



# Tavlama Benzetimi Temelli Yaklaşım ile Kapasite Kısıtlı Araç Rotalama Optimizasyonu: Karadeniz Bölgesi Örneği

Ayten Yılmaz Yalçınar<sup>1\*</sup>

<sup>1\*</sup> Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, Sakarya, Türkiye, (ORCID: 0000-0001-8160-812X), [ayteny@sakarya.edu.tr](mailto:ayteny@sakarya.edu.tr)

(İlk Geliş Tarihi Aralık 2020 ve Kabul Tarihi Ocak 2021)

(DOI: 10.31590/ejosat.851540)

**ATIF/REFERENCE:** Yılmaz Yalçınar, A. (2021). Tavlama Benzetimi Temelli Yaklaşım ile Kapasite Kısıtlı Araç Rotalama Optimizasyonu: Karadeniz Bölgesi Örneği. *Avrupa Bilim ve Teknoloji Dergisi*, (22), 239-248.

## Öz

Günümüzün en kritik ve stratejik alanlarından birisi olan lojistik sektöründe işletmeler hem araç sayılarını hem de kat ettikleri mesafeyi en küçükleyerek maliyeti minimize etmek isterler. Bu kapsamda yaygın kullanım alanına sahip bir optimizasyon çeşidi olan Araç Rotalama Problemleri (ARP), farklı durumlara farklı teknikler ile etkin çözüm alternatifleri sunmaktadır. ARP'nin önemli bir türü olan Kapasite Kısıtlı Araç Rotalama (KKAR) problemlerinde geliştirilen algoritmanın tutarlı ve iyi sonuçlar vermesi önemlidir. Bu çalışmada birçok işletme için uygulanabilecek, değişken konum (lokasyon/şehir) sayısı ve değişken araç sayısı içeren problemlere yönelik farklı modeller oluşturularak bu modeller için belirli kısıtlar doğrultusunda en kısa mesafeyi bulmayı hedefleyen bir optimizasyon modeli sunulmak amaçlanmıştır. Uygulamada meta sezgisel çözüm yöntemlerinden biri olan Tavlama Benzetimi algoritması kullanılmıştır. Online erişilebilir hazır veritabanlarından elde edilen bir veri seti ile kurulan algoritma test edilmiş, daha sonra da Karadeniz Bölgesi illerine İstanbul merkezli bir depodan dağıtım gerçekleştirildiği senaryosu ile uygulama çalışması gerçekleştirilmiştir.

**Anahtar Kelimeler:** Araç Rotalama Problemi, Kapasite Kısıtlı Araç Rotalama, Tavlama Benzetimi Yöntemi.

## Capacity Restricted Vehicle Routing Optimization with an Annealing Simulation Based Approach: Black Sea Region Case

### Abstract

In the logistics sector, which is one of the most critical and strategic areas of today, businesses want to minimize costs by minimizing both the number of vehicles and the distance they travel. In this context, Vehicle Routing Problems (ARP), which is a widely used optimization type, offers effective solution alternatives with different techniques for different situations. It is important that the algorithm developed in Capacity Restricted Vehicle Routing (KKAR) problems, which is an important type of ARP, gives consistent and good results. In this study, it is aimed to present an optimization model aiming to find the shortest distance for these models in line with certain constraints by creating different models for problems involving variable location (location / city) number and variable number of vehicles, which can be applied to many businesses. Simulated Annealing algorithm, which is one of the meta-heuristic solution methods, was used in the application. The algorithm established with a data set obtained from online accessible databases was tested, and then the application study was carried out with the scenario where distribution is made to the provinces of the Black Sea Region from the distribution center, whose central warehouse is Istanbul.

**Keywords:** Vehicle Routing Problem, Capacity Limited Vehicle Routing, Simulated Annealing Method.

\* Sorumlu Yazar: [ayteny@sakarya.edu.tr](mailto:ayteny@sakarya.edu.tr)

## 1. Giriş

Araç Rotalama problemlerinde amaç bir veya daha fazla depodan çıkan müşteri taleplerinin uygun zaman ve mesafede müşterilere ulaştırmak için en uygun rotanın belirlenmesini sağlayan eniyileme problemidir. Ürünler müşteriye ulaştırılırken rotada bulunan noktalara hangi sıralama ile uğranılacağına karar verilmesi ve en az maliyet oluşturacak rotanın saptanması, lojistik problemleri altında Araç Rotalama Problemi (ARP) kapsamında incelenir. Araç Yönlendirme Problemi (ARP), müşterilere giden ve gelen malların çeşitli depolar arasında ve müşterilere ve müşterilerden depolara nasıl dağıtıldığını ele alan problemler setidir. ARP'nin amacı, malları müşterilere en kısa zamanda ve maliyette ulaştırmak için bir depoda yada depolarda bulunan araçlar için güzergah (lar) tasarlamaktır. ARP hedefi genellikle rota maliyetini en aza indirmektir. Amaç fonksiyonları, belirli varyantlar ve uygulamalara göre değişebilir. Araçlarla ilgili maliyetleri, kat edilen mesafeyi, ulaşım araçları sayılarını en aza indirmek, taşıma ya da seyahat süresi ve araç yükünün aşılması durumunda sistemi dengeye getirmek hedeflenir (Choosak Pornsing, 2014; Sitek ve Wikarek, 2019). Araç rotalamanın temel amacı kar sağlamak amacıyla yol alacakları toplam mesafenin minimize edilmeye çalışılmasıdır. Araç Rotalama Probleminin önemli uygulama alanlarından birisi coğrafi olarak farklı bölgelerde olan müşterilere bir veya birden fazla depodan dağıtım için görevlendirilen araçların en optimum şekilde dağıtım ve toplama rotalarının tasarlanmasıdır.

Araç rotalama problemiyle ilgili ilk çalışma 1959'da araç rotalama problemini ortaya çıkaran Dantzig ve Ramser tarafından araç yönlendirme problemi için ilk matematiksel programlama formülasyonu ve algoritmik yaklaşımı önermeleri ile yapılmıştır. Araç rotalama problemleri, ürün veya hizmetin belirli bir merkezden çıkarak müşterilere ulaştırıldığı ve araçların tekrar merkeze dönmelerini sağlayan rotaların belirlendiği kombinatoriyal optimizasyon problemidir. Lojistik ve tedarik zincirinin son aşaması olan ürün dağıtım sırasında ortaya çıkan optimizasyon problemleri olarak da ifade edilebilmektedir. En iyi çözümü elde etmek amacıyla ele alınan yaklaşım ise, problem çeşitlerine uygun kısıtların oluşturulmasıdır (Kumar ve Panneerselvam, 2012; Singh ve Sharma, 2021)

Araç rotalama problemlerinde olabilecek amaç fonksiyonları, toplam taşıma maliyetlerini en aza indirmek, seyahat süresi ve araç yükü göz önünde bulundurularak araç rotalarını dengede tutabilmek, taleplerin parçalı olarak dağıtılması sonucu oluşan ceza maliyetlerini en aza indirmek ve talebi karşılayacak araç sayısını minimize etmek olarak ifade edilebilir (Toth ve Vigo, 2002)

ARP teknikleri kullanım kolaylığı ve farklı durumlar için alternatif çözümler içermesi sebebiyle literatürde oldukça farklı alanlarda, yaygın uygulama imkanı ile yer bulmuştur. Kapasite Kısıtlı Araç Rotalama Problemi çözümünde kesin çözüm yöntemi olarak doğrusal programlama, dinamik programlama, dal ve sınır algoritması ve birçok analitik çözüm yöntemi uygulanmıştır. Dorigo ve Stutzle'a göre Sezgisel Yöntem çeşitlerinden olan meta-sezgisel yöntemlerin arama uzayının yüksek kaliteli çözümleri için daha uygundur (Dorigo ve Stutzle, 2004). Raúl Baños ve arkadaşları (2013), zaman pencereli araç rotalama problemlerine tavlama benzetimi tabanlı paralel çok amaçlı yaklaşım uygulaması yapmışlar ve paralel

