



Available online at www.iujtl.com

JTL

Journal of Transportation and Logistics
7 (2) 2022



DOI: 10.26650/JTL.2022.1092188

RESEARCH ARTICLE

Bir Lojistik Firmasının En Kısa Yol Problemine Düğüm Kombinasyonu Algoritmasının Uygulanması

Applying the Node Combination Algorithm to the Shortest Path Problem for a Logistics Firm

Kevser Arman¹ , Aşegül Tuş²

Öz

Lojistik; ürünlerin taşınması, depolanması ve nihai varış noktasına ulaşması ile ilgili tüm süreçlerin yönetimidir. Lojistik faaliyetlerinin son derece karmaşık süreci, ürünlerin başlangıç noktasından varış noktasına kadar doğru bir şekilde koordinasyonunu gerektirir. Bu çalışmada, bir lojistik firması için Düzce-Artvin arasındaki toplam mesafeyi ve süreyi en aza indiren bir rota belirleme problemi, şebeke analiz yöntemlerinden biri olan En Kısa Yol (EKY) problemi olarak ele alınmıştır. Çalışmanın amacı, lojistik firmasının dağıtım faaliyetlerini optimize ederek daha yüksek düzeyde kârlılık ve müşteri hizmeti sunmaktır. Problemin çözümünde düğüm kombinasyonu algoritması kullanılmış, mesafe ve süre dikkate alınarak iki farklı rota elde edilmiştir. Bulgular, toplam minimum mesafenin 1152 km ve toplam minimum sürenin 16 saat 33 dakika olduğunu göstermektedir. Çalışmada toplam minimum mesafe dikkate alınarak elde edilen rota, Google Haritalar'ın sunduğu iki rotadan daha kısa mesafede ve sürede alternatif bir rota sunmaktadır. Ayrıca toplam minimum süre dikkate alınarak elde edilen rotanın, Google Haritalar'ın sunduğu alternatifler arasından en kısa mesafe ve süreye ait olan rota ile uyumlu olması, düğüm kombinasyonu algoritmasının uygulanabilirliğini göstermesi açısından önemlidir.

Anahtar Kelimeler: Şebeke Analizi, En Kısa Yol Problemi, Düğüm Kombinasyonu Algoritması

ABSTRACT

Logistics entails the management of all processes related to products' transportation, storage, and arrival at their destination. The highly complex process of logistics activities requires products to be accurately coordinated from their starting point to their destination. This study considers a route determination problem that minimizes the total distance and time between the cities of Düzce and Artvin in Turkey for a logistics company in terms of the shortest path problem (SPP), a network analysis method. The aim of the study is to provide a higher level of profitability and customer service by optimizing the distribution activities of a logistics company. The node combination algorithm has been used to solve the problem, with two different routes being obtained by considering distance and time. The study's findings show the total minimum distance to be 1,152 km and the total minimum time to be 16 hours and 33 minutes. The route the study obtained by considering the total minimum distance offers an alternative route in terms of both shorter distance and time compared to the two routes offered by Google Maps. In addition, having the route obtained by considering the total minimum time be compatible with the route with the shortest distance and time that is found among the alternatives offered by Google Maps is important in terms of demonstrating the applicability of the node combination algorithm used in the study.

Keywords: Network Analysis, Shortest Path Problem, Node Combination Algorithm

Başvuru/Submitted: 23.03.2021 • **Revizyon Talebi/Revision Requested:** 04.08.2022 • **Son Revizyon/Last Revision Received:** 09.08.2022 • **Kabul/Accepted:** 17.08.2022

1 Kevser Arman (Research Assistant), Pamukkale University, Department of Business Administration, Denizli, Türkiye. E-mail: karman@pau.edu.tr
ORCID: 0000-0002-4400-5976

2 **Sorumlu yazar/Corresponding author:** Aşegül Tuş (Assoc. Prof.), Associate Professor, Pamukkale University, Department of Business Administration, Denizli, Türkiye. E-mail: atus@pau.edu.tr ORCID: 0000-0003-1583-0616

Atf/Citation: Arman, K., & Tus, A. (2022). Bir lojistik firmasının en kısa yol problemine düğüm kombinasyonu algoritmasının uygulanması. *Journal of Transportation and Logistics*, 7(2), 289-302. <https://doi.org/10.26650/JTL.2022.1092188>



Extended Abstract

Logistics is the action of physically delivering businesses' goods or services to the required place at the desired time and has great importance concerning businesses' profitability. Transportation costs these days are constantly increasing, and timely delivery is one of the main issues customers focus on. For this reason, the importance of logistics activities increases daily (Kosif & Ekmekçi, 2012, p. 41). However, logistics companies spend more than half their budget on transportation activities, and obtaining the most efficient distribution path is necessary for acquiring higher profits (Zulfiqar et al., 2018, p. 371). Accordingly, one of the most important problems in logistics activities involves determining the shortest route in terms of distance that will save time and costs, and a network analysis method can be used to solve this problem. A network is an effective format used for representing complex systems. Network analysis methods are activities that ensure optimal coordination and placement of resources. Rapid progress has occurred in both the methodology and application of network analysis methods, one of these being the shortest path problem (SPP). This problem has been used to minimize a linear function representing the distance between a predetermined pair of nodes in a given network (Climaco & Martins, 1982, p. 399). Many algorithms have been developed to solve the SPP, with the Dijkstra, Prim, Bellman-Ford, Floyd-Warshall, and Johnson algorithms seen to have been widely used in the literature (Dermawan, 2019, p. 55).

