

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

KATI ATIK TOPLAMA ROTA OPTİMİZASYONU:
ÜMRANIYE İLÇESİ ÖRNEĞİ

YÜKSEK LİSANS TEZİ

UFUK DEREÇİ

Ocak 2022

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

KATI ATIK TOPLAMA ROTA OPTİMİZASYONU:
ÜMRANIYE İLÇESİ ÖRNEĞİ

YÜKSEK LİSANS TEZİ

UFUK DEREÇİ

DANIŞMAN

DR. ÖĞR. ÜYESİ M. ERKAN KARABEKMEZ

Ocak 2022

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.

Ufuk Dereci

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığını taahhüt ederim.

Dr. Öğr. Üyesi M. Erkan Karabekmez

İMZA SAYFASI

Ufuk Dereci tarafından hazırlanan ‘Katı Atık Toplama Rota Optimizasyonu: Ümraniye İlçesi Örneği’ başlıklı bu yüksek lisans tezi, Mühendislik Yönetimi Anabilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ

İMZA

Tez Danışmanı:

Dr. Öğr. Üyesi M. Erkan Karabekmez

.....

Kurumu: İstanbul Medeniyet Üniversitesi

Üyeler:

Doç. Dr. Ali Osman Kuşakçı

.....

Kurumu: İbn Haldun Üniversitesi

Dr. Öğr. Üyesi A. Salih Aydın

.....

Kurumu: İstanbul Medeniyet Üniversitesi

Tez Savunma Tarihi: 27/01/2022

ÖNSÖZ

Bu çalışmada katı atık rota optimizasyonu ve araç rotalama problemi konularını bir bütün olarak ele alarak literatürde sıklıkla kullanılan yöntemleri inceledim ve Ümraniye ilçesi özelinde çok büyük boyutlu bir gerçek hayat problemine çözüm aradım.

Tez çalışmam boyunca desteğini esirgemeyen, bilgi ve tecrübeleriyle beni yönlendiren kıymetli danışman hocam Dr. Öğr. Üyesi M. Erkan KARABEKMEZ'e teşekkürü borç bilirim.

Ayrıca veri temini ve bilgi toplama konusunda gösterdikleri destekten ötürü Ümraniye Belediyesi ve çalışanlarına da teşekkür ederim.

ÖZET
KATI ATIK TOPLAMA ROTA OPTİMİZASYONU:
ÜMRANİYE İLÇESİ ÖRNEĞİ

Ufuk Dereci

Yüksek Lisans Tezi, Mühendislik Yönetimi Anabilim Dalı

Danışman: Dr. Öğr. Üyesi M. Erkan Karabekmez

Ocak, 2022. 72 Sayfa.

Büyük şehirlerde nüfusun kalabalık, atık miktarının yüksek olması konteynerlara gidecek çöp araçlarının rotalarının belirlenmesini zorlaştırmaktadır. Doğru planlanmayan rotalar çöp toplama sürecinde gereğinden fazla yakıt, zaman ve kaynak harcanmasına sebep olur.

Araç Rotalama Problemi (ARP) belirli sayıda aracın, bir depodan başlayıp tüm müşterileri ziyaret edecek rotalarının bulunmasını amaçlayan kombinatoriyal optimizasyon problemidir. Bu çalışmada İstanbul Ümraniye ilçesinin katı atık toplama süreci ARP olarak ele alınarak tüm ihtiyacı karşılayacak en kısa rotaların oluşturulması amaçlanmıştır. Belediyeden hâlihazırda kullanılan rotalar, her bir rotanın günlük tonaj ve sefer sayıları verisi ve ilçe sınırları içerisinde yer alan tüm cadde ve sokaklardaki konteyner adet verisi alınmıştır.

Çözümde literatürde sıklıkla kullanılan En Yakın Komşu, Tasarruf, Christofides, Benzetimli Tavlama, Açgözlü İniş, Rehberli Yerel Arama ve Tabu Arama algoritmaları gibi sezgisel ve meta-sezgisel algoritmalar kullanılmıştır. Ayrıca Önce Kümele Sonra Rotala prensibine dayanan bir çözüm de incelenmiştir. Çalışma sonucunda incelenen sezgisel ve meta-sezgisel yöntemlerin devasa boyutlu gerçek hayat problemlerinde de hızlı ve olurlu sonuçlar üretebildiği, kümeleme algoritmalarının ARP'nin çözümünde etkili bir araç olabileceği gösterilmiştir.

Anahtar Kelimeler: Katı Atık Toplama Rota Optimizasyonu, Araç Rotalama Problemi, Sezgisel Algoritmalar, Önce Kümele Sonra Rotala, K-Ortalamalar Kümeleme Algoritması.

ABSTRACT
SOLID WASTE COLLECTION ROUTE OPTIMIZATION:
ÜMRANIYE MUNICIPALITY CASE

Ufuk Dereci

Master Thesis, Department of Engineering Management

Supervisor: Asst. Prof. Dr. M. Erkan Karabekmez

January, 2022. 72 Pages.

In big cities, the crowded population and high amount of waste make it difficult to determine the routes of the garbage trucks that will go to the containers. Moreover incorrectly planned routes cause waste of fuel, time and other resources in the garbage collection process.

Vehicle Routing Problem (VRP) is a combinatorial optimization problem that aims to find optimal routes of a certain number of vehicles that start from depot, visit all customers and return to the depot. In this study, it is aimed to find the shortest routes that will satisfy all the needs by considering waste collection process of İstanbul Ümraniye district as VRP. Data of the currently used routes, daily tonnages, number of truck trips and the number and location of containers in Ümraniye district were obtained from the municipality.

In the solution, popular heuristic and meta-heuristic algorithms such as Nearest Neighbor, Savings, Christofides, Simulated Annealing, Greedy Descent, Guided Local Search and Tabu Search algorithms are used. Furthermore, a “Cluster First, Route Second” principle based method is also examined. It has been shown that examined heuristics and meta-heuristics can produce fast and feasible solution in huge size real-life problem and clustering algorithms can be effective tool for solving VRP.

Keywords: Solid Waste Collection Route Optimization, Vehicle Routing Problem, Heuristic Algorithms, Cluster First Route Second, K-Means Clustering Algorithm.

İÇİNDEKİLER

ÖNSÖZ.....	v
ÖZET.....	vi
ABSTRACT	vii
İÇİNDEKİLER.....	viii
TABLolar LİSTESİ	x
ŞEKİLLER LİSTESİ.....	xi
BÖLÜM 1. Giriş	1
1.1 Araştırma amacı	2
1.2 Araştırma Motivasyonu.....	3
1.3 Araştırma kapsamı.....	3
1.4 Tezin yapısı	4
BÖLÜM 2. Literatür İncelemesi	5
2.1 Araç Rotalama Problemi	5
2.1.1 ARP Türleri	8
2.2 Atık Toplama Optimizasyonu ve Rotalama	9
2.3 K-Ortalamlar Kümeleme Yöntemi	11
BÖLÜM 3. Araştırma Metodolojisi	14
3.1 Çözüm Yöntemleri	14
3.1.1 Kesin Yöntemler	14
3.1.2 Sezgisel ve Meta-sezgisel Yöntemler	17
3.1.3 Kümeleme Algoritmaları (K-Ortalamlar Kümeleme Algoritması)	19
3.2 Önerilen Yöntemler.....	20
3.2.1 Kullanılan Yazılım ve Paket Programlar	21

3.2.2 Verilerin Alınması ve Genel Düzenlemeler	22
3.2.3 Senaryo 1: Yerel Maks – Optimal Çözüm Yaklaşımı.....	26
3.2.4 Senaryo 2: Yerel Maks– Vatandaş Memnuniyetini Gözeten Yaklaşım....	27
3.2.5 K-Ortalamlar ve OR-Tools Sıralı Çözüm	29
BÖLÜM 4. Uygulama – Ümraniye Belediyesi Örneği.....	32
4.1 Verilerin Temini ve Veri Temizleme Önışlemleri	32
4.1.1 Veri setinin düzenlenmesi.	33
4.1.2 Geopy ile koordinatların belirlenmesi.....	34
4.2 Senaryo 1: Yerel Maks – Optimal Çözüm Yaklaşımı.....	35
4.3 Senaryo 2 – Yerel Maks Vatandaş Memnuniyeti Odaklı Yaklaşım	39
4.4 K-Ortalamar ve OR-Tools Sıralı Çözüm	43
BÖLÜM 5. Sonuç ve Öneriler	49
KAYNAKÇA.....	53
EK-1 Alınan Güzergah Verileri	61
EK-2 Alınan Güzergah Tonaj Verileri.....	62
EK-3 Lokasyonlar Arası Karşılıklı Mesafe Matrisi	63

TABLÖLAR LİSTESİ

Tablo 1. Çalışmada Kullanılan OR-Tools Çalıştırma Seçenekleri	21
Tablo 2. Alınan Güzergah Verileri Formatı	32
Tablo 3. Alınan Güzergah Tonaj Verileri Formatı.....	33
Tablo 4. Güzergahlara Ait Tanımlayıcı İstatistikler (Kısaltılmış Örnek Tablo)	33
Tablo 5. Güzergahlara Ait Tanımlayıcı İstatistikler (Düzenlenmiş, Kısaltılmış Örnek Tablo)	34
Tablo 6. Koordinatların Belirlenmesi Kodu Girdi Çıktı Tablosu (Kısaltılmış Örnek Tablo)	34
Tablo 7. Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Senaryo 1	36
Tablo 8. Lokasyonlar Arası Karşılıklı Mesafe Matrisi (Kısaltılmış)- Senaryo 1	36
Tablo 9. Talep Matrisi (Kısaltılmış)- Senaryo 1	37
Tablo 10. OR-Tools Çalıştırma Sonuçları - Senaryo 1	37
Tablo 11 Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Senaryo 2	40
Tablo 12 Lokasyonlar Arası Karşılıklı Mesafe Matrisi (Kısaltılmış)	40
Tablo 13 Talep Matrisi (Kısaltılmış)- Senaryo 2	41
Tablo 14 OR-Tools Çalıştırma Sonuçları - Senaryo 2	41
Tablo 15 Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Sıralı Çözüm	44
Tablo 16 K-Ortalamlar Küme Alt ve Üst Limitleri.....	45
Tablo 17 K-Ortalamlar Çalıştırma Sonuçları	46
Tablo 18 Deney 3’e ait Küme Eleman Sayıları.....	46
Tablo 19 Elle İyileştirme Sonrasında Deney 3 Küme Eleman Sayıları	46
Tablo 20 K-Ortalamlar Sonrası OR-Tools Çözüm Sonucu.....	47
Tablo 21 OR-Tools Çalıştırma Sonuçları - Özet.....	50

ŞEKİLLER LİSTESİ

Şekil 1. Tez Akış Şeması	20
Şekil 2. Konteyner Başına Düşen Ortalama Çöp Miktarı (kg/gün)	24
Şekil 3. Konteyner Başına Ortalama Çöp Miktarı (kg/gün) Optimal Senaryo	25
Şekil 4. Konteyner Başına Ortalama Çöp Miktarı (kg/gün) V. M. Senaryosu	25
Şekil 5. Yerel Maks Mahalle Konteyner Başına Talep Katsayısı	35
Şekil 6. En İyi Deneye Ait Tüm Araçların Güzergah Haritası - Senaryo 1	38
Şekil 7. En İyi Deneye Ait 8. Nolu Aracın Güzergah Haritası – Senaryo 1	39
Şekil 8 En İyi Deneye Ait Tüm Araçların Güzergah Haritası - Senaryo 2	42
Şekil 9 En İyi Deneye Ait 17. Nolu Aracın Güzergah Haritası – Senaryo 2	43
Şekil 10 Mahalle Konteyner Başına Talep Katsayısı – Sıralı Çözüm	44
Şekil 11 İlçe Sokaklarının Küme Haritası.....	47

BÖLÜM 1. GİRİŞ

Teknolojinin gelişmesi, hızla artan kentleşme ve olağanüstü nüfus artışı ile ülkemizde ve tüm dünyada insanoğlunun çevreye bıraktı negatif etki inanılmaz boyuta gelmiştir. Başlangıçta kısıtlı olan kaynakları en iyi şekilde kullanma amacı zamanla bu kaynaklardan ortaya çıkan atıkları da en etkili şekilde değerlendirme amacına dönüşmüştür. Atık kavramı ilk olarak Çevre Kanunu'nda, "Herhangi bir faaliyet sonucunda çevreye atılan veya bırakılan zararlı maddeler" şeklinde tanımlamıştır (1983). Atık Yönetimi Yönetmeliği'nde ise "Üreticisi veya fiilen elinde bulunduran gerçek veya tüzel kişi tarafından çevreye atılan veya bırakılan ya da atılması zorunlu olan herhangi bir madde veya materyal" olarak daha detaylı bir şekilde tanımlanmıştır (2015).

Atıklar genel olarak katı, sıvı, gaz ve ambalaj atıkları şeklinde sınıflandırılır ve kendi içlerinde üretildikleri yere veya atığın niteliğine göre alt sınıflara ayrılırlar. Bu atıkların zararlarının üretim aşamasında önlenmesi veya en aza indirgenmesi, atık haline geldikten sonra toplanıp yeniden kullanılması, kullanılamayacak atıkların geri dönüştürülmesi, geri dönüştürülemeyecek atıkların imha edilmesi atık yönetiminin temel amaçlarındandır.

Ülkemizde atıkların toplanması hizmeti yerel yönetimlerin ve belediyelerin sorumluluğundadır. Özellikle büyük şehirlerde nüfusun kalabalık, sanayinin gelişmiş olmasıyla beraber atık miktarı yüksektir. 963 mahallesi bulunan İstanbul'da günlük ortalama 18 bin ton evsel atık oluşmaktadır (İstanbul Büyükşehir Belediyesi, n.d.). İlçe belediyeleri bu toplamayı yaparken çevre kirliliğini azaltmayı, vatandaşlarının memnuniyet seviyesini artırmayı amaçlar ve bu toplamayı en az maliyetle yapmayı hedefler. Bunun için kaynaklarını doğru bir şekilde planlar. Planda yapılacak herhangi hata sonucunda çöpler/atıklar konteynerlerden taşarak sokakları kirletir ve

evre kirliliğine sebep olur. Sorumlu olduėu hizmeti saėlayamayan belediye, vatandaşlarını mağdur ederek memnuniyetsizliğe sebep olur. Öte yandan hatalı bir planla çöpler düzgün bir şekilde toplanabilir ancak gereğinden fazla kaynak (personel, araç, yakıt, zaman, vd.) harcanarak israfa sebep olunabilir. Dolayısıyla bu planlanmanın yapılması kapsamlı hareket edilmesi gereken, çeşitli kısıtları olan önemli bir mühendislik problemi olarak ele alınmalıdır.

Araç rotalama problemi (ARP) literatürde çok sık çalışılan bir optimizasyon problemidir. Literatürde ilk defa 1959 yılında Dantzig ve Ramser tarafından çalışılmıştır. En temel haliyle belirli sayıda aracın, ana depodan başlayarak tüm depolara uğradıktan sonra ana depoya geri döndüğü rotaların minimum maliyetle belirlenmesidir (Dantzig & Ramser, 1959). Maliyet genellikle kat edilen mesafedir. Problemdeki kısıtlara göre ARP yeniden adlandırılarak özel problemlere dönüşebilir. (Kapasiteli Araç Rotalama Problemi (KARP), Zaman Pencereci ARP, Bölünmüş Talepli ARP, vd.)

Katı atık toplama rota optimizasyonu ile ilgili bilinen en eski çalışmalardan biri 1974 yılında Beltrami ve Bodin tarafından yapılan çalışmadır. Çalışmada New York ve Washington belediyelerinin çöp toplama sistemi Klasik ARP olarak ele alınmıştır. Tasarruf algoritmasının (Clarke & Wright, 1964) gelişmiş bir versiyonu ile çözüm aranan çalışmayla belediyelere büyük katkılar sağlanmıştır (Beltrami & Bodin, 1974).

1.1 ARAŞTIRMA AMACI

Bu tez çalışmasında İstanbul Ümraniye ilçesinin katı atık toplama süreci KARP olarak ele alınarak tüm ihtiyacı karşılayacak en kısa rotaların oluşturulması ve farklı senaryolar için rota optimizasyonun sağlanması amaçlanmıştır. Literatürde sıklıkla kullanılan yöntemlerin ve önerilen çözümün devasa boyutlu bir gerçek hayat probleminde test edilerek performans analizlerinin yapılması hedeflenmektedir.

1.2 ARAŞTIRMA MOTİVASYONU

Literatürde hem ARP ile ilgili hem de Atık Yönetimi ve Rota Optimizasyonu ile ilgili birçok çalışma olmasına rağmen iki konunun birlikte ele alındığı, Katı Atık Toplamada ARP uygulamasıyla ilgili çalışma sayısı yeterli değildir. Han ve Cueto (2015), yaptıkları çalışmada sekiz farklı veri tabanında özel anahtar kelimeler ile aramalar yaparak konuyla ilgili 1974-2015 yılları arasında yazılmış yalnızca 42 dergi ve 65 makaleye ulaşabilmişlerdir.

ARP np-zor yapısıyla her zaman kesin sonuç veren yöntemlerle çözülemediğinden sıklıkla sezgisel ve meta sezgisel yöntemler kullanılır. Literatürde bu yöntemlerin birbirleriyle performans kıyasının yapıldığı büyük ölçekli örnek veri setleri genellikle 100-200 düğümden oluşmaktadır. Bu çalışmada bir ilçenin katı atık toplama rotalaması için konteyner verisi girdi olarak kullanılacağı için problem ölçeği devasa olarak nitelendirilebilir. Bu kadar büyük ölçekli bir problemde ARP ve çözüm yöntemlerin performansının incelenmesi önemli bir yenilik olacaktır.

1.3 ARAŞTIRMA KAPSAMI

Çalışmanın kapsamı, Ümraniye Belediyesinin katı atık toplama rotasını eniyileyecek araç rotalama probleminin çözülmesidir. Klasik ARP'den farklı olarak çalışmada iki farklı senaryo üretilmiştir. Bu senaryolardan ilki ilçe sınırları içerisinde optimal çöp toplama optimizasyonunu sağlayacak senaryodur. Bu senaryoda çöp üretme derecesi bakımından diğer mahallelerden fazlasıyla ayrılan iki mahalleye çift sefer, kalan tüm mahallelere tek sefer gidilmesi planlanmaktadır. İkinci senaryoda ise vatandaş memnuniyetini artırmak için optimal senaryoyu göz ardı ederek konteynerlerin daha sık toplandığı, daha yoğun sefer sayıları içeren, çöp kamyonlarının ve işçilerinin daha eşit iş dağılımını sağlayacak, konteynerdeki çöp birikmesini önleyecek senaryodur. Bu problem için k-ortalamalar kümeleme yöntemini içeren iki kademeli önce kümele sonra rotala prensibine dayanan bir çözüm ile literatürde popüler olan tur kurucu ve tur geliştirici meta sezgisel yöntemler kullanılmıştır. Bahsedilen

yöntemlerin söz konusu problem üzerindeki etkinlikleri incelenerek çözüm kaliteleri kıyaslanmıştır.

1.4 TEZİN YAPISI

Tez genel hatlarıyla beş ana bölümden oluşmaktadır. İlk bölümde araştırmanın amacı, kapsamı ve önemi anlatılmıştır. İkinci bölümde literatürdeki atık toplama rota optimizasyonu, araç rotalama problemi ve K-Ortalama Kümeleme algoritmasıyla ilgili çalışmalar özetlenmiştir. Üçüncü bölümde araştırma modeli, verilerin toplanma, analiz ve düzenlenme yöntemleri, probleme yönelik çözüm yöntemleri ve kullanılan yazılımlar açıklanmıştır. Dördüncü bölümde, metodoloji bölümünde açıklanan yöntemler gerçek bir problem üzerinde uygulanmıştır ve iki aşamalı yeni bir çözüm yöntemi önerilmiştir. Son bölümde ise uygulama kısmındaki sonuçlar yorumlanarak araştırmanın sonuçları tartışılmıştır ve önerilerde bulunulmuştur.

BÖLÜM 2. LİTERATÜR İNCELEMESİ

Tez çalışmasının bu bölümünde ARP, atık toplama optimizasyonu ve rotalama ve K-Ortamalar Kümeleme algoritmasıyla ilgili literatürde yapılmış çalışmalar derlenmiştir. Yapılan incelemeler sonucunda yöntemlerin birçok farklı alanda kullanışlı olduğu anlaşılmıştır.

2.1 ARAÇ ROTALAMA PROBLEMİ

ARP merkez ve depolar arasındaki en iyi güzergahın bulunmasını amaçlayan np-zor karmaşıklık düzeyinde bir optimizasyon problemidir. Problemden tek bir araç söz konusu olabileceği gibi birden fazla araç da olabilir. En iyi güzergahın belirlenmesindeki amaç genellikle en kısa mesafenin bulunması olsa da araç sayısını veya zamanı en aza indirmek gibi farklı amaçlar için de problem tanımı oluşturulabilir. Problemden araçla, müşteriyle veya süreçle ilgili birtakım kısıtlar bulunabilir. Tüm bu etkenler farklı ARP türlerini ve tanımlarını doğurmuştur.

