

Makine Öğrenmesi Tabanlı Karınca Kolonisi Optimizasyonu Kullanarak Araç Rotalama

Vehicle Routing Using Machine Learning Based Ant Colony Optimization

Sinan KAMİLÇELEBİ¹ , Sümeyya İLKİN² , Suhap ŞAHİN¹ 

¹Bilgisayar Mühendisliği Bölümü, Kocaeli Üniversitesi, Kocaeli, Türkiye

²Bilişim Sistemleri Mühendisliği Bölümü, Kocaeli Üniversitesi, Kocaeli, Türkiye
(kamilcelebisinan@gmail.com, sumeyya.ilkin@kocaeli.edu.tr, suhahin@kocaeli.edu.tr)

Received: Sep.3, 2021

Accepted: Sep.16, 2021

Published: Oct.20, 2021

Özetçe— Araç rotalama problemleri birçok sektörde kullanılan çok önemli bir konu alanıdır. Araç rotalama problem çözümleri için ağırlıklı olarak kullanılan sezgisel ve metasezgisel yöntemlerin optimum bir çözüm sunmaması yeni yaklaşım ve algoritma geliştirme çalışmalarına yön vermektedir. Bu yüzden bu çalışma kapsamında zaman bağımlı araç rotalama problemine, yapay zekanın bir uygulaması olan makine öğrenmesi algoritmaları ile birlikte karınca kolonisi optimizasyonu uygulanarak oluşturulan rota planı için en uygun rotalar en kısa ve az maliyetle belirlenerek çözümler üretilmiştir. Üretilen bu çözümler farklı veri kümeleri ve sezgisel yöntemlerle de karşılaştırılmıştır. Çalışmada sezgisel optimizasyon problemlerini çözmek için makine öğrenimi ve karınca kolonisi optimizasyonu birleştirilerek literatüre gelişmiş yeni bir pratik yaklaşım sunulmuştur. Ayrıca araç rotalama problemlerinin çözümü için makine öğrenmesi ve karınca kolonisi optimizasyonunun bir arada kullanıldığı bu çalışma bu konu ile literatüre kazandırılmıştır.

Anahtar Kelimeler: Araç Rotalama, Gezgin Satıcı Problemi, Karınca Kolonisi Optimizasyonu, Kümeleme, Makine Öğrenmesi.

Abstract— Vehicle routing problems are a very important subject area used in many sectors. The fact that heuristic and metaheuristic methods, which are mainly used for vehicle routing problem solutions, do not provide an optimum solution, directs new approaches and algorithm development studies. Therefore, within the scope of this study, solutions were produced to the time-dependent vehicle routing problem by determining the most suitable routes for the route plan created by applying ant colony optimization together with machine learning algorithms, which is an application of artificial intelligence, by determining the shortest and least costly routes. These solutions were also compared with different datasets and heuristic methods. In the study, an advanced new practical approach is presented to the literature by combining machine learning and ant colony optimization to solve heuristic optimization problems. In addition, this study, in which machine learning and ant colony optimization are used together for the solution of vehicle routing problems, has been brought to the literature with this subject.

Keywords: Vehicle Routing, Traveling Salesman Problem, Ant Colony Optimization, Clustering, Machine Learning.

1. Giriş

Geçmişten günümüze, doğadaki canlılar bizlerin ve birçok araştırmacının ilgisini çekmiştir. Bu ilgi ve merakla zaman içerisinde doğadaki biyolojik canlılar yeni teknolojik sistemlerin geliştirilmesine öncü olmuştur. Mühendislikte ve günlük yaşantıda karşılaşılan çeşitli optimizasyon problemlerinin çözümünde bu biyolojik yapıli sistemlerin davranış ve iletişimleri kullanılmaktadır. Ayrıca bu sistemler günümüzde çok popüler olan robotik, yapay zekâ gibi birçok alanın da gelişmesini sağlamıştır.

Araştırmacılar, biyolojik yapıli sistemlerin davranışlarını inceleyerek başarılı sonuçlar veren çeşitli optimizasyon algoritmaları geliştirmişlerdir. Optimizasyon kelime anlamı olarak en iyileme anlamına gelir ve problemin en iyi çözümünü elde etmeyi amaçlar (Eröz ve Tanyıldızı, 2018). Geniş bir uygulama alanı olan optimizasyon yöntemleri hem bilimsel hem de endüstriyel alanda birçok teknik problemin çözümünde başarıyla kullanılır.

Literatürde ve gerçek hayatta üzerinde çok çalışılan ve ilgi çekici konulardan birisi de araç rotalama problemleridir. Birçok sektörün işletme maliyetlerinin azaltılmasında önemli rol oynayan araç rotalama problemlerinin çözümü için çeşitli sezgisel çözüm metotları önerilmiştir. Özellikle son yıllarda teknolojinin hızla gelişmesiyle birlikte büyük boyutlu optimizasyon problemlerinde daha iyi sonuç elde etmek için metasezgisel yöntemler üzerinde çalışmalar yoğunlaşmıştır. Karınca kolonisi optimizasyonu (KKO) da gerçek karıncalardan ilham alınarak geliştirilmiş ve farklı problemlere uygulanabilen önemli bir metasezgiseldir.

KKO problemlerinin çözümü için farklı yapıdaki metasezgisel yöntemlerin birleşmesiyle oluşan melez yöntemler de kullanılmaktadır. Metasezgisel optimizasyon algoritmalarının başarımı klasik (deterministik veya stokastik) yöntemlere göre daha başarılı olmasına rağmen bütün problemlerde aynı başarıyı gösterememektedir. Bu sebeple mevcut optimizasyon algoritmalarının iyileştirilmesi veya yeni metasezgisel optimizasyon algoritmalarının geliştirilme ihtiyacı ortaya çıkmaktadır (Eröz ve Tanyıldızı, 2018).

