

T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**MAKİNE ÖĞRENMESİ İLE ARAÇ ROTALAMA PROBLEMİ İÇİN
ÖNCE-KÜMELE SONRA-ROTA LA YÖNTEMLERİNİN İNCELENMESİ**

Sedat GÜZELŞEMME

YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

DANIŞMAN
Doç. Dr. Faruk SERİN

TUNCELİ – 2024

T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE ARAÇ ROTALAMA PROBLEMİ İÇİN
ÖNCE-KÜMELE SONRA-ROTALA YÖNTEMLERİNİN İNCELENMESİ

Sedat GÜZELŞEMME
(190080031)

YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

DANIŞMAN
Doç. Dr. Faruk SERİN

TUNCELİ – 2024

T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE ARAÇ ROTALAMA PROBLEMİ İÇİN
ÖNCE-KÜMELE SONRA-ROTALA YÖNTEMLERİNİN İNCELENMESİ

Sedat GÜZELŞEMME
YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

Bu tez 15/01/2024 tarihinde aşağıdaki jüri üyeleri tarafından **oybirliği** ile kabul edilmiştir.

İmza:.....

Doç. Dr. Faruk
SERİN
(Mersin Üniversitesi)

DANIŞMAN

İmza:.....

Doç. Dr. Melih YÜCESAN
(Munzur Üniversitesi)

ÜYE

İmza:.....

Dr. Öğr. Üyesi, Kenan
İNCE
(İnönü Üniversitesi)

ÜYE

Bu tez, Enstitümüz Mühendislik Yönetimi Anabilim Dalı'nda hazırlanmıştır.

Doç. Dr. Murat KORUNUR
Enstitü Müdürü
İmza ve Mühür

NOT: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı “Fikir ve Sanat Eserleri Kanunu”ndaki hükümlere tabidir.

15/01/2024

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Sedat GÜZELŞEMME
Öğrenci

Doç. Dr. Faruk SERİN
Danışman

TEŞEKKÜR

Danışman hocam Sayın Doç. Dr. Faruk SERİN başta olmak üzere, tezimin her aşamasında desteklerini benden esirgemeyen aileme ve arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Sedat GÜZELŞEMME
TUNCELİ- 2024



İÇİNDEKİLER

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	I
TEŞEKKÜR.....	II
İÇİNDEKİLER.....	III
ŞEKİLLER LİSTESİ	IV
TABLolar LİSTESİ	V
ÖZET	VI
ABSTRACT	VII
1. GİRİŞ.....	1
1.1. Literatür Taraması	3
1.2. Tezin Katkısı ve Organizasyonu	5
2. MATERYAL VE METOT.....	6
2.1. Kullanılan yazılım araçları	6
2.2. Gezgin Satıcı Problemi.....	8
2.3. Araç Rotalama Problemi	9
2.4. Önce Kümele Sonra Rotala Yaklaşımı.....	10
2.5. Makine Öğrenmesi ve Kümeleme Algoritmaları	11
2.5.1. K-Ortalama.....	12
2.5.2. DBSCAN	13
2.5.3. BIRCH	14
2.5.4. OPTICS	15
2.6. Veri seti	16
3. BULGULAR VE TARTIŞMA.....	17
4. SONUÇ VE ÖNERİLER.....	32
5. KISITLAMALAR VE GELECEK ÇALIŞMALAR.....	33
6. KAYNAKLAR	34
ÖZGEÇMİŞ	

ŞEKİLLER LİSTESİ

Şekil 3.1. Veri seti-1’de bulunan rezervuar alanların harita üzerinde gösterimi.....	17
Şekil 3.2. DBSCAN algoritmasının veri seti-1’e uygulanması sonucu elde edilen kümelerin gösterimi	18
Şekil 3.3. DBSCAN sonucu herhangi bir kümeye atanamayan rezervuar alanların haritada gösterimi	19
Şekil 3.4. DBSCAN ile kümeleme işleminden geçirilmiş gürültü içermeyen veri seti-1 gösterimi	19
Şekil 3.5. DBSCAN ile elde edilen kümelerin merkezi noktalarının haritada gösterimi....	20
Şekil 3.6. DBSCAN ile kümeleme sonucu gürültü olarak sınıflandırılan rezervuar alanı ve küme merkezlerinin bir arada gösterimi	21
Şekil 3.7. DBSCAN ile kümeleme sonucu gürültü olarak belirlenen noktaların en yakın küme merkezlerine göre ataması yapılması ile elde edilen kümeleme sonucu...	21
Şekil 3.8. DBSCAN ile kümeleme sonucu gürültü olarak belirlenen noktaların en yakın komşunun kümesine ataması yapılması ile elde edilen kümeleme sonucu	22
Şekil 3.9. Herbir kümenin rotalama işlemi sonrası belirlenen rotalar	23
Şekil 3.10. Şekil 3.9’da siyah kare ile işaretlenmiş rota alanının yakın gösterimi.....	23
Şekil 3.11. Rotalama işlemi sonucunda elde edilen bazı güzergahların gerçek yol gösterimi	24
Şekil 3.12. K-Ortalama algoritması kullanılarak kümeleme işleminden geçirilmiş veri seti-1 gösterimi	26
Şekil 3.13. K-Ortalama ile elde edilen kümelerin rotalama işlemi sonrası haritada gösterimi	26
Şekil 3.14. Şekil 3.13’de gösterilen haritada siyah kare ile işaretlenmiş alanın yakın gösterimi	27
Şekil 3.15. Rotalama işlemi sonucunda elde edilen bazı güzergahların gerçek yol gösterimi	27
Şekil 3.16. Veri seti-2’ye ait DBSCAN ile oluşturulan küme dağılımı	29
Şekil 3.17. Veri seti-2’ye ait K-Ortalama ile oluşturulan küme dağılımı	30
Şekil 3.18. Veri seti-3’e ait DBSCAN ile oluşturulan küme dağılımı	31
Şekil 3.19. Veri seti-3’e ait K-Ortalama ile oluşturulan küme dağılımı	31

TABLÖLAR LİSTESİ

Tablo 2.1. Kullanılan algoritmalara ait temel özelliklerin karşılaştırması (Xu ve Tian, 2015)	11
Tablo 3.1. DBSCAN algoritması uygulanması sonucu elde edilen kümelere ait bilgiler ..	25
Tablo 3.2. K-Ortalama algoritmasının uygulanması sonucu elde edilen kümelere ait bilgiler	28



ÖZET

Araç rotalama, lojistik endüstrisinde dağıtım operasyonlarını optimize etmek, yakıt/elektrik tüketimini azaltmak ve teslimat sürelerini iyileştirmek için kullanılmaktadır. Ancak, karmaşıklık seviyesi arttıkça büyük ölçekli rotalama problemlerinin çözümü zor hale gelebilir. Günümüzde bir çok amaç için kullanılan vasıtaların bir noktadan diğer bir noktaya ulaşana kadar izlediği güzergahın üretilmesi kaynak ve zaman ihtiyacı bakımından oldukça maliyetli ve zor bir problem olarak bilinmektedir. Özellikle güzergah oluşturmak için kullanılacak nokta sayısı fazla ise karmaşıklık seviyesi artmakta ve çözülebilir olma özelliği azalmaktadır. Bu durumda, önce kümeleme sonra rotalama yaklaşımı önemli bir çözüm sunar. Bu yaklaşım, büyük ve karmaşık rotalama problemlerini ele almak için önce kümeleme algoritmalarını kullanarak karmaşıklık olan problemi daha küçük ve anlamlı kümelere ayırmak için kullanılan bir yaklaşımdır. Oluşan her bir kümenin içindeki teslimat noktaları arasında optimal rotalar belirlenir. Böylece büyük bir rotalama problemi daha küçük ve yönetilebilir alt problemlere bölünür. Kümeleri daha küçük parçalara ayırmak, alt kümelerin daha az teslimat noktası içermesini sağlamakla beraber, rotalama algoritmalarının daha hızlı çalışmasını ve daha optimize edilmiş rotaların bulunmasını sağlar. Bu çalışmada, farklı kümeleme algoritmaları (k-ortalama, DBSCAN, BIRCH, OPTICS) üç farklı veri seti üzerinde ayrı ayrı çalıştırılarak elde edilen sonuçlar karşılaştırılmıştır. Vektör veri seti için elde edilen bulgulara göre k-ortalama en iyi sonucu ürettiği görülmüştür. Deneysel sonuçlar, önce-kümele sonra-rotalama yaklaşımının hesaplama karmaşıklığını düşürerek araç rotalama problemlerinde yol mesafesini optimize edip seyahat maliyetini düşürdüğünü göstermektedir.

Anahtar kelimeler: Makine öğrenmesi, araç rotalama, kümeleme, dbscan, k-ortalama

ABSTRACT

Investigation of Cluster-First Route-Second Methods for Vehicle Routing Problem Using Machine Learning

Vehicle routing is an important operation in the logistics industry to optimize travel, reduce fuel/electricity consumption, and improve delivery times. However, as the complexity increases, solving large-scale routing problems can become challenging. In today's world, generating the route that vehicles follow from one point to another is known to be a costly and challenging problem in terms of resources and time, given the various purposes for which vehicles are used. Particularly when the number of points used for route construction is substantial, the level of complexity escalates, and the feasibility of resolution diminishes. In this scenario, the cluster-first route-second approach presents a significant solution. This approach involves using clustering algorithms initially to decompose the complexity of the problem into smaller and meaningful clusters, aiming to address large and intricate routing problems. Optimal routes are determined between delivery points within each of the resulting clusters. Consequently, a large routing problem is divided into smaller and more manageable sub-problems. Dividing clusters into smaller segments ensures that sub-clusters contain fewer delivery points, facilitating faster operation of routing algorithms and the discovery of more optimized routes. In this study, the results obtained by different clustering algorithms (K-Means, DBSCAN, BIRCH, OPTICS) on three different datasets were compared. According to the findings obtained for the main dataset, K-Means produced the best results. Experimental results demonstrate that the cluster-first route-second approach reduces computational complexity and optimizes travel distances, thereby reducing travel costs in vehicle routing problems.

Keywords: Machine learning, vehicle routing, clustering, dbscan, k-means

1. GİRİŞ

Araç rotalama problemi'ni ilk olarak, merkezi bir noktadan birden fazla kamyonun, birden fazla benzin istasyonunun petrol talebini nasıl karşılayabileceğini ve minimum seyahat edilen mesafeyi nasıl sağlayabileceğini modelleyerek tanıtan kişiler Dantzig ve Ramser (1959) olmuştur. 1964 yılında, Clarke ve Wright, bu problemi lojistik ve taşımacılık alanında yaygın olarak karşılaşılan bir lineer optimizasyon problemine genelleştirmeyi başardı. Yaptıkları iş değişen kapasitelere sahip bir kamyon filosunu kullanarak merkezi depo etrafında coğrafi olarak dağılmış bir dizi müşteriye nasıl hizmet edeceğini belirlemek olmuştur (Braekers ve ark., 2016). Bu Araç Rotalama Problemi (ARP) olarak bilinir ve çok çalışılan konulardan biridir. Literatürde (ARP'nin eski ve yeni varyantları için) tanıtılan çözüm yöntemlerinin sayısı son yıllarda hızla artmış olup mevcut bilgisayarların işleme hızı ve bellek kapasitesi genişlediğinden, ARP'nin daha büyük örneklerini çözmeyi mümkün hale getirmekle beraber araştırma alanındaki ilerlemeyi ve ARP için ticari yazılımların geliştirilmesine hız kazandırmıştır. ARP birçok şirket tarafından çeşitli endüstri sektörlerinde kullanılmaya başlandığı için günümüzde ARP'nin çözüm üretmesi gereken problem seti büyüklüğü ARP'nin ilk zamanlarına göre daha yoğun hale gelmiştir. Yoğunluğun bu şekilde artmış olması ARP'nin çözüm zamanını uzatmakta, bazı problem setlerine de çözüm bulamayabilmektedir. Bu amaçla büyük bir problemi daha küçük anlamlı gruplara ayırarak ARP'nin bu küçük gruplara uygulanması daha efektif ve hızlı sonuç üretmektedir.

Halk sağlığını tehdit eden bulaşıcı hastalıkların çoğu sivrisinek, karasinek, tatarcık, hamamböceği, çeçe sineği, insan biti ve diğerleri gibi böcek vektörleri tarafından bulaşmaktadır (Naqqash ve ark., 2016). Karasinek, insanların yaşadığı bölgelerde gelişen zararlılardan biridir ve çöplükler, larvalarının gelişimi için özellikle uygun bir ortamdır. Karasinekler dış vücutlarında patojen mikroorganizmalar taşımalarıyla ünlüdür, bu da onları insanlar ve hayvanlar arasında cüzam, tifo, difteri, tüberküloz, dizanteri, kolera, bağırsak parazitleri ve şarbon gibi hastalıkların bulaşmasında etkili ajanlar haline getirir (Khamesipour ve ark., 2018). Bu tür hastalıkların, hızla yayılabilme ve salgına dönüşme potansiyeli yüksektir. Salgın riskinin azaltılması veya tamamen ortadan kaldırılabilmesi için doğrudan ve dolaylı hastalık bulaşma ortamlarının (rezervuarların) kontrol altına alınması gerekebilir. Bu tür bulaşıcı bir etkenin yaşayabildiği, gelişebildiği veya çoğalabildiği insan, hayvan ya da çevreye rezervuar denilmektedir. Vektör kaynaklı (karasinek, sivrisinek, fare.

vb.) hastalıkların yayılmasını önlemek amacıyla yaşam ve üreme alanlarını oluşturan rezervuarlara (dere, kanal, bodrum, fosseptik, menhol, park, bahçe, metruk ortam vb.) yaygın olarak kimyasal ilaçlar (pestisit) ile müdahale edilmektedir. Bununla birlikte, bu tür ortamlarda kimyasalların (pestisitler) kullanılması, çalışma ortamları ve binalar gibi bu yaşam ortamlarında insanların sağlığı ve refahı için potansiyel bir tehdit oluşturduğundan, bu yöntem riskleri de beraberinde getirmektedir (Duggal ve Siddiqi, 2008). Ek olarak pestisit türü kimyasalların aşırı kullanımı halinde, zararlılarda kullanılan ilaçlara karşı direnci tetikleyerek sıtma ve ishal gibi haşere kaynaklı hastalıkların yayılmasına ve hastalık kontrol programlarının başarısız olmasına neden olabilmektedir. Bu nedenle birçok ülkede bu tür zararlı kimyasalların kullanımı sıkı şekilde takip edilmektedir. Örnek olarak Türkiye'de yaşam alanlarında tarım dışı ilaçlama faaliyetleri Sağlık Bakanlığı'nın belirlediği ve denetlediği esaslara göre ilgili belediyelerde bulunan Zararlılarla Mücadele ve İlaçlama Hizmetleri Dairesi Başkanlıkları tarafından yürütülmektedir (URL-4). Türkiye, Dünya'daki birçok ülkeden daha büyük bir yüz ölçümü alanına sahiptir. Bu nedenle şehirlerin (özellikle büyükşehirlerin) yetki ve sorumluluk alanları da fazla olmaktadır. Ayrıca Dünya üzerindeki fiziksel konumu nedeniyle ülke içerisinde aynı anda birçok mevsim yaşanabilmektedir. Bu durum birçok konuda yararlı olsa da farklı mevsim ve iklim şartları yıl içerisinde vektör tür ve sayılarında da değişkenliklere yol açabilmektedir. Bundan dolayı iş gücü ve ekipman başta olmak üzere tüm kaynakların doğru ve verimli şekilde koordine edilmesi ve kullanılması daha önemli hale gelmektedir. Rezervuar alanlara pestisitlerin uygulanması aşamasında teknolojik olarak birçok farklı araçtan (drone vb.) faydalanılmaktadır (Meng ve ark., 2018; Wang ve ark., 2019). Fakat genel olarak sıvı depolayabilen ve deposundaki sıvıyı tazyikli şekilde püskürtebilme sistemine sahip araçlar kullanılmaktadır. Araçlar ilk olarak ortak bir merkezde depolarını pestisit ile doldurmaktadır. Daha sonra güzergahlarını belirlemekte ve kendilerine atanmış güzergahta bulunan tüm noktalara (rezervuar alanlara) uğrayarak ilaçlama işlemini gerçekleştirmektedirler. İnsan kaynağı, yakıt, ilaç, zaman vb. kaynakların etkili ve verimli şekilde kullanılabilmesi için araçlara uygun miktarda sıvı depolanmalı, personel ataması yapılmalı, yakıt tedariği sağlanmalı, güzergahı doğru şekilde planlanmalıdır. Aksi durumda uğranılmayan noktalar olabilir ve bu durum salgına neden olabilir ya da aynı yere birden fazla araç uğrayabilir veya ya birden fazla kez uğranabilir. Fazla ilaçlama durumunda da vektörler ilaca karşı direnç kazanabilir ve salgına neden olabilir veya alerjik durumların tetiklenmesine yol açılabilir.. Tüm bunların yanı sıra yakıt, zaman, iş gücü, ilaç gibi temel ve önemli kaynaklarda israf oluşabilmektedir. Güzergahların

optimum ve planlı bir şekilde belirlenmesi bu tür kaynakların daha iyi kullanılmasını sağlayabilir.

