

3B CBS'DE AĞ ANALİZLERİNE YÖNELİK COĞRAFI VERİTABANININ OTOMATİK OLARAK ÜRETİLMESİ

İ. R. Karas

¹Gebze Yüksek Teknoloji Enstitüsü, Jeodezi ve Fotogrametri Mühendisliği Bölümü, 41400, Gebze, Kocaeli, ragib@gyte.edu.tr

ÖZET

Ağ analizleri işlemlerinde optimum yolların belirlenmesi için kullanılan bir dizi algoritma mevcuttur. Bu algoritmalar genel olarak "En kısa yol" algoritmaları olarak adlandırılırlar. Bunlar arasında en yaygın kullanılanı ve en iyi çözümü sunanı Dijkstra algoritmasıdır. Bu bildiride Dijkstra Algoritmasının çalışma prensibi ve matematiksel altyapısına değinilecek, üç boyutlu bir ağda otomatik olarak en kısa yolların hesaplanması ve coğrafi veritabanında bir araya getirilmesine yönelik olarak geliştirilen uygulama detaylandırılacaktır.

Anahtar Sözcükler: Ağ Analizi, Coğrafi Veritabanları, Yazılım Geliştirme, Uygulama Geliştirme.

ABSTRACT

AUTOMATICALLY GENERATION OF GEO-DATABASE FOR 3D NETWORK ANALYSIS

There are several algorithms for determining of shortest paths in network analysis process. They are generally called as Shortest Path Algorithms. Dijkstra algorithm is common used alorithm and the best solution camparing in others. In this paper, mathematical infrastructure and fundamentals of Dijkstra algorithm described, and automatically generation of geo-database for 3D network analysis application is presented.

Keywords: Network Analysis, geo-database, software development, aplication development.

1. GİRİŞ

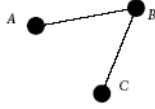
Günümüzde, iki boyutlu (2B) ağ analizi uygulamalarında büyük mesafe kaydedilmiş, ağ analizini temel alan bir çok teknoloji pratikte kullanılır hale gelmiştir. Kara yolları, tren yolları, nehirler, boru hatları, telefon ve elektrik hatları gibi birbirlerine çizgi özelliklerle bağlı sistemler ya da yapılar **ağ** olarak adlandırılır. Ağ yapıları üzerinden bir noktadan diğer bir noktaya erişebilme özelliği vardır (Esri, 1996). İnsanların bir yerden başka bir yere ulaşmaları, servis hizmetlerinin ve malların taşınması ve dağıtılması, kaynak ve enerjinin ulaştırılması ve bilgi iletişimi gibi faaliyetler, tanımlanabilen ağ yapıları içinde gerçekleşir (Güngör, 1999). **Ağ analizleri**, şebeke yapısına sahip, birbiriyle birleşen coğrafi varlıkların bağlantı şekillerinden, karar vermeye yönelik sonuç çıkarmaya yarayan konum analizleridir.

2B yada 3B, ağ analizleri için kullanılan konumsal objeler, bir şebeke yapısında düzgün hatlarla birbirine bağlanmış olarak vektörel yapıda bir araya getirilmelidirler. Ancak hat-düğüm (arc-node) topolojisi oluşturulmuş veriler ağ analizlerinin gerçekleştirilmesine olanak sağlarlar. Hat-düğüm yapısı topolojik bir yapıdır ve graf teorisini esas alır. Bu nedenle, öncelikle Graf Teorisi ve optimum yol analizlerinin yapılmasına imkan sağlayan Dijkstra algoritması hakkında bilgi verilecek, ardından üç boyutlu ağ analizi uygulaması için coğrafi veri tabanının otomatik olarak üretilmesi uygulaması tanıtılacaktır.

