



## Machine learning and mathematical programming based hybrid solution proposal for capacitated vehicle routing problem

Özgür Sanlı\*, Zühal Kartal

Department of Industrial Engineering, Faculty of Engineering, Eskisehir Technical University, 26555, Muttalip, Eskişehir, Türkiye

### Highlights:

- Mathematical Model of CVRP and decomposition methods
- Hybridizing the machine learning algorithms and mathematical modelling formulations
- Supervised and unsupervised machine learning algorithms

### Keywords:

- K-NN algorithm
- Logistics Regression
- K-Means
- CVRP
- TSP

### Article Info:

Research Article  
Received: 23.05.2022  
Accepted: 18.04.2023

### DOI:

10.17341/gazimmfd.1120276

### Correspondence:

Author: Özgür Sanlı  
e-mail:  
ozgur.sanli@gmail.com.tr  
phone: +90 507 788 0565

### Graphical/Tabular Abstract

In this study, capacitated vehicle routing problem (CVRP) is solved by using Machine Learning (ML) guided hierarchical decomposition approach, a classifying (clustering) method is used at the first stage and in the second stage routes are formed by using mathematical programming formulations. First, each node is assigned to the labeled vehicles that is determined by supervised ML algorithms. While using an unsupervised ML approach, a clustering has been made for each vehicle. After classifying (clustering) the nodes, the capacity of the vehicles is checked by capacity balancing algorithm. In case of any capacity exceeding of a vehicle, the vehicle assignment of node changes and the capacity becomes sufficient thanks to the capacity balancing algorithm. At the last stage, the nodes that are assigned to the vehicles, are solved individually by a mathematical formulation of Traveling Salesman Problem (TSP). Once all the TSP routes are combined, the solution for a capacitated vehicle routing problem has been obtained. The stages of the algorithm are summarized in Figure A. To test the suitability of the proposed methods to today's ever-changing conditions, the models under the supervised learning category were run on datasets that were not seen during the training phase.

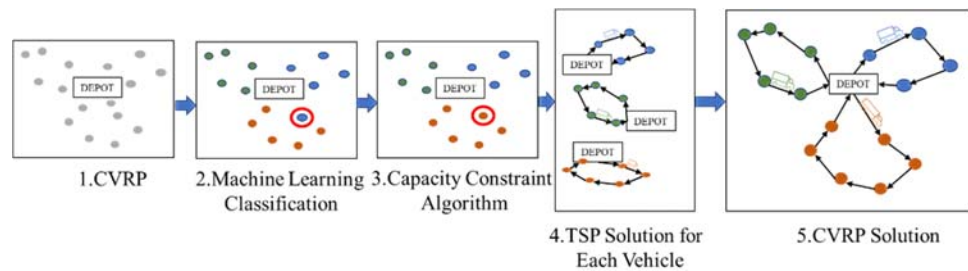


Figure A. Methodology of the ML-Assisted CVRP

**Purpose:** The aim of this study is to solve CVRP by using Machine Learning (ML) guided hierarchical decomposition approach and to show the effectiveness of the proposed approaches.

**Theory and Methods:** In this study, we proposed combining ML Models with mathematical model formulations to solve CVRP. We used K-Nearest Neighborhood and Logistic Regression in supervised learning and K-Means clustering for unsupervised learning category. Both ML Models classifies each node according to the vehicles. After assignment of nodes to the vehicles, a capacity balancing algorithm checks the capacity of each vehicle, if capacity constraint exceeds, the new assignment of nodes is performed according to probability of prediction of each vehicle for supervised ML algorithms. This procedure continues until the capacity is sufficient for each vehicle. And then, a TSP model is solved for each vehicle, and the routes are combined to obtain a whole CVRP solution.

**Results:** This model is tested on 5 different types of datasets from the literature, 2 different supervised and 1 unsupervised ML algorithms. Computational analyses of the proposed methods were conducted on the various instances. In total, 45 experiments were made and proposed model performed better solutions in some instances than CVRP mathematical model solution and a large neighborhood algorithm.

**Conclusion:** The solution of CVRP by combining supervised and unsupervised ML algorithms with mathematical models are comparable to solving the same problems with a well-known metaheuristic from the literature. Also, the hybridized algorithms could be used to solve different datasets that have not been seen during the training phase. The results demonstrate that the proposed models can produce reasonable solutions over a wide range of VRP test instances with different characteristics or sizes.



## Kapasiteli araç rotalama problemi için makine öğrenmesi ve matematiksel programlama temelli hibrid bir çözüm önerisi

Özgür Sanlı\*<sup>ID</sup>, Zühal Kartal<sup>ID</sup>

Eskişehir Teknik Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 26555, Muttalip, Eskişehir, Türkiye

### Ö N E Ç I K A N L A R

- Kapasiteli Araç Rotalama problemi matematiksel modeli ve ayırıştırma metotları
- Makine öğrenmesi algoritmaları ve matematiksel modelleme formülasyonlarının hibridleştirilmesi
- Denetimli ve denetimsiz makine öğrenmesi algoritmaları

#### Makale Bilgileri

Araştırma Makalesi

Geliş: 23.05.2022

Kabul: 18.04.2023

#### DOI:

10.17341/gazimmfd.1120276

#### Anahtar Kelimeler:

Kapasiteli araç rotalama problemi,  
makine öğrenmesi ve matematiksel modelleme hibrid yaklaşım,  
lojistik regresyon,  
K-Means, K-NN

#### ÖZ

Kapasiteli araç rotalama problemi (KARP), günümüzde kargo ve lojistik sektöründe oldukça sık karşılaşılan bir problemdir. Büyük veri çağında yaşadığımız bu günlerde, artan ihtiyaçla birlikte lojistik şirketleri hizmet verilmesi gerekli düğümlerin sayısının fazla ve yerleşimlerinin çok farklı olduğu verilerle karşı karşıya kalmaktadırlar. Dolayısıyla bu durum, mevcut çözüm teknikleri için zorlayıcı olmaktadır. Bu çalışmada, KARP çözümü için, makine öğrenmesi teknikleri ile klasik yöneylem araştırması tekniklerinin birlikte kullanılmasının çözümler üzerindeki başarısı araştırılmıştır. Bu amaçla, makine öğrenmesi teknikleri ile matematiksel programlama formülasyonlarını hibridleştiren iki aşamalı bir yaklaşım önerilmiştir. İlk aşamada makine öğrenmesi algoritmaları ile düğümlerin hangi araçlara atanacağına karar verildikten sonra ortaya çıkan kümelerin toplam talep miktarının her bir aracın kapasitesini aşmaması kapasite dengeleme algoritması adı verilen bir metot tarafından garantilenmiştir. İkinci aşamada ise, her bir araç depodan tur oluşturmak için başlar ve gezgin satıcı problemi (GSP) matematiksel modelini kullanarak en kısa kat edilen mesafeyi bulmak için atanan tüm düğümleri ziyaret eder. Bu çalışmada kullanılan makine öğrenmesi algoritmaları denetimli öğrenme kategorisi altında; K-En yakın Komşuluk algoritması (K-NN) ve lojistik regresyon (LR) algoritmaları; denetimsiz öğrenme kategorisi için, K-Ortalamalar (K-Means) algoritmasıdır. Önerilen yöntemlerin günümüzün sürekli değişen şartlarına uygunluğunu sınamak için denetimli öğrenme kategorisi altındaki modeller, eğitim aşamasında görülmemiş test örnekleri üzerinde çalıştırılmıştır. Önerilen yaklaşım için, farklı araç sayıları ile literatürden farklı özelliklere ve boyutlara sahip veri setleri kullanılarak duyarlılık analizleri gerçekleştirilmiştir. Duyarlılık analizleri ile, bu hibrid yaklaşımın bazı test problemleri üzerinde KARP çözümünden ve Geniş Komşuluk Algoritmasından daha başarılı sonuçlar ürettiği gösterilmiştir.

## Machine learning and mathematical programming based hybrid solution proposal for capacitated vehicle routing problem

### H I G H L I G H T S

- Mathematical Model of CVRP and decomposition methods
- Hybridizing the machine learning algorithms and mathematical modelling formulations
- Supervised and unsupervised machine learning algorithms

#### Article Info

Research Article

Received: 23.05.2022

Accepted: 18.04.2023

#### DOI:

10.17341/gazimmfd.1120276

#### Keywords:

Capacitated vehicle routing problem,  
machine learning and mathematical modelling hibrid approach,  
logistic regression,  
K-Means, K-NN

#### ABSTRACT

Capacity vehicle routing problem (CVRP) is a very common problem in the cargo and logistics industry today. In these days we live in the era of big data, with the increasing need, logistics companies are faced with data that the number of nodes to be served is high and their locations are constantly changing. Therefore, this situation is challenging for existing solution techniques. In this study, the success of using machine learning techniques and classical operations research techniques together on solutions for CVRP solution was investigated. For this purpose, a two-stage approach which hybridizes machine learning techniques and mathematical programming formulations is proposed. In the first stage, it was decided the nodes to be assigned to which vehicles via machine learning algorithms, then it is ensured that the resulting clusters' total demand amount do not exceed the vehicle capacity with a method, which is called capacity balancing algorithm. In the second stage, the vehicle starts from the depot and visits all the assigned nodes to find the shortest travelled distance by using the traveling salesman problem (TSP) mathematical model. The machine learning algorithms that are used in this study are for supervised learning category; K-Nearest Neighborhood (K-NN) and logistic regression (LR) algorithms and for unsupervised learning category; K-Means algorithm. In order to analyze the applicability of the proposed hybrid methods to today's ever-changing conditions, the models under the supervised learning category were run on a dataset that were not seen during the training phase. For the proposed approaches, sensitivity analyzes were performed using datasets with different characteristics and dimensions from the literature, with different number of vehicles. And it has been shown that these hybrid approaches produce better results compared to CVRP GUROBI solutions and Large Neighborhood Algorithm on some test problems.

## 1. Giriş (Introduction)

Araç rotalama problemi (ARP), bir veya birkaç depoda yerleşmiş olarak bulunan araç filosu ile belirli müşterilere yapılan ürün dağıtımını ve/veya müşterilerden ürünlerin toplanması için gerekli olan rotaların belirlenmesi problemi olarak tanımlanmaktadır (Laporte ve Semet [1]). ARP, ilk olarak 1959 yılında Dantzig ve Ramser [2] tarafından ortaya atılan “Kamyon Sevkiyatı” problemiyle literatüre sunulmuştur. Bu problemde bir depodan çıkan benzin tankerlerinin, benzin istasyonlarına en kısa yoldan nasıl dağıtılacağı sorununa çözüm aranmıştır [2]. Çalışmada ayrıca ilk defa problemin matematiksel programlama formülasyonuna yer verilmiştir. Bu öneriden birkaç yıl sonra Clarke ve Wright çalışmalarında [3], literatürde kazanım algoritması olarak bilinen etkili bir aç-gözlü algoritma önermişlerdir. Bu iki çalışmanın ardından, birçok değişik ARP versiyonu ile ilgili yüzlerce model ve algoritma ve bu algoritmalarla ilgili optimal ve yaklaşık çözümlü çok fazla sayıda çalışma yapılmıştır. İlgili okuyucular için, ARP biyografisinin anlatıldığı Laporte ve Osman [4]; Toth ve Vigo [5] çalışmaları önerilebilir. ARP'nin farklı türlerinin incelendiği çalışmalara ise, Ekşioğlu vd. [6] ve Vidal vd. [7] incelenebilir.

ARP, bir veya daha fazla depodan, kapasiteleri bilinen belirli sayıdaki aracın kullanılarak, aynı veya farklı miktarlarda taleplerde bulunan müşterilerin taleplerini karşıladıktan sonra rotalarının maliyetinin en küçüklenmeye çalışıldığı problem tipidir. Klasik ARP'da yaklaşım, araçların depoya geri döndüğünü varsayar. Gezgin Satıcı probleminin (GSP) k sayıda (kapasiteli) araçla çözülmesi sebebiyle ARP, GSP'nin geliştirilmiş bir versiyonudur (Dantzig vd. [8]). GSP'deki n düğüm için hesaplanacak rota sayısı  $n!$  iken; ARP 'de araç sayısı birden fazla olduğu için hesaplanacak rota sayısı  $2^n * n!$ 'dir. Çözüm uzayının çok büyük ve polinom olmaması nedeniyle ARP, NP-zor problem türü içerisinde.

ARP'nin en temel hali Kapasiteli Araç Rotalama Problemidir (KARP). KARP araçların homojen (eşdeğer kapasiteli) olduğunu varsaymaktadır. Müşterilerin talepleri, farklı kapasitelerde bulunan araçlarla karşılanırsa Heterojen Araç Rotalama problemi (Gendreau vd. [9]), araçların belirlenmiş rotalarda gidebilecekleri mesafeler sınırlı ise Mesafe Kısıtlı Araç Rotalama Problemi (Oropeza vd. [10]), araçların müşterilere belli bir zaman diliminde taleplerini karşılayabilecekleri probleme Zaman Pencere Araç Rotalama Problemi (Solomon [11]), birden çok deponun bulunduğu araç rotalama problemlerine ise Çok Depolu Araç Rotalama Problemi (Sujoy vd. [12]) olarak literatürde rastlanabilir.

