



## Minimum Yayılan Ağaç (MYA) Problemi: Denizli İli Hafif Raylı Sistem Proje Önerisi için Minimum Mesafeli Hat Belirleme

Müge AKAY<sup>1</sup>, Ayşegül TUŞ<sup>\*2</sup>

<sup>1</sup> Pamukkale Üniversitesi, Sosyal Bilimler Enstitüsü, Sayısal Yöntemler Bölümü, Denizli, Türkiye

<sup>2</sup> Pamukkale Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İşletme Bölümü, Denizli, Türkiye

\*atus@pau.edu.tr

(Alınış/Received: 04.10.2021, Kabul/Accepted: 06.12.2021, Yayınlanma/Published: 31.01.2022)

**Öz:** Şehir içi ulaşımda hat (güzergâh) belirleme süreci, önemli bir konudur ve bu süreç doğru bir şekilde planlanırsa yapılan yatırım veya kurulan yeni sistem, trafikte iyileşme sağlayacaktır. Bu çalışmanın amacı, şehir içi ulaşımda rahatlık, kolaylık ve zaman tasarrufu sağlamak için Denizli Belediyesi tarafından karayolu ulaşım sistemine alternatif olarak önerilen hafif raylı sistem projesi kapsamında minimum mesafeli bir hat belirlemektir. Bu nedenle hat belirleme, graf yani ağ yapısındaki sistemlerin tasarlanmasında kullanılabilen bir Minimum Yayılan Ağaç (MYA) problemi olarak ele alınmıştır. Problemin çözümünde Prim, Kruskal algoritmaları ve matris yöntemi kullanılmıştır. Her üç yöntem ile optimal sonuç elde edilmiştir. Minimum MYA mesafesi 29,09 km'dir. Ancak söz konusu yöntemler için süreçler; işlem kolaylığı, iterasyon sayısı, karmaşıklıkların minimuma indirilmesi açısından değerlendirildiğinde, daha avantajlı olan matris yönteminin gerçek hayat MYA problemlerinde kullanılmasının daha uygun olacağı sonucuna varılmıştır.

**Anahtar kelimeler:** Minimum Yayılan Ağaç, Prim Algoritması, Kruskal Algoritması, Matris Yöntemi, Raylı Sistem

### Minimum Spanning Tree (MST) Problem: Minimum Distance Line Determination for Denizli Light Rail System Project Proposal

**Abstract:** The route determination process is an important issue in urban transportation and if it is planned correctly, the investment or the new system will provide improvement in traffic. The aim of this study is to determine the line with a minimum distance within the scope of the light rail system project proposed by Denizli Municipality as an alternative to the highway transportation system in order to provide comfort, convenience and time saving in Denizli city transportation. So the line determination has been considered as the Minimum Spanning Tree (MST) problem that can be used in the design of systems in graph ie network structure. Prim, Kruskal algorithms and matrix method have been used to solve the problem. Optimal results have been obtained with all three methods. Minimum MST distance is 29,09 km. However, when the processes for these methods are evaluated in terms of ease of operation, number of iterations, and minimization of complexity, it has been concluded that the matrix method, which is more advantageous, would be more appropriate to use in real life MST problems.

**Keywords:** Minimum Spanning Tree, Prim Algorithm, Kruskal Algorithm, Matrix Method, Rail System

## 1. Giriş

Ülkemizin en önemli sorunlarından biri ulaşımdır. Nüfus artışı, ekonomik büyüme, kentleşme ve sanayileşme ile birlikte her geçen gün daha çok aracın trafiğe çıkması, ulaşım sorununa neden olmaktadır. Özellikle sabah ve akşam iş giriş-çıkış saatlerinde, ulaşımın ana arterlerine yoğun bir talep olmakta ve insanlar, özel araç (otomobil), servis araçları, minibüs, dolmuş ve belediye otobüslerinde oldukça fazla zaman kaybetmektedir. Zaman zaman bu yollar üzerinde trafiği rahatlatmak üzere yapılan seviyeli kavşaklar, ciddi bir şebeke analizi yapılmadan inşa edildiği için geçici bir çözüm olmaktadır. Bu nedenle modern uygulamaları ve teknolojik gelişmeleri göz

Atıf için/Cite as: M. Akay, A. Tuş, "Minimum Yayılan Ağaç (MYA) problemi: Denizli ili hafif raylı sistem proje önerisi için minimum mesafeli hat belirleme," *Demiryolu Mühendisliği*, no. 15, pp. 111-124, Jan. 2022. doi: 10.47072/demiryolu.1004307

önünde bulundurarak verimli ve toplumsal faydayı sağlayan ulaşım sistemleri geliştirilmelidir. Böylece daha hızlı, konforlu, güvenli ve ekonomik ulaşım hizmeti sağlanabilir. Raylı ulaşım sistemi, bu şartları sağlayan karayolu taşımacılığına alternatif modern bir toplu taşımacılık sistemidir. Arazi kullanımı, çevre sağlığı, enerji verimliliği, kaza riski, personel istihdamı, taşıma kapasitesi bakımından daha avantajlı olan bu sistem, ilk yatırım maliyeti yüksek olsa da işletme giderleri hesaba katıldığında uzun vadede daha ekonomiktir. Bu sistem sayesinde lastik tekerlekli araçlara olan talebin düşmesi, şehir içi trafik probleminin çözümüne olumlu katkıda bulunur [1, 2, 3]. Hafif raylı sistemler, elektrikle çalışan bir demiryolu sistemidir [4]. Metronun yüksek kapasitesini ve teknolojisini kapsamakla beraber tramvayın da yüzeyde gidebilme özelliğine sahiptir [5]. Bu sistemin amacı, hızlı ve güvenli bir şekilde varılmak istenen noktaya ulaşımı sağlayacak bir ulaşım altyapısını kurmak ve yolculuk taleplerini karşılayacak durakları birbirine bağlamaktır.

Raylı sistem için planlanan projelerde, özellikle hat belirlerken çeşitli kriterlerin değerlendirilerek dikkate alınması, ulaşımın iyileşmesi açısından önemlidir. Bu konuda yapılan çalışmalar incelendiğinde Gerçek vd. [6]; finansal (işletme ve bakım maliyetleri, ağ yapım maliyetleri, raylı araç satın alma maliyetleri), ekonomik (karayolu taşıtı işletme ve satın alma maliyetleri, trafik kazası maliyetleri, toplam yolculuk süresi, çevresel maliyet), sistem planlama (kapasite, erişilebilirlik, entegrasyon, alternatif ağ sisteminin ana plana uygunluğu), politika (raylı ulaşım ağının esnekliği, kamulaştırma zorluğu, yapım süresi, mevcut ağ kapasitesinin etkin kullanımı), Ludin vd. [7]; işyerine maksimum rota, minimum rahatsızlık, maksimum hareketlilik, stratejik konumu bağlama, maksimum ağ, asgari arazi edinimi, yapılandırılabilirlik, maksimum etkinlik, Alkubaisi [8], erişilebilirlik (seyahat süresi, arazi kullanımı), emniyet (siyah nokta sayısı, kavşak sayısı, hizalama), çevre (gürültü-titreşim, estetik görünüm), ekonomik (yol uzunluğu, inşaa maliyeti), nüfus gezileri (nüfus yoğunluğu, kabul, gezi sayısı), güvenlik, Kalamaras vd. [9], Hamurcu ve Eren [10, 11, 12]; ekonomik (inşaa maliyeti, kamulaştırma), çevresel etki (arazi yapısı, hassas bölgeler), sosyal etki (istihdam ve eğitim alanlarına erişim, estetik ve görsel etki, nüfus yoğunluğu, alışveriş ve yerleşim alanlarına erişim, kamu hareketliliği, erişilebilirlik), mühendislik (geliştirme ve iyileştirme yeteneği, toplam seyahat zamanı, ulaşım entegrasyon, talepleri karşılama düzeyi), Sarımehmet vd. [13]; yolcu memnuniyeti (hizmet süresi, hizmet sıklığı, kapasite, durak sayısı, direkt gidiş, fiyat), sosyallik (iş ve eğitim bağlantısını sağlaması, alışveriş-yerleşim yerlerine erişimin sağlaması), çevre (trafik, gürültü kirliliği), Banai [14], Brunner vd. [15]; iş merkezleri, düşük gelirli nüfusa hizmet veren yerler, nüfusun yoğun olduğu yerler, işletme ve sermaye maliyeti, toplu taşıma odaklı gelişme potansiyeli, mevcut yolları paylaşma imkanı, trafik problemlerinin olduğu bölgelere hizmet verme, hassas ortamlardan ve tehlikeli bölgelerden kaçınma, çekici yerler, Watanabe vd. [16]; eğitim, kontrol noktası, konut yoğunluğu ve yollara olan mesafe kriterlerini dikkate almıştır.