Literatürde ARP ile ilgili ilk çalışma 1959 yılında yazılan “The Truck Dispatching Problem” makalesidir (Dantzig & Ramser, 1959). Problem Gezgin Satıcı Probleminin (Flood, 1956) genelleştirilmiş hali olarak tanıtılmıştır ve benzin kamyonlarının optimum rotalarının bulunması için çözüm önerisinde bulunulmuştur. Herhangi pratik uygulama yapılmasa da optimale yakın çözümü verecek doğrusal programlama formülü geliştirilmiştir ve gelecekteki çalışmaların fitilini ateşlemiştir.

Clarke ve Wright (1964), bu problemin çözümü için sezgisel Tasarruf Algoritmasını geliştirmiştir. Dantzig ve Ramser çalışmalarında minimum mesafe için doldurma yaparken, Clarke ve Wright’ın önerdikleri bu yöntem en yakın kapasite için doldurma yapmaktadır. Çalışma sonucunda yapılan testlere göre önerilen yöntemin daha iyi sonuçlar verdiği belirtilmiştir.

Bertsimas (1992), ARP’de talebin belirsiz olduđu durumları inceleyen bir çalışma yaparak, toplam mesafeyi en aza indirmeyi amaçlayan iki farklı öncelik belirleme stratejisi belirlemiştir. Öncelikleri belirlerken deterministik modellerden oluşan döngü sezgiselini stokastik ARP’ye uyarlayarak en iyi ve en kötü senaryolardaki sonuçları kıyaslamıştır.

Toth ve Vigo (2002), çalışmalarında simetrik ve asimetrik maliyet/mesafe matrisine sahip KARP için döneminin en iyi dal ve sınır algoritmalarını incelemişlerdir. Ayrıca çalışmada k-ağaç (Fisher, 1994), b-eşleştirme (Miller, 1995), küme bölümlendirme (Hadjiconstantinou ve diğerleri, 1995) gibi esnetme yöntemleri de tartışılmıştır. Yapılan bu rahatlatma işlemi sonrasında problem büyüklüğü 25 noktadan 100 ve üzerine yükseltilmiştir.

Ralphs ve arkadaşları (2003), KARP için ayırma tabanlı dal ve kesim algoritması önermişlerdir. Çalışmalarında sonuçların ümit verici olduđu ve dönemi için daha önce karşılaşılan problemlerin çoğuna çözüm bulunduđu belirtilmiştir. Sonrasında günümüze kadar bu çalışmayı temel alan birçok yeni çalışma da yapılmıştır.

Baker ve Ayechev (2003), ARP’nin temel versiyonunda araç kapasitesini ve kat edilen mesafeyi kısıtlayarak çözmek için genetik algoritma (Goldberg & Holland, 1988) ve yerel arama algoritmasından oluşan hibrit bir çözüm önermişlerdir. Önerdikleri yöntemin dönemin en başarılı sezgisel algoritmaları olarak gösterilen Tabu Arama (Glover, 1989) ve Benzetimli Tavlama (Laarhoven & Aarts, 1987) algoritmalarıyla çözüm süresi ve kalitesi açısından yarışacak kadar iyi sonuçlar verdiğini ispatlamışlardır.

Amous ve arkadaşları (2017), kat edilen mesafeyi en aza indirmeyi amaçlayan KARP’yi Değişken Komşu Arama algoritması (VNS) (Mladenović & Hansen, 1997) ile ele almışlardır. Ayrıca çözüm yöntemi Değişken Komşu İniş Algoritmasıyla (Hertz & Mittaz, 2001) birlikte birçok farklı arama algoritmasını da kapsamaktadır.

Borčinová (2017), KARP çözümü için iki farklı doğrusal programlama modeli önermiştir. Önerdiği yöntemlerden ilki akış tabanlı doğrusal programlama, ikincisi güncellenmiş atama formülasyonudur. Tasarruf algoritması tabanlı karma doğrusal programlama modeli önererek, iki yöntemin de küçük ölçekli problemlerde optimal

sonucu verdiğini belirtmiştir. Bu çalışma KARP ile ilgili yapılan çalışmalardan alt turları eleme yöntemi açısından ayrılmaktadır.

Cömert ve arkadaşları (2018), KARP çözümünü hiyerarşik bir şekilde ele alarak önce kümele ardından rotalama problemine dönüştürmüşlerdir. K-ortalamalar, k-medoids (Kaufman & Rousseeuw, 1990) ve rastgele kümeleme algoritmalarıyla kümeleme yapıldıktan sonra her kümenin rotalama problemi dal ve sınır algoritmasıyla çözülmüştür. Ayrıca çalışmada bir süpermarket zinciri için uygulama gerçekleştirilmiştir ve en iyi sonucu veren kümeleme yönteminin k-medoids olduğu anlaşılmıştır. Önerilen yöntemin büyük ölçekli KARP'lerinin çalışma sürelerinin azaltacağı iddia edilmektedir.

Goli ve arkadaşları (2018), KARP için maliyetleri düşürmesinin yanı sıra müşteri memnuniyetini de göz önünde bulunduran yapay zeka destekli guguk kuşu algoritması ile çözüm önermişlerdir. Algoritmayı hızlandırmak ve çözüm kalitesini artırmak için benzetimli tavlama algoritması kullanılmıştır. Çalışmada önce GAMS (Genel Cebirsel Modelleme Sistemi, matematiksel programlama ve optimizasyon problemlerini çözmekte kullanılan bir modelleme yazılımıdır) ile optimal çözüm bulunmuştur, ardından klasik guguk kuşu algoritmasıyla (Yang & Deb, 2009) çözüm üretilmiştir. Sonuçlar kıyaslanarak önerilen metodun daha kısa sürede optima yakın sonuçlar verdiği gözlemlenmiştir.

İbrahim ve arkadaşları (2019), çalışmalarında küçük ölçekli bir veri seti üzerinde Sütun Oluşturma (Appelgren, 1969), Google OR-Tools (Perron, 2011) ve Pekiştirmeli Öğrenme algoritmalarını çalıştırarak (Kaelbling ve diğerleri, 1996) KARP için çözüm arayarak sonuçları kıyaslamışlardır. Ayrıca Pekiştirmeli Öğrenme ve Google OR-Tools algoritmaları büyük ölçekli bir veri seti üzerinde de çalıştırılarak büyük problemlerdeki yeterlilikleri kıyaslanmıştır. Sonuçta Pekiştirmeli Öğrenme yöntemi nispeten daha iyi sonuçlar verse de OR-Tools çözüm üretme hızı açısından çok daha iyi sonuç vermiştir.

Yazgan ve arkadaşları (2020), bir boya fabrikasındaki KARP için Tasarruf Algoritması ve yerel aramalı sezgisel bir algoritma kullanmışlardır. İki yöntemin sonuçları ile fabrikanın gerçek verileri kıyaslanarak en iyi sonucu veren yöntemin yerel aramalı sezgisel algoritma olduğu anlaşılmıştır.

İlhan (2020), çalışmasında KARP'yi çözmek için yeni bir popülasyon tabanlı benzetimli tavlama algoritması önermiştir. Önerdiği yöntemde yer değiştirme, ekleme ve ters çevirme olmak üzere üç rota geliştirme operatörü her tekrarda rastgele seçilerek kullanılmıştır. Performans testlerinde sıklıkla kullanılan 63 veri seti üzerinde çalıştırılan yöntem 23 örnekte optimal sonucu verirken, kalan 40 veri setinde de kıyaslanan yöntemlere göre daha dengeli sonuçlar verdiği belirtilmiştir.

Rabbouch ve arkadaşları (2020), KARP'de benzetimli tavlama algoritmasının muhtemel zayıf yönlerini geliştirecek yeni ve dinamik bir benzetimli tavlama algoritması, Ampirik Benzetimli Tavlama algoritmasını, önermişlerdir. Önerilen metot standart benzetimli tavlama algoritmasının kabul kriterini güncellemek için makul çözümlerin en kötüsünün son kısmını giderek artacak şekilde kullanır. Bu yöntemle standart benzetimli tavlamanın rastgeleliğinin azaltıldığı, yakınsamasının iyileştirildiği, genel olarak iyi bir performansa ulaşıldığı belirtilmiştir.

Hottung ve Tierney (2020), çalışmalarında ARP'ye makine öğrenmesiyle yaklaşan yöntemlerin standart optimizasyon metotlarından geride kaldığını ifade ederek, bu farkı kapatmak için yeni bir Yerel Komşu Arama (LNS) algoritması önermişlerdir. Bu yöntem derin sinir ağlarını temel alan bir yaklaşımdır. KARP üzerinde yapılan testlere göre önerilen yeni LNS algoritması, standart LNS ve literatürde tanınan Geliştirilmiş Lin Kernighan algoritmasına (Lin & Kernighan, 1973) göre iyi sonuçlar vermiştir.

Sitek ve arkadaşları (2021), posta ve kurye taşıma probleminin çözümü için ARP'nin varyantlarının kombinasyonunu kullanmışlardır. Kısıt programlama, genetik algoritma ve matematiksel modelleme yöntemlerinden oluşan hibrit bir yaklaşım uygulanmıştır. Yapılan deneylerle modelin ve yaklaşımın etkinliği kanıtlanmıştır.

2.1.1 ARP Türleri

En temel, klasikleşmiş ARP, Dantzig ve Ramser'in (1959) çalışmalarındaki gibi belirli sayıda aracın ana depodan başlayarak tüm depolara uğraması ve sonunda rotasını ana depoya dönerek tamamlamasıdır. Bu rotaların minimum maliyetle (mesafe) belirlenmesi amaçlanır. Bu probleme kapasite kısıdı da eklenerek araçların

kapasitelerinden fazla talep karşılamaması sağlanır ve problem Kapasiteli Araç Rotalama Problemi olarak adlandırılır. Müşterilerin taleplerini farklı zaman dilimlerinde gerçekleştirdiği, dolayısıyla seferlerin zaman aralıklarına göre gerçekleştirildiği problemler Zaman Pencereli ARP (Desrochers ve diğerleri, 1992) olarak adlandırılır. Müşterinin talebinin belirsiz olduğu durumlarda Belirsiz Talepli ARP (Bertsimas, 1992), müşteri talebini tek bir araç yerine birden fazla aracın parça parça karşıladığı durumlarda Bölünmüş/Parçalı Talepli ARP (Archetti & Speranza, 2008), müşteri taleplerinin farklı periyotlarda karşılandığı durumlardaysa Periyodik ARP (Gaudioso & Paletta, 1992) olarak tanımlanır. Benzer özel durumlar için farklı farklı ARP türleri de bulunmaktadır.

2.1.1.1 Kapasiteli Araç Rotalama Problemi

Bu çalışmada katı atık toplama rota problemi KARP ile ele alınmıştır. Belirli sayıda ve eşit kapasitedeki çöp kamyonları tek bir merkezden başlayarak her gün kendi güzergâhlarına ait sokaklardaki çöpleri toplayarak merkeze geri dönmektedirler. Probleme çöp toplama işlemi, kamyonların tüm sokaklara uğraması, her sokağın çöpünün tek bir kamyon tarafından toplanması ve kamyon kapasitesinin aşılması kısıtlarıyla ve tüm sürecin minimum mesafeyi veren rota ile tamamlanması amacıyla çözülmüştür.

2.2 ATIK TOPLAMA OPTİMİZASYONU VE ROTALAMA

Han ve Cueto (2015), evsel, ticari ve endüstriyel olarak sınıflandırılan atıkların rotalanması için kullanılan farklı teknikleri analiz etmişlerdir. Çalışma literatürü özetlemesi ve tasnif etmesi açısından benzer problemler için yol haritası olarak değerlendirilebilir. Çöp toplama ve rotalama optimizasyonu yapılırken çevresel etkilerin yeterince göz önünde bulundurulmadığını, KARP'nin amaçlarına sadece ekonomik değil, çevresel etkilerin de eklenmesi gerektiğini belirtmişlerdir.

Assaf ve Saleh (2017), çalışmalarında Filistin Nablus şehri için katı atık toplama problemini ARP ile ele alarak genetik algoritma temelli çözüm önermişlerdir. Çalışmada amaç çöp kamyonlarının kat ettikleri mesafeyi minimuma indirmek ve

maliyeti düşürmektir. Yapılan çalışma ile kat edilen mesafenin %66 azaltılabileceği ve çöp toplama süresinin 7 saatten 2,3 saate düşürülebileceği gösterilmiştir.

Akhtar ve arkadaşları (2017), akıllı çöp kovası konseptiyle, çöp toplama rotalarını optimize etmek için KARP'ye Geri İzlemeli Arama algoritmasıyla (Van Beek, 2006) çözüm önermişlerdir. Kat edilen mesafeyi azaltmak ve çöp boşaltma sıklığını optimal hale getirmek için çöp kovalarının eşik değeri üzerinde incelemeler yapılmıştır. Yapılan testler sonucunda toplam mesafenin yaklaşık %37, toplam yakıt masrafının yaklaşık %50 azaltılabileceği gösterilmiştir.

Erfani ve arkadaşları (2017), İran'ın Ahmadabad şehrindeki çöp kutularının saklama ve toplama için KARP'yi Coğrafi Bilgi Sistemi (CBS) ile çözmüşlerdir. Çalışma sonucunda mevcut çöp toplama sisteminin yetersiz olduğu, önerilen yöntemle tur sayısının %12,5, toplam personel sayısının %41, nihai amaç olan toplam kat edilen mesafeninse %53 iyileştirilebileceği gözlemlenmiştir.

Kuo ve Zulvia (2017), KARP için Genetik Algoritma ve Karınca Kolonisi Optimizasyon algoritmasının (Dorigo & Di Caro, 1999) kombinasyonundan oluşan hibrit bir algoritma ve 2 farklı yerel arama algoritmasını (Prim'in arama algoritması (Prim, 1957) ve 2-opt arama algoritması (Croes, 1958)) kullanarak çözüm önermişlerdir. Sekiz farklı performans testi sonrasında önerilen metodun yarışmacı sonuçlar verdiği anlaşılmıştır. Ayrıca çalışmada Endonezya'daki bir şehirde bulanık talepli çöp toplama uygulaması gerçekleştirilmiştir. Sonuçların umut verici olduğu belirtilmiştir.

Hannan ve arkadaşları (2018) KARP'nin çözümü için Güncellenmiş Parçacık Sürüsü Optimizasyonu (PSO) algoritmasını önermişlerdir (Kennedy & Eberhart, 1995) ve bu yöntemle farklı veri setleri için eşik atık seviyesi ve çizelgelemeleri içeren konseptler uygulamışlardır. Kat edilen mesafeyi minimuma indirerek optimal rotayı bulmayı amaçlayan yöntemin deneyler sonucunda atıkların toplanması ve rota optimizasyonu açısından iyi sonuçlar verdiği, atık toplama sitelerinde sosyoekonomik ve çevresel etkileri açısından önemli bir araç olabileceği vurgulanmıştır.

Asefi ve arkadaşları (2019), katı atıkların toplanıp, rotalanması ve maliyetlerin azaltılması için karışık tam sayılı doğrusal programlama modeli (MILP) oluşturmuşlardır. Çözüm için Değişken Komşu Arama algoritmasının tek başına

çalıştığı ve Değişken Komşu Arama algoritmasıyla Benzetimli Tavlama algoritmasının birlikte kullanıldığı iki farklı sezgisel yaklaşım önerilmiştir. MILP modeli CPLEX yazılımı ile çalıştırılmıştır ve önerilen iki modelin de optimale yakın sonuçlar verdiği, etkili yöntemler olduğu anlaşılmıştır.

Vu ve arkadaşları (2020), çalışmalarında Austin, Texas'taki çöp toplama sistemini, 2023 yılı için tahmin edilen verilere göre modellemişlerdir. Çöp toplama sıklığı, çöp toplama tipi, çöp kamyonu tipi ve çöp turu gibi parametreler arasındaki ilişki 48 farklı senaryo ile CBS kullanılarak incelenmiştir. Çalışma sonucunda çöp toplama sıklığının artırılmasıyla toplam çöp toplama süresinin iyileştirilebileceği anlaşılmıştır.

2.3 K-ORTALAMALAR KÜMELEME YÖNTEMİ

Karmaşık problemlerin çözümünde kullanılan en temel yöntemlerden biri problemleri daha küçük alt problemlere ayırarak parçadan bütüne ulaşmaktır. Bu doğrultuda literatürde önce grupla sonra rotala prensibiyle hazırlanmış çalışmalar (Bodin, 1975; Hiquebran ve diğerleri, 2007; Kloimüllner ve diğerleri, 2015; Comert ve diğerleri, 2018) bulunmaktadır. Bu çalışmalarda genel itibarıyla problem farklı kümeleme algoritmalarıyla gruplara ayrılmıştır ve sonrasında her bir grup farklı bir problem gibi ele alınarak kesin veya sezgisel yöntemlerle rotalanmıştır. Diğer yöntemlere kıyasla daha hiyerarşik yapıda olduğu söylenebilir. Bu tez çalışmasında literatürde en sık kullanılan kümeleme algoritmalarından biri olan K-Ortalamlar yöntemi kullanılmıştır.

Oyelade ve arkadaşları (2010), çalışmalarında yükseköğrenim gören öğrencilerin akademik başarısını ölçmek ve takip etmek için k-ortalamlar kümeleme algoritmasını kullanmışlardır. Nijerya'daki bir üniversitenin tek sömestrlik akademik sonuçları veri seti olarak kabul edilerek 79 öğrenci 4 kümeye ayrılmıştır. Yapılan kümeleme ile öğrencilerin akademik başarısıyla ilgili çıkarımlarda bulunulmuştur.

Di Giuseppe ve arkadaşları (2014), k-ortalama kümeleme algoritmasını çok değişkenli jeofizik veri analizinde araç olarak kullanmışlardır. Çalışmalarında sığ fay bölgelerini görüntülemeye yönelik bir uygulama yapmışlardır. 1980 yılındaki Güney

İtalya depreminin yüzey yırtılmasından alınan iki boyutlu yüksek çözünürlüklü manyetotellürik ve sismik veriler interpolasyon yöntemiyle homojen hale getirildikten sonra k-ortalamalar ile jeofizik karakteristiklerine göre beş kümeye ayrılmıştır. Bu kümelemenin daha kolay ve eksiksiz bir jeofizik yorumlamayı mümkün hale getirdiği belirtilmiştir.

Xiong ve arkadaşları (2017), metin kümeleme için standart k-ortalamalar kümeleme algoritmasındaki başlangıç küme merkezlerini optimize ederek kümelemenin performansını artıracak yeni bir geliştirilmiş k-ortalamalar kümeleme algoritması önermişlerdir. Önerilen metot Çince bir deney platformu üzerine test edilmiştir. Karşılaştırma sonucunda önerilen metot hız bakımından k-ortalamaların gerisinde kalsa da doğruluk bakımından üstün gelmiştir.

Chevez ve arkadaşları (2017), Arjantin'in Great La Plata bölgesindeki evsel elektrik tüketiminin belirlenmesi ve analizi için k-ortalamalar kümeleme algoritmasını kullanmışlardır. Çalışmalarında elektrik talebini etkileyen sosyo-demografik sebepler ve elektrik tüketiminin coğrafik dağılımı incelenmiştir. Çalışma sonucunda elektrik tüketimini etkileyen konumlar 8 kümeye ayrılmıştır ve her küme için karakteristik özellikler belirlenmiştir.

Abadi ve arkadaşları (2018), bir eğitim kurumunun tanıtım stratejisini belirlemek için k-ortalamalar algoritmasını kullanmışlardır. Yapılan çalışmada Endonezya'daki bir meslek lisesinin öğrencilerinin okullarını seçme sebepleri anket yapılarak öğrenilmiştir. Toplanan veri seti üzerinde k-ortalamalar kümeleme algoritması çalıştırılmıştır ve öğrenciler 5 kümeye ayrılarak, tanıtım stratejisi için 5 temel içgörü belirlenmiştir.

Tan ve arkadaşları (2019), internet ağlarına yönelik izinsiz girişlerin tespiti için k-ortalamalar kümeleme algoritması tabanlı Kendini Düzenleyen Özellik Haritası (SOM) (Kohonen, 1990) sinir ağlarını kullanmışlardır. Klasik SOM yaklaşıma k-ortalamalar tabanlı çözüm önererek kümelemedeki tekrar sayısı ciddi oranda azaltılmıştır. Ayrıca çalışmada standart k-ortalamalar kümeleme algoritmasının doğruluğunun başlangıç çözümüyle ilgili olduğu ve önerilen yöntemle kümeleme algoritmasının veriminin artırılacağı belirtilmiştir.

Wardani ve arkadaşları (2019), Endonezya'nın Semarang şehrinde 2014-2017 yılları arasında çocuklar görülen tüberküloz vakalarını k-ortalamlar ile kümelemişlerdir. Şehir yüksek, orta ve düşük yaygınlık seviyeleri olmak üzere üç kümeye ayrılarak CBS ile yoğunluk haritası oluşturulmuştur. Bu çalışma karar vericilerin önleyici kararları vermesine yardımcı olurken, strateji geliştirmek için de ilk adım özelliği taşımaktadır.

Shelly ve arkadaşları (2020), Amerikan futbolu kolej öğrencilerinin fiziksel antrenman gruplarının belirlenmesi için k-ortalamlar kümeleme algoritmasını kullanmışlardır. Çalışmada giyilebilir teknolojik ürünlerle 1. Ligdeki profesyonel atletlerin iki sezonluk verileri kayıt altına alınarak oyunun oyunculardan beklediği talep öğrenilmiştir. Sonrasında bu talebe karşılık verebilecek antrenman grupları k-ortalamlar algoritmasıyla belirlenmiştir. Elde edilen antrenman grupları, geleneksel yöntemlerle belirlenmiş antrenman gruplarıyla benzeşirken profesyoneller için sağlam bir başlangıç noktası oluşturmaktadır.

