fazla olduğu görülmektedir. Popülasyon 100 iken ise bütün denemelerde en iyi sonuca ulaşıldığı görülmektedir.

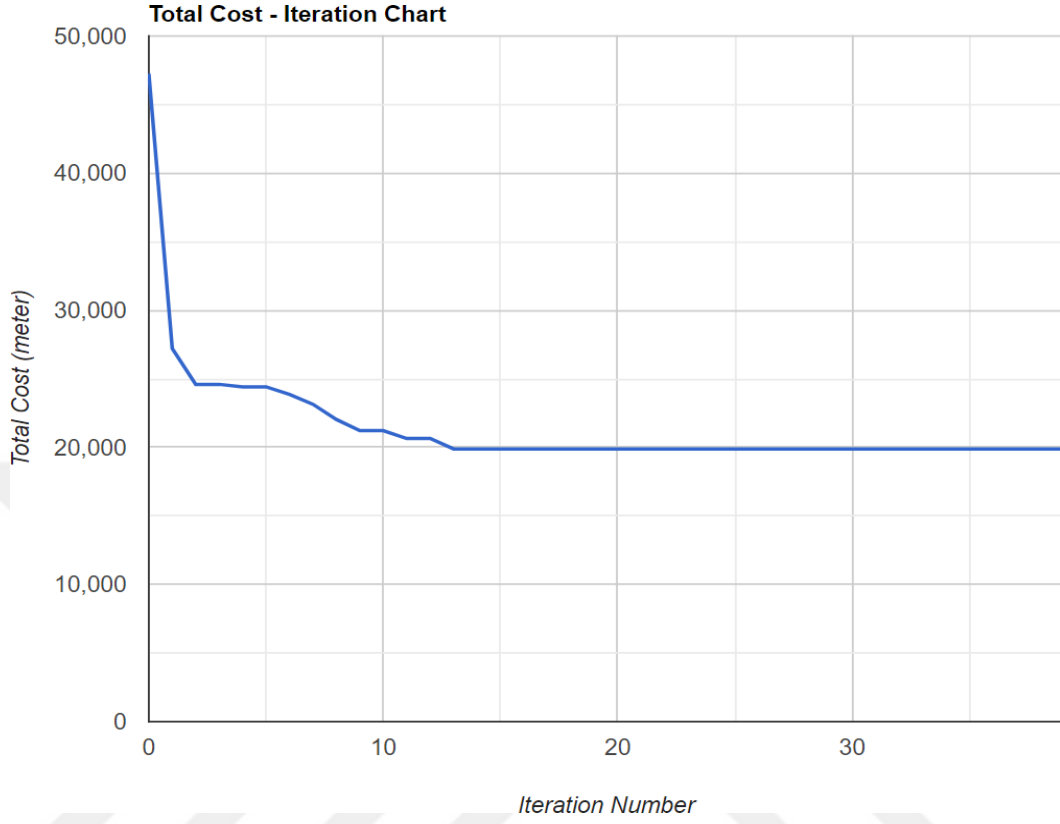
İstanbul/Küçükçekmece bölgeleri dışında İstanbul/Yenibosna ve İstanbul/Bağcılar bölgelerinde de 25'er nokta belirlenmiş olup bu noktalara ait çıktılar aşağıda gösterilmiştir.

Şekil 5.5 İstanbul/Bağcılar bölgesinde belirlenen koordinatlara ait dolaşımın optimize edilmiş rotasının harita görünümünü sunmaktadır.



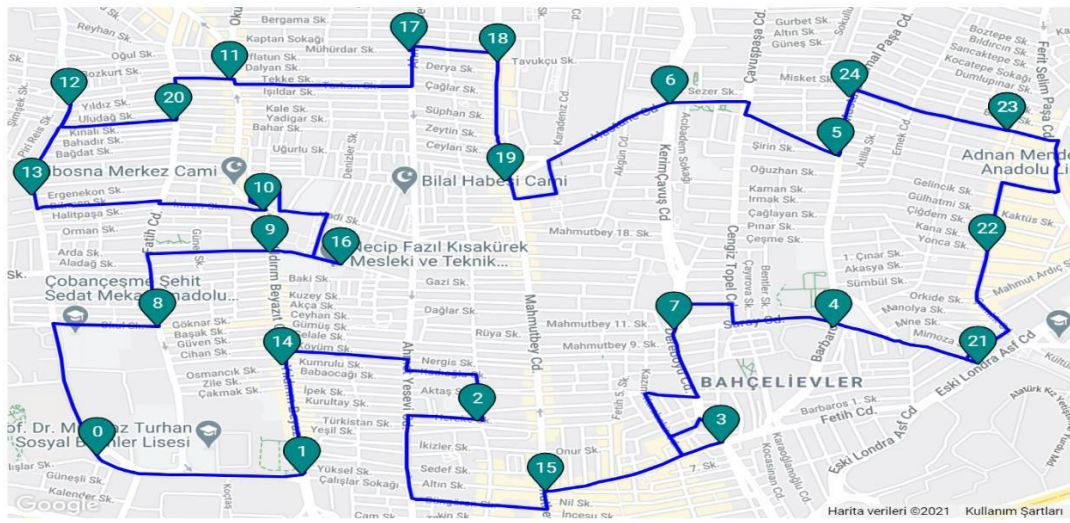
Şekil 5.5 - İstanbul/Bağcılar Dolaşı Rotası, Harita Gösterimi

