

Proje Ana Alanı : Yazılım

Proje Tematik Alanı : Akıllı Ulaşım Sistemleri

Proje Adı (Başlığı) : Elektrikli Araçların Rotalanmasında Genetik Algoritma Temelli Yeni Bir Yaklaşım Geliştirilmesi

Özet

Elektrikli araçların kullanımı her geçen gün yaygınlaşmaktadır. Buna paralel olarak bireysel kullanıma özel elektrikli araçlar için rotalama ihtiyacı ortaya çıkmaktadır. Fosil yakıtlı araçlar için kullanılan rotalama çözümleri elektrikli araçların kısa menzile sahip olması ve şarj istasyonlarının yaygın olmayışından dolayı elektrikli araçlar için uygun çözüm sunmamaktadır. Bu çalışmada bireysel kullanım için geliştirilen elektrikli araçların nispeten kısa menzil ve yetersiz şarj istasyonları kısıtları göz önüne alınarak gezgin satıcı problemine dayalı genetik algoritma kullanılarak bir model geliştirilmiştir. Problem durumuna özgü uygunluk fonksiyonu ve birey seçilimi belirlenmiştir.

Geliştirilen modelin test edilmesi için bazılarında şarj istasyonu bulunan bazılarında bulunmayan 13 şehirden oluşan bir şehir ağı ve şehirler arasındaki mesafelerin yer aldığı uzaklık matrisi kullanılmıştır. Modelin çalışma zamanı açısından performansı brute-force yaklaşımı ile karşılaştırılmıştır. İkinci olarak popülasyondaki birey sayısı ve nesil sayısına bağlı model performansı değerlendirilmiştir. Çalışma zamanı açısından benzer çalışmalarla uyumlu sonuçlar gözlemlenmiştir. Genetik algoritma parametrelerinin model üzerindeki etkilerini değerlendirmek için yapılan çalışmada popülasyondaki birey sayısının en önemli parametre olduğu sonucuna ulaşılmıştır.

Anahtar kelimeler: elektrikli araç, rotalama, genetik algoritma

Amaç

Araç rotalama problemleri (ARP) hakkında birçok çalışma bulunmaktadır. Son yıllarda elektrikli araçların yaygınlaşmasına paralel olarak az sayıda elektrikli araç rotalama problemlerinin (EARP) çözümüne yönelik çalışmalar da yapıldığı görülmektedir. Ancak yapılan çalışmalarda kargo taşımacılığı üzerine yoğunlaşıldığı ve bu sektörün ihtiyaçlarına ve doğasına göre çözümler üretildiği, bireysel araçlara yönelik çalışmaların yeterince yapılmadığı anlaşılmaktadır.

Bu çalışmanın amacı, klasik deterministik algoritmalarının uzun çalışma zamanı ve çok parametrelili problemlerde verimsizliğini göz önünde bulundurarak bu kısıtları aşabilecek meta-sezgisel algoritmalarla geliştirilmiş elektrikli bireysel kullanım amaçlı araçlar için yeni bir araç rotalama yaklaşımı geliştirmektir.

Yapılan benzer çalışmalarda olmayan elektrik şarj istasyonlarının konumlarının çalışmamızda rotalama hesaplarına dahil edilmesi ile alana katkı sağlaması da ayrıca amaçlanmıştır.

Giriş

Son zamanlarda elektrikli araçların kullanımı artan bir ivmeyle hızlanmaktadır. 2030 yılı sonuna kadar dünyadaki elektrikli otomobil sayısı 10 kat artması ve 2030 yılına gelindiğinde dünyada satılan araçların yüzde 60'ından fazlası elektrikli olması beklenmektedir[1]. Bu artan popülerlikle elektrikli araçların etkin bir şekilde rotalanması ve enerji ihtiyaçlarının optimize edilmesi, sürdürülebilir ulaşımın temel bir unsuru haline gelmiştir.

Elektrikli araçların fosil yakıtlı araçlarla kıyaslandığında mevcut pil teknolojilerinden kaynaklanan menzil ve şarj istasyonlarının yaygın bulunmamasından kaynaklanan şarj etme sorunu vardır. Bundan dolayı şehirler arası uzun mesafelerde elektrikli araç kullanımı kısıtlıdır. Bu bağlamda, elektrikli araçların etkin rota planlaması, hem enerji verimliliğini artırmak hem de sürücülerin seyahat deneyimini iyileştirmek adına kritik bir öneme sahiptir.

EAR problemi çözümlenirken fosil yakıtlı araçlarda göz ardı edilen menzil ve akaryakıt

(şarj) istasyon lokasyonunun denkleme dahil edilmesi gerekmektedir. Elektrikli araçların rota optimizasyonu, çevresel etkilerin azaltılması ve enerji kaynaklarının verimli kullanımı, uzun menzilli seyahatler açısından önemli bir adımdır.

Problem Durumu:

Rotalama problemleri, klasik (doğrusal veya konveks olmayan) deterministik programlama algoritmaları ile çözülmesi zor olan ayrık bir problemdir. Bunun nedeni, bu algoritmaların aşırı hesaplama süresinden dolayı kısıtlı oluşudur. Klasik yöntemlerin başında brute-force yöntemi gelmektedir. Bu yöntem uğranılması gereken şehir sayısına göre üssel artan olasılıkta çözüm önerilerini tek tek değerlendirir ve zaman karmaşıklığı $O(n!)$ 'dir. Bundan dolayı bu yöntem NP-hard problem olarak değerlendirilir[2].

Yukarıdaki sebeplerden dolayı rotalama problemlerinin meta-sezgisel algoritmalarla çözümünü hedefleyen çok sayıda çalışma vardır. Bu çalışmalar incelendiğinde meta-sezgisel algoritmaların rotalama problemlerinin çözümünde son derece etkin olduğu görülmektedir[3]. Bu yüksek çözüm kapasitesinden dolayı özellikle lojistik alanında rota optimizasyonu gibi çok parametrelili problemlerin üzerinde çalışmaların yoğunlaştığı görülmektedir.

Fosil yakıtlı bireysel araçların rotalama problemleri “en kısa yol” veya “en hızlı yol” dışında bir kısıtlaması olmayan bir problem türüdür. Bu konu başlığında yapılan çalışmalarda akaryakıt istasyonlarının her yerde bulunduğu ve bir depo yakıtla binlerce kilometre gidebildiği varsayılmıştır. Bunun aksine elektrikli araç rotalama probleminde en önemli nokta şarj istasyonlarının yeterince yaygın olmaması ve araç menzillerinin fosil yakıtlara göre çok kısa olmasıdır. Bu kısıtlı yönler elektrikli araç rotalama problemlerinde yeni yaklaşımlar geliştirilmesi ihtiyacını ortaya çıkarmaktadır. Uzun mesafeli yolculuklar için elektrikli araç kullanıcılarının şarj problemi yaşamadan en kısa yoldan hedeflerine ulaşabilecekleri rotaların belirlenmesi bir ihtiyaçtır.

Benzer Çalışmalar:

Dantzig ve Ramser tarafından 1959 senesinde Araç Rotalama Problemi (ARP) ilk defa çalışılmıştır. Araştırmacılar bu çalışmalarında bir kamyon filosunu lojistikte kullanımında en kısa yolu bulmaya çalışmışlardır[4]. Sonraki çalışmalarda ARP çalışmalarında fosil yakıtlı araçların rota uzunluğu veya toplam sürüş maliyetleri veya sürüş süresini en aza indirmek hedeflenmiştir. Bunun yanında araç yükleme sınırı, araç tipi, zaman penceresi, depo durumu, müşteri durumu gibi kısıtlar ele alınmıştır.[5][6][7]

Sonraki çalışmalarda ARP problemlerinin çözümünde optimizasyon algoritmalarının kullanıldığı görülmektedir. 2011 senesinde Brezina ve Cickova (Brezina ve Cickova, 2011) tarafından karınca kolonisi optimizasyonu kullanılarak GSP'nin çözümüne odaklı bir çalışma gerçekleştirilmiştir[8]. Cevre ve arkadaşları (Cevre vd., 2007) tarafından yapılan çalışmada ise genetik algoritma kullanılarak GSP optimize edilerek internet üzerinde görselleştirilmesi sağlanmıştır[9]. Alaykırın ve Engin (Alaykırın ve Engin, 2005) ise karınca kolonileri metasezgiselini kullanarak GSP üzerine bir uygulama gerçekleştirmişlerdir[10]. Shang ve arkadaşları (Shang vd., 2007) tarafından gerçekleştirilen çalışmada birliktelik kuralları tabanlı karınca koloni optimizasyonu algoritması geliştirilerek GSP'ye çözüm geliştirilmiştir[11]. Özkan ve arkadaşları (Özkan vd., 2008) tarafından yapılan çalışmada ise GSP üzerinde genetik algoritmaları yerel arama sezgileri ile birleştiren melez bir çözüm geliştirilmiştir[12].