Literatürde çok geniş bir araştırma alanına sahip olan optimizasyon konusunda uzun yıllardan beri çalışmalar yapılmış olmasına rağmen her türlü duruma uyarlanabilecek optimum bir metot geliştirilememiştir. Bu çalışmanın amacı araç rotalama problemlerinin çözümü için daha önce yapılmamış bir yöntemle hem makine öğrenmesi hem de metasezgisel KKO'yu bir arada melez bir yapı içerisinde kullanarak literatür için yeni bir yaklaşım geliştirmektir. Çalışmada zaman bağımlı araç rotalama problemini çözmek için makine öğrenmesi tabanlı KKO'dan yararlanılmıştır.

Bu çalışmanın ikinci bölümünde araç rotalama problemiyle ilgili, üçüncü bölümde ise KKO yöntemi hakkında bilgi verilmiştir. Dördüncü bölümde ilk olarak KKO kullanılarak, daha sonra ise makine öğrenmesi tabanlı KKO kullanılarak yapılan araç rotalama işlemlerine ait yöntem ve bulgular paylaşılmıştır. Beşinci bölümde ise çalışmada elde edilen sonuçların farklı yapıdaki veri kümeleri ve sezgisel yöntemler ile karşılaştırılmasına ait sonuçlar üzerinde durulmuştur. Altıncı bölümde yapılan çalışma ile ilgili genel bir değerlendirme yapılmış, araştırma bulguları ve değerlendirmeler ışığında uygulama sonuçları paylaşılarak gelecek çalışmalar için önerilerde bulunulmuştur.

2. Araç Rotalama

Uygulama alanı fazla olan araç rotalama problemleri (ARP) uzun zamandan beri pek çok araştırmacının ilgisini çekmiştir. Rotalama işleminde, belirli bir harita üzerinde bulunan herhangi iki nokta arasındaki en optimum sonuç belirlenir. ARP, bir merkezde (depoda) bulunan araçların, belirli sayıdaki müşteri taleplerini yerine getirip tekrar aynı merkeze dönmesini sağlayacak en kısa rotaların bulunmasını amaçlar.

Rotalama problemlerinin en yaygın uygulaması gezgin satıcı problemidir. Gezgin satıcı probleminde (GSP) şehirler kümesinin başlangıç ve bitiş noktası aynı olmalıdır. Ayrıca şehirler en fazla bir kez işleme alınmalıdır. Şehirler arasındaki mesafeler kullanılarak hedef için en optimum rota

bulunur (Dikmen ve diğ., 2014). ARP, GSP'ye bazı özel kısıtlar eklenerek geliştirilmiştir. Bu özel kısıtlara göre günümüzde birçok ARP türü bulunmaktadır (Yücenur ve Çetin Demirel, 2011).

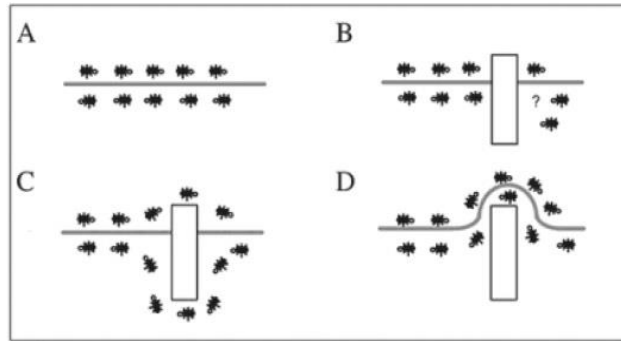
Genellikle ARP'nin çözümünde kesin, sezgisel ve metasezgisel olmak üzere üç farklı yöntem kullanılır. Kesin çözüm yöntemlerinde işlem sonucu mutlak ve uzun sürede bulunur. Ancak diğer yöntemler kesin çözümden daha az zamanda optimuma yaklaşık sonuçlar verir (Keskinürk ve diğ., 2015). Algoritma çalışma zamanları çok önemli olduğundan kesin çözüm yöntemlerini büyük boyutlu problemlerde kullanmak tavsiye edilmez (Karagul ve diğ., 2016). Araştırmalarda ARP çözümleri için çoğunlukla kullanılan sezgisel ve metasezgisel yöntemlerin ARP'nin farklı çeşitlerine optimum bir çözüm sunmaması yeni yaklaşım, model ve algoritma geliştirme çalışmalarına yön vermektedir. KKO'da GSP ve ARP çözümü için geliştirilen bu çalışmalardan birisidir.

3. Karınca Kolonisi Optimizasyonu

Tabiattaki canlı ve cansız varlıkları model olarak çeşitli aletler, makineler ve teknolojiler geliştirilmiştir. Günümüzde teknoloji ne kadar çok gelişse de doğadan model alma alışkanlığımız hiç değişmemiştir. Metasezgisel KKO'da, Dorigo (1992) tarafından doğadaki gerçek karıncaların gıda ararken en kısa yolu bulma kabiliyetlerinden esinlenerek geliştirilmiştir.

Günümüzde KKO'nun kullanım alanları oldukça fazladır. Karıncalar tabiatları gereği güzergahlarına kimyasal bir madde (feromon) bırakırlar. Bir güzergahtaki feromon miktarı, güzergahtan geçen karınca sayısına ve çevresel şartlara bağlı olarak ortamda artar veya azalır. Karıncalar bu artma veya azalma durumlarına göre hedef yol seçimine karar verirler. Bu karar verme mekanizmasına bağlı olarak ortamda bulunan feromon miktarı ile yol uzunluğu ters orantılıdır. KKO'nun bir metasezgisel olarak adlandırılmasındaki en önemli unsurda ortamdaki feromon miktarı ile başlangıç-hedef arasındaki uzaklığı birleştirerek kullanmasıdır (Pala ve Aksaraylı, 2018). Bu birleşim işlemi KKO metasezgisel yönteminin temelini oluşturmaktadır.