This study considers a route determination problem that minimizes the total distance and time between the cities of Düzce and Artvin in Turkey for a logistics company with regard to the SPP. The aim of the study is to provide a higher level of profitability and customer service by optimizing the distribution activities of a logistics company. For this purpose, the study uses the node combination method, one of the SPP methods. Although Dijkstra's algorithm has been used frequently, it may not be easily understood, especially when applying the labeling method. The process of solving the SPP with the node combination algorithm is relatively simple, straightforward, and efficient compared to the Dijkstra algorithm. For this reason, this study has chosen to use the node combination algorithm. The basic idea behind the node combination (NC) algorithm proposed by Lu and Camitz (2011) is based on combining nodes instead of keeping the labeling sets, as occurs in the Dijkstra algorithm and differs from the Dijkstra algorithm in this respect. The node combination finds the nearest neighbor to the starting node and combines this node with the starting node, calculating the SPP iteratively. It then updates the edge weights connected to the nearest node. The path expression used in the SPP can occur different variables such as distance, time, or cost for different problems (Gencer & Karamanoğlu, 2020, p. 50). The current study examines the path characteristics in terms of distance and time, with the results then being compared to Google Maps. The study's findings show the total minimum distance to be 1,152 km and the total minimum time to be 16 hours and 33 minutes. The route the study obtained by considering the total minimum distance offers an alternative route with regard to shorter distance and time compared to the two routes offered by Google Maps. Moreover, the route obtained by considering the total minimum time is compatible with the route possessing the shortest distance and time that occur among the alternatives offered by Google Maps. This study has shown the node

combination algorithm to have been successfully implemented for a real-life problem. In addition, the results the study obtained are expected to contribute to decision makers in terms of offering alternatives that are different from the routes provided by Google Maps. Different results may occur between those from the applied algorithm and those from various online map services such as Google Maps because the node combination algorithm implements the direct distance between two cities, whereas Google Maps might use the alternative route between two cities.

One of the limitations of this study is that the SPP was solved only by considering distance and time. The concept of optimal route depends on the needs or the nature of the problem. The optimal route may be the shortest distance, the lowest cost, or the shortest time (Bulut & Erol, 2018, p. 189). Moreover, various factors such as road conditions, traffic variations, and vehicle characteristics may change the shortest time route or the lowest cost route, despite not changing the shortest distance. For this reason, future studies can address the optimal route determination problem with these variables using various other SPP algorithms and programming languages. In cases where routes change due to some sudden reasons or due to a change in traffic density while traveling between nodes, the real-time dynamic route control system proposed by Bulut and Erol (2018) may be used. Optimization algorithms may differ in performance with regard to particular problems. Hybrid approaches that take advantage of this difference may provide higher performance in many cases. With the increases in processors' computational capacity, hybrid methods have started being preferred more often (Uslu et al., 2020, p. 102). In addition, many problems in real world applications cannot be solved accurately in polynomial time by any known algorithm. As the number of nodes in the route increases, so does the complexity of the computational solution and the computation time (Bulut & Erol, 2018, pp. 189–190). In this case, metaheuristic algorithms may be used as stated by Uslu et al. (2020).

1. Giriş

Lojistik; işletmelerin mal veya hizmetlerini, ihtiyaç duyulan yere istenilen zamanda fiziksel olarak yerine getirme faaliyeti olup işletmelerin kârlılığı üzerinde oldukça büyük bir öneme sahiptir. Günümüzde taşımacılık maliyetleri sürekli artmaktadır ve zamanında teslimat, müşterilerin odaklandığı temel konuların başında gelmektedir. Bu durum, lojistik faaliyetlerinin rolünü her geçen gün daha da arttırmaktadır (Kosif ve Ekmekçi, 2012: 41). Bununla birlikte lojistik firmaları, bütçelerinin yarısından fazlasını taşıma faaliyetlerine harcamaktadır ve daha iyi kâr elde etmek için dağıtımın en verimli yolunu elde etmek gereklidir (Zulfiqar vd., 2018: 371). Buna göre lojistik faaliyetlerinde en önemli problemlerden biri, zaman ve maliyet tasarrufu sağlayacak en kısa mesafeli yolun belirlenmesidir. Bu problemin çözümü için bir şebeke analiz yöntemi kullanılabilir. Şebeke, karmaşık yapıli sistemlerin temsili için kullanılan etkili bir gösterim biçimidir. Şebeke analiz yöntemleri, mevcut kaynakların optimal şekilde koordine edilmesini ve yerleştirilmesini sağlayan faaliyetler bütünüdür. Şebeke analiz yöntemlerinin hem metodolojisinde hem de uygulamasında hızlı bir ilerleme söz konusu olup bu yöntemlerden biri En Kısa Yol (EKY) problemidir. Bu problem, belirli bir şebekede önceden belirlenmiş düğüm çifti arasındaki mesafeyi temsil eden doğrusal bir fonksiyonu minimize etmek için kullanılır (Climaco ve Martins, 1982: 399). EKY problemini çözmek için birçok algoritma geliştirilmiştir. Literatürde yaygın olarak Dijkstra, Prim, Bellman-Ford, Floyd-Warshall ve Johnson algoritmaları kullanılmıştır (Dermawan, 2019: 55).

EKY algoritmaları ile ilgili yapılan çalışmalar şu şekildedir. Hall (1986), Dijkstra algoritması gibi klasik EKY algoritmalarının, hem rastgele hem de zamana bağlı seyahat sürelerine sahip ağlarda minimum seyahat süresi yolunu bulmada başarısız olabileceğine işaret etmiş, optimal rota seçimine uyarlanabilir bir karar kuralı bulmak için dinamik programlamaya dayalı bir algoritma önermiştir. Klein (1991), bulanık EKY'lerin ağdaki gerçek bir yola karşılık gelmemesi olasılığı nedeniyle engellendiğini belirtmiş ve bu sorunu aşmak için çalışmada bulanık EKY'lere dayalı yeni modeller sunmuş, ayrıca yeni modelleri çözmek için dinamik programlamaya dayalı genel bir algoritma ortaya koymuştur. Opananon ve Miller-Hooks (2005), çok kriterli, stokastik ve zamanla değişen ağlarda ayarlanabilir tercih yolu stratejilerinin belirlenmesi için bir algoritma önermiştir. Söz konusu algoritma, seyahat süreleri gözlemlendiğinde tercih edilen kritere göre en iyi yolu seçmeye izin vermiştir. Karşlı (2010), stokastik bir ulaşım ağında EKY problemini Genetik Algoritma (GA) ve Yapay Bağışıklık Sistemi (YBS) yardımıyla çözmüştür. Lu ve Camitz (2011), Dijkstra algoritmasına dayalı düğüm kombinasyonu algoritmasını önermiştir. Xi (2012), Dijkstra algoritmasını önemli ölçüde geliştirerek EKY problemlerinin çözümünde etkili bir şekilde kullanmıştır. Deng vd. (2012), EKY problemini belirsiz bir ortamda ele almak için genelleştirilmiş bir Dijkstra algoritması önermiştir. Amaliah vd. (2016), Java Adası'nda şehirler arasındaki EKY'leri bulmak için düğüm kombinasyonu algoritmasını uygulamıştır. Çalışmadan elde edilen sonuçlar, referans olarak alınan Google haritası ile düğüm kombinasyonunun doğruluğunun % 92,88 olduğunu göstermiştir. Mirino (2017), Dijkstra ve Floyd-Warshall algoritmalarını birlikte kullanmıştır. En yakın hastaneye en hızlı seyahat süresini belirlemek için Dijkstra algoritması, hastaneye en yakın mesafeyi belirlemek için ise Floyd-Warshall algoritması