Şekil 5.6 İstanbul/Bağcılar bölgesinin dolaşı rotası eniyilemesi çalışmasında iterasyona bağlı toplam mesafenin değişimini göstermektedir.



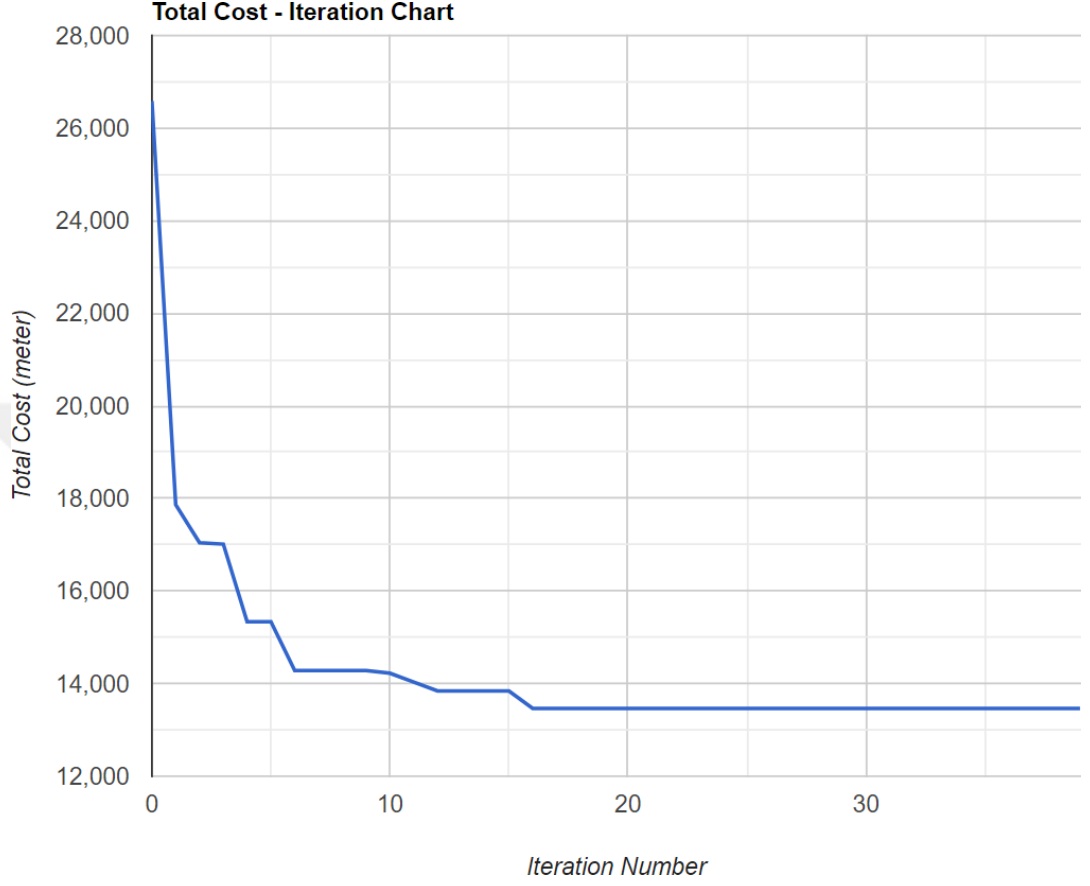
Şekil 5.6 - İstanbul/Bağcılar Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği

İstanbul/Yenibosna bölgesine ait dolaşının harita gösterimi Şekil 5.7’de sunulmuştur.



Şekil 5.5 - İstanbul/Bağcılar Dolaşı Rotası, Harita Gösterimi

İstanbul/Yenibosna bölgesinde belirlenen koordinatlara ait dolaşı rotası eniyilemesi çalışmasında iterasyona bağlı toplam mesafenin değişimi Şekil 5.8’de gösterilmektedir.