ARP'de kısıtlar ve amaç fonksiyonu değişikçe çözüm uzayı da şekil değiştirmekte ve büyümektedir, bu yüzden çözüm uzayında bulunan en iyi çözümü veren noktayı bulmak güçleşmektedir. Bu durumda optimum noktayı bulmak için farklı kesin çözüm yöntemleri geliştirilmiştir. Araç rotalama problemleri çözüm yöntemleri incelendiğinde ise, literatürde kesin çözüm yöntemleri, sezgiseller ve meta sezgisellerin sıklıkla kullanıldığı bilinmektedir. Kesin çözüm yöntemleri ile ilgili sınıflandırmalar incelendiğinde, ARP'nin en temel hali KARP için, Toth ve Vigo [13] çalışmasında, dal-sınır, dal-kesme ve küme bölme problemlerine dayalı kesin sezgisellerin yer aldığı görülmektedir. Buna göre KARP için dal-sınır (Christofides vd. [14]) ve küme bölme problemlerine (Agarwal vd. [15]) dayalı olarak türetilmiş kesin çözüm veren sezgiseller öncü olarak sayılabilir.

ARP için geliştirilen klasik sezgiseller ise, kurucu (constructive), iki aşamalı (two-phases) ve iyileştirici (improvement) sezgiseller olarak üç grupta incelenebilir. Kurucu sezgisellere örnek olarak literatürde hala daha tüm ARP tipleri ve GSP için uyarlanmış olan ve oldukça sıklıkla kullanılan Clark ve Wright kazanım algoritması [3] verilebilir.

İki aşamalı sezgiseller; “önce kümele; sonra rotala” ve “önce rotala; sonra kümele” olmak üzere ikiye ayrılır. İyileştirici (improvement) algoritmalar ise, tek bir rota iyileştirme veya birden fazla rotayı iyileştirme olarak ikiye ayrılabilir. Bu iki tip içinde rota içi ve rotalar arası ekleme (insertion), ikili yer değiştirme, 2-opt gibi algoritmalar sıralanabilir. İlgili okuyucular için yerel arama algoritmalarının ARP'ye uyarlanışını inceleyen Braysy ve Gendreau [16] kaynağı önerilebilir.

ARP için yıllar içinde çok fazla sayıda meta sezgisel çözüm yöntemleri geliştirildiği görülmektedir. Öyle ki ARP'nin çözüm algoritmaları sınıflandırıldığında %71.9'unun metasezgisel algoritmaların kullanımı yönünde olduğu görülmüştür (Ekşioğlu vd. [6]). Bu algoritmalar arasında özellikle 1990'lı yılların başlarında Tavlama Benzetimi Osman [17]; Yasaklı Arama Algoritması (Glover ve Laguna [18]; Barbarosoğlu ve Özgür [19]); Karınca Kolonisi Algoritması (Dorigo [20]; Dorigo vd. [21]; Guraksin ve Özcan [22]), Parçacık Sürüsü En iyileme Optimizasyon algoritması (Kennedy ve Eberhart [23]; Marinakis vd. [24]), Geniş Komşuluk Araması (Shaw [25]; Erdoğan [26]), Adaptif Geniş Komşuluk Araması (Pisinger ve Ropke [27]) örnek olarak verilebilir.

Araç rotalama problemleriyle ilgili olarak çok fazla sayıda araştırma yapılmış olmasına rağmen, mevcut ARP araştırmalarının çoğu, genellikle yöneylem araştırmasıyla ilgili araştırmacılar tarafından incelenmiş farklı ARP varyantlarının analitik özelliklerine ve bunlara karşılık gelen çözüm yöntemlerine odaklanır. Bu tür araştırmalarda genellikle temel amaçları ve kısıtlamaları tanımlamak için matematiksel modellerin kullanımı hakimdir (Vidal vd. [7]). Bununla birlikte, makine öğrenimi metodolojilerindeki gelişmelerle birlikte, bu topluluktaki araştırmacılar son zamanlarda yalnızca makine öğrenimi yöntemlerini kullanarak (yani, matematiksel modellerin yapılarını açıkça kullanmadan) kombinatoriyal optimizasyon problemlerini (ARP'ler dahil) çözmeye yönelik girişimlerde bulunmaktadırlar. Bu yöntemler, bazı ilerlemelere rağmen, genellikle farklı senaryolar arasında genelleme eksikliği, veri kullanımında verimsizlik ve içgörülerini keşfedip çözüm yapılarını yorumlamama gibi sorunlarla karşılaşmaktadır. Bu amaçla makine öğrenmesini geleneksel optimizasyona dayalı tekniklerle birleştiren hibrid yöntemlerin kullanıldığı yeni bir araştırma yönüne eğilim başlamıştır (Bai vd. [28]).

Literatür incelendiğinde, ARP'nin yıllar içerisinde çok kapsamlı bir şekilde çalışıldığı ve ARP'yi çözmek için birçok etkili çözüm algoritması geliştirildiği görülebilir. Bu yüzden, bu çalışmadaki öncelikli amaç, araç rotalama problemini çözmenin sınırlarını daha fazla zorlamak yerine, makine öğrenmesi ve yöneylem araştırması tekniklerinin birlikte nasıl kullanılacağını ve yorumlanacağını göstermeye örnek bir çalışma olmaktır. Bu amaçla, gerçek hayat lojistik problemlerinin dinamikliğini yansıtacak şekilde, eğitim ve test için farklı özelliklere ve boyutlara sahip çeşitli veri setleri kullanılmıştır. Böylelikle, bu çalışma bir yandan gerçek hayatta kullanım açısından geliştirilebilir olmasına rağmen, bir yandan da çalışmadaki önerinin hangi tip veri setlerinde iyi performans gösteremeyebileceği de irdelenmiştir. Literatürde GSP için önerilmiş bu tarz bir çalışmaya Sun [29] çalışmasında rastlanılabilir.

Bilindiği üzere makine öğrenmesi, bilgisayarların insanlara benzer şekilde öğrenmesini sağlamak amacıyla çeşitli teknik ve algoritmaların geliştirilmesini sağlayan bilimsel bir çalışma alanıdır [30]. Makine öğrenmesi gözetimli ve gözetimsiz öğrenme olmak üzere ikiye ayrılır. Gözetimli öğrenmede, bilgisayar daha önce doğru sınıflandırılmış bir veri kaynağı ile eğitilir. Bu eğitimden çıkarılan anlamlı sonuçlar daha önce bilinmeyen verilerde kullanılır. Başlıca gözetimli öğrenme algoritmaları Karar Ağaçları, Destek Vektör

Makinelere, Lojistik Regresyon (LR) ve K-En Yakın Komşuluk Algoritması (K-NN) olarak sıralanabilir. Gözetimsiz öğrenmede, bilgisayar ortaya konulan modelde gizli kalıpları kendisi belirlemeye çalışarak anlamlı bir sonuç çıkarmaya çalışır. Gözetimsiz öğrenmede bilgisayar, gözetimli öğrenmeye göre daha çok veriye ihtiyaç duyar. Gözetimsiz öğrenme algoritmalarından başlıcaları ise K-Medoids, K-Means ve DBSCAN olarak sıralanabilir. İstatistik, optimizasyon ve olasılık gibi matematiğin alt unsurlarından oluşan Makine Öğrenmesinin, ARP'nin çözümünde kullanılması ilk olarak 1985 yılında Hopfield ve Tank tarafından ortaya atılmıştır [31]. Dal-kesme metodunu öğretme (Baltean vd. [32]), önemli düğümleri öğrenerek çözme metodu (He vd. [33]), sezgisel metotları kullanarak öğrenme (Khalil vd. [34]) literatürde başlıca bilinen gözetimli öğrenme çalışmalarıdır. Gözetimsiz öğrenme algoritmaları ile ilgili çalışmalara örnek olarak; dal stratejisini öğrenme (Khalil vd. [35]), değişken seçim politikasını öğrenme (Variable Election Policy) (Lederman vd. [36]), kısıt karşılama problemi (Constraint Matching Problem) (Amizadeh vd. [37]) çalışmaları verilebilir.

Literatür incelendiğinde ARP'nin çözümü için, az da olsa makine öğrenmesi algoritmaları yardımıyla birleştirilmiş çözüm yöntemlerine de rastlanmaktadır. Bu çözüm yöntemleri; ARP için geliştirilen sezgisel çözüm yöntemleri kategorisinde incelenebilir. Önce Rotala sonra Kümele (ÖRSK) ve Önce Kümele sonra Rotala (ÖKSR) algoritmaları bu yöntemlere örnektir. ÖRSK algoritmalarında öncelikle bütün müşterileri kapsayan bir GSP çözümü alınır. Bu GSP tarafından oluşturulan çözüm, araçların rotalarına kapasite kısıtlarını aşmayacak şekilde bölünür. ÖKSR' de ise, ÖRSK' de yapılan işlemin tam tersi yapılır, önce araçlara müşteriler atanır, araçların müşterilere en kısa yoldan gidebilmesi için GSP matematiksel modelleri ya da çözümü için geliştirilmiş algoritmalarla baş vurulur. Literatürde, ÖRSK' ye ilişkin başlıca çalışmalar Beasley [38], Montoya [39], ÖKSR' ye; Fisher ve Jaikumar [40], Donda ve Cerdá [41] örnek olarak verilebilir.

Bu çalışmanın literatüre bir diğer katkısı ise, gözetimli öğrenmede sınıflandırma için kullanılan LR ve K-NN gibi algoritmaların kümelendirme amacıyla ARP'de kullanılışı sonucunda performanslarının da incelenmesine olanak sağlanmıştır. Önerilen çalışmada, KARP'nin çözümü için makine öğrenmesi algoritmalarıyla matematiksel modelleme formülasyonlarının çalıştırılmasına yönelik olarak önce kümele, sonra rotalama çözüm yöntemine dayalı hibrid bir algoritma önerisinde bulunulmuştur. Araçlara atanacak düğümlere gözetimli ve gözetimsiz makine öğrenmesi algoritmalarıyla karar verildikten sonra, bir araca atanmış düğümlerin çözümü için GSP matematiksel modelleme formülasyonu ticari çözücüye belli bir süre verilerek çalıştırılmıştır. Makine öğrenmesi ile araçlara atanacak düğümler belirlenirken, LR, K-NN ve K-Means algoritmalarından faydalanılmıştır. Araçlara atanan düğümler belirlendikten sonra kapasite aşımı olup olmadığı kontrol edildikten sonra, eğer böyle bir durumla karşılaşıldıysa kapasite dengeleme algoritması adını verdiğimiz bir algoritma ile araçlara olan atamalar güncellenmektedir. Kapasite dengeleme algoritmasında, LR ve K-NN algoritmaları çalıştırdıktan sonra kapasitesi aşılana ait düğümlerden en düşük olasılıklı düğüm seçilerek, bu düğümün ikinci olarak en yüksek olasılıklı araçta atanması sağlanır. K-Means algoritması çalıştırdıktan sonra ise diğer küme merkezlerine hangi düğüm en yakınsa o düğümün atanması gerçekleştirilir. Eğer seçilen diğer araçta da kapasite uygunluğu yoksa atama üçüncü araçta yapılmaktadır. Bu işlem, araç kapasitesi aşılmayana dek sürdürülür. Ardından elde edilen atamalar, yukarıda belirtildiği gibi ticari bir çözücüye gezgin satıcı problemi formülasyonu çözdürülerek aynı amaç fonksiyonu dahilinde, rotalar elde edilir. Gözetimli öğrenmedeki deneyler, algoritmanın görmediği problemler üzerinde gerçekleştirilmiş, dolayısıyla gerçek hayat rotalamala problemlerindeki dinamiklikte test edilmiş olmaktadır. Bilindiği kadarıyla literatürde, makine

öğrenme algoritmalarını ve matematiksel programlama modellerini hibridleştirip ARP'de uygulayan benzer bir çalışmaya rastlanmamıştır.