Şekil 1'de karıncaların yiyecek arama davranışları gösterilmektedir. Şekil 1'e göre, a) Başlangıç ve hedef güzergahı arası karıncaların feromon salgılaması (b) karınca güzergahı üzerine bir engel konulması (c) karıncaların güzergahlarına konulan engel ile oluşan yeni yolu keşfetmesi (d) karıncaların ortamdaki feromon miktarına göre en kısa yolu belirlemesi (Eren Şenaras ve İnanç, 2017).



Şekil 1. Karıncaların yiyecek arama davranışları

Karıncı kolonisinde karınca turu oluşturulurken popülasyondaki her bir karınca Eş. (1)'de verilen matematiksel ifadeyi kullanarak yol seçim işlemi gerçekleştirir (Eren Şenaras ve İnanç, 2017).

$$P_{i,j}^l = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^l} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ eğer } j \in N_i^l \quad (1)$$

P_{ij}^l : i noktasından j noktasına l karıncasının yol seçme olasılığı

τ_{ij} : İki nokta (i, j) arasındaki feromon değeri

η_{ij} : İki nokta (i, j) arasındaki sezgisel değeri

α : Feromon kuvvetlendirme oranı

β : Sezgisellik kuvvetlendirme oranı

N : Noktalar kümesi

Metasezgisel KKO'nun temelindeki koloni algoritmasına ait sözde kod Şekil 2'de gösterilmiştir.

```
Algorithm 1 Karınca Koloni Algoritması
1: procedure KKOMETASEZGISEL
2:   Parametreleri belirle.
3:   Feromon izlerini baslat.
4:   while maksimum iterasyon sayısı sağlanmadıysa do
5:     Karıncalar oluşturularak her dugume rastsal olarak yerleştirilir.
6:     Her bir karınca belirlenen parametrelere göre bir sonraki şehri
7:     -Esitlik (1)'deki denklemden verilen lokal arama olasılığına bağlı
8:     -olarak seçmek suretiyle rota oluşturularak turunu tamamlar.
9:     Her bir karınca tarafından kat edilen yolların uzunluğu (rota mesafesi)
10:    -hesaplanır.
11:    Hesaplanan rota mesafesine göre lokal feromon güncellemesi yapılır.
12:    -(Rota mesafesine göre yollardaki feromon değerlerini arttırma.)
13:    -(Feromon buharlastırma katsayısına göre bütün yollardaki feromon
14:    -değerlerini azaltma.)
15:    En iyi çözüm oluşturulur.
16:    En iyi çözüm için global feromon güncellemesi yapılır.
17:   close;
```

Şekil 2. Karınca koloni algoritması sözde kodu

Günümüzde birçok alanda başarıyla kullanılan karınca koloni algoritması (KKA), gezgin satıcı ve diğer optimizasyon problemlerinin çözümünde iyi sonuçlar vermiştir.

4. Uygulama

Bu bölümde ilk olarak KKO ile zaman bağımlı (pencereli) araç rotalama işlemi yapılmıştır. Daha sonra denetimli ve denetimsiz makine öğrenmesi algoritmalarıyla metasezgisel KKA için bir makine öğrenme sistemi tasarlanmıştır. Tasarlanan bu sistem için araç rotalama işlemi Python programlama dili temelinde çeşitli Python veri bilimi ve scikit-learn makine öğrenmesi kütüphaneleri kullanılarak yapılmıştır.

Çalışmada rotalama problemi olarak her alanda karşılaşılan ve önemli bir konu alanı olan zaman bağımlı araç rotalama problemi (ZBARP) kullanılmıştır. Uygulamamızdaki ZBARP'de bir rotada zaman cinsinden gidilebilecek mesafe temel kısıt olarak belirlenmiştir. Zamana bağlı diğer parametrelerimiz Tablo 1'de gösterilmiştir. Amacımız ise sınırlı bir rota süresi içerisinde hem gidilebilecek her yere gitmek hem de bunu en az maliyetle (zaman, uzunluk-km) yapmaktır. Zaman kısıtıyla birlikte tasarladığımız model, bize gidilebilecek en optimum rota için ne kadar süreye ihtiyacımız olduğu bilgisini oluşturduğu rotayla göstermiştir. Çalışmada kısıtımız zaman (dakika) olduğu için oluşturduğumuz matrisleri tahmini hıza bölerek mesafe matrisi, zaman matrisine dönüştürülmüştür.

Tablo 1. Araç Rotalamada Kullanılan Zaman Kısıt Parametreleri

Parametreler	Değer
Günlük zaman sınırı (dakika)	600
Ziyaret edilen yerlerde harcanacak zaman sınırı (dakika)	30
Ortalama araç hızı (m/min)	667

Not: (1 km/h = 16,67 m/min)

Uygulamayı gerçekleştirmek için bir rota senaryosu oluşturulmuştur. Çalışmadaki gerçek amaçlarımızdan birisi de olan bu senaryo, "Türkiye'nin Kocaeli ilinde yer alan turistik (müze, tema parkı, manzara), tarihi ve doğal alanlara belirlenen bir başlangıç noktasına göre ilk olarak sadece KKO kullanarak daha sonra ise denetimli ve denetimsiz makine öğrenmesi tabanlı KKO ile zaman bağımlı araç rotalama planının" yapılmasıdır. Bu rotalama işlemi için geliştirmiş olduğumuz yazılım ile dünyanın herhangi bir noktası için rota planı oluşturulabilir. Ve oluşturulmak istenilen rota planı farklı şekillerde dinamik olarak özelleştirilerek (farklı ağ çeşitlerinden veri çekilmesi gibi) uygulanabilir.