Şekil 5.8 - İstanbul/Bağcılar Dolaşı Rotası, İterasyon-Toplam Mesafe Grafiği

En iyi ve en kötü sonuçlar kıstas alınan sonuç parametresine (Toplam İterasyon, En İyi İterasyon, Çalışma Süresi) göre değişebilir. Farklı parametreler ile daha fazla test yaparak farklı sonuçları değerlendirmekte fayda vardır. Buna rağmen grafiklerden açıkça görülmektedir ki, önerilen yöntem hızlı ve kabul edilebilir sonuçlara ulaşmada başarılı olmuştur.

6. SONUÇLAR

Son yıllarda akıllı şehir kavramı, neredeyse tüm gelişmekte olan ve gelişmiş ülkelerde giderek artan bir ilgi görmektedir. Akıllı şehirlerdeki kritik problem, atık toplama ve yönetimi yaklaşımları ile ilgilidir. Bu yazıda, verimli bir atık toplama mekanizması önerilmiştir. Öncelikle sensöre dayalı bir sistem kullanılarak kutulardaki atık miktarı ölçülmekte ve bu bilgiler bir sunucuya gönderilmekte, ardından minimum maliyetle çöp toplama kamyonları için genetik algoritma tabanlı bir yol planlanmaktadır.

Çalışmamızda İstanbul ilinin farklı bölgelerinde bulunan çöp konteynırlarının doluluk durumlarına göre toplanma listesine alınması ve rota oluşturulması uygulaması geliştirilip testleri yapılmıştır. Rota oluşturulurken Google Maps Distance Matrix API servisi ile mesafeler canlı olarak alınıp Google Maps Directions API rota anlık olarak oluşturulmuştur. Uygulamamız C# dilinde ve .Net Framework'ünde geliştirilmiş olup sonuçlar web ortamında görselleştirilmiştir. Rota çıktısı Google Maps Javascript API servisi entegrasyonu ile harita üzerinde gösterilmiştir.

Şehirlerin, özellikle de büyük şehirlerin yalnızca atık değil kentselleşmenin artmasıyla çok sayıda sorunu bulunmaktadır. Çalışmanın devamında, nokta sayısının artması durumunda paralel hesaplama yöntemi ile daha hızlı sonuçlar elde etmek üzere çalışma geliştirilebilir. Ayrıca bahsedilen sorunların çözümüne katkı için yapılan bu çalışmanın, gelecek dönemde çoklu araç rotalama (MTSP), diğer araçlarla haberleşme, trafik yoğun saatlerin tespiti ile uygun zamanda çıkma vb. iyileştirmeler ile Akıllı Şehirler oluşturma projesine daha fazla katkıda bulunabilecektir.

7. KAYNAKLAR

- [1] F. Anindra, D. M. Min and H. L. H. S. Warnars, Smart City Implementation Modelling in Indonesia with Integration Platform Approach, International Conference on Information Management and Technology (ICIMTech), Jakarta, Indonesia, 2018.
- [2] C. Sađırođlu, Akıllı Őehir Stratejisi ve Eylem Planı, [Online].Eriřilebilirlik: https://yte.bilgem.tubitak.gov.tr/sites/images/bilgem/tubitak_bilgem_yte_akillisehirlersunumu_301117.pdf, [Eriřim Tarihi Aralık 2020].
- [3] R. Fujdiak, P. Masek, P. Mlynek, J. Misurec and E. Olshannikova, Using genetic algorithm for advanced municipal waste collection in Smart City, 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Prague, 2016.
- [4] D. Hoornweg and P. Bhada-Tata, What a Waste : A Global Review of Solid Waste Management, Urban development series; pp. 15, World Bank, Washington, DC., 2012.
- [5] European Commission, Science for Environment Policy, [Online].Eriřilebilirlik: https://ec.europa.eu/environment/integration/research/newsalert/pdf/ecosystem_services_biodiversity_IR11_en.pdf , [Eriřim Tarihi Haziran 2020].
- [6] V. Catania and D. Ventura, An Approach for Monitoring and Smart Planning of Urban Solid Waste Management Using Smart-M3 Platform, Proceedings of 15th Conference of Open Innovations Association FRUCT, St. Petersburg, 2014.
- [7] A. Caragliu, C. Del Bo and P. Nijkamp, Smart Cities in Europe, Journal of Urban Technology, pp. 65-82, [Eriřim Tarihi 10 Ađustos 2020].
- [8] R. Hollands, Will the Real Smart City Please Stand Up?, Newcastle University, [Online].Eriřilebilirlik: https://www.researchgate.net/publication/248930334_Will_the_Real_Smart_City_Please_Stand_Up, [Eriřim Tarihi Temmuz 2020].