Son yıllardaki çalışmalarda ise elektrikli araçların rotalama problemlerinin incelendiği görülmüştür. Elektrikli araçların rota optimizasyonu nispeten yeni bir araştırma konusudur. Erdoğan ve arkadaşları (Erdoğan vd., 2012) tarafından yapılan çalışmada alternatif enerji yakıtlı araçların rota optimizasyon modeli sunulmuş olup, alternatif enerji kaynağı ve maksimum yakıt kapasitesi sınırı dikkate alınarak seyahat mesafesini en aza indirmeyi amaçlamaktadır[13]. Schneider ve arkadaşları (Schneider vd., 2014) tarafından yapılan çalışmada ise şarj

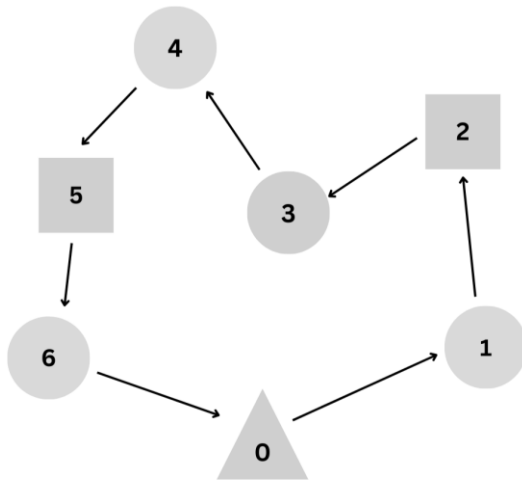
istasyonunun tanıtıldığı ve batarya kapasitesi (yani, bir tam şarjdan sonra maksimum seyahat mesafesi) gibi bazı kısıtlamaların geleneksel araçlara ek olarak dikkate alındığı elektrikli araçların rota optimizasyonu incelenmiştir[14]. Lin ve arkadaşları (Lin vd., 2014) ARP için maliyet ve CO₂ emisyonlarını en aza indirerek genetik algoritma esaslı bir yöntem önermiştir[15]. Costa ve arkadaşları(Costa vd.2018) benzer bir çalışmayı da 2018’de yapmıştır. Bu çalışmada CO₂ emisyonunu en aza indirmek için benzer şekilde genetik algoritmayı kullanmışlardır[16]. Elektrikli araç rotalama problemlerinde bu çalışmaya yakın bir araştırma Conrad ve Figliozi tarafından yapılmıştır ancak bu çalışmada elektrik şarj istasyonlarının her noktada bulunduğu varsayılmıştır. Bu varsayım günümüz itibariyle pratikte geçersiz bir varsayımdır[17].

Problem Tanımı:

EARP tanımlanan noktalar üzerinde aşağıdaki varsayımlar altında en küçük maliyetli rotaların tespiti problemidir:

- Araç her noktaya kesinlikle bir kez uğramalı.
- Araç rotası başlangıç noktasından başlamalı ve tekrar aynı noktada son bulmalı.
- Şarj istasyonlarının bulunduğu noktalarda araç bataryası şarj edilmeli.
- Araç menzili başlangıç noktasından ilk ziyaret edilecek şarj istasyonunun bulunduğu noktaya ve şarj istasyonun bulunduğu noktadan diğer şarj istasyonunun bulunduğu noktaya ve son şarj istasyonunda başlangıç noktasına kadar olan mesafeye eşit veya büyük olmalıdır.

Şekil 1’de yukarıdaki EARP tanımı için bir rotalama planı verilmiştir. Daireler ziyaret edilecek noktaları, kareler şarj istasyonu bulunan noktaları ve üçgen başlangıç/bitiş noktasını göstermektedir.



Şekil 1: EARP için örnek rotalama

Modelin matematiksel gösterimi için Kuyzu tarafından önerilen benzer bir modelin EARP için düzenlenmiş hali kullanılmıştır[18]. Elektrikli Araç Rotalama Problemi (EARP) notasyonel

olarak şu şekilde tanımlanabilir.

Kümeler

V: tüm noktalar,

0: başlangıç ve bitiş noktası, $0 \in V$

F: şarj istasyonu olan noktalar, $F \subseteq V$

A: noktalar arasında mesafeler, $\{(i,j): i,j \in V, i \neq j\}$

C: bütün noktaların ziyaret edildiği rotalar

X: iki nokta arasındaki yol,

Parametreler

m: araç bataryasına bağlı menzil

F_c : C rotasının uzunluğu $\forall r \in R$

Karar değişkenleri

$$x_{ij} = \begin{cases} 1 & \text{eğer araç } i \text{ noktasından } j \text{ noktasına geçiyorsa } (i, j \in V) \\ 0 & \text{diğer durumlarda} \end{cases}$$

$$F_i = \begin{cases} 1 & \text{eğer noktada şarj istasyonu varsa } (i \in V) \\ 0 & \text{diğer durumlarda} \end{cases}$$

Amaç Fonksiyonu

$$\text{En küçük } F_c = \sum_{i,j \in V, i \neq j} A_{ij} X_{ij} \quad (1)$$

Kısıtlar

$$\sum_{i,j \in V, i \neq j} X_{ij} = 1 \quad i, j \in V \quad (2)$$

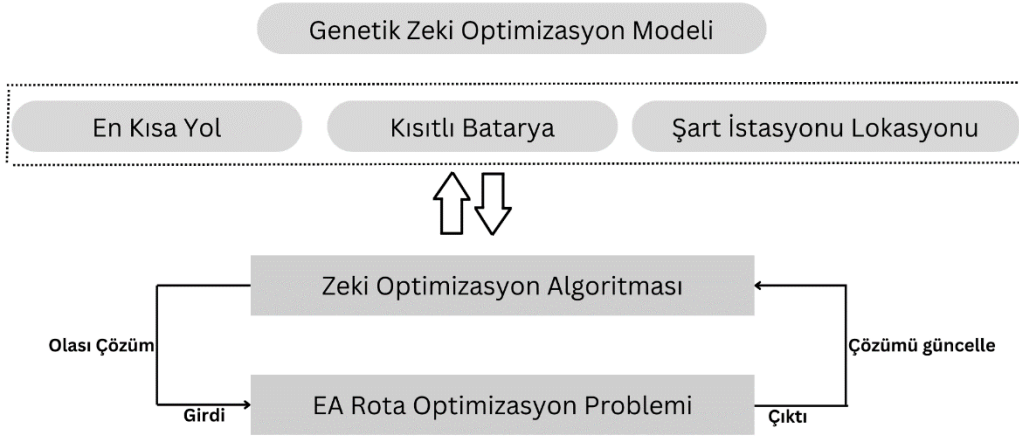
$$\sum_{i,j \in V, i \neq j} A_{ij} X_{ij} - \sum_{j \in V} F_j \leq m \quad i, j \in V \quad (3)$$

Matematiksel modelde amaç fonksiyonu (1) en kısa rotayı göstermektedir. Kısıtlar (2) her noktanın bir kez ziyaret edilmesi gerektiğini, (3) rota parçalarının toplamının araç menziline büyük olamayacağını göstermektedir.