4.1. Karınca Kolonisi Optimizasyonu Kullanarak Araç Rotalama

Belirlediğimiz bölgenin (Kocaeli ili) detaylı yol bilgisi, açık kaynaklı bir harita platformu olan OpenStreetMap üzerinden çevrimiçi olarak temin edilmiştir. Oluşturduğumuz rota senaryosundaki hedeflere gitmek için web üzerinden Overpass API ile sorgu gönderilerek çalışma veri kümesi oluşturulmuştur. Bu veri kümesine göre Kocaeli ilinde turistik, tarihi ve doğal olmak üzere toplam 59+1 seyahat noktası bulunmaktadır. Ayrıca rotamız için "Kocaeli-Milli İrade Meydanı" başlangıç noktası olarak belirlenmiştir.

Çalışmamız için gerekli olan veri kümesi oluşturulduktan ve çeşitli veri ön işleme adımları yapıldıktan sonra Kocaeli il yollarının ve bu yolların kesişim bilgisi kullanılarak bir graf oluşturulmuştur. Oluşturduğumuz bu graftaki başlangıç ve hedef noktalarına ulaşmamızı sağlayacak en optimum yol Dijkstra algoritması ile tek tek yolların uzunlukları toplanarak hassas bir şekilde bulunmuştur. Bu işlemi gitmek istediğimiz her bir noktaya uyguladığımızda ise rota planımız için gerekli olan, $N \times N^*$ büyüklüğünde bir mesafe matrisi elde edilmiştir. (N: Gidilecek yer sayısı.)

KKO karar verirken oluşturduğumuz mesafe ve feromon matrisleri olmak üzere iki temel bilgiyi, Tablo 2'de belirtilen bizim belirlediğimiz temel algoritma parametrelerine göre birleştirerek kullanır. Tablo 2'deki KKO parametreleri, ilgili problemimize göre denemeler yapılarak en etkili sonuçlar veren değerler seçilerek belirlenmiştir. Çalışmamızda KKO karar mekanizması oluşturulduktan sonra, Şekil 2'de verilen KKA'ya ait sözde kod kullanılarak Tablo 1'deki zaman kısıtlarına göre en optimum rota ve zaman değerleri belirlendi ve elde edilen sonuçlar harita üzerinde Şekil 3'teki gibi görselleştirildi.



Şekil 3. KKO kullanılarak zaman bağımlı araç rotalama işlemine ait Kocaeli ili harita görseli

Tablo 2. Karınca Kolonisi Algoritma Parametreleri

Parametreler	Değer
İterasyon-Adım sayısı (i)	100
Karınca sayısı (k)	Şehir Sayısı (n)
Feromon kuvvetlendirme oranı (α)	1,00
Sezgisellik kuvvetlendirme oranı (β)	1,00
Feromon buharlaşma oranı (ρ -Rho)	0,05

KKO ile ZBARP uygulaması için buraya kadar yapılan tüm işlemler çalışmamızda kullandığımız farklı veri kümeleri ve yöntemlerinde hiçbir değişiklik göstermemektedir. Çalışmamızda ZBARP için KKO'daki karınca koloni sınıfına ek olarak ZBARP algoritma sınıfı yazılmıştır. ZBARP sınıfı, araç sürücüsü için en optimum değerleri belirleyerek rota ve zaman bilgilerini yazdırır. KKO, ZBARP sınıfını çağırarak işlem yaptığından bu çalışmadaki diğer tüm bölümlerde tekrardan ZBARP üzerinde durulmamıştır.

KKO algoritmamıza göre oluşturduğumuz Kocaeli ili zaman bağımlı araç rota planı için sonuçlarımız (F0) Tablo 3'te gösterilmiştir. Ayrıca bu bölümdeki uygulamamızı karşılaştırmak için aynı sistem parametreleriyle geliştirilen başka bir KKO çalışması da (FB- KKO Candidate List Strategy) test amaçlı olarak kullanılmıştır.

Tablo 3. Karınca Kolonisi Optimizasyonu Kullanarak Zaman Bağımlı Araç Rotalama Sonuçları

Şehir Sayısı	Araç Sayısı	Yöntem	Toplam Süre (Dakika)
60	1	F0	Dorigo ACO (Dorigo ve Birattari, 2011), (Dorigo ve Di Caro, 1999)
		FB	Başarsoft KKO CLS (Bajpai ve Yadav, 2015)

4.2. Makine Öğrenmesi Tabanlı Karınca Kolonisi Optimizasyonu Kullanarak Araç Rotalama (MLBACO)

Bu bölümde, Tablo 4'teki denetimli ve denetimsiz makine öğrenmesi yöntemleri daha önce belirleyerek kullandığımız zaman bağımlı araç rotalama planına ve parametrelerine göre KKO'ya sırasıyla uygulanmıştır.

Çalışmamızda Kocaeli ili için tam veri kümesinin çerçevesi rota planımıza ve seçtiğimiz coğrafi bölgeye bağlı olmak üzere (Düğüm Sayısı+1 (59+1), 5)'dir. Tablo 4'te yer verilen makine öğrenmesi yöntemlerini çalışmamıza uygulanmadan önce coğrafi veri kümesi üzerinde bazı veri ön işleme adımları yapılmıştır. Bu işlemler: JSON formatındaki coğrafi veri kümesinin parse işlemi ile ayıklanması ve anlamlı biçime dönüştürülmesi, veri kümesi hakkında bilgi edinme, tanımlayıcı istatistikler, veri bütünlüğü kontrolü, veri keşfi analizi, label encoder, veri içindeki gereksiz kolonların atılması, gerekli veri tipi dönüşüm işlemlerinin yapılmasıdır. Veri ön işleme adımından sonra her bir noktanın kartezyen koordinatlarına göre unique bir alana sahip yeni bir özellik çıkarımı (osmid) yapılmıştır. Özellik çıkarımından sonra koordinatlarda eğer farklı bir koordinat türünde (EUC 2D, ATT) veri varsa veri normalleştirme işlemi uygulanmıştır. Veri normalleştirme işleminin ardından özellik seçimi (y, x, name, osmid, distance) yapılmıştır. Özellik seçiminden sonra ise sırasıyla makine öğrenme algoritmaları KKO'ya entegre edilmiştir.