Literatür incelendiğinde, çeşitli tipteki araç rotalama problemlerinin çözümü için kümeleme ve ardından rotalamanın gerçekleştirildiği çalışmalara rastlanmıştır. Ele alınan yöntemde en yakın çalışmalar, kümeleme algoritmaları ve matematiksel modellerin birlikte kullanılmasına yönelik olanlar Donda ve Cerdá [41] ve Asis vd. [42] örnek olarak verilebilir. Donda ve Cerdá [41] çok depolu heterojen filolu, zaman pencereli araç rotalama problemi için 3 aşamalı bir sezgisel önerisinde bulunmuştur. Buna göre ilk aşamada uygun kümeler bulunduktan sonra, ikinci aşamada kümeleme tabanlı bir matematiksel modelleme formülasyonu kümelerin araçlara atanması işlemini gerçekleştirir. Son aşamada ise, araç rotaları yine matematiksel modelleme formülasyonu yardımıyla bulunmuştur. Asis vd. [42] çalışmasında ise ham petrol arzının operasyonel yönetimi için matematiksel modelleme tabanlı bir kümeleme algoritması sunulmuştur. Kapasiteli araç rotalama probleminin çözümü için ise K-Means yöntemi ile kümeleme ve ardından rotalama ile çözüm arayan Rautela vd. [43] çalışmasına rastlanmıştır. Rautela vd. [43] K-Means yöntemini araçların kapasitesi aşılmayan bir çözüm bulunana dek çalıştırdıktan sonra, en ucuz link algoritmasıyla araçlara atanan düğümlerin rotalamasını gerçekleştiren iki aşamalı bir algoritma önerisinde bulunmuşlardır. Geetha vd. [44] ise kapasiteli araç rotalama probleminden esinlenerek, bir düğümün bir araca atanması işleminde müşterilerin talebini ve kümeye olan uzaklığını birlikte dikkate alan bir K-Means algoritması sunulmuştur. Alesiani vd. [45] çalışmalarında, K-Means algoritmasıyla indirgenmiş sayıdaki küme merkezi bulduktan sonra, bu merkezlerin rotalanmasını da çözüm için tspy Python hazır program paketini kullanmışlardır. K-Means algoritması ile çözüm için çalışmamıza en çok benzerlik gösteren çalışma ise, Mostafa ve Eltawir [46] tarafından ele alınmıştır. Yazarlar, heterojen filolu araç rotalama problemi için, K-Means yöntemini her bir kümeye benzer sayıda düğüm atanması yapan bir mekanizmanın çalıştırılmasının ardından, rotalama problemini gezgin satıcı problemi matematiksel modelleme formülasyonu ile çözmüşlerdir. Ancak çalışmada, kapasite kısıtının aşılması durumunda nasıl bir yaklaşım izlendiğine yer verilmemiştir. Duan vd. [47] konvülsiyonel sinir ağıları ile düğümler ve linkler için birtakım öznelikler belirleyerek düğümlerin sırasını ve benzer şekilde linklerin çözüm içinde bulunma olasılıklarını tahminleyecek şekilde bir modelle ARP'yi çözmüşlerdir. Ancak, çözüm tekniği olarak her bir düğümün araca atanmasını takip ederek, aracın kapasitesinin aşılmasına izin verilmeyen bir yaklaşım izlemişlerdir. Düğüm sırası ve link sırası için çalışmamıza benzer olasılık tabanlı bir model kurulmasına rağmen, yazarlar, araçların en küçüklenmesini amaç fonksiyonlarında ele aldıklarını belirtmiş ancak araçlar için sabit bir katsayı almayarak, araçların en küçüklenmesini nasıl başardıklarına dair bir açıklamaya yer vermemişlerdir. Bu çalışmada ise, K-Means algoritması ile birlikte, LR ve K-NN algoritmalarının da kapasite aşılması durumunda dengelemesine ilişkin geliştirilmiş olarak kullanılacak gözetimli makine öğrenmesi metotları için olasılık tabanlı algoritmalar ve ticari çözücülerin birlikte kullanıldığı bir yaklaşım sunulmuştur. Dolayısıyla, bu çalışmada, gerçek hayat dinamikliğini yansıtacak şekilde gözetimli makine öğrenmesi tekniklerinin eğitim ve test için ayrı veriler üzerinden çalışmasını sağlayan, gözetimli ve gözetimsiz makine öğrenmesi tekniklerinin ARP'de kullanımını garantiye alacak şekilde kapasite aşımını engelleyecek bir algoritma yardımıyla sınıflandırma ve kümeleme yapması, ardından veri setleri dahilinde rotaların özelliklerini inceleyen bir çalışma literatürde bilindiği kadarıyla henüz yer almamaktadır. Makine öğrenmesi algoritmaları ile ARP'nin çözümüne, ilgili olan okuyucular bir araştırma çalışması Czuba ve Pierzchala [48] inceleyebilirler. Ayrıca ARP'nin çözümü için Nazari vd. [49] gösterici ağı (pointer network) ve Kool vd. [50] ağı

politika sunumu ile tahmin (greedy policy rollout) gibi yöntemler uygulayarak ARP'yi çözmüşlerdir.

Çalışmanın bundan sonraki bölümleri şu şekilde organize edilmiştir. İkinci bölümde kullanılan matematiksel modeller, üçüncü bölümde; makine öğrenmesi algoritmaları sunulmuştur. Dördüncü bölümde, sayısal sonuçlar, literatürde bulunan veri setleri üzerinden yapılan çalışma sonuçları verilmiştir. Beşinci bölümde, sonuç kısmına yer verilmiştir.

## 2. KARP ve GSP MATEMATİKSEL MODELLERİ (CVRP and TSP MATHEMATICAL MODELS)

Bu bölümde, çalışmada kullanılan KARP ve GSP matematiksel modelleri hakkında bilgi verilmiştir. KARP matematiksel modeli, makine öğrenmesiyle sınıflandırılan (kümelendirilen) ve ardından bu düğümlerin GSP matematiksel modeli ile rotaların oluşturulması sonucu oluşan çözümlerin karşılaştırılması ve ayrıca LR ve K-NN algoritmalarının eğitilmesi amacıyla kullanılmıştır.

### 2.1. KARP Matematiksel Modeli (CVRP Mathematical Model)

Kapasiteli (homojen) araç rotalama problemi, aynı kapasiteye sahip homojen araçların, bir depodan başlayarak, tüm düğümlerin araç kapasitelerini aşmayacak şekilde, araçlar tarafından katedilen toplam mesafenin en küçüklenmesi amacını içerir. Araç rotalama problemlerinin araç-akış formülasyonuna dayalı olarak iki indisli ve üç indisli olarak modellenilebildiği bilinmektedir. Bu bölümde ise, KARP'nin çözümü için iki indisli araç-akış formülasyonuna dayalı karma tamsayılı matematiksel programlama modeli kullanılmıştır [51]. Alt-tur-engelleme kısıtları için Miller-Tucker-Zemlin tarafından önerilmiş kısıtlar kullanılmıştır [52].

### 2.2. GSP Matematiksel Modeli (TSP Mathematical Model)

Gezgin satıcı problemi, bir depodan başlayarak tüm düğümlerin ziyaret edilmesi sırasında toplam katedilen mesafenin en küçüklenmesini amaçlamaktadır. Bu çalışmada, makine öğrenmesi algoritmalarının kapasite aşımı olmadan araçlara göre sınıflandırma (kümeleme) yapmasının ardından, her bir araç içinde bulunan düğümler GSP modeli ile çözdürülmüştür. Bu çalışmada GSP'nin iki indisli matematiksel programlama modeli kullanılmıştır (Dantzig vd. [8]; Applegate [53]).

## 3. Makine Öğrenmesi Algoritmaları (Machine Learning Algorithms)

Makine öğrenmesi algoritmaları, temel olarak iki çeşit çıktı verirler. Çıktı sayısal ise regresyon, kategorik ise sınıflandırma problemi olarak adlandırılır. Sınıflandırma problemleri, birçok uygulamada kullanılmaktadır. Bunlar veri madenciliği (Hand [54], Wu [55]), protein etkileşimi (Bock & Gough [56]), kanser teşhisi (Mangasarian vd. [57]), iflas tahmini (Min ve Lee [58]) gibi alanlardır. Makine öğrenmesi algoritmaları genel olarak gözetimli, gözetimsiz ve yarı gözetimli olmak üzere üç kategoride toplanmaktadır. Bu çalışmada ise gözetimli makine öğrenmesi algoritmalarından olan LR ve K-NN algoritmaları çoklu sınıflandırma problemi tahmininde bulunmak için kullanılmıştır. Gözetimsiz bir makine öğrenmesi algoritması olan K-Means yöntemi ile de kümeleme yapılmıştır.

Bu çalışmada gözetimli makine öğrenmesi algoritmaları, ARP'nin çözümünde, araçların hangi müşterilere atanacağı konusunda bir tahminde bulunarak, sınıflandırmayı gerçekleştirmektedir. Kullanılan K-NN ve LR algoritmaları müşterileri araç bazında sınıflayarak, atamaları gerçekleştirmektedir. Gözetimli öğrenme algoritmalarında

sınıflandırma işlemi yapılmadan önce, sınıflandırılması yapılmış bir veri seti holdout yöntemine göre [59] eğitim seti ve test seti olarak ikiye ayrılır. Bu çalışmada ise %70 eğitim - %30 test verisi bölütlemesi kullanılmıştır. Eğitim seti olarak literatürde Solomon [11] (1987) tarafından önerilmiş düğümlerin rassal olarak türetilmiş verilere dayanan R101[11] veri seti kullanılmıştır. Bu veri seti zaman pencereli araç rotalama problemi için türetilmiş olup, 100 adet düğüm içermektedir. R101 veri setinde düğümlerin X ve Y koordinatlarına ek olarak, talep değerleri de verilmiştir. Bu veri seti düğümlerin dağıtım alanına homojen dağıtılmış olması sebebiyle kullanılmıştır. Ancak zaman penceresi değerlerine ilişkin veriler test aşamasında kullanılmamıştır.

Algoritmaların performans karşılaştırmasında kullanılan değerlendirme metriği ise doğruluk (accuracy) testidir. Kullanılan sınıflandırma fonksiyonunun kalitesi bu doğruluk testi ile ölçülmüştür. Bu test, tahmin edilen sınıflandırmanın, bilinen sınıflandırma üzerinden yüzde kaçını doğru tahmin ettiği ile ölçülür.

### 3.1. Lojistik Regresyon Algorithm (Logistic Regression Algorithm)

Lojistik regresyon (LR) bir sınıflandırma algoritmasıdır. Gözetimli sınıflandırma problemlerinde sınıfların ayrık olduğu göz önüne alındığında, algoritmaların amacı sınıflar arasındaki karar sınırlarını bulmaktır. Karar sınırları, bir sınıfın örneklerini diğerinden ayırır. Problem örneğine bağlı olarak, karar sınırları karmaşık veya geometrik olarak doğrusal olmayan bir yerleşim gösterebilir. Genel olarak, farklı makine öğrenmesi algoritmaları, karar sınırlarının şekliyle ilgili farklı varsayımlara sahiptir. LR durumunda, varsayım, karar sınırlarının doğrusal olduğudur. Bu algoritma sonucunda elde edilen karar sınırları, hiper düzlemler şeklinde oluşmaktadır.

LR'nin amacı, özniteliklerde (features) bulunan bilgileri kullanarak belirli bir gözlemin sınıfının doğru bir tahminine sahip olmak için veri noktalarını bölmenin bir yolunu bulmaktır. LR, lineer sınıflandırma denkleminin logaritmik maksimizasyon haline dönmüş şeklindedir. Bu algoritma ile, yapılacak olan sınıflandırma LR modelinde çok kısa sürelerde yapılabilmektedir. Makine öğrenmesi algoritmaları, bir sınıflandırıcıyı belirli bir sınıflandırma algoritması kullanarak bir veri kümesi üzerinde eğitirken, veri noktalarını belirli sınıflara ayıran Karar Sınırı (Decision Boundary) adı verilen bir hiper düzlemler kümesinin tanımlanması gerekir. Problemden kullanılan modelde LR algoritması düğümlerin X ve Y koordinatlarını öznitelik olarak aldığı durumda sınıflandırmayı tahmin etmektedir. LR'nin amacı bilindiği üzere ikili (binary) sınıflandırma yapmaktır. Ancak, bu çalışmada araç sayıları 2 ve 2'den daha çok sayıda ele alınmıştır. Bu amaçla LR'nin çoklu sınıflandırma hali için Softmax Regresyon kullanılmıştır. Softmax regresyon, lojistik regresyonun çoklu sınıflandırma için geliştirilmiş halidir [60]. Yalnızca bir bağımsız değişkenin ve bir bağımlı değişkenin olduğu lojistik regresyon modeli Eş. 1'deki gibidir (Anderson [61]; Demaris [62]):

$$P(y = 1|X = x) = h_{\theta}(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} = \frac{1}{1 + e^{-\beta_0 + \beta_1 x}} = \frac{1}{1 + e^{-\theta^T x}} \quad (1)$$

Buna karşılık softmax regresyonu, lojistik regresyonun birden fazla sınıfı ele almak istenildiği duruma genelleştirilmesidir. Lojistik regresyonda yukarıdaki gibi etiketlerin ikili olduğu varsayılmıştır:  $y^{(i)} \in \{0,1\}$ . Softmax regresyonu ise K 2'den fazla değer aldığıda uygulanır ( $y^{(i)} \in \{1, \dots, K\}$ ). Burada K, sınıfların sayısıdır. Bir test girdisi x göz önüne alındığında, hipotezin, her bir değer için, K'nin farklı olası değerlerinin her birini alan sınıf etiketinin olasılığı aşağıdaki gibi hesaplanacaktır. Böylece, hipotez  $P(y = k|x)$   $k = 1, \dots, K$  olacak şekilde K-boyutlu bir vektör çıkaracaktır. Hipotez bu kez Eş. 2'deki gibi olacaktır:

$$h_{\theta}(x) = \begin{bmatrix} P(y = 1|x; \theta) \\ P(y = 2|x; \theta) \\ \vdots \\ P(y = K|x; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^K \exp(\theta^{(j)T} x)} \begin{bmatrix} \exp(\theta^{(1)T} x) \\ \exp(\theta^{(2)T} x) \\ \vdots \\ \exp(\theta^{(K)T} x) \end{bmatrix} \quad (2)$$

$\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(k)} \in \mathfrak{R}^n$  modelin parametreleri olduğu durumda,  $\frac{1}{\sum_{j=1}^K \exp(\theta^{(j)T} x)}$  terimi dağılımı normalleştirir ve olasılıklar toplamını 1'e eşitler. Şekil 1'de LR ile R101 eğitim veri seti ile elde edilen karar sınır grafikleri sunulmuştur. Bu çizilen sınıra göre ise, araç sınıflandırmaları yapılmıştır.

