

Kapasite Kısıtlı Araç Rotalama Problemi için Melez Bir Algoritma

Bilal BABAYIGIT*, Kadir YILDIZ²⁺

¹Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Erciyes Üniversitesi, Kayseri, Türkiye

²Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Erciyes Üniversitesi, Kayseri, Türkiye

*Sorumlu yazar: bilalb@erciyes.edu.tr

+Konuşmacı: kadiryildizmeb@gmail.com

Sunum/Bildiri Şekli: Sözlü / Tam Metin

Özet – Son yıllarda yaşanan ekonomik daralma sonucunda üretim maliyetlerini azaltmak ve kar marjlarını arttırmak firmalar için önemli bir hedef haline gelmiştir. Özellikle lojistik ve dağıtım şirketlerinin operasyonel işlemleri araç rotalama problemi olarak ele alınarak değerlendirilmesi gerekmektedir. Araç rotalama problemi (ARP) kombinasyonel optimizasyon problemlerinden ve NP-Zor sınıfında yer almaktadır. ARP'nin temel amacı belirli sayıda müşteriye hizmet vermek koşuluyla minimum mesafeli rotaları oluşturmaktır. Problem kapasite, zaman penceresi, mesafe gibi kısıtlar eklenerek farklı türlere ayrılabilir. Kapasite kısıtlı ARP'nde bir rotada yer alan müşteri talepleri toplamının araç kapasitesini aşmaması gerekmektedir. Literatürde problemin çözümünde kesin, sezgisel ve metasezgisel çözüm yöntemleri mevcut olmakla birlikte bu çalışmada kapasite kısıtlı araç rotalama probleminin çözümü için en yakın komşu algoritması, Yapay Arı Koloni (YAK) algoritması ve 2-opt algoritmalarının birlikte kullanıldığı melez bir yapı önerilmiştir. Önerilen algoritmada başlangıç çözümlerinin yarısı rassal diğer yarısı ise en yakın komşu yöntemi ile oluşturulmuştur. Başlangıç çözümlerine uygulanan YAK algoritması ile elde edilen en iyi çözüme 2-opt algoritması uygulanarak tur maliyetleri azaltılmıştır. Önerilen algoritma literatürde halihazırda var olan test problemleri üzerinde denenmiş ve karşılaştırmalı sonuçları elde edilmiştir. Önerilen algoritmanın en iyi çözümlere yakın değerler ürettiği gözlemlenmiştir. Bu çalışmada sunulan çözüm yöntemi literatüre önemli katkılar sağlayabilir ve çok çeşitli kombinasyonel optimizasyon problemlerinin çözümünde kullanılabilir.

Anahtar kelimeler – Kombinasyonel Optimizasyon, Kapasite Kısıtlı Araç Rotalama Problemi, Yapay Arı Koloni Algoritması, En Yakın Komşuluk, 2-opt Algoritması

I. GİRİŞ

Klasik ARP'nde, aynı tip ve kapasiteye sahip olan homojen bir araç filosu, merkezi bir depodan hareket ederek talepleri önceden bilinen bir grup müşteriye hizmet vermektedir. Problemden, her müşteri sadece bir araçtan hizmet almakta ve her araç sadece bir rota izlemektedir. Araçlar için kapasite kısıtının yanı sıra, araçların depodan hareket edip rotanın sonunda depoya geri dönme zorunluluğu vardır. Problemden söz konusu koşullar altında toplam yolculuk maliyetini en küçükleyen rotalar kümesinin bulunması amaçlanmaktadır [1]. ARP'de ilk çalışma 1959 yılında Dantzig ve Ramser tarafından yayınlanan "Truck Dispatching Problem" isimli makaledir [2]. 1964 yılında ise Clarke ve Wright geliştirdikleri tasarruf algoritması ile ARP çözümüne katkıda bulunmuşlardır [3]. ARP'ne, araç kapasiteleri, araç sayısı, dağıtımda uygulanan zaman sınırı, mesafe sınırı, depo sayısı, yük dağıtım sırasında toplama yapılması gibi araç, müşteri veya fiziki koşullardan kaynaklı bazı kısıtlamalar eklenebilir. Bu kısıtlar ARP'nin çeşitlenmesi ve alt başlıklar altında değerlendirilmesine yol açar. ARP 'de problem boyutu müşteri sayısına göre üstel orantılı olarak artmaktadır. Bu durum ARP 'ni NP (Non-deterministic Polynomial-time)-hard sınıfı bir kombinatoriyal optimizasyon problemi yapmaktadır [4]. Günümüzde lojistiğin gelişmesi ve artan müşteri talepleri araştırmacıların daha optimal çözümler ve algoritmalar geliştirmelerine yol açmıştır. Bu nedenle problemin çözümünde kesin algoritmaların varlığı bilinmesine rağmen daha kısa sürede optimal çözüm üreten yerel arama

algoritmaları, sezgisel ve metasezgisel yöntemler daha çok tercih edilmektedir. Bu algoritmalarından bazıları Genetik algoritmalar, Parçacık Sürü optimizasyonu, Tabu Arama, Benzetilmiş Tavlama, Karınca Kolonisi, Yapay Arı Koloni algoritmaları şeklinde sıralanabilir.