4.2.1 Denetimsiz Makine Öğrenme Algoritmalarının (F1-F4) KKO'ya Entegre Edilmesi

Özellik seçim işleminin ardından hazırlanan veri kümesi seçilen makine öğrenmesi algoritmasına girdi olarak gönderilip yeni oluşan coğrafi veri kümesi çıktı olarak alınmıştır. Daha sonra Şekil 2'deki KKA sözde kodu adım ikiden sonra KKA için NetworkX and OSMnx kütüphanelerinin yardımıyla mesafe matrisi ve bu mesafe matrisinden de feromon ve ortam bilgisi matrisi oluşturulmuştur. Oluşturulan bu matrisler, karınca ve koloni sınıflarında ardından bu sınıfları kullanacak olan ZBARP sınıfında NumPy kütüphanesi üzerinden kullanılmıştır. Makine öğrenmesi algoritmalarıyla yeni oluşturulan matrisler Şekil 2'deki KKA sözde kodu adım dokuzdaki rota mesafesinin hesaplanmasında kullanılmıştır. Her bir karınca en optimum sonucu makine öğrenmesi algoritma çıktıları ile yeni oluşan mesafe ve feromon matrislerini birleştirerek bulmuştur. Ayrıca kümeleme algoritmalarında veri kümemiz için optimum küme sayısı (k) dinamik olarak sistem parametreleriyle bulunmuştur.

4.2.2 Denetimli Makine Öğrenme Algoritmalarının (F5-F7) KKO'ya Entegre Edilmesi

Özellik seçim işleminin ardından ilk olarak coğrafi veri kümesinin %70'i eğitim ve %30'u ise test olarak bölümlenmiştir. İki nokta verildiğin de aralarındaki seyahat süresini ve mesafesini hesaplayabilmek için seçilen makine öğrenmesi algoritması ile eğitim kümesi için bir tahmin modeli oluşturulmuştur. Daha sonra bu eğitim modeli kullanılarak tahmini noktalar sıralamasına sahip yeni bir coğrafi veri kümesi elde edilmiştir. Yeni oluşan coğrafi veri kümesindeki tahmin sırasına göre coğrafi noktalar arasında her bir karıncanın bir turu ne kadar mesafede (zamanda) alacağını hesaplamak için bir döngü oluşturuldu ve skor elde edildi. Noktalar arasındaki seyahat süresi veya mesafesi için en düşük skora sahip karınca en optimum sonucu bulmaktadır. Şekil 2'deki KKA sözde kodu adım ikiden sonra yeni oluşan veri kümesindeki tahmin sırasına göre mesafe matrisi ve bu mesafe matrisinden de feromon ve ortam bilgisi matrisi hesaplanmıştır. Son olarak tahmin modeli ile oluşturulan bu matrisler Şekil 2'deki KKA sözde kodu adım dokuzdaki noktalar arasındaki rota mesafesinin tahmini olarak hesaplanmasında kullanılmıştır. Coğrafi noktalar arasındaki yapılan bu tahmini hesaplamalar makine öğrenmesi özellik seçimindeki etiketlerle oluşturduğumuz ayrı bir model verisine göre yapılmıştır. Sonuç olarak oluşturulan tahmin modeli ile karıncalar rota planı için en optimum sonucu bulmuştur.

Çalışmamızda Tablo 4'teki denetimsiz ve denetimli makine öğrenme algoritmalarını KKO ile bütünleştirirken algoritma iterasyon sonucunda global olarak en optimum sonucu bulan karıncanın rotası nihai çıktı olarak alınmıştır.

Tablo 4. Makine Öğrenmesi Tabanlı KKO Kullanarak Zaman Bağımlı Araç Rotalama Sonuçları

Şehir Sayısı	Araç Sayısı	Makine Öğrenmesi Yöntemleri	Toplam Süre (Dakika)	
60	1	F1	KMeans	1999,897
		F2	Hierarchical Clustering	2050,494
		F3	DBSCAN	1770,542
		F4	HDBSCAN	2006,675
		F5	KNeighborsClassifier (KNN)	1992,280
		F6	XGBoost	1405,176
		F7	LightGBM	1229,228

Tablo 4'te yer alan ve KKO ile bütünleştirilen makine öğrenmesi algoritmaları için sırasıyla model performans değerlendirilmesi yapılmıştır. Her bir model performans değerlendirme işleminde "Confusion Matrix" kullanılmıştır. Bu karışıklık (hata) matrisinde girdi değerleri olarak makine öğrenmesi algoritmalarından elde edilen tahmini değerler ile veri kümemize ve istatistiksel değerlere göre oluşturulan sentetik veriler kullanılmıştır.

Tablo 3'teki elde ettiğimiz sonuçlara göre Tablo 4 kıyaslandığında makine öğrenmesi başarılı bir şekilde toplam seyahat süresini azaltarak optimize etmiştir. KKO ile ZBARP için tasarladığımız makine öğrenme sistemine ait Tablo 5'te yer alan metrik performans analiz sonuçları incelendiğinde, denetimli bir öğrenme algoritması olan "LightGBM" en başarılı, denetimsiz "HDBSCAN" makine öğrenme yöntemi ise en başarısız algoritmadır.