Gezgin satıcı problemi (Travelling Salesman Problem - TSP) (Menger, 1932), kombinatoriyal optimizasyon alanında 1900 lü yılların başından itibaren sıklıkla kullanılan bir yaklaşımdır (Pop ve ark., 2024). İlaçlanacak rezervuar alanlar göz önünde bulundurulduğu zaman Gezgin Satıcı Probleminde temel amaç tüm rezervuar alanları en düşük maliyete göre turlamasının sağlanması olarak söylenebilir. Gezgin Satıcı Problemlerinde kapasite ve zaman ile ilgili kısıtlamalar bulunmamaktadır. Bu nedenle eğer araç sayısının birden fazla olması, vektörlere özgü ilaçlama zamanları vb. değişkenler göz önüne alındığı zaman Dantzig ve Ramser (Dantzig ve Ramser, 1959) tarafından önerilen Araç Rotalama Problemi (Vehicle Routing Problem - VRP) yaklaşımı daha uygun bir yaklaşım olacaktır (Bai ve ark., 2023). Bu yaklaşım sayesinde araç sayısı, kapasite vb. birçok değişken göz önünde bulundurulabilmektedir. Araç Rotalama Problemleri için kesin algoritmalar, sezgisel yöntemler ve hiyerarşik yöntemler gibi bir çok farklı yaklaşım bulunmaktadır (Chen ve ark., 2010; Hasugian ve ark., 2023; Hawks, 2023; Sajid ve ark., 2022). Hiyerarşik yöntemlerden birisi olan önce kümele, sonra rotala yaklaşımının ilk olarak önceden belirlenmiş noktaları belirli kriterlere göre gruplamakta ve daha sonra ise gruplar içerisindeki noktalara göre en az maliyete sahip güzergahı hesaplamaya dayanmaktadır.

1.1. Literatür Taraması

Yapılan çalışmada problem çözümü için hiyerarşik bir yaklaşım olan önce kümele, sonra rotala yaklaşımı önerilmektedir. Kümeleme aşamasında Makine öğrenmesi yöntemleri, rotalama aşamasında ise kesin algoritmalar kullanılmaktadır. Literatürde ilgili yöntem ve tekniklerin kullanım alanlarının çok geniş olduğu görülmektedir. Nallusamy ve ark., (2009) tarafından yapılan çalışmada hangi şehirlere hangi araçların hizmet verdiğini belirlemek için K-ortalama algoritmasından yararlanılmıştır. Elango ve ark., (2011) çoklu robot sistemlerindeki robotları ilk olarak k-ortalama yöntemi ile kümelemekte ve daha sonra ise çoklu gezici satıcı problemi (MTSP) yöntemi ile robot ile görevler arasındaki mesafeyi azaltarak maliyeti düşürdüklerini öne sürmektedirler. Yantur ve ark. tarafından yapılan çalışmada, beyaz eşyaların ana depodan Ankara ve çevre illerdeki evlere veya bayilere teslimine yönelik rotaların teslimat süresi penceresi kısıtı dikkate alınarak hesaplaması yapılmaktadır. “Esnek zaman pencereli araç rotalama” problemleri sınıfında olan problem

için DBSCAN kümeleme yöntemi özelleştirilerek kullanılmakta ve oluşan kümelere göre de güzergâh tahmini yapılmaktadır. Uygulama sonucunda önerilen yaklaşımları sayesinde araç sayısında %30'luk bir iyileşme, araç doluluk oranlarında ise ortalama %94,89 luk bir artışın elde edildiğini belirtmişlerdir (Yantur ve ark., 2024). Alişan ve arkadaşları (Alisan ve ark., 2023) bir çalışmada böceklerin genellikle insanların yaşadığı alanlarda insan sağlığına ve çevreye olumsuz etkileri olan organizmalar olarak bilindiğini belirtmiştir. Bu problemi çözmek için, çalışmalarında önceden belirlenmiş böcek yuvalarının koordinatlarını kullanmışlardır. Bu yuvalar, belirlenmiş ölçütlere göre belediye sınırları içinde k-ortalama yöntemi kullanılarak kümelemişlerdir. Daha sonra her kümeye ait aracın kat etmesi gereken en düşük mesafeye sahip güzergahın elde edilebilmesi için TSP yöntemine dayalı Google or-tools kullanılarak en az maliyete sahip güzergâh hesaplamaları gerçekleştirmişlerdir. Wu ve arkadaşları (Wu ve ark., 2023) yenilikçi bir EMT algoritması olan EMT-RD'yi sunmaktadır. Bu algoritma, gerçek ve kesikli çözümler arasında bilgi transferini kolaylaştırmak için çift yönlü uyarlamalı bir codec kullanmaktadır. EMT-RD algoritması, her görev için gerçek ve kesikli çözümlerin transferine izin verir ve her görev için gerçek çözümlerden kesikli çözümlere bilgi dönüştüren bir kodlama kuralı kullanır. Ayrıca, kesikli çözümlerden gerçek çözümlere dönüştüren bir çözümleme kuralı, arama deneyimlerini oluşturur ve gerçek çözümler arasında bilgi transferini kolaylaştırır. Mekanizma, farklı görevler arasında transfer işlemlerini başlatan uyarlanabilir bir transfer stratejisi ile tamamlanmaktadır. Her görev, bu çalışma için geliştirilen 168 zaman penceresi çoklu görev örneği çiftleri ve bu çalışma için oluşturulan dört gerçek dünya problemi çifti üzerinde kapsamlı deneylerle ele alındığında, önerdikleri EMT-RD algoritmasının rakiplerini geride bıraktığını vurgulamaktadırlar. Lopez-Castro ve arkadaşları (López-Castro ve ark., 2023) heterojen filooya sahip ham süt toplama rotalarını belirlemeye odaklanmış ve karbondioksit operasyonel emisyonlarının ortalamasını en aza indirmeyi amaçlamışlardır. Problemleri çözmek için, Birleşik ve Sınırlı (Branch and Bound) formülasyona dayalı kesin bir yöntem ve Yinelemeli Yerel Arama (ILS) temelli çoklu başlangıçlı bir heuristik kullanılmıştır. Analiz, rota tasarımı açısından, çevresel bir yaklaşımın araçların seyahat ettiği mesafeyi artırtabileceğini (%60'a kadar) ancak genel emisyonları azaltabileceğini ve ayrıca büyük ve kirletici araçların bir kısmının daha düşük emisyonlu küçük araçlarla değiştirilmesinin önemini ortaya koymuşlardır.

1.2. Tezin Katkısı ve Organizasyonu

Yapılan çalışmada VRP çözümü için hiyerarşik bir yaklaşım olan önce kümele, sonra rotala yaklaşımları incelenmiştir. Yaklaşım iki aşamadan oluşmaktadır. İlk aşama olan kümeleme aşamasında, farklı algoritmalarından (DBSCAN, BIRCH, OPTICS, K-Ortalama, gibi) faydalanılarak kümeleme işlemleri gerçekleştirildi. Rotalama aşamasında ise oluşan her küme için az maliyete sahip güzergâh oluşturuldu. Daha sonra ise kullanılan kümeleme yöntemlerinin etkililiğini göstermek için 3 gerçek veri seti üzerinde uygulanarak sonuçlar karşılaştırıldı.

Kümeleme ve araç rotalama alanındaki yaklaşımlar, kümeleme yöntemlerinin yapısı ve uygulaması, kümeleme ve araç rotalamanın birlikte uygulanması ile ilgili bir kılavuzun oluşturulması ve sunulan yaklaşımların uygulanabilirliğinin gösterilmiş olması bu tez çalışmasının katkıları arasında sayılabilir. Bu tez çalışmasındaki yaklaşımdan yararlanarak hesaplama zaman karmaşıklığı yüksek olan araç rotalama problemi alt kümelere indirgenerek hesaplama ve çözüm geliştirilme gibi konularda kolaylık sağlayacaktır.

Bu tez dokümanı şöyle düzenlenmiştir: Birinci bölümde giriş yapıp literatür ve tezin katkısından bahsedilmiştir. İkinci bölümde ise materyal ve metotlar açıklanmıştır. Üçüncü bölümde, çalışmada yapılan deneysel uygulama ve çıktılarına yer verilmiştir. Son bölümde sırasıyla sonuçlar, öneriler, kısıtlamalar ve potansiyel gelecek çalışmalar hakkında bilgiler sunulmaktadır.

2. MATERYAL VE METOT

Kümeleme ve rotalama yöntemlerinin etkili bir şekilde uygulanıp hızlı ve efektif bir şekilde geliştirilebilmek adına belli başlı seçimler yapılmış olup programlama dili ve kütüphaneleri için Python dili ve desteklediği kütüphaneler yönünde tercih yapılmıştır. Yaygın olarak kullanılan ve python desteği sunan scikit-learn ve pycaret temel makine öğrenmesi algoritmaları için tercih edilip veri işleme ve görselleştirme için pandas, numpy ve matplotlib kütüphaneleri kullanılmıştır. Ayrıca harita ve coğrafi verilerle çalışıldığı için folium ve osmnx kütüphaneleri tercih edilmiştir. Optimizasyon problemi çözümü için de ortools kütüphanelerinden faydalanılmıştır.

2.1. Kullanılan yazılım araçları

Makine öğrenmesi alanında etkili bir uygulama geliştirebilmek için uygun programlama dili ve bu dilin desteklediği kütüphaneleri seçmek büyük önem arz etmektedir. Makine öğrenmesi projeleri çoğunlukla Python dili gibi yüksek seviyeli diller ile geliştirilmektedir. Python, makine öğrenmesi uygulamaları için geniş bir ekosistem sunmakla beraber güçlü ve popüler kütüphane desteği sağlar. Örneğin, scikit-learn ve pycaret çoğunlukla temel makine öğrenmesi algoritmaları için tercih edilirken veri işleme ve görselleştirme göz önünde bulundurulunca pandas, numpy, seaborn ve matplotlib gibi kütüphaneler yaygın olarak kullanılabilir. Bu kütüphaneler, veri analizi, veri temizleme ve elde edilen sonuçların görselleştirilebilmesi gibi önemli adımları az zaman ve efor ile yapılabilmesine yardımcı olur. Ayrıca harita ve coğrafi verilerle çalışırken folium ve osmnx gibi kütüphaneler tercih edilebilir. Optimizasyon problemi çözümleri için ise ortools gibi kütüphaneler ön plana çıkmaktadır. Tüm bu kütüphaneler, farklı ihtiyaçlara yönelik çözümler sunar ve uygulamanın geliştirilmesi için kolaylık sunar. Doğru dili ve uygun kütüphaneleri seçmek, uygulamanın performansı ve genişletilebilir olma özelliği üzerinde büyük etkiye sahiptir. Python, Guido van Rossum tarafından geliştirilmiş ve 1991 yılında piyasaya sürülmüş yorumlanabilen ve yüksek seviyeli bir programlama dilidir. Python dilinin hedefleri arasında basitlik, okunabilirlik ve üretkenlik yer almaktadır. Dilin tasarım felsefesi, kodunun sade ve anlaşılır olmasını sağlar. Python gibi yorumlanan dillerin avantajlarından biri, yazılım geliştirme sürecini hızlandırmaya yardımcı olabilmesidir. Kodunuzda hızlı bir şekilde değişiklikler yapabilir, hataları hızlıca tespit edebilir ve test

edebilirsiniz. Ancak yorumlanan dillerin bir dezavantajı, genellikle derlenmiş dillere göre daha yavaş olmalarıdır. C, C++, Java gibi diller kaynak kodunun çalışmadan önce bir derleyici tarafından önceden derlenmesini gerektirir. Derleyici, kaynak kodu daha düşük seviyeli makine diline dönüştürür ve bu derlenmiş kod daha sonra doğrudan çalıştırılabilir dosyaya dönüştürülür. Derlenen diller genellikle daha hızlıdır, ancak geliştirme süreci daha fazla zaman alabilir. Python'un yorumlanan bir dil olması, hızlı prototip üretme, dinamik kod değişiklikleri ve kolay hata ayıklama gibi avantajlar sağlar. Göz önünde tutulması gereken önemli noktalardan birisi kolay okunabilir olmasıdır. Okunabilirlik, hazırlanmış bir uygulamanın komutlarının farklı insanlar tarafından rahatlıkla anlaşılabilir olması, genişletilebilir ve sadeleştirilebilir olma durumu olarak açıklanabilir. Tasarlanma aşamasında okunabilirlik özelliğine oldukça önem verilen Python programlama dilinde, ifade blokları oluşturulurken araç karakterleri yerine, boşluktan oluşan girintiler kullanılır. Birçok yüksek seviyeli programlama dilinde, komutlar çalıştırıldığı zaman bilgisayarın anlayabileceği yapıya dönüştürülmektedir. Python dili, basit işlemlerden, üst düzey karmaşık işlemlere kadar kullanılabilirliği kolay olan kütüphaneler sunmaktadır. Ayrıca “Matlab”, “R” veya “C” gibi dillerle entegre olarak çalışabilmektedir. TIOBE (The Importance Of Being Earnest) tarafından yapılan araştırmada, arama motorlarına ait veriler analiz edilerek, arama motorları aracılığıyla aranılan programlama dilleri güncel sıralamasında Python dili en çok kullanılan genel programlama dili olarak 1. sırada bulunmaktadır (URL-2). Google Trends verilerine dayanılarak yapılan farklı bir çalışmada, sorgu index sayılarına bakıldığında, Python’ın ilk sırada olan popüler bir programlama dili olduğu sonucuna ulaşılmıştır (URL-3). Yaygın olarak kullanılmasından dolayı birçok kütüphanenin oluşturulmasında da rol almıştır. Scikit-learn, Python tabanlı bir makine öğrenimi kütüphanesidir. Bu kütüphane, çeşitli temel ve gelişmiş makine öğrenimi tekniklerini uygulamak için geniş bir araç seti sunar. Scikit-learn, açık kaynaklı olması, kolay kullanımı ve Python'un genel popülerliği nedeniyle araştırmacılar, veri bilimcileri ve endüstri profesyonelleri arasında yaygın olarak tercih edilmektedir.

Scikit-learn, sınıflandırma, regresyon, kümeleme, boyut azaltma gibi çeşitli temel modelleri destekler. Bununla birlikte, kapsamlı bir model seçimi, özellik seçimi ve değerlendirme araçları koleksiyonu sunar. Bu araçlar, kullanıcıların model performansını değerlendirmelerine, en iyi modeli seçmelerine ve veri özelliklerini en iyi şekilde kullanmalarına olanak tanır. Scikit-learn'ün sunduğu temel özellikler arasında, çapraz doğrulama, hiper parametre ayarlama, öznelik ölçeklendirme ve öznelik mühendisliği

gibi veri ön işleme adımları da yer alır. Bu adımlar, veri setlerini model eğitimi için hazırlamak için kritik öneme sahiptir. Ayrıca, Scikit-learn belgelendirilmiş API'leri ve standartlaştırılmış arayüzleriyle birlikte donatılmıştır. Bu da, farklı algoritmalar arasında kolayca geçiş yapmayı veya yeni algoritmalar geliştirmeyi kolaylaştırır. Kütüphane, bir dizi kullanıcı dostu fonksiyon ve sınıf aracılığıyla model oluşturmayı, eğitmeyi, tahmin üretmeyi ve sonuçları değerlendirmeyi sağlar. Genel olarak, Scikit-learn, Python'u temel alarak makine öğrenimi uygulamaları geliştirmek isteyen araştırmacılar ve endüstri profesyonelleri için güçlü ve esnek bir araçtır. Özellikle, model oluşturma, veri analizi ve özellik mühendisliği gibi makine öğrenimi süreçlerinin birçok yönünü desteklemesiyle dikkat çeker. Bu, disiplinler arası bir yaklaşımla, teorik bilgiyi pratik uygulamalarla birleştirme imkanı sunar. Coğrafi bilgi sistemleri (Geographic Information Systems - GIS) ve ağ analizi gibi uygulamalar göz önünde bulundurulduğunda Osmnx kütüphanesi ön plana çıkmaktadır. Osmnx, Python programlama dili için geliştirilmiş açık kaynak kodlu bir kütüphanedir. Osmnx, genellikle 'open street map' gibi açık veri kaynaklarından coğrafi veri toplamak, analiz etmek ve görselleştirmek için güçlü araçlar sunar. Haritalar oluşturmak ve bu haritaları etkileşimli olarak görselleştirmek için ise folium kütüphanesi kullanılmaktadır. Folium, kullanıcıların python programlarında haritaları oluşturmak için basit bir arayüz sunarak HTML veya JavaScript bilgisine ihtiyaç duymadan, python kodu kullanarak haritaları hızlı bir şekilde oluşturulmasına yardımcı olur.