### 3.2. K-En Yakın Komşuluk Algoritması (K-Nearest Neighborhood)

K-NN, sınıflandırma işleminde bulunulacak örnek veri noktasının bulunduğu sınıfın (öğrenim kümesi) ve en yakın komşunun (elemanın), k değerine (benzerliğe) göre belirlendiği bir gözetimli makine öğrenme yöntemi olarak ifade edilmektedir. Bu algoritma parametrik olmayan gözetimli bir öğrenme yöntemidir, ilk olarak Fix ve Hodges tarafından 1951 yılında ortaya atılmıştır [63]. Sınıflandırma ve regresyon için kullanılır. Her iki durumda da girdi, bir veri setindeki en yakın k eğitim örneğinden oluşur. En yakın komşu sınıflandırıcılar, analogi yoluyla, yani belirli bir test seti ile ona benzer eğitim setlerini karşılaştırarak öğrenmeye dayanmaktadır. Eğitim seti n özniteliği ile tanımlanır ve her set, n boyutlu uzayda bir noktayı temsil etmektedir. Böylelikle, tüm eğitim gözlemleri n-boyutlu bir model uzayında saklanmaktadır [63]. Bilinmeyen bir yeni bir gözlem seti verildiğinde, bir k-en yakın komşu sınıflandırıcısı, bilinmeyen sete en yakın olan k eğitim gözlemleri için desen uzayını genellikle Öklid mesafesi gibi bir uzaklık metriği ile aramaktadır (Eş. 3). Ardından arama sonucu ile elde edilen k adet komşu gözlem noktasına göre yeni veri setinin sınıfını belirlemektedir (Çılgın vd. [64]).

$$\text{dist}(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (3)$$

K-NN algoritmasının çoklu sınıflandırma problemlerine de uygulanabilir olduğu bilinmektedir. K-NN algoritmasında çoklu sınıflara dair olasılıklar hesaplanırken, çoğunluk oylama (majority voting labeling) olarak adlandırılan bir yaklaşım kullanılmaktadır. Bu yaklaşıma göre, N komşuluk sayısı değeri sonucunda, her bir

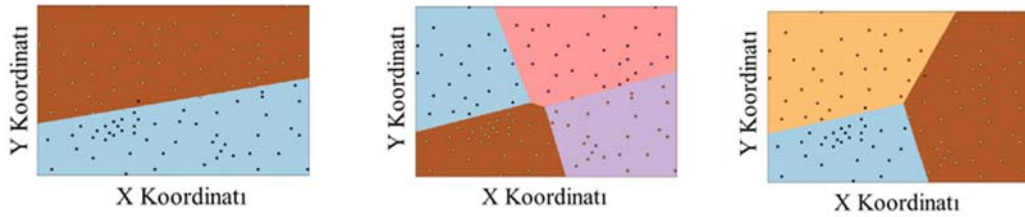
düğümün, belirli bir hedef sınıfa ait olması sayısı toplanarak, bu değer N'e bölünür ve bir olasılık elde edilir. K-NN algoritması, öznitelik hiper düzlemindeki (ve bunların görel mesafe ölçümleri) verilerin dağıtımının yerel geometrisine dayanan bir algoritmadır. Bu nedenle karar sınırı, doğrusal olmayan bir şekilde oluşur. Ele alınan örneklerde LR algoritmasında olduğu gibi eğitim ve test setlerine ayrılan veri, X ve Y koordinatları öznitelik olarak kullanılarak tahminde bulunmuştur. R101 eğitim veri setine ait K-NN algoritması kullanılarak yapılan sınıflandırmadan elde edilen karar sınırları Şekil 2'de verilmiştir.

### 3.3. K-Ortalamalar Algoritması (K-Means Algorithm)

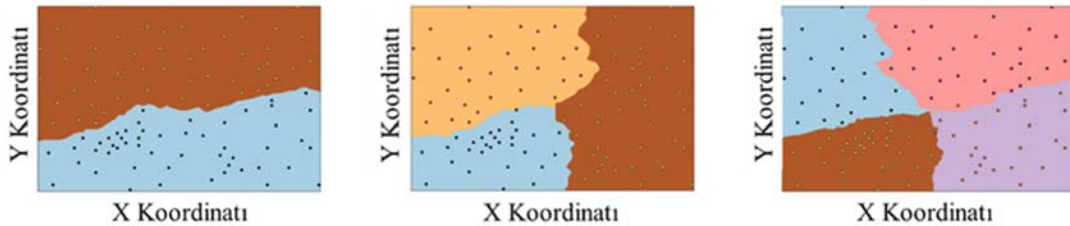
K-Ortalamalar (K-Means) kümeleme algoritması, gözetimsiz makine öğrenmesi algoritmaları içerisinde sınıflandırılır. Yöntem n adet veri nesnesinden oluşan bir veri kümesini, her gözlemin en yakın ortalamaya sahip kümeye (küme merkezleri veya küme merkezi) ait olduğu k kümeye bölmeyi amaçlayan bir yöntemdir. K-Means algoritması, kümeler arası benzerlikleri en aza indirirken, küme içi benzerlikler en büyüklenmeye çalışma mantığına dayalıdır. Algoritmanın temel çalışma prensibi karesel hataları en küçüklemeye dayanır. Algoritma iki adımda çalışır; ilk olarak, veride bulunan her bir düğümü en yakın kümelene bölgesine atar, ardından her bir kümeye ait merkezi verinin ortalamasına göre belirler. K-Means algoritması gözetimsiz bir öğrenme algoritması olduğu için herhangi bir eğitim ve test veri seti olarak ayırma ihtiyacı duymamaktadır. Bu algoritmada da yine, gözetimli makine öğrenme algoritmalarında olduğu gibi veri setinde bulunan X ve Y koordinatları öznitelik olarak alınmış ve bu değerlere göre kümele gerçekleştirilmiştir. R101 eğitim veri setine ait K-Means karar sınır grafiği Şekil 3'te sunulmuştur.

### 3.4. Kapasite Dengeleme Algoritması (Capacity Balancing Algorithm)

Araç rotalama probleminin sınıflandırma aşamasında yukarıda bahsedilen karar sınırı değerlerine göre düğümlerin araçlara ataması yapılmaktadır. Farklı veri setlerinde, araca yapılan atamalar sonucu aracın taşınması gereken toplam talep miktarı, aracın kapasitesinin üzerinde gerçekleşebilmektedir. Araçların kapasitesinin aşıldığı durumlarda, kapasite dengeleme algoritması (KDA) adını verdiğimiz bir algoritma ile araçların kapasitesinin aşılması garantilenmektedir. Denetimli makine öğrenmesi algoritmaları sınıflandırma yapılacaksa bir kararın seçilip seçilmemesi üzerine kurgulanmışlardır. Ancak, sınıflandırmalar çoklu kararları içerecek



Şekil 1. LR Algoritması ile Tahmin Sınıfının 2 Araçlı, 3 Araçlı ve 4 Araçlı Karar Sınırları (Decision Boundaries of LR Algorithm Prediction Classification for 2 Vehicles, 3 Vehicles and 4 Vehicles)



Şekil 2. K-NN algoritması ile Tahmin Sınıfının 2 Araçlı, 3 Araçlı ve 4 Araçlı Karar Sınırları Karar Sınırları (Decision Boundaries of K-NN Algorithm Prediction Classification for 2 Vehicles, 3 Vehicles and 4 Vehicles)

şekilde güncellenebilmektedirler. Bu güncellemeler sayesinde çok sınıflı LR ve K-NN algoritmalarından her bir düğümün her bir araca atanmasına (karara) ait olasılıklar elde edilmiştir. Kapasite dengeleme algoritması, kapasitesi aşılmış bir araca atanmış düğümler arasında, en az olasılıkta bulunan düğümün o araca atanmasını iptal ederek, bu düğümün eğer kapasitesi yeterliyse ikinci sıradaki en yüksek olasılıkta olan araca atanmasını yapılmasını sağlar. LR algoritması için olasılık değerlerinin nasıl hesaplandığı Bölüm 3.1.'de ve K-NN algoritması için olasılık değerlerinin nasıl hesaplandığı ise Bölüm 3.2.'de ayrıntılı olarak verilmiştir. K-Means algoritmasında ise, olasılıklar yerine direk olarak öklit uzaklıklarına göre bir değerlendirme yapılmaktadır. Bu algoritma sonucunda elde edilen kümelerde bir aracın kapasitesi aşılmışsa, bu aracın düğümlerinden diğer araçların merkezlerine olan en kısa mesafede olan düğümler seçilerek uzaklıkları sıralanır. Bu düğümler arasından en kısa mesafedeki düğüm ve araç seçilerek, eldeki aracın düğüm listesinden çıkartılarak diğer aracın kapasitesi yeterliyse ekleme şeklinde gerçekleştirilmektedir. Bu prosedür, tüm araçların kapasiteleri yeterli hale gelene dek sürdürülür.

Kapasite dengeleme algoritması ile LR ve K-NN algoritmalarında hesaplanan olasılıklara dair nasıl bir prosedür işletildiğine dair bir örnek Tablo 1.'de verilmiştir. İki araçlı bir problem örneği için LR algoritması ile düğümlerin Araç 1'e ve Araç 2'ye atanma olasılıkları Tablo 1'deki gibi hesaplanmıştır. Eğer araçlara atamalar sonundaki kontrollerde 1. Aracın kapasitesi aşılsa, Araç 1'e atanacak düğümler içerisinde en düşük olasılıklı düğüm; yaklaşık 0,59 olasılıkla 47 numaralı düğüm (indis numarası: 15), Araç 1'e atanan düğümler içerisinde çıkarılarak Araç 2'ye atanmaktadır. Bu yeniden atama sonrası hala daha kapasite aşımı durumu varsa, Araç 1'e atanan düğümlerden 2. en düşük olasılıklı düğüm çıkartılarak, kapasite aşımı engellenene dek bu işleme devam edilir.

#### 4. Sayısal Sonuçlar (Numerical Results)

Yapılan çalışmada, Python ve Python kütüphaneleri; Ski-learn, Python-MIP, Pandas ve Numpy kullanılmıştır. Bu kütüphaneler

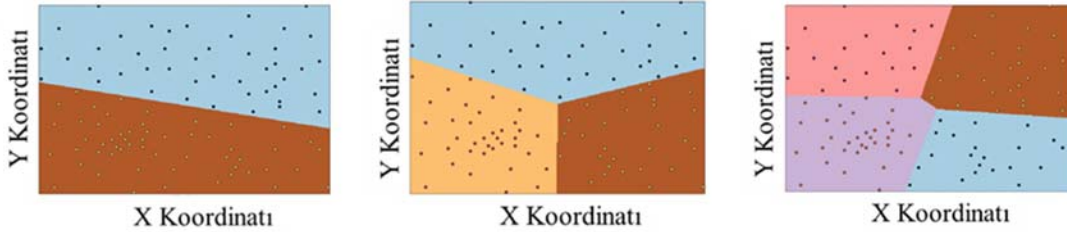
yardımıyla matematiksel modelleme formülasyonları ve makine öğrenmesi algoritmaları tek bir yazılım platformunda hazırlanmıştır. Bütün problemler, Python-MIP platformu kullanılarak Gurobi 9.12 ticari çözücüsü ile çözülmüştür ve test problemlerine ait sonuçlar AMD Ryzen 5 3500 U with Radeon Vega Mobile Gfx 2.10 GHz işlemcili bilgisayar çalıştırılarak alınmıştır.

#### Parametreler:

Bu çalışmada kullanılan LR ve K-NN algoritmaları daha yüksek bir başarı elde etmek için belirli hiperparametrelere gereksinim duymaktadır. Python Scikit-Learn kütüphanesi kullanılarak oluşturulan hiperparametre optimizasyonunda Izgara Arama (Grid Search) yaklaşımı kullanılmıştır. Bu yaklaşım ile olası parametre kombinasyonları deneyerek en iyi parametre kümesi bu iki yaklaşım için elde edilmiştir. Buna göre K-NN algoritmasında kullanılan komşuluk sayısı parametresi 'K' 15, ağırlık (weight) parametresi ise öklit uzaklıkları olarak alınmıştır. K-Means algoritmasında ise, K sayısı araç sayısı olarak belirlenmiştir.

#### Test Verileri:

Çalışma boyunca LR, K-NN ve K-Means algoritmaları 5 adet veri seti kullanılarak test edilmiştir. Bu veri setlerinde, ARP için 2-3-4 araçlı versiyonlar olmak üzere algoritmalar çalıştırılmış olup toplam 45 adet test problemi üzerinde deney sonuçları alınmıştır. Problemlerdeki araç kapasiteleri, toplam veri setindeki talep miktarının araç sayısına bölünmüş, ardından çıkan sonucun %10 fazlası alınarak belirlenmiştir. KARP matematiksel modelleme formülasyonu GUROBI 9.12 çözücüsü ile her bir problem tipi için 6 saat (21600 saniye) boyunca çalıştırılmış ve çözücü tarafından sunulan en iyi değerler ve alt sınır değerleri raporlanmıştır. Kullanılan veri setleri ise, 100 düğümlü; RC101[11], C101[11], 240 düğümlü Golden\_1[67], CMT3 [66] ve Tai100a[65] olup, veri setlerinin



Şekil 3. K-Means Algoritması ile Tahmin Sınıfının 2 Araçlı, 3 Araçlı ve 4 Araçlı Karar Sınırlar (Decision Boundaries of K-MEANS Algorithm Prediction Classification for 2 Vehicles, 3 Vehicles and 4 Vehicles)

Tablo 1. LR Algoritması ile Düğümlerin Araçlara Atanma Olasılıkları İlişkin Bir Örnek (An Example for Probabilities of Node-to-Vehicle Assignment Using LR Algorithm)

İndis	Düğüm numarası	X-Koordinatı	Y-Koordinatı	Talep	LR ile Araç Tespiti	Araç 1 için Olasılıklar	Araç 2 için Olasılıklar
1	62	4	-6	4	1	0,99999237029208	0,000000762970792
2	63	18	-4	394	1	0,99999720737659	0,00000002792623
3	64	10	-1	3	1	0,999996921283165	0,000000003078717
4	65	16	-10	60	1	0,99999974018208	0,000000259817920
5	66	96	5	899	1	0,99999999598236	0,000040176319000
6	67	60	28	169	1	0,998704918496372	0,000129508150000
7	68	75	-19	10	1	0,99999999999826	0,001735267900000
8	69	107	5	47	1	0,99999999908748	0,000091251858000
9	70	70	-4	37	1	0,99999999747109	0,000025289059000
10	71	110	-2	23	1	0,9999999997214	0,000027854605000
11	72	90	3	25	1	0,99999999626492	0,000037400000000
12	73	80	-17	444	1	0,99999999999786	0,000214000000000
13	74	59	-21	3	1	0,99999999999379	0,006210000000000
14	75	83	14	99	1	0,99999877730699	0,000000001220000
15	47	46	38	119	1	0,587662138056631	0,412337862000000

yerleşimine ait şema grafikleri bu sırada Şekil 4’de sunulmuştur. Bu veri setlerine ait düğüm talepleri de değiştirilmeden deneyler boyunca kullanılmıştır. Veri setleri literatürde sıklıkla kullanılan ARP’nin farklı türleri için türetilmiş ve çeşitli özellikleri yansıtacak şekilde seçilmiştir.