Tablo 5. Makine Öğrenme Algoritmalarının Performans Sonuçları
(Tablo 4'deki Kocaeli Test Durumu için)

Makine Öğrenmesi Yöntemleri	Accuracy	Precision	F1-Score	Recall	MSE
F1	0,60	0,68	0,71	0,75	0,40
F2	0,52	0,66	0,61	0,57	0,48
F3	0,67	0,71	0,65	0,60	0,43
F4	0,57	0,67	0,38	0,40	0,53
F5	0,50	0,61	0,50	0,28	0,60
F6	0,74	0,62	0,69	0,72	0,05
F7	0,87	0,95	0,86	0,79	0,33

5. Çalışmanın Farklı Veri Kümesi ve Sezgisel Yöntemlerle Karşılaştırılması

Yapılan çalışmanın bu son bölümünde ise önerdiğimiz yeni yaklaşım başta olmak üzere Tablo 3 ile Tablo 4'deki tüm yöntemler farklı büyüklüklerdeki TSPLIB (TSP Library) (Reinelt, 1994) ve Solomon (Solomon, 1987) veri kümeleri üzerinde sırasıyla çalıştırılarak karşılaştırma işlemleri yapılmıştır. Ayrıca karşılaştırma işlemlerinde literatürde rotalama ve diğer problemler için kullanılan bazı sezgisel yöntemlere de yer verilmiştir. Yapılan karşılaştırma işlemlerinde 105 şehre kadar veri kümesine yer verilmesinin nedeni uygulamanın yapıldığı bilgisayar performansının düşük olmasıdır. Çalışma için yazılan algoritmalar daha büyük veri kümeleri üzerinde de çalışmaktadır.

5.1. TSP Veri Kümesi Kullanılarak Kıyaslamalı Sonuçlar

Çalışmamız, seçilen beş farklı TSP veri kümesi üzerine başarılı bir şekilde uygulanmıştır. Tablo 6'da makine öğrenmesi yöntemlerinden en başarılı sonuç veren algoritmaya (F7) yer verilmiştir. Tablo 6'daki "Optimum Çözüm" başlıklı sütunda verilen değerler TSPLIB tarafından bulunmuş sonuçlardır.

Tablo 6. TSPLIB Karşılaştırmalı Örneklerini Çözmek için
MLBACO Uygulamasına Göre Sonuçlar

TSP Test Örnekleri	Şehir Sayısı	Araç Sayısı	Optimum Çözüm (Km)	Önerilen Algoritma Çözümü (Km)	
				F0	F7
burma14	14	1	3323	2891	1964
ulysses22	22		7013	5983	4883
berlin52	52		7542	5086	4956
eil76	76		538	453	351
lin105	105		14379	12990	9268

Karşılaştırma işlemlerinde kullanılan her bir TSP test örneği için literatür araştırması yapılmıştır. Tablo 7-11'de önerdiğimiz yaklaşımla literatür taramasındaki karşılaştırmalı sonuçlara ait bilgilere yer verilmiştir.

Tablo 7. burma14 TSPLIB Örneği için Kıyaslamalı Test Sonuçları

TSP Test Durumu	Referans	Optimum Çözüm (Km)	Metot
burma14	[F7]	1964	Önerilen Algoritma (MLBACO)
	(Ratanavilisagul ve Pasaya, 2018)	3323	ACO+2Opt Algorithm
	(Ratanavilisagul, 2017)	3323	ACO Pheromone Mutation
	(Cheng ve diğ., 2016)	3087	PSO+3Opt Algorithm

Tablo 8. ulysses22 TSPLIB Örneği için Kıyaslamalı Test Sonuçları

TSP Test Durumu	Referans	Optimum Çözüm (Km)	Metot
ulysses22	[F7]	4883	Önerilen Algoritma (MLBACO)
	(Ratanavilisagul ve Pasaya, 2018)	7013	ACO+2Opt Algorithm
	(Ratanavilisagul, 2017)	7018,84	ACO Pheromone Mutation
	(Castillo ve diğ., 2015)	7018	Fuzzy Logic, ACO

Tablo 9. berlin52 TSPLIB Örneği için Kıyaslamalı Test Sonuçları

TSP Test Durumu	Referans	Optimum Çözüm (Km)	Metot
berlin52	[F7]	4956	Önerilen Algoritma (MLBACO)
	(Ebadinezhad, 2020)	7542	ACO, Dynamic Evaporation Strategy
	(Oonsrikaw ve Thammano, 2018)	7544,4	Simulated Annealing ACO
	(Necula ve diğ., 2017)	7545,83	Bee-Inspired Algorithms

Tablo 10. eil76 TSPLIB Örneği için Kıyaslamalı Test Sonuçları

TSP Test Durumu	Referans	Optimum Çözüm (Km)	Metot
eil76	[F7]	351	Önerilen Algoritma (MLBACO)
	(Ebadinezhad, 2020)	541,6	ACO, Dynamic Evaporation Strategy
	(Liu ve diğ., 2020)	538	ACO, Epsilon Greedy
	(Necula ve diğ., 2017)	541,2	Bee-Inspired Algorithms

Tablo 11. lin105 TSPLIB Örneği için Kıyaslamalı Test Sonuçları

TSP Test Durumu	Referans	Optimum Çözüm (Km)	Metot
lin105	[F7]	9268	Önerilen Algoritma (MLBACO)
	(Ebadinezhad, 2020)	14379	ACO, Dynamic Evaporation Strategy
	(Oonsrikaw ve Thammano, 2018)	14383	Simulated Annealing ACO
	(Necula ve diğ., 2017)	14380,23	Bee-Inspired Algorithms

TSP veri kümeleri üzerindeki karşılaştırmalı sonuçlar incelendiğinde önerilen algoritmamız (F7) diğer sezgisel yöntemlere göre daha başarılı sonuçlar vermiştir.