Shamrat ve arkadaşları (2020), yaptıkları çalışmayla kariyer portallarındaki benzer işleri ayırt etmek için k-ortalamlar kümeleme algoritmasını kullanmışlardır. Benzer işlerin saçılım grafiği ile görsel olarak kümelenmesiyle işlerin mevcut yoğunlukları belirlenerek iş pazarının genel durumu özetlenmiştir. Yapılan çalışma iş arayanların pazardaki talebi öğrenip yeteneklerini bu ihtiyaçlar doğrultusunda geliştirmesi açısından yol haritası sunmaktadır.

Zhang ve arkadaşları (2020), akıllı bina uygulamasında kullanılan, nesnelerin internetiyle birbirine bağlanan elektrikli cihazların kimlik tanımlaması için k-ortalamlar kümeleme algoritmasını kullanmışlardır. Çalışmada cihazların kümelenmesi için gerekli olan küme sayısı dirsek metoduyla belirlenerek, cihazlar farklı karakteristik özelliklere göre kümelenmiştir. Uygulama sonucunda geliştirilen cihaz tanımlama modelinin, benzer cihazların tanınarak eşleştirilmesi konusunda başarılı olduğu gözlemlenmiştir.

BÖLÜM 3. ARAŞTIRMA METODOLOJİSİ

Bu bölümde KARP için çözüm yöntemleri incelenerek bu yöntemlerin iki farklı senaryoda kullanılışı gösterilmiştir. Kümeleme ve sezgisel yöntemlerin sıralı olarak kullanıldığı, önce kümele sonra rotala prensibine dayanan bir çözüm de incelenmiştir. Ayrıca tez çalışması boyunca kullanılan yazılım ve paketler, kullanılan algoritmalara ait parametreler, verilerin alınış şekli ve türü, senaryoların hikayesi anlatılmıştır.

3.1 ÇÖZÜM YÖNTEMLERİ

ARP NP-zor optimizasyon problemi olduğu için her zaman optimal sonucu veren yöntemler bulunmamaktadır. Özellikle problem ölçeği büyüdükçe kesin sonuç veren yöntemler makul süre içerisinde sonuç verememekte veya sonuç üretememektedir. Bu yüzden optimal sonuca yaklaşan, hızlı çözüm üreten sezgisel yöntemler geliştirilmiştir. Sezgisel algoritmaların yaygınlaşmasıyla ve gelişmesiyle meta-sezgisel algoritmalar kullanılmıştır. Ayrıca birçok alanda kullanılan kümeleme algoritmaları da ARP'nin çözümünde kullanılan yöntemlerden biridir. Bu tez çalışmasında KARP için çözüm yöntemleri incelenmiştir.

3.1.1 Kesin Yöntemler

KARP'nin kesin yöntemlerle çözümü için doğrusal programlama modeli incelenmelidir. Klasik KARP'ye ait doğrusal programlama modeli aşağıdaki gibidir (Altabeeb ve diğerleri, 2019). N kümesi n adet düğümden oluşmaktadır, bu çalışma kapsamında düğümler sokakları ifade etmektedir. Her sokağın q kg çöp talebi bulunmaktadır. Sokaklar arası uzaklık c_{ij} ile ifade edilmiştir. K araç sayısını, Q araç kapasitesini temsil etmektedir.

$$\min \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} X_{ij}^k \quad (1)$$

Denklem 1’de görüleceği üzere amaç fonksiyonu araçların kat ettiği toplam mesafeyi en aza indirmeyi amaçlamaktadır. Denklemdeki X_{ij}^k k aracının i düğümünden j düğümüne gidip gitmediğini ifade etmektedir, 0 veya 1 değerini alır.

Kısıtlar:

$$\sum_{k=1}^K \sum_{i=0}^N X_{ij}^k = 1 \quad j \in \{1, \dots, N\}: i \neq j \quad (2)$$

$$\sum_{k=1}^K \sum_{j=0}^N X_{ij}^k = 1 \quad j \in \{1, \dots, N\}: i \neq j \quad (3)$$

$$\sum_{i=0}^N \sum_{j=0}^N X_{ij}^k q_i \leq Q \quad k \in \{1, \dots, K\} \quad (4)$$

$$\sum_{j=1}^N X_{ij}^k \leq 1 \quad \text{for } i = 0 \quad k \in \{1, \dots, K\} \quad (5)$$

$$\sum_{j=1}^N X_{ji}^k \leq 1 \quad \text{for } i = 0 \quad k \in \{1, \dots, K\} \quad (6)$$

$$\sum_{j=1}^N X_{ij}^k = \sum_{j=1}^N X_{ji}^k \quad \text{for } i = 0 \quad k \in \{1, \dots, K\} \quad (7)$$

Denklem 2 ve 3 sokakların yalnızca bir araç tarafından ziyaret edilmesinin kısıdıdır. Denklem 4 araç kapasitesinin aşılmamasını kontrol eder. Denklem 5, 6 ve 7 her turun 0 düğümünden yani depodan başlamasını ve bitmesini sağlar.

3.1.1.1 Dal ve sınır Algoritması

Dal ve sınır algoritması temelde bir optimizasyon problemini dallara (alt problemlere) ayırarak tüm alternatif sonuçlar içerisinde en iyisini seçmeyi amaçlar. Doğrusal programlama çözümü tam sayılı çıkmadığında dal ve sınır algoritması kullanılarak çözüm uzayı aşağıdaki gibi yeniden değerlendirilir (Toth & Vigo, 2002).

- DP0. Doğrusal programlamadan elde edilen en iyi çözüm alt sınır (AS) kabul edilir. Tam sayı çıkmayan bir değişken ele alınır ve bu değişkene en yakın alt ve üst tam sayı değerleri çözüm uzayından çıkartılır.
- Bir sonraki DP_i aşamalarında her iki alt ve üst tam sayı değerleri DP0'a kısıt olarak eklenir ve yeniden çözülür.
- Dallandırmaya devam edilip edilmeyeceğine karar vermek için çözüm alt sınırla kıyaslanır. Eğer mevcut çözüm alt sınırdan daha iyiye (minimizasyon problemleri için daha küçük, maksimizasyon problemleri için daha büyük alt sınır aranır), tüm değişkenler tam sayılı ve $DP_i=AS$ ise sonuç optimal kabul edilir. Şartlar sağlanmıyorsa yeni dallar eklenerek şartlar sağlanana kadar çözüme devam edilir.
 - Dal ve sınır algoritmasında ilerlemek için iki farklı yöntem bulunmaktadır. Bunlardan ilki LIFO yani ilk önce son üretilen alt problemin ilk çözülmesidir. Diğeriyse atlamalı yöntemdir. En iyi Z değerine sahip alt problem dallandırılır.

3.1.1.2 Kesme düzlemi algoritması

Dal ve sınır algoritmasında olduğu gibi kesme algoritması da başlangıç çözümü olarak doğrusal programlama çözümünü alır. Bu çözüme yeni kısıtlar ekleyerek çözüm kümesini (düzlemini) daraltır.

Başlangıçta tam sayı kısıdı görmezden gelinir. Çözüm düzlemi incelenerek; tam sayılı çözümleri içeride, tam sayılı olmayan çözümleri dışarıda bırakacak bir kesme yapılır. Yeni kesme kısıdı başlangıç doğrusal programlama modeline eklenerek yeniden çözülür. Tüm değişkenler tam sayılı çözüme ulaşana kadar bu süreç tekrarlanır.

3.1.2 Sezgisel ve Meta-sezgisel Yöntemler

Sezgisel yöntemler genel olarak tur kurucu ve tur geliştirici algoritmalar olarak ikiye ayrılabilir. Tur kurucu algoritmalar, problem için uygun çözüm bulurlar. En iyi çözümü bulma gayesi güdülmez. Tur geliştirici algoritmalar ise başlangıçta var olan bir uygun çözümü alarak geliştirmeyi amaçlarlar. En iyi çözümü bulma garantisi vermese de en iyi çözüme yaklaşma hedefiyle çalışırlar.

3.1.2.1 En yakın komşu algoritması

Gezgin satıcı problemine çözüm için geliştirilmiştir. Hızlı ve basit olmasından dolayı ARP’de de sıklıkla tur kurucu algoritma olarak kullanılmaktadır. Başlangıç noktasından başlayarak sürekli olarak mevcut noktaya en yakın ziyaret edilmemiş noktaya gitmek hedeflenmektedir. En yakın noktanın talebi araç kapasitesini aşacaksa, o nokta pas geçilir ve sonraki nokta hedeflenir. Araç kapasitesi dolana veya tüm talepler karşılanana kadar turlar oluşturulur. Tur sona erdiğinde ziyaret edilmemiş nokta varsa yeni tur oluşturulur.

3.1.2.2 Tasarruf algoritması

Tasarruf algoritması düğümlere ayrı ayrı gidip gelmek yerine iki düğümü birbirine bağlayarak tasarrufu maksimize etmeyi amaçlayan bir tur kurucu algoritmadır.

Başlangıçta başlangıç noktası (0.düğüm) ve tüm düğümler arasında gidip gelen turlar düşünülür. Sonrasında herhangi iki düğüm arasında yeni bir bağlantı kurulur ve kurulan bağlantıyla tasarruf sağlanmış olur. Bu tasarruf S_{ij} ile gösterilir, tüm düğümler için aşağıdaki formülle hesaplanır. C_{ij} iki düğüm arasındaki maliyeti yani mesafeyi ifade etmektedir (Clarke & Wright, 1964).

$$S_{ij} = C_{i0} + C_{0j} - C_{ij} \quad (6)$$

Tüm tasarruf değerleri büyükten küçüğe sıralanır ve en fazla tasarrufu sağlayacak ikililer sırayla, araç kapasite kısıdını göz önünde bulundurarak aynı rotaya eklenir.

3.1.2.3 Benzetimli tavlama algoritması

Benzetimli tavlama algoritması, yerel minimumdan kurtulmak için yerel arama algoritmasının rastgele arama, kabul ve durma kriterlerine farklı açıdan yaklaşmaktadır. Yerel arama algoritmalarının aksine benzetimli tavlama algoritması iyileştirici hamleleri tercih ederken, belirli ihtimaller dahilinde iyileştirici olmayan hamleleri de tercih edebilir. Bu ihtimaller sıcaklık parametresiyle kontrol edilir ve “0” olma eğilimindedir (Osman, 1993).

Algoritma bir başlangıç çözümü ve sıcaklık değeri olarak başlar, yeni çözümler üreterek ürettiği çözümleri başlangıç çözümüyle kıyaslar. Kabul kriteri ve sıcaklığa göre çözümü günceller veya yeni çözümler üretmeye devam eder (Zhan ve diğerleri, 2016).

Başlangıçta genellikle yüksek sıcaklık değeri tercih edilir, böylelikle ayrık değerlere daha fazla müsaade edilir. Sıcaklık soğumaya başladıkça (T değeri azaldıkça), ayrık değerler kabul kriterinden geçememeye başlar ve çözüm kümesi daralarak belirli noktaya kümelenir.

3.1.2.4 Tabu arama algoritması

Tabu arama algoritması yerel minimumdan kaçmak için hafızasını kullanır, belirli süre için tabu listesi oluşturur ve çözüm uzayını genişleterek global minimumu bulmayı hedefler. Komşu arama algoritmaları genellikle tekrarlamalı ve iyileştirmeli ilerlerler. Bulunan en iyi olurlu çözüm ve bir sonraki çözüm kıyaslandığında çözüm geliştirilmiyorsa çözüm uzayında ilerlenmez. Ancak Tabu arama algoritması yerel minimumdan kaçmak için kendi oluşturduğu tabu listesine göre kötü çözümü de kabul edebilir (Hertz ve diğerleri, 1995).

Tabu listesi kısa hafıza da bulunur ve algoritmanın belirli süreyle seçimini yasakladığı noktalardan oluşur. Noktalar “tabu tenure” denilen önceden belirlenmiş hamle sayısı kadar tabu listesinde tutulur. Algoritmada tabu yıkma ve durdurma olmak üzere iki önemli kriter bulunmaktadır. Tabu yıkma kriteri sağlandığında algoritmanın tabu olarak algıladığı hamleler mümkün hale gelir. Durdurma kriteri sağlandığındaysa süreç sonuçlanır.

Başlangıçta rastgele bir çözüm oluşturulur ve en iyi çözüm kabul edilir. Komşular arasından yeni bir aday çözüm listesi oluşturulur ve çözümler değerlendirilir. Çözüm listesindeki en uygun çözüm seçilir ve en iyi çözümle kıyaslanır. Durdurma kriteri yeterli değilse tabu listesi güncellenir ve aday çözümler arasına geri dönülür. Tüm adımlar durdurma kriteri sağlanana kadar tekrarlanır ve son çözüm en iyi çözüm kabul edilir.

3.1.2.5 Rehberli yerel arama algoritması

Rehberli yerel arama (Guided Local Search), amaç fonksiyonuna ceza faktörünü ekleyerek çalışan meta-sezgisel bir algoritmadır. Aramalarından deneyim elde eder ve daha önce ziyaret edilen yerel minimumlara tekrar uğrayan aramaları cezalandırır. Yerel arama sırasında tek bir değere takılmadan çözüm alanını genişletmeyi ve çeşitlendirmeyi amaçlar (Kilby ve diğerleri, 1999).

3.1.3 Kümeleme Algoritmaları (K-Ortalamlar Kümeleme Algoritması)

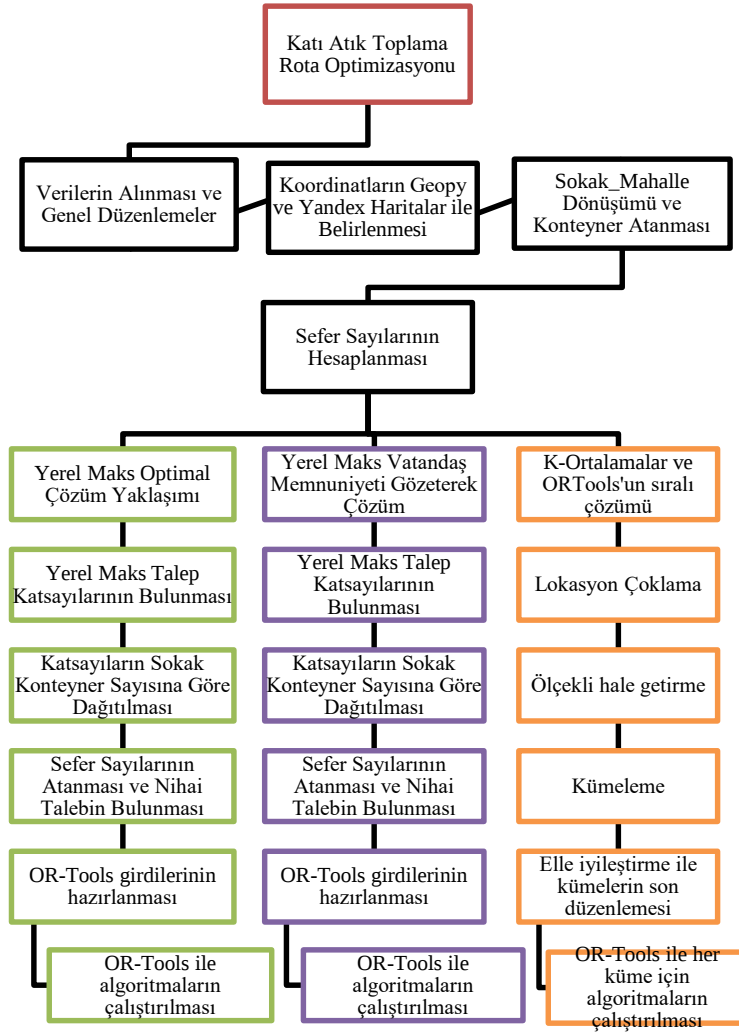
Kümeleme algoritmaları ARP’de genellikle iki aşamalı çözüm yöntemlerinde kullanılmaktadır. Noktalar (sokak, depo, şehir vb.) öncelikle K-Ortalamlar ve K-Medoids gibi kümeleme algoritmalarıyla benzerliklerine göre belirli sayıda kümeye ayrılır, daha sonra kümeler kesin veya sezgisel yöntemlerle ayrı ayrı rotalanır.

İlk olarak MacQueen (1967) tarafından önerilen yöntemde, başlangıçta K küme için K tane rastgele küme merkezi atanır. Tüm noktaların küme merkezlerine olan uzaklıkları hesaplanır ve her nokta kendisine en yakın kümeye atanır. Atama sonrasında her kümenin orta noktası belirlenir ve yeniden tüm noktalar kendisine en yakın kümeye atanır. Her atama sonrası yeni küme belirleme ve yeniden atama işlemi tekrarlanır. Yeni atama yapılamayacağı an kümeleme sona erer.

Bu tez çalışmasında kullanılan k-ortalamlar kümeleme algoritmasına standart K-Ortalamlar’dan farklı olarak küme kapasitesi limitleri eklenmiştir. Dolayısıyla her nokta kendisine en yakın kümeye atandıktan sonra küme kapasitesi ayrıca kontrol edilir, alt ve üst limitlere uymayan küme varsa kümeleme yeniden yapılır. Yeniden atama yapılamayacağı ana kadar kümeleme devam eder.

3.2 ÖNERİLEN YÖNTEMLER

Bu çalışmada iki farklı senaryo ve iki farklı çözüm yöntemi incelenmiştir. Öncelikle ARP'nin optimal çözüm senaryosunda en yoğun iki lokasyon hariç tüm güzergahlara tek aracın gittiği senaryo incelenmiştir. Sonrasında vatandaş memnuniyetini gözeten senaryo, güzergahlara eklenen hesaplanmış sefer sayıları ile incelenmiştir. Son olarak, K-Ortalamlar kümeleme algoritması ve sezgisel algoritmaların birlikte kullanıldığı bir çözüm ile ARP'ye çözüm aranmıştır. Tez çalışmasındaki genel çözüm akışı Şekil 1'de gösterilmiştir.



Şekil 1. Tez Akış Şeması

3.2.1 Kullanılan Yazılım ve Paket Programlar

Verilerin genel düzenlenmesi, diğer yazılımlar için hazır hale getirilmesi ve kümeleme sonrası düzenleme işleminde MS Excel kullanılmıştır.

Sokak koordinatlarının bulunmasında ve tüm algoritmaların çalıştırılmasında Python yazılımı kullanılmıştır. Çalışmada kullanılan Python kütüphaneleri şunlardır:

Geopy: paketiyle (*GeoPy's Documentation!*, n.d.) Yandex Haritalar veri tabanına erişim sağlanarak tüm sokakların koordinatları ve konumları belirlenmiştir.

OR-Tools: Google'ın geliştirdiği, optimizasyon problemlerinde kullanılan OR-Tools paketiyle (*Google Developers*, n.d.) tur kurucu ve tur geliştirici algoritmalar sırasıyla çalıştırılarak KARP'ye çözüm aranmıştır. OR-Tools ARP'nin çözümünde sezgisel ve meta-sezgisel yöntemlerin çalıştırılmasına olanak sağlayan bir optimizasyon aracıdır. Girdi olarak lokasyonların koordinatlarını, taleplerini (kg), karşılıklı mesafe matrisini, araç sayısını ve kapasitelerini ve hangi lokasyonun merkez depo olduğu bilgisini alır; veri modelini oluşturur. Daha sonra ilk sonuç üretme stratejisi (first solution strategy), yerel arama seçeneği (local search option) ve zaman limiti (time limit) girdilerini de alarak OR-Tools belirlenen zaman limiti ve sezgisel algoritmalarla göre çalışır, sonuç üretir. İlk sonuç üretme stratejisi, problemin çözümü için ilk makul çözümün oluşturulmasını sağlar. Yerel arama seçeneğiyle üretilen makul çözümü geliştirmeyi amaçlayan meta-sezgisel yöntemlerin çalıştırılmasını sağlar. Tez çalışmasında kullanılan ilk sonuç üretme stratejileri ve yerel arama seçenekleri Tablo 1'de gösterilmiştir. Zaman limiti, arama algoritmasının her bir nokta için arama yapabileceği maksimum zaman sınırını ifade etmektedir.

Tablo 1. Çalışmada Kullanılan OR-Tools Çalıştırma Seçenekleri

İlk Sonuç Üretme Stratejileri	Yerel Arama Seçenekleri
Path Cheapest Arc (En Ucuz Yay)	Greedy Descent (Açgözlü İniş)
Path Most Constrained Arc (En Kısıtlı Yay)	Guided Local Search (Rehberli Yerel Arama)
Christofides	Simulated Annealing (Benzetimli Tavlama)
Savings (Tasarruf)	Tabu Search (Tabu Arama)

K-Means-Constrained: paketiyle (Levy-Kramer, n.d.) K-Ortalamlar kümeleme algoritması çalıştırılmıştır ve her lokasyon bir kümeye (araca) atanmıştır. Kullanılan paket sayesinde Standart K-Ortalamlar Kümeleme algoritmasından farklı olarak kümelerin eleman sayıları minimum ve maksimum değerler ile sınırlandırılabilir. Kümeleme yaparken kapasite sınırını da hesaplama dahil etmek için bu paket kullanılmıştır.