uygulanmıştır. Fitro (2018), Coğrafi Bilgi Sistemlerine dayalı EKY probleminin çözümü için düğüm kombinasyonu ve Dijkstra algoritmalarını birlikte kullanmıştır. Dermawan (2019), bir tren yolculuğunda EKY problemi için Dijkstra ve Floyd-Warshall algoritmalarını karşılaştırmıştır. Çalışmadan elde edilen sonuçlar, Dijkstra algoritmasının dört parametre (zaman karmaşıklığı, bellek karmaşıklığı, tamamlama düzeyi ve optimizasyon düzeyi) üzerinde Floyd-Warshall algoritmasından daha iyi performansa sahip olduğunu ortaya koymuştur. Ekmen (2020), EKY problemlerinde Dijkstra, Bellman-Ford, Johnson ve Floyd-Warshall algoritmalarını analiz edip karşılaştırmıştır. Çalışmadan elde edilen sonuçlar, EKY kapsamında kullanılacak algoritmaların çeşitli faktörlere göre seçilmesi gerektiğini vurgulamaktadır.

Ulaştırma problemleri EKY algoritmaları kullanılarak verimli bir şekilde çözülmektedir (Gencer ve Karamanoğlu 2020: 3). Literatür incelendiğinde, bu konu kapsamında daha önce yapılan çalışmalar şu şekildedir. Yuan ve Wang (2009), acil durum lojistik yönetiminde yol seçimi problemi için değiştirilmiş bir Dijkstra algoritması tasarlamıştır. Tirastittam ve Waiyawuththanapoom (2014), toplu taşıma planlama sistemini tasarlamak için Dijkstra algoritmasını kullanmıştır. Ojekudo ve Akpan (2017), bir boya firmasının üretim tesisinden yedi farklı bayiye EKY problemi için Dijkstra algoritmasını kullanmıştır. Bulunan EKY ile firmanın boya taşıma maliyeti azaltılmıştır. Zulfikar vd. (2018), optimal dağıtım rotası planlama problemi için Dijkstra ve sweep algoritmalarını kullanmıştır. Rosita vd. (2019), dağıtım için en uygun rotayı elde etmek amacıyla vektör normalizasyon tekniği ile Dijkstra algoritmasını birleştirmiştir. Özdemir vd. (2021), ipek yolu koridorları arasında EKY problemi için Dijkstra algoritmasını kullanmıştır.

Bu çalışmanın amacı, bir lojistik firmasının Düzce-Artvin arası gerçekleştirdiği dağıtım faaliyetlerini optimize ederek daha yüksek seviyede kârlılık ve müşteri hizmeti sağlamaktır. Yapılan çalışmalar incelendiğinde EKY problemlerinin çözümünde farklı algoritmaların gerçek hayat problemlerinde uygulanabilirliği yönünde bir boşluk olduğu görülmüştür. Dijkstra algoritması, sıkça kullanılmasına rağmen özellikle etiketleme yöntemi uygulanırken kolayca anlaşılabilir. Düğüm kombinasyonu algoritması ile EKY bulma süreci, Dijkstra algoritmasına göre nispeten basit, anlaşılır ve verimlidir. Bu nedenle bu çalışmada bir lojistik firmasının EKY probleminin çözümünde düğüm kombinasyonu algoritması kullanılmıştır.

Düğüm kombinasyonu algoritması ile ilgili ulusal literatürde ilk kez çalışma yapıyor olması, algoritmanın ulusal literatüre kazandırılması açısından önemlidir. Çalışma, şu şekilde düzenlenmiştir. İkinci bölümde EKY problemi hakkında kısa bir bilgi verilerek Dijkstra ve düğüm kombinasyonu algoritmaları anlatılmıştır. Üçüncü bölümde, Düzce-Artvin arası EKY problemi, düğüm kombinasyonu algoritması ile çözülmüştür. Son bölümde ise sonuçlara ilişkin değerlendirmeler yapılmıştır.

2. En Kısa Yol (EKY) Problemi

EKY probleminde amaç, bir şebekede yer alan başlangıç ile hedef düğümler arasındaki minimum toplam mesafeye sahip yolu veren rotayı bulmaktır (Hillier and Lieberman, 2001: 411). Şebekeler ise bir amacı gerçekleştirmek için gerekli faaliyetler ve olaylardan

meydana gelen, faaliyet ve olayların birbirleri ile bağlantı ve ilişkilerini gösteren şema olarak tanımlanabilir (Öztürk, 2009: 525). Bir şebeke, düğüm ve bağlantı olarak iki tür sembol ile tanımlanır. Tüm faaliyetler oklar ile ve tüm olaylar da daire ile gösterilip; şebekede yer alan daireler düğüm olarak, düğümler ile ilişkili olan oklar ise dal ya da bağlantı olarak ifade edilir (Winston ve Goldberg, 2004: 413). Şebekelerin kullanımı, yönetsel problemin görselleştirilmesine ve anlaşılmasına yardımcı olur (Render vd., 2015: 342). Şebekeler, günlük hayatımızda ulaşım, elektrik, iletişim gibi çok sayıda ortamda ve çeşitli şekillerde ortaya çıkar (Ojekudo ve Akpan, 2017: 20). İki şehir arasındaki en kısa yolun belirlenmesi, bir doğalgaz boru hattı ağının tasarımı, bir projenin faaliyetleri için zaman çizelgesinin belirlenmesi ve bir boru hattı ağı aracılığı ile petrol sahalarından rafinerilere minimum maliyetli akış programının belirlenmesi gibi birçok yöneylem araştırması durumu, şebekeler ile modellenilebilir ve şebeke analiz yöntemleri ile çözülebilir (Taha, 2017: 247). Bu çalışmada, bir lojistik firması için Düzce'den Artvin'e minimum mesafeyi ve süreyi veren rota, EKY problemi ile belirlenmeye çalışılmıştır. EKY problemlerinde kullanılan yol ifadesi farklı problemler için mesafe, süre ve maliyet vb. şekillerde ortaya çıkabilir (Gencer ve Karamanoğlu 2020: 50). Bu çalışmada düğümler arası nitelikler, mesafe ve süre olarak iki açıdan incelenmiş ve sonuçlar karşılaştırılmıştır. Çalışmada ele alınan problemin çözümünde Dijkstra algoritmasına dayalı düğüm kombinasyonu algoritması kullanıldığı için bu bölümde öncelikle Dijkstra algoritması, ardından düğüm kombinasyonu algoritması hakkında bilgi verilmiştir.