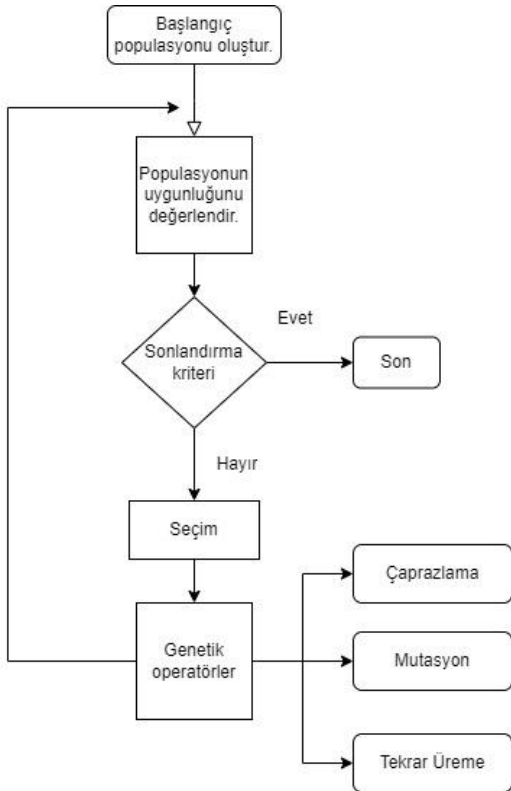
Yöntem

Bu çalışma elektrikli araç rotalama problemi için şarj istasyonlarının lokasyonunu ve elektrikli aracın menzili gibi kısıtları göz önüne alan, genetik algoritmayla geliştirilmiş, gezgin satıcı problemi temelli, en kısa rotayı bulak için bir çözüm modeli önermektedir.

Önerilen model için 13 düğümlü şehir şebekesi üzerinde vaka çalışması yapılmıştır. Bu modelde bazı şehirlerde şarj istasyonu bulunduğunu bazı şehirlerde ise şarj istasyonu bulunmadığını varsayılmıştır. Bu model ile Şekil 2’de gösterildiği gibi araç rotalama problemi, genetik algoritma ile çözülmektedir. Genetik algoritmada rastsal olarak seçilen rotalar GA algoritmalar ile eniyilenmektedir.



Şekil 2: EA için genetik algoritma rota optimizasyon yapısı



Şekil 3: Genetik algoritmaya ait temel çalışma akış diyagramı

Genetik algoritmanın temel çalışma prensibini gösteren akış diyagramı Şekil 3’de

verilmiştir.

Genetik Algoritma Parametreleri

Birey: Bu çalışmada olası çözüm rotalarının her biri birey olarak belirlenmiştir. Örnek birey "0 → 1 → 2 → 3" gibi bir dizidir.

Gen: Bir bireyi oluşturan ve bir şehri temsil eden her bir sayı gendir. Çalışmamızda bir birey toplam şehir sayısı kadar genden oluşmaktadır. Gezgin satıcı problemi gereği bir bireyde bulunan genler tekrarsızdır.

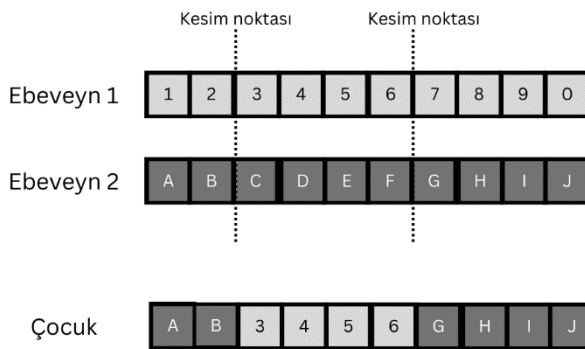
Birey Sayısı: Oluşturulan popülasyonda kaç birey bulunacağını belirleyen parametredir. Bu çalışmada popülasyonu oluşturacak birey sayısı 200 olarak belirlenmiştir. Bu sayı çalışma esnasında örnek problemlerin çözümünde gözlemlenen sonuçlar dikkate alınarak sezgisel olarak belirlenmiştir.

Uygunluk Değerinin Belirlenmesi: Önerilen model en kısa mesafeli rotayı belirlerken aynı zamanda bu rotadaki şarj istasyonları arasındaki mesafenin aracın menzilinden küçük olma durumunu denetler. Eğer menzil şarj istasyonları arasındaki mesafeden küçükse bu rota için 100000 gibi büyük bir sayı döndürür. Eğer menzil şarj istasyonları arasındaki mesafeden büyükse söz konusu rotanın mesafe matrisine göre mesafesini döndürür.

Birey Seçim Yöntemi: Bireylerin çaprazlama için hangi yönteme göre seçileceğini belirleyen parametredir. Çalışmamızda uygunluk değeri belirlenmiş bireylerden seçkinler yöntemi ve rulet seçim yöntemi hibrit olarak kullanılmıştır.

Yeni Nesil Seçimi: Oluşturulacak yeni nesil şu şekilde tespit edilir: Popülasyon uygunluk değerine göre sıralanır. Popülasyondaki birey sayısı ile seçkinler katsayısı çarpılarak seçkin birey sayısı bulunur. Oluşturulacak yeni nesle bulunan seçkin bireyler atanır. Geriye kalan bireyler çaprazlama işlemi sonucunda oluşturulacak bireylerle tamamlanır.

Çaprazlama: Yeni neslin değişik gen yapısına sahip olması için çaprazlama işlemi yapılır. Çalışmamızda literatürde "iki nokta çaprazlama" olarak adlandırılan Şekil 4'de gösterilen yöntem kullanılmıştır. Bu yönteme göre ebeveyn bireyler rastsal olarak belirlenen iki noktadan üç parçaya bölünmüştür. Yeni bir birey oluşturmak için 1. ebeveynin ikinci parçası alınır ve bu parçanın önüne 2. ebeveynin ilk parçası, arkasına ise 2. ebeveynin üçüncü parçası eklenir. Problemin doğası gereği bir kromozomda her genden sadece 1 tane bulunabileceğinden parça eklemelerde önceden var olan gen kontrolü yapılır.

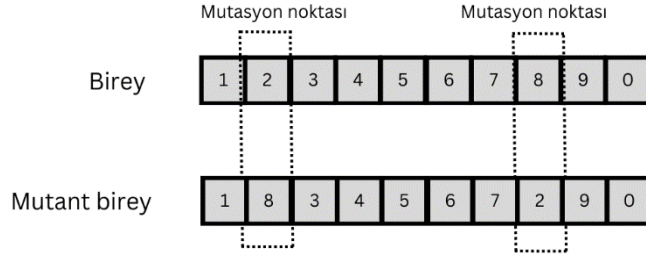


Şekil 4: İki noktalı çaprazlama örnek gösterimi

Eğer bu gen hali hazırda yeni bireyde varsa gen eklenmeyerek sonraki gene geçilir. Yeni neslin oluşturulması için çaprazlama ile belirlenecek bireyler şu şekilde oluşturulur: Popülasyondan uygunluk değeri sıralamasına göre sondan seçkin birey sayısı kadar birey

çıkarılır. Geriye kalan bireyler ikili gruplara ayrılarak çaprazlamaya tabii tutulur.

Mutasyon: Önerilen modelde iki nokta mutasyonu kullanılmıştır. İki nokta mutasyonunda bireyin genetik kodundaki iki belirli noktadaki gen değerleri birbirleriyle değiştirilmiştir. Bu, genetik materyalin belirli bir aralığındaki genleri değiştirmek için kullanılmıştır.