3.2.2 Verilerin Alınması ve Genel Düzenlemeler

Ümraniye Belediyesi Strateji Geliştirme ve Temizlik İşleri Müdürlükleriyle yapılan görüşmeler neticesinde çöp toplama süreciyle ilgili bilgiler alındı. Veri talebinde bulunularak kamyon güzergah bilgileri, mahalleler için tonaj bilgileri ve sokak konteyner sayıları öğrenildi.

Excel’de yapılan düzenleme ve birleştirme işlemleri sonucunda her mahalle için toplam metraj, konteyner sayısı ve 1 Haziran 2020 ve 30 Haziran 2021 tarihleri arasındaki tonaj ve sefer sayıları tek tabloda birleştirilmiştir.

Günlük güzergah tonaj verileri incelendiğinde, bazı tarihlerde bazı sokakların olağan dışında tonaj verisine sahip olduğu belirlenmiştir. Belediyeye yapılan görüşme neticesinde sokakta düzenlenen düğün merasimi, cenaze toplanması, yol çalışması gibi zorunlu sebeplerden sokağa giremeyen kamyonların çöp toplama hizmetini bir sonraki gün gerçekleştirdiği öğrenilmiştir. Bu durum veri setinde ayrık tonaj verisi değerleri oluşturmaktadır. Bu olağan dışı verilerin veri setinden çıkartılması gerekmektedir.

Her güzergaha ait tonaj verileri için çeyrekler arası uzaklıklar (IQR) belirlenerek alt ve üst limitler bulunmuştur. Bu limitlerin dışına çıkan ayrık tonaj verileri veri setinden çıkartılmıştır.

3.2.2.1 Geopy ile sokakların koordinatlarının belirlenmesi

Geopy bir coğrafi bilgi sistemi kütüphanesidir. Bu kütüphane sayesinde adresi bilinen herhangi bir sokağın gerçek konumu ve (x,y) koordinatları kolayca bulunabilmektedir (Bu işleme “geocoding” denir.). Ayrıca koordinatları bilinen iki

konum arasındaki jeodezik uzaklık (İki yüzey noktası arasındaki en kısa mesafeyi belirten eğrinin uzunluğudur.) da rahatlıkla hesaplanabilmektedir.

Kütüphane içerisinde yer alan “geocoders” metodu internete, “geocoding” servisine, bağlanarak veri çeker. Nominatim, Google Maps, Bing Maps, AzureMaps, ArcGis, TomTom Geocoder ve Yandex Maps gibi ücretli ve ücretsiz birçok servis bulunmaktadır. Ücretli servisler genellikle sorgu başına farklı fiyat politikası izlerken, ücretsiz servisler genellikle kullanıcıya günlük sorgu limitleri koyarak hizmet verirler. Bu tez çalışmasında ücretsiz bir servis olan, Türkiye’de kaliteli bir harita ağına sahip Yandex Maps servisine erişim sağlanmıştır.

Yazılan koda, örneğin “Pırlı Sokak, Ümraniye” sorgusu gönderildiğinde “Pırlı Sok., Elmalıkent Mah., Ümraniye, İstanbul, Türkiye” ve sokağın koordinatları olan “(29.092866, 41.039532)” çıktısı alınmaktadır. Bu şekilde Ümraniye ilçe sınırları içerisinde yer alan tüm sokak ve caddelerin gerçek konumları ve (x,y) koordinatları belirlenmiştir. Çıktı listesi Excel dosyası olarak kaydedilmiştir ve ilerleyen aşamalarda kullanılmak üzere saklanmıştır.

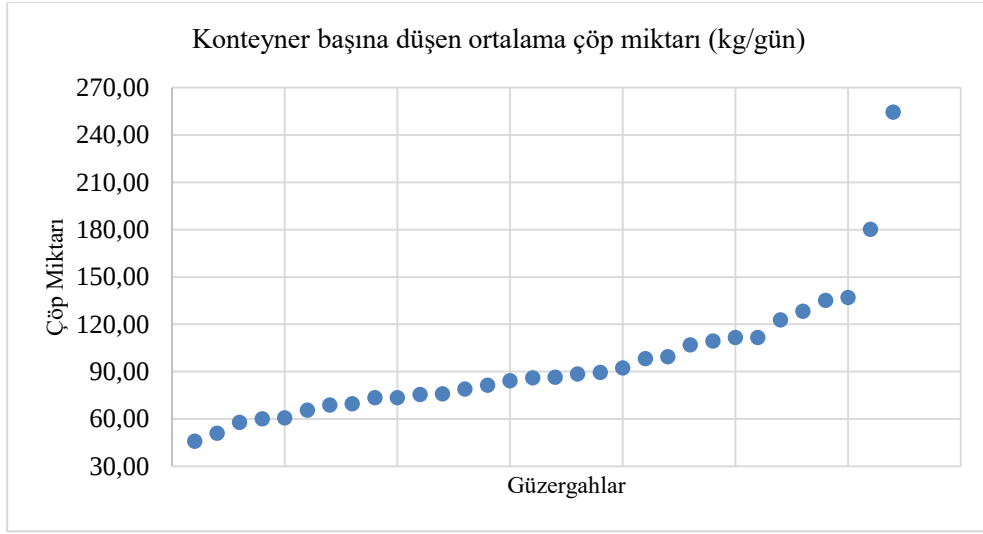
3.2.2.2 Sokak-Mahalle dönüşümü ve konteyner sayılarının atanması

Belediyeden alınan lokasyon verileri incelerken bazı kayıtların hatalı olduğu fark edilmiştir. Bazı sokalar birden fazla mahalleye kayıtlı gözükmemektedir. Problemin bu şekilde çözülmesi, aynı noktaya birden fazla gidilmesine sebep olacağı için tekrarlayan lokasyonlar veri setinden kaldırılmıştır. 3.2.2.1 başlığında oluşturulan Excel dosyasında gerekli düzenlemeler yapıldıktan sonra yeni bir sütun oluşturularak konteyner sayıları da belediyeden alınan veri setinden çekilerek eklenmiştir.

İlçe sınırları içerisinde çöp kovası konmasını gerektirecek kadar büyük olmayan, yeterince çöp üretmeyen veya kamyonların girebileceği kadar geniş olmayan sokaklar bulunmaktadır. Kamyonlar bu noktalarda konteyner olmasa da yol üstü çöp toplama işlemi yapmaktadır. Dolayısıyla bu durumun da probleme entegre edilmesi gerekmektedir. Bunun için konteyner sayısı 0 olan sokakların konteyner sayıları 0,2 olarak düzenlenmiştir. Böylece ilerleyen başlıklarda hesaplanacak katsayılarla çarpılarak çöp miktarı belirlenecek, araç rotalarına bu noktalar da dahil edilecektir.

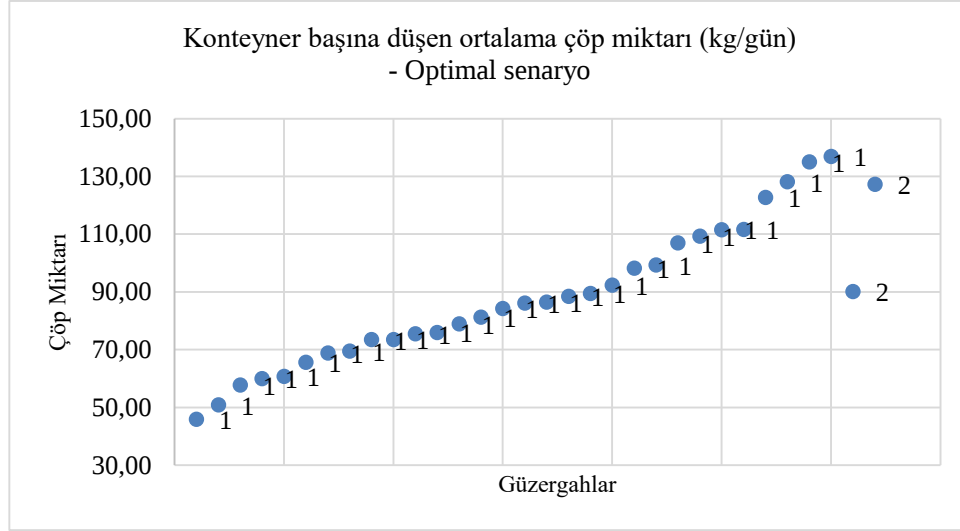
3.2.2.3 Sefer sayısı kavramı ve hesaplanması

Her güzergaha ait alt ve üst limitler bulunup bu limitler dışında kalan veriler veri seti dışına atıldıktan sonra, her güzergaha ait ortalama (μ) ve standart sapma (σ) değerleri hesaplanmıştır. $\mu+2\sigma$ ile her güzergahın üst limiti belirlenmiştir. Bu limit değerler her güzergahın kendi konteyner sayısına bölünerek konteyner başına düşen ortalama çöp miktarı hesaplanmıştır. Bu değerler Şekil 2’de gösterilmektedir.



Şekil 2. Konteyner Başına Düşen Ortalama Çöp Miktarı (kg/gün)

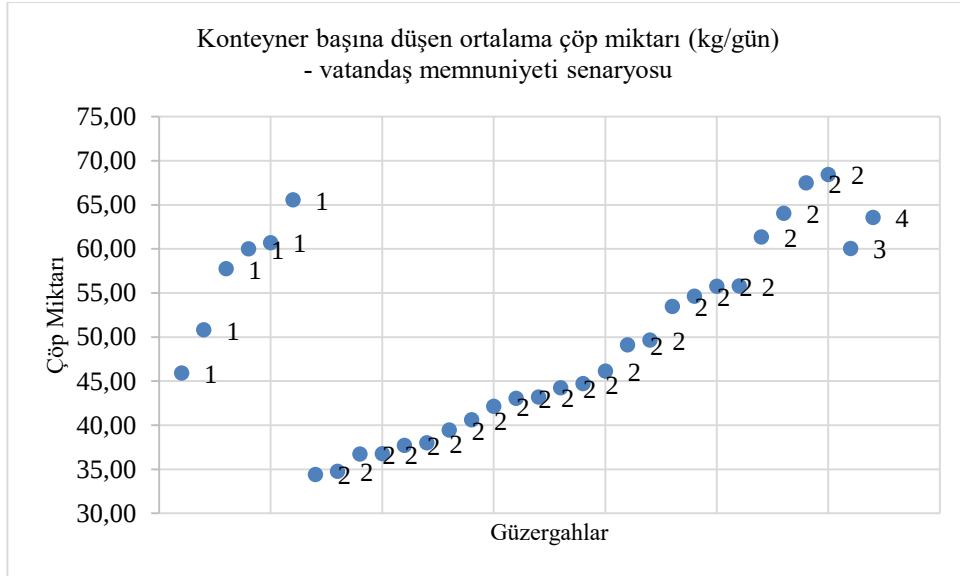
Şekil 2’de görüldüğü üzere grafiğin en sağındaki iki güzergah dışındaki tüm güzergahlar belirli bir aralığa yığılmışken bu iki güzergah sınırlar dışındadır. Bu güzergahlara da diğerleri gibi tek araçla hizmet vermek aşırı yüklenmeye sebep olabilir. Bu yüklenme sonucunda fazla çalışan görevlilerin performansı düşebilir veya araçlarda birtakım arızalar oluşabilir. Bu iki noktaya günde iki sefer gidilmesi ile konteyner başına düşen çöp miktarını, Şekil 3’te gösterildiği üzere 46 kg ve 136 kg aralığına sınırlandırabiliriz.



Şekil 3. Konteyner Başına Ortalama Çöp Miktarı (kg/gün) Optimal Senaryo

Çalışmanın ilerleyen kısımlarında, Senaryo 1 – Optimal Çözüm Yaklaşımında, Şekil 3’te gösterilen sefer sayıları ile probleme çözüm aranacaktır.

Öte yandan Şekil 3’te görüldüğü üzere 46-136 kg aralığı hala çok geniş bir aralıktır. Farklı güzergahlara günde farklı sefer sayılarında gidilmesi ile tüm problemde daha dar bir konteyner başına düşen çöp miktarı aralığı yakalanabilir. Böylece tüm ilçe sınırları içerisinde her mahalle ve sokak için benzer çöp toplama sıklığı elde edilir. Araçlar daha efektif kullanılabilir, çalışanlar her güzergahta eşit iş yüküne sahip olur.



Şekil 4. Konteyner Başına Ortalama Çöp Miktarı (kg/gün) V. M. Senaryosu

Şekil 4'te görüldüğü üzere sefer sayıları en az yoğunluğa sahip 6 güzergah için 1 tur, en yoğun iki güzergah için sırasıyla 3 ve 4 tur, kalan tüm güzergahlar için 2 tur olarak atandığında, konteyner başına düşen çöp miktarı aralığı 35-68 kg aralığına kadar daralmaktadır. Çalışmanın ilerleyen kısımlarında, Senaryo 2 – Memnuniyeti Gözetken Çözüm Yaklaşımında, Şekil 4'te gösterilen sefer sayıları ile probleme çözüm aranacaktır.

3.2.3 Senaryo 1: Yerel Maks – Optimal Çözüm Yaklaşımı

Bu senaryoda çöp kamyonlarının rotalanması için tüm sokakların günlük tonajının yerel maksimumlarını (yerel maks) temel alan, en uygun çözüm yaklaşımını benimsemiş bir yöntem uygulanmıştır. Şekil 3'te gösterilen sefer sayılarıyla probleme çözüm aranmıştır. Yaklaşımın genel hatları şu şekildedir:

- Her güzergaha ait tonaj verilerinin alt ve üst limitleri belirlenip limit dışına taşan ayırık değerler veri setinden çıkarıldıktan sonra her güzergaha ait maksimum tonaj değerleri bulunmuştur. Bulunan maksimum değerler her güzergahın kendi konteyner sayısına bölünerek her güzergah için maksimum çöp üretimi durumunda (yerel maksimum), konteyner başına düşen çöp miktarı (güzergah tonaj katsayısı) hesaplanmıştır.
- Hesaplanan değerler her güzergah için hesaplanan (Şekil 3'te gösterilen) sefer sayılarına bölünerek, her güzergah için tek seferde konteyner başına düşen çöp miktarı katsayısı bulunmuştur.
- Hesaplanan katsayılar problemdeki tüm sokaklara dağıtıldı. Sokaktaki konteyner sayısı, sokağın ait olduğu güzergahın katsayısı çarpılarak sokak çöp miktarları hesaplanmış oldu. Algoritmaların hata vermemesi için değerler en yakın tam sayıya yuvarlanmıştır.
- OR-Tools veri modelini oluşturmak için istediği girdiler oluşturulmuştur.
 - Daha önce belirlenen lokasyon koordinatları, OR-Tools'a $[n \times 1]$ matrisi ile (n =lokasyon sayısı) girilmiştir.
 - Tez çalışmasının bu bölümünde hesaplanıp en yakın tam sayıya yuvarlanan sokak çöp miktarları OR-Tools'a $[n \times 1]$ boyutlu talep matrisi ile girilmiştir.

- $[n \times n]$ boyutlu karşılıklı mesafe matrisi lokasyonların birbirlerine olan jeodezik uzaklıklarından oluşmaktadır. Geopy ile oluşturulmuştur. Değerler km'den m'ye çevrilmiştir. Küsuratlı değerler en yakın tam sayıya yuvarlanmıştır.
- Araç sayısı ve kapasiteleri belediyeden alınan bilgiler dahilinde eşit kapasitelerde girilmiştir.
- Probleme ilgili temel girdiler sonrasında OR-Tools'a algoritmanın çalışmasıyla ilgili girdiler girilmiştir.
 - Çalışmada kullanılan ilk sonuç üretme stratejileri ve yerel arama seçenekleri Tablo 1'de gösterilmiştir. Bu seçenekler ikili kombinasyonlar halinde tek tek denenerek problem için en iyi sonucu veren algoritma ikilisine yanıt aranmıştır.
 - Tez çalışmasında 100, 500, 1000 gibi zaman limitleri ile deneyleri yapılmıştır. Zaman limiti artırıldığında çözüm kalitesini artırıp artırmadığı gözlemlenmiştir.
- Algoritmanın çalışması için gerekli olan tüm girdiler de girildikten sonra OR-Tools çalıştırılır ve sonuçlar ekrana yazdırılır.
- Rotalara ilişkin güzergah bilgileri, amaç fonksiyonu, çalışma süresi, mesafe gibi bilgiler Excel dosyasına kaydedilmiştir. Saçılım grafiği kullanılarak her bir araca ve tüm probleme ait araç rotaları oluşturulmuştur. Koordinat düzlemindeki her bir nokta sokağı ifade etmektedir.

3.2.4 Senaryo 2: Yerel Maks– Vatandaş Memnuniyetini Gözeten Yaklaşım

İlk senaryoda iki güzergaha ek sefer eklenerek konteynerlerdeki ortalama çöp birikme miktarları düzenlense de iki güzergah arasındaki fark neredeyse üç kata kadar ulaşabilmektedir. Aynı işi gören, aynı kaynağı kullanan iki güzergah arasındaki bu farklılık mahalleler arasında çöp toplanma sıklığında dengesizliğe, belediye araç ve personeli arasında da iş yükü farklılığına sebep olacaktır. Dolayısıyla bu senaryoda yoğun mahalleler için sefer sayıları düzenlenerek konteyner başına ortalama çöp miktarı farkı düşürülmüştür. Şekil 4'te hesaplanıp gösterilen sefer sayılarıyla probleme çözüm aranmıştır. Yaklaşımın genel hatları şu şekildedir:

- Her güzergaha ait tonaj verilerinin alt ve üst limitleri belirlenip limit dışına taşan ayırık değerler veri setinden çıkarıldıktan sonra her güzergaha ait maksimum tonaj değerleri bulunmuştur. Bulunan maksimum değerler her güzergahın kendi konteyner sayısına bölünerek her güzergah için maksimum çöp üretimi durumunda (yerel maksimum), konteyner başına düşen çöp miktarı (güzergah tonaj katsayısı) hesaplanmıştır.
- Hesaplanan değerler her güzergah için hesaplanan (Şekil 4'te gösterilen) sefer sayılarına bölünerek, her güzergah için tek seferde konteyner başına düşen çöp miktarı katsayısı bulunmuştur.
- Hesaplanan katsayılar problemdeki tüm sokaklara dağıtıldı. Sokaktaki konteyner sayısı, sokağın ait olduğu güzergahın katsayısı çarpılarak sokak çöp miktarları hesaplanmış oldu. Algoritmaların hata vermemesi için değerler en yakın tam sayıya yuvarlanmıştır.
- OR-Tools veri modelini oluşturmak için istediği girdiler oluşturulmuştur.
 - Daha önce belirlenen lokasyon koordinatları, OR-Tools'a $[n \times 1]$ matrisi ile (n =lokasyon sayısı) girilmiştir.
 - Tez çalışmasının bu bölümünde hesaplanıp en yakın tam sayıya yuvarlanan sokak çöp miktarları OR-Tools'a $[n \times 1]$ boyutlu talep matrisi ile girilmiştir.
 - $[n \times n]$ boyutlu karşılıklı mesafe matrisi lokasyonların birbirlerine olan jeodezik uzaklıklarından oluşmaktadır. Geopy ile oluşturulmuştur. Değerler km'den m'ye çevrilmiştir. Küsuratlı değerler en yakın tam sayıya yuvarlanmıştır.
 - Araç sayısı ve kapasiteleri belediyeden alınan bilgiler dahilinde eşit kapasitelerde girilmiştir.
- Probleme ilgili temel girdiler sonrasında OR-Tools'a algoritmanın çalışmasıyla ilgili girdiler girilmiştir.
 - Çalışmada kullanılan ilk sonuç üretme stratejileri ve yerel arama seçenekleri Tablo 1'de gösterilmiştir. Bu seçenekler ikili kombinasyonlar halinde tek tek denenerek problem için en iyi sonucu veren algoritma ikilisine yanıt aranmıştır.

- Tez çalışmasında 100, 500, 1000 gibi zaman limitleri ile deneyleri yapılmıştır. Zaman limiti artırıldığında çözüm kalitesini artırıp artırmadığı gözlemlenmiştir.
- Algoritmanın çalışması için gerekli olan tüm girdiler de girildikten sonra OR-Tools çalıştırılır ve sonuçlar ekrana yazdırılır.
- Rotalara ilişkin güzergah bilgileri, amaç fonksiyonu, çalışma süresi, mesafe gibi bilgiler Excel dosyasına kaydedilmiştir. Saçılım grafiği kullanılarak her bir araca ve tüm probleme ait araç rotaları oluşturulmuştur. Koordinat düzlemindeki her bir nokta sokağı ifade etmektedir.

3.2.5 K-Ortalamlar ve OR-Tools Sıralı Çözüm

Senaryo 1 ve Senaryo 2’den farklı olarak bu çözüm yönteminde sezgisel veya meta-sezgisel algoritmalar kullanılmadan önce makine öğrenmesine dayanan k-ortalamlar kümeleme algoritmasıyla güzergahlar kümelenmiştir ve sonrasında her bir güzergah kendi içerisinde ayrı ayrı rotalanmıştır. İki aşamalı ARP olarak değerlendirilebilir.