yaklaşımın hesaplama maliyetinde sıralı versiyonlarla karşılaştırıldığında daha düşük sonuçlar elde etmişler aynı zamanda çözüm kalitesinde eksilme yaşamadıkları gibi çalışma zamanından da kazanç elde etmişlerdir (Baños vd., 2013). Yine paralel tavlama benzetimi yaklaşımı üzerine çalışan Dong ve arkadaşları ileriki yıllarda paralel tavlama benzetimi algoritmasını kullanarak eş zamanlı topla-dağıt araç rotalama problemi adlı çalışmaları ile paralel tavlama benzetimi algoritmasının çalışma süresinden literatürde bulunan diğer beş algoritmaya göre daha iyi bir sonuç elde etmişlerdir (Mu vd.,2016). Mahmudy, zaman pencereli araç rotalama probleminin optimizasyonu için geliştirilmiş tavlama benzetimi çalışması ile komşu çözümleri etkin şekilde keşfetmek amacıyla özel işlevler geliştirmiştir. Önerilen yaklaşım, literatürde mevcut olan iyi bilinen benchmark problemleriyle karşılaştırıldığında iyileştirilmiş tavlama benzetiminin 82.29 saniyelik çözüm süresi ile o umut verici sonuçlar elde edilmiştir (Mahmudy, 2014). Bir başka çalışmada, kapasite kısıtlı araç rotalama çalışmasında Tavlama Benzetimi algoritmasının diğer tüm algoritmalarından daha iyi performans gösterdiğini ve çoğu örnek için optimum çözümleri sunduğu ifade edilmektedir (Wei vd., 2017). Ayrı teslimat problemi eklenerek oluşturulan bir problemin çözümünde araç rotalama için etkili iki algoritma olan Tabu Araması ve Tavlama Benzetimi Algoritmaları ayrı ayrı uygulanmıştır. Elde edilen sonuçlara göre ise orta büyüklükteki problem örnekleri için Tabu Arama Algoritması ile daha iyi sonuçlar elde edilirken, büyük boyutlu problemler için ise Tavlama Benzetimi Algoritması makul çözüm süresinde daha iyi çözümler elde edilmiştir (Wang vd., 2017). Yine hibrit bir algoritma olan ve yeni bir yaklaşım olarak önerilen çalışmada ise Tavlama Benzetimi Tabanlı Simbiyotik Organizma Arama Optimizasyonu ile gezgin satıcı problemi üzerinde durulmuştur. Çözüm sonuçlarının diğer algoritmalarla rekabet edebilecek etkinlikte olduğuna ulaşılmıştır (Ezugwua vd., 2017). STCDARP için sezgisel bir çözüm yaklaşımı geliştiren Kızıloğlu (2017), hibrit sezgiseller ile bir çözüm sunarak Tavlama Benzetiminin başlangıç çözümler için uygunluğunu test etmiş. Ayrıca küçük boyutlu ARP için optimal sonuçlar elde edilmiştir. TB yardımıyla oluşturulan çözümler Tasarruf Algoritmasına göre daha iyi sonuç vermiştir (Kızıloğlu, 2017). Bir diğer çalışmada Eş Zamanlı Topla Dağıt Araç Rotalama Problemi için Karınca Kolonisi Algoritması ve Değişken Komşuluk Arama Algoritması'nın melez bir metasezgisel çalışma ile çözüm sunulmuştur. Bu çalışmada Karınca Kolonisi Algoritmasının uzun süreli hafıza yapısından ve Değişken Komşuluk Algoritmasının yoğun yerel arama sunan yapısından faydalanarak daha optimal bir sonuç almak hedeflenmiştir. Bu algoritma için önerilen test sonuçlarına baktığımızda istikrarlı ve iyi bir sonuç verdiği gözlemlenmiştir ve bunlar sayısal veriler ile çalışmada desteklenmiştir (Kaya,2017).

Görüldüğü gibi mesafe ve araç sayısında yaşanan iyileştirmeler ile maliyette de düşüklüğe sebep olan ARP teknikleri literatürde farklı tür ve alanlarda uygulama imkanı bulmuştur. Bu öneme istinaden bu makale çalışmasında Araç Rotalama Problemlerinde yer alan değişen lokasyon (şehir) sayıları ve değişen araç sayıları için farklı modeller oluşturularak bu modeller için belirli kısıtlar doğrultusunda en kısa mesafeyi bularak optimizasyonu sağlayan ve birçok işletme için uygulanabilecek genel bir araç rotalama problemi çözümü elde etmek amaçlanmıştır. Uygulama; koordinatların, araç sayısının ve şehir sayısının bilindiği modeller için uygulanmaktadır. Çalışma genel bir tanıtımın ve literatürün sunulduğu giriş bölümünün ardından yöntemlerin sunulduğu ikinci bölüm,

uygulama çalışmasının ve bulgularının aktarıldığı üçüncü bölüm, tartışma ve sonuçların yer aldığı dördüncü bölümü içermektedir.

## 2. Materyal ve Metot

Araç rotalama problemlerinde eniyileme çalışmaları; minimize edilmek istenen rota sayısı, toplam rota uzunluğu, rota süresi ve maksimize edilmek istenen müşteri memnuniyeti gibi bazı ölçütler temelinde gerçekleştirilmektedir. Araç rotalama problemlerinde kısıtların ve önceliklerin farklı olması birçok ARP tekniğinin ortaya çıkmasına sebep olmuştur. ARP, tarihsel süreci içinde gerçek hayat uygulamaları ile birlikte farklı yapılara evrilmiş ve bu yapılar kısıtlarına ve ele alınan problemin karakteristiklerine göre yeni araç rotalama problem türleri olarak literatürdeki yerlerini almışlardır. ARP'nin başlıca türleri; kapasite kısıtlı (KARP), mesafe ve kapasite kısıtlı (MKARP), zaman pencereli (ZPARP), geri toplamalı (GTARP), dağıtım ve toplamalı (DTARP), açık (AARP), çoklu depo (ÇDARP), bölünmüş teslimatlı (BTARP), periyodik (PARP), heterojen filolu (HFARP), simetrik ARP, asimetric ARP, homojen ARP, Statik ARP, Dinamik ARP ve bulanık araç rotalama problemi (BARP) olarak ifade edilebilirler (Karagül ve diğerleri, 2016, Koç, 2012, Erol 2006).

### 2.1. Kapasite Kısıtlı Araç Rotalama Problemi

İşletmelerin üretim sürecinde olumsuzluklara sebep olacak, kapasitesi dar olan kaynaklar bulunabilmektedir. Mevcut iş taleplerini zamanında, istenilen şekilde karşılayabilmek için yeterli kapasiteye sahip olmak gereksinimi duyarlar (Akçimen ve Antmen, 2019). Lojistik alanında hizmet veren işletmeler için de araçlardaki kapasite kısıtı, ele alınan problem için önemli bir unsurdur.

Kapasite Kısıtlı Araç Rotalama Problemi önceden belli olan müşterilerin taleplerini karşılamak için oluşturulan ve aynı zamanda bu taleplerin parçalanmasına izin vermeyerek ürün miktarının araç kapasitesine göre değiştiği yani kapasite kısıtının olduğu araç rotalama çeşididir. Müşteriler ve talepleri deterministiktir. Kapasite Kısıtlı Araç Rotalama Probleminde araçlar kapasite kısıtına sahip olduğundan talepler karşılanırken aracın kapasitesinin aşılmamasına dikkat edilmelidir. Araç sayısının sabit ve kapasiteli olduğu işletmeler için uygulanabilecek bir rotalama problemidir (Ekizler,2011).