- [9] A.C. Mendes and A. Henriques: Smart cities precisam de apoios., [Online].Erişilebilirlik: <https://www.jornaldenegocios.pt/negocios-iniciativas/cidades-inteligentes/detalhe/almeida-henriques-smart-cities-precisam-de-apoios>, [Erişim Tarihi Haziran 2020].
- [10] M. A. Mohammed, M. K. A. Ghani, R. Hamed, S. Mostafa, D. A. Ibrahim, H. K. Jameel and A. H. Alallah, Solving Vehicle Routing Problem by Using Improved Genetic Algorithm for Optimal Solution, Journal of Computational Science, vol. 21, pp. 255-262, 2017.
- [11] R. Arakaki and F. L. Usberti, Hybrid Genetic Algorithm for the Open Capacitated Arc Routing Problem, Computers & Operations Research, vol.90, pp. 221-231, 2017.
- [12] S. Wøhlk, A Decade of Capacitated Arc Routing, Part of the Operations Research/Computer Science Interfaces book series, pp. 29-48, 2008.
- [13] P. Lacomme, C. Prins and W. Ramdane, A Genetic Algorithm for the Capacitated Arc Routing Problem and Its Extensions, Part of the Lecture Notes in Computer Science book series, LNCS, vol. 2037 , pp. 473-483, 2001.
- [14] H. Han and E. Ponce-Cueto, Waste Collection Vehicle Routing Problem: A Literature Review, [Online].Erişilebilirlik: https://www.researchgate.net/publication/282415372_Waste_Collection_Vehicle_Routing_Problem_A_Literature_Review/link/570644e208ae0f37fee19ea5/download/, [Erişim Tarihi Ağustos 2020].
- [15] J. Evans and E. Minieka, Optimization Algorithms for Networks and Graphs. Second Editions, Revised and Expanded, [Online].Erişilebilirlik: https://www.researchgate.net/publication/327765806_Optimization_algorithms_for_networks_and_graphs_Second_edition_revised_and_expanded/, [Erişim Tarihi Ağustos 2020].
- [16] A. P. Punnen and G. Gutin, The Travelling Salesman Problem: Applications, Formulations and Variations, Kluwer Academic Publishers, 2002.

- [17] N. Christofides, Technical Note-Bounds for the Travelling-Salesman Problem, *Operations Research*, 20 (5), 1044-1056, 1972.
- [18] R. M. Karp and M. Held, The Traveling Salesman Problem and Minimum Spanning Trees, *Operations Research*, c. 18, pp. 1138-1162, 1970.
- [19] G. Rinaldi and M. Padberg, Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut, *Operations Research Letters*, c. 6, pp. 1-7, 1987.
- [20] K. Alaykırın ve O. Engin, Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 20(1), sf. 69-76, 2005.
- [21] M. L. Fisher, Optimal Solution of Vehicle Routing Problems using Minimum k-trees, *Operations Research*, 42(4), 626-642, 1994.
- [22] G. Dantzig, S. Johnson and R. Fulkerson, Solution of a Large-Scale Travelling Salesman Problem, *Journal of Operations Research Society*, c.2, pp. 393-410, 1954.
- [23] G. Clarke and J. W. Wright, Scheduling of Vehicles From a Central Depot to a Number of Delivery Points, *Operations Research*, 12(4), 568-581, 1964.
- [24] C. Gencer ve M. Eryavuz, Araç Rotalama Problemine Ait Bir Uygulama, *Süleyman Demirel Üniversitesi, İktisadi ve İdari Bilimler Fakültesi*, 6(1), 139-155, 2001.
- [25] T. Eren, H. Güvez ve M. Dege, Kırıkkale’de Araç Rotalama Problemi İle Tıbbi Atıkların Toplanması, *International Journal of Engineering Research and Development*, 4(1), 41-45, 2012.
- [26] K. Çalışkan, Karınca Kolonisi Optimizasyonu İle Araç Rotalama Probleminin Maliyetlerinin Kümeleme Tekniği İle İyileştirilmesi, *TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi*, 84s, Ankara, 2011.