#### 4.1. Kapasite Dengeleme Algoritması Sonuçları (Capacity Balancing Algorithm Results)

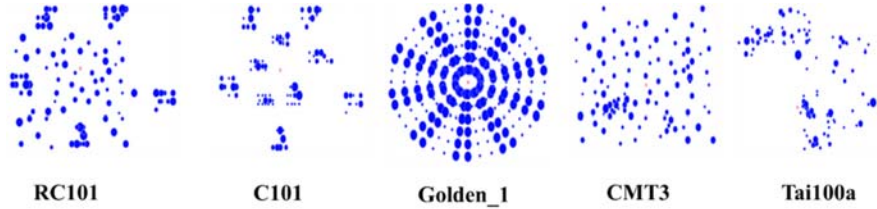
Çalışma kapsamında değerlendirilen 45 test probleminin 37 ’sinde araç kapasitesi aşıldığı için, araçların yüklerinin kapasiteleri aşmayacak hale getirilmesi kapasite dengeleme algoritması sayesinde gerçekleştirilmiştir. Denetimsiz makine öğrenmesi olan K-Means algoritmasında ise, kapasite dengelemek için düğümlerin merkezlere olan uzaklıklarına ilişkin bir dengeleme yapılmıştır. Tablo 2’de kullanılan 5 veri setinde (45 test problemi) kapasite dengeleme algoritmasının kullanıldığı ve kullanılmadığı durumlar verilmiştir. Tablo 2’den elde edilen sonuçlara göre, kapasite dengeleme algoritması özellikle 2 araçlı 15 problem verisi için 9 problem verisi tarafından kullanılmış, 3 araçlı ve 4 araçlı örnekler için ise sadece K-Means algoritması Golden\_1 ve CMT3 veri seti tarafından ihtiyaç duyulmamıştır. K-Means algoritması küme içi benzerliklerin maksimizasyonunu dikkate alarak aradıkları sonuçta çalıştığı için öznitelik olarak alınan X ve Y koordinatlarına göre düğümlerin belli noktalarda toplanması sonucu kümeleme yapılmaktadır, Golden\_1 ve CMT3 veri setinin bu yüzden kapasite aşımıyla karşılaşmadığı düşünülmektedir.

#### 4.2. Matematiksel Model ve Algoritmalar Ait Sayısal Sonuçlar (Computational Results of Mathematical Models and Algorithms)

Bu bölümde öncelikle, araç sayılarının algoritmalar üzerindeki etkisi incelenmek istenmiştir. Bu amaçla, her bir araç sayısına ait algoritma

sonuçları ve KARP’ın GUROBI 9.12 çözücüsü ile 6 saat çalıştırılmasından elde edilen çıktılar karşılaştırılmıştır. Ayrıca GUROBI 9.12 tarafından verilen alt sınır değerleri de Tablo 3’de yer almaktadır.

Tablo 3’ten görüleceği üzere, K- Means & GSP çözümü, RC101 ve Tai100a veri setleri olmak üzere, KARP ’ın GUROBI ile çözdürülmesinden daha iyi sonuç vermiştir. Bu iki veri setinde K-Means&GSP çözümü KARP alt sınırına yaklaşık %90 ve %80 civarında yaklaşmışlardır. LR&GSP çözümü ise, RC101 ve Tai100a veri setlerinde KARP çözümünden daha iyi sonuç verirken, CMT3 veri setinde ise, KARP çözümü ile aynı sonuca erişmiştir. K-NN&GSP çözümü incelendiğinde ise, algoritmanın K-Means&GSP çözümü aynı davranışı sergilemiş olduğu görülmektedir. Algoritmaların tutarlı bir şekilde RC101 ve tai100a veri setlerinde daha iyi sonuç vermesinin nedeni, bu veri setlerindeki düğüm dağılımlarının rassal bir şekilde yerleşime dayalı olduğu, bu nedenle de, algoritmaların yaptığı sınıflandırmanın KARP çözümüne göre daha başarılı olduğu düşünülmektedir. Ayrıca amaç fonksiyonları ve kullanılan CPU süreleri incelendiğinde ise, RC101 veri setinde tüm makine öğrenmesi algoritmalarının KARP alt sınırına göre %50 daha az bir süre içinde ortalama olarak %10 ve daha üzeri oranlarda daha iyi bir sonuç verdiği görülmektedir. Makine öğrenmesi & GSP algoritmaları, C101 ve Golden\_1 veri setlerinde ise her bir algoritma 6 saat sürelerinin tamamını kullanmıştır. Ancak bu veri setlerinde, makine öğrenmesi algoritmalarının hiçbirisi KARP’ın sonucuna (veya alt sınıra) erişememiş, ortalama olarak KARP’a %90 yakınlıkta amaç fonksiyonu değerlerine erişebilmiştir. KARP alt sınırına yakınlık değerleri ise C101 için yaklaşık %45 seviyesindeyken, Golden\_1 için bu oran en fazla %4 civarında gerçekleşmiştir. Burada dikkat çekmek istediğimiz nokta, Golden\_1 240 düğümlü bir veri seti olmasına rağmen, düğümlerin dağılımının eğitim setine benzemesi, düğüm



Şekil 4. RC101, C101, Golden\_1, CMT3 ve Tai100a Veri Setlerine ait Şema Grafikleri  
(Schema Graphics of RC101, C101, Golden\_1, CMT3 and Tai100a Datasets)

Tablo 2. Kapasite Dengeleme Algoritmasının Kullanıldığı Test Problemi Örnekleri  
(Test Problem Samples that uses Capacity Balancing Algorithm)

Kapasite Dengeleme Algoritması Kullanılan Model ve Verisetleri	LR			K-NN			K-Means		
	2-Araçlı	3-Araçlı	4-Araçlı	2-Araçlı	3-Araçlı	4-Araçlı	2-Araçlı	3-Araçlı	4-Araçlı
RC101	+	+	+	+	+	+	-	+	+
C101	+	+	+	+	+	+	+	+	+
Tai100a	-	+	+	-	+	+	+	+	+
Golden_1	+	+	+	+	+	+	-	+	-
CMT3	-	+	+	-	+	+	+	-	+

Tablo 3. 2-Araçlı Makine Öğrenmesi Modelleri ve GSP Çözüm değerlerinin KARP ile Karşılaştırılması  
(2-Vehicle Machine Learning Models&GSP Solutions comparison with CVRP)

RC101	C101		Tai100a		Golden 1		CMT3			
	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)		
K-Means	668,7	11011,93	569,87	21600	1046,4	11022,88	4335,3	21600	673,01	3086,32
LR	675,5	11478	573,45	21600	1035,2	10843,48	4337,5	21600	651,3	48,76
K-NN	692,8	10858,63	578,74	21600	1033,8	10821,55	4352	21600	659,4	34,56
KARP	763	21600	531,98	21600	1148,0	21600	4313,8	21600	651,3	21600
KARP Alt Sınır	608,3	21600	390,39	21600	866,07	21600	4174,4	21600	640,2	21600



sayısı artsa bile başarıya sebep olmuştur. CMT3 veri setinde makine öğrenmesi modellerinin KARP 'a göre %25 daha az CPU süresi kullanılarak KARP tarafından elde edilen amaç fonksiyonu değerine %95 oranında yaklaşıldığı tespit edilmiştir. LR&GSP ve K-NN&GSP sonuçlara CMT3 veri setinde 1 dakikadan daha az bir CPU süresinde eriştiği görülmektedir. Dolayısıyla, algoritmaların aynı zamanda veri setlerine bağlı olarak da performanslarının değiştiği aşikardır. KARP alt sınır değeri CMT3 bazında incelendiğinde ise, ortalama olarak alt sınıra yüzde olarak en fazla yaklaşım oranı görülmektedir (%1.7). C101 veri setinde ise en iyi çözüm GUROBI çözücüsünden elde edilmesine rağmen KARP'ın kendi çözümünün alt sınıra olan uzaklığı %36 civarında gerçekleşmiştir.

Algoritmaların veri setlerine göre davranışını daha iyi analiz edebilmek için, 2 araçlı veri setlerinde tüm algoritmalarda KARP 'tan daha iyi sonuç veren Tai100a veri setinin, tüm algoritmalar için rotaları çizilmiş ve Şekil 5'de sunulmuştur. Tai100a veri seti çözümünde kullanılan algoritmaların düğüm sayısı, araçların kullanılan kapasiteleri ve rota uzunlukları Tablo 4'de sunulmuştur. Bu veri setinde sadece K-Means algoritması kapasite dengeleme algoritmasını kullanmıştır.

Tai100a veri seti için KARP çözüm sonucu incelendiğinde, rotalardaki düğüm sayılarının dağılımının diğer algoritmalara oranla daha dengesiz bir şekilde yer aldığı görülmektedir (Tablo 4). Rotalar talep miktarlarına bağlı olarak da değişim göstermektedir, ancak, makine öğrenmesi algoritmalarının rotalardaki düğüm sayılarının birbirine daha yakın olması araç kapasitesinin aşılmasının göstergesidir. K-Means&GSP 'nin çözümü, kapasite dengeleme algoritmasıyla dengelemesi sonucunda, araçların atandıkları düğüm sayılarının birbirine en yakın şekilde oluştuğu görülmektedir. Ancak

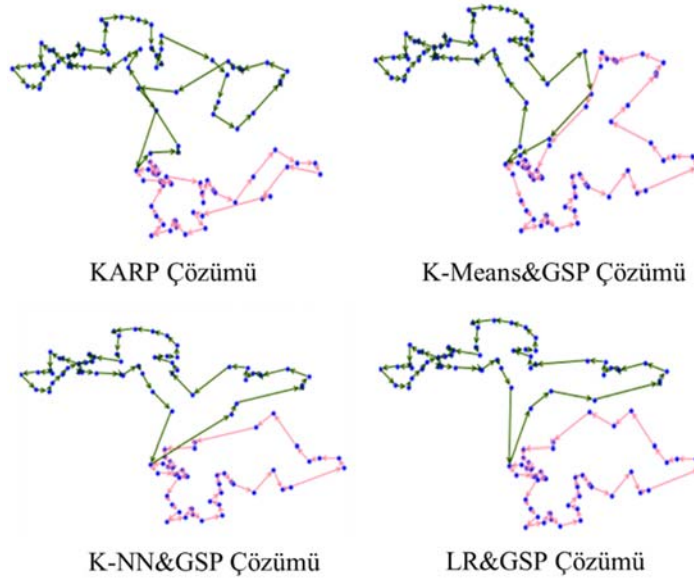
K-Means&GSP 'nin çözümünden elde edilen amaç fonksiyonu değeri, diğer iki algoritmaya kıyasla küçük bir farkla daha yüksek orandadır. Rotaların yapısı incelendiğinde ise, KARP'ın çözümünün, rotalarının uzunluklarının birbirlerinden oldukça farklı yerleşmiş olması amaç fonksiyonunda en yüksek değeri de beraberinde getirmiştir. Makine öğrenmesi algoritmalarıyla alınan çözümlerde ise, düğüm sayıları benzer dağıldığından rota uzunlukları da birbirine benzer oluşmuştur ve bu sebeple amaç fonksiyonu değerlerinin KARP matematiksel modelinin çözümünden daha düşük çıktığı yorumu yapılabilir (Şekil 5).

3-Araçlı Makine Öğrenmesi Modelleri ve ardından oluşan sınıflandırmaların (kümelerin) GSP ile çözümünün KARP'ın GUROBI 9.12 ile çözdürülmesi sonucu algoritmaların performans karşılaştırma verileri Tablo 5'te sunulmuştur.