2.1. Dijkstra Algoritması

EKY problemlerinde en sık kullanılan algoritmalarından biri olan Dijkstra algoritması (1959), bir başlangıçtan (kaynak) bir hedefe (varış) EKY'yi bulmak için kullanılır (Ojekudo ve Akpan, 2017: 23). Dijkstra algoritması, yalnızca ağırlıkları negatif bir değer olmayan bir yol için problemi çözer (Dermawan, 2019: 55). Bu algoritma, aynı zamanda bir açgözlü (greedy) algoritma biçimidir. Açgözlü algoritmada, her adım için tüm seçenekler arasından optimal olan seçilir ve değerlendirmeye alınır (Erol ve Bulut, 2017: 2). Dijkstra algoritması, her adımda optimal çözümü bulmayı amaçlamaktadır. Dijkstra algoritmasında geçici ve kalıcı olmak üzere iki tür düğüm etiketi vardır. Bir düğüme giden daha kısa bir yol bulunabiliyorsa geçici etiket değiştirilir. Tüm durumlar incelendiğinde ve daha kısa bir yol bulunamadığında geçici etiketin durumu, kalıcı olarak işaretlenir. Tablo 1'de pseudo kodu verilen Dijkstra algoritmasının adımları şu şekilde ifade edilebilir (Taha, 2017: 255-256):

Adım 1: Her düğüm için geçici bir mesafe değeri atanır. Başlangıç düğümü (Düğüm 1), kalıcı etiket [0, -] ile diğer tüm düğümler ise sonsuz ile işaretlenir.

Adım 2: Düğüm 2 için tüm komşu düğümler göz önünde bulundurulur ve mevcut düğüm için geçici yollar hesaplanır. Hesaplanan geçici yollar arasından en kısa olan mesafe seçilir ve ilgili düğüm kalıcı olarak işaretlenir. Söz konusu durum, diğer tüm düğümler için tekrarlanır.

Adım 3: Tüm düğümler kalıcı etiketlere sahipse işlem sonlandırılır, aksi takdirde adım 2 tekrarlanır.

Tablo 1: Dijkstra algoritmasının pseudo kodu

1: $d[s] := 0, v_u := v_s, V := V - \{s\}$	<i>/*Initialization*/</i>
2: while $d[u] < \infty$ and $ V > 0$	
3: $V := V - \{u\}$	<i>/*mark u as visited*/</i>
4: for each j in V	
5: $d[j] := \min \{d[j], d[u] + W[u,j]\}$	<i>/*relaxation*/</i>
6: $d[u] :=$ the smallest value in d for nodes in V	
<i>/*at the end of the algorithm, vector d contains the corresponding distances*/</i>	
Kaynak: Lu ve Camitz (2011)	

2.2. Düğüm Kombinasyonu Algoritması (Node Combination Algorithm)

Lu ve Camitz (2011) tarafından önerilen düğüm kombinasyonu (NC) algoritmasının temel fikri, etiketleme kümelerini Dijkstra algoritmasında olduğu gibi tutmak yerine düğümleri birleştirmeye dayanır. Bu yönü ile Dijkstra algoritmasından farklılaşır. Düğüm kombinasyonu, başlangıç düğümünün en yakın komşusunu bulur ve bu düğümü başlangıç ile birleştirerek EKY'leri yinelemeli olarak hesaplar. Daha sonra en yakın düğüme bağlı kenar ağırlıklarını günceller. Düğüm kombinasyonu algoritması, negatif olmayan kenar ağırlıkları olan yönlendirilmemiş ağlar için sunulmuştur. V düğümlü, E bağlantılarından oluşan negatif olmayan ağırlıklı bir ağ $G = (V, E, W)$ verildiğinde, ağırlık matrisi; $W_{N \times N}$, v_s ; başlangıç düğüm, $d(v_j)$; v_s ile v_j arasındaki mesafeyi kaydedecek olan vektör olarak tanımlandığında NC algoritmasının adımları, aşağıdaki gibidir (Lu ve Camitz, 2011: 6403):

Adım 1: Başlangıç düğümü belirlenir. Burada $d(v_s) = 0$ olarak atanır.

Adım 2: Başlangıç düğümü ile bağlantılı en küçük ağırlıklara bakılarak en yakın düğüm bulunur. $w_{sk} = \min \{w_{sj}\}$ yapan v_s 'nin komşularından v_k seçilir. $d(v_k) = w_{sk}$ olup v_s için bitişik düğüm yoksa durulur.

Adım 3: Bulunan düğüm birleştirilir ve v_k silinir ($V = V - v_k$). Eğer $V = \emptyset$ ise durulur.

Adım 4: Kenar ağırlıkları güncellenir. Her kenar e_{kj} için $w_{sj} = \min \{w_{sj}, w_{sk} + w_{kj}\}$ güncellenir ve adım 1'e geri dönlür.

Tablo 2'de düğüm kombinasyonu algoritmasının pseudo kodu verilmiştir.

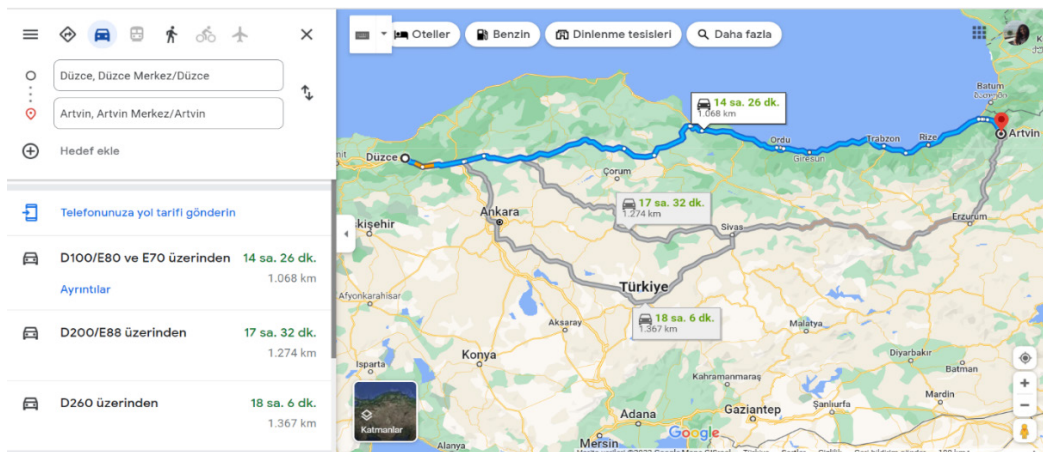
Tablo 2: Düğüm kombinasyonu algoritmasının pseudo kodu