Bu çalışmada materyaller ve metot başlığı altında Kombinasyonel optimizasyon, Gezgin satıcı Problemi, Araç Rotalama Problemi hakkında bilgilendirmeler yapılmıştır. kullanılacak metotlar ile bunların uygulanış biçimleri açıklanmıştır. Çalışmada, tur kurucu sezgisel olarak En Yakın Komşu algoritması, temel algoritma olarak YAK algoritması ve tur geliştirici algoritma olarak 2-opt algoritması incelenmiştir. Sonuçlar bölümünde, geliştirilen yazılım ile literatürde var olan veri setleri denenmiş ve elde edilen bulgulara çalışmada yer verilmiştir. Tartışma bölümünde elde edilen sonuçlar literatürde var olan sonuçlarla karşılaştırmalı olarak ele alınmıştır. Sonuç bölümünde elde edilen veriler yorumlanmış, gelecek çalışmalara ışık tutması açısından önerilerde bulunulmuştur.

II. MATERYALLER VE METOT

Bu bölümde kombinasyonel optimizasyon, gezgin satıcı problemi ve ARP hakkında bilgilere yer verilmiştir. Ayrıca bu bölümde ARP çözüm yöntemleri ile çalışmada kullanılan metot hakkında bilgiler açıklanmıştır.

A. Kombinasyonel Optimizasyon

Optimizasyon temel olarak işgücü, zaman, kapasite, hammadde gibi kaynakların verimli kullanılması ve bunun

sonucunda kar, kapasite kullanımı, verimliliğin artırılarak maliyetlerin azaltılması olarak tanımlanabilir [5]. Başka bir ifadeyle optimizasyon bir probleme ait çözüm uzayında olası en iyi çözüme ulaşmak için geliştirilen yöntem ve metotlar bütünüdür.

Optimizasyonda öncelikle karar değişkenleri ya da tasarım parametreleri kümesinin tanımlanması gerekir. Bir sonraki aşamada bu parametrelere sadık kalarak problem tipine göre maliyet ya da kar fonksiyonları ile problem ile ilgili kısıtlamaları içeren sınırlama fonksiyonları tanımlanmalıdır. Problem tipi en küçükleme temeline dayanıyorsa maliyet fonksiyonunun daha düşük sayısal değerler üretmesi beklenir. Problem tipi en büyükleme temeline dayanıyorsa bu durumda kar fonksiyonunun daha büyük sayısal değerler üretmesi beklenir. Sınırlamalar karar parametreleri referans alınarak parametrelerin alamayacağı değerleri ifade etmektedir [6].

Optimizasyon problemleri karar değişkenlerinin alabileceği değerlerin sürekli veya ayrık olmasına göre sürekli ve kombinasyonel optimizasyon problemleri olarak gruplandırılır. Kombinasyonel optimizasyon problemleri ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması, sıralanması ve seçilmesi işlemleri temeline dayanmaktadır.

B. Gezin Satıcı Problemi

Gezin Satıcı probleminin temelleri William Rowan Hamilton'un kendi adını taşıyan Hamilton turuna dayanmaktadır. Hamilton turu, bir grafda seçilen bir başlangıç düğümünden hareket ederek tüm düğümlere bir kez uğrayıp başlangıç noktasına dönen çevrimdir [7].

Optimum rota oluşturmaya yönelik bilinen en eski problemlerden biri Gezin Satıcı problemidir. Yöneylem araştırmasının en ilgi çekici problemlerinden biri olan Gezin satıcı probleminde temel amaç bir noktadan veya şehirden başlayarak n adet noktayı veya şehri en küçük Hamilton turu oluşturacak şekilde ziyaret etmek ve başlangıç noktasına geri dönmektir. En küçük Hamilton turu, toplam mesafeyi veya maliyeti en küçükleyen tur olarak tanımlanabilir [8].

Problemin matematiksel ifadesi için uzaklık veya maliyet matrisi $D = (d_{i,j})$ tanımlanmalıdır. Burada her $d_{i,j}$ i şehirden j şehrine gitmek için gerekli olan mesafe veya maliyeti temsil etmektedir. Problemin çözümü n elemanlı problem kümesinin minimum mesafe veya maliyeti oluşturacak şekilde yerleştirildiği $P(i_1, i_2, \dots, i_n)$ bir permütasyona dayanmaktadır. Burada toplam maliyet, maliyet matrisi kullanılarak $d_{i_1, i_2} + d_{i_2, i_3} + \dots + d_{i_{n-1}, i_n} + d_{i_n, i_1}$ şeklinde ifade edilir.

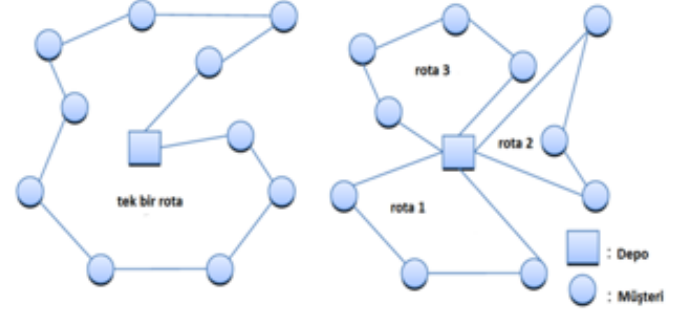
n şehirli problemin çözüm uzayının $(n-1)!/2$ olduğu düşünüldüğünde problemin kesin çözümünün ne denli zor olduğu anlaşılmaktadır. Üstelik probleme eklenecek her yeni şehrin çözüm uzayını üstel olarak arttıracığı ve kesin çözümün bulunmasının daha da zorlaşacağı açıktır [9].