Bu çalışmada, şehir içi ulaşımı iyileştirmeye yönelik Denizli ilinde yapılması planlanan hafif raylı sistem için minimum mesafeli bir hat belirlenmesi amaçlanmıştır. Hafif raylı sistemde hat belirleme problemini çözmek için öncelikle bir grafa, grafi oluşturmak için ise duraklara ihtiyaç vardır. Bunun için Denizli Belediyesi'nin ulaşımı kolaylaştırmak amacıyla başlattığı "Hafif Raylı Sistem Projesi" dikkate alınmıştır. Denizli Belediyesi, karayolu ulaşım sistemine alternatif olarak önerdiği hafif raylı sistem projesi kapsamında yöneticiler, yolcular ve çevresel faktörleri dikkate alarak Denizli Ulaşım Ana Planı'ndaki koşullar çerçevesinde şehir içi hat alternatiflerini belirlemiştir. Bu çalışmada, Servergazi Devlet Hastanesi-Pamukkale Üniversitesi Kalp Merkezi ana hattı esas alınıp bu hattı içine alan bölgedeki 21 merkezi yer, durak olarak varsayılmıştır. Graf oluşturulduktan sonra hafif raylı sistem için minimum mesafeli hat belirleme problemi, Minimum Yayılan Ağı (MYA) olarak ele alınmıştır.

MYA problemi; ulaşım, lojistik, iletişim ağları tasarımı, su, doğalgaz, elektrik, telefon, internet hatlarının yerleştirilmesi, görüntü işleme, taksonomi, küme analizi, devre tasarımı, kablosuz telekomünikasyon ağ tasarımı, dağıtım sistemleri, sosyo-coğrafi alanların bölgeselleştirilmesi,

alanların homojen, bitişik bölgelere gruplanması, yerleştirilmesi dahil olmak üzere birçok alanda uygulanmıştır [17-18]. Örneğin, yayın probleminde, aynı mesajların iletişim ağı içindeki tüm bilgisayarlara iletilmesi gerekir. Bu tür bir iletişim ağı, MYA problemi olarak modellenilebilir [19]. MYA problemi için çok sayıda algoritma geliştirilmiştir. İlk algoritma, Boruvka (1926) tarafından geliştirilmiştir. Boruvka, çalışmasında Moravia şehri için etkili bir elektrik dağıtım ağı tasarlamayı amaçlamıştır. Prim algoritması, ilk olarak Jarnik (1930) tarafından icat edilmiş, daha sonra Prim (1957) tarafından yeniden formüle edilmiştir. Kruskal algoritması, Kruskal (1956) tarafından geliştirilmiştir. Boruvka (1926) ve Kruskal (1956) gibi iyi bilinen klasik algoritmalar, MYA'yı hesaplamak için ayrık küme birleştirme algoritmalarıdır. Prim (1957) bunu ikili yığın, d-yığın ve f-yığın ile gerçekleştirmiştir. Dijkstra (1959), Sollin (1961), Yao (1975), Cheriton ve Tarjan (1976), Fredman (1987) MYA problemi için farklı algoritmalar önermiştir. Pierce (1975), MYA problemi ile ilgili tüm klasik algoritmaları analiz etmiştir. Haymond, Jarvis ve Shier (1984), MYA algoritmaları için çeşitli hesaplama yöntemleri geliştirmiştir. Fredman ve Tarjan (1985), Prim, Kruskal ve Boruvka algoritmalarından esinlenerek Fibonacci yığınlarını kullanmış ve düğümler arasında daha az iterasyon sayısı veren bir algoritma sunmuştur. Matsui (1994), MYA problemi için doğrusal bir zaman algoritması oluşturmuştur. Yapılan çalışma daha önce Cheriton ve Tarjan (1976)'ın çalışmasına yapısal anlamda benzese de Matsui (1994)'nin algoritması, faaliyetleri yeniden düzeltme zorunluluğu olmadığı için kullanımı daha kolay olup Prim algoritmasına göre daha düşük yayılmayla sonuçlanmıştır. Bader (2006), düzensiz yapıli graflarda kolayca uygulanabilen MYA algoritmasını, Osipov (2009) ise Filter-Kruskal algoritmasını önermiştir [20]. Donkoh [21], Batı Afrika gaz boru hattı projesi için optimum bir boru hattı bağlantısı elde etmek için Prim algoritmasını ve Steiner ağaç algoritmalarını faktör derecelendirme yöntemiyle uygulamış, elde ettiği sonuç ile orijinal ağırlıktan yaklaşık %10,3 azalma sağlamıştır. Arogunde [22], Nijerya'daki eyaletlerden Odeda yerel yönetiminde yerel bir erişim ağını modellemek için Prim algoritmasını kullanmıştır. Çevik vd. [23], MYA problemini kullanarak İç Anadolu Bölgesi'nde bir kargo firmasının dağıtım güzergahını belirlemiştir. Gitonga [24], Kenya'daki Chuka Üniversitesi'nde bir yerel alan ağı tasarlayarak, MYA problemi için Prim algoritmasını uygulamış ve fiber optik ağa sahip düğümler tarafından temsil edilen çeşitli üniversite binaları ara bağlantısının toplam maliyetini en aza indirmeyi başarmıştır. Biswas vd. [20], MYA problemi için döngü algılama tabanlı bir algoritma önermiştir. Bu algoritmanın en büyük avantajı, ağırlıklı grafin kenarlarını sıralama ihtiyacını ortadan kaldırmasıdır. Akpan vd. [25], MYA probleminde Kruskal algoritmasını kullanarak bir yer altı kablo tesisatı için kablo kanallarının üç besleme ucundan ihtiyaç duyulan bir inşaat sahasındaki dört konuma gönderilmesindeki ulaştırma sorununa odaklanmıştır. Marpaung, F. [26], MYA problemlerini çözmek için Prim ve Boruvka algoritmalarını karşılaştırmış ve Prim algoritmasının daha verimli olduğunu göstermiştir. Rachmawati ve Pakpahan [27], MYA probleminin çözümünde Kruskal ve Boruvka algoritmalarını kullanarak karşılaştırmalı bir analiz yapmıştır. Yapılan testler sonucunda Kruskal algoritmasının MYA'yı bulmada Boruvka'dan daha etkili olduğu görülmüştür. Jahan vd. [28], MYA'yı belirlemek için Prim algoritmasından esinlenerek yeni bir yaklaşım olan matris yöntemini önermiştir. Ayrıca derece kısıtlı MYA, olasılıksal MYA, kuadratik MYA, minimum Steiner ağaç problemi, kapasiteli MYA, genelleştirilmiş MYA ve bulanık MYA, MYA'nın uzantıları olan NP-zor problemlerdir ve bu problemleri çözmeyi amaçlayan polinom zamanlı algoritmik yaklaşımlar yoktur [18-19].