2.2. Gezgin Satıcı Problemi

Gezgin satıcı problemi (TSP) günümüze kadar birçok türe sahip olan ve birçok araştırmacı tarafından ele alınıp üzerinde çalışılan bilgisayar bilimindeki en eski problemlerden biri olarak kabul edilir. Seyahat eden satıcı problemi olarak da ifade edilebilen gezgin satıcı problemi çok zor bir kombinasyonel optimizasyon problem olarak değerlendirilir (Garey ve Johnson, 2009; Wei ve ark., 2021). Problemin tanımı şudur: Verilen bir şehir kümesinden tam bir tur bulma; her bir şehrin sadece bir kez ziyaret edilmesi gerekmekte ve tüm şehirler bu tur dikkate alınarak ziyaret edilmelidir. Şehirlerarası mesafeye dayalı olarak, simetrik ve asimetrik olmak üzere iki farklı türde problem bulunur. Simetrik TSP'de, iki şehir arasındaki ters yönlü mesafe, bu iki şehir arasındaki normal mesafe kadardır, ancak asimetrik olanında eşit değildir. Asimetrik olanlar, tek yönlü yollar ve trafik sıkışıklıkları nedeniyle gerçek yaşam problemleri için daha uygun görülebilir.

TSP'nin tanımı kolaydır, ancak orta veya büyük boyutlu bir örnek için bir çözüm elde etmek çok zor olabilmektedir. Optimum bir çözüm için tüm olası yolları karşılaştırıp en kısa olanı seçmek ana hedef olarak kabul edilmektedir. Bir problemin n şehirlik bir seti için $n!$ çözüm bulunabilmektedir ve büyük örnekler için çok fazla olası çözüm olduğu görülmektedir. Karmaşıklık teorisinde, TSP, NP-Zor sınıfında olduğu gösterilen temel problemlerden biridir (Garey ve Johnson, 2009). Bu nedenle, TSP için optimum çözümü polinomsal zaman içinde bulmak imkansızdır. Optimum birini bulabilmek için tüm olası çözümleri karşılaştırmak gerekmektedir. Bu problemi çözmek için birçok yöntem kullanılmakta ve çözüm teknikleri birçok alanda uygulanmaktadır. Yöntemleri iki ana kategoriye ayırmak mümkündür: kesin algoritmalar ve sezgisel algoritmalar (Burkhovetskiy ve Steinberg, 2017; Couto, 2018; Geng ve ark., 2011; Liao ve ark., 2012; Wei ve ark., 2021). Kesin algoritmalar en iyi sonuçları üretmektedir ancak daha büyük örnekler için daha fazla işlem süresi ihtiyacı duymaktadırlar. Diğer bir yaklaşım ise sezgisel algoritmalar; bunlar en iyi sonuçları garanti etmezler, ancak kesin algoritmaların ihtiyaç duydukları işlem süresinden daha az işlem süresi gerektirirler. Sezgisel algoritmalar, kullanım tekniklerine göre iki grupta değerlendirilebilirler. İlk grup, boş bir çözümle başlayan başlangıç çözümlerini oluşturmak için yapılan yapılandırmacı sezgisel algoritmalar ve ikinci grup, başlangıç çözümlerle başlayan çözümlerin kalitesini artırmak için kullanılan iyileştirici sezgisel algoritmalar. Bu nedenle, genellikle, iyileştirici sezgisel algoritmalar, yapılandırmacı sezgisel algoritmaların ardından çözümleri iyileştirmek için kullanılır.

2.3. Araç Rotalama Problemi

Araç Rotalama Problemi (VRP) bir veya daha fazla depodan coğrafi olarak dağılmış müşterilere hizmet etmek üzere atanmış araçların optimal rotalarını tasarlama problemini ifade etmektedir (Laporte ve ark., 1988). VRP ürün ve hizmet dağıtımını etkileyen ve bu nedenle literatürde popüler olan alanlardan biridir. Taşıma problemlerini çözmede tanıtılmasından bu yana önemli başarılar elde edilmiştir. Özellikle, optimizasyon tekniklerinin kullanılmasıyla etkili sonuçlar elde edilmiştir. Güçlü formülasyonlarla kesin yöntemler ve heuristikler üzerine bazı teknikler önerilmiş ve bu teknikler VRP'nin çözümünde geliştirilmiştir (Sajid ve ark., 2022).

VRP'de bir rota oluştururken dikkate alınması gereken faktörler şunlardır (Cordeau ve ark., 2002):

- Her oluşturulan rota depoda başlar ve biter,
- Her müşteri sadece bir kez ziyaret edilir,
- Her müşteri sadece bir araç tarafından ziyaret edilir,
- Rotalardaki müşteri talepleri araç kapasitesini aşmamalıdır,
- Herhangi bir rotanın süresi belirtilen zamanı aşmamalıdır,
- Toplam rota maliyetini minimize etmeye yönelik bir amaç olmalıdır.

2.4. Önce Kümele Sonra Rotala Yaklaşımı

Yıllar içinde, Araç Rotalama Problemi için birçok çözüm yöntemi önerildi. Optimal çözümleri aramak amacıyla kullanılan kesin yöntemler genellikle birkaç yüz müşteri içeren problemlerle başa çıkabilmektedir. Ancak problem boyutu arttıkça, sezgisel yöntemlere başvurulması gereklidir (Koç ve Laporte, 2018). Önce kümele, sonra rotala, araç rotalama problemlerine yaklaşmanın yollarından bir tanesidir. Araç rotalama probleminin NP-zor problem olması nedeniyle bu yaklaşımda eldeki problem alt parçalara ayrılmaktadır. Geleneksel kümeleme yöntemleri toplam rota uzunluğunu azaltmayı hedefler. Fakat önce kümele sonra rotala yaklaşımında problem, her birinin bir gezgin satıcı problemi olduğu alt problemlere bölünür. Var olan en son teknoloji gezgin satıcı problemi çözümleri, binlerce müşteriye kadar olan büyük problemleri yönetebilmektedir. Zorluk, rotaların toplam uzunluğunun her bir kümede kolektif olarak en aza indirilmesi için kümelemenin gerçekleştirilmesidir (Fisher ve Jaikumar, 1981). Geleneksel k-ortalama ve k-medyan kümeleme yaklaşımları, kümeleme sonuçlarını genellikle sırası ile küresel ve elmas şeklinde bulmaktadır. Ancak bu yöntemler, kümelemelerinde depo konumunu dikkate almamaktadırlar (Bramel ve ark., 1992; Bramel ve Simchi-Levi, 1995; URL-1). Orta ölçekli problemler için Lagrangian tekniğini kullanıyorlar. Asimptotik olarak optimal olmalarına rağmen, sezgisel yaklaşımları diğer sezgisel yöntemlerle hesaplama açısından rekabet edemiyor (URL-5).

2.5. Makine Öğrenmesi ve Kümeleme Algoritmaları

Son yıllarda, Makine Öğrenimi (Machine Learning - ML), bilgisayar biliminde hızla büyüyen bir alan haline gelmiştir. ML, Yapay Zekâ (Artificial Intelligence - AI) alanının özel bir bölümüdür ve makinelerin veriden bilgi çıkarmalarını sağlayan algoritmaları ve modelleri tasarlar. Bu, makinelerin açık talimatlar olmadan örneklerden ve deneyimlerden öğrenerek zaman içinde performanslarını geliştirebilmeleri anlamına gelir. ML algoritmalarının insan müdahalesi olmadan veriden öğrenme yeteneği birkaç avantaja sahiptir. Örneğin, değişen ortamlara uyum sağlayabilir, yeni verilerden öğrenip sürekli olarak performansını artırabilirler (Shaheamlung ve ark., 2020). Bu, sistemlerin insan müdahalesine ihtiyaç duymadan geliştirilmesine olanak tanır. Genel olarak, ML'ın hızlı büyümesi inovasyon ve ilerleme için heyecan verici fırsatlar yaratmış ve gelecekte teknoloji ve toplumun şekillenmesinde önemli bir rol oynamaya devam etmesi beklenmektedir.

ML algoritmaları terimi, veriden öğrenip zaman içinde daha iyi hale gelebilen bir grup algoritmayı ifade eder. Bu sayede sistem, gelecekte ne olacağını tahmin edebilir ve tek bir kullanıcının veri setinden örnek bir veri kümesi üzerinde desenler bulabilir. Keşifsel veri madenciliği ve bilgisayar tabanlı algoritmalar kullanılarak makineler, farklı kategoriler için temel davranışsal desenleri öğrenip, eğitilmiş verilere dayanarak sonuçları tahmin edebilirler. ML süreci, gerçek sonuçları içeren örneklerle başlar ve algoritmalar veri setinde desen arar. Bu, veri seti içinde sunulan senaryolara dayanarak gelecekte bilinçli kararlar alabilme yeteneğini kolaylaştırır (Valente Klaine ve ark., 2019).

Tablo 2.1. Kullanılan algoritmalara ait temel özelliklerin karşılaştırması (Xu ve Tian, 2015)

Algoritma	Karmaşıklık (zaman)	Ölçeklenebilirlik	Veri setinin şekli	Veri girişi sırasına duyarlı	Gürültü/dışarıya duyarlı
<i>K-Ortalama</i>	$O(knt)$	Orta	Convex	Yüksek	Yüksek
<i>BIRCH</i>	$O(n)$	Yüksek	Convex	Orta	Düşük
<i>DBSCAN</i>	$O(n*\log n)$	Orta	Arbitrary	Orta	Düşük
<i>OPTICS</i>	$O(n*\log n)$	Orta	Arbitrary	Düşük	Düşük

Xu ve Tian'nın 2015 senesinde kümeleme algoritmaların temel özelliklerini Tablo 2.1'de gösterildiği gibi belirtmiştir. Veri analizinde sıkça kullanılıp benzer özelliklere sahip veri noktalarını gruplandırmak için tasarlanmış olan algoritmalar, veri setlerindeki yapıyı

keşfetmek, veriler arasındaki ilişkileri ortaya çıkarmak ile beraber anlamlı bilgiler elde etmek için kullanılabilir.

2.5.1. K-Ortalama

MacQueen (1967) tarafından önerilen k-ortalama (k-means) kümeleme yöntemi, deterministik, sayısal, denetimsiz ve yinelemeli özelliklere sahiptir. Bu yöntem, N adet nesneyi k adet kümeye bölmeyi amaçlar. K-ortalama algoritması, uygun başlangıç merkez noktalarını seçmek için belirli bir parametreye sahip değildir. Ancak, farklı başlangıç merkez noktalarının seçilmesi, özellikle örneklemin içinde birçok aykırı değer bulunduğunda, kümeleme sonuçları arasında büyük farklılıklara neden olabilir. Yöntem, kümeleme çalışmalarında sıklıkla tercih edilen basit ve hızlı bir yöntemdir. İki aşamalı yöntemin ilk adımında, k değeri önceden belirlenir ve bu değere göre k merkezi rastgele seçilir. Ardından, her veri nesnesi en yakın merkeze atanır. Genellikle, Öklid mesafesi kullanılarak her veri nesnesi ile küme merkezleri arasındaki mesafe belirlenir. Tüm veri nesneleri bir kümeye atanana kadar bu adım tekrarlanır ve ön gruplama tamamlanır. Daha sonra, ön gruplama sonucunda oluşan kümelerin anlamı yeniden hesaplanır. Bu yinelemeli süreç, kriter fonksiyonu minimum değerine ulaşana kadar devam eder (Na ve ark., 2010). Aşağıda bu süreç adım adım açıklanmıştır.

K-Ortalama kümeleme algoritmasının adımları:

Adım 1: Belirlenen küme sayısına (K) sahip olan ilk merkez noktaları rastgele seçilir ve başlangıç kümeleri oluşturulur. Bu kümeler $\{C_1, C_2, C_3, C_4, \dots, C_k\}$ şeklinde K küme olarak ifade edilir. C_i kümesinin vektör ortalaması (küme merkezi) M_i Eşitlik (2.1)'deki şekilde ifade edilir. Burada, X_i değeri, C_i kümesine ait i'inci örneği ve n ise veri kümesinin içerdiği veri öğelerinin sayısını sembolize eder.

$$M_K = \frac{1}{n_K} \sum_{i=1}^{n_K} X_{iK} \quad (2.1)$$

Adım 2: Küme içi değişimler, "Karesel Hata Formülü" kullanılarak hesaplanır. C_i için karesel hata formülü, Eşitlik (2.2)'de verilmiştir. Tüm kümeler için karesel hata, Eşitlik (2.3)'de gösterildiği gibi küme içindeki değişimlerin toplamını ifade eder. Karesel hata yönteminin amacı, verilen K değerine bağlı olarak E değerini minimize eden K tane kümeyi bulmaktır.

$$e_i^2 = \sum_{i=1}^{n_K} (X_{iK} - M_K)^2 \quad (2.2)$$

$$E_K^2 = \sum_{i=1}^K e_i^2 \quad (2.3)$$

Adım 3: Her düğüm en yakın merkeze atanarak küme oluşturulur.

Adım 4: Küme sayısı aynı tutularak her kümenin yeni merkezi tekrar hesaplanır.

Adım 5: Kümelerin elemanları sabit kalana kadar ya da algoritmanın durma koşulları sağlanana kadar Adım 2'den itibaren tekrarlanır (Sariman, 2014).

Elbow Method: Elbow metodu, kümeleme algoritmasında optimal küme sayısını belirlemede önemli bir rol oynayan yaygın olarak kullanılan bir sezgisel yöntemdir. Bu yöntem, algoritmada kullanılan küme sayısına karşı küme içi kareler toplamını (Within-Cluster Sum of Squares - WCSS) çizmeyi içerir. WCSS, her veri noktasının atanmış küme merkezi ile arasındaki toplam mesafenin ölçüsüdür ve tüm kümeler üzerinde toplanır. Elbow metodunun temel amacı, WCSS'deki azalma hızının yavaşlamaya başladığı "dirsek noktasını" belirlemektir. Bu, kullanılan küme sayısı ile kümeleme algoritması tarafından açıklanan varyans miktarı arasında bir denge noktasını gösterir.

2.5.2. DBSCAN

Yoğunluk tabanlı kümeleme teknikleri, klasik kümeleme yöntemlerinden farklı olarak uzaklık yerine yoğunluğa dayalı bir yaklaşım kullanır. Yoğunluk tabanlı kümeleme, veri uzayındaki veri noktalarının yoğunlaştığı bölgeleri belirleyerek bu bölgeler arasındaki belirli ölçüm ve parametrelere dayanarak kümeleme işlemini gerçekleştirir (Kaya, 2018). DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algoritması yüksek yoğunluklu bölgeleri bir araya getirerek ve düşük yoğunluklu bölgeleri ayrı olarak tanımlayarak veriyi gruplara ayırır (Ester ve ark., 1996). Kümeler, yoğunluğu yüksek bölgelerin, yoğunluğu düşük bölgelerden ayrılmasıyla belirlenir (Tan ve ark., 2018). Bu özellik, yoğunluk tabanlı yöntemlerde keşfedilen kümelerin yoğunluğun arttığı yönde genişleyebileceği anlamına gelir, bu da farklı şekillerde kümeler elde edilmesine olanak tanır (Berkhin, 2006; Gan ve ark., 2020). DBSCAN kümeleme algoritmasının adımları aşağıda verilmiştir.

DBSCAN kümeleme algoritmasının adımları:

Adım 1: Veri tabanındaki her bir gözlem arasından "ziyaret edilmemiş" bir rastgele başlangıç noktası (gözlem) seçilir.

Adım 2: Seçilen noktanın merkezinde, bir eşik değeri (eps) yarıçapında bir daire çizilir.

Adım 3: Bu daire içinde yer alan noktalar (gözlemler veya komşular), bir eşik değeri (minpts) değerine eşit veya büyükse, merkezdeki nokta çekirdek nokta (core object) olarak adlandırılır ve bu daire içindeki tüm noktalar aynı küme içinde kümelendir. Çekirdek nokta "ziyaret edildi" olarak işaretlenir.

Adım 4: Eğer daire içindeki noktalar, minpts değerinden küçükse, bu nokta "gürültü (aykırı değer, noise)" olarak işaretlenir. Gürültü olarak işaretlenen noktalar daha sonra bir kümeye dahil edilebilir.

Adım 5: Yeniden "ziyaret edilmemiş" bir nokta bulunur. Eğer bu nokta bir kümeye aitse ve bu noktanın Eps yarıçapında çizilen daire içinde, MinPts değerine eşit veya daha fazla nokta varsa, bu nokta çekirdek nokta olarak adlandırılır ve daire içindeki tüm noktalar aynı küme içine dahil edilir. Eğer MinPts değerinden daha az nokta varsa, bu nokta daha fazla noktaya ulaşmak için kullanılamayacağından "sınır değeri (border object)" olarak adlandırılır. Nokta "ziyaret edildi" olarak işaretlenir.

Adım 6: Tüm noktalar gürültü veya ziyaret edildi olarak işaretlenene kadar işlemler tekrarlanır. Tüm noktalar işaretlendiğinde işlem sonlanır.