1: $W[s,u] := 0, v_u := v_s, V := V - \{s\}$	<i>/*Initialization*/</i>
2: while $W[s,u] < \infty$ and $ V > 0$	
3: $V := V - \{u\}$	<i>/*node combination*/</i>
4: for each j in V	
5: $W[s,j] := \min \{W[s,j], W[s,u] + W[u,j]\}$	<i>/*updating edge weights*/</i>
6: $v_u :=$ the nearest neighbour of s in V	<i>/*finding the nearest neighbour*/</i>
<i>/*at the end of the algorithm, the sth row in W contains the corresponding distances*/</i>	
Kaynak: Lu ve Camitz (2011)	

Düğüm kombinasyonu algoritması ile Dijkstra algoritması arasındaki temel fark, mesafeleri belirlenmiş ziyaret edilen (çözülmüş) düğümler kümesidir. Düğüm kombinasyonu algoritmasında düğümler yeni kaynak düğümde birleştirilir, bu kümenin sürdürülmesine gerek olmadığı anlamına gelir. İkinci olarak, gevşeme doğrudan kenar ağırlığı üzerinde yapılır, bu da geçici mesafeleri kaydetmek için ek bellek gerekmediği anlamına gelir. Üçüncüsü, düğüm kombinasyonu algoritması, başlangıç düğümün en yakın komşusunu tekrar tekrar bularak gerçekleştirilir, bu da EKY'leri bulma sürecini daha anlaşılır hale getirir. Deneysel değerlendirmeler, düğüm kombinasyonu algoritmasının daha az bellek kullanarak Dijkstra algoritması ile aynı sürede EKY'leri bulduğunu ortaya koymaktadır. Düğüm kombinasyonu algoritmasının ağırlık matrisleri üzerinde uygulanması da daha uygundur çünkü bulunacak minimum eleman ve değiştirilecek elemanlar, aynı satırdadır. Düğümlerin birleşimi, kenar ağırlıkları sonsuza ayarlanarak gerçekleşirse düğüm kombinasyonu algoritması, MATLAB gibi matematik platformlarında ve sonsuzluk kavramının uygulandığı diğer programlama dillerinde uygun olan vektörler tarafından gerçekleştirilebilir. Sonuç olarak, Dijkstra algoritmasını anlamamanın alternatif bir yolu olarak düğüm kombinasyonu algoritması, EKY'leri bulma sürecini çok daha basit, anlaşılır kılar ve bellek tasarrufu sağlar (Lu ve Camitz, 2011: 6408).

3. Uygulama

Bu çalışmada, lojistik alanında faaliyet gösteren bir firmanın Düzce-Artvin arası dağıtım faaliyetlerinde toplam mesafe ve süreyi minimum kılan rota belirleme problemi, düğüm kombinasyonu algoritması ile gerçekleştirilmiştir. Oluşturulacak şebeke için Google Haritalar'ın Düzce'den Artvin'e sunmuş olduğu üç alternatif rota dikkate alınmıştır. Bu alternatif rotalar, Şekil 1'de görülmektedir.



Şekil 1. Düzce'den Artvin'e alternatif rotalar

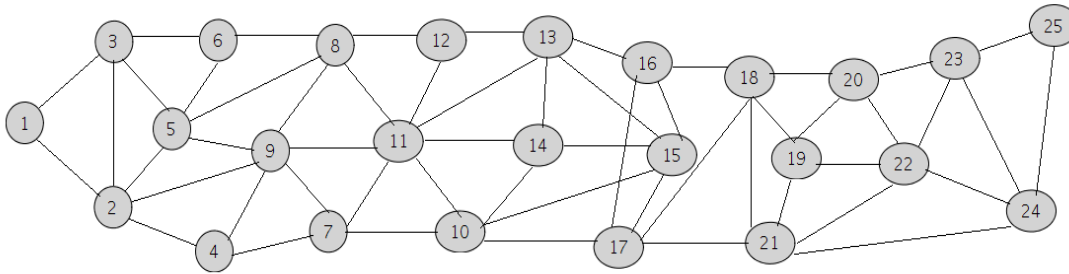
Kaynak: <https://www.google.com.tr/maps> (09.03.2022)

Düzce-Artvin arası EKY problemi için belirlenen şehirler, Tablo 3'te yer almaktadır. Bu şehirler, Şekil 1'deki alternatif rotalar üzerinde bulunan şehirlerdir. Ancak, en uzun mesafe ve en fazla süreye sahip üçüncü rotada yer alan Kırşehir, Nevşehir ve Kayseri şehirleri,

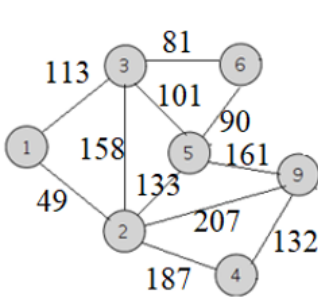
minimum mesafe ve süreyi bulmada etkili olmayacağı için şebekeye dahil edilmemiştir. Toplam 25 şehir (düğüm) ve 54 bağlantıdan oluşan şebeke, Şekil 2'deki gibidir.