Hat belirleme problemi için graf teorisi uygulamalarından MYA kullanılarak yapılan çalışma sayısı oldukça sınırlıdır. Çevik vd. [23], bir kargo firmasının İç Anadolu bölgesindeki iller ve bu illerin ilçeleri arasındaki dağıtım hattını MYA ile belirlemiştir. Shahin ve Jaferi [29], Tahran'dan İran'ın il merkezlerine araba aküsü tedarik problemi için bir graf geliştirmiş, sezgisel Prim algoritması kullanarak bu grafin MYA'sını elde etmiştir. Problem, metasezgisel genetik algoritma ile de çözülmüş ve sonuç, Prim algoritması ile karşılaştırılarak büyük problemler için genetik algoritmanın kullanılabileceği gösterilmiştir. Karayollarındaki yük vagonlarının yoğunluğunu azaltmak için karayolu trafiği ve kazaları, yakıt tüketimi ve müşteri teslim süresinin azaltılabileceği bir senaryo önerilmiştir. Zhou vd. [30], bireysel ilgi ve ihtiyaçlara dayalı en uygun

turizm rotasını elde etmek için MYA'yı kullanmıştır. Hem MYA problemi hem de MYA probleminin çözümünde kullanılan algoritmaların gerçek hayatta uygulanabilirliğini göstermesi bakımından yapılan çalışmanın literatüre katkı sağlayacağı düşünülmektedir. Çalışma, matris yönteminin ilk kez Türkçe literatüre kazandırılması açısından da önem taşımaktadır. Ayrıca bu problemin çözümü ile elde edilen MYA yapısı, hafif raylı sistem projesi için minimum mesafeli bir hat belirleme konusunda Denizli Belediyesi'ne bir öneri niteliğindedir. Önerilen hattın gerçek hayatta başarı ile uygulanabildiği takdirde Denizli şehir içi ulaşım problemini büyük oranda çözeceği düşünülmektedir.

Çalışma şu şekilde düzenlenmiştir. İkinci bölümde MYA problemi ve bu problemin çözümünde kullanılan Kruskal, Prim algoritmaları ve matris yöntemi açıklanmıştır. Uygulama bölümünde Denizli ilinde yapılması önerilen bir hafif raylı sistem için MYA olarak ele alınan hat belirleme problemi, ikinci bölümde açıklanan yöntemler ile çözülmüş ve karşılaştırmalı analiz yapılmıştır. Son bölümde ise sonuçlar yorumlanmış ve önerilerde bulunulmuştur.

## 2. Metot

MYA, klasik graf teorisinde iyi bilinen şebeke optimizasyon problemlerinden biridir [19]. Graf teorisi, bir toplulukta bulunan nesnelere arasındaki ilişkileri modeller. Graf, düğümler ve bu düğümleri birbirine bağlayan kenarlardan oluşan bir tür ağ yapısıdır [31]. Bir graf, her bir kenarı bir sayı (mesafe, maliyet, zaman, miktar gibi) ile ilişkilendirilmişse ağırlıklı, her bir kenarın belirli bir yönü varsa yönlü, belirli bir yönü yoksa da yönsüz bir graftır. Bir graf üzerinde işlem yaparken genelde işlem kolaylığı sağlamak ve güçlü bir temsili veri elde etmek için ağaç yapısına çevrilmektedir. Ağaç yapısı, sahip olduğu düğümler arasındaki kenarlar ile herhangi bir döngü oluşturmayan, yönsüz ve ağırlıklı bir graftır [32]. Diğer bir ifade ile herhangi bir grafın "yayılan ağaç" yapısındaki gösteriminde, çizilen ağaç hiçbir döngü olmaksızın bütün düğümleri kapsamaktadır. Bir graftan birden fazla yayılan ağaç yapısı çıkabilir. Bu ağaç yapılarından, kenar ağırlıklarının toplamı en düşük olan ağaç ise MYA olarak adlandırılmaktadır. Yani bir graf için MYA, diğer her bir kapsayan ağacın ağırlığından daha az veya bu ağırlığa eşit olan bir kapsayan ağaçtır [28]. Bir grafın bütün düğümlerini kapsayan MYA yapısı, graftaki  $n$  adet düğüme sahipse MYA yapısında  $n-1$  kadar da kenar var demektir [32].

MYA, ağacın ağırlığının kenarlarının ağırlıklarının toplamına karşılık geldiği, ağırlıklı, yönsüz ve bağlantılı bir grafın tüm düğümlerini kapsayan minimum ağırlıkta bir ağaç elde etme problemi ile ilgilenir [20]. MYA probleminin çözümü için bir önceki bölümde de anlatıldığı gibi çok sayıda algoritma vardır. Bu algoritmalar en uygun sonuca ulaşmak için her adımda uygulanacak belirli bir stratejiyi izler ve her adımda, bir sonraki adımları düşünmeden sadece bulunduğu durumdaki en iyi seçeneğe yönelir [33]. Bu çalışmada sadece uygulamada kullanılacak olan Prim, Kruskal algoritmaları ve matris yöntemi açıklanmıştır.

### 2.1. Prim algoritması

Prim algoritması, matematikçi bilim adamı Vojtěch Jarník (1930) tarafından ortaya atılmış, daha sonra Robert C. Prim (1957) ve Edsger W. Dijkstra (1959) tarafından tekrar keşfedilmiştir. Prim algoritmasında aşağıdaki adımlar izlenir [34-35]:

Adım 1: Graftan rastgele bir düğüm seçilerek ağaç oluşturmaya başlanır. Yani MYA başlangıçta sadece seçilen düğümü içerir.

Adım 2: Bu düğümü henüz ağaçta olmayan düğümlere bağlayan kenarlardan minimum ağırlıktaki kenar bulunur ve ağaca eklenir.

Adım 3: Graftaki tüm düğümler ağaçta oluncaya kadar 2. adım tekrarlanır. Bu işlemler yapılırken döngü oluşmamasına dikkat edilmelidir. Tüm düğümler birbirine bağlandığında ağaçtaki kenar sayısı, düğüm sayısının 1 eksiği yani  $n-1$  kadardır ve algoritma sonlandırılır.

## 2.2. Kruskal algoritması

Kruskal algoritması, Joseph Kruskal (1956) tarafından önerilmiştir. Prim algoritmasından tek farkı bir sonraki kenarı nasıl seçtiğidir. Bu algoritmanın adımları aşağıdaki gibidir [33]:

Adım 1: Graftaki bütün kenarlar ağırlıklarına göre küçükten büyüğe doğru sıralanarak S kenarlar kümesi oluşturulur.

Adım 2: Bu kümeden henüz ağaçta olmayan minimum ağırlıklı kenar seçilir. Bu kenarın ağaca katılması, bir döngü oluşmasına sebep olmuyorsa ağaca dahil edilir. Seçilen kenar, bir düğüm oluşmasına sebep olmuyorsa ağaca dahil edilir ve S'den çıkarılır.

Adım 3: Ağaçtaki kenar sayısı, düğüm sayısının 1 eksiği (n-1)'e ulaşana kadar 2. adım tekrarlanır. En son kalan tek ağaç, grafin MYA yapısını temsil eder. S boş küme ve n adet toplam düğüm sayısına karşılık, oluşturulmakta olan MYA yapısına n-1 sayıda kenar eklenmiş ise işlem sonlandırılır.

Prim ve Kruskal algoritmaları ile aynı MYA bulunur. İki algoritma arasındaki fark, Prim algoritmasında eklenecek kenarın ağaca bağlı olması gerekirken Kruskal algoritmasında kenarların aralarında bağlantı olup olmadığına bakılmaksızın yayılan ağaca eklenmesidir [36].

## 2.3. Matris yöntemi

Matris yöntemi, Jahan vd. (2020) tarafından Prim algoritmasından esinlenerek geliştirilmiştir. Yöntemin adımları aşağıdaki gibidir [28]:

Adım 1: Öncelikle düğümler arasındaki kenar ağırlıklarını (mesafeleri) gösteren bir ağırlık (uzaklık) matrisi (D) oluşturulur. n düğümlü bir ağda bağlantısız küme (DC) = {1, ..., k, s, j, ..., n} olarak tanımlanır. Tablo 1'de gösterilen  $D = [d_{ij}]$  ağırlık matrisinde  $d_{ij}$ , i ve j düğümleri arasındaki kenar ağırlığını temsil eder.

**Tablo 1.** Ağırlık matrisi (D)

	1	2	3	...	n
1	-	$d_{12}$	$d_{13}$	...	$d_{1n}$
2	$d_{21}$	-	$d_{23}$	...	$d_{2n}$
3	$d_{31}$	$d_{32}$	-	...	$d_{3n}$
...	...	...	...	...	...
n	$d_{n1}$	$d_{n2}$	$d_{n3}$	...	-

Adım 2: Başlangıç için rastgele seçilen düğüm 1, bağlantılı küme (C)'nin elemanı, ağdaki diğer tüm düğümler ise bağlantısız küme (DC)'nin elemanları olarak düşünülür. 1. iterasyonda, Tablo 2'de görüldüğü gibi ilk satıra düğüm 1 eklenir.