Şekil 5: İki noktalı mutasyon örneği gösterimi

Proje İş-Zaman Çizelgesi

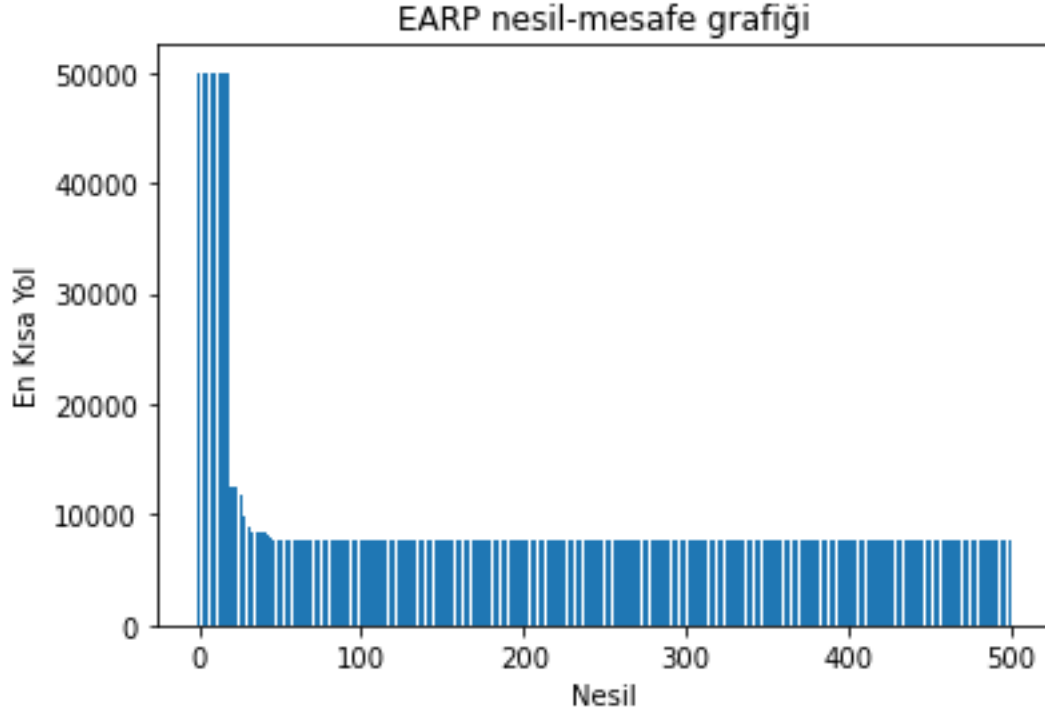
AYLAR						
İşin Tanımı	Ağustos	Eylül	Ekim	Kasım	Aralık	Ocak
Literatür Taraması	X	X				
Yöntem Seçimi		X	X			
Verilerin Toplanması ve Analizi			X	X	X	
Proje Raporu Yazımı					X	X

Bulgular

Geliştirilen model python programlama dilinde kodlanmıştır. Bütün sayısal analizler Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz işlemcili, 8,00 GB RAMli, NVIDIA GeForce GTX 1050 ekran kartı olan bir bilgisayarda çalıştırılmış ve oluşan sonuçlar değerlendirilmiştir.

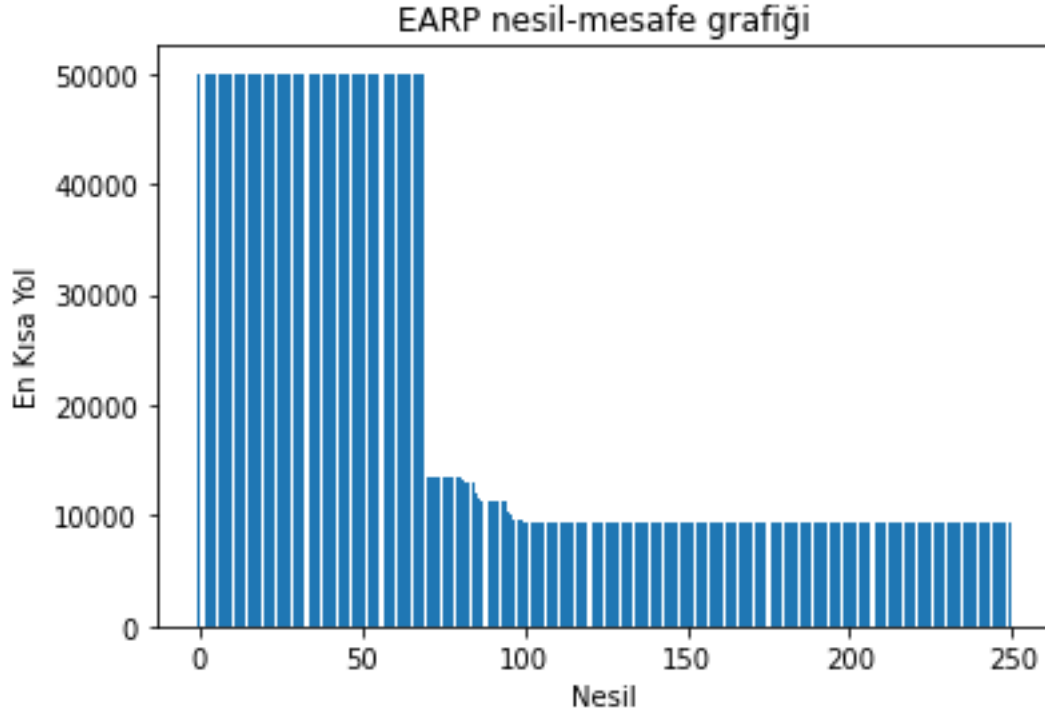
Örnek şehir ağı için en kısa rota brute-force ve genetik algoritma yöntemiyle bulunmuştur. Bu aşamada şarj istasyonu işleme dahil edilmemiştir. Burada iki algoritma çalışma zamanı açısından karşılaştırılmıştır. Çalışma sonrasında örnek şehir ağı için brute-force yöntemiyle sonuca yaklaşık 13 dakikada ulaşılmış, genetik algoritma ile yakın sonuç 15 saniyede bulunmuştur.

İkinci aşamada genetik algoritma parametrelerinin etkileri incelenmiştir. Geliştirilen modelin etkinliğinin tespiti için popülasyondaki birey sayıları sabit tutularak nesil sayıları değiştirilerek modelin performansı analiz edilmiş, diğer taraftan nesil sayıları sabit tutularak popülasyondaki birey sayıları değiştirilmiştir. Elde edilen veriler aşağıdaki grafiklerde gösterilmiştir.



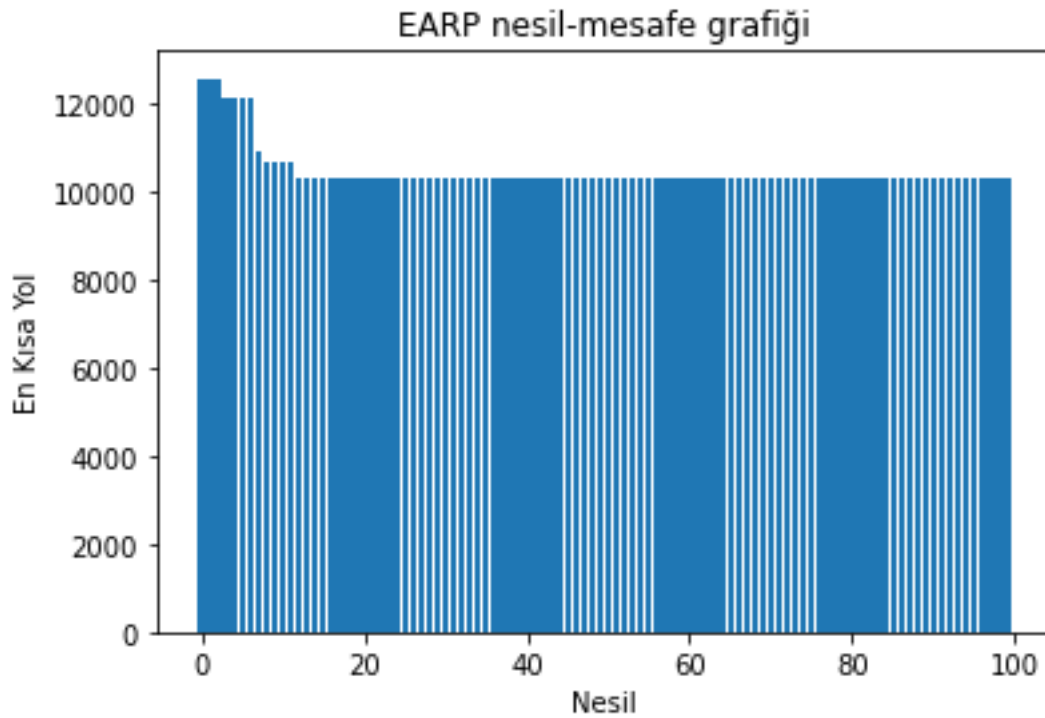
Şekil 6: 30 birey ve 500 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 1: Modeli test etmek için 30 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 500 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 6’da görülmektedir. 50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