Diğer bileşenlerin nasıl hesaplandığı ve algoritmanın çalıştırılması ile ilgili genel işleyiş aşağıdaki gibidir:

- Tonaj verilerine ait alt ve üst limitler belirlenip limit dışına çıkan veriler veri setinden çıkartıldıktan sonra kalan veri setinde her güzergah için $u+2std$ ile yeni üst limit bulunmuştur. Bu üst limit her güzergahın kendi konteyner sayısına bölünerek güzergahtaki konteyner başına düşen çöp miktarı bulunmuştur.
- Bulunan değerler her güzergah için hesaplanan sefer sayılarına bölünerek, her güzergah için tek seferde konteyner başına düşen çöp miktarı katsayısı hesaplanmıştır.
- Hesaplanan katsayılar problemdeki tüm sokaklara dağıtıldı. Sokaktaki konteyner sayısı ile, sokağın ait olduğu güzergahın katsayısı çarpılarak sokak çöp miktarları hesaplanmış oldu. Algoritmaların hata vermemesi için değerler en yakın tam sayıya yuvarlanmıştır.
- K-Ortalamlar Kümeleme algoritması normal şartlarda sadece uzaklık ve yakınlığa göre kümeleme yapmaktadır. Standart k-ortalamlar ile probleme

özüm arandığında küme içerisinde çok fazla miktarda öpü olan sokak da çok az miktarda öpü olan sokak da aynı ağırlığa sahip olmaktadır. Bu tez alışmasında öp miktarlarına göre ağırlıklandırma yapmak için algoritmaya her sokağın tek bir sefer yazılması yerine talep miktarı kadar tekrar tekrar yazılmasıyla kümeleme algoritması manipüle edilmiştir.

- Örneğın Acısu Caddesinin 168 kg'lık öp miktarı bulunmaktadır. Algoritmaya Acısu Caddesi tek bir sefer girildiğinde yalnızca 34 kg'lık öpü bulunan Buse Sokağıyla aynı ağırlığa sahip olacaktır. Yapılan manipölasyonla Acısu Caddesi 168 defa, Buse Sokağı 34 defa girildiğinde lokasyonların küme içerisindeki ağırlığı da önemli hale gelecektir ve standart k-ortalamlar algoritması ağırlıklandırılmış k-ortalamlar kümeleme algoritmasına dönüşecektir.
- Ancak her bir lokasyonun talebi kadar girilmesi problem boyutunu 2997 sokaktan 323802 lokasyon dilimine dönüştürecektir. Bu dramatik artışla algoritmanın alışması zorlaşacaktır. Bu yüzden öp miktarları 25'lik sistemde ölçeklendirilmiştir. Örneğın Acısu Caddesinin 168 kg'lık talebi (150-175) aralığına denk geldiğı için ağırlık katsayısı 7 olmuştur. Tüm lokasyonların ölçek değeri kadar algoritmaya girilmesiyle problem büyüklüğü 14548'e düşürölmüştür. Oluşturulan sokak listesi ve ağırlık katsayıları Excel tablosuna kaydedilmiştir.
- K-Means-Constrained algoritması girdi olarak koordinat listesi, küme sayısı, minimum ve maksimum küme boyutu, n_init ve max_iter bilgilerini istemektedir.
 - Küme sayısı bu tez alışması için araç sayısı olan 32 olarak belirlenmiştir.
 - Minimum ve maksimum küme boyutunu belirlemek kümenin toplam eleman sayısı problemdeki araç sayısına bölünmüştür. Çıkan değeri %20 ıkartılıp, eklenerek minimum ve maksimum küme boyutları belirlenmiştir.
 - N_init ve max_iter değeri için farklı farklı değeri girilerek algoritma alıştırılmıştır ve en iyi sonucu veren değeri aranmıştır.

- Algoritma çıktısı Excel dosyasına kaydedilerek, pivot tablo yardımıyla gruplandırılmıştır. İnceleme sonucunda bazı lokasyonların ağırlık dilimlerinin iki farklı kümeye atandığı anlaşılmıştır. Aynı lokasyonun iki farklı kümede olması problemin çözümünde hatalara sebep olacaktır. Hatalı lokasyonların tüm problem içerisinde çok küçük bir orana sahip olması (%0,17) sebebiyle elle iyileştirme yapılması uygun görülmüştür.
 - Yapılan iyileştirme işleminde azınlık dilimler çoğunluk dilimlerle aynı kümeye atanmıştır. Örneğin, 85 numaralı lokasyonun 11 diliminin 9'u 27. kümeye atanmışken 2 dilim 7. kümeye atanmıştır. Yapılan iyileştirme ile bu iki dilim de 27. kümenin elemanı haline getirilmiştir.
- Elle iyileştirme sonucunda nihai kümeler ve elemanları belirlenmiştir. Kümelere ayrılan sokaklar ayrı ayrı problemler olarak ele alınmıştır ve OR-Tools paketi yardımıyla ARP çözülmüştür.
 - Diğer senaryolardaki OR-Tools çözümlerinin aksine bu kısımda çok araçlı tek bir problem yerine tek araçlı birden fazla problem bulunmaktadır.
- Girdi olarak yine her bir alt problemin (kümenin) lokasyonların koordinatlarını, taleplerini (tek araçla tüm talep karşılayabileceği için tüm talepler eşit alınmıştır), karşılıklı mesafe matrisini, araç sayısını (tek araç) ve kapasitesini ve hangi lokasyonun merkez depo olduğu bilgisini alır; veri modelini oluşturur.
- Problemle ilgili temel girdiler sonrasında OR-Tools'a ilk sonuç üretme stratejisi ve arama seçeneği girilirken bu aşamada birden fazla yöntem denemek yerine ilk iki senaryoda iyi sonuçlar verdiği anlaşılan strateji ve seçenek kullanılmıştır. Ayrıca zaman limiti daha küçük ölçekli bir problem olduğu için 100 olarak belirlenmiştir.
- Algoritmanın çalışması için gerekli olan tüm girdiler de girildikten sonra OR-Tools çalıştırılmıştır ve sonuçlar ekrana yazdırılmıştır. Rotalara ilişkin güzergah bilgileri, amaç fonksiyonu, çalışma süresi, mesafe gibi bilgiler Excel dosyasına kaydedilmiştir.

BÖLÜM 4. UYGULAMA – ÜMRANIYE BELEDİYESİ ÖRNEĞİ

Bu tez çalışmasında Ümraniye Belediyesinin katı atık toplama rota optimizasyonu ARP olarak ele alınarak incelenmiştir. Bu başlıkta alınan verilerin içeriği, verilerin incelenmesi, düzenlenmesi ve farklı senaryolar altında OR-Tools ve K-Ortalamalar Kümeleme algoritması tabanlı çözümü uygulanmıştır.

4.1 VERİLERİN TEMİNİ VE VERİ TEMİZLEME ÖNİŞLEMLERİ

Belediye ile yapılan görüşmeler sonucunda güzergah bilgileri, mahalleler için tonaj bilgileri ve sokak konteyner sayıları öğrenilmiştir. Alınan verilere ait Excel dosyalarında kullanılan format aşağıdaki gibidir.

Tablo 2. Alınan Güzergah Verileri Formatı

METRAJ	MAHALLE	CADDE-SOKAĞI	KONTEYNER ADET VE KAPI NUMARASI	SAYI	BÖLGE ARACI
...	...	...	...	...	...

3327 satırlık veri içermektedir, Ek-1’de sunulmuştur. Tablo 2’de metraj sütunu sokakların uzunluklarını, mahalle sütunu sokağın belediye veri tabanında bağlı olduğu mahalleyi, cadde-sokağı sütunu sokak veya cadde ismini belirtirken sayı sokakta bulunan konteyner sayısını belirtmektedir. Ayrıca bölge aracı sütunuyla belediyenin mevcut sistemde kullandığı güzergahlar belirtilmiştir. İlçe halihazırdaki uygulamada 36 bölgeye/güzergaha bölünmüştür. Bu güzergahlardan üçü ana arterlerden, biri organize bölgesinden oluşmaktadır. Tüm güzergahların çöpü belirli plan doğrultusunda toplanırken bu üç ana arter gün içinde sürekli olarak gezilerek çöplerin birikmesine müsaade edilmemektedir. Bu yüzden bu bölgelerden alınan verilerin veri setinden çıkartılması uygun görülmüştür.

Güzergah ve sokak tablosu incelendiğinde bazı sokakların aynı anda iki farklı mahalleye kaydedildiği dolayısıyla lokasyon ve konteyner sayılarında çift sayım işlemi olduğu gözlenmiştir. Bu hatayı ortadan kaldırmak için tekrarlayan sokaklar elimine edilerek yeni liste oluşturulmuştur. Bu listeye göre rotalama probleminde 2998 lokasyon ve 6233 konteyner bulunmaktadır.

Tablo 3. Alınan Güzergah Tonaj Verileri Formatı

TARİH	PLAKA	AD SOYAD	BÖLGESİ	1.SEFER TONAJ	2.SEFER TONAJ	3.SEFER TONAJ	TOPLAM TONAJ	SEFER SAYISI
...	...	...	...	...	...	...	...	...

14256 satırlık veri içermektedir, Ek-2’de sunulmuştur. Tablo 3’te her satır bir güzergaha ait belirli tarihteki tonaj verisini ifade etmektedir. 1 Haziran 2020 ve 30 Haziran 2021 tarihleri arasında her gün için her bir güzergahın çöpünün hangi araçla, hangi şoför tarafından kaç seferde toplandığı bilgisini göstermektedir.

Veriler incelendiğinde tüm güzergahlar içerisinde hiç sefer yapılmamış 210 gün, tek sefer yapılmış 1177 gün, iki tur yapılmış 12444 gün ve üç tur yapılmış 425 gün olduğu gözlenmiştir.

Ayrıca tonaj verilerinde organize bölgesi için yeterli veri olmamasından dolayı güzergah listesinden ana arterlerden sonra organize bölgesi de çıkartılmıştır. Problem 32 bölge, 32 araç olarak ele alınacaktır.

4.1.1 Veri setinin düzenlenmesi.

Tablo 4. Güzergahlara Ait Tanımlayıcı İstatistikler (Kısaltılmış Örnek Tablo)

Tarih	ATAKENT	ESENEVLER	İSTİKLAL - NAMİKKEMAL	MEHMET AKİF	NECİP FAZIL
"Tarih"	"Tonaj"	"Tonaj"	"Tonaj"	"Tonaj"	"Tonaj"
Q1	13.880	15.960	13.880	15.400	16.920
Q3	15.840	17.400	15.140	17.280	18.340
IQR	1.960	1.440	1.260	1.880	1.420
LB	10.940	13.800	11.990	12.580	14.790
UB	18.780	19.560	17.030	20.100	20.470

Tablo 4’te bazı güzergahlara ait tanımlayıcı istatistikler paylaşılmıştır. Her güzergaha ait çeyreklik bilgileri Q1 ve Q3 satırlarında hesaplanarak IQR satırında çeyrekler arası uzaklıklar bulunmuştur. Sonrasında alt ve üst sınır (LB, UB) belirlenerek bu limitler arasında olmayan veriler tablodan çıkartılmıştır.

Tablo 5. Güzergahlara Ait Tanımlayıcı İstatistikler (Düzenlenmiş, Kısaltılmış Örnek Tablo)

Tarih	ATAKENT	ESENEVLER	İSTİKLAL - NAMIKKEMAL	MEHMET AKİF	NECİP FAZİL
"Tarih"	"Tonaj"	"Tonaj"	"Tonaj"	"Tonaj"	"Tonaj"
Ortalama Tonaj	14.768	16.779	14.525	16.445	17.799
Max Tonaj	17.760	19.500	16.590	20.080	20.440
Min Tonaj	11.800	14.080	12.000	12.820	15.120
Std. Sapma	1.285	1.037	919	1.432	1.039
u-2STD	12.198	14.705	12.688	13.581	15.720
u+2STD	17.338	18.852	16.362	19.308	19.877

Limitler dışına taşan verilerin silinmesi sonrasında kalan verilere ait tanımlayıcı istatistikler Tablo 5’te gösterilmiştir. Her güzergaha ait ortalama, maksimum, minimum tonajlar hesaplanmıştır. Standart sapmalar bulunarak her güzergaha ait alt ve üst sınır belirlenmiştir.

4.1.2 Geopy ile koordinatların belirlenmesi.

Python Geopy kütüphanesi kullanılarak Yandex Haritalar üzerinden problemdeki tüm cadde ve sokaklara ait konumlar ve koordinatlar belirlenmiştir. Koda ait örnek girdi ve çıktı tablosu Tablo 6’da gösterilmiştir.

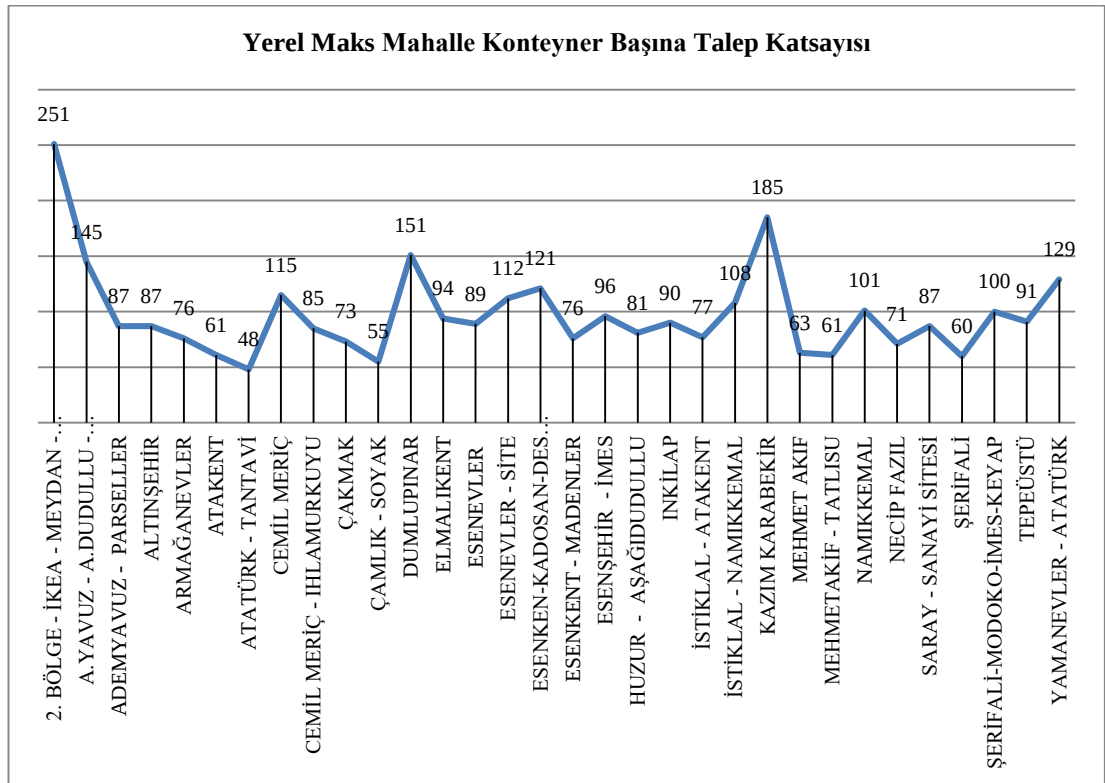
Tablo 6. Koordinatların Belirlenmesi Kodu Girdi Çıktı Tablosu (Kısaltılmış Örnek Tablo)

	Girdi	Çıktı		
	Adres	Konum	Xkord	Ykord
0	PINARCIK SOKAK, Ümraniye	Pınarcık Sok., Tatlısu Mah., Ümraniye, İstanbul, Türkiye	41,00257	29,133991
1	PINARLI SOKAK, Ümraniye	Pınarlı Sok., Tepeüstü Mah., Ümraniye, İstanbul, Türkiye	41,01748	29,131224
2	PIRILTI SOKAK, Ümraniye	Pırılı Sok., Elmalıkent Mah., Ümraniye, İstanbul, Türkiye	41,03953	29,092866
3	PIRLANTA SOKAK, Ümraniye	Pırlanta Sok., Esenşehir Mah., Ümraniye, İstanbul, Türkiye	41,00612	29,168127
...	...	...	...	...

Ayrıca “3.2.2.3 Sefer sayısı kavramı ve hesaplanması” başlığında anlatılan yöntem ile üç farklı senaryo/metot için sefer sayıları hesaplanmıştır. Tablo 6’ya bu sefer sayıları ve sokaklara it konteyner sayıları da eklenerek üç farklı veri girdi dosyası oluşturulmuştur. Bu tablolar ilerleyen bölümlerde detaylandırılacaktır.

4.2 SENARYO 1: YEREL MAKS – OPTİMAL ÇÖZÜM YAKLAŞIMI

Yerel maks senaryosu, isminden de anlaşılacağı üzere her güzergah için kendi maksimum tonaj verisinin kendi konteyner sayısına bölünerek yerel maksimum taleplerin gerçekleştirildiği senaryodur. Optimizasyonu yapılacak problem için en kötü senaryo göz önünde bulundurularak en fazla çöpün üretildiği durum incelenmektedir. Yapılan hesaplama sonucu maksimum talep durumunda, her bir mahallenin konteyner başına çöp miktarı Şekil 5’te gösterilmiştir.



Şekil 5. Yerel Maks Mahalle Konteyner Başına Talep Katsayısı

Sokak listesi tablosuna Şekil 5’te hesaplanan maksimum talep katsayıları ve Şekil 3’te gösterilen hesaplanmış optimum sefer sayıları da eklenerek aşağıdaki nihai tablo, Tablo 7 oluşturulmuştur.

Tablo 7. Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Senaryo 1

	Konum	Xkord	Ykord	D.Konteyner Sayısı	R.Talep Katsayısı	Sefer Sayısı	KS*TK/SS
0	Hekimbaşı Katı Atık Aktarma Merkezi	41,05846964	29,11273772	0	0	0	0
1	3016. Sok., Esenevler Mah., Ümraniye, İstanbul, Türkiye	41,008843	29,097232	0,2	112	1	22
2	Neval Sok., Necip Fazıl Mah., Ümraniye, İstanbul, Türkiye	41,015022	29,164866	12	251	2	1506
3	Acun Sok., Ümraniye, İstanbul, Türkiye	41,008673	29,130559	10	63	1	630
...	...	...	...	...	...	...	...

Tablo 7 OR-Tools’un çalışması için gerekli tüm verileri içeren özet tablonun küçük bir örneğidir. Konum, Xkord ve Ykord sütunları Yandex Haritalar’dan otomatik çekilmiştir. D.Konteyner Sayısı sütunu güzergahtaki düzenlenmiş konteyner sayılarını içermektedir. R.Talep Katsayısı Şekil 5’ten hesaplanarak çekilmiştir. Sefer sayıları Şekil 3’te hesaplanmış ve çekilmiştir. Nihai talep (Konteyner Sayısı)*(R.Talep Katsayısı)/(Sefer Sayısı) formülüyle hesaplanmıştır ve küsuratlı değerler en yakın tam sayıya yuvarlanmıştır.

OR-Tools çalıştırılması için gerekli olan girdiler: lokasyonlar arası karşılıklı mesafe matrisi, lokasyonlara ait talep matrisi, araç sayısı ve kapasiteleri ve depoya konumudur. Tablo 7’den gerekli sütunlar alınarak girdiler aşağıdaki gibi hazırlanmıştır.

Tablo 8. Lokasyonlar Arası Karşılıklı Mesafe Matrisi (Kısaltılmış)- Senaryo 1

	0	1	2	...	...	...	2997
0	0	6399	6747	...	...	...	6485
1	6399	0	2104	...	...	...	335
2	6747	2104	0	...	...	...	2431
...	...	...	...	0	...	...	...
...	...	...	...	...	0	...	...
...	...	...	...	...	...	0	...
2997	6485	335	2431	...	...	...	0

Tablo 8, Tablo 7'nin Xkord ve Ykord sütunlarını kullanarak Python Geopy kütüphanesi yardımıyla hesaplanmış 2998x2998 boyutlu matristir, Ek-3'de sunulmuştur. Hücre değerleri iki lokasyon arasındaki jeodezik uzaklığı metre cinsinde ifade etmektedir.

Tablo 9. Talep Matrisi (Kısaltılmış)- Senaryo 1

id	0	1	2	3	...	...	...	2997
talep	0	600	426	1672	...	...	...	20

Talep matrisi Tablo 7'nin KS*TK/SS sütunudur. 1x2997 boyutundadır ve hücre değerleri lokasyona ait çöp miktarı kg cinsinden ifade etmektedir.

Ana depo 0. lokasyon olarak belirlenmiştir. Tüm araçlar depodan tura başlayacak ve turu depoda bitirecektir.