C. Araç Rotalama Problemi

Araç Rotalama Problemi bir takım kısıtlar eşliğinde coğrafi olarak dağılık şehirler veya müşteriler için bir veya birden çok depoda yer alan araç filosuna ait rotaların oluşturulmasını içeren problem sınıflarının tümüne verilen genel bir isimdir [10].

Araç Rotalama Probleminin (ARP) temel bir türü olan Kapasite Kısıtlı Araç Rotalama Problemi (KKARP) statik ve deterministik bir yapıdadır. KKARP'de tüm müşteriler teslimatlarla bire bir eşleşmektedir. Talepler önceden bilinir ve

bölünemez. Araçlar homojen yapıdadır ve eşit taşıma kapasitesine sahiptir. Dağıtım merkezi tek bir depodan yapılmaktadır. Araçlar için sadece kapasite kısıtı uygulanmaktadır. Tüm müşterilere hizmet verme zorunluluğunun olduğu problemde amaç fonksiyonu toplam maliyeti (rota sayısını ve/veya uzunluklarını veya seyahat sürelerini) minimize etmektir.



Şekil 1. a) Gezin satıcı problemi b) Araç rotalama problemi

D. ARP Matematiksel Modeli

KKARP teorik olarak graf problemi olarak tanımlanabilir. $G = (V, A)$ tam grafında $V = \{0, 1, \dots, n\}$ ve $A = \{(i, j) : i, j \in V, i \neq j\}$ şeklinde ifade edilir. Burada V talepleri karşılanacak coğrafi olarak dağılmış müşterilerin oluşturduğu düğümler kümesini temsil etmektedir. $\{0\}$ numaralı düğüm ise depo konumudur. A bir rotada iki düğüm noktası tarafından oluşturulan hatları ifade eder. i ve j notasyonları mesafe ve uzaklık matrisi hesaplamalarında kullanılan müşteri çiftlerini temsil etmektedir. $d_{i,j}$: i müşterisi ile j müşterisi arasındaki mesafeyi ve $c_{i,j}$: i müşterisi ile j müşterisi arasındaki mesafeden elde edilen uzaklık matrisini ifade etmektedir. $d_{i,j} = d_{j,i}$ ve $c_{i,j} = c_{j,i}$ kabulleri ile problemimizin simetrik olacağını garantiliyoruz. Eğer G yönlendirilmiş bir graf ise $c_{i,j}$ matrisi ve problemimiz asimetrik bir yapı kazanır. Bu bilgiler ışığında KKARP problemi V kümesindeki düğümlerin A kümesindeki hatları oluşturmasını sağlayan bağlantılar kümesi şeklinde tanımlanır [11].

Problem çözümü için tanımlanması gereken diğer notasyonlar sırasıyla her biri pozitif değere sahip olan müşteri talepleri (q_i) ve homojen taşıma kapasitesine (Q) sahip araç sayıları (K) şeklinde ifade edilir.

Yukarıdaki kabuller ve kısıtlamaları göz önünde bulundurarak problemin matematiksel modelini aşağıdaki gibi ifade edebiliriz [12];

Notasyonlar;

$G = (V, A)$
 $V = \{v_0, v_1, v_2, v_3, v_4, \dots, v_n\}$ burada v_0 depoyu v_1, v_2, \dots, v_n düğümleri müşterileri temsil etmektedir.
 q_i : i düğümüyle temsil edilen müşteri talep miktarı, $i \in V$
 $d_{i,j}$: v_i ve v_j düğümleri (müşterileri) arasındaki mesafe
 $K = \{k_1, k_2, \dots, k_m\}$ araç filosu
 Q : Homojen yapıdaki her bir aracın kapasitesi, $k_i \in K$

Karar Değişkenleri;

$x_{i,j}^k = \begin{cases} 1, & k \text{ i. müşteriden j. müşteriye giderse} \\ 0, & \text{aksi durumda} \end{cases}$
 $y_i^k = \begin{cases} 1, & k \text{ aracının i. müşteriye hizmet vermesi} \\ 0, & \text{aksi durumda} \end{cases}$