2.5.3. BIRCH

BIRCH algoritması (Zhang ve ark., 1996), verilen veriler için bir Kümeleme Özellik Ağacı (Clustering Feature Tree - CFT) olarak adlandırılan bir ağaç oluşturur. Veri esas olarak kayıplı bir şekilde bir dizi Kümeleme Özellik düğümüne (Clustering Feature nodes - CF Düğümleri) sıkıştırılır. CF Düğümleri, Kümeleme Özellik alt kümeleri (CF Alt Kümeler) olarak adlandırılan bir dizi alt küme içerir ve bu CF Alt Kümeler, terminal olmayan CF Düğümleri içinde bulunabilir. CF Alt Kümeleri, bellekte tam giriş verilerini saklama ihtiyacını ortadan kaldıran kümeleme için gerekli bilgileri içerir. Bu bilgiler şunları içerir: Bir alt kümedeki örnek sayısı, Doğrusal Toplam-Tüm örneklerin toplamını içeren n-boyutlu bir vektör, Kare Toplam-Tüm örneklerin kare L2 normunun toplamı, Merkezler-Yeniden hesaplama gereksinimini önlemek için doğrusal toplam/n_samples, Merkezlerin kare normu.

BIRCH algoritmasının eşik (threshold) ve dallanma faktörü (branching factor) olmak üzere iki parametresi bulunmaktadır. Dallanma faktörü, bir düğümdeki alt küme sayısını sınırlar, eşik ise giren örnek ile mevcut alt kümeler arasındaki mesafeyi sınırlar. Bu algoritma, giriş verilerini doğrudan CFT'nin yapraklarından elde edilen bir dizi alt kümeye indirgeyerek bir örnek veya veri azaltma yöntemi olarak görülebilir. Bu azaltılmış veriler daha sonra bir küresel kümeleyiciye verilerek daha fazla işlenebilir. Bu küresel kümeleyici, $n_küme$ ile ayarlanabilir. Eğer $n_küme$ belirlenmezse, yapraklardaki alt kümeler doğrudan okunur, aksi takdirde bir küresel kümeleme adımı bu alt kümeleri küresel kümeler (etiketler) olarak etiketler ve örnekler en yakın alt kümenin küresel etiketine atanır.

Algoritmanın çalışma mantığı şu şekilde: Yeni bir örnek, CF Ağacı'nın köküne (bir CF Düğümü) eklenir. Ardından, bu örnek kökün alt kümesi ile birleştirilir; bu birleşmeden sonra en küçük yarıçaplı alt küme, eşik ve dallanma faktörü koşulları tarafından sınırlanır. Eğer bu alt kümenin çocuk düğümü varsa, bu işlem yaprak bir düğüme ulaşıncaya kadar tekrarlanır. Yaprakta en yakın alt kümeyi bulduktan sonra, bu alt kümenin özellikleri ve üst alt kümeleri yinelemeli (rekürsif) olarak güncellenir. Yeni örneği ve en yakın alt küme ile birleştirilen alt kümenin yarıçapı, eşik karesinden daha büyükse ve alt küme sayısı dallanma faktöründen büyükse, bu yeni örneğe geçici bir alan tahsis edilir. İki en uzaktaki alt küme alınır ve bu alt kümeler arasındaki mesafeye dayalı olarak alt kümeler iki gruba ayrılır. Eğer bu bölme düğümünün bir üst alt kümesi varsa ve yeni bir alt küme için yer varsa, üst alt küme ikiye bölünür. Yer yok ise, bu düğüm tekrar ikiye bölünür ve bu süreç kök düğüme ulaşıncaya kadar özyinelemeli olarak devam eder.

2.5.4. OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure), veri kümesindeki yoğunluk tabanlı bir kümeleme algoritmasıdır. Bu algoritma, veri setinde belirlenmiş bir noktanın çevresindeki yoğunluğu ölçer ve bu bilgiyi kullanarak veri noktalarını birbiriyle ilişkilendirir. OPTICS algoritması, DBSCAN algoritmasındaki epsilon (Eps) ve minimum nokta (MinPnts) değerlerinden oluşan bağımlılığı azaltmak üzere Ankerst ve arkadaşları tarafından geliştirilmiştir (Ankerst ve ark., 1999). Algoritma girdi olarak sadece minimum nokta parametresine ihtiyaç duymaktadır.

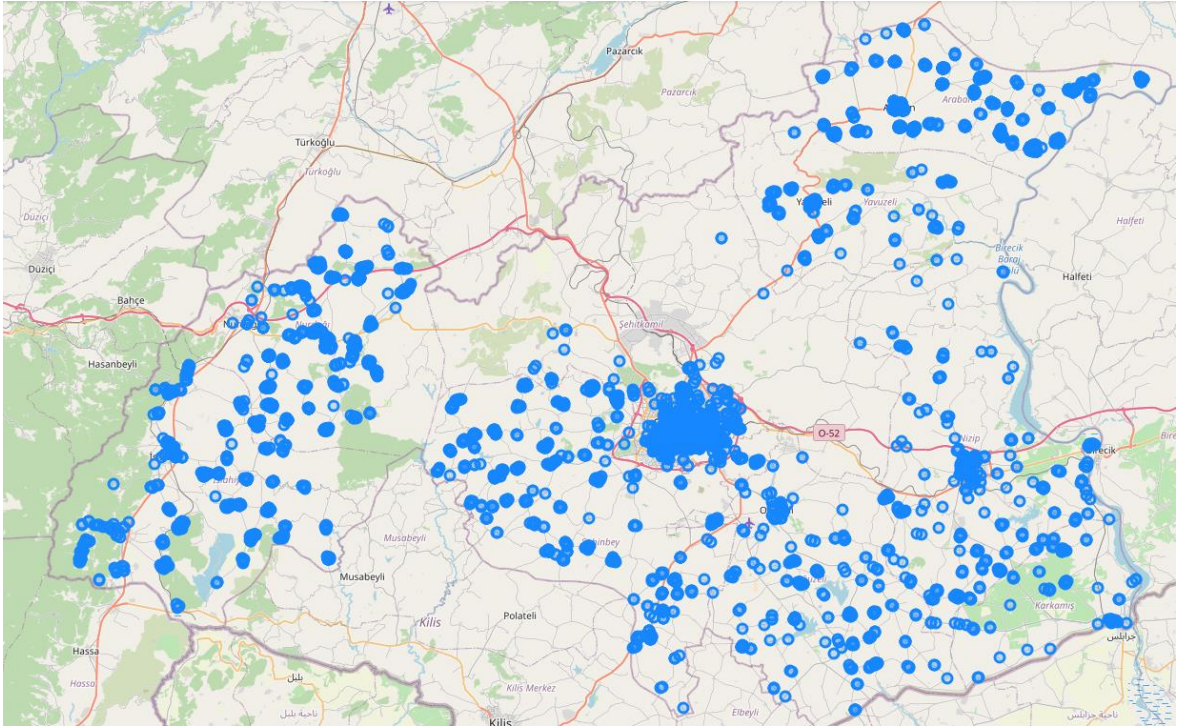
Algoritma, her bir noktayı bir erişilebilirlik mesafesi ile ilişkilendirerek çalışır. Bu mesafe, veri kümesindeki diğer noktalardan bu noktaya ulaşmanın ne kadar kolay olduğunu ölçer. Benzer erişilebilirlik mesafesine sahip noktaların aynı kümede olma olasılığı yüksektir. OPTICS algoritmasının temel adımları arasında yoğunluk eşiği parametresinin tanımlanması, her bir noktanın en yakın komşularına olan mesafesinin hesaplanması ve erişilebilirlik grafiğinin oluşturulması bulunmaktadır. OPTICS, DBSCAN gibi diğer yoğunluk tabanlı kümeleme algoritmalarının aksine, farklı yoğunluklara ve şekillere sahip kümelerle başa çıkabilir ve önceden belirli sayıda kümenin belirlenmesini gerektirmez, bu da küme seçiminde daha fazla esneklik sağlar.

2.6. Veri seti

Bu çalışmada üç farklı veri seti kullanılmıştır. Veri seti-1, 10 Şubat 2016 ile 17 Haziran 2019 tarihleri arasında, Gaziantep ilinin 9 ilçesi ve 296 mahallesinde gerçekleşen 4.658 vektörle mücadele işlemlerinden elde edilen bilgilerden yararlanılarak oluşturulmuş koordinat bilgilerini içermektedir. Veri seti-2, Bulgaristan'daki 1954 gerçek yerleşim yerinin GPS koordinatlarını içermektedir (URL-6, 2023). Veri seti-3'ün içerisinde ise Romanya'daki 2950 gerçek yerleşim yerinin GPS koordinatları bulunmaktadır (URL-7, 2023).

3. BULGULAR VE TARTIŞMA

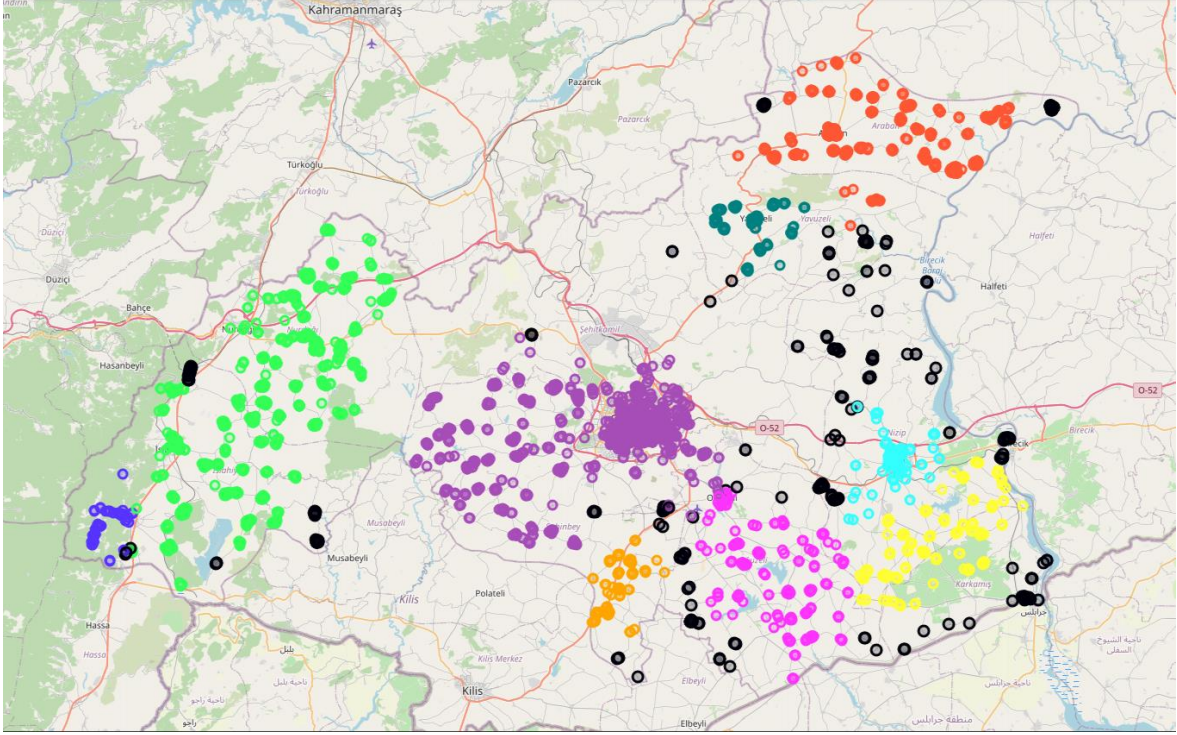
Çalışmada kullanılan veri seti-1 4658 noktaya ait koordinat verisi barındırmaktadır. Bu pestisit kullanılarak karasinek vektöründen arındırılması gereken rezervuar yerleri Şekil 3.1’de harita üzerinde gösterilmektedir. Yoğunluğu bu şekilde yüksek olan veri seti ile bir araç rotalama problemi veya gezgin satıcı problemi için veri bütün bir şekilde ele alındığında hesaplama maliyeti yüksek çıkmaktadır.



Şekil 3.1. Veri seti-1’de bulunan rezervuar alanların harita üzerinde gösterimi

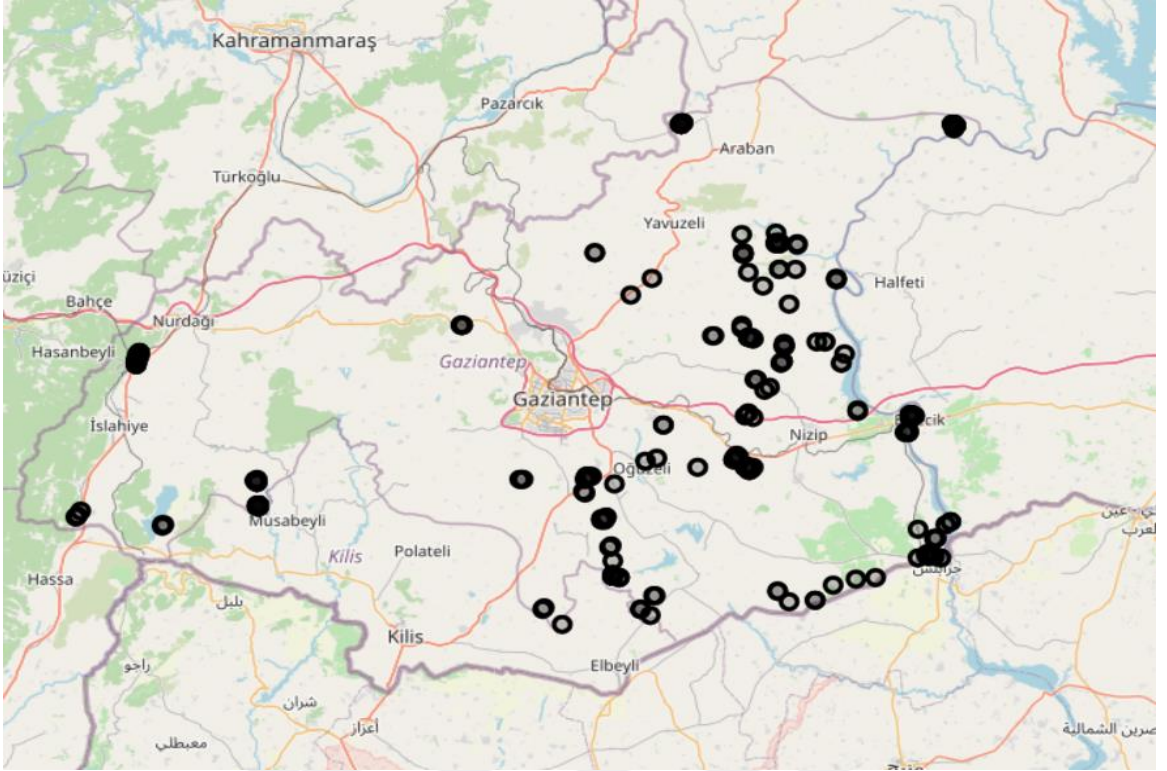
Bazı alanlarda rezervuar sayısının yoğunlaştığı ve diğer alanlarda ise bunun aksine rezervuar sayısının seyrekleştiği görülmektedir. Bunu göz önünde bulundurunca alanların bir bütün olarak ele alınması yerine, bölünerek daha küçük gruplar halinde ayrı ayrı çözüm üretilebileceği görülebilmektedir. DBSCAN algoritması veri seti üzerinde, manuel parametre düzenlemeleri yapılarak, uygulanmış olup elde edilen sonuçlara göre noktalar 9 kümeye ayrılmıştır. Parametreler sırası ile epsilon değeri 0.05247, minimum nokta sayısı da 50 seçilmiştir. Gaziantep idari bölgeleri de düşünüldüğünde böyle bir kümelemenin uygun olabileceği düşünülebilir.

DBSCAN algoritmasının veri setlerine uygulanmasında herhangi bir kümeye yerleştirilemeyen düğümler de bulunabilmektedir. DBSCAN algoritmasının veri setine uygulanması sonucu gürültü olarak elde edilen rezervuar alanlar diğer kümelerle birlikte toplu olarak Şekil 3.2’te (siyah renk), ayrı olarak ise Şekil 3.3’te harita üzerinde gösterilmektedir.