5.2. Solomon Veri Kümesi Kullanılarak Kıyaslamalı Sonuçlar

Bu bölümde çalışmamıza uygun olarak seçilen üç farklı Solomon veri kümesi kullanılmıştır. Tablo 12’de makine öğrenme yöntemlerinden en başarılı sonuç veren algoritmaya (F7) ait sonuçlar paylaşılmıştır.

Tablo 12. Solomon Karşılaştırmalı Örneklerini Çözmek için MLBACO Uygulamasına Göre Sonuçlar

Solomon Test Örnekleri	Şehir Sayısı	Araç Sayısı	Önerilen Algoritma Çözümü (Dakika)	
			F0	F7
C2	100	1	896,081	596,081
R2			885,794	584,229
RC2			927,658	624,349

Karşılaştırma işlemlerindeki her bir Solomon test örneği için de literatür araştırması yapılmıştır. Tablo 13-15’de önerdiğimiz yaklaşımla literatür taramasındaki karşılaştırmalı sonuçlara ait bilgilere yer verilmiştir.

Tablo 13. C2 Solomon Örneği için Kıyaslamalı Test Sonuçları

Solomon Test Durumu	Referans	Optimum Çözüm (Dakika)	Metot
C2	[F7]	596,081	Önerilen Algoritma (MLBACO)
	(Wang ve diğ., 2020)	1086,5	Improved ACO, MOSA
	(Zhang ve diğ., 2019)	591,56	Multi-Objective ACO
	(Ye ve diğ., 2018)	590	Multi-Type Ant System, NNS

Tablo 14. R2 Solomon Örneği için Kıyaslamalı Test Sonuçları

Solomon Test Durumu	Referans	Optimum Çözüm (Dakika)	Metot
R2	[F7]	584,229	Önerilen Algoritma (MLBACO)
	(Wang ve diğ., 2020)	1697,94	Improved ACO, MOSA
	(Liu ve diğ., 2020)	1313,7	Improved Ant Colony Algorithm
	(Zhang ve diğ., 2019)	1252,37	Multi-Objective ACO

Tablo 15. RC2 Solomon Örneği için Kıyaslamalı Test Sonuçları

Solomon Test Durumu	Referans	Optimum Çözüm (Dakika)	Metot
RC2	[F7]	624,349	Önerilen Algoritma (MLBACO)
	(Wang ve diğ., 2020)	1739,7	Improved ACO, MOSA
	(Liu ve diğ., 2020)	1488,6	Improved Ant Colony Algorithm
	(Zhang ve diğ., 2019)	1423,7	Multi-Objective ACO

5.3. KKO CLS (FB) Kullanılarak Kıyaslamalı Sonuçlar

Son olarak Tablo 16-17’de her bir TSPLIB ve Solomon veri kümeleri için önerdiğimiz algoritma (F7) ile Başarsoft KKO CLS (FB) yöntemi karşılaştırılmıştır. Bu karşılaştırma işlemleri için FB yöntemi

çalışma boyunca kullandığımız sekiz farklı veri kümesi (TSPLIB-Solomon) üzerine sırasıyla uygulanmıştır. Sonuçlar incelendiğinde sunduğumuz yaklaşımın en optimum değerleri başarılı bir şekilde verdiği görülmektedir.

Tablo 16. TSPLIB Karşılaştırma Örneklerinin KKO CLS Uygulamasına Göre Test Sonuçları

TSP Test Örnekleri	Şehir Sayısı	Optimum Çözüm (Km)		
		TSPLIB	Önerilen Algoritma [F7]	KKO CLS
burma14	14	3323	1964	3031
ulysses22	22	7013	4883	7278
berlin52	52	7542	4956	9937
eil76	76	538	351	641
lin105	105	14379	9268	16748

Tablo 17. Solomon Karşılaştırma Örneklerinin KKO CLS Uygulamasına Göre Test Sonuçları

Solomon Test Örnekleri	Şehir Sayısı	Optimum Çözüm (Km)	
		Önerilen Algoritma [F7]	KKO CLS
C2	100	596,081	675
R2		584,229	777
RC2		624,349	709

6. Sonuç ve Öneriler

Bu çalışmada ilk olarak araç rotalama ve KKO ile ilgili kavramsal ve ilişkisel bilgiler verilmiştir. Kavramsal çerçeve hakkında genel olarak bilgi verilmesinin ardından Kocaeli ili için oluşturulan rota planı çerçevesinde KKO ile ZBARP çözülmüştür. Daha sonra makine öğrenmesi yaklaşımıyla metasezgisel KKO bütünleştirilmiş ve zaman bağımlı dinamik hedef yapısına sahip araç rotalama işlemi yapılmıştır. Son bölümde ise bu çalışma ile sunulan yeni yaklaşıma ait sonuçlar farklı veri kümeleri ve sezgisel algoritmalar ile karşılaştırılmıştır.

Farklı yapıdaki veri kümeleri ve sezgisel yöntemler ile yapılan karşılaştırmalar neticesinde metasezgisel KKO'nun makine öğrenmesi teknikleri ile birlikte ele alınması ZBARP'nin çözümünde başarılı sonuçlar vermiştir. Çalışma birçok açıdan fayda sağlayacaktır. Ayrıca çeşitli şirket ya da diğer kurumlara daha uygun dağıtım rotası belirlenmesiyle maliyet kaybı önlenerek ciddi bir maliyet tasarrufu sağlanırken, kaza riski, çevre kirliliği, trafik sorunu gibi konularda da önemli katkıların olabileceği söylenebilir.