Kapasite Kısıtlı Araç Rotalama Problemi Matematiksel Modeli ve formülasyonları aşağıda verilmiştir (Bozyerli vd. 2014, Keskinürk vd. 2016, Kulaç ve Çağıl, 2019):

$Q_i$ : araç kapasitesi,  $k_i \in K$   
 $K = \{k_1, k_2, \dots, k_m\}$  araç filosu  
 $N$ : müşteri veya durak sayısı,  
 $q_i$ :  $i$  düğümünün talep miktarı  
 $d_{ij}$ :  $i$  ve  $j$  müşterileri arasındaki mesafe,  
 $V_0$ : depo olmak üzere;

Değişkenler:

$$x_{ij}^k \begin{cases} 1, & \text{eğer } k \text{ aracı } i \text{ düğümünden sonra} \\ & j \text{ düğümünü ziyaret ederse} \\ 0, & \text{aksi taktirde} \end{cases}$$

$$y_i^k \begin{cases} 1, & \text{eğer } i \text{ düğümüne } k \text{ aracı} \\ & \text{hizmet verirse} \\ 0, & \text{aksi taktirde} \end{cases}$$

$$\text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \quad (2,1)$$

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

$$\sum_{j \in V} x_{ij}^k + \sum_{j \in V} x_{ji}^k = 1 \quad \forall i \in V, k \in K \quad (2,3)$$

$$\sum_{k=1}^K \sum_{j=1}^N x_{0j}^k = K \quad (2,4)$$

$$\sum_{j \in V} x_{0j}^k = 1 \quad \forall k \in K \quad (2,5)$$

$$\sum_{j \in V} x_{j,n+1}^k = 1 \quad \forall k \in K \quad (2,6)$$

$$x_{ji}^k = 1 \rightarrow y_i - q_i = y_j \quad \forall ij \in V, \forall k \in K \quad (2,7)$$

$$y_0 = Q, 0 \leq y_i \quad \forall i \in V \quad (2,8)$$

$$\sum_{i=1}^N q_i \sum_{j=0}^N x_{ji}^k \leq Q \quad k \in \{1, \dots, m\} \quad (2,9)$$

$$x_{ji}^k \in \{0,1\} \quad \forall ij \in V, \forall k \in K \quad (2,10)$$

Araç rotalama probleminde, araçların gideceği toplam mesafenin minimizasyonu için Denklem (2.1)'de gösterilen amaç fonksiyonu kullanılır. Kısıt (2.2) her  $i$ - $j$  bağlantısına bir aracın hizmet vermesi, kısıt (2.3) ise geri dönüşlerin engellenmesi ile ilgili kısıttır. Kısıt (2.4) depodan çıkan araç sayısı ile toplam araç sayısının eşit olması gerektiğini göstermektedir. Kısıt (2.5) ve (2.6) aracın depodan ve  $j$ . düğümünden bir defa çıkacağını gösterir. Kısıt (2.7) aracın  $i$ - $j$  düğümüne atanması durumunda  $i$  düğümünden  $j$  düğümüne geldiğinde geriye kalan kapasiteyi göstermektedir. Kısıt (2.8)'e göre aracın başlangıç kapasitesi  $Q$  olacak, kısıt (2.9)'a göre ise bir araca atanan müşterilerin toplam talebi aracın kapasitesini aşmasına izin verilmeyecektir. Kısıt (2.10) ise  $x_{ij}^k$  değişkeninin tam sayı olması ile ilgili kısıttır (Uyumaz, 2017).

### 2.2. Tavlama Benzetimi (TB) Yöntemi

KARP'nin çözümünde kullanılan metasezgisel yöntemlerden birisi olan Tavlama Benzetimi yönteminin çıkışı metaller için yapılan tavlama işlemine dayanmaktadır. Nasıl ki demir tavlama işlemi sırasında bir demir parçayı ısıtıp sonra soğumaya bırakılıyorsa, herhangi bir sayısal ölçüme de benzeri yaklaşım uygulanabileceği düşüncesi üzerine kuruludur. Bu yöntem S. Kirkpatrick ve arkadaşları C. D. Gelatt ve M. P. Vecchi tarafından 1983 yılında geliştirilmiştir. Tavlama Benzetimi Algoritması uygulaması kolay ve yerel optimumdan kurtulma olasılığı yüksek olan bir algoritmadır. Ayrık optimizasyon problemleri için çok fazla kullanılan Tavlama Benzetimi sürekli optimizasyon teknikleri için de kullanılabilir. Algoritmanın temel prensibi, iyi çözümü feda ederek yerine kötü çözümü kabul etme olasılığı olan  $p$  değerinin dinamik bir şekilde ilerleyen iterasyonlarda azalmasıdır (Şahin ve Eroğlu,2014). Tavlama Benzetiminin iki çeşidi vardır. İlki Hızlı Tavlama Benzetimi yöntemidir. TB, Tavlama Benzetiminin aksine düz fonksiyonlar için değil konveks fonksiyonlar için kullanılır. Hızlı Tavlama Benzetimi,

TB yöntemine göre daha hızlı sonuç vermektedir. İkinci Tavlama Benzetimi çeşidi ise Genelleştirilmiş Tavlama Benzetimi yöntemidir. Bu yöntemde tek bir sıcaklık kullanılır ve kombinatoriyal optimizasyon problemleri için daha uygundur. TB yöntemi uygulaması üç aşamadan oluşmaktadır. Bu aşamalar başlangıç durumuna getirme, seçim ve bitirme ile güncelleme aşamalarıdır. İlk aşamada TB parametreleri oluşturularak mevcut çözüm en iyi sonuç olarak atanır. İkinci aşamada elde edilen değerler baştaki parametreler ile karşılaştırılır. Bu karşılaştırma sonucunda çözüm ya bitirilir ya da devam eder. Üçüncü aşamada ise yeni çözüm en iyi çözüm olarak atanır ve algoritma ikinci aşamaya geri gider (Taş, 2007)

Standart bir TB algoritmasının temel adımları aşağıda verilmektedir (Erol, 2006):

Adım 1. Rasgele olarak bir başlangıç çözümü üret ve en iyi çözüm S olarak ata. Ayrıca t iterasyon indeksini 0 olarak ata.

Adım 2. Bir başlangıç sıcaklık değeri TB belirle ve mevcut sıcaklık değeri  $T_0 = T_B$  olarak ata.

Adım 3. En iyi çözümden hareketle rasgele komşu bir çözüm  $S^1 \in N(S)$  oluştur.

Adım 4. Üretilen  $S^1$  çözümüyle S çözümünün amaç fonksiyonu değerleri arasındaki farkı hesapla

$$(\delta = C(S^1) - C(S))$$

Adım 5. Eğer  $S^1$ , S'den daha iyi ( $\delta < 0$ ) ise S çözümüne  $S^1$  çözümünü ata.  $S^1$ , S'den daha kötü fakat mevcut  $T_t$  sıcaklığında  $e^{(-\delta/T)} > \theta$  sağlanıyorsa ( $\theta$ , 0 ile 1 arasında rasgele üretilmiş bir sayıdır). S çözümü ile  $S^1$  çözümünü yer değiştir. Yoksa  $S^1$  mevcut çözüm olarak muhafaza et.

Adım 6. T sıcaklığını Denklem (2.11) veya Denklem (2.12)'deki formüle göre değiştir.

$$T_t = R \cdot T_{t-1} \quad (0 < R < 1) \quad (2.11)$$

$$T_t = t / (1 + \beta t) \quad (\beta \text{ uygun küçük bir değerdir}) \quad (2.12)$$

Adım 7. Durdurma kriteri sağlanıyorsa araştırmayı durdur, aksi halde iterasyon indeksi t'yi bir artırarak üçüncü adıma git.

### 3. Uygulama ve Bulgular

Tek depolu kapasite kısıtlı araç rotalama problemlerinde, deponun tek bir noktada olması tüm araçların bu depodan ayrılıp tur tamamlandığında yine bu depoya dönmelerini zorunlu kılar. Bu durumda işletme, araç kapasitesi ve araç sayısını müşteri taleplerini karşılayacak şekilde belirlemelidir. Kapasitenin yetersiz, araç sayısının gereğinden az olması müşteri ihtiyacının karşılanamamasına sebep olurken, gereğinden fazla olması ise gereksiz maliyete ve israfa neden olmaktadır. Burada denge noktasını sağlamak önemli bir unsurdur. Denge noktası sağlandıktan sonra araçların katedikleri toplam mesafeyi de minimize edilerek katlanılan maliyet azaltılabilir. Bu çalışmada uygun araç sayısı ve araç kapasiteleriyle müşteri taleplerini eksiksiz karşılamak aynı zamanda toplam mesafeyi minimize etmek hedeflenmiştir. Bu amaçla, aşağıdaki aşamalar sırasıyla gerçekleştirilmiştir.

#### 3.1. Problem veri setinin elde edilmesi ve mesafeler matrisinin oluşturulması

Uygulama çalışmasında kullanılmak üzere, Uchoa ve arkadaşlarının üzerinde çalıştıkları ve optimum sonuçları elde ettikleri kapasite kısıtlı araç rotalama problemi olan X grubu veri setine erişilmiş, bu veri seti ile algoritma test edilmiştir. Veri setinden seçilen 10 tane örnek ile çözümler gerçekleştirilmiştir. Seçilen veri setleri aşağıdaki gibidir (Tablo1):

Tablo 1. Veri seti (<http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>)

|            |            |            |            |           |            |            |            |            |           |
|------------|------------|------------|------------|-----------|------------|------------|------------|------------|-----------|
| X-n101-k25 | X-n106-k14 | X-n110-k13 | X-n115-k10 | X-n120-k6 | X-n125-k30 | X-n129-k18 | X-n134-k13 | X-n139-k10 | X-n143-k7 |
|------------|------------|------------|------------|-----------|------------|------------|------------|------------|-----------|

Tabloda görülen “n” harfinin yanındaki sayı müşteri sayısını, “k” harfinin yanındaki sayı ise araç sayısını ifade etmektedir.