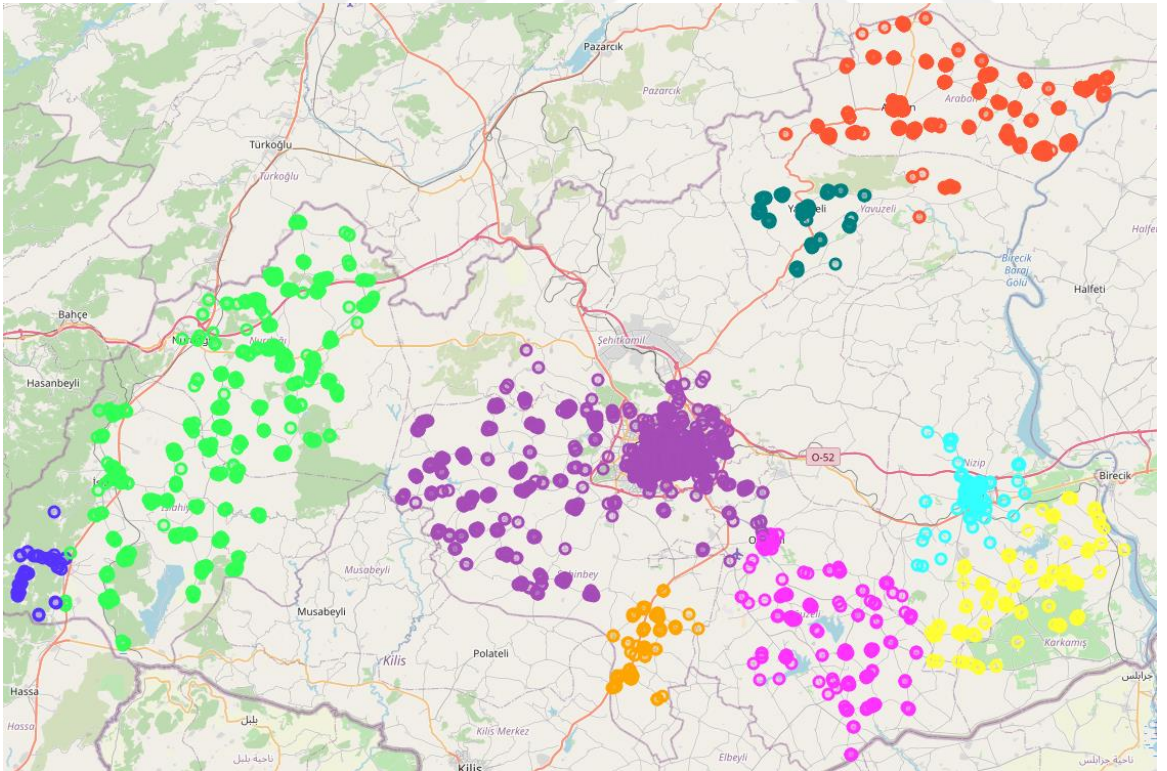


Şekil 3.2. DBSCAN algoritmasının veri seti-1’e uygulanması sonucu elde edilen kümelerin gösterimi

Karasek vektörüne ait veri setinde yer alan dezenfeksiyon işlemi gerektiren rezervuar yerlerinin gürültüden arındırılmış noktalara ait gerçek koordinatlar kullanılarak işaretleme şekil 3.4’te harita üzerinde gösterilmektedir. DBSCAN algoritmasının veri setine uygulanması sonucu elde edilen 9 kümenin merkez noktaları da Şekil 3.5’te gösterilmektedir. DBSCAN algoritmasının veri setine uygulanması sonucu elde edilen ve gürültü olarak sınıflandırılan rezervuar alanları ile beraber oluşturulan kümelerin küme merkezlerinin bir arada gösterimi Şekil 3.6’da gösterilmektedir.

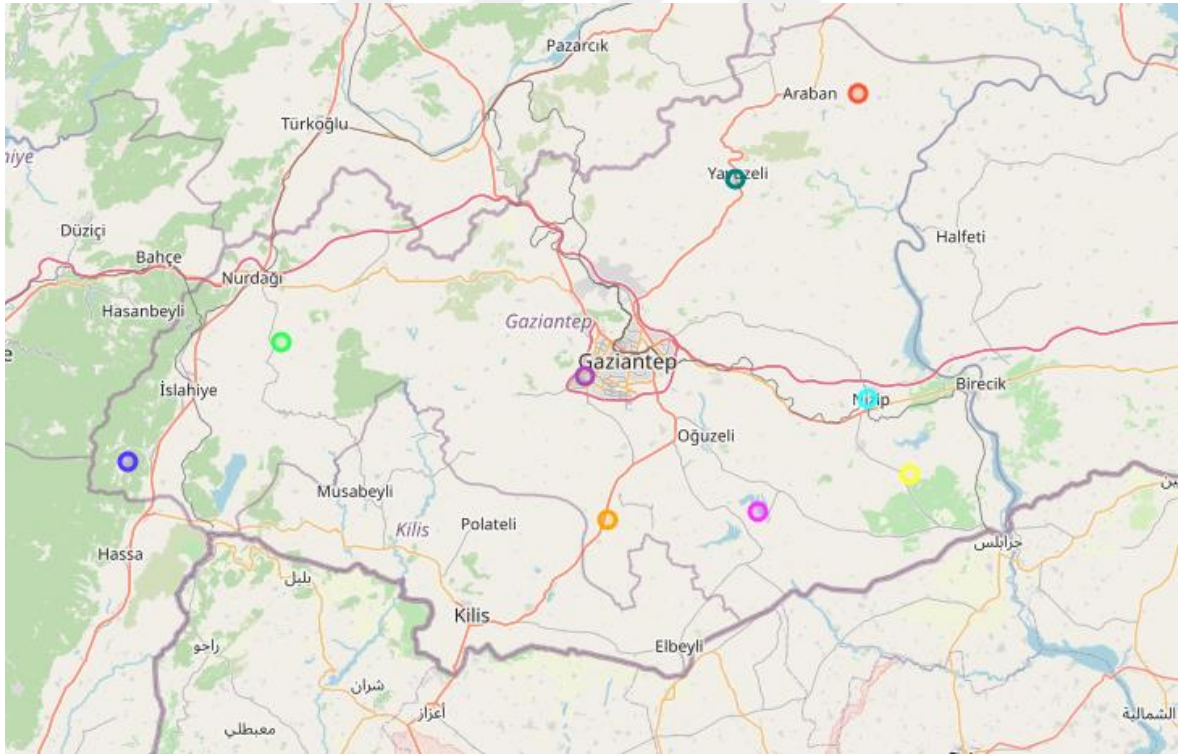


Şekil 3.3. DBSCAN sonucu herhangi bir kümeye atanamayan rezervuar alanların haritada gösterimi

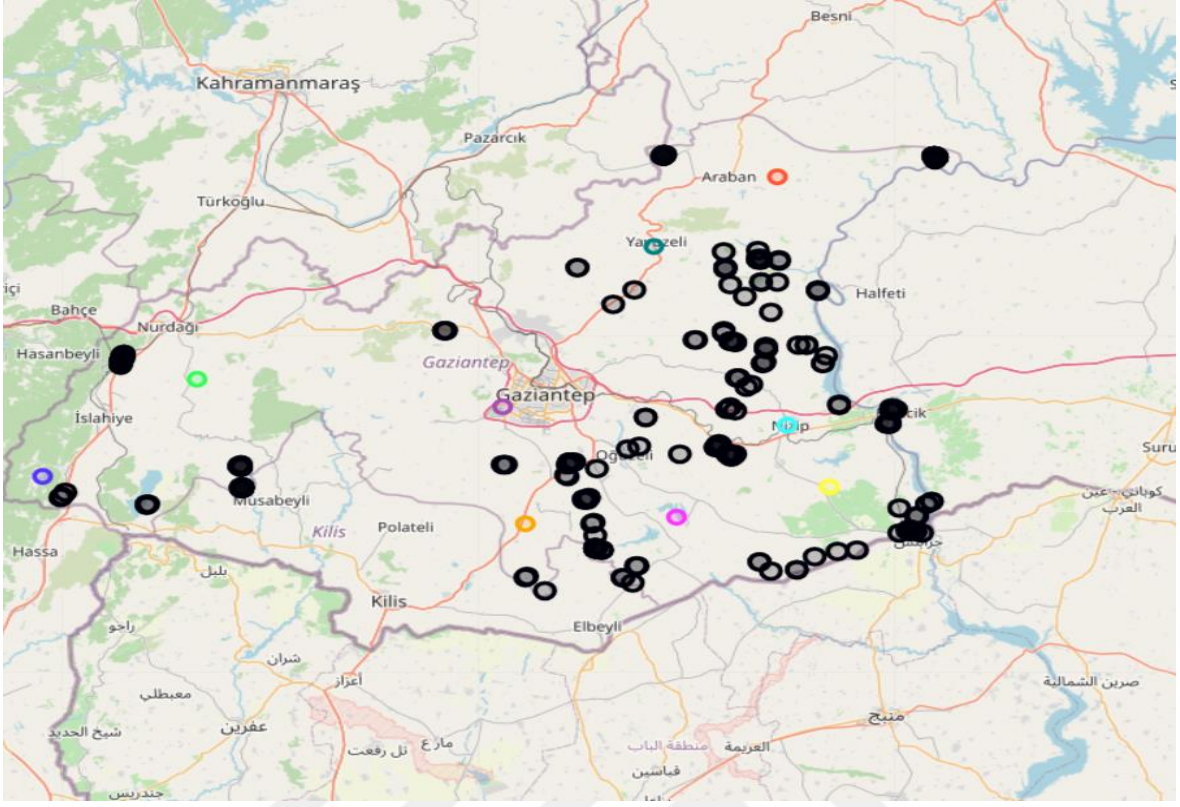


Şekil 3.4. DBSCAN ile kümeleme işleminden geçirilmiş gürültü içermeyen veri seti-1 gösterimi

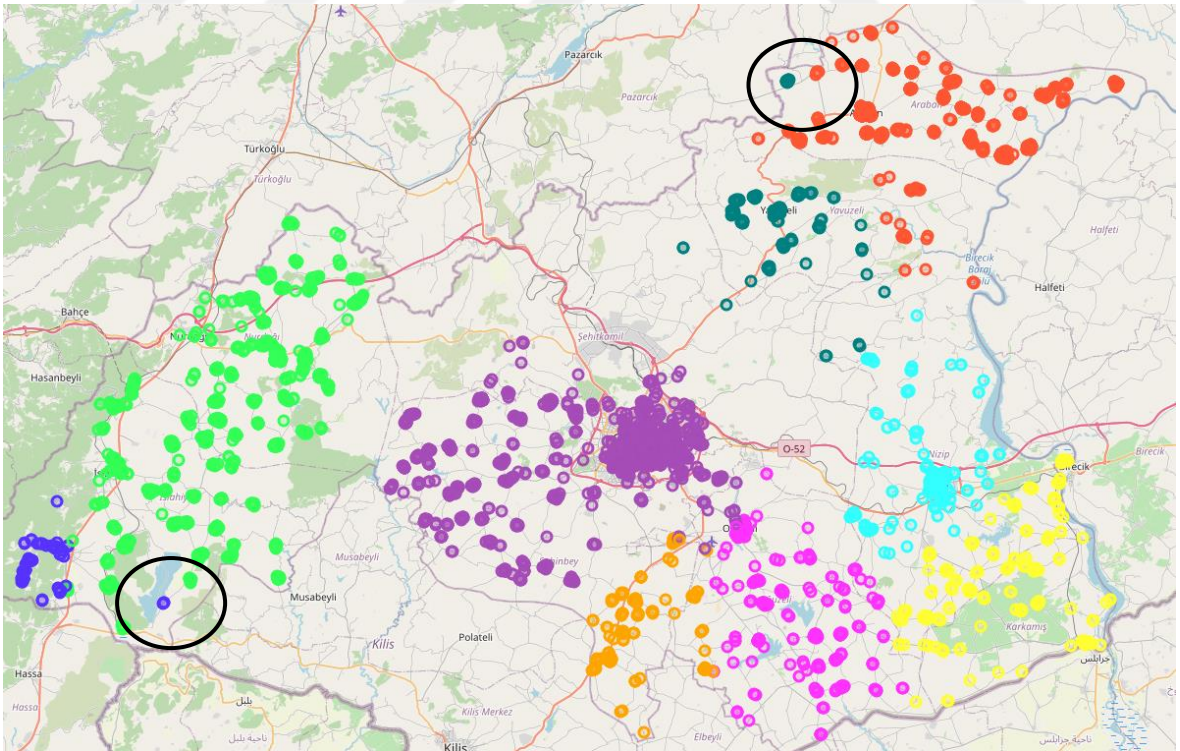
Gürültü olarak adlandırılan rezervuar alanlarının kümelere ait merkezi noktaları göz önünde bulundurularak en yakın kümelere eklenmesi sonucu elde edilen güncel kümeler ise Şekil 3.7’de harita üzerinde gösterilmektedir. Gürültü olarak sınıflandırılan rezervuar alanların kümelerin merkezi noktaları ile arasındaki mesafeye göre atanması sonucu elde edilen güncellenmiş dağılımın gösterildiği Şekil 3.7’de küme 1 (mavi renk) ve küme 2’ye (yeşil renk) atanan ancak uygun bir atama olarak değerlendirilemeyecek noktaların da (siyah çember içerisinde alınmıştır) olduğu görülmektedir. Gürültü olarak sınıflandırılan rezervuar alanlar gürültü olarak sınıflandırılmayan en yakın rezervuar alanın bulunduğu kümeye atanarak gerçekleştirilen küme güncelleme yaklaşımı bu durumu ortadan kaldıracabilmektedir. Gürültü olarak adlandırılan rezervuar alanlarının en yakın komşu göz önünde bulundurularak en yakın komşunun kümesine eklenmesi sonucu elde edilen güncellenmiş kümeler Şekil 3.8’de harita üzerinde gösterilmektedir.



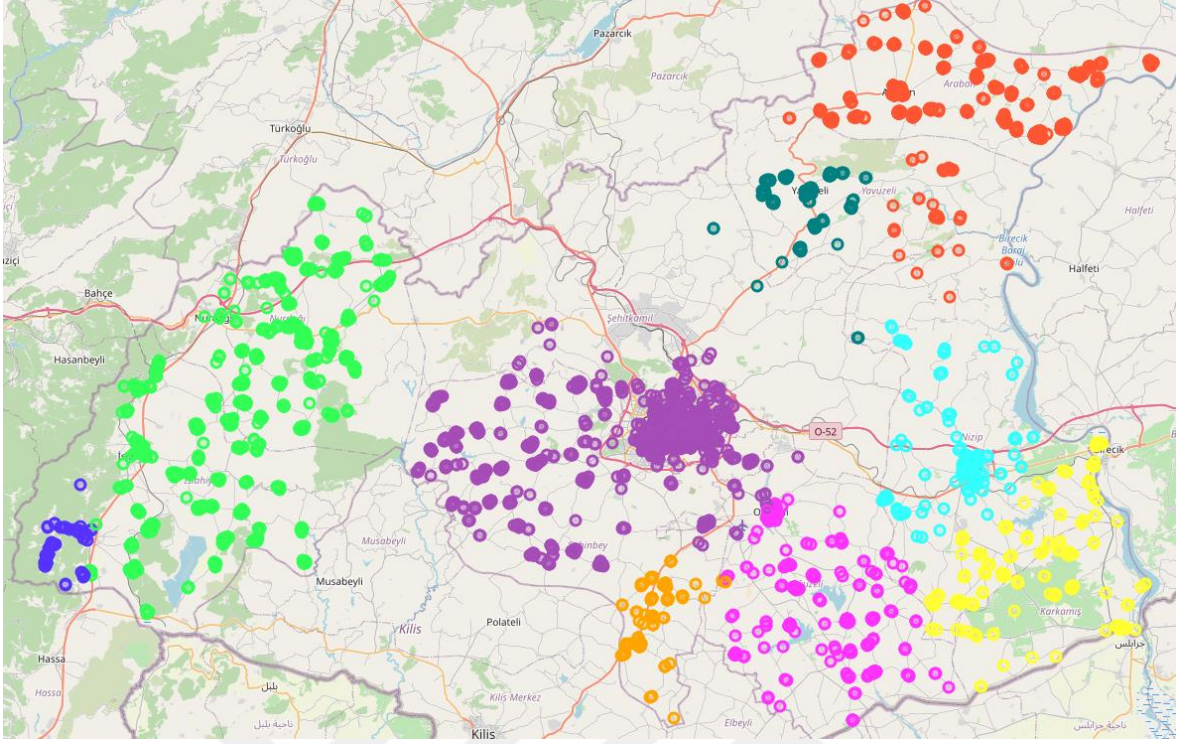
Şekil 3.5. DBSCAN ile elde edilen kümelerin merkezi noktalarının haritada gösterimi



Şekil 3.6. DBSCAN ile kümeleme sonucu gürültü olarak sınıflandırılan rezervuar alanı ve küme merkezlerinin bir arada gösterimi