Tablo 3: Düzce – Artvin arası EKY problemi için belirlenen şehirler

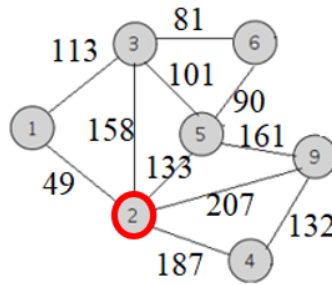
Düğüm	Şehir	Düğüm	Şehir	Düğüm	Şehir
1	Düzce	10	Yozgat	19	Gümüşhane
2	Bolu	11	Çorum	20	Trabzon
3	Zonguldak	12	Sinop	21	Erzincan
4	Ankara	13	Samsun	22	Bayburt
5	Karabük	14	Amasya	23	Rize
6	Bartın	15	Tokat	24	Erzurum
7	Kırıkkale	16	Ordu	25	Artvin
8	Kastamonu	17	Sivas		
9	Çankırı	18	Giresun		



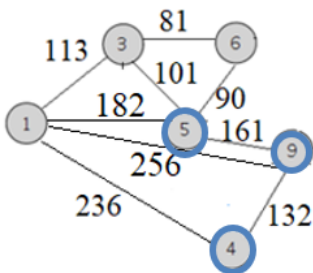
Şekil 2. 25 Şehir Arasındaki Bağlantıları Gösteren Şebeke



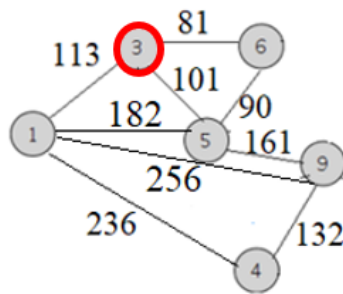
Şekil 3. Başlangıç Düzümü: Düzüm 1



Şekil 4. Düzüm 1'e En Yakın Düzüm: Düzüm 2

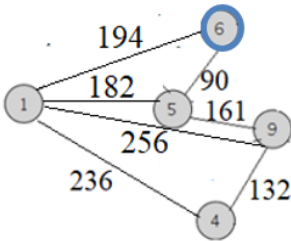


Şekil 5. Düzüm 2'nin Silinmesi

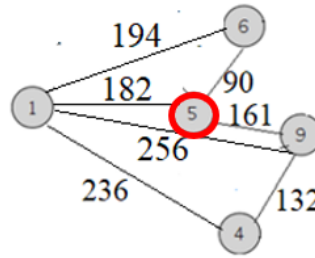
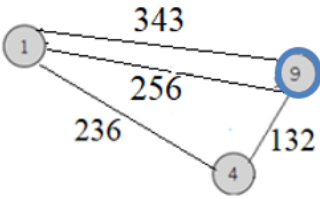


Şekil 6. Düzüm 1'e En Yakın Düzüm: Düzüm 3

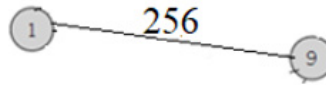
Şekil 2'deki şebekede başlangıç şehri, Düzce (Düğüm 1) ve hedef şehir, Artvin (Düğüm 25)'dir. Google Haritalar'ın "Yön" özelliği kullanılarak 25 şehir için doğrudan bağlantıya sahip olanlar belirlenmiş olup, mesafe ve süreler elde edilmiştir. Şehirler arası farklı mesafelerin ve sürelerin olduğu durumlarda iki şehir arasındaki en kısa mesafe ve süre dikkate alınmıştır. Düğüm kombinasyonu algoritmasının şebeke üzerinden işleyişi, Düzce (Düğüm 1) - Çankırı (Düğüm 9) arası toplam mesafeyi minimum kılan EKY problemi örnek alınarak Şekil 3-12'de gösterilmiştir. Başlangıç düğümüne en yakın düğüm, kırmızı renk ile; en yakın düğümün silinmesi sonucu, en yakın düğümün komşu kenarlarının başlangıç düğümünden uzaklıklarının güncellendiği düğüm(ler) ise mavi renk ile belirtilmiştir.



Şekil 7. Düğüm 3'ün Silinmesi

Şekil 8. Düğüm 1'e En Yakın Düğüm:
Düğüm 5

Şekil 9. Düğüm 5'in Silinmesi



Şekil 10. Düzce'den Çankırı'ya EKY

Çalışmada ilk olarak mesafeler dikkate alınarak çözüm gerçekleştirilmiştir. Buna göre Düzce'den Artvin'e minimum mesafeyi veren rotayı belirleme problemi, düğüm kombinasyonu algoritması kullanılarak MATLAB programında çözülmüş ve Şekil 11'deki sonuç ekranı elde edilmiştir.

```
Command Window
e =
    1152

L =
fx 25 23 20 18 16 13 12 8 5 2 1
```

Şekil 11. MATLAB programında düğüm kombinasyonu algoritması sonuç ekranı (mesafe)

Şekil 11'deki sonuç ekranına göre Düzce'den Artvin'e minimum mesafe, 1152 km olup rota şu şekildedir: Düğüm 1 (Düzce) → Düğüm 2 (Bolu) → Düğüm 5 (Karabük) →

Düğüm 8 (Kastamonu) → Düğüm 12 (Sinop) → Düğüm 13 (Samsun) → Düğüm 16 (Ordu) → Düğüm 18 (Giresun) → Düğüm 20 (Trabzon) → Düğüm 23 (Rize) → Düğüm 25 (Artvin).

Toplam minimum mesafe yerine toplam minimum sürenin ele alınmasının rota üzerindeki etkisini değerlendirmek için Düzce'den Artvin'e minimum süreyi veren rota belirleme problemi çözülerek Şekil 12'deki sonuç ekranı elde edilmiştir.

```
Command Window
e =
    994
L =
fx
25  23  20  18  16  13  11  9  2  1
```

Şekil 12. MATLAB programında düğüm kombinasyonu algoritması için sonuç ekranı (süre)

Buna göre Düzce'den Artvin'e minimum süre, 16 saat 33 dakika olup rota şu şekildedir: Düğüm 1 (Düzce) → Düğüm 2 (Bolu) → Düğüm 9 (Çankırı) → Düğüm 11 (Çorum) → Düğüm 13 (Samsun) → Düğüm 16 (Ordu) → Düğüm 18 (Giresun) → Düğüm 20 (Trabzon) → Düğüm 23 (Rize) → Düğüm 25 (Artvin). Çalışmada elde edilen rotalar, Şekil 13'te harita üzerinde gösterilmiştir.

Tablo 4'te ise Google Haritalar'ın sunduğu ve düğüm kombinasyonu algoritmasından elde edilen tüm sonuçlar, detaylı bir şekilde yer almaktadır.