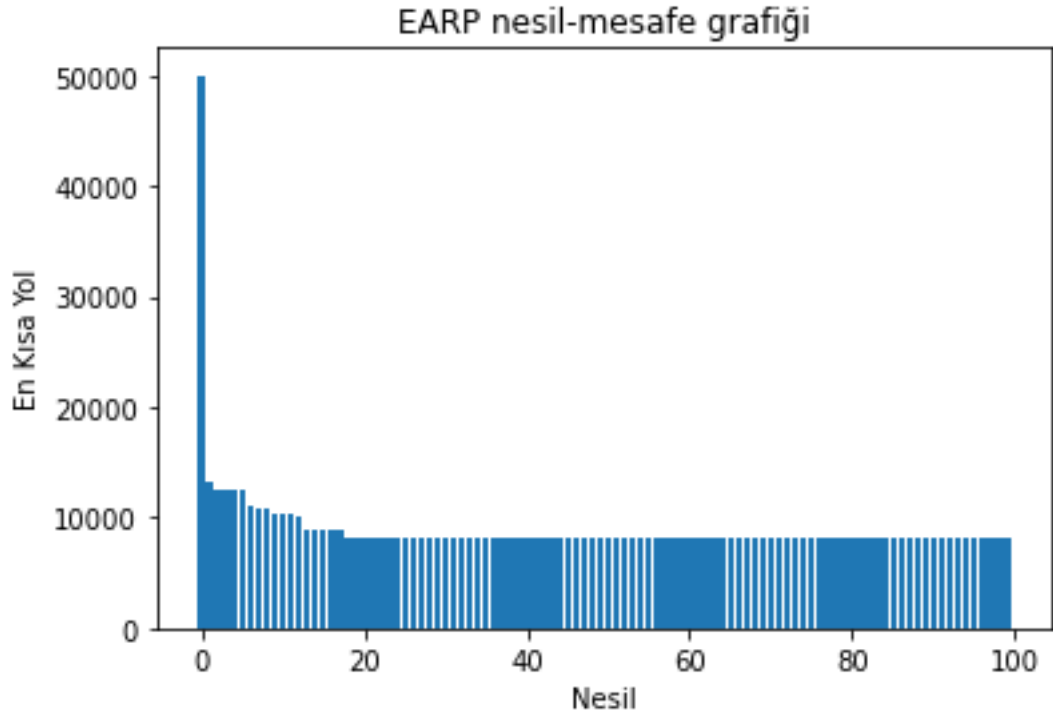
Şekil 7: 30 birey ve 250 nesi parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 2: Modeli test etmek için 30 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 250 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 6’ta görülmektedir. 75. nesle kadar en kısa yolun değişmediği 75. nesilden sonra hızla düştüğü 100. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



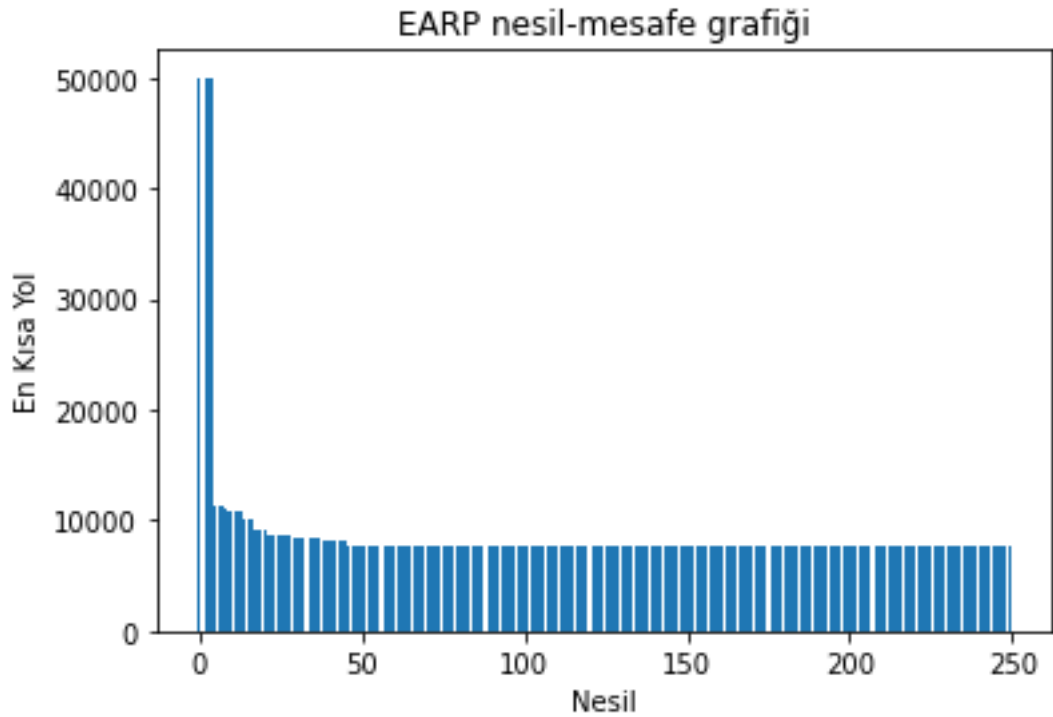
Şekil 7: 30 birey ve 100 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 3: Modeli test etmek için 30 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 100 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 7’de görülmektedir. 20. nesle kadar en kısa yolun hızla düştüğü 20. nesilden sonra optimum değerden uzak noktada yatay seyretmeye başladığı görülmüştür.



Şekil 8: 50 birey ve 100 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiđi

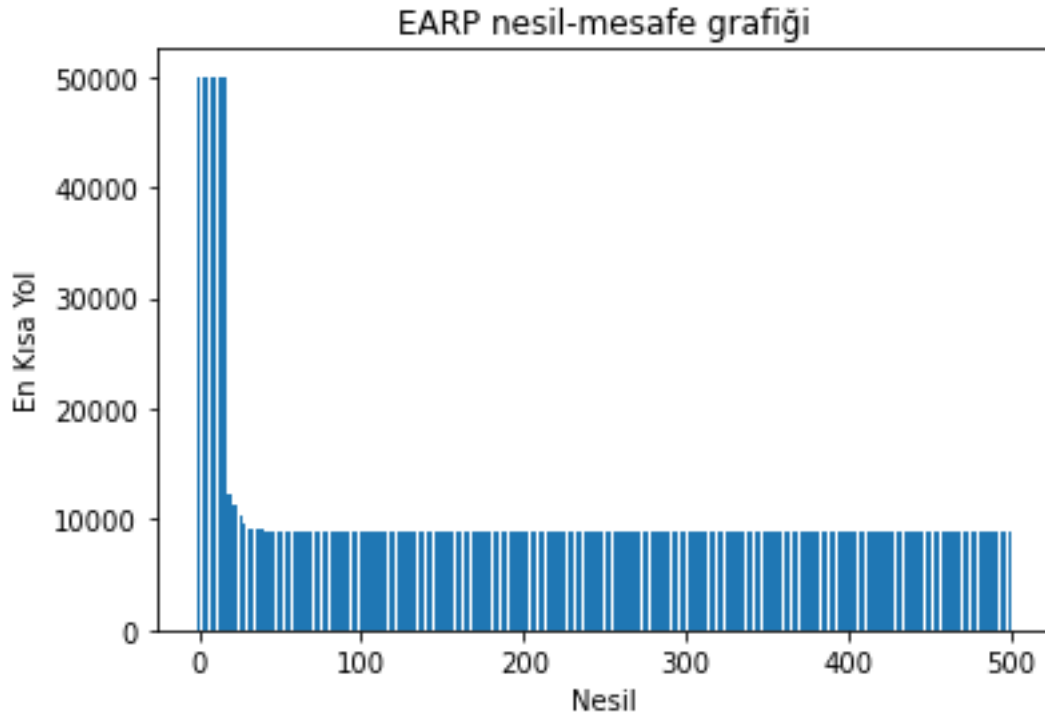
Durum 4: Modeli test etmek için 50 bireyli bir popölasyon oluşturulmuştur. Genetik algoritma 100 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 8’de görölmektedir. 20. nesle kadar en kısa yolun hızla düştüğü 20. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görölmüştür.