K-Means&GSP 'nin çözümü RC101, Tai100a ve CMT3 veri setlerinde KARP matematiksel çözümünden daha iyi sonuç vermiştir. Özellikle KARP matematiksel modelinin CPU süresinde göz önüne alındığında, Tai100a (98,25 sn.) ve CMT3 (12,46 sn.) veri setlerinde, K-Means&GSP 'nin çok daha kısa CPU süresinde daha iyi sonuçlara eriştiği görülmektedir. LR&GSP çözümü incelendiğinde C101 veri seti dışında, bütün veri setlerinde daha az CPU süresinde, algoritmanın KARP çözümüne göre daha iyi sonuçlar elde ettiği görülmektedir. Özellikle CMT3 veri setinde 2,44 saniyede KARP'tan çok daha iyi bir sonuca ulaşılmıştır. K-NN&GSP çözümü sonuçları, tüm veri setlerinin ortalamaları değerlendirildiğinde, KARP matematiksel modelinin sonucuna CPU süresinin yaklaşık %30'unu kullanarak %90'lar seviyesinde yaklaştığı tespit edilmiştir. 2 araçlı veri setleri çözümünde olduğu gibi, tüm algoritmaların KARP alt sınırına en yakın olduğu veri seti CMT3 iken, en uzak olduğu veri

**Tablo 4.** Tai100a veriseti çözümünde kullanılan algoritmaların düğüm-araç kapasitesi-rota uzunlukları (Nodes-Vehicle Capacity-Route Length of Algorithms that is found in Solution of Tai100a Dataset)

	KARP		K-Means		LR		K-NN	
	1.Araç	2.Araç	1.Araç	2.Araç	1.Araç	2.Araç	1.Araç	2.Araç
Rotalanan Düğüm Sayısı	40	60	52	48	46	54	45	55
Kullanılan Araç Kapasitesi	6957	8246	8055	7148	7364	7839	7245	7958
Araçların Kullandığı Rota Uzunlukları	425,76	722,24	523,87	522,52	431,74	602,13	441,96	593,28
Toplam Uzunluk	1148		1046,39		1033,87		1035,24	



**Şekil 5.** 2 araçlı Tai100a veri seti için tüm algoritmaların rotaları (Routes of all Algorithms for 2 Vehicle Tai100a Dataset)

setleri ise C101 ve Tai100a olmuştur. C101 ve Tai100a alt sınırları KARP'nın eldeki çözümlerine göre yaklaşık %36 ve %38 oranında uzaktır.

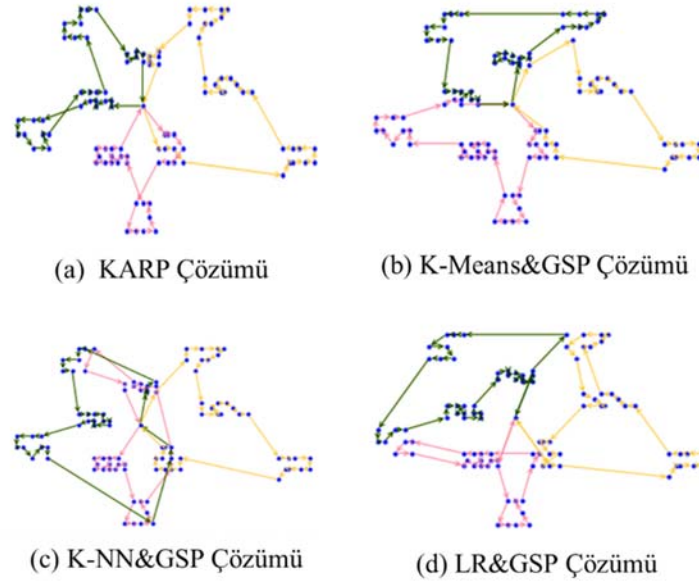
3-Araçlı Makine Öğrenmesi algoritmaları ve GSP'nin çözüm sonuçları C101 veri seti özelinde incelendiğinde, tümünün KARP çözümüne göre daha yüksek çıktığı görülmektedir (Tablo 6). Bu nedenle algoritmaların bu veri seti üzerinde davranışlarını yakından incelemek için C101 veri setine göre algoritmaların rotaları Şekil 6'da çizilerek, incelenmiştir. C101 veri seti çözümünde kullanılan algoritmaların düğüm sayısı, araçların kullanılan kapasiteleri ve rota uzunlukları ise Tablo 6'da sunulmuştur. Bu problem setinde, bütün makine öğrenmesi algoritmaları sınıflandırma ve kümelendirme yaptıktan sonra kapasite dengeleme algoritmasının kullanılmasına ihtiyaç duyulmuştur. C101 veri seti için araçların kapasiteleri 663 olarak hesaplanmıştır. Çözümlerde kapasitenin üst sınırına çok yakın olarak araç kapasitelerinin kullanımı dikkati çekmektedir. Bu durumun nedenini kullanılan üç algoritmanın da kapasite dengeleme

prosedürüne ihtiyaç duyması olarak açıklanabilir. Şekil 6a'da KARP çözümü incelendiğinde, 1. ve 2. aracın rotalarının birbirini kestiği bir duruma rastlanmıştır. Bu durumun nedeninin araçların kapasite kısıtından kaynaklanabileceği düşünülmektedir. K-Means&GSP ve LR&GSP çözümlerinde rotaların birbirini kestiği bir durumla karşılaşmamıştır. Bu durumun KARP çözümüne en yakın amaç fonksiyonu değerinin bu iki çözümden gelmeye neden olduğu düşünülmektedir. Ayrıca bu veri seti literatürden bilindiği üzere kümelenmiş verileri içermektedir. LR&GSP çözümü, diğer makine öğrenmeleri algoritmaları arasında KARP matematiksel model çözümüne ve alt sınırına en yakın sonuç bulan algoritmadır. KARP çözümüne benzer şekilde, 3 aracın rotasının çakışması K-NN&GSP çözümü sonucunda izlenebilir. K-NN&GSP çözümünün amaç fonksiyonları açısından en kötü sonuç vermesinin nedeninin rotaların yerleşimi kaynaklı olduğu sonucu çıkarılabilir (Şekil 6c.).

K-Means&GSP çözümünde ise en yüksek düğüm sayısına sahip rotayla karşılaşmıştır (Tablo 6). Ancak buna rağmen, 40 düğümlü en

**Tablo 5.** 3-Araçlı Makine Öğrenmesi Modelleri ve GSP Çözüm Değerlerinin KARP ile Karşılaştırılması (3-Vehicle Machine Learning Models comparison with CVRP)

RC101	C101		Tai100a		Golden 1		CMT3			
	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)		
K-Means	698,1	703,47	583,6	3994,05	1177,2	98,25	4527,3	8097,1	685,4	12,46
LR	704,0	455,89	579,1	14691,14	1051,1	2562,25	4393,3	10266,22	664,3	2,44
K-NN	774,5	132,13	654,7	14888,13	1439,9	117,31	4602,7	21600	706,8	9,51
KARP	720,23	21600	564,08	21600	1209,32	21600	4521,28	21600	696,94	21600
KARP Alt Sınır	612,45	21600	410,56	21600	872,01	21600	4119,38	21600	639,62	21600



**Şekil 6.** 3 araçlı C101 veri seti için tüm algoritmaların rotaları (Routes of all Algorithms for 3 Vehicle C101 Dataset)

**Tablo 6.** C101 Veriseti çözümleri için düğüm-araç kapasitesi-rota uzunlukları (Nodes-Vehicle Capacities-Route Lengths for the solutions of C101 Dataset)

	KARP			K-Means			LR			K-NN		
	1.Araç	2.Araç	3.Araç	1.Araç	2.Araç	3.Araç	1.Araç	2.Araç	3.Araç	1.Araç	2.Araç	3.Araç
Rotalanan Düğüm Sayısı	38	28	34	40	25	35	28	37	35	28	38	34
Araç Kapasitesi	650	510	650	660	490	660	490	660	660	510	640	660
Araçların Rota Uzunlukları	227,9	142,0	194,0	208,6	188,1	186,8	152,7	210,1	216,2	181,6	243,1	229,9
Toplam Rota	564,07			583,61			579,16			654,76		

fazla düğüm sayısı içeren rotanın uzunluğu 208,6 br olarak gerçekleşmiş olup, birbirine yakın şekilde dengeli bir şekilde yerleşen kümenin rotalandığında uzunluğunun, düğüm sayısının daha az olduğu (örneğin K-NN 38 düğüm:243,1 br) rotalardan daha kısa mesafelerde gerçekleştiği gözlenmiştir. Şekil 6'da görülen veri setinde (C101) makine öğrenmesi algoritmalarının kümelendirme sonuçlarının GSP ile çözümlerinin, KARP' a göre daha yüksek amaç fonksiyonu bulmasının sebebi, gözetimli öğrenmede kullanılan veri setinin R101 veri seti olması; bu veri setinin de rassal ve homojen olarak dağıtılmış düğümlerden oluşması ve burada kullanılan C101 veri setinin kümelenebilir bir veri seti olması şeklinde açıklanabilir.

4-Araçlı Makine Öğrenmesi Modellerinin, KARP Matematiksel Modelleme Formülasyonu Sonucu ile Karşılaştırılması Tablo 7'de sunulmuştur.

K-Means&GSP algoritması sonuçları RC101, Golden\_1 ve CMT3 veri setlerinde KARP için ticari çözümden alınan çözüme göre daha iyi sonuç vermiştir. Amaç fonksiyonu olarak daha düşük sonuç alınamayan C101 ve Tai100a veri setlerinde ise, KARP için GUROBI 9.12 ticari çözücüsünün verdiği çözüme sırasıyla %92 ve %98 oranında yaklaşmış olup ticari çözüme göre kullanılan süreler de oldukça düşük olarak gerçekleşmiştir (790 sn; 1330 sn). Burada alt sınıra olan yakınlık ise sırasıyla %57,88 ve %50,72 olarak gerçekleşmiştir. LR&GSP ile çözümünde ise, RC101 ve CMT3 veri setlerinde daha iyi sonuç elde edilmiş olup, CMT3 veri setinde sadece 8,44 saniyede bu çözüme ulaşılmıştır. C101, Tai100a ve Golden\_1 veri setlerinde ise, amaç fonksiyonlarının KARP' a göre yakınlık yüzdeleri sırasıyla; %90, %96 ve %97 KARP alt sınır dikkate

alındığında yakınlık yüzdeleri ise yaklaşık %62, %54 ve %17 oranında gerçekleşmiştir. K-NN&GSP çözümleri ise, 3 araçlı çözümlerdekine benzer bir trend göstermiş ve RC101, C101, tai100a veri setlerinde en yüksek amaç fonksiyonu değerini türetmiştir. Kullanılan CPU süreleri ise, KARP çözümü için verilen sürenin %25 ve %30 civarı arasında değişim göstermiştir. Dolayısıyla önerilen algoritmaların oldukça kısa CPU sürelerinde, daha iyi çözümlere erişebildiği görülmüştür. Ancak Tablo 7'den izlenebileceği üzere hiçbir çözüm KARP alt sınır değerlerine erişememiştir.

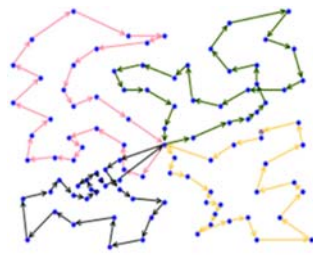
4-Araçlı makine öğrenmesi algoritmalarının özellikle CMT veri setinde daha iyi sonuçlar elde ettiği görülmektedir. Bu amaçla bu veri setinde algoritmaların davranışını izlemek için oluşan rotalar Şekil 7'de çizilerek gösterilmiştir.

CMT3 veriseti çözümünde kullanılan algoritmaların düğüm sayısı, araçların kullanılan kapasiteleri ve rota uzunlukları Tablo 8'de sunulmuştur. Bütün makine öğrenmesi algoritmaları kapasite dengeleme algoritmasını kullanmışlardır.