Denklemsel İfadeler;

$$\text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{i,j} x_{i,j}^k \quad (1)$$

$$\sum_{k \in V} \sum_{j \in N} x_{i,j}^k = 1, \quad \forall i \in V \quad (2)$$

$$\sum_{j \in V} x_{i,j}^k + \sum_{j \in V} x_{j,i}^k = 1, \quad \forall i \in V, k \in K \quad (3)$$

$$\sum_{k=1}^K \sum_{j=1}^N x_{0,j}^k = K \quad (4)$$

$$\sum_{j \in V} x_{0,j}^k = 1, \forall k \in K \quad (5)$$

$$\sum_{j \in V} x_{j,n+1}^k = 1, \forall k \in K \quad (6)$$

$$x_{i,j}^k = 1 \Rightarrow y_i - q_i = y_j, \forall i, j \in V, k \in K \quad (7)$$

$$y_0 = Q, 0 \leq y_i, \forall i \in V \quad (8)$$

$$\sum_{i=1}^N q_i \sum_{j=0, j \neq i}^N x_{i,j}^k \leq Q \quad k \in \{1, \dots, m\} \quad (9)$$

$$x_{i,j}^k \in \{0,1\}, \forall i, j \in V, k \in K \quad (10)$$

Denklem (1) tüm araçlar tarafından katedilen toplam mesafenin minimize edilmesi için kullanılan amaç fonksiyonu gösterimidir. (2) kısıtı i, j çiftlerinin oluşturduğu her bir rota hattına sadece bir aracın hizmet vermesini sağlar. (3) kısıtı rota boyunca araçların ziyaret ettiği bir düğüme geri dönmesini engeller. (4) kısıtı rotalamada kullanılacak her aracın başlangıç noktasının depo konumu olmasını sağlar. (5) kısıtı her aracın depodan sadece bir kez çıkışını garantiler. (6) kısıtı her müşterinin sadece bir araç tarafından hizmet almasını garantiler. (7) kısıtı i ve j düğümler arası geçişler sonucunda araç kapasitesindeki azalmayı ve kalan araç kapasitesini gösterir. (8) kısıtı rotalamada kullanılacak her aracın başlangıç kapasite değerinin Q olacağını garantiler. (9) kısıtı bir rotaya atanan düğüm noktalarındaki müşteri talepleri toplamının o rotada kullanılan araç kapasitesini aşamayacağını gösterir. (10) kısıtı ise (2) ve (3) kısıtları için referans gösterilen bir tam sayı kısıtı olarak ifade edilebilir.

E. ARP Çözüm Yöntemleri

ARP çözümleri kesin yöntemler, sezgisel yöntemler ve metasezgisel yöntemler alt başlıkları altında toplanır.

Matematiksel tabanlı olan kesin çözüm yöntemlerinden bazıları dal-sınır, dal-kesme, dinamik programlama ve dal-kesme –fiyat algoritması olarak sıralanabilir.

Sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için çeşitli alternatif hareketlerden etkili olanlara karar vermek amacıyla tanımlanan kriterler veya

bilgisayar metotlarıdır. Bu tür algoritmalar yakınsama özelliğine sahiptir, ama kesin çözümü garanti edemezler ve sadece kesin çözüm yakınındaki bir çözümü garanti edebilirler [13].

Metasezgisel yöntemler çözüm uzayında optimal sonuca ulaşacak şekilde arama yapmak için, farklı özelliklere sahip sezgisel algoritmaların zekice biraraya getirilmesinden oluşmuş iterasyona dayalı çözüm yöntemleridir. Her iterasyonda bir çözüm veya bir çözüm kümesi elde edilir. Metasezgisel yöntemlerin çoğu deterministik değildir. Genellikle stokastik ancak bilinçli arama yaparlar [14].

Bu çalışmada sezgisel ve metasezgisel algoritmaların birlikte kullanıldığı hibrit bir yapı önerilmiştir. Kullanılan algoritmalar En yakın komşu algoritması, 2-opt algoritması ve yapay arı koloni algoritması şeklinde sıralanmaktadır.

- *En Yakın Komşu Algoritması*

Algoritma rastgele seçilen bir şehirden başlar. Seçilen şehrin uzaklık matrisindeki satırı taranarak en yakın komşusu seçilir ve rotaya eklenir. Tüm şehirler rotaya eklenene kadar en yakın komşu ekleme işlemi tekrarlanır. Algoritma en son rotaya eklenen şehirden depoya olan mesafenin de maliyet değerine eklenmesiyle son bulur. Yöntem tek bir rota üzerinden hareket edilen seri ve aynı anda birden fazla rota üzerinden hareket edilen paralel olmak üzere iki gruba ayrılır. Karmaşıklık düzeyi $Q(n^2)$ olan yöntemde rotaya eklenen son şehirlerin toplam maliyet üzerindeki etkileri çok yüksek olabilmektedir.

- *2-opt Algoritması*

2-opt ve 3-opt algoritmaları tur geliştirici sezgisel olarak kullanılmaktadır. 2-opt sezgiselinde birbirine komşu olmayan iki farklı kenarın yer değiştirmesinin toplam maliyet üzerinde olumlu etki yapması amaçlanmaktadır. Seçilen iki kenarda bu kenarları oluşturan düğüm noktaları kombinasyonel olarak denenir ve maliyeti azaltacak şekilde kenarların konumları yer değiştirilir. Bu işlemin sonucu olarak kenarları oluşturan düğüm noktaları da değişmiş olur. Şekil 3'de 2-opt sezgiselinin çalışma prensibi gösterilmiştir. Buna göre A ve B düğümlerinin oluşturduğu kenar ile D ve E düğümlerinin oluşturduğu kenar, düğüm bağlantıları değişecek şekilde A ve D düğümleri ve bunlara ait kenar ile E ve B düğümleri ve bunlara ait kenar olmak üzere düzenleniyor[15].