Şekil 9: 50 birey ve 250 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiđi

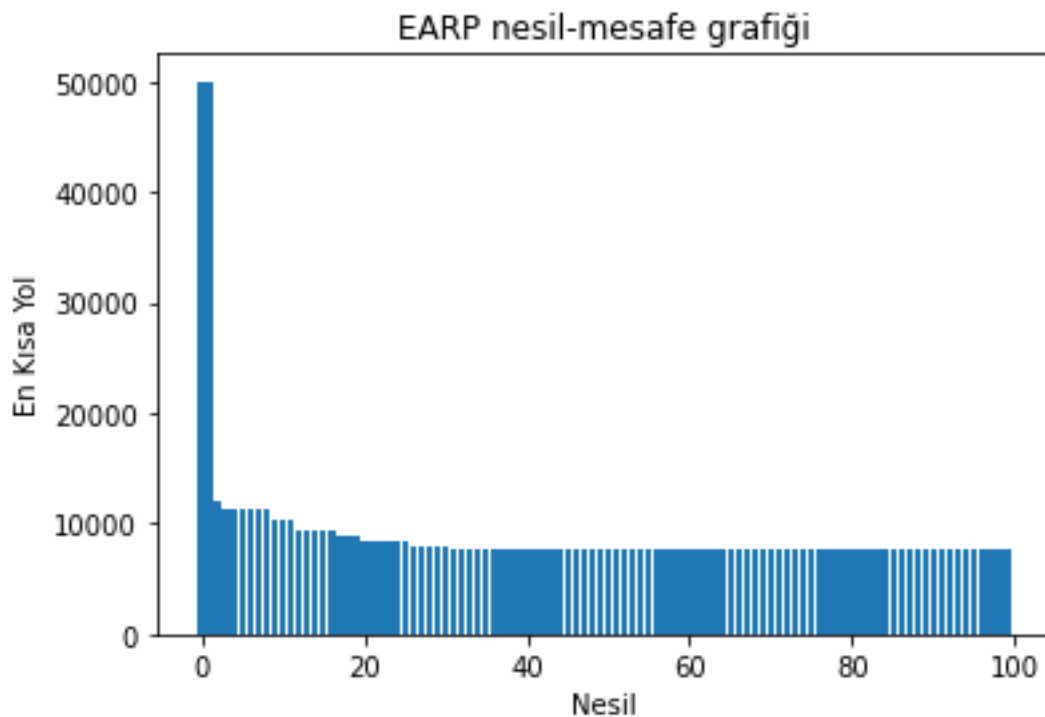
Durum 5: Modeli test etmek için 50 bireyli bir popölasyon oluşturulmuştur. Genetik algoritma 250 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 9’da görölmektedir.

50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



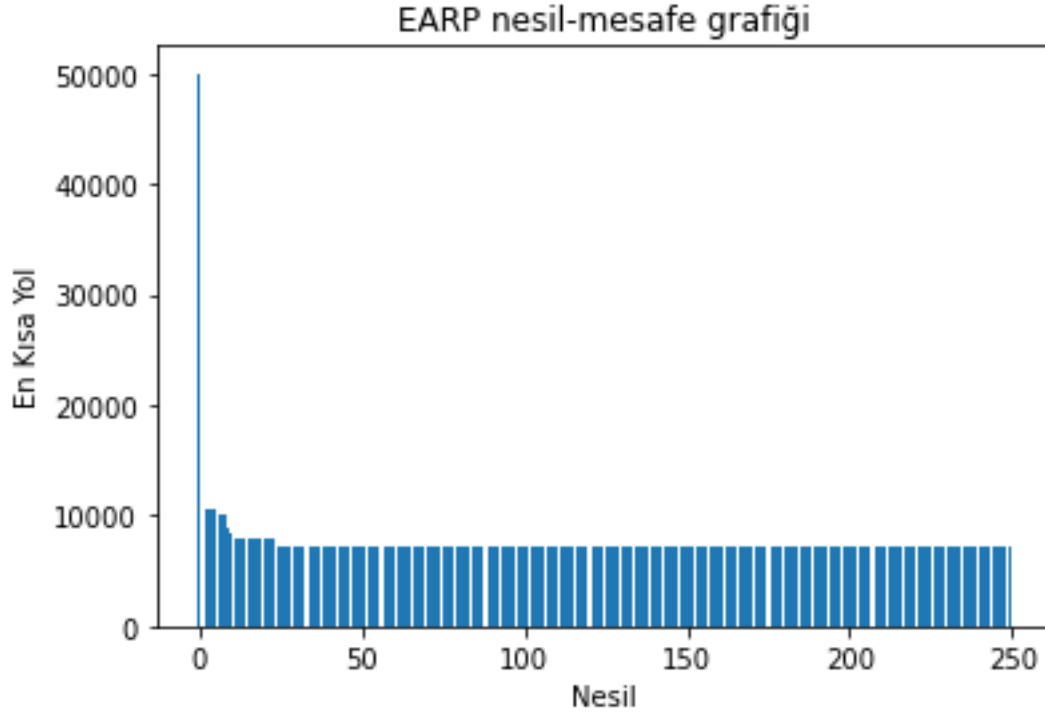
Şekil 10: 50 birey ve 500 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 6: Modeli test etmek için 50 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 500 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 10’da görülmektedir. 50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



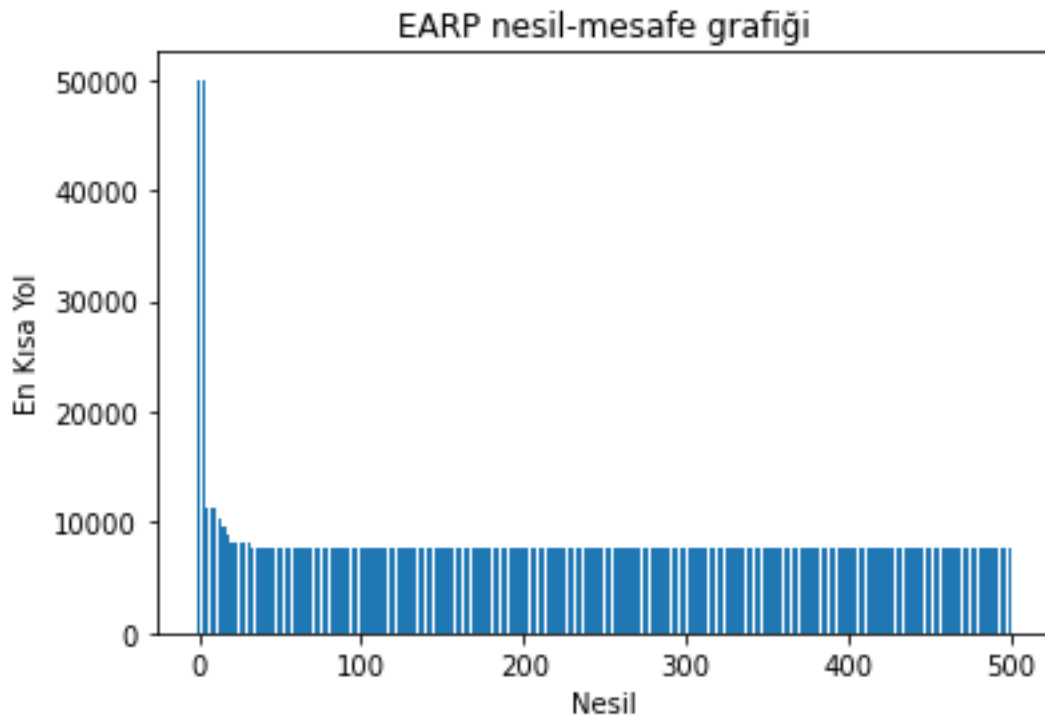
Şekil 11: 100 birey ve 100 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 7: Modeli test etmek için 100 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 100 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 10’da görülmektedir. 40. nesle kadar en kısa yolun hızla düştüğü 40. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