Şekil 7 üzerindeki çözümler incelendiğinde, KARP çözümünde 1. ve 4. Araç rotalarının birbiri içinden geçtiği duruma rastlanmıştır. Araç rotalarının birbiri içinden geçtiği durum sadece KARP çözümünde izlenmiştir (Şekil 7a). Bu tarz bir çözümün KARP'ın amaç fonksiyonu değerinin en yüksek oluşmasına sebep olduğu düşünülmektedir. LR&GSP ve K-NN&GSP çözümünden elde edilen rotaların ise birbirine çok benzer olduğu görülmektedir (Şekil 7c., Şekil 7d.). Tablo 8 incelendiğinde, araç rotalarının uzunluklarının da birbirlerine çok yakın şekilde oluştuğu gözlemlenmektedir. Ayrıca, en başarılı

**Tablo 7.** 4-Araçlı Makine Öğrenmesi Modelleri ve GSP Çözüm Değerlerinin KARP ile Karşılaştırılması (4-Vehicle Machine Learning Models comparison with CVRP)

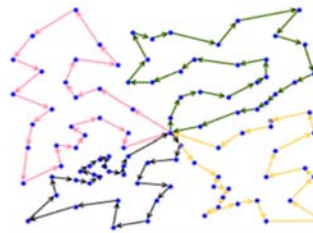
RC101	C101		Tai100a		Golden_1		CMT3			
	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)	Amaç F.	Toplam Süre (s)		
K-Means	795,8	172,64	665,5	790,22	1330,6	5418,18	4716,4	5396,5	711,5	16,05
LR	802,0	749,11	684,5	5663,93	1360,8	5571,99	4915,9	7432,02	701,1	8,44
K-NN	836,9	417,46	714,7	454,51	1635,1	10815,48	4822,7	5483,51	710,3	38,87
KARP	814,3	21600	610,8	21600	1309,5	21600	4800,1	21600	719,7	21600
KARP Alt Sınır	615,3	21600	421,5	21600	882,8	21600	4174,5	21600	654	21600



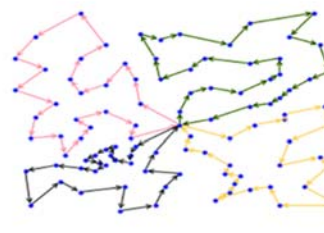
(a) KARP Çözümü



(b) K-Means&GSP Çözümü



(c) K-NN&GSP Çözümü



(d) LR&GSP Çözümü

**Şekil 7.** 4 araçlı CMT3 veri seti için tüm algoritmaların rotaları (Routes of all Algorithms for 4 Vehicle CMT3 Dataset)

**Tablo 8.** CMT3 veriseti çözümünde kullanılan algoritmaların düğüm-araç kapasitesi-rota uzunlukları (Nodes-Vehicle Capacity-Route Length of Algorithms that is found in Solution of CMT3 Dataset)

	KARP				K-Means			
	1.Araç	2.Araç	3.Araç	4.Araç	1.Araç	2.Araç	3.Araç	4.Araç
Rotalanan Düğüm Sayısı	24	26	25	25	25	22	26	27
Kullanılan Araç Kapasitesi	361	378	354	365	371	322	385	380
Araçların Kullandığı Rota Uz.	162,01	191,75	205,69	160,24	174,29	191,79	155,44	190,04
Toplam Rota	719,69				711,56			
	LR				K-NN			
	1.Araç	2.Araç	3.Araç	4.Araç	1.Araç	2.Araç	3.Araç	4.Araç
Rotalanan Düğüm Sayısı	24	27	24	25	24	28	22	26
Kullanılan Araç Kapasitesi	312	386	361	399	338	395	329	396
Araçların Kullandığı Rota Uz.	197,32	191,45	162,01	150,35	206,45	197,33	159,5	147,07
Toplam Rota	701,13				710,35			

amaç fonksiyonuna LR&GSP çözümü tarafından ulaşıldığı görülmektedir. LR&GSP çözümü incelendiğinde ise (Şekil 7.d., Tablo 8) rotaların düğüm sayıları birbirine çok yakın şekilde ve düşük uzaklık toplamlarıyla oluşmuştur. En düşük rota uzunluğa sahip K-NN&GSP çözümünün ise (147,07 br); K-Means&GSP çözümüne rota yerleşimleri açısından oldukça farklılık göstermesine rağmen amaç fonksiyonları açısından oldukça yakın olması dikkat çekicidir. Bu durumun nedeni, CMT3 veri setinin rassal bir şekilde dağılmış düğümlerden oluşması nedeniyle, bu durumun farklı rota yerleşimlerinin bile benzer amaç fonksiyonu değerlerine yol açabileceği şeklinde yorumlanabilir.

Bu bölümde yapılan bir diğer analiz ise, algoritmaların tüm araç sayıları dikkate alındığında KARP çözümü ile karşılaştırılması olmuştur. K-Means&GSP için 3 ve 4 araçlı örneklerde genel ortalamada KARP'tan daha büyük amaç fonksiyonu sonuçları alındığı görülmektedir. Aynı problem örneklerinde, çözüm almak için gereken sürelerde ortalama olarak ticari çözümcüye verilen sürenin yaklaşık %11,5 civarında gerçekleşmiştir. Ancak 2 araçlı örneklerde bu çözüm tekniği daha başarılı şekilde, özellikle düğümlerin rassal olarak dağılmış veri setlerinde sonuç KARP çözüm sonucuna göre daha düşük oluşmuştur. Bu örneklerde ayrıca KARP'ın ticari çözümcüdeki süresinin yaklaşık %63'ü kullanılmıştır. Dolayısıyla, K-Means&GSP çözüm tekniği araç sayısı düşük olan örneklerde genel olarak daha başarılı çözümler vermiştir. K-NN&GSP çözümleri incelendiğinde ise, genel toplamda KARP modelinin çözümü için verilen sürenin %36,68'ini kullanarak %4,70 daha büyük amaç fonksiyonuna sahip bir sonuç elde ettiği tespit edilmiştir. K-NN&GSP çözümünün araç sayısı arttıkça KARP'a göre sonuçtaki etkinliği düşmektedir. Ancak KARP çözümüne göre toplamda kullanılan sürenin en düşük olduğu çözüm yöntemi K-NN&GSP çözümü olmuştur. Dolayısıyla, karar vericiler için daha kısa sürede daha hızlı çözümlerin önemli olduğu durumlarda bu çözüm tekniği bir tercih sebebi olabilir. Verilen LR&GSP çözümlerinde ise, genel toplamda KARP modeline verilen sürenin %34,86'sını kullanarak, KARP çözümüne göre ortalama %1,34 daha iyi bir sonuç elde edebilmiştir. LR&GSP çözümleri incelendiğinde, 4 araçlı test örnekleri için KARP'a göre bir düşüş olmasına rağmen, tüm test problemleri açısından performansları incelendiğinde, en başarılı yöntem olmuştur. Ayrıca süre olarak da KARP çözümü için gereken sürenin yaklaşık üçte biri kullanıldığı için, karar vericiler hem hızlı hem de başarılı sonuçlar veren bir yöntem olarak LR&GSP çözümünü tercih edebilirler. Son olarak, algoritmaların başarısı karşılaştırıldığında ise, K-NN&GSP çözümü kullanılan süre olarak en başarılı algoritma olduğu, amaç fonksiyonları olarak karşılaştırıldığında ise, LR&GSP çözümünün en başarılı sonuçlara ulaşabildiği izlenmiştir. Ancak, bu sonuçlar kullanılan yapılan analizlerden de izlenebileceği gibi veri setine ve kullanılan araç sayısına bağlı olarak değişiklik gösterebilmektedir.

#### 4.3. Önerilen Hibrid Yaklaşımın Geniş Komşuluk Araması Algoritmasıyla Karşılaştırılması (Comparison of proposed approach with Large Neighborhood Search Algorithm)

Bu bölümde son olarak önerilen yaklaşımın başarısı, literatürde araç rotalama problemleri için en başarılı metasezgisellerden biri olan Geniş Komşuluk Araması (GKA) Algoritması ile karşılaştırılmıştır. Geniş komşuluk arama algoritması Shaw [25] tarafından geliştirilmiş olup, çözümü bozma ve tamir etme operatörlerinin ardı ardına çalışması prensibine sahiptir (Shaw [25]). Yazar çalışmasında, topla dağıt araç rotalama problemine ait bir başlangıç çözümüne bir adet bozma (destroy) ve bir adet onarım (repair) operatörünü ardışık olarak uygulayarak ilk çözümü kademeli olarak geliştirmeye çalışmıştır. Geniş komşuluk arama algoritmasının temel fikri, bir başlangıç çözümünden başlayıp daha iyi bir çözüme doğru ard arda bozma ve tamir etme operatörlerinin aşamalı şekilde uygulanarak gelişme sağlanmasına dayanmaktadır. Bu temel prensibe göre, eldeki çözümün büyük bir kısmı değiştirilerek daha iyi bir çözüm aranır.

Bu çalışmada, önerilen yöntemin başarısını karşılaştırmak için Erdoğan [26] tarafından açık kaynak kodlu olarak literatüre sunulan VRP SpreadSheet Solver olarak adlandırılan GKA algoritması kullanılmıştır. Erdoğan [26] çalışmasında belirtildiği üzere, ARP için geliştirilmiş bu Excel tabanlı çözücü 50 düğümlük ARP örneklerini 15 dakika süre ile ve düğüm sayısı arttıkça sürenin de doğrusal olarak arttırılmasıyla 200 düğümlük örnekleri 60 dakika (3600 saniye) süre içinde başarıyla çözmektedir. Bu çalışmada ise, önerilen yaklaşımlarla eşit bir ortamın sunulması için standart parametreler ile her bir problem tipi 6'şar saat çalıştırılmıştır. Tablo 9'da ise yaklaşıma bütünsel olarak bakılmış olup, her bir problem tipi ve veri seti için algoritmaların bulunduğu en iyi değerler ile GKA algoritması tarafından elde edilen değerlerle karşılaştırılmıştır.

Tablo 9'dan görüleceği üzere önerilen yaklaşımla elde edilen sonuçlar, literatürde oldukça başarılı olarak bilinen GKA algoritmasına yakın veya daha iyi sonuçlar şeklinde ortaya çıkmıştır. GKA algoritmasından yüzde olarak en uzak sonuç %11 iken (Tai100a\_4v), en iyi çözümün bulunamadığı en yakın uzaklık ise %0,04 (Golden\_1\_2v) şeklinde gerçekleşmiştir. Önerilen yaklaşımlarla, Tablo 9'dan görülen yüzde yakınlıklar en iyi çözümlere %0,8 uzakken, GKA algoritmasıyla en iyi çözümlere olan yüzde yakınlık değerleri %4,12 oranında gerçekleşmiştir. Eğitim setinin bazı parametreler kullanılarak bir miktar kümelennmiş hale çevrilmesi ile elde edilmiş RC101 veri setinde K-means&GSP çözüm algoritmasının baskın olduğu görülmektedir. Golden\_1 veri setinde de aynı durumla karşılaşılmıştır. Bunun nedeni, RC101 ve Golden\_1 veri setlerindeki düğümlerin dağılımının bir miktar benzemesi olarak açıklanabilir. CMT3 veri setinde ise, tüm araçlar için LR

**Tablo 9.** LR; K-NN ve K-Means Algoritmalarından Alınan En İyi Sonuçlarla GKA Algoritmasının Sonuçları  
(Comparison of LNS vs LR; K-NN and K-Means Algorithms' Best Results)

Test verisi_ Araç Sayısı	Algoritma	En İyi Değer	%Yüzde Yakınlık	Süre (sn)	GKA	%Yüzde Yakınlık	Süre (sn)
RC101_2v	K-Means	668,69	0,0	11011,93	684,46	0,023	21600
RC101_3v	K-Means	698,1	0,0	703,47	738,42	0,054	21600
RC101_4v	K-Means	795,85	0,0	172,64	802,26	0,007	21600
C101_2v	LR	573,45	0,01	21600	567,54	0,0	21600
C101_3v	K-Means	583,6	0,0	3994,05	624,21	0,065	21600
C101_4v	K-Means	665,55	0,0	790,22	726,04	0,083	21600
tai100a_2v	K-NN	1033,87	0,0	10821,55	1042,47	0,008	21600
tai100a_3v	LR	1051,1	0,0	2562,25	1084,02	0,03	21600
tai100a_4v	K-Means	1330,66	0,11	5418,18	1192,25	0,0	21600
Golden_1_2v	K-Means	4335,38	0,004	21600	4317,82	0,0	21600
Golden_1_3v	K-Means	4527,3	0,0	8097,1	4565,64	0,008	21600
Golden_1_4v	K-Means	4716,44	0,0	5396,5	4757,48	0,008	21600
CMT3_2v	LR	651,3	0,0	48,76	688,58	0,054	21600
CMT3_3v	LR	583,61	0,0	3994,05	724,49	0,194	21600
CMT3_4v	LR	701,13	0,0	8,44	765,69	0,084	21600
	Ortalama	1527,74	0,008	6414,6	1552,09	0,0412	21600

algoritmasının başarılı olduğu görülmektedir. CMT3 veri seti eğitim setine en çok benzeyen veri seti olarak dikkat çekmektedir. Tai100a ve C101 veri setleri ise ele alınan R101 veri setinden çok farklı yerleşime sahip oldukları için algoritmaların başarısı açısından bir eğilim gözlenmemiştir. Dolayısıyla daha geliştirilebilir özneliklerin kullanılmasıyla algoritmaların performanslarının da iyileştirilebileceği aşikardır.

## 5. Sonuçlar (Conclusions)

Bu çalışmada, makine öğrenmesi ve yönelem araştırması tekniklerinin birlikte kullanılmasına ve geliştirilmesine örnek teşkil edecek hibrid bir yöntem önerisinde bulunulmuştur. Bu amaçla, araç rotalama problemi üzerinde çalışılmış olup, gözetimli makine öğrenme algoritmalarından LR ve K-NN algoritmaları, gözetimsiz makine öğrenme algoritmalarından ise K-Means algoritmaları literatürde en sık kullanımına rastlanılan algoritmalar olduğu için bu çalışmada yer almışlardır. Yöntem olarak, kapasiteli araç rotalama problemi öncelikle literatürden rassal olarak türetilmiş bir veri seti için çözdürülmüş ve gözetimli makine öğrenme algoritmaları bu veri seti çözümü üzerinden eğitilmiştir. Eğitilen algoritmalar gerçek hayatta özellikle lojistik şirketlerinin karşılaştığı hizmet verilmesi gereken düğüm yerlerinin farklı yerlerde konumlandırılması ve düğüm sayılarının fazlalığı dikkate alındığından, literatürden araç rotalama problemlerinin çeşitli tipleri için sıklıkla kullanılan başka veri setleri üzerinde sınıflandırma veya kümeleme yapmak üzere kullanılmışlardır. Gözetimli makine öğrenme algoritmalarından elde edilen sınıflandırmalar ile, gözetimsiz makine öğrenme algoritmalarından elde edilen kümeleme sonuçları, her bir küme tek bir GSP oluşturulacak şekilde karma tamsayı matematiksel programlama modeli ile elde edilmiştir. Araçlara atanan düğümlerin araç kapasitesini aşması durumunda kapasite dengeleme algoritması adı verilen prosedür devreye girmektedir. Literatürde bilindiği kadarıyla makine öğrenmesi ve yönelem araştırması tekniklerini bir araya getirerek, KARP çözümü dahilinde inceleyen böyle bir çalışmaya rastlanmamıştır.