- *Açgözlü (Greedy) Algoritması:*

Açgözlü yaklaşım ile bir problemin çözümünde anlık olarak kısıtlı seçenekler arasından en fazla kazanç elde edilen nokta tercih edilir. ARP için Açgözlü sezgiseli her adımda daha önce kullanılmamış olan en kısa değerli kenarları ekleyerek tur oluşturur ve bunu yaparken alt tur oluşumunu engellemeye çalışır. Algoritmanın genel adımları aşağıdaki gibidir.

Adım 1: Oluşturulan tüm kenarlar için büyükten küçüğe sıralama yapılır.

Adım 2: En kısa kenar alt tur oluşturmayacak şekilde ilgili tura eklenir.

Adım 3: Bütün düğümler tura katılana kadar 2. adım yenilenir. Yapay Arı Koloni Algoritması(YAK)

Arı kolonisi gibi sürü tabanlı algoritmalarda sürü zekasını yansıtan en belirgin özellikler iş paylaşımı yapabilme ve

kolonide bulunan her türün öz-kontrol yapısı ile organize olabilesidir. Bir arı kolonisi belirli işleri(görevleri) yerine getirmek için özelleştirilmiş ve otonom yapıya sahip arı türlerinden oluşmaktadır. Tereshko'nun sürü zekasının temellerini oluşturan bal arısı kolonisinde yiyecek aramanın reaktif difüzyon denklemleri ile modellenmesi çalışmasında yiyecek arama modeli yiyecek kaynakları, görevli işçi arılar ve görevsiz işçi arılardan meydana gelmektedir [16].

Karaboğa [17] tarafından literatüre kazandırılan Yapay Arı Koloni Algoritması (YAK) gerçek arı davranışlarının modellenmesi temeline dayanmaktadır. Bir besin kaynağının tek bir görevli arı tarafından işlenebilmesi, işçi arıların sayısının gözcü arıların sayısına eşit olması gibi kabullerin yapıldığı algoritma da nektar miktarı biten besin kaynağındaki görevli arı kâşif arıya dönüşmektedir. Bir optimizasyon probleminde yiyecek kaynağının konumları çözüm uzayında bulunan olası çözümler kümesini temsil etmektedir. Kaynakların nektar miktarı ise her bir çözüme ait uygunluk değerlerinden elde edilir. Diğer optimizasyon algoritmalarında olduğu gibi ABC optimizasyon algoritmasında da temel amaç çözüm uzayının taranarak optimum değerleri elde etmektir. Probleminde bulunan optimum çözüm nektar miktarı en fazla olan besin kaynağını temsil etmektedir. Modele ait aşamalar aşağıdaki gibidir;

F. Önerilen Algoritma

Bu çalışma kapsamında HA-1 (Hibrit Algoritma -1) olarak isimlendirilen hibrit algoritma tur kurucu sezgisel, ana algoritma ve tur geliştirici sezgisel olmak üzere 3 aşamadan oluşan bir yapıdadır. Tur kurucu sezgisel olarak En yakın komşu algoritması ana algoritma olarak YAK ve tur geliştirici olarak da 2-opt algoritmaları kullanılmıştır.

Kodlama Düzeni: Çözüm uzayı içinden seçilen başlangıç çözümleri kümesi literatürde var olan kodlama türlerinden permütasyon kodlama ile üretilmiştir. Çözümler tüm müşteri numaralarının yerleştirildiği dizi formatında üretilmektedir.

Başlangıç Çözümlerinin (Besin Kaynakları) Üretilmesi: Üretilen başlangıç çözümleri, çözüm uzayı araştırmasını hem global hemde lokal çözüm noktalarında yapabilmesi için iki yöntemle oluşturulmuştur. Başlangıç çözümlerinin yarısı rastgele permütasyon kodlama ile üretilmiştir. Oluşturulan çözüm dizilerindeki müşteri numaraları rastgele seçilerek sırası ile diziyeye eklenmiştir. Başlangıç çözümlerinin diğer yarısı en yakın komşuluk tabanlı bir yerleştirme yöntemi ile oluşturulmuştur. Yöntemde rastgele seçilen bir müşteri numarası ile başlanarak her seferinde diziyeye son eklenen elemana en yakın mesafedeki müşteri numarası eklenmiştir. Tüm müşteri numaraları eklendiğinde dizi sonlandırılır ve başlangıç çözümü üretilmiş olur.

Uygunluk Değerlerinin (Nektar Miktarlarının) Hesaplanması: Oluşturulan her başlangıç çözümü için KKARP'nde bulunan amaç fonksiyonu uygunluk değeri hesaplanmasında kullanılmıştır. Uygunluk değerlerinin hesaplanmasında kullanılan uygunluk fonksiyonu aşağıda ifade edilmiştir.

$$\text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{i,j} x_{i,j}^k \quad (11)$$

Burada $d_{i,j}$: i ve j müşterileri arasındaki mesafe(maliyet), $K = \{k_1, k_2, \dots, k_m\}$ araç filosu ve $x_{i,j}^k$ karar değişkenidir.