Çalışmada İstanbul ili depo noktası olarak seçilmiş, Karadeniz bölgesi illeri de dağıtım alanları olarak kullanılmıştır. Karadeniz bölgesinin seçilmesinin nedeni il sayısının fazlalığı ve bunun getireceği farklı boyutlarda deneme imkanınıdır. Uygulama çalışması ile belirlenen bölgeden farklı sayıda iller baz alınarak farklı boyutlarda problemlerle çalışma imkanı elde edilmiştir. İllerin koordinatları Karayolları Genel Müdürlüğü'nden alınmıştır. Veri setinde uygulanan koordinatlar kullanarak mesafeler matrisleri oluşturulmuştur ve bu matris MATLAB' de tavlama benzetimi modelinin verileri kısmına işlenmiştir. Veri setindeki tüm veriler için benzer çalışma gerçekleştirilmiştir. Örnek olarak X-n101-k25 veri setine yönelik MATLAB programı ekran görüntüsü Şekil 1'de verilmiştir.

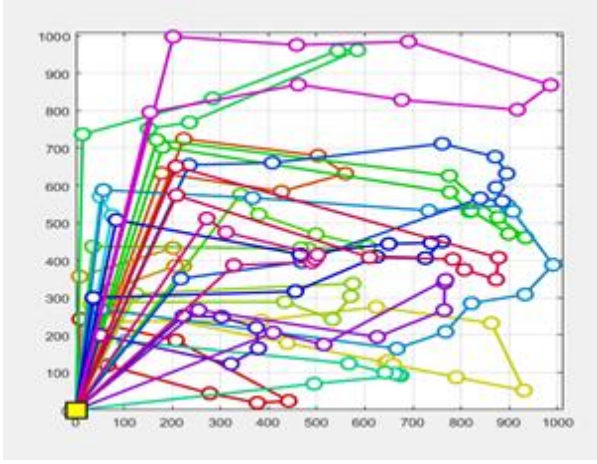
Şekil 1. X-n101-k25 veri seti MATLAB ekran görüntüsü

| Field | Value          |
|-------|----------------|
| I     | 101            |
| J     | 25             |
| r     | 1x101 double   |
| c     | 1x25 double    |
| xmin  | 0              |
| xmax  | 1000           |
| ymin  | 0              |
| ymax  | 1000           |
| x     | 1x101 double   |
| y     | 1x101 double   |
| x0    | 365            |
| y0    | 689            |
| d     | 101x101 double |
| d0    | 1x101 double   |
| eta   | 0.8000         |

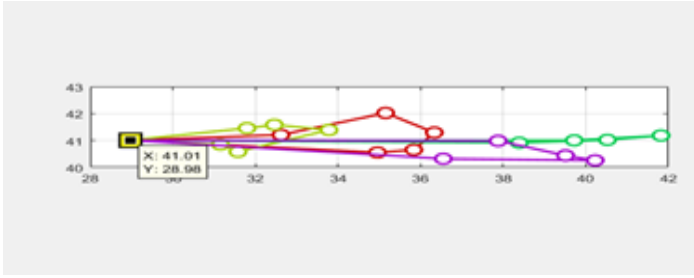
Şekil 1'de I şehir sayısını, j araç sayısını, r talep sayısını, c araç kapasitesini, x ve y min-max değerleri grafiğin alt üst sınırları, x ve y koordinatları, x0 ve y0 deponun koordinatları, d

mesafeler matrisi, d0 deponun müşterilere olan uzaklığı ve eta komşu seçme parametresidir. Algoritma çalıştırıldığında istenen model seçilip çözümü yapılmaktadır. X-n129-k18 veri setinin çözümünde elde edilen rotalara ait gösterim Şekil 2’de, Problem 26’ya ait araç rotası görünümü ise Şekil 3’te verilmiştir.

Şekil 2. X-n129-k18 veri setinin çözümünde oluşan rotalar



Şekil 3. Problem 26 araç rotası



### 3.2. Parametrelerin Belirlenmesi

Tavlama benzetimi algoritmasında kullanılan parametreler ve bunlara ek iterasyon sayısı parametresi farklı değerler alındığında farklı sonuçlar elde edilmektedir. Bu yüzden oluşturulan tavlama benzetimi algoritması X veri setinden orta ölçekli bir problem olan X-n129-k18 veri setinde farklı parametre değerleri ile denenip en iyi sonucu veren parametre değerleri seçilmiştir. Bahsedilen parametreler;

- Eta ( $\eta$ ): Komşu seçme fonksiyonunda kullanılan 0-1 arası rassal sayı
- Alpha ( $\alpha$ ): Soğutma Katsayısı 0-1 arası sayı
- IT<sub>1</sub>: Her sıcaklıkta gerçekleştirilecek iterasyon sayısı
- T<sub>0</sub>: Başlangıç Sıcaklığı
- IT<sub>2</sub>: İterasyon sayısı’dır.

Orta ölçekteki bu problem ile test edilmesi planlanan her bir değer, algoritmada 50 defa çalıştırılmıştır. Bunun sonucunda minimum mesafe, maksimum mesafe, ortalama mesafe ve mesafelerin standart sapması değerleri bulunmuştur. Sonuçlara göre standart sapmanın en düşük olduğu parametre değeri algoritmanın bu değerinde daha kararlı olduğunu gösterir. Seçim yapılırken standart sapma da göz önünde bulundurulmuştur.

Soğutma Katsayısının ( $\alpha$ ) Belirlenmesi:

Alpha parametresi için 0,5; 0,7 ve 0,99 değerleri verilmiştir. Sabit tutulacak parametre değerleri;

Eta=0,8; IT<sub>1</sub>= 100; T<sub>0</sub>=250; IT<sub>2</sub>=1000 olarak belirlenmiştir.

Tablo 2. Tavlama Benzetimi Algoritması alpha değeri test sonuçları

| Alpha | Min Mesafe | Ortalama Mesafe | Max Mesafe  | Standart Sapma |
|-------|------------|-----------------|-------------|----------------|
| 0,5   | 27286,2707 | 28020,73553     | 29169,86882 | 0,026211474    |
| 0,7   | 27041,2525 | 28248,77362     | 29746,99052 | 0,042745967    |
| 0,99  | 27092,9852 | 28012,27829     | 28817,55759 | 0,032817506    |

Tablo 2.’de görülen TB alpha değeri test sonuçları içerisinde ortalama mesafe ve standart sapma değerine bakılarak en uygun çözüm sonucunu veren 0,99 değeri alpha değeri olarak seçilmiştir.

Eta değerinin ( $\eta$ ) belirlenmesi:

Eta parametresi için 0,2; 0,5 ve 0,8 değerleri verilmiştir. Sabit tutulacak parametre değerleri;

Alpha=0,99; IT<sub>1</sub>=100; T<sub>0</sub>=250; IT<sub>2</sub>= 1000 olarak belirlenmiştir.

Tablo 3. Tavlama Benzetimi Algoritması eta değeri test sonuçları

| Eta | Min Mesafe | Ortalama Mesafe | Max Mesafe  | Standart Sapma |
|-----|------------|-----------------|-------------|----------------|
| 0,2 | 28167,5073 | 28909,68764     | 29699,04714 | 0,025672376    |
| 0,5 | 27294,877  | 28214,77305     | 29230,67298 | 0,032603347    |
| 0,8 | 27446,2635 | 28098,34372     | 29415,88078 | 0,023207069    |

Tablo 3.’de görülen TB eta değeri test sonuçları içerisinde ortalama mesafe ve standart sapma değerine bakılarak en uygun çözüm sonucunu veren 0,8 değeri eta değeri olarak seçilmiştir.

Her Sıcaklıkta Gerçekleştirilecek İterasyon Sayısının (IT<sub>1</sub>) Belirlenmesi:

Her sıcaklıkta gerçekleştirilecek iterasyon sayısı için 50, 100, 200 ve 300 değerleri verilmiştir. Sabit tutulacak parametre değerleri;

Alpha=0,99; Eta=0,8; T<sub>0</sub>=250; İterasyon Sayısı=1000 olarak belirlenmiştir.