Sonuç olarak, tedarik zincirinden lojistiğe birçok alanda önemli bir karar alanı olan araç rotalama ile ilgili yapılan çalışmamızda KKO ve makine öğrenmesinin birlikte kullanılması ile sunduğumuz yaklaşım (MLBACO) veri kümesinin büyük olduğu problem setleri için iyi çözümler üretmiştir. Bu çalışma, konu alanı ile ilgili yapılacak gelecekteki benzer çalışmalara önemli ölçüde rehberlik edecektir. Ayrıca yapılacak ileriki çalışmalarda, GSP veya KKO çözümünde farklı yapay zekâ veya makine öğrenmesi teknikleri de bir arada kullanılarak yeni melez algoritmalar geliştirilebilir.

Kaynaklar

- Bajpai A., Yadav R. (2015) Ant Colony Optimization (ACO) for The Traveling Salesman Problem (TSP) Using Partitioning, *International Journal of Scientific & Technology Research*, 4(09): 376-381.
- Castillo O., Neyoy H., Soria J., et al. (2015) A New Approach for Dynamic Fuzzy Logic Parameter Tuning in Ant Colony Optimization and its Application in Fuzzy Control of a Mobile Robot, *Applied Soft Computing*, vol. 28, pp. 150-159.
- Cheng B., Lu H., Huang Y., Xu K. (2016) An Improved Particle Swarm Optimization Algorithm Based on Cauchy Operator and 3-Opt for TSP, *17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pp. 177-182.
- Dikmen H., Dikmen H., Elbir A., Ekşi Z., Çelik F. (2014) Gezgin Satıcı Probleminin Karınca Kolonisi ve Genetik Algoritmalarla Eniyilemesi ve Karşılaştırılması, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 18(1): 8-13.
- Dorigo M., Birattari M. (2011) Ant Colony Optimization, *In Encyclopedia of Machine Learning*, pp. 36-39, Boston, MA: Springer.
- Dorigo M., Di Caro G. (1999) Ant Colony Optimization: A New Meta-heuristic, *Congress on Evolutionary Computation-CEC 99*, vol. 2, pp. 1470-1477.
- Ebadinezhad S. (2020) DEACO: Adopting Dynamic Evaporation Strategy to Enhance ACO Algorithm for the Traveling Salesman Problem, *Engineering Applications of Artificial Intelligence*, vol. 92.
- Eren Şenaras A., İnanç Ş. (2017) GSP Çözümü için Karınca Kolonisi Optimizasyonu, *International Congress of Management Economy and Policy (ICOMEPE)*, İstanbul, Türkiye, 20-21 Mayıs, 58-67.
- Eröz E., Tanyıldızı E. (2018) Güncel Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması, *International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Türkiye.
- Karagul K., Tokat S., Aydemir E. (2016) Kapasite Kısıtlı Araç Rotalama Problemlerinde Başlangıç Rotalarının Kurulması için Yeni Bir Algoritma, *Mühendislik Bilimleri ve Tasarım Dergisi*, 4(3): 215-226.
- Keskintürk T., Topuk N., Özyeşil O. (2015) Araç Rotalama Problemleri ile Çözüm Yöntemlerinin Sınıflandırılması ve Bir Uygulama, *İşletme Bilimi Dergisi*, Cilt:3, Sayı:2.
- Liu C., Kou G., Zhou X., et al. (2020) Time-Dependent Vehicle Routing Problem with Time Windows of City Logistics with a Congestion Avoidance Approach, *Knowledge-Based Systems*, vol. 188.
- Liu Y., Cao B., Li H. (2020) Improving Ant Colony Optimization Algorithm with Epsilon Greedy and Levy Flight, *Complex & Intelligent Systems*, 7:1711-1722.
- Necula R., Breaban M., Raschip M. (2017) Tackling Dynamic Vehicle Routing Problem with Time Windows by means of Ant Colony System, *IEEE*.
- Oonsrikaw Y., Thammano A. (2018) Enhanced Ant Colony Optimization with Local Search, *17th International Conference on Computer and Information Science (ICIS)*, pp. 291-296.
- Pala O., Aksaraylı M. (2018) Çok Amaçlı Kapasite Kısıtlı Araç Rotalama Problemi Çözümünde Bir Karınca Kolonisi Optimizasyon Algoritması Yaklaşımı, *Alphanumeric Dergisi*, Cilt:6, Sayı:1.
- Ratanavilisagul C. (2017) Modified Ant Colony Optimization with Pheromone Mutation for Travelling Salesman Problem, *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 411-414.
- Ratanavilisagul C., Pasaya B. (2018) Modified Ant Colony Optimization with updating Pheromone by Leader and Re-initialization Pheromone for Travelling Salesman Problem, *International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, pp. 1-4.

- Reinelt G. (1994) *The Traveling Salesman, Computational Solutions for TSP Applications*, Springer-Verlag Berlin Heidelberg.
- Solomon M. M. (1987) Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows Constraints, *Operations Research*, vol. 35, No. 2, pp. 254-265.
- Ye D., Wanhong Z., Hongwei L., Yonghui Z. (2018) Multi-Type Ant System Algorithm for the Time Dependent Vehicle Routing Problem with Time Windows, *Journal of Systems Engineering and Electronics (BIAI)*, vol. 29, pp. 625–638.
- Yücenur G. N., Çetin Demirel N. (2011) Çok Depolu Araç Rotalama Problemlerinin Çözümü için Genetik Algoritma ve Karınca Kolonisi Optimizasyonundan Oluşan Melez Algoritma Tasarımı, *Mühendislik ve Fen Bilimleri Dergisi*, 340-350.
- Zhang H., Zhang Q., Ma L., et al. (2019) A Hybrid Ant Colony Optimization Algorithm for a Multi-Objective Vehicle Routing Problem with Flexible Time Windows, *Information Sciences*, vol. 490, pp. 166-190.
- Wang Y., Wang L., Chen G., et al. (2020) An Improved Ant Colony Optimization Algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice, *Swarm and Evolutionary Computation (SWEVO)*, vol. 55.