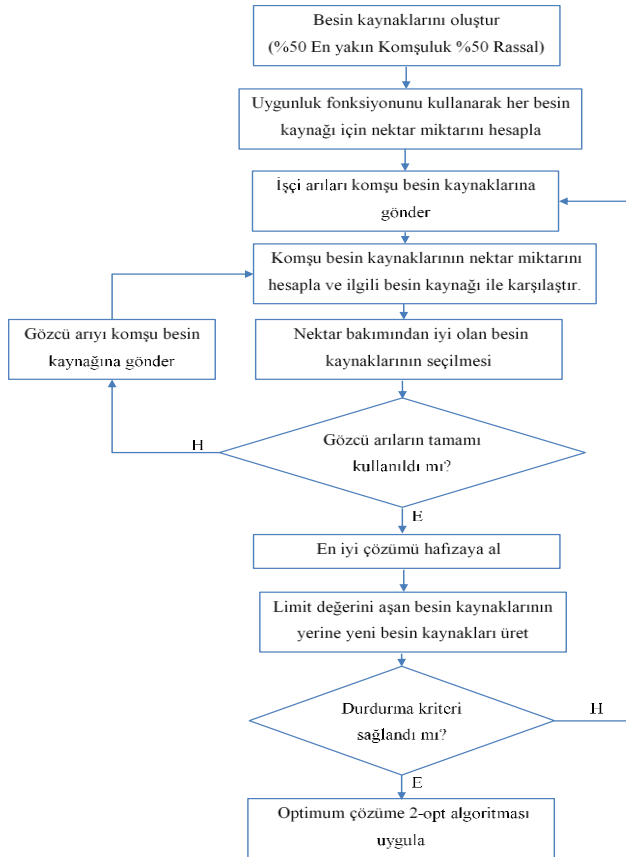
HA-1 Algoritmasına Uygulanan YAK Algoritmasına ait Süreçler: Nektar miktarları hesaplanan besin kaynaklarına işçi arılar gönderilir. Gönderilen işçi arılar görevlendirildikleri besin kaynağının komşuluğundaki besin kaynaklarını araştırır. Bu arama işlemi bir besin kaynağında yer alan müşterinin komşuluğunda yer alan müşteri ile yer değiştirmesi temeline dayanır. Bu işlem için 0-1 arasında değiştirme oranı(d_o) belirlenir. Her müşteri için 0-1 arasında üretilen rassal sayı, değiştirme oranı altında ise komşu yer değiştirme işlemi gerçekleştirilir. Aksi durumda ilgili müşteri dizideki yerinde kalır. Değiştirme işlemi sonucu komşu besin kaynaklarının (çözümlerin) nektar miktarları (uygunluk değerleri) hesaplanır. Elde edilen komşu besin kaynağı var olan besin kaynağı ile karşılaştırılır ve algoritmaya daha iyi olan besin kaynağı ile devam edilir. İşçi arılar besin kaynaklarını tarama işlemini sonlandırdığında gözcü arıların daha spesifik arama yapabilmeleri için besin kaynağı seçim işlemi yapılır. Algoritmada bu işlem rulet tekeri yöntemi kullanılarak gerçekleştirilmiştir. Seçilen besin kaynakları yeterli sayıya ulaştığında seçim işlemi sonlandırılır ve gözcü arılar besin kaynaklarına gönderilir. Gözcü arılarda işçi arı safhasında olduğu gibi komşu besin kaynaklarını belirleyerek bu besin kaynaklarına ait nektar miktarlarını (uygunluk değerlerini) hesaplar. Elde edilen komşu besin kaynağı var olan besin kaynağından daha iyi ise besin kaynağı değiştirilir. Tüm gözcü arılar besin kaynaklarını incelediğinde o ana kadar ki en iyi çözüm değeri hafızaya alınır. Programda belirlenen limit değerini aşan çözümler için rassal olarak oluşturulan yeni çözümlerle (besin kaynakları) ile algoritmaya devam edilir. Algoritma da işlenen besin kaynaklarının geliştirilemediğini belirten limit değerinin aşıldığı durumlarda ilgili besin kaynağı değiştirilir. Bu işlem için değiştirme oranı 1 olarak belirlenmiştir. Algoritmada durdurma kriteri sağlanana kadar işçi arıların besin kaynağına gönderildiği safhadan itibaren algoritma tekrarlanır.

Durdurma Kriteri: İterasyon temeline dayanan metasezgisel algoritmalarda alt ve üst sınırların belirlendiği eşik değerlerinin aşılması, iterasyon sayısı, algoritmanın çalışma süresi, algoritmanın belirli bir noktaya yakınsaması gibi durdurma kriterleri mevcuttur. Bu çalışmada iterasyon sayısı durdurma kriteri olarak alınmıştır. Her veri setinde oluşturulan algoritma 100 kez tekrarlanmıştır.

Tur Geliştirici Sezgisel Uygulanması: Yapılan çalışmada, elde edilen en optimum çözümde tur maliyetini optimize etmek için 2-opt algoritması uygulanmıştır. Her bir alt turun optimize edilmesi temeline dayanan yaklaşımda alt turlardan elde edilen tasarruf miktarları toplamı kadar rota iyileştirilmesi yapılmıştır. Veri setleri üzerinde uygulanan 2-opt algoritmasının tur maliyetlerini önemli ölçüde düşürdüğü gözlemlenmiştir. HA-1 algoritmasının uygulama adımları aşağıda belirtilmiştir.