Tablo 4. Tavlama Benzetimi Algoritması her sıcaklıkta gerçekleştirilecek iterasyon sayısı test sonuçları

| IT <sub>1</sub> | Min Mesafe | Ortalama Mesafe | Max Mesafe  | Standart Sapma |
|-----------------|------------|-----------------|-------------|----------------|
| 50              | 28178,2276 | 29117,44968     | 30159,40523 | 0,032256331    |
| 100             | 27446,2635 | 28081,92672     | 29415,88078 | 0,022636025    |
| 200             | 26550,7717 | 27292,86636     | 27868,01267 | 0,027190059    |
| 300             | 26095,2773 | 26871,84851     | 27648,23826 | 0,028899061    |

Tablo 4’de görülen TB sıcaklık iterasyon test sonuçları içerisinde ortalama mesafe ve standart sapma değerine bakılarak en uygun çözüm sonucunu veren 300 sayısı her sıcaklıkta gerçekleştirilecek iterasyon sayısı olarak seçilmiştir.

Başlangıç sıcaklık değerinin (T<sub>0</sub>) belirlenmesi:

Başlangıç sıcaklık değeri için 100, 175, 250 ve 300 değerleri verilmiştir. Sabit tutulacak parametre değerleri;

Alpha=0,99; Eta= 0,8; IT<sub>1</sub>=100; IT<sub>2</sub>=1000 olarak belirlenmiştir.

Tablo 5. Tavlama Benzetimi algoritması başlangıç sıcaklık değeri test sonuçları

| T0  | min mesafe | ortalama mesafe | max mesafe  | standart sapma |
|-----|------------|-----------------|-------------|----------------|
| 100 | 26916,5081 | 28010,99207     | 28932,49429 | 0,039073373    |
| 175 | 27000,1969 | 28071,1878      | 28949,59901 | 0,038152676    |
| 250 | 27092,9852 | 27986,95543     | 28609,91848 | 0,031942391    |
| 300 | 26947,2147 | 28087,7523      | 29281,22133 | 0,040606224    |

Tablo 5'te görülen TB başlangıç sıcaklık değeri test sonuçları içerisinde ortalama mesafe ve standart sapma değerine bakılarak en uygun çözüm sonucunu veren 250 değeri başlangıç sıcaklık değeri olarak seçilmiştir.

İterasyon sayısının (IT<sub>2</sub>) belirlenmesi:

İterasyon sayısı için 400, 600, 800 ve 1000 değerleri verilmiştir. Sabit tutulacak parametre değerleri;

Alpha=0,99; Eta=0,8; IT<sub>1</sub>=100; T<sub>0</sub>=250 olarak belirlenmiştir.

Tablo 6. Tavlama Benzetimi Algoritması İterasyon Sayısı test sonuçları

| İterasyon sayısı | min mesafe | ortalama mesafe | max mesafe  | standart sapma |
|------------------|------------|-----------------|-------------|----------------|
| 400              | 29175,0619 | 30010,77256     | 31304,04221 | 0,027847022    |
| 600              | 27746,7248 | 28838,17635     | 30499,90887 | 0,037847453    |
| 800              | 27039,7798 | 28265,17248     | 29429,19497 | 0,043353449    |
| 1000             | 27092,9852 | 27961,05342     | 28777,50697 | 0,03104562     |

Tablo 6'da görülen TB iterasyon sayısı test sonuçları içerisinde ortalama mesafe ve standart sapma değerine bakılarak en uygun çözüm sonucunu veren 1000 sayısı iterasyon sayısı olarak seçilmiştir. Testler sonucunda tavlama benzetimi algoritmasının parametre değerleri aşağıdaki gibi bulunmuştur:

Eta ( $\eta$ ):0,8; Alpha ( $\alpha$ ): 0,99; IT<sub>1</sub>: 100; T<sub>0</sub>: 250; IT<sub>2</sub>: 1000

### 3.3. Algoritmanın Test Edilmesi

Algoritmanın tutarlılığını ve verdiği sonuçların ne derece doğru olduğunu görmek amacıyla literatürde kullanılan X veri setinden 10 adet problem kullanılmıştır. X grubu veri setine ait problemlerin farklı boyutlarda olması, algoritmaların etkinliğini ve farklı boyutlardaki etkisinin görülmesine yardımcı olacaktır. Her bir problem algoritmada 10 defa çalıştırılmıştır. Böylelikle ortalama bir sonuç elde edilmiştir. Tablo 7' de sırasıyla problemin ismi, elde edilen ortalama mesafe sonucu, bilinen optimum sonuç ve standart sapma verilmiştir.

Tablo 7. Tavlama Benzetimi Algoritması Çözüm Sonuçları

| Problem    | Ortalama Mesafe | Bilinen Optimum | Yüzde Sapma |
|------------|-----------------|-----------------|-------------|
| X-n101-k25 | 28757,03922     | 27591           | -4,23%      |
| X-n106-k14 | 27238,38872     | 26362           | -3,32%      |
| X-n110-k13 | 13652,26869     | 14971           | 8,81%       |
| X-n115-k10 | 11298,81684     | 12747           | 11,36%      |
| X-n120-k6  | 11517,86568     | 13332           | 13,61%      |
| X-n125-k30 | 53000,53559     | 55539           | 4,57%       |
| X-n129-k18 | 26849,18296     | 28940           | 7,22%       |

|            |             |       |        |
|------------|-------------|-------|--------|
| X-n134-k13 | 10077,43232 | 10916 | 7,68%  |
| X-n139-k10 | 13087,59289 | 13590 | 3,70%  |
| X-n143-k7  | 13722,16945 | 15700 | 12,60% |

Tablo 7'deki elde edilen sonuçlara göre ilk 2 problemde tavlama benzetimi ile elde edilen sonuçlar bilinen optimum sonuçlara çok yakın değerler olarak görülmektedir. Diğer 8 problemde ise bilinen optimum sonuçtan daha iyi sonuçlar elde edilmiştir. Yüzde Sapma sütunundaki negatif sapmalar bilinen optimum sonuçtan daha yüksek değerler bulunduğu için negatif olarak ifade edilmiştir. Pozitif sapmalar ise bilinen optimum sonuçtan daha iyi sonuç elde edildiği için pozitif olarak ifade edilmiştir. Elde edilen sonuçlarla algoritmanın tutarlı ve bilinen optimumdan daha iyi sonuçlar verdiğini gözlemlenmiştir.

### 3.4. Algoritmanın Pratikte Uygulanması

Kurulan algoritmanın literatürdeki veri setleri ile testinin ardından bu makalenin uygulama kısmında kullanılan problem veri setleri ile pratikte bir uygulamada karşılığı görülmek istenmiştir. Bu amaçla İstanbul depo merkezi olmak üzere, dağıtım yerleri olarak Karadeniz Bölgesi illeri koordinatları kullanılmıştır.

#### 3.4.1. Karadeniz illeri için mesafeler matrisi oluşturulması

Karadeniz Bölgesi illerine ait koordinatları ve talep miktarları Tablo 8'de görülmektedir. Çalışmada kullanılan il talep bilgileri ise random sayı ataması ile elde edilmiştir.