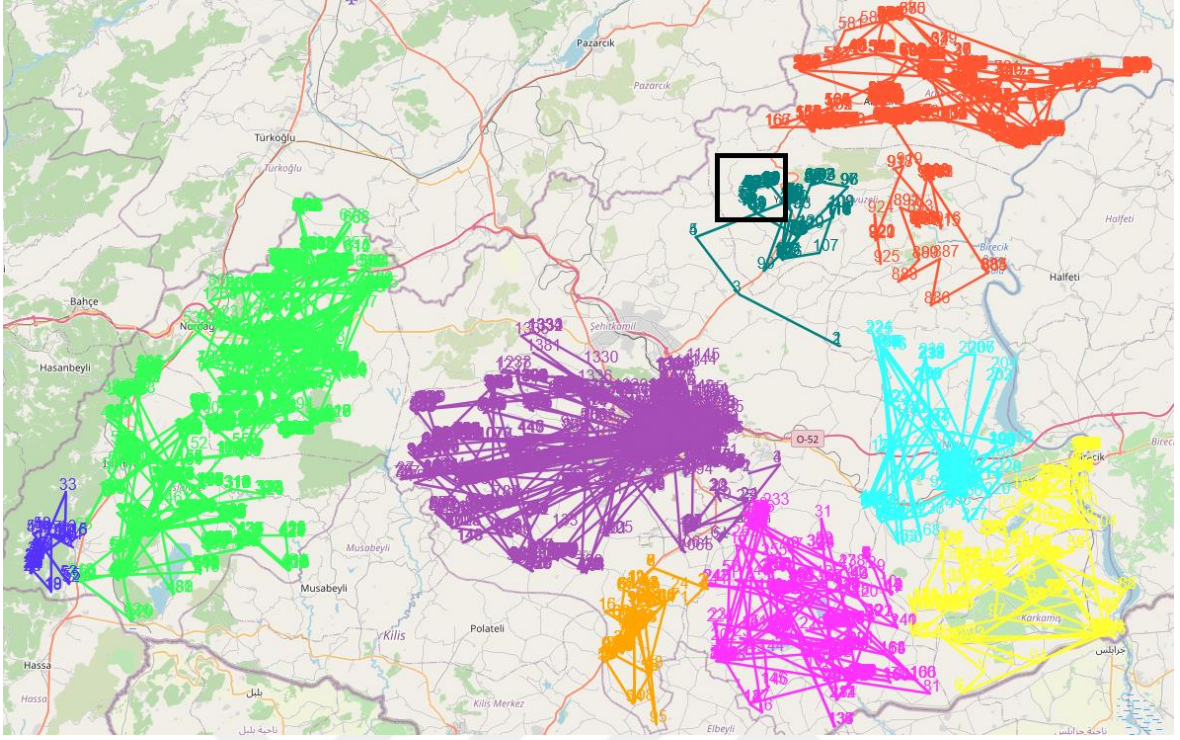


Şekil 3.7. DBSCAN ile kümeleme sonucu gürültü olarak belirlenen noktaların en yakın küme merkezlerine göre ataması yapılması ile elde edilen kümeleme sonucu



Şekil 3.8. DBSCAN ile kümeleme sonucu gürültü olarak belirlenen noktaların en yakın komşunun kümesine ataması yapılmış ile elde edilen kümeleme sonucu

Karasinek vektörüne ait veri setinde bulunan pestisit kullanılarak dezenfekte edilmesi gereken rezervuar alanlara ait gerçek koordinatlar gürültüden arındırılmış ve önce kümeleme işleminden geçirilerek elde edilen kümelerin sonra ayrı ayrı rotalama işleminden geçirilmesi sonucu elde edilen çıktı Şekil 3.9’da harita üzerinde gösterilmektedir. İlk bakışta şekildeki rotalar karmaşık görülebilmektedir. Ama aslında bu şekilde olmadığı rotalama çıktısına yakından bakıldığında anlaşılabilmektedir. Karasinek vektörüne ait veri setinin önce kümelenmesi sonra rotalanması işleminden elde edilen sonuç haritasının yakınlştırılarak bir parçası Şekil 3.10’da gösterilmiştir.

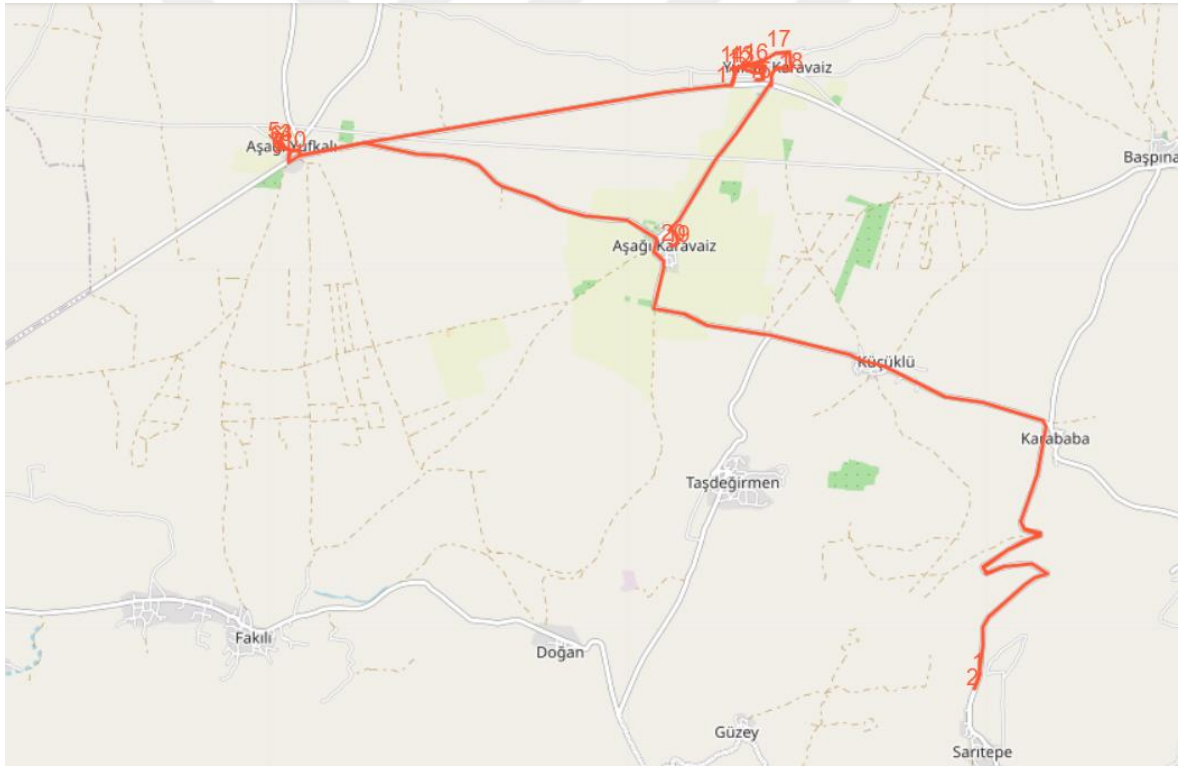


Şekil 3.9. Herbir kümenin rotalama işlemi sonrası belirlenen rotalar



Şekil 3.10. Şekil 3.9’da siyah kare ile işaretlenmiş rota alanının yakın gösterimi

Rezervuar yerleri arasındaki mesafe hesaplanırken öklit veya haversine gibi mesafe ölçümleri kullanılabilir. Ancak noktalar arasında dere, dağ vb. karayolunu uzatan durumlar olması halinde bu mesafeler yanıltıcı olabilmektedirler. Çünkü iki nokta arasındaki mesafe (kuş bakışı mesafesi) belirtilen ölçümlere göre kısa olmasına rağmen, bu noktalar arasındaki gerçek araç seyahat mesafesi çok daha uzun olabilir. Belirtildiği üzere, bu yöntemlerin hesaplanma maliyeti daha düşük olmakla beraber dezavantajları da bulunmaktadır. Bu dezavantajı ortadan kaldırmak için hesaplamalarda ve görsel rotalarda gerçek araç seyahat mesafesinin kullanılması daha faydalı olacaktır. Şekil 3-11’de gerçek araç seyahat mesafelerine göre yapılan hesaplamalarla oluşturulan rotanın bir kesiti gösterilmektedir. Burada, rotanın harita üzerinden kolay anlaşılabilmesi için tüm rota detayları şekle eklenmemiştir.



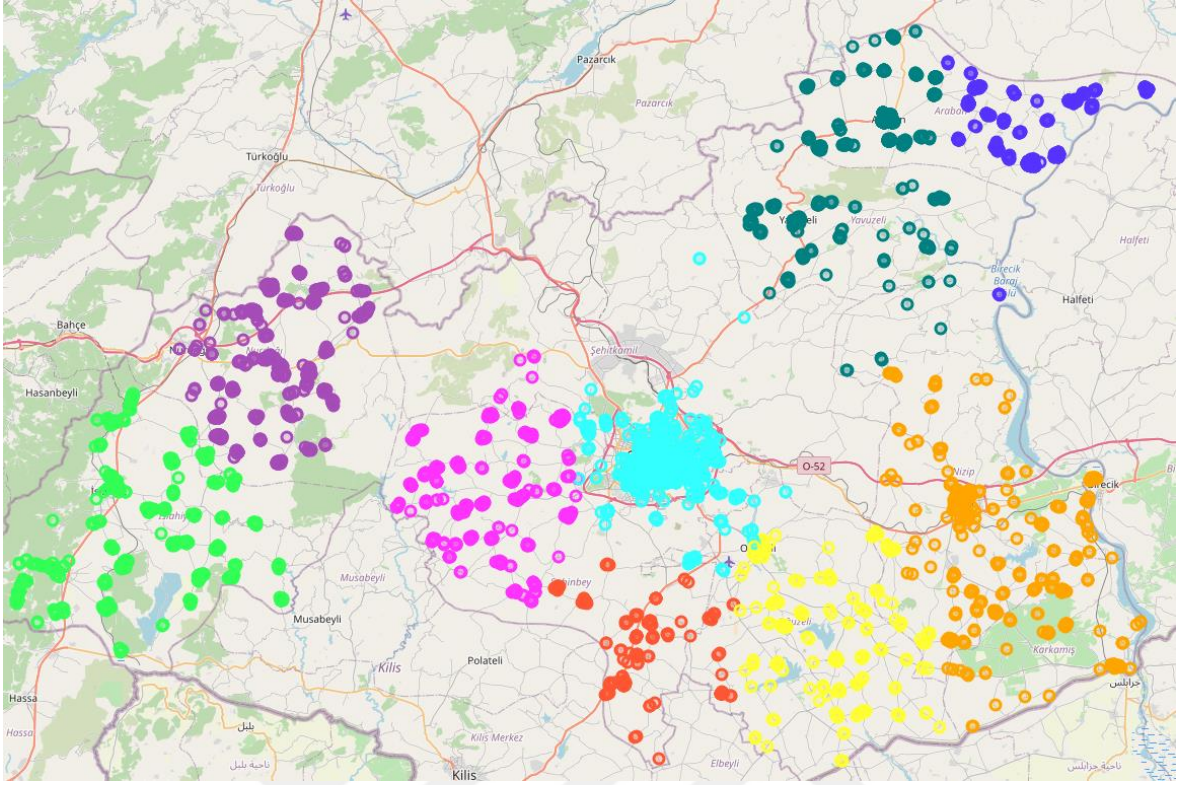
Şekil 3.11. Rotalama işlemi sonucunda elde edilen bazı güzergahların gerçek yol gösterimi

Küme sonucu elde edilen her bir kümeye ait seyahat mesafesi ve toplam seyahat mesafesi bilgileri Tablo 3.1’de sunulmuştur.

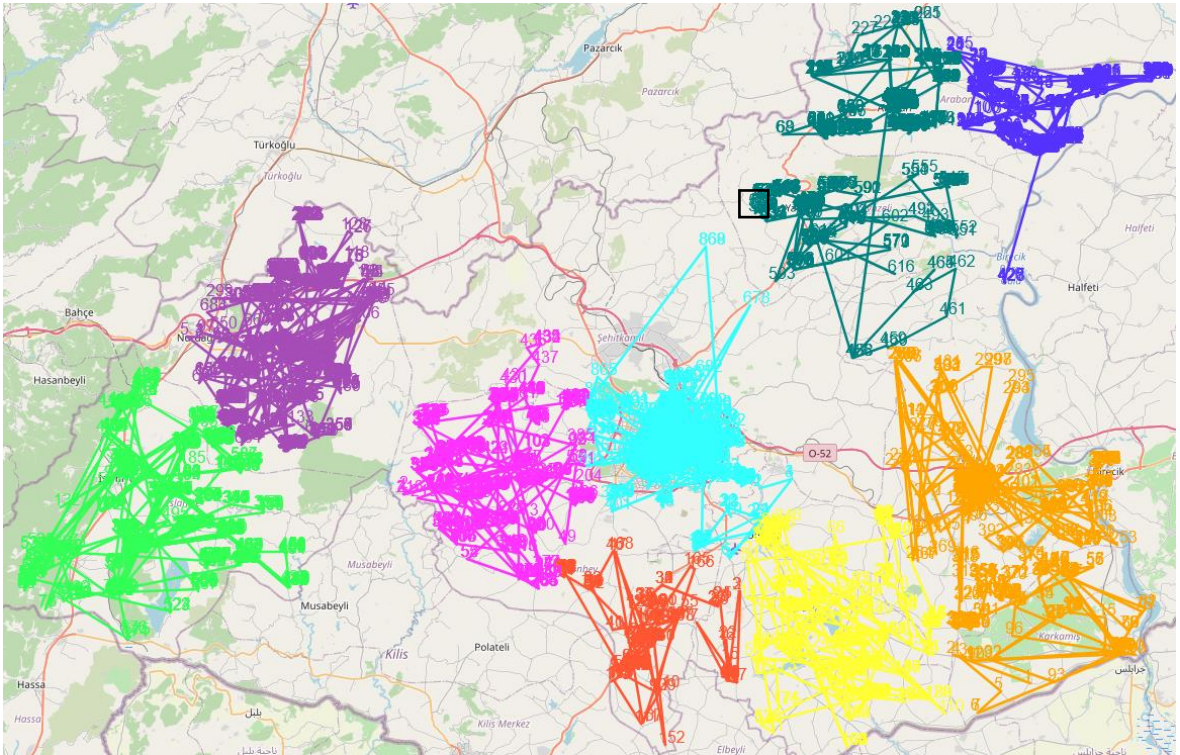
Tablo 3.1. DBSCAN algoritması uygulanması sonucu elde edilen kümelere ait bilgiler

Küme	Rezervuar Sayısı	Mesafe
1	106	139937
2	258	309235
3	29	24680
4	88	148446
5	65	79929
6	93	160311
7	371	299862
8	33	45891
9	34	45323
	Toplam	1208291

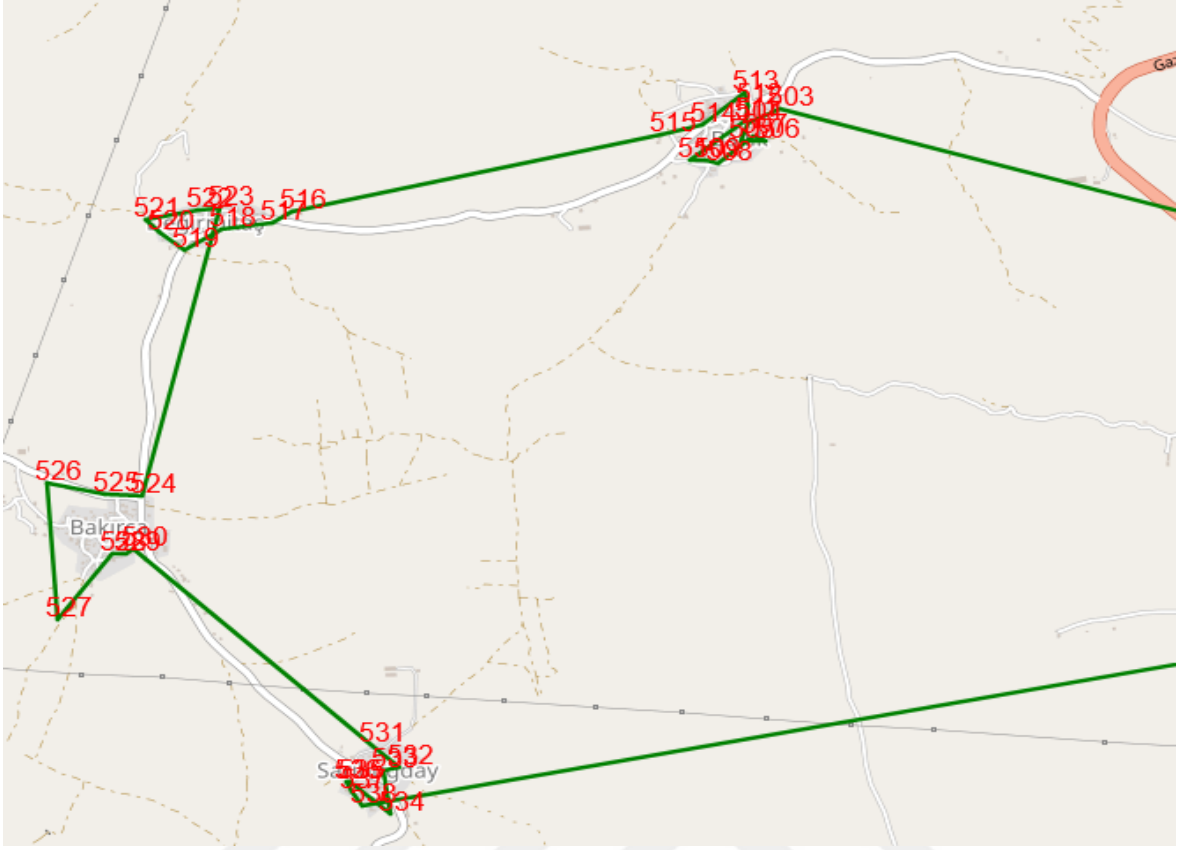
Farklı kümeleme yöntemlerinin başarıya etkisini değerlendirebilmek için veri seti-1 K-Ortalama kümeleme algoritmasıyla da kümelendi. K-Ortalama algoritmasında DBSCAN algoritmasında olduğu gibi gürültü olarak sınıflandırılan noktalar bulunmamaktadır. Bu sebep ile DBSCAN algoritmasının ürettiği gürültü kümesi için ayrıca uygulanan adımların burada K-Ortalama algoritması için uygulanması gerekmemektedir. Veri seti-1 için K-Ortalama ile oluşturulan modelin silhoutte değeri 0.5406, Calinski-Harabasz değeri 15570.7305 ve Davies-Bouldin değeri 0.6618 olarak ölçülmüştür. Deney sonucu 9 kümeye ayrılmış rezervuar alan dağılımları Şekil 3.12’te, her bir kümeye ait optimum olarak belirlenen güzergah Şekil 3.13’de, ortalama işleminin uygulanması sonucu üretilen güzergah haritasının bir kesiti yakınlaştırılmış hali Şekil 3.14’te, gerçek araç seyahat mesafelerine göre yapılan hesaplamalarla oluşturulan rotanın bir kesiti de Şekil 3.15’de gösterilmektedir.