Tablo 1. HA-1 Algoritması işlem adımları

Adım 1: Permütasyon kodlama türü kullanarak besin kaynaklarının yarısını en yakın komşu yöntemi ile diğer yarısını rassal olarak oluşturma
 Adım 2: Uygunluk fonksiyonunu kullanarak besin kaynaklarının nektar miktarlarını hesaplama
 Adım 3: TEKRARLA
 Adım 4: İşçi arılarının komşu besin kaynaklarına gönderilmesi
 Adım 5: Komşu besin kaynağının nektar miktarlarının hesaplanarak ilgili besin kaynağı ile karşılaştırılıp iyi olan besin kaynağının seçilmesi
 Adım 6: Nektar bakımından iyi olan besin kaynaklarının belirlenen seçim yöntemi ile seçilmesi
 Adım 7: Gözcü arıların besin kaynaklarına dağıtılarak komşu besin kaynaklarının incelenmesi
 Adım 8: Komşu besin kaynağının nektar miktarlarının hesaplanarak ilgili besin kaynağı ile karşılaştırılıp iyi olan besin kaynağının seçilmesi
 Adım 9: En iyi çözümün hafızaya alınması
 Adım 10: Limit değerini aşan ve nektar miktarı tükenen besin kaynağının yeni üretilen besin kaynağı ile değiştirilmesi
 Adım 11: Belirlenen durdurma kriteri sağlanana kadar algoritmanın iteratif olarak 3. adımdan itibaren çalıştırılması
 Adım 12: Elde edilen optimum çözüme 2-opt algoritmasının uygulanması



Şekil 2. HA-1 algoritması akış diyagramı

III. BULGULAR VE TARTIŞMA

Literatürde KKARP için yapılan hemen hemen tüm yeni metodların denendiği veri setleri mevcuttur. Bunlardan bazıları Augerat vd. [18] tarafından önerilen A ve P sınıfı veri setleri, Christofides ve Eilon [19] tarafından önerilen E sınıfı veri seti ve Fisher [20] tarafından önerilen F sınıfı veri seti şeklinde sıralanabilir. Belirtilen test problemleri üzerinde HA-1 algoritmalarının etkinliği araştırılmıştır. Yapılan simülasyonlardan elde edilen sonuçlara çalışmada yer verilmiştir.

No	Veri Seti	Müşteri	Kapasite	Rota	En İyi	HA-1	Fark	Fark(%)
1	A-n32-k5	32	100	5	784	809,33	25,33	3,23
2	A-n33-k5	33	100	5	661	692,82	31,82	4,81
3	A-n33-k6	33	100	6	742	753,25	11,25	1,52
4	A-n34-k5	34	100	5	778	781,96	3,96	0,51
5	A-n36-k5	36	100	5	799	817,48	18,48	2,31
6	A-n37-k6	37	100	6	949	967,5	18,5	1,95
7	A-n39-k6	39	100	6	831	874,17	43,17	5,19
8	A-n44-k6	44	100	6	937	998,04	61,04	6,51
9	A-n63-k10	63	100	10	1314	1394,36	80,36	6,12
10	A-n69-k9	69	100	9	1159	1249,25	90,25	7,79
11	P-n16-k8	16	35	8	450	450	0	0
12	P-n19-k2	19	160	2	212	219,75	7,75	3,66
13	P-n20-k2	20	160	2	216	218,3	2,3	1,06
14	P-n21-k2	21	160	2	211	211	0	0
15	P-n22-k2	22	160	2	216	221,12	5,12	2,37
16	P-n22-k8	22	3000	8	603	603	0	0
17	P-n50-k7	50	150	7	554	597,17	43,17	7,79
18	P-n55-k10	55	115	10	694	742,1	48,1	6,93
19	E-n22-k4	22	6000	4	375	381,84	6,84	1,82
20	E-n23-k3	23	4500	3	569	568,56	-0,44	-0,08
21	E-n30-k3	30	4500	3	534	547,74	13,74	2,57
22	E-n33-k4	33	8000	4	835	870,07	35,07	4,2
23	E-n76-k10	76	140	10	830	894,65	64,65	7,79
24	F-n45-k4	45	2010	4	724	770,3	46,3	6,4

Şekil 3. HA-1 algoritması ile elde edilen sonuçlar

Önerilen algoritma literatürde KKARP için sıklıkla kullanılan 4 farklı veri setine ait problemlerde denenmiş ve en iyi çözümlere yakın değerler elde edilmiştir. Müşteri sayıları 16 ile 76 arasında değişen problemlerde çalıştırılan algoritma her problemin en iyi çözüm değeri ile karşılaştırılmıştır. P-n21-k2, P-n22-k8 ve P-n16-k8 problemlerinde en iyi çözüme ulaşılmıştır. E-n23-k3 probleminde en iyi çözümden % 0.08 daha iyi sonuç elde edilmiştir. En yüksek yüzdelik fark % 7.79 olarak A-n69-k9, P-n50-k7 ve E-n76-k10 problemlerinde gerçekleşmiştir. Problemin NP-Hard sınıfı ve çözüm uzayı büyüklüğü dikkate alındığında sunulan çözüm yöntemi literatüre önemli katkılar sağlayabilir ve çok çeşitli kombinasyonel optimizasyon problemlerinin çözümünde kullanılabilir.