Tablo 8: Karadeniz Bölgesi illeri koordinat ve talep bilgileri

| İLLER     | X         | Y         | TALEPLER |
|-----------|-----------|-----------|----------|
| İSTANBUL  | 41,00527  | 28,97696  | 0        |
| AMASYA    | 40,64991  | 35,83532  | 50       |
| ARTVİN    | 41,18277  | 41,818292 | 70       |
| BOLU      | 40,575977 | 31,578809 | 50       |
| ÇORUM     | 40,550556 | 34,955556 | 60       |
| GİRESUN   | 40,912811 | 38,38953  | 70       |
| GÜMÜŞHANE | 40,438588 | 39,508556 | 40       |
| KASTAMONU | 41,38871  | 33,78273  | 70       |
| ORDU      | 40,983879 | 37,876411 | 70       |
| RİZE      | 41,02005  | 40,523449 | 80       |
| SAMSUN    | 41,292782 | 36,33128  | 80       |
| SİNOP     | 42,02314  | 35,153069 | 30       |
| TOKAT     | 40,316667 | 36,55     | 80       |
| TRABZON   | 41,00145  | 39,7178   | 30       |
| ZONGULDAK | 41,456409 | 31,798731 | 60       |
| BAYBURT   | 40,255169 | 40,22488  | 60       |
| BARTIN    | 41,581051 | 32,460979 | 10       |
| KARABÜK   | 41,2061   | 32,62035  | 20       |
| DÜZCE     | 40,843849 | 31,15654  | 60       |

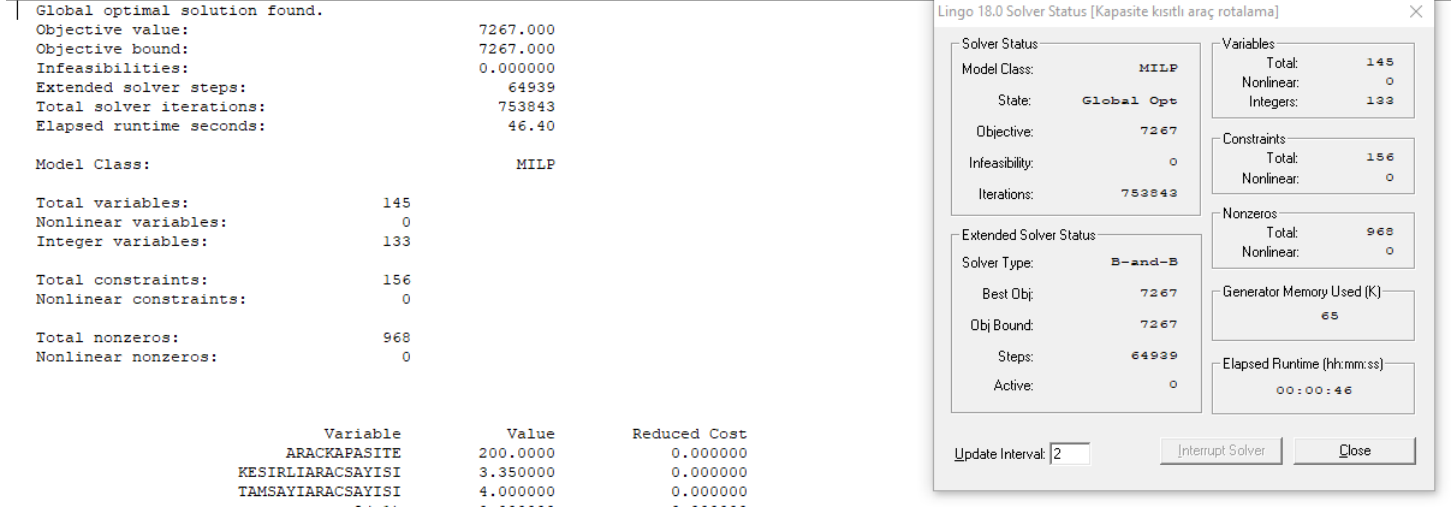
Eldeki bilgilerle uygun araç sayısı ve optimum mesafenin bulunması amacıyla aşağıda detayları verilen doğrusal programlama modeli kullanılmıştır.

### 3.5. Doğrusal Programlama ile uygun araç sayısı ve optimum mesafe bulunması

Ele alınan problemin optimum araç sayısı ve mesafe hesabı için kurulan matematiksel modeli farklı araç sayıları, kapasiteler ve müşteri sayıları kullanılarak elde edilen veri setlerinin

LINGO paket programında çalıştırılmasıyla yapılmıştır. Karadeniz bölgesinde yer alan 18 il birer azaltılarak problemler elde edilmiştir. LINGO’da çözdürülen bu veri setleriyle global optimum (GO) ve geçerli çözümlere (GÇ) ulaşılmıştır. Ayrıca belirlenen kapasite ve taleplerde kaç adet araç gerekliliği

sorgulanıp program çıktılarında sonuçları elde edilmiştir (Program çıktılarına ait bir örnek ekran görüntüsü Şekil 4’de verilmiştir). Tablo 8’de farklı boyutlardaki problemler ve LINGO çözümleri listelenmiştir.



Şekil 4: Örnek bir LINGO ekran görüntüsü

Tablo 9’da verilen çözümlerden de görüleceği gibi küçük boyuttaki problemlerin çözümü 2 ila 60 dakika arasında değişen sürelerde elde edilmiştir. Çözüm sonuçları global optimum (GO) çözüm olarak bulunmuştur. Büyük boyutta olan problemlerin çözümü ise 1 ila 22 saat arasında süren değişen süreler sonucunda durdurularak elde edilmiştir ve geçerli çözüm (GÇ) olarak tabloya aktarılmıştır. Tabloda görülen araç sayıları ise çözüm sonucu elde edilen, belirlenen kapasite ve talepler doğrultusunda gereken optimum araç sayısını göstermektedir.

|    |    |     |      |   |    |
|----|----|-----|------|---|----|
| 22 | 17 | 200 | 9449 | 5 | GÇ |
| 23 | 17 | 250 | 7784 | 4 | GÇ |
| 24 | 17 | 300 | 7133 | 4 | GÇ |
| 25 | 18 | 200 | 9554 | 4 | GÇ |
| 26 | 18 | 250 | 7739 | 4 | GÇ |
| 27 | 18 | 300 | 7183 | 4 | GÇ |

Tablo 9. Çözümde kullanılacak farklı boyuttaki problemler ve LINGO sonuçları

| Problem No | Müşteri Sayısı | Araç Kapasitesi i | Mesafe (LP) | LINGO Araç Sayısı | Çözüm Türü |
|------------|----------------|-------------------|-------------|-------------------|------------|
| 1          | 10             | 200               | 6757        | 4                 | GO         |
| 2          | 10             | 250               | 5914        | 3                 | GO         |
| 3          | 10             | 300               | 5218        | 3                 | GO         |
| 4          | 11             | 200               | 7267        | 4                 | GO         |
| 5          | 11             | 250               | 6014        | 3                 | GO         |
| 6          | 11             | 300               | 5707        | 3                 | GO         |
| 7          | 12             | 200               | 7738        | 4                 | GO         |
| 8          | 12             | 250               | 6391        | 3                 | GO         |
| 9          | 12             | 300               | 6029        | 3                 | GO         |
| 10         | 13             | 200               | 8062        | 4                 | GO         |
| 11         | 13             | 250               | 7002        | 4                 | GO         |
| 12         | 13             | 300               | 6117        | 3                 | GO         |
| 13         | 14             | 200               | 8685        | 5                 | GO         |
| 14         | 14             | 250               | 7227        | 4                 | GO         |
| 15         | 14             | 300               | 6154        | 3                 | GO         |
| 16         | 15             | 200               | 9390        | 5                 | GÇ         |
| 17         | 15             | 250               | 7669        | 4                 | GÇ         |
| 18         | 15             | 300               | 6615        | 3                 | GÇ         |
| 19         | 16             | 200               | 9446        | 5                 | GÇ         |
| 20         | 16             | 250               | 7725        | 4                 | GÇ         |
| 21         | 16             | 300               | 7055        | 4                 | GÇ         |

### 3.6. Belirlenen problemlerin tavlama benzetimi algoritması ile çözümü

Karadeniz bölgesinin illeriyle farklı boyutlarda oluşturulan problemler MATLAB’da oluşturulan tavlama benzetimi algoritması ile çözülmüştür. Tablo 10’da sonuçlar verilmiştir. Araç sayısına, araç kapasitelerine ve müşteri sayılarına göre araçların aldığı mesafeler değişiklik göstermektedir. Müşteri sayısı arttıkça katedilen mesafe artmaktadır. Aynı şekilde araç kapasiteleri arttıkça da mesafenin azaldığı gözlemlenmiştir. Araç sayıları ve kapasiteler orantılı şekilde verilmeye çalışılmıştır ve kaç aracın gerekli olduğu, kapasitenin ne kadar olması gerektiği konusunda karşılaştırma yapılabilecek şekilde değerler verilmiştir.