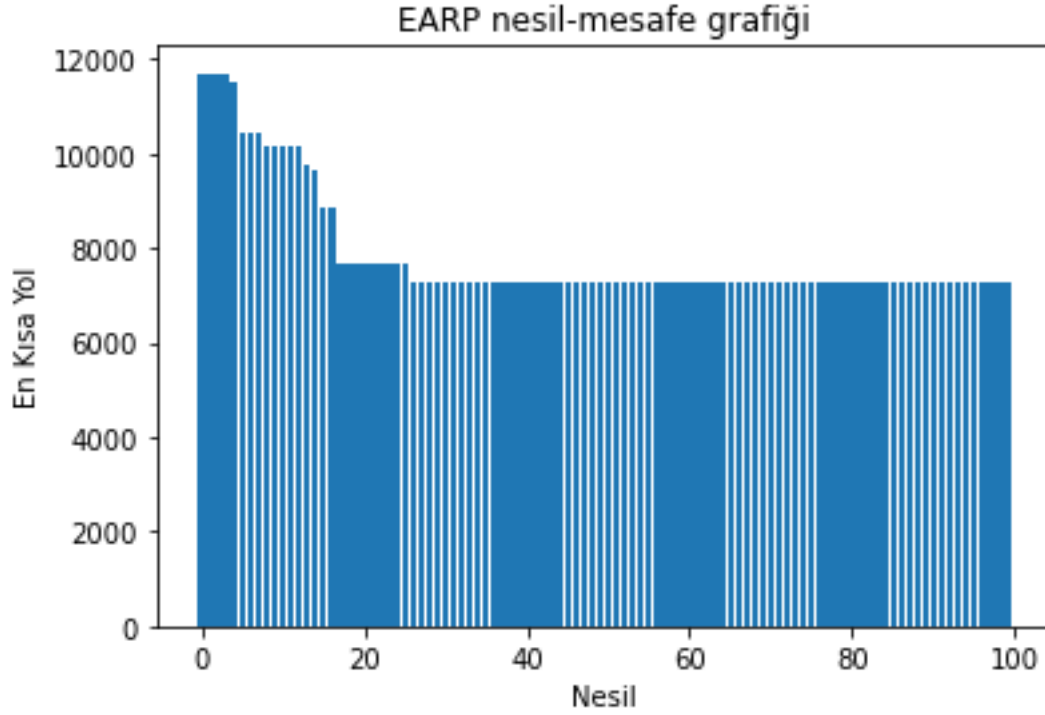
Şekil 12: 100 birey ve 250 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 8: Modeli test etmek için 100 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 250 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 12’de görülmektedir. 30. nesle kadar en kısa yolun hızla düştüğü 30. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



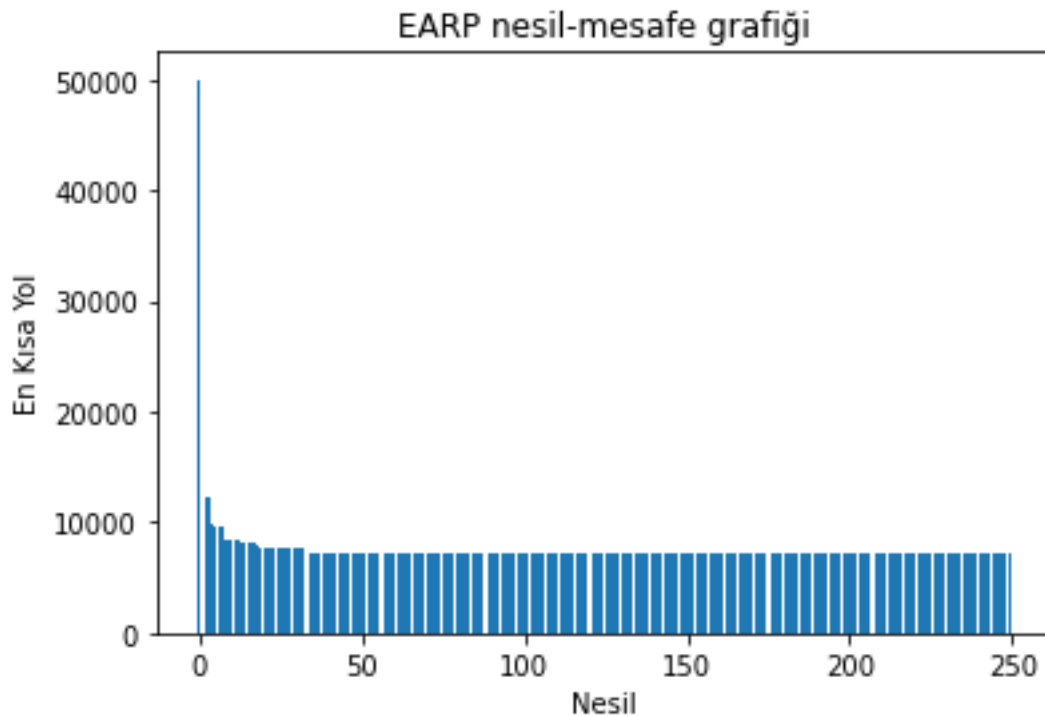
Şekil 13: 100 birey ve 500 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 9: Modeli test etmek için 100 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 500 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 13'te görülmektedir. 50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



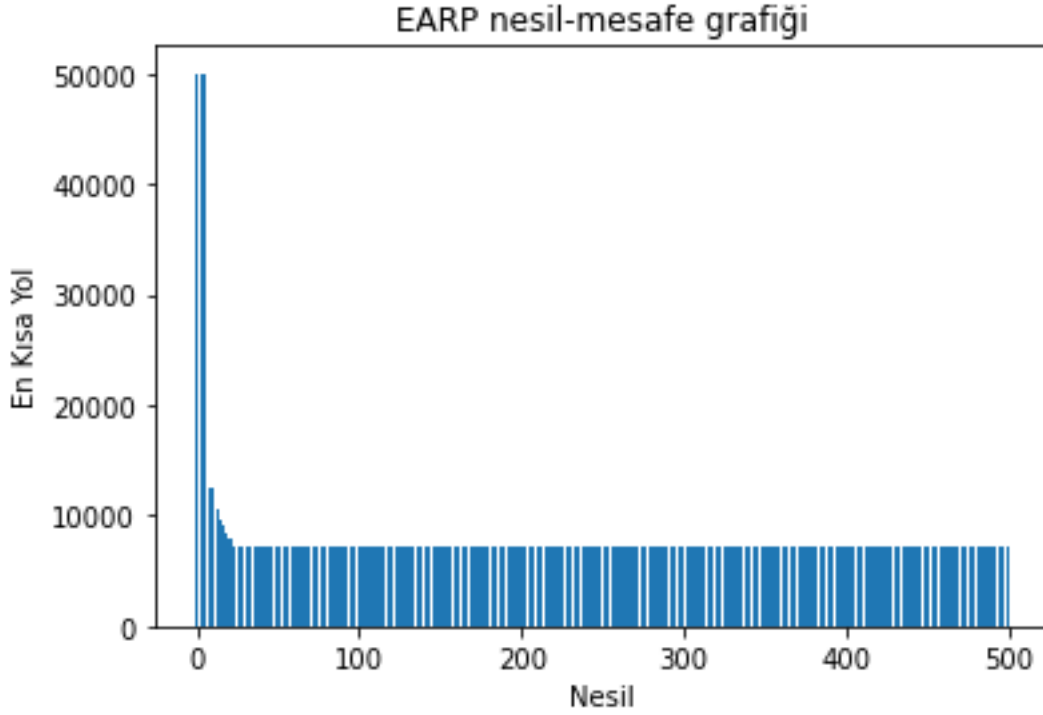
Şekil 14: 200 birey ve 100 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 10: Modeli test etmek için 200 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 100 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 14'de görülmektedir. 30. nesle kadar en kısa yolun hızla düştüğü 30. nesilden sonra optimuma uzak bir değerde yatay seyretmeye başladığı görülmüştür.