IV. SONUÇ

Geliştirilen hibrit HA-1 algoritmasında tur kurucu olarak en yakın komşu algoritması ana algoritma olarak YAK algoritması ve tur geliştirici olarak 2-opt algoritması kullanılmıştır. Önerilen algoritma literatürde KKARP için sıklıkla kullanılan 4 farklı veri setine ait problemlerde denenmiş ve en iyi çözümlere yakın değerler elde edilmiştir. Yapılan çalışmadan elde edilen sonuçlar ışığında sonraki çalışmalar için aşağıdaki öneriler yapılabilir;

- Önerilen algoritmalar daha büyük boyutlu problemler için test edilebilir.

- ARP çözümünde kullanılan diğer metasezgisel yöntemlerle uygulama yapılarak sonuçlar karşılaştırılabilir.
- Başka melez sistemler kullanılarak mevcut durum iyileştirilebilir.
- Önerilen algoritma ARP'nin başka türleri için test edilebilir.
- Ele alınacak problemlerde trafik yoğunluğu, yolların ve coğrafi yapının özellikleri dikkate alınabilir.
- İstatistiki veri tutularak taleplerin geliş frekansı tespit edilir ve öğrenme algoritmaları yardımıyla talep tahmin çizelgesi oluşturulabilir.
- Önerilen algoritmalar günümüzde yavaş yavaş hayatımıza giren hava araçları rotalamasında da kullanılabilir.
- Önerilen algoritmalar doğal afetleri izleme süreçlerini hızlandırmak için belirli bir araç filosunun rotalanmasında kullanılabilir.

Sonuç olarak mevcut çalışmayı, test problemlerinde denenmesi ve geliştirilen çözüm yöntemlerinin özgünlüğü kapsamında değerlendirdiğimizde literatüre katkı sağlayacağı düşünülmektedir.

KAYNAKÇA

- [1] M. Özcanlı, H. Serin, "Evaluation Of Soybean/Canola/Palm Biodiesel Mixture As An Alternative Diesel Fuel", Journal of Scientific & Industrial Research, vol. 70, pp.466-470, 2011.
- [2] G. B. Dantzig, J. M. Ramser, "The truck dispatching problem", Management Science, vol. 6, pp. 81-91, 1959.
- [3] G. Clarke, J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research, vol. 12, pp. 568-581, 1964.
- [4] Ç. Alabaş, B. Dengiz, "Yerel Arama Yöntemlerinde Yöre Yapısı: Araç Rotalama Problemine Bir Uygulama", Yöneylem Araştırması/Endüstri Mühendisliği 24. Ulusal Kongresi, 2010, Gaziantep – Adana, p. 333 – 335.
- [5] M. Türkay, Optimizasyon modelleri ve çözüm metodları. (Web Sayfası: <http://home.ku.edu.tr/~mturkay/indr501/Optimizasyon.pdf>), (Erişim tarihi: Nisan 2019).
- [6] D. Karaboğa, Yapay Zeka Optimizasyon Algoritmaları. Atlas Yayın Dağıtım, İstanbul, 2004.
- [7] N. L. Biggs, E. K. Llyod, R. J. Wilson, Graph Theory. Clarendon Press, Oxford, UK. 1976.
- [8] R. Matai, S. P. Singh, M. L. Mittal, Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches. pp. 1-24. In: Traveling Salesman Problem: Theory and Applications (Eds. D. Davendra), Intech Open Access Publisher, 2010.
- [9] B. Gavish, S. C. Graves, "The travelling salesman problem and related problems", Technical Report OR 078-78, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA. 1978..
- [10] C. Rego, "Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms", Parallel Computing, vol 27, pp. 201-222, 2001.
- [11] P. Toth, D. Vigo, The Vehicle Routing Problem Society for Industrial and Applied Mathematics, Philadelphia, pp. 367, 2002.
- [12] A. H. El Hassani, A. Koukam, L. Bouhaf, A hybrid ant colony system approach for the capacitated vehicle routing problem and the capacitated vehicle routing problem with time Windows. pp. 57-70. In: Vehicle Routing Problem, (Eds. T. Caric, H. Gold), INTECH Open Access Publisher, Croatia, 2008.
- [13] V. Erol, "Araç Rotalama Problemleri için Popülasyon ve Komşuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması", Yıldız Teknik Üniversitesi, Sistem Mühendisliği, Yüksek Lisans Tezi, Sabancı Kütüphanesi, İstanbul, 2006.
- [14] C. Blum, A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM Computing Surveys, vol. 35, 3, pp. 268-308, 2003.
- [15] D. Korżinek, R. Skoczylas, System for a Stationary Robot that Draws Vector Images Based on Photographic Analysis. Polish-Japanese Academy of Information Technology, Faculty of Multi-Agent Systems and Robotics, Yüksek Lisans Tezi, Warsaw, 2005.