**Tablo 2.** 1. iterasyon  $C = \{1\}$

	1	2	3	...	k	...	n
1	-	$d_{12}$	$d_{13}$	...	$d_{1k}$	...	$d_{1n}$

Adım 3: Düğüm 1'e bağlantısı olan diğer tüm düğümler arasından minimum ağırlık belirlenir. 2. iterasyonda,  $d_{1k}$ 'nin minimum ağırlık olduğu varsayılarak ikinci satıra düğüm k eklenir ve Tablo 3 oluşturulur. Bu durumda k, C'nin elemanı olup DC'den çıkarılır. Tablo 3'te döngüden kaçınmak için  $d_{k1}$  ağırlığı göz ardı edilir ve (-) konulur.

**Tablo 3.** 2. iterasyon  $C = \{1, k\}$ 

	1	2	3	...	k	...	n
1	-	$d_{12}$	$d_{13}$	...	$d_{1k}^*$	...	$d_{1n}$
k	-	$d_{k2}$	$d_{k3}$	...	-	...	$d_{kn}$

Adım 4: Bu işlem, grafın tüm düğümleri  $C'$ 'de olana kadar tekrarlanır. Bir ağda  $n$  adet düğüm varsa, süreci tamamlamak için  $n - 1$  iterasyona ihtiyaç vardır. Genel olarak, herhangi bir iterasyonda,  $C'$ 'deki  $s$  düğümünden en kısa  $d_{sj}$  mesafesine sahip bağlı kümeye girmek için  $j$  seçilirse ve  $i$ ,  $C'$ 'de herhangi bir düğüm ise, sonraki iterasyonda Tablo 4'te gösterildiği gibi  $d_{sj}$  dışında tüm  $d_{ij}$  ve  $d_{ji}$  mesafeleri için (-) konulur. Böylece, her iterasyonda grafın tüm düğümleri  $C'$ 'de olana kadar bir düğüm dahil edilir.

**Tablo 4.** n-1. iterasyon  $C = \{1, k, \dots, s, j\}$ 

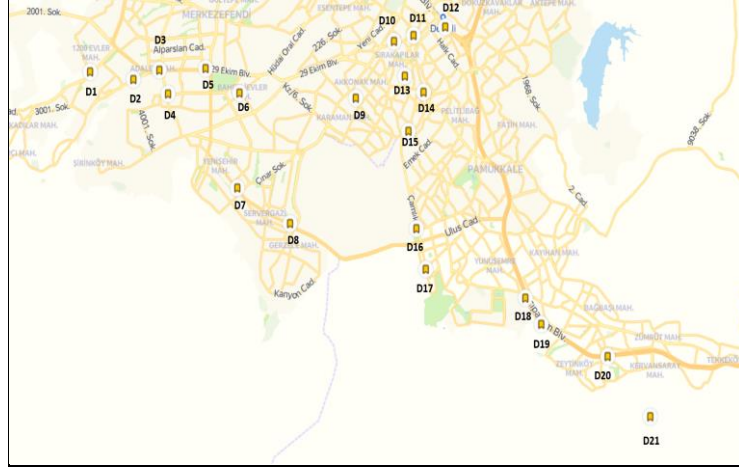
	1	2	3	...	k	...	s	...	j	...	n
1	-	$d_{12}$	$d_{13}$	...	$d_{1k}^*$	...	$d_{1s}$	...	-	...	$d_{1n}$
k	-	$d_{k2}$	$d_{k3}$	...	-	...	-	...	-	...	$d_{kn}$
...	...	...	...	...	...	...	...	...	...	...	...
s	-	$d_{s2}$	$d_{s3}$	...	-	...	-	...	$d_{sj}^*$	...	$d_{sn}$
j	-	$d_{j2}$	$d_{j3}$	...	-	...	-	...	-	...	$d_{jn}$

### 3. Uygulama

Bu çalışmada Denizli'de şehir içi ulaşımda diğer taşıma sistemleri ile entegre olacak şekilde yapılması planlanan hafif raylı sistem projesi için minimum mesafeli bir hat belirleme problemi, MYA olarak ele alınmıştır. Öncelikle hafif raylı sistem için bir graf oluşturulmuştur. Graf oluşturmada Denizli Belediyesi'nin daha önce "Hafif Raylı Sistem Projesi" için hazırladığı fakat uygulamaya geçirmedikleri çalışmalar ve şehir haritası dikkate alınmıştır. Belediyenin bu proje için belirlediği hatlardan biri, Servergazi Devlet Hastanesi-Pamukkale Üniversitesi Kalp Merkezi'dir. Yapılan literatür araştırması ve uzman görüşleri doğrultusunda Yandex haritalardan bu hattı içine alan bölgedeki mevcut otobüs durakları da dikkate alınarak belirlenen 21 merkezi yer, grafi oluşturan duraklar olarak varsayılmıştır. Yürüyüş mesafesinde olan birbirine çok yakın mesafeli yerler birleştirilerek tek bir durak olarak düşünülmüştür. Durak olarak varsayılan bu yerler; yolculuk talebinin yüksek olduğu, diğer toplu ulaşım sistemleri ile entegrasyonun sağlandığı, nüfusun yoğun olduğu eğitim, hastane gibi önemli özel ve kamu kurumları, alışveriş merkezleri, sanayi ve ticaret alanları olup literatürde hat belirlemede kullanılan kriterleri destekler niteliktedir. Belirlenen durak isimleri ve kısaltmaları, Tablo 5'te verilmiştir. Bu sisteme ilişkin graf, Şekil 1'de gösterilmiştir. Duraklar arası mesafeler, Yandex haritalar yardımı ile belirlenmiştir. Bu aşamada Yandex haritalardan tüm duraklar, haritada bulunarak işaretlenmiş ve her bir durağın diğer tüm duraklar ile arasındaki mesafeler, km birimi cinsinden hesaplanmıştır. Duraklar arası mesafeyi gösteren D uzaklık matrisi, Tablo 6'da verilmiştir. Problemin çözümünde Prim, Kruskal algoritmaları ve matris yöntemi kullanılmıştır.

**Tablo 5.** Hafif raylı sistem için duraklar

Kısaltma	Duraklar	Kısaltma	Duraklar	Kısaltma	Duraklar
D1	1200 Evler	D8	Başak Kavşağı	D15	Lise Caddesi
D2	Servergazi Devlet Hastanesi	D9	Akkonak Parkı	D16	Tiyatro Kavşağı
D3	Aydem	D10	Devlet Hastanesi	D17	Çamlık Caddesi
D4	Bahçelievler	D11	Bayramyeri	D18	PAÜ Merkez Kampüsü
D5	Adalet Sarayı	D12	Otogar	D19	Cerrahi Hastanesi
D6	Denizli Özel Sağlık Hastanesi	D13	Saltak	D20	Bağbaşı Belediye Kavşağı
D7	Teras Park	D14	Delikliçınar Meydanı	D21	Teleferik