Önerilen hibrid yaklaşımlardan KARP matematiksel modelinin çözümünden çok daha kısa sürede başarılı sonuçlara erişebilmiştir. Elde edilen en iyi çözüm değerleri ayrıca literatürden GKA'ya dayanan bir açık kaynak kodlu çözücü ile de karşılaştırılmış ve bu algoritmaya yakın ya da daha başarılı sonuçlar elde edilmiştir. Ele alınan yaklaşımın süre ve çözüm kalitesi açısından gelişime açık yönlerinin olduğu aşikardır. Özellikle GSP çözümü için GUROBI

tarafından 21,600 saniyenin tümünün kullanıldığı problemler, düğümlerin kümelenecek şekilde dağılımı olduğu problemlerdir. Bu tarz veri setlerinde de daha başarılı olabilecek farklı özneliklerin kümeleme ve sınıflandırma algoritmaları olarak kullanılmalarıyla daha geliştirilmiş çözümler elde edilebilir. Bundan sonraki çalışmalar için, önerilen yöntemler zaman pencereli araç rotalama problemini içerecek özneliklerin dahil edilmesiyle, daha karmaşık araç rotalama problemlerini çözecek hale getirilebilir.

## Kaynaklar (References)

- Laporte G., Semet F., Classical heuristics for the vehicle routing problem, The Vehicle Routing Problem, Editör: P Toth., D Vigo., SIAM, Philadelphia, A.B.D., 109-128, 2002.
- Dantzig G.B., Ramser J.H., The truck dispatching problem, Management Science, 6 (1), 80-91, 1959.
- Clarke G., Wright J.W., Scheduling of vehicles from a depot to a number of delivery points, Operations Research, 12, 568-581, 1964.
- Laporte G., Osman H.I., Routing Problems: A bibliography, Annals of Operations Research, 227-262, 1995.
- Toth P., Vigo D., Vehicle Routing Problems, Methods and Applications, MOS-SIAM Series on Optimization, Second Edition, U.S.A., 2014.
- Eksioglu B., Vural A.V., Reisman A., The vehicle routing problem: A taxonomic review, Computers and Industrial Engineering, 57 (4), 1472-1483, 2009.
- Vidal T., Laporte G., Matl P., A concise guide to existing and emerging vehicle routing problem variants, European Journal of Operational Research, 286 (2), 401-416, 2020.
- Dantzig G., Fulkerson R., Johnson S., Solution of a large-scale traveling-salesman problem, Journal of the Operations Research Society of America, 2 (4), 393-410, 1954.
- Gendreau M., Laporte G., Musaraganyi C., Taillard E.D., A tabu search heuristic for the heterogeneous fleet vehicle routing problem, Computers and Operations Research, 26, 1153-1173, 1999.
- Oropeza A., Cruz-Chávez M., Cruz-Rosal Martín H., Bernal P., Abarca J.C., Unsupervised clustering method for the capacitated vehicle routing problem, Ninth Electronics, Robotics and Automotive Mechanics Conference, NW Washington, DC, U.S.A., November 19-23, 2012.
- Solomon M., Algorithms for the vehicle routing and scheduling problems with time window constraints, Operation Research, 35, 254-265, 1987.
- Sujoy R., Andrei S., Jean B., Mourad D., The multi-depot split-delivery vehicle routing problem: Model and solution algorithm, Knowledge-Based Systems, 71, 238-265, 2014.
- Toth P., Vigo D., The Vehicle Routing Problem, SIAM, Philadelphia, U.S.A., 2000.

14. Christofides N., Mingozzi A., Toth P., State-space relaxation procedures for the computation of bounds to routing problems, *Networks*, 11, 145-164, 1981.
15. Agarwal Y., Mathur K., Salkin H.M., A set-partitioning-based exact algorithm for the vehicle routing problem, *Networks*, 19, 731-749, 1989.
16. Braysy I., Gendreau M., Vehicle routing problem with time windows, part I: Route construction and local search algorithms, *Transportation Science*, 39 (1), 104-118, 2005.
17. Osman I.H., Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, 41, 421-451, 1993.
18. Glover F., Laguna M., *Tabu search, Modern Heuristic Techniques for Combinatorial Problems*, Editör: Reeves C.R., 70-150, Blackwell, Oxford, İngiltere, 1993.
19. Barbarosoglu G., Ozgur D., A tabu search algorithm for the vehicle routing problem, *Computers and Operations Research*, 26, 255-270, 1999.
20. Dorigo M., *Optimization, learning and natural algorithms*, Doktora Tezi, Elektronik Departmanı, Milano Politeknik Üniversitesi, Milano, İtalya, 1992.
21. Dorigo M., Ant colony optimization., *Scholarpedia*, 2 (3), 1461, 2008.
22. Guraksin A.M., Ozcan A., ACO-based approach for integrating product lifecycle management with MRO services in aviation industry, *Soft Computing*, 27, 337-361, 2023.
23. Kennedy J., Eberhart R., Particle swarm optimization, *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, Perth, Australia, 1942-1948, November 27-December 1, 1995.
24. Marinakis Y., Marinaki M., Migdalas A., Particle swarm optimization for the vehicle routing problem: A survey and a comparative analysis, Editör: Martı R., Pardalos P., Resende M., *Handbook of Heuristics*, Springer, Cham, 2018.
25. Shaw P., Using constraint programming and local search methods to solve vehicle routing problems, Editör: Maher M., Puget J.F., *Principles and Practice of Constraint Programming-CP98*, 417-431, Berlin Heidelberg, Springer, Germany, 1998.
26. Erdoğan G., An open source spreadsheet solver for vehicle routing problems, *Computers and Operations Research*, 84, 62-72, 2017.
27. Ropke S., Pisinger D., An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, 40, 455-472, 2006.
28. 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., Analytics and machine learning in vehicle routing research, *International Journal of Production Research*, 61 (3), 4-30, 2023.
29. Sun Y., Ernst A., Li X., Jake W., Generalization of machine learning for problem reduction: a case study on travelling salesman problems, *OR Spectrum*, 43, 607-633, 2021.
30. Blum C., Roli A., Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Computing Surveys*, 35 (3), 268-308, 2003.
31. Hopfield J.J., Tank D.W., "Neural" computation of decisions in optimization problems, *Biological Cybernetics*, 52 (3), 141-152, 1985.
32. Baltean-Lugoian R., Bonami P., Misener R., Tramontani A., Selecting cutting planes for quadratic semidefinite outer-approximation via trained neural networks, 2018.
33. He H., Daume III H., Eisner J.M., Learning to search in branch and bound algorithms in *Advances in neural information processing systems*, NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2, 3293-3301, Montreal, Canada, December 8-13, 2014.
34. Khalil E.B., Dilkina B., Nemhauser G. L., Ahmed S., Shao Y., Learning to run heuristics in tree search, *International Joint Conferences on Artificial Intelligence Organization*, 659-666, Melbourne, Australia, August 19-25, 2017.
35. Khalil E.B., Le Bodic P., Song L., Nemhauser G., Dilkina B., Learning to branch in mixed integer programming, 13. Association for the Advancement of Artificial Intelligence Conference, 30 (1), Phoenix, Arizona, February 12-17, 2016.
36. Lederman G., Rabe M.N., Lee E.A., Seshia S.A., Learning heuristics for automated reasoning through deep reinforcement learning, arXiv preprint arXiv:1807.08058, 2018.
37. Amizadeh S., Matuskevich S., Weimer M., Learning to solve circuit-sat: An unsupervised differentiable approach, 6. International Conference on Learning Representations Conference, Vancouver, Canada, April 30-May 3, 2018.
38. Beasley J.E., Route first—Cluster second methods for vehicle routing, *Omega*, 11 (4), 403-408, 1983.
39. Montoya J.A., Guéret C., Mendoza J.E., Villegas J.G., A route first cluster-second heuristic for the green vehicle routing problem, ROADEF 2014, Bordeaux, France, February 24-26, 2014.
40. Fisher M.L., Jaikumar R., A generalized assignment heuristic for vehicle routing, *Networks*, 11 (2), 109-124, 1981.
41. Dondo R., Cerdá J., A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows, *European Journal of Operational Research*, 176, 1478-1507, 2007.
42. Asis L.S., Eduardo C., Grossmann E.I., A MILP-based clustering strategy for integrating the operational management of crude oil supply, *Computers and Chemical Engineering*, 145, 2021.
43. Rautela A., Sharma S.K., Bhardwaj P., Distribution planning using capacitated clustering and vehicle routing problem, *Journal of Advances in Management Research*, 16 (5), 781-795, 2018.
44. Geetha S., Poonthalir G., Vanathi P.T., Improved K-Means algorithm for capacitated clustering problem, *INFOCOMP Journal of Computer Science*, 8 (4), 52-59, 2009.
45. Alesiani F., Ermiş G., Konstantinos G., Constrained clustering for the capacitated vehicle routing problem, *Applied Artificial Intelligence*, 36 (1), 2022.
46. Mostafa N., Eltwair A., Solving the heterogeneous capacitated vehicle routing problem using K-Means clustering and valid inequalities, *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Rabat, Morocco, April 11-13, 2017.
47. Duan L., Zhan Y., Hu H., Gong Y., Wei J., Zhang X., Xu Y., Efficiently solving the practical vehicle routing problem, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, CA, U.S.A., July 6-10, 2020.
48. Czuba P., Pierzchala D., Machine Learning methods for solving vehicle routing problems, 36th IBIMA Conference, 4-5 November, 2020.
49. Nazari M., Afshin O., Snyder L.V., Takac M., Reinforcement learning for solving the vehicle routing problem, 32. Conference on Neural Information Processing Systems, Montreal, Canada, 2-8 December, 2018.
50. Kool W., Herke van H., Max W., Attention, Learn to Solve Routing Problems!, 7. International Conference on Learning Representations, New Orleans, LA, 1-25, May 6-9, 2019.
51. Christofides N., Mingozzi A., Toth P., Sandi C., *Combinatorial Optimization*, Wiley, Chichester, U.K., 315-338, 1979.
52. Miller E., Tucker A.W., Zemlin R.A., Integer programming formulations and traveling salesman problems, *Journal of the ACM*, 7, 326-329, 1960.
53. Applegate D.L., Bixby R.E., Chvatal V., Cook W.J., *The traveling salesman problem: a computational study*, Princeton University Press, U.S.A., 2006.
54. Hand D.J., *Principles of data mining*, Drug safety, 30 (7), 621-622, 2007.
55. Wu X., Kumar V., Quinlan J.R., Ghosh J., Yang Q., Top 10 algorithms in data mining, *Knowledge and Information Systems*, 14 (1), 1-37, 2008.
56. Bock J.R., Gough D.A., Predicting protein-protein interactions from primary structure, *Bioinformatics*, 17 (5), 455-460, 2001.
57. Mangasarian O.L., Street W.N., Wolberg W.H., Breast cancer diagnosis and prognosis via linear programming, *Operations Research*, 43 (4), 570-577, 1995.
58. Min J.H., Lee Y.C., Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters, *Expert Systems with Applications*, 28 (4), 603-614, 2005.
59. Sammut C., Webb G.I., *Encyclopedia of Machine Learning*, 1031, Springer New York, U.S.A., 2011.
60. <http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/>, Erişim tarihi Aralık 19, 2022.
61. Anderson J.A., Logistic discrimination, *Handbook of Statistics*, 2, 169-191, 1982.
62. DeMaris A., A Tutorial in Logistic Regression, *Journal of Marriage and Family*, 57 (4), 956-968, 1995.
63. Fix E., Hodges L., Discriminatory Analysis, *Nonparametric Discrimination: Consistency Properties*, *International Statistical Review*, 57 (3), 238-247, 1989.
64. Çılgın C., Gökçen H., Gökçen Y., Sentiment Analysis of public sensitivity to COVID-19 vaccines on Twitter by majority voting classifier-based machine learning, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38 (2), 1093-1104, 2022.

65. Rochat Y., Taillard E., Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics*, 1 (1), 147-167, 1995.
66. Christophides N., Mingozzi A., Toth P., Sandi C., *Combinatorial Optimization*, John Wiley & Sons, Wiley-Interscience Series in Discrete Mathematics and Optimization, New York, U.S.A., 1979.
67. Golden B.L., Raghavan S., Wasil E.A., *The Vehicle Routing Problem: Latest Advances and New Challenges*, Operations Research/Computer Science Interfaces Series, Springer New York, NY, U.S.A., 2008.
68. Prescient&Strategic Intelligence, *On-Demand Logistics Market Report: By Vehicle Type (Light Commercial Vehicle, Medium/Heavy Commercial Vehicle), End Use (B2B, B2C), Application (E-Commerce, Industrial, Moving and Shifting, P2P Delivery) – Latest Trends, Recent Developments, and Demand Forecast Through 2030*, <https://www.psmarketresearch.com/market-analysis/on-demand-logistics-market>, January 2020, Erişim Tarihi Ocak 11, 2022.
69. Tek Ö.B., Ozgul E., *Modern Pazarlama İlkeleri: Uygulamalı Yönetimsel Yaklaşım, Birleşik Matbaacılık*, 3. Baskı, İzmir, 2010.