Şekil 15: 200 birey ve 250 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 11: Modeli test etmek için 200 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 250 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 15’te görülmektedir. 50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.



Şekil 16: 200 birey ve 500 nesil parametreleriyle çalıştırılan modelin nesil-mesafe grafiği

Durum 12: Modeli test etmek için 200 bireyli bir popülasyon oluşturulmuştur. Genetik algoritma 500 nesil olacak şekilde eğitilmiştir. Elde edilen sonuçlar şekil 16’da görülmektedir. 50. nesle kadar en kısa yolun hızla düştüğü 50. nesilden sonra optimum değere ulaştığı ve yatay seyretmeye başladığı görülmüştür.

Sonuç ve Tartışma

Bu çalışmada bazılarında şarj istasyonu bulunup bazılarında bulunmayan 13 şehirden oluşan şehir ağında bir en kısa rota problemi çalışılmıştır. En kısa yol rota problemini bulmak için genetik algoritma kullanılmıştır. Çalışmanın ilk aşamasında genetik algoritma ile brute force yöntemi sonucunun bulunması için çalışma zamanı açısından değerlendirilmiştir. Bulunan sonuçlara göre genetik algoritmayla çözümün daha kısa zamanda sonuç verdiği gözlemlenmiştir. Bu sonuç Ergun ve arkadaşlarının çalışmasıyla uyumludur[3].

İkinci aşamada genetik algoritmanın popülasyondaki birey sayısı ve nesil sayıları değiştirilerek 12 durum ile modelin performansı ölçülmüştür. 13 şehirden oluşan problem durumu için önerdiğimiz model ile elde edilen sonuçlar şu şekildedir:

- Nesil sayısının sabit tutulup popülasyondaki birey sayısının değiştirildiği durumlarda birey sayısına doğru orantılı olarak model performansının arttığı gözlemlenmiştir. En iyi sonucun 200 bireyli popülasyonda elde edildiği gözlemlenmiştir.
- Popülasyondaki birey sayısının sabit tutulduğu nesil sayısının değiştirildiği durumlarda ise modelin belirli bir nesle kadar performansının arttığı sonrasında yatay seyrettiği gözlemlenmiştir.
- Nesil sayısının model üzerinde doğrusal veya ters orantılı olabilecek bir etkisi görülmemiştir. 100, 250 ve 500 nesil sayılarını farklı birey sayısı ile girdiğimizde

20-75 nesil arasında hızla düşüşteyken ondan sonraki tüm nesillerde sabit kalıp herhangi bir değişim yakalanmamıştır. 75'ten fazla girildiği takdirde model için nesil sayısının çok bir etkisi yoktur.

- Yapılan çalışmada popülasyondaki birey sayısının değiştirildiğinde gözlemlenen sonuçlarda modeldeki en önemli parametre olduğu gözlemlenmiştir.

Öneriler

Yapılan çalışmada metasezgisel optimizasyon algoritmalarından genetik algoritma ile başarılı sonuçlar alınmıştır. Genetik algoritmanın yanında diğer metasezgisel algoritmalar elektrikli araç rotalama problemlerinin çözümünde kullanılabilir. Bununla birlikte elektrikli araç rotalama problemlerinde daha farklı kısıtlar tespit edilip modele eklenerek bu çalışma genişletilebilir. Bu çalışmada araç her şarj istasyonu olan şehirde şarj edilerek yola devam ettirilmiştir ve şehir ile şarj istasyonu arasındaki mesafe toplam mesafeye istemli bir şekilde eklenmemiştir. Benzer bir çalışmada gidilebilecek şarj istasyonu sınırı konularak veya şehir ile şarj istasyonu arasındaki mesafe matrisi eklenerek model genişletilebilir.

Kaynaklar

- [1] Global, E. V. (2023). Global EV Outlook 2023: Catching up With Climate Ambitions.
- [2] Stefan, I., Toth, P., & Vigo, D. (2014). The family of vehicle routing problems. *Vehicle Routing: Problems, Methods and Applications. Filadelfia: Society for Industrial and Applied Mathematics*, 119.
- [3] Ergun, Ö., Kuyzu, G., Savelsbergh, M., 2007. Shipper collaboration. *Computers & Operations Research*, 34, 1551–1560.
- [4] Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- [5] Marinakis, Y., Marinaki, M., & Dounias, G. (2010). A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 23(4), 463-472.
- [6] Qinghua, Z., Yao, L., Guoquan, C., Zhuan, W., Haiqin, H., & Kui, L. (2008, July). Improved genetic algorithm for variable fleet vehicle routing problem with soft time window. In *2008 6th IEEE International Conference on Industrial Informatics* (pp. 233-238). IEEE.
- [7] Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & operations research*, 31(12), 1985-2002.
- [8] Brezina Jr, I., & Čičková, Z. (2011). Solving the travelling salesman problem using the ant colony optimization. *Management Information Systems*, 6(4), 10-14.
- [9] Cevre, U., Özkan, B., & Uğur, A. (2007). Gezgin satıcı probleminin genetik algoritmalarla eniyilemesi ve etkileşimli olarak Internet üzerinde görselleştirilmesi. *XI I. "Türkiye'de İnternet" Konferansı, Ankara*, 8-10.
- [10] ALAYKIRAN, K., & Engin, O. (2005). Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde Bir Uygulaması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 20(1).
- [11] Shang, G., Lei, Z., Fengting, Z., & Chunxian, Z. (2007, August). Solving traveling salesman problem by ant colony optimization algorithm with association rule. In *Third International Conference on Natural Computation (ICNC 2007)* (Vol. 3, pp. 693-698). IEEE.
- [12] Özkan, B., Cevre, U., & Uğur, A. (2008). Melez bir eniyileme yöntemi ile rota planlama. *Akademik Bilişim Konferansı, Çanakkale*, 1-9.
- [13] Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing

problem. *Transportation research part E: logistics and transportation review*, 48(1), 100-114.

[14] Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation science*, 48(4), 500-520.

[15] Lin, C., Choy, K. L., Ho, G. T., & Ng, T. W. (2014). A genetic algorithm-based optimization model for supporting green transportation operations. *Expert systems with applications*, 41(7), 3284-3296.

[16] Costa, P. R. D. O., Mauceri, S., Carroll, P., & Pallonetto, F. (2018). A genetic algorithm for a green vehicle routing problem. *Electronic notes in discrete mathematics*, 64, 65-74.

[17] Conrad, R. G., & Figliozzi, M. A. (2011, May). The recharging vehicle routing problem. In *Proceedings of the 2011 industrial engineering research conference* (Vol. 8). IISE Norcross, GA.

[18] Kuyzu, G., 2017. Lane covering with partner bounds in collaborative truckload transportation procurement. *Computers & Operations Research* 77, 32–43.