**Şekil 1.** Raylı hat sistemine ilişkin graf**Tablo 6.** D uzaklık matrisi

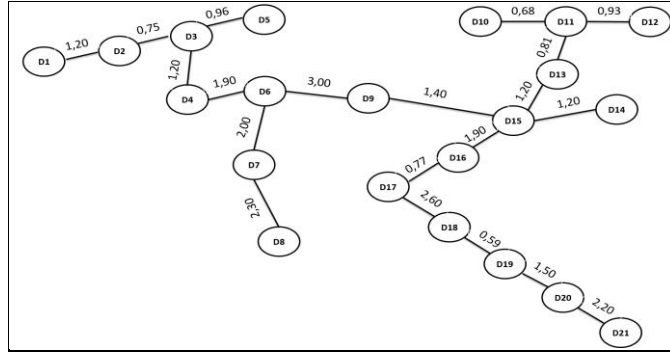
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21
D1	-	1,20	1,80	2,80	2,80	4,60	4,70	7,70	6,50	7,00	7,70	9,20	8,10	8,70	7,90	9,20	10,00	15,00	14,00	15,00	17,00
D2	1,20	-	0,75	1,30	1,50	2,90	3,50	6,10	5,10	5,80	6,50	8,00	6,50	7,50	6,50	8,00	8,50	13,00	12,00	14,00	16,00
D3	1,80	0,75	-	1,20	0,96	2,80	2,90	6,20	4,50	5,20	5,90	7,50	6,40	6,90	6,10	7,40	8,40	13,00	11,00	13,00	15,00
D4	2,80	1,30	1,20	-	1,60	1,90	2,60	4,80	4,70	5,20	5,80	7,40	6,00	7,20	6,00	7,10	7,60	12,00	10,00	12,00	14,00
D5	2,80	1,50	0,96	1,60	-	2,30	2,50	5,40	4,40	5,10	5,60	6,60	5,70	6,80	5,60	6,60	7,40	12,00	11,00	13,00	15,00
D6	4,60	2,90	2,80	1,90	2,30	-	2,00	3,30	3,00	3,70	4,40	6,00	4,30	5,50	4,30	6,00	6,80	12,00	9,90	11,00	14,00
D7	4,70	3,50	2,90	2,60	2,50	2,00	-	2,30	3,40	4,60	5,00	6,50	5,00	5,80	4,30	5,40	4,70	7,40	7,70	9,40	12,00
D8	7,70	6,10	6,20	4,80	5,40	3,30	2,30	-	3,50	4,60	5,30	6,90	5,10	5,80	4,60	3,90	4,40	8,90	8,20	9,10	11,00
D9	6,50	5,10	4,50	4,70	4,40	3,00	3,40	3,50	-	1,50	2,00	3,60	1,60	2,60	1,40	3,10	3,90	8,10	6,20	7,70	11,00
D10	7,00	5,80	5,20	5,20	5,10	3,70	4,60	4,60	1,50	-	0,68	2,20	0,95	2,10	2,10	4,00	4,90	9,10	7,40	8,80	11,00
D11	7,70	6,50	5,90	5,80	5,60	4,40	5,00	5,30	2,00	0,68	-	0,93	0,81	2,10	2,10	4,00	4,90	8,90	6,60	8,10	10,00
D12	9,20	8,00	7,50	7,40	6,60	6,00	6,50	6,90	3,60	2,20	0,93	-	2,40	2,10	3,10	4,80	5,70	7,80	6,00	7,40	9,70
D13	8,10	6,50	6,40	6,00	5,70	4,30	5,00	5,10	1,60	0,95	0,81	2,40	-	1,40	1,20	3,10	4,00	7,70	5,80	7,30	9,60
D14	8,70	7,50	6,90	7,20	6,80	5,50	5,80	5,80	2,60	2,10	2,10	2,10	1,40	-	1,20	3,60	4,50	7,90	5,50	7,00	9,10
D15	7,90	6,50	6,10	6,00	5,60	4,30	4,40	4,60	1,40	2,10	2,10	3,10	1,20	1,20	-	1,90	2,80	6,70	4,80	6,30	8,50
D16	9,20	8,00	7,40	7,10	6,60	6,00	5,40	3,90	3,10	4,00	4,00	4,80	3,10	3,60	1,90	-	0,77	3,30	3,90	5,20	7,60
D17	1,00	8,50	8,40	7,60	7,40	6,80	4,70	4,40	3,90	4,90	4,90	5,70	4,00	4,50	2,80	0,77	-	2,60	3,20	4,70	6,80
D18	1,50	13,00	13,00	12,00	12,00	12,00	7,40	8,90	8,10	9,10	8,90	7,80	7,70	7,90	6,70	3,30	2,60	-	0,59	2,10	4,10
D19	1,40	12,00	11,00	10,00	11,00	9,90	7,70	8,20	6,20	7,40	6,60	6,00	5,80	5,50	4,80	3,90	3,20	0,59	-	1,50	3,80
D20	1,50	14,00	13,00	12,00	13,00	11,00	9,40	9,10	7,70	8,80	8,10	7,40	7,30	7,00	6,30	5,20	4,70	2,10	1,50	-	2,20
D21	1,70	16,00	15,00	14,00	15,00	14,00	12,00	11,00	11,00	11,00	10,00	9,70	9,60	9,10	8,50	7,60	6,80	4,10	3,80	2,20	-

### 3.1. Prim algoritması ile çözüm

Prim algoritması için çözüme rastgele seçilen D1 düğümü ile başlanmıştır. D1 düğümüne en yakın mesafede 1,20 km ile D2 olduğu için D1 düğümü, D2'ye bağlanmıştır. Sonra D1 veya D2 düğümlerinden ulaşılacak en kısa mesafeye bakılmıştır. En kısa mesafe 0,75 km olup D2 düğümü D3 düğümüne bağlanmıştır. D1, D2 ve D3 düğümlerine bakıldığında en kısa mesafenin D3 ve D5 arasındaki 0,96 km olduğu görülmüş ve ağaca D5 eklenmiştir. Çözüme bu şekilde devam edildiğinde 20 iterasyon sonunda tüm düğümler birbirine bağlanmış olup Tablo 7'de görüldüğü gibidir. Bu işlemler yapılırken döngü oluşmamasına dikkat edilmiştir. 20 iterasyon sonunda Şekil 2'deki gibi bir MYA elde edilmiştir. Toplam minimum mesafe 29,09 km'dir.

**Tablo 7.** Prim algoritması hesaplamaları

İterasyon	Kenar	Mesafe	İterasyon	Kenar	Mesafe	İterasyon	Kenar	Mesafe
1	D1-D2	1,20	8	D6-D9	3,00	15	D15-D16	1,90
2	D2-D3	0,75	9	D9-D15	1,40	16	D16-D17	0,77
3	D3-D5	0,96	10	D15-D14	1,20	17	D17-D18	2,60
4	D3-D4	1,20	11	D15-D13	1,20	18	D18-D19	0,59
5	D4-D6	1,90	12	D13-D11	0,81	19	D19-D20	1,50
6	D6-D7	2,00	13	D11-D10	0,68	20	D20-D21	2,20
7	D7-D8	2,30	14	D11-D12	0,93			



**Şekil 2.** Prim algoritması ile elde edilen MYA

### 3.2. Kruskal algoritması ile çözüm

Aynı problem, Kruskal algoritması ile çözülmek istendiğinde öncelikle kenarlar arasındaki mesafeler küçükten büyüğe sıralanarak bir S kenarlar kümesi oluşturulmuştur. Sırasıyla 0,59 km (D18-D19 arası), 0,68 km (D10-D11 arası), 0,75 km (D2-D3 arası), 0,77 km (D16-D17 arası), 0,81 km (D11-D13 arası), 0,93 km (D11-D12 arası) seçilmiştir. 0,93 km'den sonra minimum mesafenin D10 ve D13 düğümleri arasındaki 0,95 km olan kenar olduğu görülmüştür. Fakat bu düğümler, bir döngü yaratacağı için seçilmemiştir. İşlemler, bu şekilde döngü oluşturmadan tüm kenarlar seçilene kadar tekrarlanmış ve Tablo 8'de gösterilmiştir. Kruskal algoritması ile 47. iterasyon sonunda ulaşılan MYA, Prim algoritması ile elde edilen MYA ile aynı olup toplam minimum mesafe 29,09 km'dir.