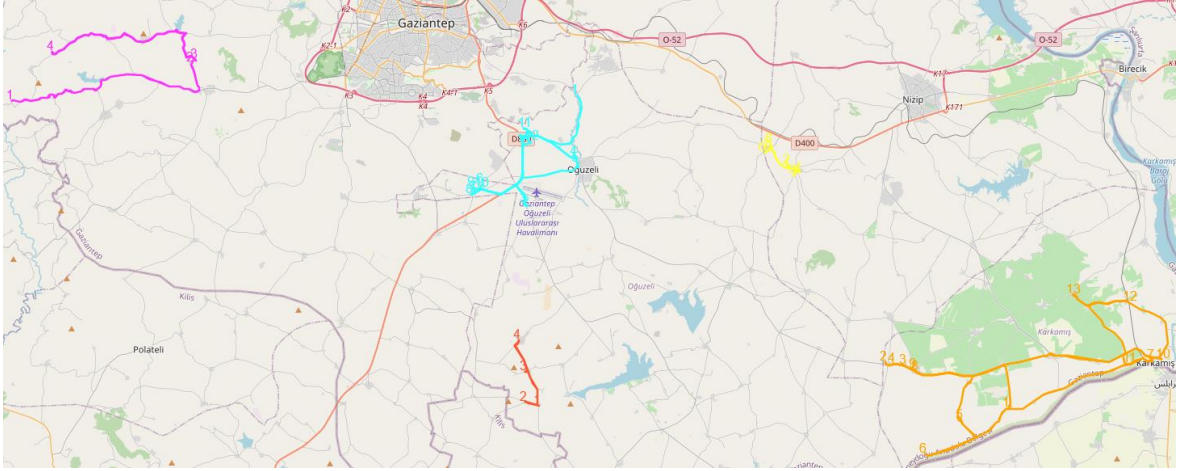
Şekil 3.12. K-Ortalama algoritması kullanılarak kümeleme işleminden geçirilmiş veri seti-1 gösterimi



Şekil 3.13. K-Ortalama ile elde edilen kümelerin rotalama işlemi sonrası haritada gösterimi



Şekil 3.14. Şekil 3.13’de gösterilen haritada siyah kare ile işaretlenmiş alanın yakın gösterimi



Şekil 3.15. Rotalama işlemi sonucunda elde edilen bazı güzergahların gerçek yol gösterimi

K-Ortalama algoritmasının kümeleme sonucu elde edilen her bir kümeye ait seyahat mesafesi ve toplam seyahat mesafesi bilgileri ise Tablo 3.2’de verilmiştir.

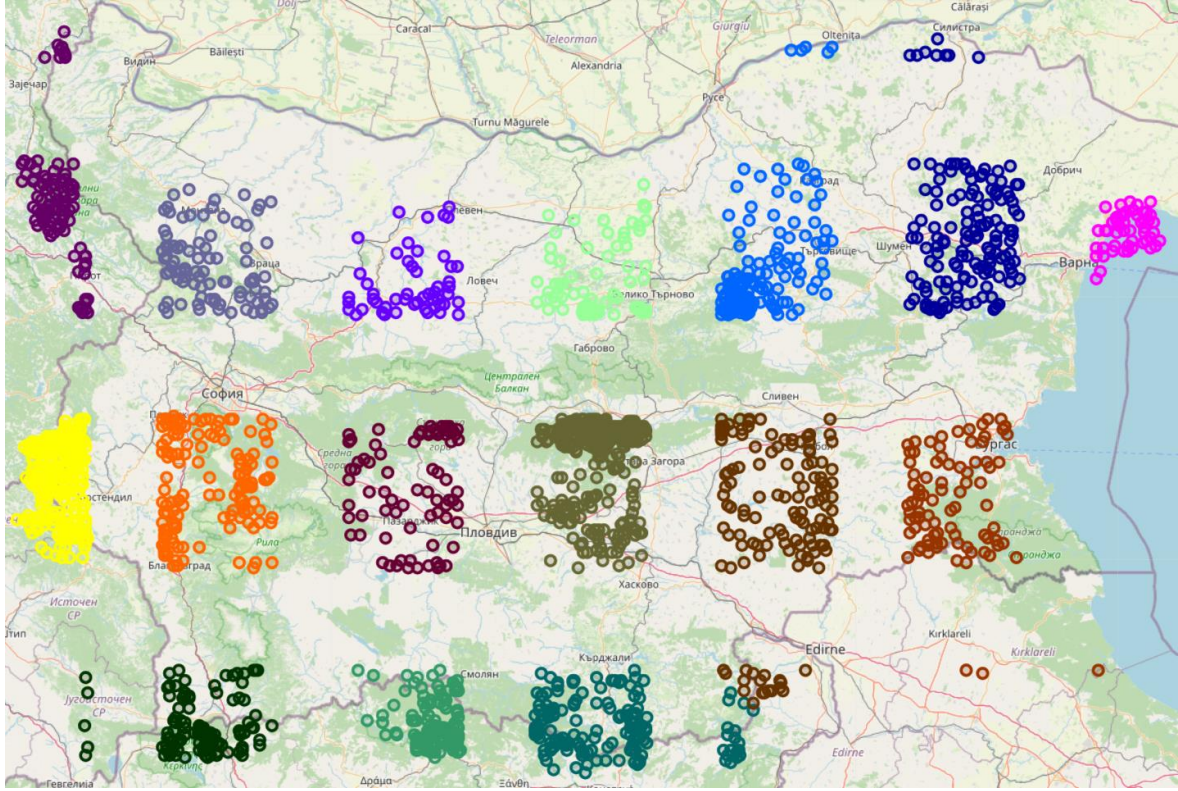
Tablo 3.2. K-Ortalama algoritmasının uygulanması sonucu elde edilen kümelere ait bilgiler

Küme	Rezervuar Sayısı	Mesafe
1	55	99287
2	136	173169
3	58	66482
4	107	180743
5	261	151188
6	96	147049
7	124	149485
8	143	235670
9	82	139575
	Toplam	1342648

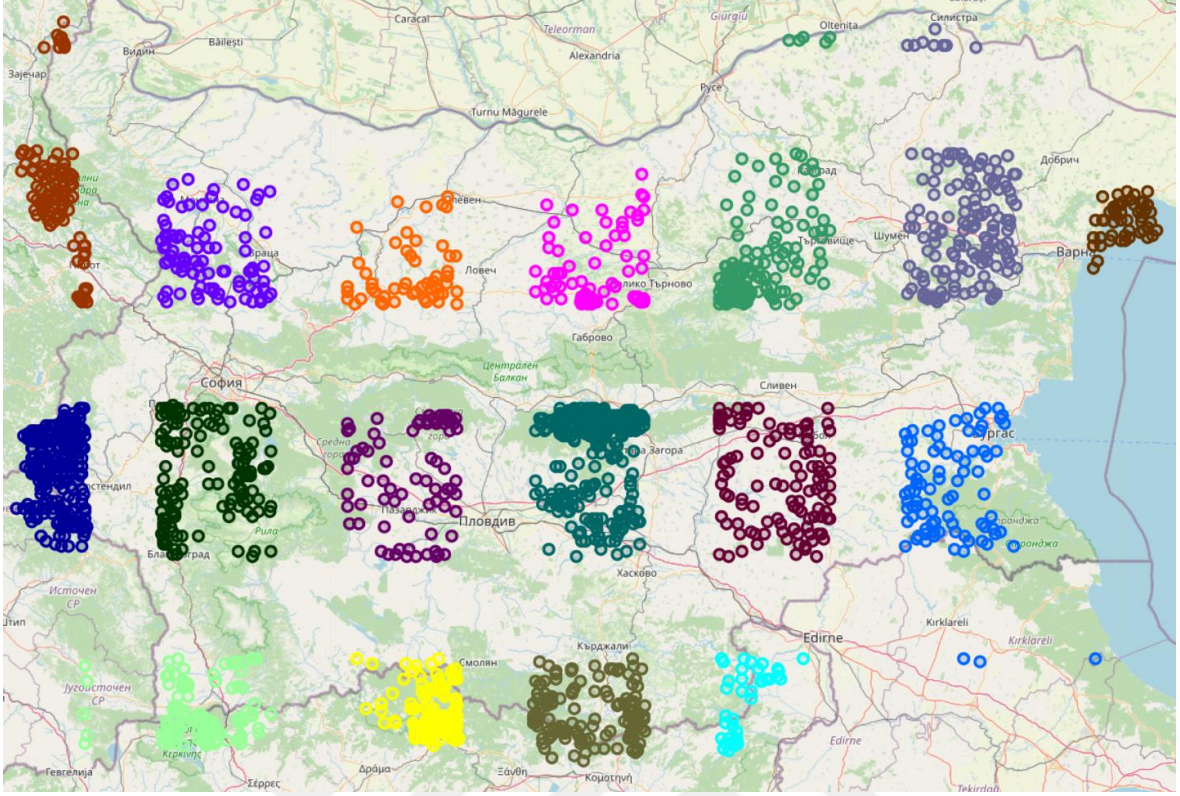
K-Ortalama ve DBSCAN algoritmalarının yanı sıra OPTICS ve BIRCH algoritmaları da denendi. OPTICS algoritması en iyi sonucu epsilon değeri 0.001 minimum nokta 20 değerlerini kullandığımızda üretti ve seyahat mesafesi 1064485 metre olarak hesaplandı. BIRCH algoritmasının en iyi sonucu ise threshold değeri 0.001 küme sayısı 250 değerlerini kullandığımızda seyahat süresininin 10703 metre olduğu görülmüştür. Ancak burada çok fazla kümele oluşmaktadır. Yapılan deneysel çalışmalarda OPTICS ve BIRCH algoritmalarının sonucu, belirlediğimiz problemin çözümü için, K-Ortalama ve DBSCAN algoritmaları kadar başarılı olmadığından, K-Ortalama ve DBSCAN algoritmalarının sonuçları üzerinde değerlendirmeler yapıldı.

Deneysel çalışmalar için ana veri olarak veri seti 1 üzerinde duruldu. Ancak literatürdeki çalışmalarla karşılaştırmalar yapabilmek adına deneysel çalışmalarda iki veri seti daha kullanıldı. Bunlardan biri Bulgaristan ülkesine ait literatürde kullanılan bir veri setidir (Veri seti 2) (URL-6, 2023). Bu veri setinin DBSCAN algoritması kullanılarak kümelere ayrılması sonucu 1954 nokta toplam 16 kümeye ayrılmıştır. İlk aşamada toplamda 66 nokta sınıflandırılmamıştır. Bu noktalar en yakın kümelere dağıtılmıştır. Kümelerin toplam uzunluğu 6239 kilometre 360 metre olarak hesaplanmıştır. Kümeleme sonucu Şekil 3.16’da sunulmuştur. Şekil 3.17’de ise K-Ortalama algoritması kullanılarak veri seti-2’nin 17 farklı kümeye ayrıldığı görülmektedir. K-Ortalama algoritması ile elde edilen kümelere ait toplam uzunluk 6206 kilometre 608 metre olarak hesaplanmıştır. Bulgaristan ülkesine ait veri seti-2’yi oluşturan koordinat noktalarının harita üzerinde kümelere göre gösterildiği Şekil 3.16 ve Şekil 3.17 incelendiğinde koordinat noktalarının doğal olarak kümelenmiş ve

Şekil 3.16’de pembe renk ile gösterilen küme ve Şekil 3.17’de kahverengi olarak gösterilen kümenin deniz üzerinde olduğu görülmektedir. Bu durum ilgili veri setinin oluşturulması aşamasında enlem ve boylam koordinat değerlerinin virgülden sonra sadece iki basamağının alınması nedeniyle olduğu söylenebilir.