Şekil 13. Elde edilen rotalar

4. Sonuç ve öneriler

Lojistik sektörü, dünyada her geçen gün büyümeye devam eden sektörlerin başında olup, lojistik faaliyetleri işletmelerin faaliyet alanına bakılmaksızın genel kârlılığı üzerinde yüksek bir etkiye sahiptir. Lojistik faaliyetleri oldukça karmaşık bir süreç içermekte olup, bu durum mal veya hizmetlerin doğru bir koordinasyon ile başlangıç noktasından varış noktasına teminini gerektirir. Lojistik firmaları ise bu işlemleri başarılı bir şekilde gerçekleştirmek

Tablo 4: Düzce'den Artvin'e alternatif rotaların karşılaştırılması

Alternatif Rota I (Google Haritalar)
Rota: Düzce → Bolu → Çankırı → Çorum → Amasya → Samsun → Ordu → Giresun → Trabzon → Rize → Artvin
Süre: 14 saat 26 dakika
Mesafe: 1068 km
Alternatif Rota II (Google Haritalar)
Rota: Düzce → Bolu → Çankırı → Çorum → Yozgat → Sivas → Erzincan → Erzurum → Artvin
Süre: 17 saat 32 dakika
Mesafe: 1274 km
Alternatif Rota III (Google Haritalar)
Rota: Düzce → Bolu → Ankara → Kırşehir → Nevşehir → Kayseri → Sivas → Erzincan → Erzurum → Artvin
Süre: 18 saat 7 dakika
Mesafe: 1367 km
Düğüm Kombinasyonu Algoritması I (mesafe)
Rota: Düzce → Bolu → Karabük → Kastamonu → Sinop → Samsun → Ordu → Giresun → Trabzon → Rize → Artvin
Süre: 16 saat 52 dakika
Mesafe: 1152 km
Düğüm Kombinasyonu Algoritması II (süre)
Rota: Düzce → Bolu → Çankırı → Çorum → Samsun → Ordu → Giresun → Trabzon → Rize → Artvin
Süre: 16 saat 33 dakika
Mesafe: 1154 km

için bu sektörde rol göstermektedir. Lojistik yönetimindeki temel problemlerden biri en kısa mesafe veya en kısa süreyi veren yol seçimidir. Lojistik alanında EKY problemi oldukça karşılaşılan bir durumdur ve bu problemi çözmek için birçok algoritma geliştirilmektedir. Bu çalışmanın temel amacı, bir lojistik firmasının Düzce - Artvin arası dağıtım faaliyetlerinde toplam mesafeyi ve süreyi minimum kılan rota belirleme problemini düğüm kombinasyonu algoritması ile gerçekleştirmektir. EKY problemi kapsamında en sık kullanılan algoritmalarından biri Dijkstra algoritmasıdır ancak Dijkstra algoritmasının kullanımı çok basit değildir ve fazla işlem gerektirmektedir. Dijkstra algoritmasına dayalı düğüm kombinasyonu algoritması ise daha az işlem ile sonuçların kolay ve başarılı şekilde elde edilmesini sağlamaktadır. Çeşitli şebekeler üzerinde yapılan değerlendirmeler, düğüm kombinasyonu algoritmasının Dijkstra algoritması kadar verimli olduğunu ortaya koymuştur. Bu nedenle çalışmada ele alınan problemin çözümünde düğüm kombinasyonu algoritması kullanılmıştır. Çalışmada ele alınan problem için toplam minimum mesafe ve toplam minimum süre açısından iki çözüm gerçekleştirilmiştir. Çalışmadan elde edilen bulgular, Google Haritalar'ın sunduğu alternatif rotalar ile karşılaştırılmıştır. Düğüm kombinasyonu algoritması ile elde edilen sonuçlar, Google Haritalar'ın sunduğu alternatif rota I ile uyumlu olup, alternatif rota II ve III'ten daha kısa mesafe ve sürede sonuçlar ortaya koymaktadır. Çalışma sonucunda toplam minimum mesafe ve süre sırasıyla 1152 km ve 16 saat 33 dakika olarak bulunmuştur.

Ulusal literatürde düğüm kombinasyonu algoritmasının kullanıldığı bir çalışma yapılmamıştır. Bu çalışmada düğüm kombinasyonu algoritmasının gerçek hayat problemi

için başarılı bir şekilde gerçekleştirildiği gösterilmektedir. Buna ek olarak bu çalışmada elde edilen rotaların, Düzce-Artvin arası EKY problemi için Google Haritalar'dan farklı alternatif rotalar sunarak karar vericilere katkı sağlaması beklenir. Uygulanan algoritma sonucunda elde edilen sonuç ile Google haritalar gibi çeşitli çevrimiçi harita hizmetleri arasında farklı sonuçlar olması mümkündür. Çünkü düğüm kombinasyonu algoritması, iki şehir arasındaki doğrudan mesafeyi uygularken Google haritalar, iki şehir arasındaki alternatif yolu kullanabilir. Bu çalışmanın sınırlılıklarından biri EKY'nin sadece mesafeler ve süreler dikkate alınarak bulunmasıdır. Örneğin; yol durumu, trafiğin seyri, araç özellikleri vb. durumlar, en kısa mesafeyi değiştirmese bile en kısa süreli yol ya da en az maliyetli yol rotasını değiştirecektir. Bu nedenle, gelecekte yapılacak çalışmalarda farklı EKY algoritmaları ve programlama dilleri kullanılarak bahsedilen değişkenler ile optimal rota belirleme problemi ele alınabilir.

Hakem Değerlendirmesi: Dış bağımsız.

Çıkar Çatışması: Yazar çıkar çatışması beyan etmemişlerdir.

Finansal Destek: Yazar finansal destek beyan etmemişlerdir.

Yazar Katkıları: Çalışma Konsepti/Tasarım- K.A., A.T.; Veri Toplama- K.A., A.T.; Veri Analizi/Yorumlama- K.A., A.T.; Yazı Taslağı- K.A., A.T.; İçeriğin Eleştirel İncelemesi- K.A., A.T.; Son Onay ve Sorumluluk- K.A., A.T.

Peer Review: Externally peer-reviewed.

Conflict of Interest: Author declared no conflict of interest.

Financial Disclosure: Author declared no financial support.

Author Contributions: Conception/Design of Study- K.A., A.T.; Data Acquisition- K.A., A.T.; Data Analysis/ Interpretation- K.A., A.T.; Drafting Manuscript- K.A., A.T.; Critical Revision of Manuscript- K.A., A.T.; Final Approval and Accountability- K.A., A.T.