Tablo 8. Kruskal algoritması hesaplamaları

İterasyon	Kenar	Mesafe	Seçim	İterasyon	Kenar	Mesafe	Seçim
1	D18-D19	0,59	Seçildi	25	D9-D11	2	Seçilmedi
2	D10-D11	0,68	Seçildi	26	D10-D14	2,1	Seçilmedi
3	D2-D3	0,75	Seçildi	27	D11-D14	2,1	Seçilmedi
4	D16-D17	0,77	Seçildi	28	D12-D14	2,1	Seçilmedi
5	D11-D13	0,81	Seçildi	29	D10-D15	2,1	Seçilmedi
6	D11-D12	0,93	Seçildi	30	D11-D15	2,1	Seçilmedi
7	D10-D13	0,95	Seçilmedi	31	D18-D20	2,1	Seçilmedi
8	D3-D5	0,96	Seçildi	32	D10-D12	2,2	Seçilmedi
9	D3-D4	1,2	Seçildi	33	D20-D21	2,2	Seçildi
10	D1-D2	1,2	Seçildi	34	D5-D6	2,3	Seçilmedi
11	D13-D15	1,2	Seçildi	35	D7-D8	2,3	Seçildi
12	D14-D15	1,2	Seçildi	36	D12-D13	2,4	Seçilmedi
13	D2-D4	1,3	Seçilmedi	37	D5-D7	2,5	Seçilmedi
14	D9-D15	1,4	Seçilmedi	38	D9-D14	2,6	Seçilmedi
15	D13-D14	1,4	Seçildi	39	D4-D7	2,6	Seçilmedi
16	D2-D5	1,5	Seçilmedi	40	D17-D18	2,6	Seçildi
17	D9-D10	1,5	Seçilmedi	41	D1-D4	2,8	Seçilmedi
18	D19-D20	1,5	Seçildi	42	D1-D5	2,8	Seçilmedi
19	D4-D5	1,6	Seçilmedi	43	D3-D6	2,8	Seçilmedi
20	D9-D13	1,6	Seçilmedi	44	D15-D17	2,8	Seçilmedi
21	D1-D3	1,8	Seçilmedi	45	D2-D6	2,9	Seçilmedi
22	D4-D6	1,9	Seçildi	46	D3-D7	2,9	Seçilmedi
23	D15-D16	1,9	Seçildi	47	D6-D9	3	Seçildi
24	D6-D7	2	Seçildi				

### 3.3. Matris yöntemi ile çözüm

Matris yöntemi ile çözüm için rastgele seçilen D1 düğümü, ilk satıra eklenerek bir tablo oluşturulmuştur. 1. iterasyonda, D1'e bağlantısı olan diğer tüm düğümler arasından 1,20 km minimum mesafeli D2 seçilmiş ve tablonun ikinci satırına eklenmiştir. 2. iterasyonda D1 ve D2 düğümlerine minimum mesafeli bağlantısı olan 0,75 km ile D3 olup bu düğüm, 3. iterasyonda tablonun üçüncü satırına eklenmiştir. Bu şekilde işlemlere devam edilerek 20 iterasyon sonucunda Tablo 9 elde edilmiştir. Bu yöntem ile yine diğer algoritmalarda olduğu gibi kenarları {(1,2), (2,3), (3,4), (3,5), (4,6), (6,7), (6,9), (7,8), (9,15), (15,13), (15,14), (15,16), (13,11), (11,10), (11,12), (16,17), (17,18), (18,19), (19,20), (20,21)} ve minimum mesafesi 29,09 km olan bir MYA elde edilmiştir.

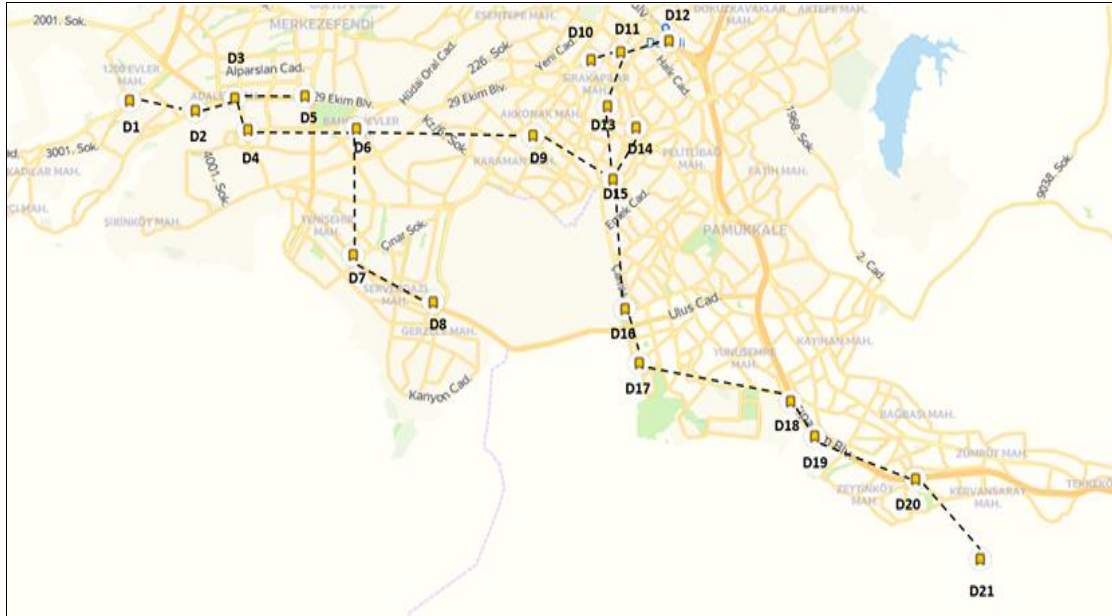
**Tablo 9.** Kruskal algoritması hesaplamaları

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21
D1	-	1,20*	1,80	2,80	2,80	4,60	4,70	7,70	6,50	7,00	7,70	9,20	8,10	8,70	7,90	9,20	10,00	15,00	14,00	15,00	17,00
D2	-	-	0,75*	1,30	1,50	2,90	3,50	6,10	5,10	5,80	6,50	8,00	6,50	7,50	6,50	8,00	8,50	13,00	12,00	14,00	16,00
D3	-	-	-	1,20*	0,96*	2,80	2,90	6,20	4,50	5,20	5,90	7,50	6,40	6,90	6,10	7,40	8,40	13,00	11,00	13,00	15,00
D5	-	-	-	-	-	2,30	2,50	5,40	4,40	5,10	5,60	6,60	5,70	6,80	5,60	6,60	7,40	12,00	11,00	13,00	15,00
D4	-	-	-	-	-	1,90*	2,60	4,80	4,70	5,20	5,80	7,40	6,00	7,20	6,00	7,10	7,60	12,00	10,00	12,00	14,00
D6	-	-	-	-	-	-	2,00*	3,30	3,00*	3,70	4,40	6,00	4,30	5,50	4,30	6,00	6,80	12,00	9,90	11,00	14,00
D7	-	-	-	-	-	-	-	2,30*	3,40	4,60	5,00	6,50	5,00	5,80	4,30	5,40	4,70	7,40	7,70	9,40	12,00
D8	-	-	-	-	-	-	-	-	3,50	4,60	5,30	6,90	5,10	5,80	4,60	3,90	4,40	8,90	8,20	9,10	11,00
D9	-	-	-	-	-	-	-	-	-	1,50	2,00	3,60	1,60	2,60	1,40*	3,10	3,90	8,10	6,20	7,70	11,00
D15	-	-	-	-	-	-	-	-	-	2,10	2,10	3,10	1,20*	1,20*	-	1,90*	2,80	6,70	4,80	6,30	8,50
D14	-	-	-	-	-	-	-	-	-	2,10	2,10	2,10	1,40	-	-	3,60	4,50	7,90	5,50	7,00	9,10
D13	-	-	-	-	-	-	-	-	-	0,95	0,81*	2,40	-	-	-	3,10	4,00	7,70	5,80	7,30	9,60
D11	-	-	-	-	-	-	-	-	-	0,68*	-	0,93*	-	-	-	4,00	4,90	8,90	6,60	8,10	10,00
D10	-	-	-	-	-	-	-	-	-	-	-	2,20	-	-	-	4,00	4,90	9,10	7,40	8,80	11,00
D12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4,80	5,70	7,80	6,00	7,40	9,70
D16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0,77*	3,30	3,90	5,20	7,60
D17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2,60*	3,20	4,70	6,80
D18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0,59*	2,10	4,10
D19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1,50*	3,80
D20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2,20*