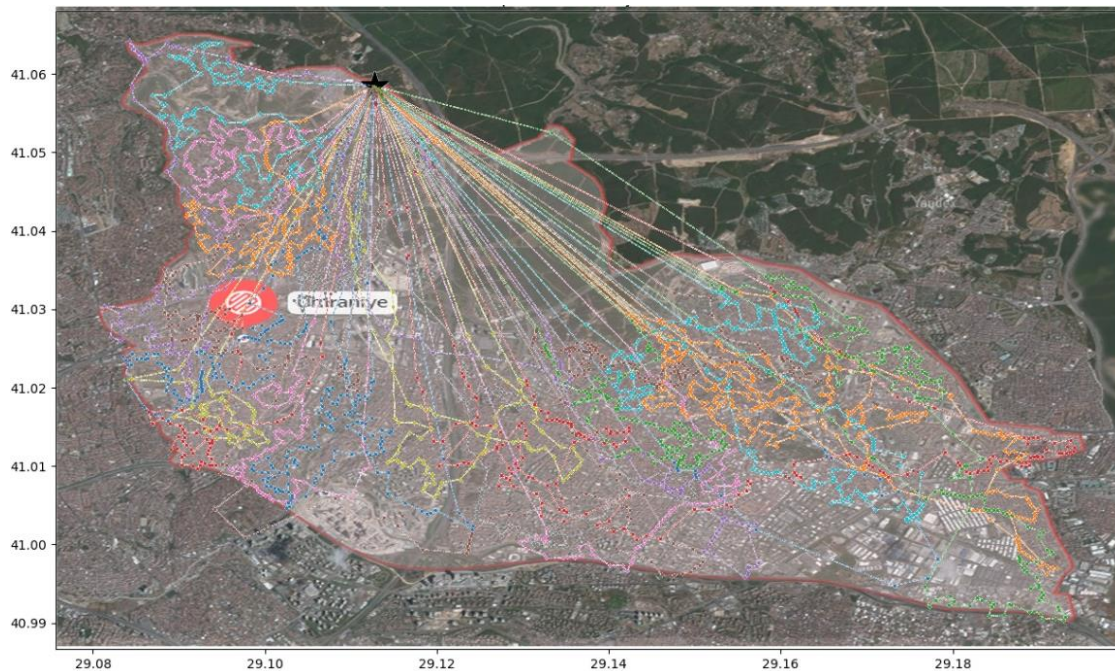
Araç sayısı 32'dir. Tüm araçlar eşit kapasiteye sahiptir. Araç kapasiteleri algoritmanın toplam kat edilen mesafeyi azaltmak için araç sayısında tasarrufa gitmemesi ve problemin tam olarak 32 araçla çözülmesi için (Toplam Talep)/(Araç Sayısı = 32) formülüyle 18 Ton olarak hesaplanmıştır ve OR-Tools'a girilmiştir.

Tüm girdilerin belirlenmesiyle ilk sonuç üretme, yerel arama stratejisi ve limitler de belirlenerek OR-Tools çalıştırılmıştır. Sonuçlar aşağıdaki tabloda özetlenmiştir.

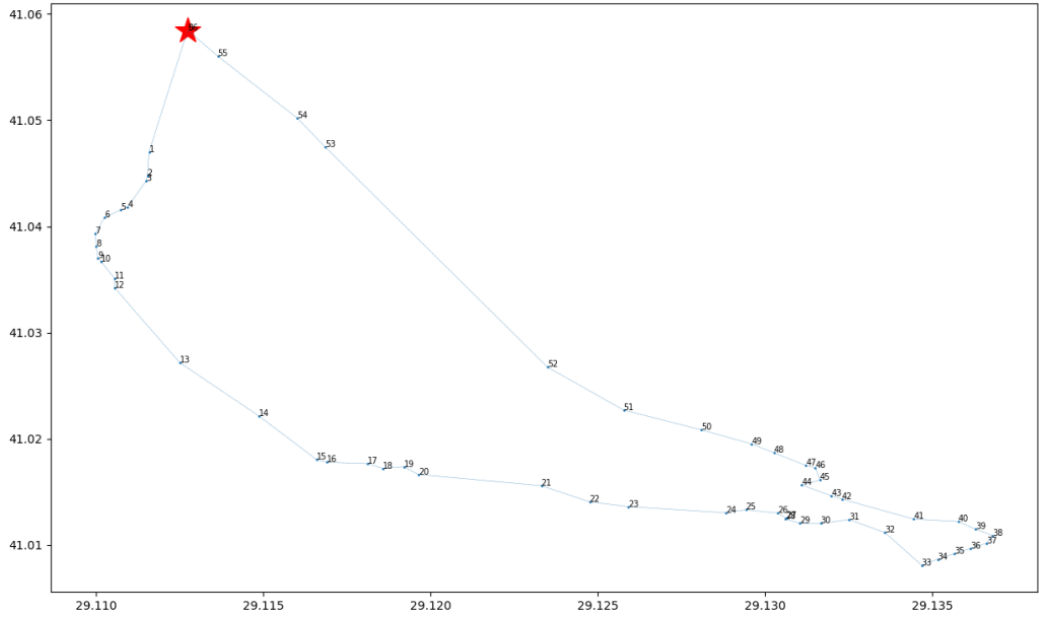
Tablo 10. OR-Tools Çalıştırma Sonuçları - Senaryo 1

İlk Sonuç Üretme Stratejisi	Yerel Arama Stratejisi	Amaç Fonksiyonu (m)	Çalışma Süresi (sn)	Çözüm Kalitesi
En Ucuz Yay	Rehberli Yerel Arama	525.582	640	99,97%
En Ucuz Yay	Benzetimli Tavlama	525.443	634	100,00%
En Ucuz Yay	Tabu Arama	525.443	621	100,00%
En Ucuz Yay	Açgözlü İniş	525.471	630	99,99%
Tasarruf	Rehberli Yerel Arama	571.122	677	92,00%
Tasarruf	Benzetimli Tavlama	571.352	646	91,96%
Tasarruf	Tabu Arama	571.389	676	91,96%
Tasarruf	Açgözlü İniş	571.122	604	92,00%
Christofides	Rehberli Yerel Arama	558.127	645	94,14%
Christofides	Benzetimli Tavlama	558.522	662	94,08%
Christofides	Tabu Arama	558.522	657	94,08%
Christofides	Açgözlü İniş	558.127	675	94,14%
En Kısıtlı Yay	Rehberli Yerel Arama	535.856	665	98,06%
En Kısıtlı Yay	Benzetimli Tavlama	534.379	629	98,33%
En Kısıtlı Yay	Tabu Arama	534.471	627	98,31%
En Kısıtlı Yay	Açgözlü İniş	534.757	640	8,26%

Tablo incelendiğinde En Ucuz Yay ile üretilen ilk sonucun Tabu Arama veya Benzetimli Tavlama algoritmasıyla geliştirilmesi en iyi sonucu vermektedir. Ancak çözüm süresi de göz önünde bulunduğunda Tabu Arama algoritmasının kısmen daha iyi sonuç verdiği söylenebilir.



OR-Tools'tan elde edilen tek bir araca ait örnek güzergah haritası ve çıktısı ve örnek bir araç için aşağıdaki gibidir. Şekil 7'de en iyi deney sonucuna ait rastgele seçilen 8 nolu aracın güzergah haritası gösterilmiştir. Kırmızı yıldız ana depoyu, noktalar ziyaret edilen lokasyonları, sayılar ziyaret sırasını göstermektedir.



Şekil 7. En İyi Deneye Ait 8. Nolu Aracın Güzergah Haritası – Senaryo 1

“Route for vehicle 8:

0 Load(0) -> 1297 Load(18) -> 1001 Load(37) -> 806 Load(56) -> 1930 Load(74) -> 1727 Load(92) -> 2476 Load(272) -> 1664 Load(290) -> 1980 Load(380) -> 1525 Load(398) -> ... (*güzergah listesi kısaltılmıştır). -> 1735 Load(17954) -> 2672 Load(17972) -> 0 Load(17972)

Distance of the route: 12960 m

Load of the route: 17972 kg”

8. Araca ait bu örnek çıktı, rotanın 0. konumdan başladığını, 1297. Konuma giderek 18 kg’lık çöp alındığını, daha sonra 1001. konuma giderek 19 kg’lık daha çöp alındığı ve araç kapasitesinin 37 kg olduğunu anlatmaktadır. Araç 2672. konuma geldiğinde araç kapasitesi 17.972 kg olmuştur ve araç kapasitesi dolduğu için başlangıç konumuna geri dönerek turu tamamlamıştır. Rotanın toplam mesafesi 12.960 m, rotada toplanan toplam çöp miktarı 17.972 kg’dır.

4.3 SENARYO 2 – YEREL MAKS VATANDAŞ MEMNUNİYETİ ODAKLI YAKLAŞIM

Bu senaryoda 4.2’de incelenen yerel maks senaryosundan farklı olarak güzergahlar sefer sayılarına bölünerek çöplerin daha sık toplandığı, birikmenin önüne geçmeyi amaçlayan durum incelenmiştir.

Şekil 5’te hesaplanan maksimum talep durumunda her mahalle için konteyner başına çöp miktarı katsayısı bu senaryoda da kullanılmıştır. Bu katsayılar sokak listesi tablosuna eklenmiştir. Ayrıca Şekil 4’te gösterilen, 1-4 aralığında değişen sefer sayıları da tabloya eklenmiştir ve aşağıdaki nihai tablo, Tablo 11 oluşturulmuştur.

Tablo 11 Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Senaryo 2

	Konum	Xkord	Ykord	D.Konteyner Sayısı	R.Talep Katsayısı	Sefer Sayısı	KS*TK/SS
0	Hekimbaşı Katı Atık Aktarma Merkezi	41,0584696 4	29,1127377 2	0	0	0	0
1	3016. Sok., Esenevler Mah., Ümraniye, İstanbul, Türkiye	41,008843	29,097232	0,2	112	2	11
2	Neval Sok., Necip Fazıl Mah., Ümraniye, İstanbul, Türkiye	41,015022	29,164866	12	251	4	753
3	Acun Sok., Ümraniye, İstanbul, Türkiye	41,008673	29,130559	10	63	1	630
4	Asil Cad., Topağacı Mah., Ümraniye, İstanbul, Türkiye	41,049326	29,105775	3	185	3	185
...	...	...	...	...	...	...	...

Tablo 11 OR-Tools’un çalışması için gerekli tüm verileri içeren özet tablonun küçük bir örneğidir. Konum, Xkord ve Ykord sütunları Yandex Haritalar’dan otomatik çekilmiştir. D.Konteyner Sayısı sütunu güzergahtaki düzenlenmiş konteyner sayılarını içermektedir. R.Talep Katsayısı Şekil 5’te hesaplanarak çekilmiştir. Sefer sayıları Şekil 4’te hesaplanmış ve çekilmiştir. Nihai talep (Konteyner Sayısı)*(R.Talep Katsayısı)/(Sefer Sayısı) formülüyle hesaplanmıştır ve küsuratlı değerler en yakın tam sayıya yuvarlanmıştır.

Tablo 12 Lokasyonlar Arası Karşılıklı Mesafe Matrisi (Kısaltılmış)

	0	1	2	...	...	...	2997
0	0	6399	6747	...	...	...	6485
1	6399	0	2104	...	...	...	335
2	6747	2104	0	...	...	...	2431
...	...	...	...	0	...	...	...
...	...	...	...	...	0	...	...
...	...	...	...	...	...	0	...
2997	6485	335	2431	...	...	...	0

Senaryolar farklı olsa da tüm sokaklar birebir aynı olduğu için karşılıklı mesafe tablosu, Tablo 8 bu senaryoda da kullanılmıştır. Hücre değerleri iki lokasyon arasındaki jeodezik uzaklığı metre cinsinde ifade etmektedir. Tablo 11’in Xkord ve Ykord sütunlarını kullanarak Python Geopy kütüphanesi yardımıyla hesaplanmış 2998x2998 boyutlu matristir.

Tablo 13 Talep Matrisi (Kısaltılmış)- Senaryo 2

id	0	1	2	3	...	...	...	2997
talep	0	300	213	836	...	...	...	10

Talep matrisi Tablo 11'in KS*TK/SS sütunudur. 1x2997 boyutundadır ve hücre değerleri lokasyona ait çöp miktarını kg cinsinden ifade etmektedir.

Ana depo 0. lokasyon olarak belirlenmiştir. Tüm araçlar depodan tura başlayacak ve turu depoda bitirecektir.

Araç sayısı 32'dir. Tüm araçlar eşit kapasiteye sahiptir. Araç kapasiteleri algoritmanın toplam kat edilen mesafeyi azaltmak için araç sayısında tasarrufa gitmemesi ve problemin tam olarak 32 araçla çözülmesi için (Toplam Talep)/(Araç Sayısı = 32) formülüyle 10500 kg olarak hesaplanmıştır ve OR-Tools'a girilmiştir.

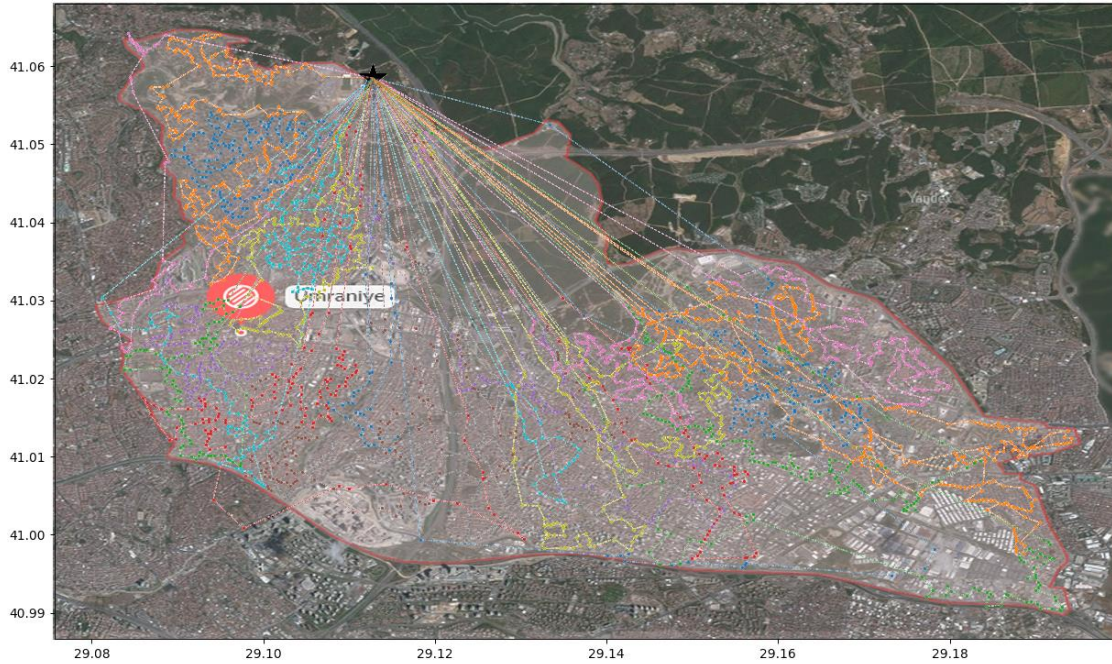
Tüm girdilerin belirlenmesiyle ilk sonuç üretme, yerel arama stratejisi ve limitler de belirlenerek OR-Tools çalıştırılmıştır. Sonuçlar Tablo 14'te özetlenmiştir.

Tablo 14 OR-Tools Çalıştırma Sonuçları - Senaryo 2

İlk Sonuç Üretme Stratejisi	Yerel Arama Stratejisi	Amaç Fonksiyonu (m)	Çalışma Süresi (sn)	Çözüm Kalitesi
En Ucuz Yay	Rehberli Yerel Arama	533.302	1210	100,00%
En Ucuz Yay	Benzetimli Tavlama	533.716	1151	99,92%
En Ucuz Yay	Tabu Arama	537.074	1218	99,30%
En Ucuz Yay	Açgözlü İniş	533.302	1178	100,00%
Tasarruf	Rehberli Yerel Arama	573.573	1180	92,98%
Tasarruf	Benzetimli Tavlama	573.568	1238	92,98%
Tasarruf	Tabu Arama	573.527	1168	92,99%
Tasarruf	Açgözlü İniş	573.573	914	92,98%
Christofides	Rehberli Yerel Arama	553.680	1193	96,32%
Christofides	Benzetimli Tavlama	555.590	1206	95,99%
Christofides	Tabu Arama	555.590	1246	95,99%
Christofides	Açgözlü İniş	553.680	1136	96,32%
En Kısıtlı Yay	Rehberli Yerel Arama	535.098	1155	99,66%
En Kısıtlı Yay	Benzetimli Tavlama	535.945	1202	99,51%
En Kısıtlı Yay	Tabu Arama	534.845	1173	99,71%
En Kısıtlı Yay	Açgözlü İniş	534.698	1150	99,74%

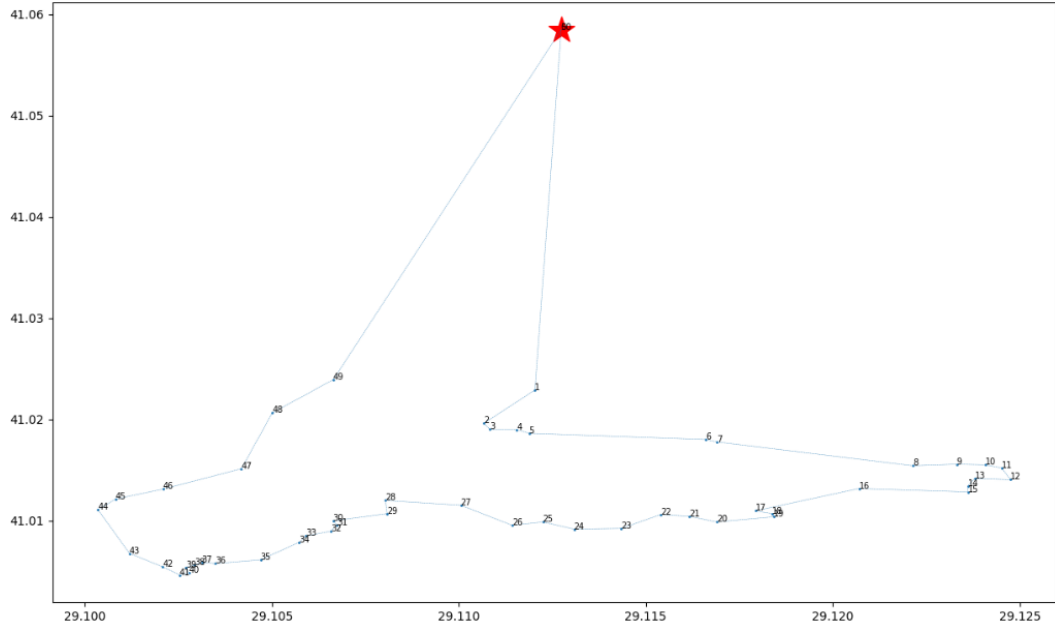
Tablo 14 incelendiğinde tüm algoritmalar içerisinde en iyi sonucu veren ilk çözüm stratejisinin En Ucuz Yay algoritması olduğu anlaşılmıştır. Bu yöntem ile üretilen ilk sonuçlar Rehberli Yerel Arama ve Açgözlü İniş algoritmasıyla geliştirildiğinde minimum amaç fonksiyonu değeri elde edilmiştir. Çözüm süresi de göz önünde bulundurulduğunda bu senaryo için En Ucuz Yay ve Açgözlü İniş algoritmalarının birlikte kullanılmasının en efektif sonucu verdiği söylenebilir. Ayrıca Tasarruf algoritmasıyla ilk çözüm üretildikten sonra Açgözlü İniş algoritmasıyla sonucun iyileştirilmesinin tüm deneyler arasında en hızlı çözüm olduğu ancak çözüm kalitesinin geride kaldığı anlaşılmıştır.

Tüm deneylerde 32 aracın tamamı kullanılarak %100 araç doluluk oranı sağlanırken, problem için en kısa rota uzunluğunun 533302 metre olduğu belirlenmiştir. En iyi çözüme ilişkin tüm araçların güzergah haritası Şekil 8’de gösterilmiştir. Her güzergah farklı renkle, Hekimbaşı Katı Atık Toplama Merkezi siyah yıldızla gösterilmiştir. Tüm turlar buradan başlamakta ve burada sona ermektedir.



Şekil 8 En İyi Deneye Ait Tüm Araçların Güzergah Haritası - Senaryo 2

OR-Tools’tan elde edilen en iyi deneyde, tek bir araca ait örnek güzergah haritası ve çıktısı aşağıdaki gibidir. Şekil 9’da en iyi deney sonucuna ait 17. Aracın güzergah haritası rastgele seçilerek gösterilmiştir. Kırmızı yıldız ana depoyu, noktalar ziyaret edilen lokasyonları, sayılar ziyaret sırasını göstermektedir.