Kaynakça

- Amaliah, B., Faticah, C., & Riptianingdyah, O. (2016). "Finding the Shortest Paths Among Cities in Java Island Using Node Combination Based on Dijkstra Algorithm". *International Journal on Smart Sensing & Intelligent Systems*, 9(4), 2219-2236.
- Bulut, F., & Erol, H. M. (2018). "A Real-Time Dynamic Route Control Approach on Google Maps Using Integer Programming Methods". *International Journal of Next-Generation Computing*, 189-202.
- Climaco, J. C. N., & Martins, E. Q. V. (1982). "A Bicriterion Shortest Path Algorithm". *European Journal of Operational Research*, 11(4), 399-404.
- Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). "Fuzzy Dijkstra Algorithm for Shortest Path Problem Under Uncertain Environment". *Applied Soft Computing*, 12(3), 1231-1237.
- Dermawan, T. S. (2019). "Comparison of Dijkstra dan Floyd-Warshall Algorithm to Determine the Best Route of Train". *IJID (International Journal on Informatics for Development)*, 7(2), 54-58.
- Dijkstra EW. (1959) "A Note on Two Problems in Connexion with Graphs". *Numerische Mathematik*, 1(1), 269-271.
- Düzce'den Artvin'e Alternatif Rotalar, <https://www.google.com/maps> (09.03.2022).
- Ekmen, D. E. (2020). "A Study on Performance Evaluation of Optimization Algorithms in the Shortest Path Problem", (Unpublished Master Thesis), Ankara Yıldırım Beyazıt University Graduate School of Natural and Applied Sciences, Ankara.
- Erol, M. H., & Bulut, F. (2017, April). "Real-Time Application of Travelling Salesman Problem Using Google Maps API". In *2017 Electric Electronics, Computer Science, Biomedical Engineering's Meeting (EBBT)* (pp. 1-5). IEEE.

- F.S. Hillier, & G.L. Lieberman (2001). *Introduction to Operations Research*, (Seventh Edition), McGraw-Hill.
- Fitro, A., P Sulistio Ilham, A., B Saeful, O., & Frendianata, I. (2018). "Shortest Path Finding in Geographical Information Systems Using Node Combination and Dijkstra Algorithm". *International Journal of Mechanical Engineering and Technology*, 9(2), 755-760.
- Gencer C., & Karamanaoğlu Y. E. (2020). *Şebeke Optimizasyonu*. Nobel Yayınevi, Ankara.
- Hall, R. W. (1986). "The Fastest Path Through a Network with Random Time-Dependent Travel Times". *Transportation Science*, 20(3), 182-188.
- Karlı, N. (2010). "Akıllı Ulaşım Sistemleri için Yapay Bağışıklık Sistemleri ve Genetik Algoritma ile Yeni Stokastik En Kısa Yol Algoritmalarının Geliştirilmesi", (Yayınlanmamış Doktora Tezi), Atatürk Üniversitesi Fen Bilimleri Enstitüsü, Erzurum.
- Klein, C. M. (1991). "Fuzzy Shortest Paths". *Fuzzy Sets and Systems*, 39(1), 27-41.
- Kosif, B., & Ekmekçi, İ. (2012). "Araç Rotalama Sistemleri ve Tasarruf Algoritması Uygulanması". *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 11(21), 41-51.
- Lu, X., & Camitz, M. (2011). "Finding the Shortest Paths by Node Combination". *Applied Mathematics and Computation*, 217(13), 6401-6408.
- Mirino, A. E. (2017). "Best Routes Selection Using Dijkstra and Floyd-Warshall Algorithm". In *2017 11th International Conference on Information & Communication Technology and System (ICTS)* (pp. 155-158). IEEE.
- Ojekudo, N. A., & Akpan, N. P. (2017). "An Application of Dijkstra's Algorithm to Shortest Route Problem". *IOSR Journal of Mathematics (IOSR-JM)*, 13(3), 20-32.
- Opananon, S., & Miller-Hooks, E. (2005). "Adjustable Preference Path Strategies for Use in Multicriteria, Stochastic, and Time-Varying Transportation Networks". *Transportation Research Record*, 1923(1), 137-143.
- Özdemir, S., Sacar, Ö., & Özcan, E. "Dijkstra Algoritması Kullanılarak İpek Yolu Koridorları Arasında En Kısa Ulaştırma Güzergâhının Belirlenmesi". *Demiryolu Mühendisliği*, (13), 97-105.
- Öztürk, A. (2009). *Yöneylem Araştırması*, (12. Baskı). Ekin Basım Yayın Dağıtım, Bursa.
- Render, B., Stair Jr, R. M., Hanna, M. E. & Trevor, S. T. (2015). *Quantitative Analysis for Management*, 13e. Pearson Education.
- Rosita, Y. D., Rosyida, E. E., & Rudiyanto, M. A. (2019). "Implementation of Dijkstra Algorithm and Multi-Criteria Decision-Making for Optimal Route Distribution". *Procedia Computer Science*, 161, 378-385.
- Shu-Xi, W. (2012). "The Improved Dijkstra's Shortest Path Algorithm and Its Application". *Procedia Engineering*, 29, 1186-1190.
- Taha, H. A. (2017). *Operations Research: An Introduction*. Pearson Education Limited.
- Tirastittam, P., & Waiyawuththanapoom, P. (2014). "Public Transport Planning System by Dijkstra Algorithm: Case Study Bangkok Metropolitan Area. World Academy of Science, Engineering and Technology International Journal of Social, Behavioral, Educational, Economic", *Business and Industrial Engineering*, 8(1), 54-59.
- Uslu, M. F., Uslu, S., & Bulut, F. (2020). "An Adaptive Hybrid Approach: Combining Genetic Algorithm and Ant Colony Optimization for Integrated Process Planning and Scheduling". *Applied Computing and Informatics*, 18 (1/2), 101-112.
- Winston, W. L., & Goldberg, J. B. (2004). *Operations Research: Applications and Algorithms* (Fourth Edition). Belmont: Thomson Brooks/Cole.
- Yuan, Y., & Wang, D. (2009). "Path Selection Model and Algorithm for Emergency Logistics Management". *Computers & Industrial Engineering*, 56(3), 1081-1094.
- Zulfıqar, O.M., Isnanto, R.R. & Nurhayati, O.D. (2018). "Optimal Distribution Route Planning Based on Collaboration of Dijkstra and Sweep Algorithm", *10th International Conference on Information Technology and Electrical Engineering (ICITEE) Information Technology and Electrical Engineering*, Bali, Indonesia.