- [27] O. Çetin, Akaryakıt Dağıtımında Araç Rotalama Problemi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, 185s, İstanbul, 2013.
- [28] S. Longhi, D. Marzioni, E. Alidori, G. D. Buo, M. Prist, M. Grisostomi and P. Matteo, Solid Waste Management Architecture Using Wireless Sensor Network Technology, 5th International Conference on New Technologies, Mobility and Security (NTMS), Istanbul, 2012.
- [29] S. Chaudhari and V. Bhole, Solid Waste Collection as a Service using IoT solution for Smart Cities, IEEE International Conference on Smart City and Emerging Technologies ICSCET, Mumbai, January 2018,
- [30] J. M. Gutierrez, M. Jensen, M. Henius and T. Riaz, Aalborg University, Fredrik Bajers vej 7, 9220 Aalborg Ø, Denmark, NetPlan A/S, Hedeager 3, 8200 Aarhus N, Denmark, RadioAnalyzer, Algade 15, 9000 Aalborg, Denmark, 2015.
- [31] M. V. B. Delgado, J. L. R. Gazquez, P. Jimenez and P. Pavon-Marino, Optimal Path Planning for Selective Waste Collection in Smart Cities, [Online].Erişilebilirlik: https://www.researchgate.net/publication/332727478_Optimal_Path_Planning_for_Selective_Waste_Collection_in_Smart_Cities/link/5cc70667299bf12097883770/download/, [Erişim Tarihi Nisan 2020].
- [32] S. Idwan, I. Mahmood, J. A. Zubairi and I. Matar, Optimal Management of Solid Waste in Smart Cities using Internet of Things, Springer Science+Business Media, LLC, Wireless Personal Communications, Issue 1/2020, 2020.
- [33] M. Mitchell, Genetic algorithms: An overview. Complexity, 1(1), 31–39, [Online].Erişilebilirlik: https://iustlive.com/Notifications/Department/DOEE_160620055411.pdf, [Erişim Tarihi Temmuz 2020]
- [34] J. H. Holland, Genetic Algorithms, Scientific American, vol. 267, No. 1, pp. 66-73, [Online].Erişilebilirlik: <https://www.jstor.org/stable/24939139>, [Erişim Tarihi Temmuz 2020].

- [35] J. Carr, An Introduction to Genetic Algorithms, Whitman College Mathematics Department, Computer Science, [Online].Erişilebilirlik: <https://www.semanticscholar.org/paper/An-Introduction-to-Genetic-Algorithms-Carr/e9f8d49686a4c8d99d0a5ceba85c4508c30d57c4>, [Erişim Tarihi Mayıs 2020].
- [36] Ç. Taşkin ve G. G. Emel, Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi, [Online].Erişilebilirlik: <https://www.researchgate.net/publication/312447324>, [Erişim Tarihi Nisan 2020].
- [37] D. Beasley, D. Bull, and R. Martin, An overview of Genetic Algorithms: Pt1, Fundamentals, University of Bristol, vol.15, pp. 58-69, [Online].Erişilebilirlik: <https://research-information.bris.ac.uk/en/publications/an-overview-of-genetic-algorithms-pt1-fundamentals>, 1993.
- [38] I. M. Oliver, D. J. Smith, and J. R. C. Holland, A Study of Permutation Crossover Operators on the TSP, Proceedings of the 2nd International Conference on Genetic Algorithms on Genetic Algorithms and their Application, pp. 224-230, 1987.
- [39] T. D. Gwiazda, Genetic Algorithms Reference, Volume I Crossover for Single-Objective Numerical Optimization Problems, [Online].Erişilebilirlik: http://www.tomaszgwiazda.com/Genetic_algorithms_reference_first_40_pages.pdf, [Erişim Tarihi Mayıs 2020].
- [40] L. Davis, Applying Adaptive Algorithms to Epistatic Domains, Proceedings of the 9th International Joint Conference on Artificial Intelligence, vol. 1, pp. 162-164, 1985.
- [41] L. D. Whitley, T. Starkweather and D. Fuquay, Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator, Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 133-140, 1989.
- [42] G. G. Mitchell, Evolutionary Computation Applied to Combinatorial Optimisation Problems, Ph.D. Thesis, School of Electronic Engineering, Dublin City University, 2007.

- [43] G. A. Croes, A Method for Solving Traveling Salesman Problems, Operations Res. 6, pp. 791-
- [44] M. M. Flood, The Travelling Salesman Problem, Operations Res. 4, pp. 61-67, [Online].Erişilebilirlik: <https://doi.org/10.1287/opre.4.1.61>, [Erişim Tarihi Şubat 2020].
- [45] M. Vivet, vivet/GoogleApi, Github, 2018.
- [46] berlin52.tsp, - 52 locations in Berlin, Germany, [Online].Erişilebilirlik: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp>, [Erişim Tarihi Mayıs 2020]