Şekil 9 En İyi Deneye Ait 17. Nolu Aracın Güzergah Haritası – Senaryo 2

“Route for vehicle 17:

0 Load(0) -> 610 Load(12) -> 2066 Load(73) -> 293 Load(622) -> 83 Load(634) -> 2103 Load(646) -> ... (Rota kısaltılmıştır) -> 1605 Load(10468) -> 2164 Load(10480) -> 1764 Load(10488) -> 420 Load(10500) -> 0 Load(10500)

Distance of the route: 15144 m

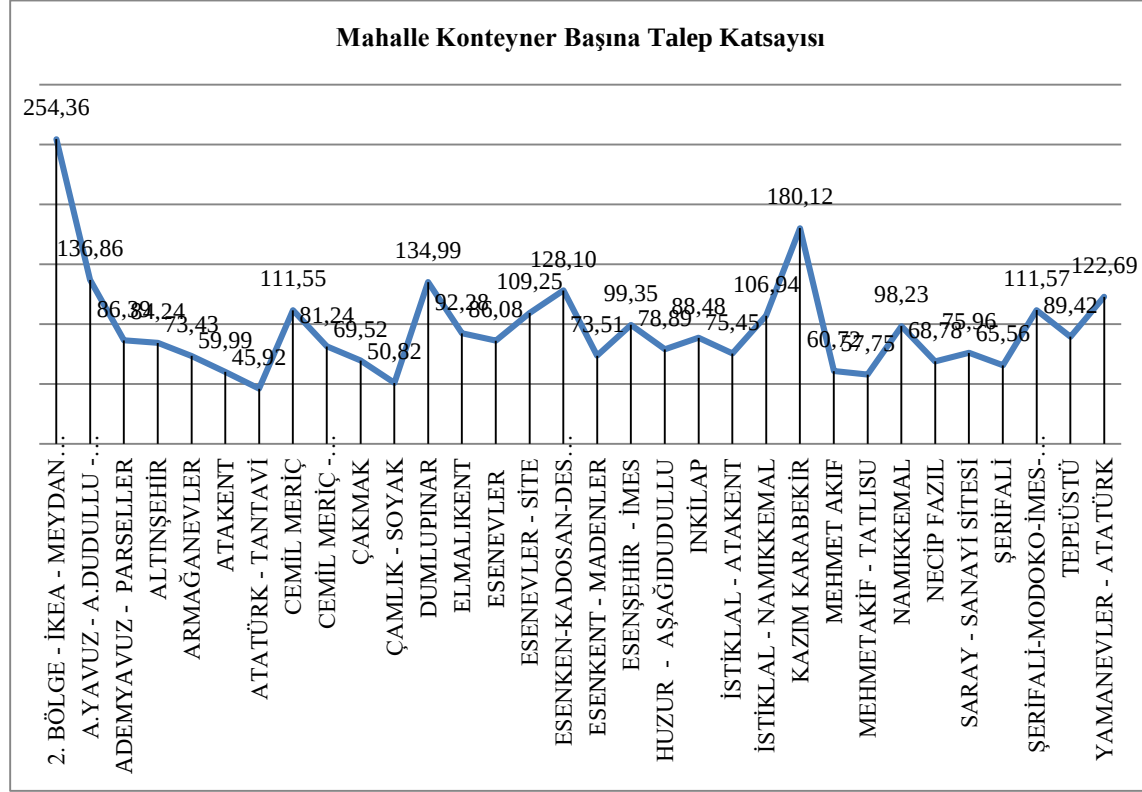
Load of the route: 10500 kg”

17. Araca ait bu örnek çıktı, rotanın 0. konumdan başladığını, 610. konuma giderek 12 kg’lık çöp alındığını, daha sonra 2066. konuma giderek 61 kg’lık daha çöp alındığı ve araç kapasitesinin 73 kg olduğunu anlatmaktadır. Araç 420. konuma geldiğinde araç kapasitesi 10500 kg olmuştur ve araç kapasitesi dolduğu için başlangıç konumuna geri dönerek turu tamamlamıştır. Rotanın toplam mesafesi 15144 m, rotada toplanan toplam çöp miktarı 10500 kg’dır.

4.4 K-ORTALAMAR VE OR-TOOLS SIRALI ÇÖZÜM

K-Ortalamar kümeleme algoritmasını ve OR-Tools girdilerini özetleyecek nihai veri tablosunu oluşturmak için öncelikle her bir bölge için güzergah başına düşen çöp miktarının hesaplanması gerekmektedir. Bu katsayılar yerel maks senaryolarındaki

gibi yerel maksimum değerler yerine Tablo 5’deki her güzergahın kendi üst sınırının ($u+2\text{std}$), konteyner sayısına bölünmesiyle oluşturulmuştur. Şekil 10’da gösterilmiştir.



Şekil 10 Mahalle Konteyner Başına Talep Katsayısı – Sıralı Çözüm

Sokak listesi tablosuna Şekil 10’da hesaplanan talep katsayıları ve Şekil 4’te gösterilen, 1-4 aralığında değişen sefer sayıları tabloya eklenmiştir. Hesaplanan sokak talepleri 25’lik sistemde ölçeklendirilmiştir ve yeni bir sütun olarak tabloya eklenmiştir.

Tablo 15 Nihai Veri Tablosu (Kısaltılmış Örnek Tablo)– Sıralı Çözüm

	Konum	Xkord	Ykord	D.Konteyner Sayısı	R.Talep Katsayısı	Sefer Sayısı	KS* TK/SS	Ölçek Değeri
0	Hekimbaşı Katı Atık Aktarma Merkezi	41,05846964	29,11273772	0	0	0	0	0
1	3016. Sok., Esenevler Mah., Ümraniye, İstanbul, Türkiye	41,008843	29,097232	0,2	109,25	2	11	1
2	Neval Sok., Necip Fazıl Mah., Ümraniye, İstanbul, Türkiye	41,015022	29,164866	12	254,36	4	763	31
3	Acun Sok., Ümraniye, İstanbul, Türkiye	41,008673	29,130559	10	60,718	1	607	25
4	Asil Cad., Topağacı Mah., Ümraniye, İstanbul, Türkiye	41,049326	29,105775	3	180,12	3	180	8
..	...	...	...	...	...	...	...	...

Tablo 15, K-Ortalamlar kümeleme algoritmasının çalışması için gerekli tüm verileri içeren özet tablonun küçük bir örneğidir. Konum, Xkord ve Ykord sütunları Yandex Haritalar'dan otomatik çekilmiştir. D.Konteyner Sayısı sütunu güzergahtaki düzenlenmiş konteyner sayılarını içermektedir. R.Talep Katsayısı Şekil 10'da hesaplanarak çekilmiştir. Sefer sayıları Şekil 4'te hesaplanmış ve çekilmiştir. Nihai talep $(\text{Konteyner Sayısı}) * (\text{R.Talep Katsayısı}) / (\text{Sefer Sayısı})$ formülüyle hesaplanmıştır ve küsuratlı değerler en yakın tam sayıya yuvarlanmıştır. Ölçek Değeri başlıklı sütunda nihai taleplerin 25'lik ölçeklendirilmiş karşılıkları bulunmaktadır. Her bir sokak ölçek değeri kadar sayıda tekrar edilerek k-ortalamlar kümeleme algoritmasına girdi olarak girilerek kümeleme yaparken sokak ağırlıkları da sürece dahil edilmiştir.

Kümeleme için kullanılan K-Means-Constrained algoritması girdi olarak koordinat listesi, küme sayısı, minimum ve maksimum küme boyutu, n_init ve max_iter bilgilerini istemektedir.

Koordinat listesi Tablo 15'teki Xkord ve Ykord sütunlarının ölçek değeri kadar tekrar tekrar yazılmasıyla oluşturulan listedir.

Tablo 16 K-Ortalamlar Küme Alt ve Üst Limitleri

Toplam Küme Eleman Sayısı	Araç Sayısı	Ortalama Küme Büyüklüğü	Minimum Küme Boyutu	Maksimum Küme Boyutu
14.521	32	453,78	363,03	544,54

Toplam küme eleman sayısı araç sayısına bölünerek tek bir küme için ortalama eleman sayısı belirlenmiştir. Sonrasında bu ortalama değer %20 alt ve üst limitleri hesaplanarak kümelerin minimum ve maksimum eleman sayısı belirlenmiştir.

N_init ve max_iter değerleri için farklı farklı değerler girilerek algoritma çalıştırılmıştır ve en iyi sonucu veren değer aranmıştır. Çalıştırma sonuçları Tablo 17'de özetlenmiştir.

Tablo 17 K-Ortalamalar Çalıştırma Sonuçları

Deney No	n_init	max_iter	Çalışma Süresi	Tekrarlayan Değer Sayısı
1	25	5000	11 hr 18 min 43 sec	36
2	50	1500	6 hr 46 min 37 sec	27
3	100	2000	20 hr 31 min 13 sec	25
4	100	5000	42 hr 32 min 9 sec	36
5	10	3000	2 hr 34 min 24 sec	35
6	10	1500	1 hr 31 min 27 sec	35

Kümeleme sonuçlarını kıyaslarken tüm deneylerde bazı lokasyonların ağırlık dilimlerinin iki farklı kümeye atandığı anlaşılmıştır. Bu yüzden çalışma süresi yerine, iki farklı kümeye atanarak tekrara sebep olan değerlerin en az olduğu deneyin en iyi sonucu verdiği düşünülerek deney 3 en iyi sonuç kabul edilmiştir. Deney 3'e ait küme eleman sayıları Tablo 18'de gösterilmiştir.

Tablo 18 Deney 3'e ait Küme Eleman Sayıları

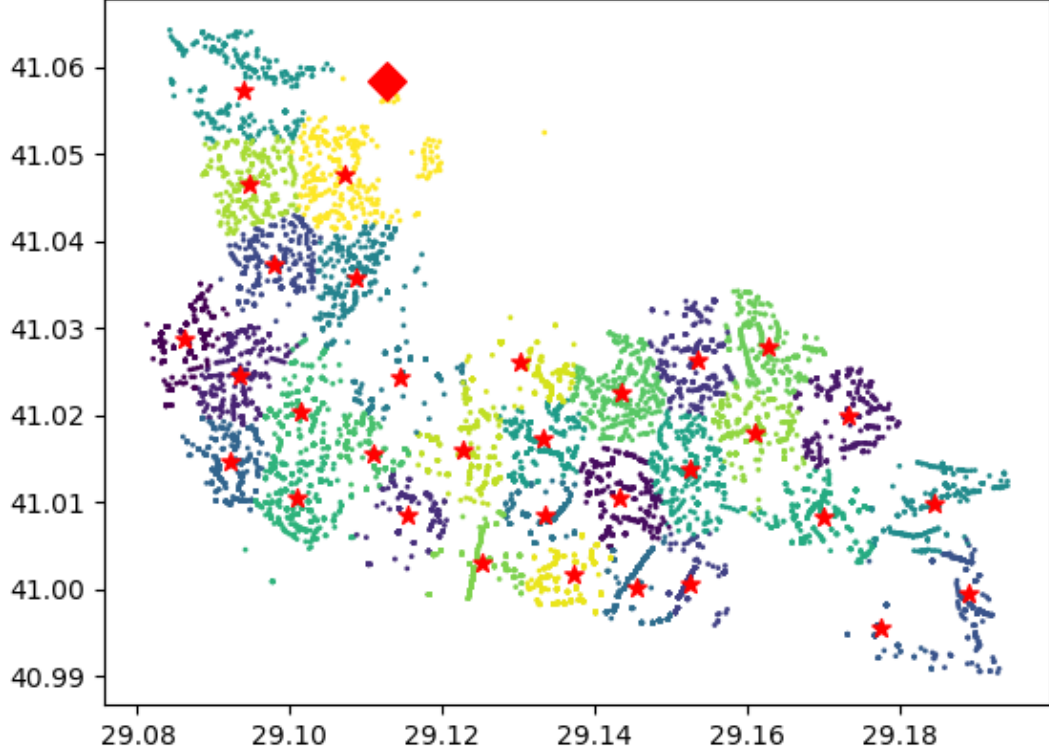
Küme No	0	1	2	3	4	5	6	7	8	9	10
Eleman Sayısı	505	532	363	467	545	420	545	462	363	363	545
Küme No	11	12	13	14	15	16	17	18	19	20	21
Eleman Sayısı	545	545	397	503	363	363	465	463	467	545	545
Küme No	22	23	24	25	26	27	28	29	30	31	
Eleman Sayısı	545	422	364	433	363	390	446	364	520	363	

Aynı lokasyonun iki farklı kümede olması problemin çözümünde hatalara sebep olacaktır. Hatalı lokasyonların tüm problem içerisinde çok küçük bir orana sahip olması (%0,17) sebebiyle elle iyileştirme yapılması uygun görülmüştür. Yapılan elle iyileştirme sonrasında küme eleman sayıları Tablo 19'daki gibi olmuştur.

Tablo 19 Elle İyileştirme Sonrasında Deney 3 Küme Eleman Sayıları

Küme No	0	1	2	3	4	5	6	7	8	9	10
Eleman Sayısı	506	536	363	457	547	420	536	460	363	363	554
Küme No	11	12	13	14	15	16	17	18	19	20	21
Eleman Sayısı	555	546	399	504	367	364	466	466	462	543	546
Küme No	22	23	24	25	26	27	28	29	30	31	
Eleman Sayısı	542	422	366	432	362	391	447	363	511	362	

İyileştirme sonrasında ilçe sokaklarının küme haritası Şekil 11'deki gibidir. Her küme farklı renkle ifade edilmiştir. Kırmızı yıldızlar küme merkezlerini, kırmızı karo ise katı atık toplama merkezini göstermektedir.



Şekil 11 İlçe Sokaklarının Küme Haritası

Bu aşamadan sonra her bir kümenin OR-Tools ile ayrı ayrı rotalanmasına geçilmiştir. Her kümenin elemanlarının lokasyon listesi ve bu lokasyonların karşılıklı mesafe matrisi oluşturularak OR-Tools'a girilmiştir. Tek araçla tüm talebin karşılanması kesin olduğu için OR-Tools tek araçla ve homojen taleple çalıştırılmıştır. Arama limiti problem daha küçük parçalara ayrıldığı için 100 olarak belirlenmiştir.

Tablo 20 K-Ortalamlar Sonrası OR-Tools Çözüm Sonucu

Küme No	0	1	2	3	4	5	6	7	8
Mesafe	13.816	18.164	22.496	15.627	14.492	18.299	17.535	14.756	25.400
Küme	9	10	11	12	13	14	15	16	17
Mesafe	21.265	16.028	16.317	13.680	11.315	14.499	25.790	14.143	14.996
Küme	18	19	20	21	22	23	24	25	26
Mesafe	22.049	22.303	19.064	13.823	12.186	20.196	20.635	15.470	20.661
Küme	27	28	29	30	31	<u>Toplam</u>			
Mesafe	13.597	14.526	12.621	16.816	20.751	<u>553.316</u>			

Sonuçlar Tablo 20’de birleştirilmiştir. Her küme bir aracı, her mesafe araçların kat ettikleri mesafeyi, toplam başlığı tüm araçların toplam mesafesini göstermektedir. Araç rotaları önceki senaryolarda olduğu gibi kayıt altına alınmıştır. Hiyerarşik çözüm yöntemiyle problem için en kısa rota uzunluğunun 553316 metre olduğu belirlenmiştir.

BÖLÜM 5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında katı atık optimizasyonu ve araç rotalama probleminin birlikte incelenerek İstanbul Ümraniye ilçesinin katı atık toplama rota optimizasyonun sağlanması amaçlanmıştır. Problem Kapasiteli Araç Rotalama Problemi olarak ele alınmıştır ve çözüm yöntemleri incelenmiştir. Kesin yöntemler, sezgisel yöntemler ve kümeleme algoritmaları olmak üzere üçe ayrılan çözüm yöntemleri genel hatlarıyla özetlenerek problemin çözümü için sezgisel, meta-sezgisel yöntemler ve önce kümele, sonra rotala prensibine dayanan sıralı bir çözüm yöntemi kullanılmıştır.

Problem mahallelerin çöp üretme katsayılarına göre sefer sayılarının düzenlendiği iki farklı senaryo ile desteklenmiştir. İlk senaryo çöp üretme derecesi bakımından diğer mahallelerden fazlasıyla ayrılan iki mahalleye çift sefer, kalan tüm mahallelere tek sefer gidilmesi senaryosudur. İkinci senaryo ise çöp kamyonlarının ve işçilerin daha eşit iş dağılımını sağlayacak, konteynerdeki çöp birikmesini önleyecek memnuniyet senaryosudur.

Uygulama aşamasında algoritmaların çalıştırılması için çeşitli Python kütüphanelerinden faydalanılmıştır. Bu paketlerden biri olan, literatürde sezgisel ve meta-sezgisel yöntemlerin kıyaslanmasında sıklıkla kullanılan OR-Tools paketi ile incelenen algoritmalar çalıştırılmıştır. K-Means-Constrained paketiyle, K-Ortalamlar Kümeleme algoritması çalıştırılmıştır ve önce kümelenen sokaklar daha sonra OR-Tools ile tekrar rotalanmıştır.

OR-Tools çalıştırılmasıyla elde edilen Senaryo 1 ve Senaryo 2 sonuçları Tablo 21’de özetlenmiştir. İlk senaryo için yapılan deneyler sonucunda en iyi sonucu veren algoritma ikilisinin En Ucuz Yay ve Tabu Arama algoritmaları olduğu anlaşılmıştır. Senaryoya ait en kısa rotaların uzunluğu 525443 metredir. %100 araç kullanım oranı sağlarken, araç başına düşen ortalama çöp miktarı 16420 kg’dır.

Tablo 21 OR-Tools Çalıştırma Sonuçları - Özet

Senaryo 1				Senaryo 2			
İlk Sonuç Üretme Seçeneği	Yerel Arama Stratejisi	Amaç Fonksiyonu (m)	Çalışma Süresi (s)	İlk Sonuç Üretme Seçeneği	Yerel Arama Stratejisi	Amaç Fonksiyonu (m)	Çalışma Süresi (s)
EUY	RYA	525,582	640	EUY	RYA	533,302	1210
EUY	BT	525,443	634	EUY	BT	533,716	1151
EUY	TA	525,443	<u>621</u>	EUY	TA	537,074	1218
EUY	Aİ	525,471	630	EUY	Aİ	533,302	<u>1178</u>
TAS	RYA	571,122	677	TAS	RYA	573,573	1180
TAS	BT	571,352	646	TAS	BT	573,568	1238
TAS	TA	571,389	676	TAS	TA	573,527	1168
TAS	Aİ	571,122	604	TAS	Aİ	573,573	914
CHR	RYA	558,127	645	CHR	RYA	553,68	1193
CHR	BT	558,522	662	CHR	BT	555,59	1206
CHR	TA	558,522	657	CHR	TA	555,59	1246
CHR	Aİ	558,127	675	CHR	Aİ	553,68	1136
EKY	RYA	535,856	665	EKY	RYA	535,098	1155
EKY	BT	534,379	629	EKY	BT	535,945	1202
EKY	TA	534,471	627	EKY	TA	534,845	1173
EKY	Aİ	534,757	640	EKY	Aİ	534,698	1150

EUY: En Uygun Yay, TAS: Tasarruf, CHR: Christofides, EKY: En Kısıtlı Yay, RYA: Rehberli Yerel Arama, BT: Benzetimli Tavlama, TA: Tabu Arama Aİ: Açgözlü İniş

İkinci senaryo için yapılan deneyler sonucunda En Ucuz Yay ve Açgözlü İniş algoritmalarının birlikte kullanılmasının en iyi sonucu verdiği anlaşılmıştır. Senaryoya ait en kısa rotaların uzunluğu 533302 metredir. %100 araç kullanım oranı sağlanırken, araç başına düşen ortalama çöp miktarı 16665 kg'dır.

Her iki senaryo için de tüm deney sonuçları incelendiğinde sezgisel ve meta-sezgisel algoritmaların devasa boyutlu KARP'lerinde dahi hızlı ve olurlu sonuçlar verebildiği anlaşılmıştır.

Uygulama bölümünün son kısmında, Senaryo 2 için sıralı çözüm uygulanmıştır. Cömert ve arkadaşları (2018) çalışmasında KARP'nin boyutunu küçültmek için kümeleme algoritmalarına kapasite kısıdı eklemiştir. Bu çalışmadaysa sokak talepleri hesaplanan sefer sayılarına bölünerek sonrasında 25'lik dilimler halinde ölçeklendirilmiştir. Bu ölçeklendirme sayesinde K-Ortalamlar Kümeleme

algoritmasının problem boyutu yaklaşık %95 küçültülmüştür ve kümeleme algoritmasına kapasiteler de dahil edilmiştir. 25'lik dilimlerin algoritmaya tekrar tekrar saydırılmasıyla tek bir elemanın küme içerisindeki ağırlığına da anlam kazandırılmıştır. Kapasite kısıdına göre yapılan kümeleme sonrasında OR-Tools ile her bir küme kendi içerisinde rotalanmıştır ve en kısa rota uzunluğunun 553316 metre olduğu anlaşılmıştır.

İkinci senaryo için uygulanan iki farklı çözüm yaklaşımı sonucunda, kümeleme yapmadan En Ucuz Yay veya En Kısıtlı Yay algoritmalarıyla ilk çözümün bulunduktan sonra diğer tur geliştirici algoritmalarla sonucun iyileştirilmesinin daha iyi sonuçlar verdiği anlaşılmıştır. Sıralı çözümü içeren yaklaşımsa, daha uzun çalışma süresi gerektirse de Christofides ve Tasarruf algoritmasıyla üretilen ve daha sonra diğer algoritmalarla iyileştirilen deneylere göre daha iyi sonuçlar vermiştir.

Belediyeden alınan güzergah sokak listesi ve sokak uzunluklarına göre güzergahlarda bulunan sokakların toplam uzunluğu 803897 metre, araç başına kat edilen ortalama mesafe 25065 metredir. Vu ve arkadaşları (2020) çalışmalarında CBS kullanarak 48 farklı senaryoyu incelemişler ve kamyon kapasitesini artırarak tüm senaryolar için kat edilen mesafenin %21'e kadar azaltılabileceğini göstermişlerdir. Bu çalışmada da Senaryo 1 ve Senaryo 2'nin uygulanmasıyla araçların kat ettiği toplam jeodezik mesafenin sırasıyla %35 ve %34 , önce kümele sonra rotala yönteminin uygulamasıyla %31 iyileştirilebileceği anlaşılmıştır.