Tablo 10. Tavlama Benzetimi Algoritması Çözüm Sonuçları

| Problem No | Müşteri Sayısı | Araç Kapasitesi | Araç Sayısı | Mesafe(TB) |
|------------|----------------|-----------------|-------------|------------|
| 1          | 10             | 200             | 4           | 6757       |
| 2          | 10             | 250             | 3           | 5914       |
| 3          | 10             | 300             | 3           | 5218       |
| 4          | 11             | 200             | 4           | 7267       |
| 5          | 11             | 250             | 3           | 6014       |
| 6          | 11             | 300             | 3           | 5707       |
| 7          | 12             | 200             | 4           | 7738       |
| 8          | 12             | 250             | 4           | 6391       |
| 9          | 12             | 300             | 3           | 6029       |
| 10         | 13             | 200             | 4           | 8062       |
| 11         | 13             | 250             | 3           | 7002       |
| 12         | 13             | 300             | 3           | 6117       |

|    |    |     |   |      |
|----|----|-----|---|------|
| 13 | 14 | 200 | 5 | 8685 |
| 14 | 14 | 250 | 4 | 7227 |
| 15 | 14 | 300 | 3 | 6154 |
| 16 | 15 | 200 | 5 | 9390 |
| 17 | 15 | 250 | 4 | 7669 |
| 18 | 15 | 300 | 3 | 6615 |
| 19 | 16 | 200 | 5 | 9446 |
| 20 | 16 | 250 | 4 | 7725 |
| 21 | 16 | 300 | 4 | 6671 |
| 22 | 17 | 200 | 5 | 9449 |
| 23 | 17 | 250 | 4 | 7728 |
| 24 | 17 | 300 | 4 | 7075 |
| 25 | 18 | 200 | 5 | 9512 |
| 26 | 18 | 250 | 5 | 7739 |
| 27 | 18 | 300 | 4 | 7075 |

Araçların her birinin katettiği mesafe ve doluluk oranlarına bakmak için Problem 25,26 ve 27'yi ele alınarak aşağıdaki sonuçlara ulaşılmıştır.

$$Doluluk Oranı = \frac{Araçtaki toplam talep}{Araç Kapasitesi}$$

Problem 25 Çözüm Sonuçları:

- 1.Araç Rota:  
[İstanbul→Kastamonu→Bartın→Zonguldak→Düzce→İstanbul]
- 2.Araç Rota:  
[İstanbul→Rize→Artvin→Trabzon→Karabük→İstanbul]
- 3.Araç Rota:  
[İstanbul→Bolu→Ordu→Rize→İstanbul]
- 4.Araç Rota:  
[İstanbul→Sinop→Giresun→Gümüşhane→Bayburt→İstanbul]
- 5.Araç Rota:  
[İstanbul→Çorum→Tokat→Amasya→İstanbul]

Tablo 12. Problem 26 Çözüm Sonuçları

| Araç No | Mesafe (km) | Araç Kapasite | Araçtaki Toplam Talep | Doluluk oranı |
|---------|-------------|---------------|-----------------------|---------------|
| 1       | 1685        | 250           | 240                   | 96,00%        |
| 2       | 1107        | 250           | 250                   | 100,00%       |
| 3       | 2600        | 250           | 250                   | 100,00%       |
| 4       | 0           | 250           | 0                     | 0,00%         |
| 5       | 2347        | 250           | 250                   | 100,00%       |

Problem 27 Çözüm Sonuçları:

- 1.Araç Rota:  
[İstanbul→Samsun→Ordu→Giresun→Tokat→İstanbul]
- 2.Araç Rota:  
[İstanbul→Bolu→Düzce→İstanbul]
- 3.Araç Rota:  
[İstanbul→Trabzon→Rize→Artvin→Bayburt→Gümüşhane→İstanbul]

4.Araç Rota:  
[İstanbul→Çorum→Amasya→Sinop→Kastamonu→Karabük→Bartın→Zonguldak→İstanbul]

Tablo 13. Problem 27 Çözüm Sonuçları

| Araç No | Mesafe (km) | Araç Kapasite | Araçtaki Toplam Talep | Doluluk oranı |
|---------|-------------|---------------|-----------------------|---------------|
| 1       | 1973        | 300           | 300                   | 100,00%       |
| 2       | 524         | 300           | 110                   | 36,67%        |
| 3       | 2811        | 300           | 280                   | 93,33%        |
| 4       | 1767        | 300           | 300                   | 100,00%       |

## 4. Sonuç ve Tartışma

Araç rotalama problemi türlerinden olan kapasite kısıtlı araç rotalama probleminde müşteri ihtiyaçlarını karşılarken maliyeti minimize etmede araç sayısı ve kapasitesi önemli kavramlardır. Tüm kısıtları karşılamak için belirlemelerin yerinde ve tutarlı yapılması gerekmektedir. Matematiksel modelleme ile problem boyutu büyüdükçe problemin çözümü uzun süreleri bulmaktadır. Bu yüzden problemde metasezgisel çözüm yöntemlerinden olan Tavlama Benzetimi algoritması kullanılmıştır. 5 adet önemli parametrenin kullanıldığı algoritma ile orta ölçekli bir problemde belirlenen farklı parametreler denenerek çözdürülmüş ve en iyi sonuçları veren parametreler çözümde kullanılmak üzere seçilmiştir. Bu değerler:

- Eta ( $\eta$ ) (Komşu seçme fonksiyonunda kullanılan 0-1 arası rassal sayı): 0,8
- Alpha ( $\alpha$ )(Soğutma Katsayısı 0-1 arası sayı): 0,99
- IT<sub>1</sub> (Her sıcaklıkta gerçekleştirilecek iterasyon sayısı): 300
- T<sub>0</sub> (Başlangıç Sıcaklığı):250
- IT<sub>2</sub> (İterasyon sayısı):1000' dir.

Algoritmanın tutarlılığını ölçmek amacıyla literatürde bulunan farklı boyutta problem veri setleri ele alınarak Tavlama Benzetimi algoritması ile her problem 50 kez çalıştırılmış ve Tablo 14'de çıkan ortalama sonuçlar bilinen optimum sonuçlarla karşılaştırılmıştır.

Tablo 14. X grubu veri seti algoritma sonuçları ve bilinen optimum sonuçlar

| Problem    | Ortalama Mesafe | Bilinen Optimum |
|------------|-----------------|-----------------|
| X-n101-k25 | 28757,03922     | 27591           |
| X-n106-k14 | 27238,38872     | 26362           |
| X-n110-k13 | 13652,26869     | 14971           |
| X-n115-k10 | 11298,81684     | 12747           |
| X-n120-k6  | 11517,86568     | 13332           |
| X-n125-k30 | 53000,53559     | 55539           |
| X-n129-k18 | 26849,18296     | 28940           |
| X-n134-k13 | 10077,43232     | 10916           |
| X-n139-k10 | 13087,59289     | 13590           |
| X-n143-k7  | 13722,16945     | 15700           |

Bunun sonucunda literatürdeki sonuçlara ya çok yakın ya da daha iyi sonuçlar elde edildiği görülmüştür. Karadeniz bölgesi illerinin farklı boyutlarda kullanıldığı problem Lineer Programlama- LINGO paket programı ile elde edilen sonuçlar



ve MATLAB programı ile Tavlama Benzetimi algoritması çalıştırılarak elde edilen sonuçlar Tablo 30'da karşılaştırılmalı olarak sunulmuştur.

Tablo 15. Lineer Programlama ile Tavlama Benzetimi Algoritması sonuçları kıyaslama tablosu