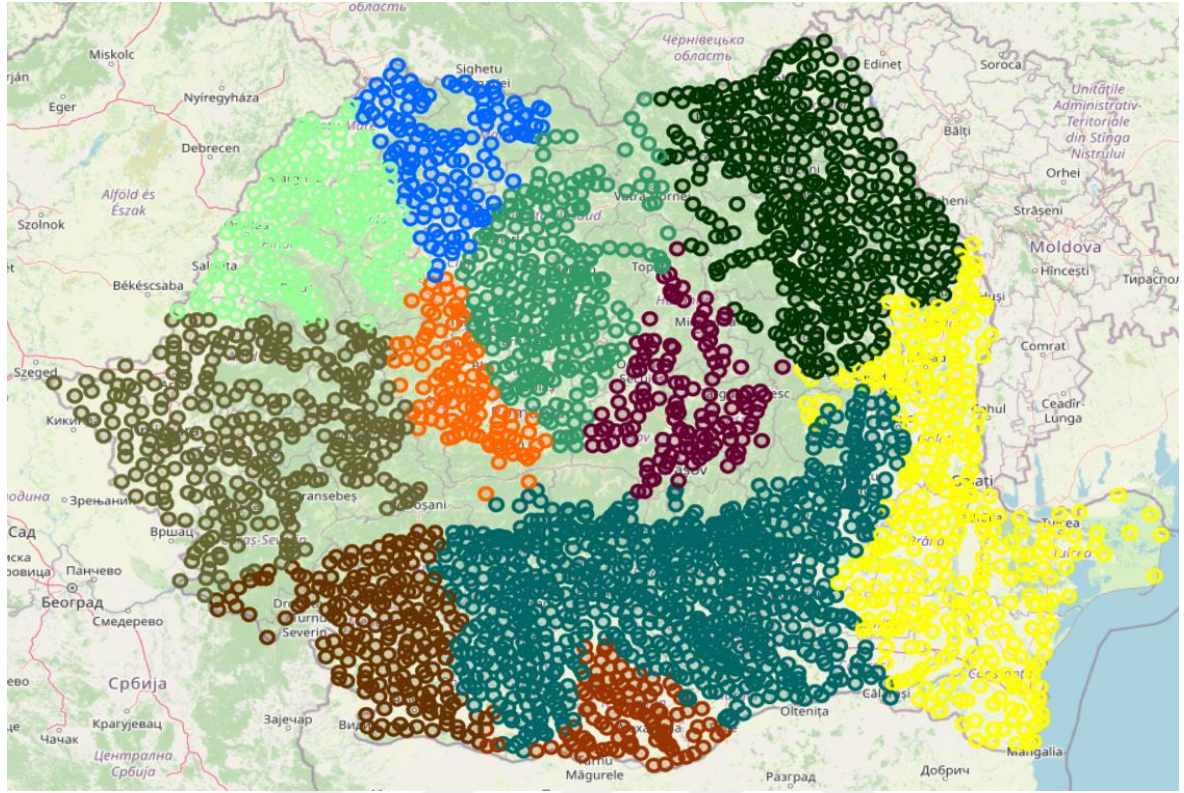


Şekil 3.16. Veri seti-2'ye ait DBSCAN ile oluşturulan küme dağılımı

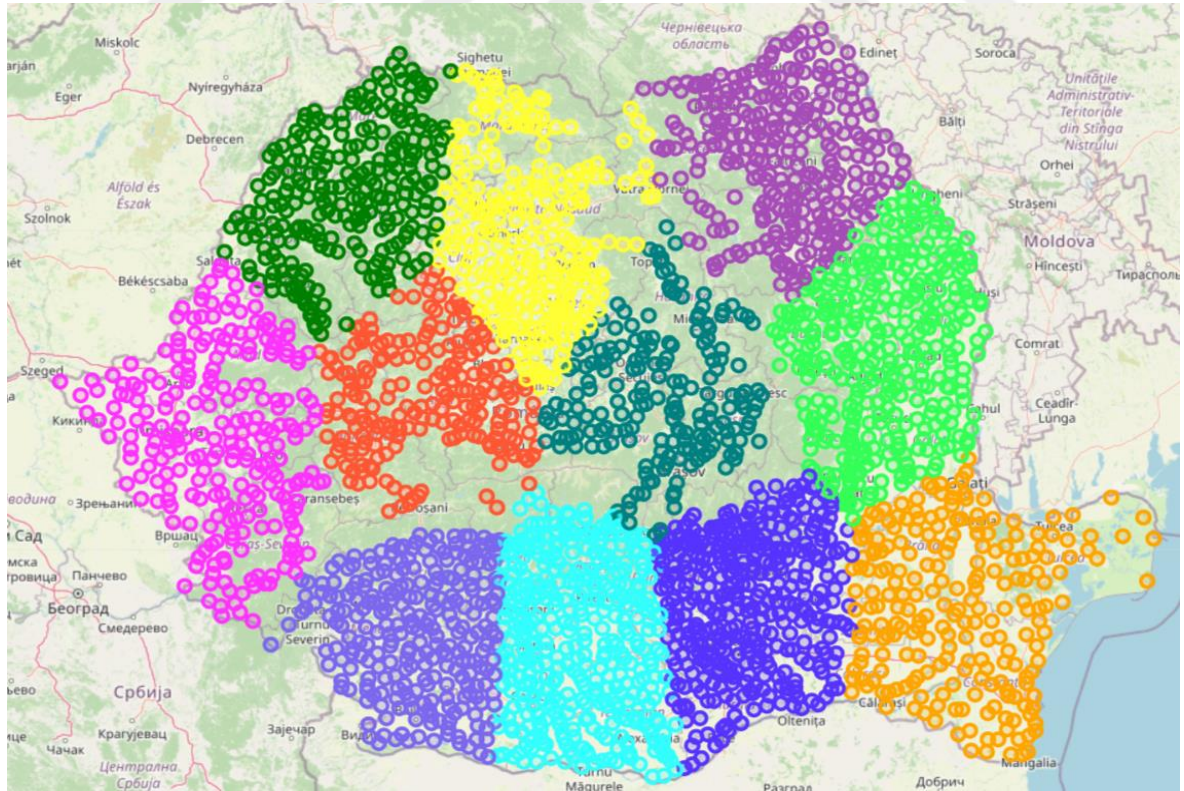


Şekil 3.17. Veri seti-2'ye ait K-Ortalama ile oluşturulan küme dağılımı

Romanya ülkesine ait veri seti (veri seti 3) de literatürde kullanılan bir diğer veri setidir (URL.7, 2023). DBSCAN algoritması (Şekil 3.18) kullanılarak kümelere ayırma sonucu 2950 nokta toplam 11 kümeye ayrılmıştır. İlk aşamada toplamda 1436 nokta sınıflandırılmamıştır. Bu noktalar en yakın kümelere dağıtılmıştır. Kümelerin toplam uzunluğu 13622 kilometre 677 metre olarak hesaplanmıştır. DBSCAN ile yapılan kümelemede oluşan kümelerden birine düşen nokta sayısı çok fazla olduğu için rotalama işlemi bu küme için yapılamamıştır. Rotalama işleminin bu küme için de yapılabilmesi için birbirine yakın olan noktalar birleştirilmiştir. Bu sebepten dolayı veri seti-3 toplam rotalama süresi görece düşük çıkmıştır. Şekil 3.18’de ise K-Ortalama algoritması kullanılarak veri set 3’ün 11 farklı kümeye ayrıldığı görülmektedir. K-Ortalama algoritması ile elde edilen kümelere ait toplam uzunluk 20718 kilometre 263 metre olarak hesaplandı. Romanya ülkesine ait veri seti 3’ü oluşturan koordinat noktalarının harita üzerinde kümelere göre gösterildiği Şekil 3.18 ve Şekil 3.19 incelendiğinde koordinat noktalarının uygun kümelendiği görülebilmektedir.



Şekil 3.18. Veri seti-3'e ait DBSCAN ile oluşturulan küme dağılımı



Şekil 3.19. Veri seti-3'e ait K-Ortalama ile oluşturulan küme dağılımı

4. SONUÇ VE ÖNERİLER

Yapılan çalışmada araç sayısına göre güzergahların belirlenmesi amacı ile önce kümele sonra rotala yaklaşımı kullanılmıştır. Bulgaristan (veri seti-2) ve Romanya (veri seti-3)'ya ait olan gerçek koordinat verilerini içeren veri setlerini Crişan ve arkadaşları bir çalışmalarında rotalama için kullanmışlardır (Crişan ve ark., 2019). Çalışmalarında veri seti-2 için en iyi sonucun 10845 kilometre ve veri seti-3 için en iyi sonucun 21742 kilometre olduğu belirtilmiştir. Veri seti-2'ye K-Ortalama algoritması uygulanması sonucu elde edilen sonuç 6206 kilometre 608 metre ve DBSCAN uygulanması sonucu 6239 kilometre 360 metre olup veri seti-3 için K-Ortalama algoritmasının sonucu 20718 kilometre 263 metre ve DBSCAN algoritmasının sonucu 13622 kilometre 677 metre olduğu görülmüştür. Sonuçlar karşılaştırıldığında, önce kümele sonra rotala yaklaşımının başarılı olduğu görülebilmektedir.

Literatürde bulunan verilerin yanı sıra Gaziantep vektörle mücadele veri seti (veri seti-1) üzerinde yapılan deneysel çalışmalarda da başarılı sonuçlar elde edilmiştir. Burada, OPTICS ve BIRCH algoritmalarından ziyade DBSCAN ve K-Ortalama algoritmalarının daha iyi sonuçlar sunduğu görülmektedir. Deneysel çalışmalar veri seti-1 için K-Ortalama algoritmasının ile kümeleme yapıldıktan sonra rotalama işleminin yapılması daha iyi sonuç verdiğini göstermektedir.

Bu çalışmada incelenen önce kümele sonra rotala yaklaşımlarının verimli sonuçlar ürettiği ve maliyeti düşürdüğü görülmüştür. Günümüzde araç filosu yöneterek hizmet veren birçok kurum ve kuruluş bulunmakta ve araç rotalama problemi ile uğraşmaktadır. Bu problemin çözümü için, özellikle çok sayıda seyahat noktasının bulunma durumlarında, önce kümele sonra rotala yaklaşımı iyi bir tercih olacaktır.

5. KISITLAMALAR VE GELECEK ÇALIŞMALAR

Bu tezin deneysel çalışmalarında herbir küme için bir araç ve bir araç da sadece bir kümedeki noktaları gezecek şekilde hesaplamalar yapılmıştır. Hesaplama kaynakları kısıtlı olduğundan, karşılaştırma amaçlı, kümeleme yapılmadanki dağılım ve çok fazla nokta içeren kümeler için araç rotalama yapılamadı. Bazı çok sayıda nokta içeren kümelerin rotalamalarının yapılabilmesi için yakın noktalar birleştirilerek hesaplamalar yapıldı. Bu durumlar dezavantaj olarak değerlendirilebilir.

Bu tezin konusu için gelecek çalışmalarda, kümeleme esnasında olası rotalama maliyeti göz önünde bulundurularak kümelemeyi de rota maliyetine göre belirleme çalışmaları yapılabilir. Bu durumda küme içi uyumu belirlerken sadece mesafeler değil de olası rota maliyeti de hesaba katılmış olur. Bununla beraber herbir küme için çoklu araç, bir araç için birden fazla küme ve her iki durum için de zaman ve kapasite kısıtlaması durumlarında algoritmaların uygunluğu ve performansları değerlendirilebilir. Kümeleri belirli kısıtlar altında alt kümelere ayırma durumları üzerinde de çalışılabilir. Bu bağlamda yeni yöntemlerin geliştirilmesi çalışmaları da literatüre katkı sunacaktır.

6. KAYNAKLAR

- Alişan, Y., Serin, F., Mete, S., Özceylan, E.,** 2023. Analysing vehicle routing in urban pest control using a cluster-first route-second approach. *2023 International Conference on Information Technology (ICIT)*, 643-646.
- Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J.** 1999. OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49-60.
- Bai, R., Chen, X., Chen, Z.-L., Cui, T., Gong, S., He, W., Jiang, X., Jin, H., Jin, J., Kendall, G., Li, J., Lu, Z., Ren, J., Weng, P., Xue, N., Zhang, H.,** 2023. Analytics and machine learning in vehicle routing research. *International Journal of Production Research*, 61(1):4-30.
- Berkhin, P.,** 2006. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, 25-71.
- Braekers, K., Ramaekers, K., Van Nieuwenhuysse, I.,** 2016. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300-313.
- Bramel, J., Coffman, E.G., Shor, P.W., Simchi-Levi, D.** 1992. Probabilistic analysis of the capacitated vehicle routing problem with unsplit demands. *Operations Research*, 40(6):1095-1106.
- Bramel, J., Simchi-Levi, D.,** 1995. A location based heuristic for general routing problems. *Operations Research*, 43(4):649-660.
- Burkhovetskiy, V.V., Steinberg, B. Y.,** 2017. Parallelizing an exact algorithm for the traveling salesman problem. *Procedia Computer Science*, 119:97-102.
- Chen, P., Huang, H., Dong, X.-Y.,** 2010. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37(2):1620-1627.
- Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F.,** 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512-522.
- Couto, B.,** 2018. Using matrices and hungarian method to solve the traveling salesman problem, 1-13.
- Crişan, G. C., Iantovics, L. B., Nechita, E.,** 2019. Computational Intelligence for Solving Difficult Transportation Problems. *Procedia Computer Science*, 159:172-181.
- Dantzig, G. B., Ramser, J. H.,** 1959. The Truck Dispatching Problem. *Management Science*, 6(1):80-91.

- Duggal, N., Siddiqi, Z.,** 2008. Providing decision making analytical tools to IPM managers through web based: Electronic pest monitoring, and pesticide use reporting system. *6th International Conference on Urban Pests, Budapest, Hungary, 13-16 July 2008*, 275-280.
- Elango, M., Nachiappan, S., Tiwari, M. K.,** 2011. Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms. *Expert Systems with Applications*, 38(6):6486-6491.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X.,** 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *AAAI Press*, 96(34):226-231.
- Fisher, M. L., Jaikumar, R.,** 1981. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109-124.
- Gan, G., Ma, C., Wu, J.,** 2020. Data Clustering: Theory, Algorithms, and Applications, Second Edition. *Society for Industrial and Applied Mathematics*. 41-49.
- Garey, M. R., Johnson, D. S.,** 2009. Computers and intractability: A guide to the theory of NP-completeness (27. print). W.H.Freeman Company, NewYork. 175s.
- Geng, X., Chen, Z., Yang, W., Shi, D., Zhao, K.,** 2011. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, 11(4):3680-3689.
- Hasugian, I. A., Lestari, N., Buchari, B., Dewi, R. A.** 2023. Implementation of heuristic algorithm on capacitated vehicle routing problem (CVRP) for product distribution. *AIP Conference Proceedings*, 2741-2751.
- Hawks, A. H.,** 2023. 12 Capacitated Vehicle. *Intelligent Techniques for Cyber-Physical Systems*, 183s.
- Hiquebran, D. T., Alfa, A. S., Shapiro, J. A., Gittoes, D. H.** 1993. A Revised Simulated Annealing and Cluster-First Route-Second Algorithm Applied to the Vehicle Routing Problem. *Engineering Optimization*, 22(2):77-107.
- Khamesipour, F., Lankarani, K. B., Honarvar, B., Kwent, T. E.,** 2018. A systematic review of human pathogens carried by the housefly (*Musca domestica* L.). *BMC Public Health*, 18(1):1049-1054.
- Koç, Ç., Laporte, G.,** 2018. Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, 91:79-91.
- Laporte, G., Nobert, Y., Taillefer, S.,** 1988. Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems. *Transportation Science*, 22(3):161-172.
- Liao, Y.-F., Yau, D.-H., Chen, C.-L.** 2012. Evolutionary algorithm to traveling salesman problems. *Computers & Mathematics with Applications*, 64(5):788-797.

- López-Castro, L. F., Solano-Charris, E. L., Pagès-Bernaus, A.,** 2023. Environmental approach for the design of raw milk collection routes with a heterogeneous fleet. *Computers and Electronics in Agriculture*, 211:1-17.
- MacQueen, J.,** 1967. Classification and analysis of multivariate observations. *5th Berkeley Symp. Math. Statist. Probability*, 281-297.
- Meng, Y., Lan, Y., Mei, G., Guo, Y., Song, J., Wang, Z.,** 2018. Effect of aerial spray adjuvant applying on the efficiency of small unmanned aerial vehicle for wheat aphids control. *International Journal of Agricultural and Biological Engineering*, 11(5):30-45.
- Menger, K.,** 1932. Das botenproblem. *Ergebnisse eines mathematischen kolloquiums*, 2(4):11-12.
- Na, S., Xumin, L., Yong, G.,** 2010. Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm. *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, 63-67.
- Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., Parthiban, P.,** 2009. Optimization of non-linear multiple traveling salesman problem using k-means clustering, shrink wrap algorithm and meta-heuristics. *International Journal of Nonlinear Science*, 8(4):480-487.
- Naqqash, M. N., Gökçe, A., Bakhsh, A., Salim, M.,** 2016. Insecticide resistance and its molecular basis in urban insect pests. *Parasitology Research*, 115(4):1363-1373.
- Pop, P. C., Cosma, O., Sabo, C., Sitar, C. P.,** 2024. A comprehensive survey on the generalized traveling salesman problem. *European Journal of Operational Research*, 314(3):819-835.
- Sajid, M., Jawed, M. S., Abidin, S., Shahid, M., Ahamad, S., Singh, J.,** 2022. Capacitated Vehicle Routing Problem Using Algebraic Harris Hawks Optimization Algorithm. *Intelligent Techniques for Cyber-Physical Systems*, 183-210.
- Sariman, G.,** 2014. Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma: K-Means ve K-Medoids Kümeleme Algoritmalarının Karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 15(3):192-202.
- Shaheamlung, G., Kaur, H., Kaur, M.,** 2020. A Survey on machine learning techniques for the diagnosis of liver disease. *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 337-341.
- Tan, P.-N., Steinbach, M., Kumar, V.,** 2016. Cluster Analysis: Basic Concepts and Algorithms. *Introduction to data mining*. Pearson Education. Harlow. 719s.

- URL-1**, 2024. https://www.academia.edu/19188331/Revisiting_Cluster_First_Route_Second_for_the_Vehicle_Routing_Problem. Revisiting Cluster First, Route Second for the Vehicle Routing Problem. 06 Ocak 2024.
- URL-2**, 2024. <https://www.tiobe.com/tiobe-index/>. TIOBE Index. | TIOBE - The Software Quality Company. 06 Ocak 2024.
- URL-3**, 2024. <https://pypl.github.io/PYPL.html/>. PYPL Index | PopularitY of Programming Language. 06 Ocak 2024.
- URL-4**, 2024. <https://www.mevzuat.gov.tr/mevzuatmetin/1.5.5393.pdf/>. Mevzuat. 15 Ocak 2024.
- URL-5**, 2023. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-3995.2000.tb00200.x>. Classical and modern heuristics for the vehicle routing problem Laporte 2000 International Transactions in Operational Research Wiley Online Library. 06 Kasım 2023.
- URL-6**, 2023. <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QCHUMD>. Bulgaria TSP instance with GPS coordinates and GEO norm. 06 Kasım 2023.
- URL-7**, 2023. <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/WI0Z8E>. Romania TSP instance with GPS coordinates and GEOM norm. 06 Kasım 2023.
- Valente Klaine, P., Onireti, O., Souza, R. D., Imran, M. A.**, 2019. The role and applications of machine learning in future self organizing cellular networks. *Next-Generation Wireless Networks Meet Advanced Machine Learning Applications*. IGI Global. Hershey, PA. 300s.
- Wang, G., Lan, Y., Yuan, H., Qi, H., Chen, P., Ouyang, F., Han, Y.**, 2019. Comparison of Spray Deposition, Control Efficacy on Wheat Aphids and Working Efficiency in the Wheat Field of the Unmanned Aerial Vehicle with Boom Sprayer and Two Conventional Knapsack Sprayers. *Applied Sciences*, 9(2):120-138.
- Wei, X., Ma, L., Zhang, H., Liu, Y.**, 2021. Multi-core-, multi-thread-based optimization algorithm for large-scale traveling salesman problem. *Alexandria Engineering Journal*, 60(1), 189-197.
- Wu, Y., Cai, Y., Fang, C.**, 2023. Evolutionary multitasking for bidirectional adaptive codec: A case study on vehicle routing problem with time windows. *Applied Soft Computing*, 145:205-222.
- Xu, D., Tian, Y.**, 2015. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165-193.

Yantur, T. G., Uygun, Ö., Erkan, E. F., 2023. A Clustering Approach for the Metaheuristic Solution of Vehicle Routing Problem with Time Window. *Advances in Intelligent Manufacturing and Service System Informatics*, Sakarya, Türkiye, Mayıs 26-27.

Zhang, T., Ramakrishnan, R., Livny, M. 1996. BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25(2):103-114.