### 3.4. Karşılaştırmalı Analiz

MYA problemini çözmek için kullanılan Prim ve Kruskal algoritmaları ile matris yöntemi karşılaştırıldığında toplam minimum MYA mesafesi 29,09 km'dir. Şekil 3'te, elde edilen MYA'nın harita görüntüsü verilmiştir. Her üç yöntemde aynı MYA ve optimal sonuç elde

edilmesine rağmen süreçler farklıdır. Yöntemlerin, süreç bazında karşılaştırması Tablo 10'daki gibidir. Tablo 10'da görüldüğü gibi matris yönteminin avantajı, Prim ve Kruskal algoritmalarının aksine her iterasyonda grafla çalışmaya gerek olmamasıdır. Bu yöntemde öncelikle uzaklık matrisi belirlendikten sonra problem, mesafeler kullanılarak çözülür. Bu yöntem için gereken iterasyon sayısı, Prim algoritması ile aynı iken Kruskal algoritmasından daha azdır, fakat işlem kolaylığı düşünüldüğünde Prim algoritmasından daha pratiktir. Prim algoritması, her iterasyonda bir ağaç oluşturarak çalışmaktadır. Ancak Kruskal algoritmasının bazı iterasyonları ağaç olmayan, orman olarak tanımlanan yapıyı oluşturabilir. Ayrıca Prim algoritması ve matris yöntemi ile Kruskal algoritmasının aksine graftaki herhangi bir düğüm seçilerek çözüme başlanabilir. Bu çalışmada 21 düğüm göz önüne alındığında, birçok kısıt eklenmesi gerektiği için oluşturulan model çok karmaşık olacaktır. Prim algoritması ile çözümde daha fazla düğüm olduğu varsayılırsa, çözümün ilerleyen aşamalarında birçok düğüme bakılarak bunlar arasındaki minimum mesafenin bulunması gerektiği için bazı düğümlerin gözden kaçırılacağı düşünülmektedir. Kruskal algoritmasında ise S kenarlar kümesi oluşturulduğu için düğüm sayısı artsa bile minimum mesafenin gözden kaçırılması ihtimali azalmaktadır.



Şekil 3. MYA harita görüntüsü

Tablo 10. Yöntemlerin süreç bazında karşılaştırması

	Prim Algoritması	Kruskal Algoritması	Matris Yöntemi
Her iterasyonda grafla çalışmaya gerek var mı?	Evet	Evet	Hayır
Graftaki herhangi bir düğüm seçilerek çözüme başlanabilir mi?	Evet	Hayır	Evet
İterasyon sayısı	20	47	20
İşlem kolaylığı	Zor	Zor	Kolay

#### 4. Sonuç

Şehir içi ulaşımda, ulaşım sistemlerinin sürücü ve araç odaklı olmaktan ziyade trafikte geçirilen sürenin azaltılmaya çalışılarak şehirde yaşayanların hızlı, konforlu ve güvenli bir şekilde ulaşımına öncelik verilerek planlanması, şehir içi ulaşımda memnuniyetin sağlanması açısından önem arz etmektedir. Son yıllarda Denizli ilinde artan nüfus nedeniyle oluşan trafik yoğunluğunda iyileşme sağlamak için ulaşım yatırımları, yeni ulaşım planlama kararları

çerçevesinde mevcut ulaşım sistemleri ile birlikte çalışan toplu ulaşım odaklı hafif raylı sistemler olarak düşünülmektedir. Bu sebeple bu çalışmada yapılması planlanan hafif raylı sistem için minimum mesafeli bir hat belirleme problemi, MYA olarak ele alınmıştır. Literatüre bakıldığında MYA'yı çözmek için bir dizi algoritma geliştirildiği görülmektedir. Bu çalışmada çözüm analizinde literatürde en yaygın kullanılan Prim ve Kruskal algoritmaları ile Prim algoritması fikrinden esinlenerek yakın zamanda önerilen matris yöntemi dikkate alınmıştır. Söz konusu problem, mevcut ağda çözümlenerek aynı MYA elde edilmiş ve toplam minimum MYA mesafesi 29,09 km olarak hesaplanmıştır. Ancak her üç yöntemde aynı MYA ve optimal sonuç elde edilmesine rağmen süreçler farklıdır. Sonuç olarak matris yönteminin çözümü, Prim ve Kruskal algoritmalarına göre daha kolaydır ve gerçek hayatta bu konu ile ilgili yapılacak çalışmalarda rahatlıkla kullanılabilir. Bundan sonraki MYA ile ilgili çalışmalar için problemde çok fazla kenar sayısı varsa işlem kolaylığı, iterasyon sayısı, karmaşıklıkların minimuma indirilmesi açısından matris yönteminin kullanılması önerilebilir.

Çalışmada belirlenen hafif raylı sistem hattı, belediyenin başlatmış olduğu hafif raylı sistem projesi ile uyumlu olup sadece bir öneri niteliğindedir. Durak sayısı ve durak yerleri değiştirilerek duyarlılık analizleri yapılabilir. Böylece, minimum mesafeli hattaki değişimler izlenerek bunlar arasından optimal olan hat belirlenebilir. Yapılan çalışmada temel amaç, MYA problemi ile minimum mesafeli bir hat belirlemek olup çalışmanın bazı sınırlılıkları vardır. Durak yerlerini belirlemek, hattın etkinliği bakımından önemli olup çok ciddi bir süreç ve konu ile ilgili teknik bilgi gerektiren ayrı bir çalışma konusudur. Bu çalışmada, yapılan literatür araştırması ve uzman görüşleri doğrultusunda grafi oluşturmak için belediyenin planladığı hattı içine alan bölgedeki mevcut otobüs durakları da dikkate alınarak belirlenen 21 merkezi yer, grafi oluşturan duraklar olarak varsayılmıştır. Bu yerler; yolculuk talebinin yüksek olduğu, diğer toplu ulaşım sistemleri ile entegrasyonun sağlandığı, nüfusun yoğun olduğu eğitim, hastane gibi önemli özel ve kamu kurumları, alışveriş merkezleri, sanayi ve ticaret alanlarıdır. Farklı kriterler dikkate alınarak farklı yöntemler ile uzman bilgisi dahilinde durak yerleri yeniden belirlenebilir. Bunun için matematiksel modeller kurulabilir. Ayrıca hat belirleme, sadece hat mesafesine bağlı olmayıp maliyet, süre gibi farklı değişkenler de dikkate alınabilir. İleriki çalışmalarda, konu ile ilgili raylı sistem tasarım uzmanları, belediye yöneticileri ve akademisyenler ile disiplinlerarası daha detaylı bir çalışma/proje yapılabilir.

Denizli iline ait trafik sorunları düşünüldüğünde bölgeye raylı sistem hizmetinin sunulmasının şehir içi trafikte bir iyileşme sağlayacağı muhakkaktır. Fakat bu çok önemli bir iş gücü, çok büyük bir bütçe ve uzun bir süreç gerektirmektedir. Bir raylı sistem hattının hayata geçirilmesi, uzunluğuna ve durak sayısına göre bazen yıllar alabilmektedir. Raylı sistemlerin yatırım maliyetleri; genel olarak hatta, hattaki tünel, viyadük ve yüzey kesimlerinin uzunluklarına, zemin cinsine ve yapılış şekline bağlı olarak değişmektedir. Bunun için öncelikle bölge ile ilgili jeolojik ve jeoteknik etüd yapılmalı, raylı sistemin geçeceği yolun eğimi, virajı, elektrifikasyon sistemi gibi kriterler dikkate alınmalıdır. Kısaca, böyle bir yatırım kararı alınırken ekonomik, sosyal ve çevresel kriterler birlikte değerlendirilmelidir.

### Kaynakça

- [1] A.Y. Gündüz, M. Kaya, C. Aydemir, "Kentiçi ulaşımında karayolu ulaşımına alternatif sistem: Raylı ulaşım sistemi", *Akademik Yaklaşımlar Dergisi*, cilt 2, sayı 1, ss. 134-151, 2011.
- [2] H. G. Önder, F. Ademar, "Türkiye'deki kentiçi raylı toplu taşıma sistemlerinin ulaşım ana planları bağlamında değerlendirilmesi", *Demiryolu Mühendisliği*, sayı 10, ss. 31-45, 2019.
- [3] H. Kutlu, H. Ulvi, F. Akdemir, "Gelişmekte Olan Ülkelerde Raylı Sistem Yatırım Kararlarını Etkileyen Ölçütlerin Belirlenmesi: AB ve Türkiye Özelinde Bir Araştırma", *Demiryolu Mühendisliği*, sayı 9, ss. 61-78, 2019.
- [4] S. T. Altuntaş, Y. Eyigün, "Sürdürülebilir kentiçi ulaşım politikaları raylı sistemler örneği", *Teknoloji ve Uygulamalı Bilimler Dergisi*, cilt 3, sayı 2, ss. 217-233, 2021.