Bu tez çalışmasıyla literatürde sıklıkla çalışılan ARP'nin, katı atık toplama rota optimizasyonu problemi ile birlikte ele alınabileceği, incelenen sezgisel ve meta-sezgisel çözüm yöntemlerinin devasa ölçekli problemlerde de hızlı ve olurlu çözümler vererek çalışabildiği, önce kümele sonra rotala prensibine dayanan yöntemin etkili bir araç olabileceği anlaşılmıştır.

Bu tez çalışmasında sokakların birbirlerine olan uzaklıkları jeodezik mesafe olarak hesaplanmıştır. Dolayısıyla sokaklar arasındaki gerçek mesafede ve jeodezik mesafe arasında oluşabilecek farklılıklar bu çalışmanın zayıf noktası olarak değerlendirilebilir. Bu zayıflığı ortadan kaldırmak için ücretli servislerin veri tabanına erişerek mesafe matrisleri yeniden düzenlenip algoritmalar yeniden çalıştırılabilir.

Çalışmaya akıllı teknolojilerin eklenmesiyle çöp konteynerlerinin doluluk oranları tam zamanlı olarak tahmin edilebilir, dolayısıyla geçmiş veriler doğrultusunda sefer sayıları oluşturmak yerine anlık rota optimizasyonunun sağlanması mümkün olabilir. Çöp kamyonları olağan rutinlerini takip etmek yerine sıra dışı çöp üretim senaryolarına cevap verebilir. Yakıt, zaman ve iş gücünden tasarruf mümkün olabilir.

Çalışmanın yöntemi Ümraniye ilçesi özelinde uygulanmış olsa da benzer ihtiyaca sahip diğer il ve ilçe belediyeleri de bu çalışmadan faydalanabilirler. Ölçek değeri ihtiyaç büyüklüğüne göre yeniden düzenlenerek kümeleme yapılır, küme sonuçlarına göre bu çalışmadaki gibi elle iyileştirme ihtiyacı doğabilir. Bu durumda belirtilen kurala göre elle iyileştirme yapılarak probleme çözüm üretilir.

Çalışma yukarıda bahsedilen zayıflık -jeodezik mesafe ve gerçek uzaklık farkı- ücretli servisler kullanılarak ortadan kaldırıldığında belediyelerin çöp rotalama optimizasyonu ihtiyacına yanıt verecek bir çözüm haline gelecektir. Yine yukarıda bahsedildiği üzere çalışmaya eklenecek sensör ve diğer akıllı teknolojilerle çöp doluluk oranları anlık olarak takip edilerek sadece kritik düzeyi geçen konteynerler belirlenerek problem boyutu daha da küçültülebilir ve daha hızlı ve ekonomik çözümler elde edilebilir. Tüm süreç takip edilebilir bir hale gelir ve operasyon maliyetleri ve kaynak harcamaları azalır.

KAYNAKÇA

- Abadi, S., Mat The, K. S., Nasir, B. M., Huda, M., Ivanova, N. L., Sari, T. I., Maselena, A., Satria, F., & Muslihudin, M. (2018). Application Model Of K-Means Clustering: Insights Into Promotion Strategy Of Vocational High School. *International Journal Of Engineering And Technology(UAE)*, 7(2.27 Special Issue 27), 182–187. <https://doi.org/10.14419/Ijjet.V7i2.11491>
- Akhtar, M., Hannan, M. A., Begum, R. A., Basri, H., & Scavino, E. (2017). Backtracking Search Algorithm In CVRP Models For Efficient Solid Waste Collection And Route Optimization. *Waste Management*, 61, 117–128. <https://doi.org/10.1016/J.Wasman.2017.01.022>
- Altabeeb, A. M., Mohsen, A. M., & Ghallab, A. (2019). An Improved Hybrid Firefly Algorithm For Capacitated Vehicle Routing Problem. *Applied Soft Computing Journal*, 84, 105728. <https://doi.org/10.1016/J.Asoc.2019.105728>
- Amous, M., Toumi, S., Jarboui, B., & Eddaly, M. (2017). A Variable Neighborhood Search Algorithm For The Capacitated Vehicle Routing Problem. *Electronic Notes In Discrete Mathematics*, 58, 231–238. <https://doi.org/10.1016/J.Endm.2017.03.030>
- Appelgren, L. H. (1969). A Column Generation Algorithm For A Ship Scheduling Problem. 3(1), 53–68. <https://doi.org/10.1287/TRSC.3.1.53>
- Archetti, C., & Speranza, M. G. (2008). The Split Delivery Vehicle Routing Problem: A Survey. *Operations Research/ Computer Science Interfaces Series*, 43, 103–122. https://doi.org/10.1007/978-0-387-77778-8_5
- Asefi, H., Lim, S., Maghrebi, M., & Shahparvari, S. (2019). Mathematical Modelling And Heuristic Approaches To The Location-Routing Problem Of A Cost-Effective Integrated Solid Waste Management. *Annals Of Operations Research*, 273(1–2), 75–110. <https://doi.org/10.1007/S10479-018-2912-1>
- Assaf, R., & Saleh, Y. (2017). Vehicle-Routing Optimization For Municipal Solid Waste Collection Using Genetic Algorithm: The Case Of Southern Nablus City. *Civil And Environmental Engineering Reports*, 26(3), 43–57. <https://doi.org/10.1515/Ceer-2017-0034>

- Atık Yönetimi Yönetmeliği, (2015).
- Baker, B. M., & Ayechew, M. A. (2003). A Genetic Algorithm For The Open Vehicle Routing Problem. *Computers & Operations Research*, 30(5), 787–800. [https://doi.org/https://doi.org/10.1016/S0305-0548\(02\)00051-5](https://doi.org/https://doi.org/10.1016/S0305-0548(02)00051-5)
- Beltrami, E. J., & Bodin, L. D. (1974). Networks And Vehicle Routing For Municipal Waste Collection. *Networks*, 4, 65–94.
- Bertsimas, D. J. (1992). A Vehicle Routing Problem With Stochastic Demand. *Operations Research*, 40(3), 574–585. <https://doi.org/10.1002/Cpa.1955.3160080113>
- Bodin, L. D. (1975). A Taxonomic Structure For Vehicle Routing And Scheduling Problems. *Computers & Urban Society*, 1(1), 11–29. [https://doi.org/10.1016/0305-7097\(75\)90003-4](https://doi.org/10.1016/0305-7097(75)90003-4)
- Borčinová, Z. (2017). Two Models Of The Capacitated Vehicle Routing Problem. *Croatian Operational Research Review*, 8(2), 463–469. <https://doi.org/10.17535/Crorr.2017.0029>
- Çevre Kanunu, (1983).
- Chévez, P., Barbero, D., Martini, I., & Discoli, C. (2017). Application Of The K-Means Clustering Method For The Detection And Analysis Of Areas Of Homogeneous Residential Electricity Consumption At The Great La Plata Region, Buenos Aires, Argentina. *Sustainable Cities And Society*, 32, 115–129. <https://doi.org/10.1016/J.Scs.2017.03.019>
- Clarke, G., & Wright, J. W. (1964). Scheduling Of Vehicles From A Central Depot To A Number Of Delivery Points. *Operations Research*, 12(4), 568–581. <https://doi.org/10.1287/Opre.12.4.568>
- Comert, S. E., Yazgan, H. R., Kır, S., & Yener, F. (2018). A Cluster First-Route Second Approach For A Capacitated Vehicle Routing Problem: A Case Study. *International Journal Of Procurement Management*, 11(4), 399–419. <https://doi.org/10.1504/IJPM.2018.092766>
- Croes, G. A. (1958). A Method For Solving Traveling-Salesman Problems. 6(6),

791–812. <https://doi.org/10.1287/OPRE.6.6.791>

- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. In *Management Science* (Pp. 80–91.).
- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A New Optimization Algorithm For The Vehicle Routing Problem With Time Windows. *40*(2), 342–354. <https://doi.org/10.1287/OPRE.40.2.342>
- Di Giuseppe, M. G., Troiano, A., Troise, C., & De Natale, G. (2014). K-Means Clustering As Tool For Multivariate Geophysical Data Analysis. An Application To Shallow Fault Zone Imaging. *Journal Of Applied Geophysics*, *101*, 108–115. <https://doi.org/10.1016/J.Jappgeo.2013.12.004>
- Dorigo, M., & Di Caro, G. (1999). Ant Colony Optimization: A New Meta-Heuristic. *Proceedings Of The 1999 Congress On Evolutionary Computation, CEC 1999*, *2*, 1470–1477. <https://doi.org/10.1109/CEC.1999.782657>
- Erfani, S. M. H., Danesh, S., Karrabi, S. M., & Shad, R. (2017). A Novel Approach To Find And Optimize Bin Locations And Collection Routes Using A Geographic Information System. *Waste Management And Research*, *35*(7), 776–785. <https://doi.org/10.1177/0734242X17706753>
- Fisher, M. L. (1994). Optimal Solution Of Vehicle Routing Problems Using Minimum K-Trees. *42*(4), 626–642. <https://doi.org/10.1287/OPRE.42.4.626>
- Flood, M. M. (1956). The Traveling-Salesman Problem. *4*(1), 61–75. <https://doi.org/10.1287/OPRE.4.1.61>
- Gaudio, M., & Paletta, G. (1992). A Heuristic For The Periodic Vehicle Routing Problem. *26*(2), 86–92. <https://doi.org/10.1287/TRSC.26.2.86>
- Glover, F. (1989). Tabu Search—Part I. *1*(3), 190–206. <https://doi.org/10.1287/IJOC.1.3.190>
- Goldberg, D. E., & Holland, J. H. (1988). Genetic Algorithms And Machine Learning. *Machine Learning*, *3*(2), 95–98.
- Goli, A., Aazami, A., & Jabbarzadeh, A. (2018). Accelerated Cuckoo Optimization Algorithm For Capacitated Vehicle Routing Problem In Competitive

- Conditions. *International Journal Of Artificial Intelligence*, 16(1).
- Hadjiconstantinou, E., Christofides, N., & Mingozzi, A. (1995). A New Exact Algorithm For The Vehicle Routing Problem Based On q -Paths And k -Shortest Paths Relaxations. *Annals Of Operations Research*, 61(1), 21–43. <https://doi.org/10.1007/BF02098280>
- Han, H., & Cueto, E. P. (2015). Waste Collection Vehicle Routing Problem: Literature Review. *PROMET - Traffic&Transportation*, 27(4), 345–358. <https://doi.org/10.7307/Ptt.V27i4.1616>
- Hannan, M. A., Akhtar, M., Begum, R. A., Basri, H., Hussain, A., & Scavino, E. (2018). Capacitated Vehicle-Routing Problem Model For Scheduled Solid Waste Collection And Route Optimization Using PSO Algorithm. *Waste Management*, 71, 31–41. <https://doi.org/10.1016/J.Wasman.2017.10.019>
- Hertz, A., & Mittaz, M. (2001). A Variable Neighborhood Descent Algorithm For The Undirected Capacitated Arc Routing Problem. 35(4), 425–434. <https://doi.org/10.1287/TRSC.35.4.425.10431>
- Hertz, A., Taillard, E., & De Werra, D. (1995). A Tutorial On Tabu Search. *Proc. Of Giornate Di Lavoro AIRO*, 95, 13–24.
- Hiquebran, D. T., Alfa, A. S., Shapiro, J. A., & Gittos, D. H. (2007). A Revised Simulated Annealing And Cluster-First Route-Second Algorithm Applied To The Vehicle Routing Problem. 22(2), 77–107. <https://doi.org/10.1080/03052159308941327>
- Hottung, A., & Tierney, K. (2020). Neural Large Neighborhood Search For The Capacitated Vehicle Routing Problem. *Frontiers In Artificial Intelligence And Applications*, 325, 443–450. <https://doi.org/10.3233/FAIA200124>
- Ibrahim, A. ., Lo, N., Abdulaziz, R. O., & Ishaya, J. A. (2019). Capacitated Vehicle Routing Problem. *International Journal Of Research -GRANTHAALAYAH*, 7(3), 310–327. <https://doi.org/10.29121/Granthaalayah.V7.İ3.2019.976>
- Ilhan, I. (2020). A Population Based Simulated Annealing Algorithm For Capacitated Vehicle Routing Problem. *Turkish Journal Of Electrical Engineering And Computer Sciences*, 28(3), 1217–1235.

<https://doi.org/10.3906/Elk-1902-122>

İstanbul Büyükşehir Belediyesi. (N.D.). *No Title*. <https://atikyonetimi.ibb.istanbul/>

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal Of Artificial Intelligence Research*, 4, 237–285. <https://doi.org/10.1613/JAIR.301>

Kaufman, L., & Rousseeuw, P. (1990). Finding Groups In Data: An Introduction To Cluster Analysis. In *Wiley Series In Probability And Statistics*. John Wiley & Sons.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings Of ICNN'95 - International Conference On Neural Networks*, 4, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>

Kilby, P., Prosser, P., & Shaw, P. (1999). Guided Local Search For The Vehicle Routing Problem With Time Windows. *Meta-Heuristics: Advances And Trends In Local Search Paradigms For Optimization*, 473–486. https://doi.org/10.1007/978-1-4615-5775-3_32

Kloimüller, C., Papazek, P., Hu, B., & Raidl, G. R. (2015). A Cluster-First Route-Second Approach For Balancing Bicycle Sharing Systems. *International Conference On Computer Aided Systems Theory - EUROCAST 2015*, 9520, 439–446. https://doi.org/10.1007/978-3-319-27340-2_55

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings Of The IEEE*, 78(9), 1464–1480. <https://doi.org/10.1109/5.58325>

Kuo, R. J., & Zulvia, F. E. (2017). Hybrid Genetic Ant Colony Optimization Algorithm For Capacitated Vehicle Routing Problem With Fuzzy Demand - A Case Study On Garbage Collection System. *2017 4th International Conference On Industrial Engineering And Applications, ICIEA 2017*, 244–248. <https://doi.org/10.1109/IEA.2017.7939215>

Laarhoven, P. J. M. Van, & Aarts, E. H. L. (1987). Simulated Annealing. *Simulated Annealing: Theory And Applications*, 7–15. https://doi.org/10.1007/978-94-015-7744-1_2

- Levy-Kramer, J. (N.D.). *Welcome To K-Means-Constrained's Documentation! — K-Means-Constrained 0.5.1 Documentation*. Retrieved November 21, 2021, From <https://Joshlk.Github.io/K-Means-Constrained/>
- Lin, S., & Kernighan, B. W. (1973). An Effective Heuristic Algorithm For The Traveling-Salesman Problem. <https://doi.org/10.1287/Opre.21.2.498>, 21(2), 498–516. <https://doi.org/10.1287/OPRE.21.2.498>
- Macqueen, J. (1967). Some Methods For Classification And Analysis Of Multivariate Observations. In *Proceedings Of The 5th Berkeley Symposium On Mathematical Statistics And Probability*, 1(14), 281–297.
- Miller, D. L. (1995). A Matching Based Exact Algorithm For Capacitated Vehicle Routing Problems. <https://doi.org/10.1287/Ijoc.7.1.1>, 7(1), 1–9. <https://doi.org/10.1287/IJOC.7.1.1>
- Mladenović, N., & Hansen, P. (1997). Variable Neighborhood Search. *Computers & Operations Research*, 24(11), 1097–1100. [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)
- OR-Tools | Google Developers. (N.D.). Retrieved November 21, 2021, From <https://developers.google.com/Optimization>
- Osman, I. H. (1993). Metastrategy Simulated Annealing And Tabu Search Algorithms For The Vehicle Routing Problem. *Annals Of Operations Research* 1993 41:4, 41(4), 421–451. <https://doi.org/10.1007/BF02023004>
- Oyelade, O. J., Oladipupo, O. O., & Obagbuwa, I. C. (2010). Application Of K Means Clustering Algorithm For Prediction Of Students Academic Performance. *International Journal Of Computer Science And Information Security*, 7(1), 292–295.
- Perron, L. (2011). Operations Research And Constraint Programming At Google. *Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics)*, 6876 LNCS(2), 2–2. https://doi.org/10.1007/978-3-642-23786-7_2
- Prim, R. C. (1957). Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal*, 36(6), 1389–1401. <https://doi.org/10.1002/J.1538->

- Rabbouch, B., Saâdaoui, F., & Mraïhi, R. (2020). Empirical-Type Simulated Annealing For Solving The Capacitated Vehicle Routing Problem. *Journal Of Experimental And Theoretical Artificial Intelligence*, 32(3), 437–452. <https://doi.org/10.1080/0952813X.2019.1652356>
- Ralphs, T. K., Kopman, L., Pulleyblank, W. R., & Trotter, L. E. (2003). Affection Rate Determination Of Effective Factors On Acceptance Of E-Banking. *Mathematical Programming*, 94, 343–359.
- Shamrat, F. M. J. M., Tasnim, Z., Mahmud, I., Jahan, N., & Nobel, N. I. (2020). Application Of K-Means Clustering Algorithm To Determine The Density Of Demand Of Different Kinds Of Jobs. *International Journal Of Scientific And Technology Research*, 9(2), 2550–2557.
- Shelly, Z., Burch, R. F. V., Tian, W., Strawderman, L., Piroli, A., & Bichey, C. (2020). Using K-Means Clustering To Create Training Groups For Elite American Football Student-Athletes Based On Game Demands. *International Journal Of Kinesiology And Sports Science*, 8(2), 47–63. <https://doi.org/10.7575//Aiac.ijkss.V.8n.2p.47>
- Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., & Banaszak, Z. (2021). Optimization Of Capacitated Vehicle Routing Problem With Alternative Delivery, Pick-Up And Time Windows: A Modified Hybrid Approach. *Neurocomputing*, 423, 670–678. <https://doi.org/10.1016/J.Neucom.2020.02.126>
- Tan, L., Li, C., Xia, J., & Cao, J. (2019). Application Of Self-Organizing Feature Map Neural Network Based On K-Means Clustering In Network Intrusion Detection. *Computers, Materials And Continua*, 61(1), 275–288. <https://doi.org/10.32604/Cmc.2019.03735>
- Toth, P., & Vigo, D. (2002a). Models, Relaxations And Exact Approaches For The Capacitated Vehicle Routing Problem. *Discrete Applied Mathematics*, 123(1–3), 487–512. [https://doi.org/10.1016/S0166-218X\(01\)00351-1](https://doi.org/10.1016/S0166-218X(01)00351-1)
- Toth, P., & Vigo, D. (2002b). Branch-And-Bound Algorithms For The Capacitated VRP. In *The Vehicle Routing Problem* (Pp. 29–51). Society For Industrial And

- Applied Mathematics. <https://doi.org/10.1137/1.9780898718515.CH2>
- Van Beek, P. (2006). Backtracking Search Algorithms. *Foundations Of Artificial Intelligence*, 2(C), 85–134. [https://doi.org/10.1016/S1574-6526\(06\)80008-8](https://doi.org/10.1016/S1574-6526(06)80008-8)
- Vu, H. L., Ng, K. T. W., Fallah, B., Richter, A., & Kabir, G. (2020). Interactions Of Residential Waste Composition And Collection Truck Compartment Design On GIS Route Optimization. *Waste Management*, 102, 613–623. <https://doi.org/10.1016/J.Wasman.2019.11.028>
- Wardani, R. S., Purwanto, Sayono, & Paramananda, A. (2019). Clustering Tuberculosis In Children Using K-Means Based On Geographic Information System. *AIP Conference Proceedings*, 2114(June). <https://doi.org/10.1063/1.5112483>
- Welcome To Geopy's Documentation! — Geopy 2.2.0 Documentation. (N.D.). Retrieved November 21, 2021, From <https://Geopy.Readthedocs.Io/En/Stable/>
- Xiong, C., Hua, Z., Lv, K., & Li, X. (2017). An Improved K-Means Text Clustering Algorithm By Optimizing Initial Cluster Centers. *Proceedings - 2016 7th International Conference On Cloud Computing And Big Data, CCBBD 2016*, 265–268. <https://doi.org/10.1109/CCBD.2016.059>
- Yang, X. S., & Deb, S. (2009). Cuckoo Search Via Lévy Flights. *2009 World Congress On Nature And Biologically Inspired Computing, NABIC 2009 - Proceedings*, 210–214. <https://doi.org/10.1109/NABIC.2009.5393690>
- Yazgan, H. R., Comert, S. E., & Kılıç, E. N. (2020). Araç Rotalama Probleminin Sezgisel Algoritmalar İle Çözülmesi: Bir Boya Fabrikasında Uygulama. *Journal Of Turkish Operations Management*, 4(2), 549–563.
- Zhan, S. H., Lin, J., Zhang, Z. J., & Zhong, Y. W. (2016). List-Based Simulated Annealing Algorithm For Traveling Salesman Problem. *Computational Intelligence And Neuroscience*, 2016. <https://doi.org/10.1155/2016/1712630>
- Zhang, G., Li, Y., & Deng, X. (2020). K-Means Clustering-Based Electrical Equipment Identification For Smart Building Application. *Information (Switzerland)*, 11(1), 1–18. <https://doi.org/10.3390/Info11010027>

EK-1 ALINAN GÜZERGAH VERİLERİ

EK-2 ALINAN GÜZERGAH TONAJ VERİLERİ

EK-3 LOKASYONLAR ARASI KARŞILIKLI MESAFE MATRİSİ