| Problem No | Müşteri Sayısı | Araç Kapasitesi | Araç Sayısı | Mesafe(TB) | Mesafe (LP) | LINGO Gerekli Araç Sayısı | Çözüm Türü |
|------------|----------------|-----------------|-------------|------------|-------------|---------------------------|------------|
| 1          | 10             | 200             | 4           | 6757       | 6757        | 4                         | GO         |
| 2          | 10             | 250             | 3           | 5914       | 5914        | 3                         | GO         |
| 3          | 10             | 300             | 3           | 5218       | 5218        | 3                         | GO         |
| 4          | 11             | 200             | 4           | 7267       | 7267        | 4                         | GO         |
| 5          | 11             | 250             | 3           | 6014       | 6014        | 3                         | GO         |
| 6          | 11             | 300             | 3           | 5707       | 5707        | 3                         | GO         |
| 7          | 12             | 200             | 4           | 7738       | 7738        | 4                         | GO         |
| 8          | 12             | 250             | 4           | 6391       | 6391        | 3                         | GO         |
| 9          | 12             | 300             | 3           | 6029       | 6029        | 3                         | GO         |
| 10         | 13             | 200             | 4           | 8062       | 8062        | 4                         | GO         |
| 11         | 13             | 250             | 3           | 7002       | 7002        | 4                         | GO         |
| 12         | 13             | 300             | 3           | 6117       | 6117        | 3                         | GO         |
| 13         | 14             | 200             | 5           | 8685       | 8685        | 5                         | GO         |
| 14         | 14             | 250             | 4           | 7227       | 7227        | 4                         | GO         |
| 15         | 14             | 300             | 3           | 6154       | 6154        | 3                         | GO         |
| 16         | 15             | 200             | 5           | 9390       | 9390        | 5                         | GÇ         |
| 17         | 15             | 250             | 4           | 7669       | 7669        | 4                         | GÇ         |
| 18         | 15             | 300             | 3           | 6615       | 6615        | 3                         | GÇ         |
| 19         | 16             | 200             | 5           | 9446       | 9446        | 5                         | GÇ         |
| 20         | 16             | 250             | 4           | 7725       | 7725        | 4                         | GÇ         |
| 21         | 16             | 300             | 4           | 6671       | 7055        | 4                         | GÇ         |
| 22         | 17             | 200             | 5           | 9449       | 9449        | 5                         | GÇ         |
| 23         | 17             | 250             | 4           | 7728       | 7784        | 4                         | GÇ         |
| 24         | 17             | 300             | 4           | 7075       | 7133        | 4                         | GÇ         |
| 25         | 18             | 200             | 5           | 9512       | 9554        | 4                         | GÇ         |
| 26         | 18             | 250             | 5           | 7739       | 7739        | 4                         | GÇ         |
| 27         | 18             | 300             | 4           | 7075       | 7183        | 4                         | GÇ         |

Sonuçlar değerlendirildiğinde LINGO'da müşteri sayısı az olan yani küçük boyuttaki problemlerde çözüm süresi de kısa sürmüştür ve program optimum çözümü vermiştir. Müşteri sayısı arttıkça yani problem boyutu büyüdükçe çözüm çok uzun sürdüğü için belli süre bekletilip program durdurulmak zorunda kalmıştır ve geçerli çözümlere ulaşılmıştır. MATLAB'de çalıştırılan Tavlama Benzetimi algoritması ile çözülen küçük boyutlu problemlerin sonuçları elde edilen global optimum sonuçlarla aynı değerleri vermiştir. Tavlama Benzetimi Algoritmasının bazı problemlerde geçerli çözümlerden daha iyi sonuçlar vermesi ise lineer programlamada çok uzun süre çalışıp optimal çözüm elde edilemediği için durdurulması bir sebep olarak düşünülebilir. LINGO'daki çözüm sonucu elde edilen gerekli araç sayısı ise bazı problemlerde tavlama benzetimi algoritmasında kullanmak üzere belirlenen araç sayısından az bulunmuştur. LINGO'da elde edilen araç sayıları kullanılması gereken optimum araç sayılarıdır ve daha doğru bir sonuç vermektedir. Lineer programlama sonucu bazı problemlerde belirlenen araç kapasitesinde kullanılması gereken araç sayıları Tavlama benzetiminde kullanılan araç sayıları ile  $\pm 1$  fark olacak şekilde değişiklik göstermiştir. Tüm bu elde edilen çözümlerden belirlenen uygun parametrelerle Tavlama Benzetimi

algoritmasının kısa sürede elde edilen sonuçlarla tutarlı ve optimum sonuçlara yakın ya da eşit sonuçlar sunduğu görülmektedir.

## Kaynakça

- Akçimen, C., Antmen, Z. (2019). Kısıtlar Teorisinde Kapasite Kısıtı ve Bir Üretim İşletmesinde Uygulaması, Avrupa Bilim ve Teknoloji Dergisi, (15), 618-626. DOI: 10.31590/ejosat.545139
- Baños, R., Ortega, J., Gil, C., Fernández, A., & Toro, F. (2013). A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows, Expert Systems with Applications 40 (5), 1696–1707.
- Bozyer, Z., Alkan, A., Fırlalı, A. (2014). "Kapasite Kısıtlı Araç Rotalama Probleminin Çözümü için Önce Grupla Sonra Rotala Merkezli Sezgisel Algoritma Önerisi", Bilişim Teknolojileri Dergisi, 7(2). <https://dergipark.org.tr/en/pub/gazibtd/issue/6631/88031>
- Choosak, P. (2014). A particle swarm optimization for the vehicle routing problem. University of Rhode Island, Dissertation, Doctor of Philosophy in Industrial and Systems Engineering.
- Dorigo, M., Birattari, M. & Stutzle, T. (2006). Ant colony optimization, in *IEEE Computational Intelligence Magazine*, 1, 4, 28-39, doi: 10.1109/MCI.2006.329691.
- Ekizler, H. (2011). Araç rotalama probleminin çözümünde karınca kolonisi optimizasyonu algoritmasının kullanılması. Yüksek Lisans Tezi. İstanbul Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı.
- Erol, V. (2006). Araç rotalama sistemleri için popülasyon ve komşuluk tabanlı metasezgisel bir algoritmanın tasarımı ve uygulaması. Yüksek Lisans Tezi. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Ezugwua, A.E., Adewumi, A.O., Frincu, M.E. (2017). Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem, Expert Systems With Applications, 77, 189–210
- Karagül, 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, 2016.
- Kaya, C. (2017). Eş zamanlı topla dağıt araç rotalama problemi için karınca koloni sistemi ile güçlendirilmiş değişken komşuluk arama algoritması. Yüksek Lisans Tezi. Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı.
- Kızıloğlu, K. (2017). Stokastik talepli çok Depolu Araç Rotalama Problemi İçin Sezgisel Bir Çözüm Yaklaşımı. Yüksek Lisans Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü.
- Koç, Ç., (2012). Zaman bağımlı araç rotalama problemi. Yüksek Lisans Tezi. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü.
- Kulaç, S., Çağlı, G. (2019). Bir İç Lojistik Sistem Probleminin, Araç Rotalama Çözüm Yöntemleri ile İyileştirilmesi. International Journal of Engineering Research and Development, 11 (2), 528-541. DOI: 10.29137/umagd.495518
- Kumar, S.N., Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. Intelligent Information Management, 04, 03. [10.4236/iim.2012.43010](https://doi.org/10.4236/iim.2012.43010)

- Mahmudy, W.F. (2014). Improved simulated annealing for optimization of vehicle routing problem with time windows (VRPTW), *Jurnal Ilmiah KURSOR*. 7 (3), 109-116,
- Mu, D., Wang, C., Zhao, F. & Sutherland, J.W. (2016). Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm, *Int. J. Shipping and Transport Logistics* 8, 1, ,81–106.
- Singh, V.P., Sharma, K. (2021). Capacitated Vehicle Routing Problem with Interval Type-2 Fuzzy Demands. In: Kalamkar V., Monkova K. (eds) *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*. [https://doi.org/10.1007/978-981-15-3639-7\\_11](https://doi.org/10.1007/978-981-15-3639-7_11)
- Sitek, P., Wikarek, J. (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach, *Ann Oper Res* 273, 257–277, <https://doi.org/10.1007/s10479-017-2722-x>
- Şahin, Y., Eroğlu, A. (2014). Kapasite kısıtlı araç rotalama problemi için metasezgisel yöntemler: Bilimsel yazın taraması, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 19 (4), 337-355.
- Taş, O. (2007). Havayolu Şirketlerinde Uçuşların Atanması Probleminin Tavlama Benzetimi Yöntemi İle Çözülmesi. Yüksek Lisans Tezi. Çukurova Üniversitesi.
- Toth, P., Vigo, D. (2002). The Vehicle Routing Problem, *Society For Industrial And Applied Mathematics*. [https://epubs.siam.org/doi/pdf/10.1137/1.9780898718515\\_fm](https://epubs.siam.org/doi/pdf/10.1137/1.9780898718515_fm)
- Uyumaz, D. (2017). Kapasite kısıtlı araç rotalama probleminin paralel genetik algoritma ile çözümü. Yüksek Lisans Tezi. Karabük Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı.
- Wang, J., Jagannathan, A.K.R., Zuo, X., Murray, C.C. (2017). Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries, *Computers & Industrial Engineering*, 112, 84–98.
- Wei, L., Zhang, Z., Zhang, D., Leung, S.C.H. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints, *European Journal of Operational Research* 265, 843–859. <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/> <https://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Root/Uzakliklar/ilmesafe.xls>