- [5] S. Şahin, “Malatya’da kentiçi ulaşım ve toplu taşıma sistemlerinin karşılaştırmalı incelenmesi: Trambüs ve hafif raylı sistem örneği”, Yüksek Lisans Tezi, İnönü Üniversitesi, Sosyal Bilimler Enstitüsü, Malatya, 2020.
- [6] H. Gerçek, B. Karpak, T. Kılınçaslan, “A multiple criteria approach for the evaluation of the rail transit networks in Istanbul”, *Transportation*, vol. 31, no. 2, pp. 203-228, 2004.
- [7] A. Ludin, S. Latip, “Using multi-criteria analysis to identify suitable light rail transit route”, *Jurnal Alam Bina*, vol. 9, no. 1, pp. 131-142, 2007.
- [8] M. I. T. Alkubaisi, “Predefined evaluating criteria to select the best tramway route”, *Journal of Traffic and Logistics Engineering*, vol. 2, no. 3, pp. 211-217, 2014.
- [9] G. S. Kalamaras, L. Brino, G. Carrieri, C. Pline, P. Grasso, “Application of multicriteria analysis to select the best highway alignment”, *Tunnelling and Underground Space Technology*, vol. 15, no. 4, pp. 415-420, 2000.
- [10] M. Hamurcu, T. Eren, “Ankara Büyükşehir Belediyesi’nde çok ölçütlü karar verme yöntemi ile monoray güzergâh seçimi”, *Transist*, sayı 8, ss. 410-419, 2015.
- [11] M. Hamurcu, T. Eren, “Using ANP-TOPSIS methods for route selection of monorail in Ankara”, *In 28th European Conference on Operational Research*, Poznan, Poland, 2016, pp. 3-6.
- [12] M. Hamurcu, T. Eren, “Raylı sistem projeleri kararında AHS-HP ve AAS-HP kombinasyonu”, *Gazi Mühendislik Bilimleri Dergisi*, cilt 3, sayı 3, ss.1-13, 2017.
- [13] B. Sarimehmet, M. Hamurcu, E. Tamer, “Çok kriterli karar verme: Kırıkkale YHT istasyonu-şehir bağlantısının sağlanması”, *Demiryolu Mühendisliği*, sayı 11, ss. 26-40, 2020.
- [14] R. Banai, “Public transportation decision-making: A case analysis of the Memphis light rail corridor and route selection with analytic hierarchy process”, *Journal of Public Transportation*, vol. 9, no. 2, pp. 1-24, 2006.
- [15] I. M. Brunner, K. Kim, E. Yamashita, “Analytic hierarchy process and geographic information systems to identify optimal transit alignments”, *Transportation Research Record*, vol. 2215, no. 1, pp. 59-66, 2011.
- [16] K. Watanabe, K. Gotoh, K. Tachiiri, “Route selection for a new transportation system in hillside urban areas: A case study in Nagasaki, Japan”, *Journal of Urban Planning and Development*, vol. 132, no. 2, pp. 89-96, 2006.
- [17] R. L. Graham, P. Hell, “On the history of the minimum spanning tree problem”, *Annals of the History of Computing*, vol. 7, no. 1, pp. 43-57, 1985.
- [18] P.C. Pop, “The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances”, *European Journal of Operational Research*, vol. 283, no. 1, pp.1-15, 2020.
- [19] A. Dey, A. Pal, H. V. Long, “Fuzzy minimum spanning tree with interval type 2 fuzzy arc length: formulation and a new genetic algorithm”, *Soft Computing*, vol. 24, no. 6, pp. 3963-3974, 2020.
- [20] O. P. Biswas, M. Goel, H. Negi, M. Datta, “An efficient greedy minimum spanning tree algorithm based on vertex associative cycle detection method”, *Procedia Computer Science*, vol. 92, pp. 513-519, 2016.
- [21] E. K. Donkoh, S. K. Amponsah, K. F. Darkwah, “Optimal pipeline connection for the West African gas pipeline project”, *Research Journal of Applied Sciences, Engineering and Technology*, vol. 3, no. 2, pp. 67-73, 2011.
- [22] O. T. Arogundade, B. Sobowale, A. T. Akinwale, “Prim algorithm approach to improving local access network in rural areas”, *International Journal of Computer Theory and Engineering*, vol. 3, no.3, pp. 413-417, 2011.
- [23] Çevik, S. S. Karaca, M. Özkan, “En küçük yayılma modeli ile İç Anadolu Bölgesinde bir kargo firmasının dağıtım güzergâhının belirlenmesi”, *Karamanoğlu Mehmetbey Üniversitesi Sosyal ve Ekonomik Araştırmalar Dergisi*, cilt 2011, sayı 2, ss. 1-9, 2011.
- [24] C. K. Gitonga, “Prims algorithm and its application in the design of university LAN networks”, *International Journal*, vol. 3, no. 10, pp. 131-136, 2015.
- [25] N. P. Akpan, I. A. Iwok, “A minimum spanning tree approach of solving a transportation problem”, *International Journal of Mathematics and Statistics Invention*, vol. 5, no. 3, pp. 09-18, 2017.
- [26] F. Marpaung, “Comparative of Prim’s and Boruvka’s algorithm to solve minimum spanning tree problems”, *Journal of Physics: Conference Series*, vol. 1462, IOP Publishing, 2020. doi:10.1088/1742-6596/1462/1/012043
- [27] D. Rachmawati, F. Y. P. Pakpahan, “Comparative analysis of the Kruskal and Boruvka algorithms in solving minimum spanning tree on complete graph”, in *International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 2020, IEEE, pp. 55-62.

- [28] S. Jahan, T. Hossain, F. A. Simi, “Matrix method for determining minimum spanning tree”, *Journal of Applied Mathematics and Computation*, vol. 4, no. 4, pp. 118-122, 2020.
- [29] A. Shahin, F. Jaferi, “The shortest route for transportation in supply chain by minimum spanning tree”, *International Journal of Logistics Systems and Management*, 22(1), 43-54, 2015.
- [30] X. Zhou, J. Yang, J. J. Wu, Z. G. Shi, K. L., Gao, H. W., Liu (2018), “Interest tourism route plan algorithm based on improved minimum spanning tree”, *DEStech Transactions on Computer Science and Engineering, International Conference on Modeling, Simulation and Optimization (MSO 2018)*, ISBN: 978-1-60595-542-1, 270-273.
- [31] K. A. Tin, “Applications of the shortest spanning tree and path on graph theory”, *IRE Journals*, vol 1, no. 2, pp. 3-6, 2019.
- [32] J. A. Bondy, U. S. R. Murty, *Graph theory with applications*. London: Macmillan Press, 1976.
- [33] J. B. Kruksal, “On the shortest spanning subtree of a graph and the traveling salesman problem”, *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48-50, 1956.
- [34] R. C. Prim, “Shortest connection networks and some generalizations”, *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389-1401, 1957.
- [35] E. W. Dijkstra, “A note on two problems in connexion with graphs”, *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.
- [36] G. D. Kurt, “Kablosuz ağlarda en az kesinti ihtimali işbirlikli yol atama”, Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, 2011.

### Özgeçmiş



#### Müge AKAY

1994 tarihinde doğmuştur. Lisans Eğitimini Ege Üniversitesi'nde tamamlamıştır. Pamukkale Üniversitesi'nde sayısal yöntemler alanında yüksek lisans yapmaktadır. İç denetim uzmanı olarak çalışmaktadır.

E-Posta: makay202@posta.pau.edu.tr



#### Ayşegül TUŞ

1980 tarihinde doğmuştur. Lisans eğitimini Hacettepe Üniversitesi'nde tamamlamıştır. Pamukkale Üniversitesi'nde çalışmaktadır. İlgi alanına giren araştırma konuları; yöneylem araştırması, karar verme ve bulanık mantıktır.

E-Posta: atus@pau.edu.tr

### Beyanlar:

Bu makalede bilimsel araştırma ve yayın etiğine uyulmuştur.

Yazarların katkıları: Müge AKAY: Fikir oluşturma, literatür taraması, metodoloji, görselleştirme, yazma. Ayşegül TUŞ: Taslak hazırlama, kaynak inceleme, doğrulama, kontrol, elde edilen sonuçları değerlendirme.