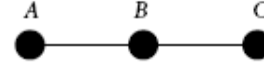
1.1 Graf Teorisi

Aralarında söyleyiş bakımından benzerlik olsa da grafların her hangi bir fonksiyonun grafiği ile ilgisi yoktur. Basitçe bir graf, düğüm olarak adlandırılan noktalar ve bu noktaları birleştiren hatlardan oluşan ve geometrik bir bilgi vermeyip, sadece düğümler arasındaki ilişkiyi gösteren çizgiler topluluğudur (Worboys, 1995). Pek çok problem, nesneler ve onlar arasındaki bağlantılar/ilişkiler kullanılarak biçimlendirilir. Örneğin, Berlin’le Antalya ve Chicago arasında uçak seferi varsa, ama Antalya’yla Chicago arasında uçak seferi yoksa bu durum, Şekil 1’deki gibi gösterilebilir. Burada düğümler şehirleri, düğümlerin arasındaki hatlar da “uçak seferi var” ilişkisini simgeler. Aynı şekil üç kişi arasındaki tanışıklık ilişkisini göstermek için de kullanılabilir. Örneğin Barış, Ahmet ve Canan’ı tanıyor, fakat Ahmet’le Canan birbirini tanımıyorsa, bu durumu gösteren graf Şekil 1’deki gibi öncekiyle aynı olacak, fakat bu kez düğümler insanları, hatlar da tanışıklık ilişkisini simgeleyecektir (Küçükçifçi vd., 2003). Graf, yukarıdaki örneklerdeki gibi durumları modelleyen matematiksel bir nesnedir. Graf teorisi, birleştirici (combinatorial) matematiğin, üzerinde uzun yıllardan beri çalışılan bir temel alt dalıdır. Hat ve düğümlerden oluşan

graflar, geometrik değil ilişkisel bilgiler içerirler. Hatların boy ve şeklinin, doğrusal ya da eğrisel oluşunun ve düğümlerin konumunun bir önemi yoktur. Her bir hat iki düğüm arasındaki bir ilişkiyi simgelediğinden, önemli olan tek şey var olup olmadıklarıdır. Dolayısı ile Şekil 1'de görülen graf daha sade bir yapıda Şekil 2'deki gibi de çizilebilir.

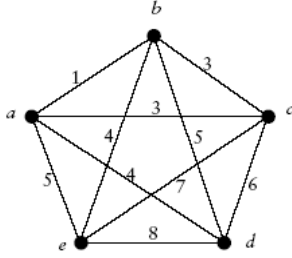


Şekil 1: Nesneler ve aralarındaki ilişkiler

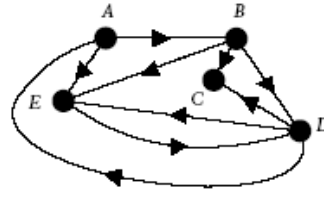


Şekil 2: Aynı grafın farklı gösterimi

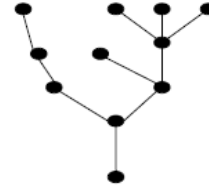
Graf yapısındaki hatlar değer alabilir ve bu değerler grafın yapısına katılabilir. Bir grafın üzerindeki hatların değerleri eşit değilse ve her biri farklı bir değer alabiliyorsa bu tip graflara **maliyetli yada ağırlıklı graf** (weighted graph) denir (Şekil 3). Bütün hatların değeri aynı ise bu graf maliyetli graf olarak anılamaz. Ağırlıkların bir anlamı yoktur ve her hattın değerinin 1 olduğu basit graf gibi değerlendirilir. Şehirlerin arasındaki mesafelerin hatlara değer olarak atandığı yol haritasını temsil eden graflar maliyetli graflar için örnek verilebilir (Çölkesen, 2001). Yada iş akış şemalarındaki, her işin bitirme süresini gösteren graflar da yine maliyetli graflar için bir başka örnektir. Eğer bir graftaki hatlar yön bilgisine sahipse bu tür graflara **yönlü graf** (directed graph) denir (Şekil 4). Bu yön bilgisi bağlantının nereden başlayıp nereden bittiğini belirtir. Yön bilgisi olan graflarda düğümler arasındaki bağlantının yönü vardır. Eğer iki yönde bağlantı varsa ters yönde iki ayrı hat kullanılır. Graf yapısında bütün kenarlar aynı çeşittir. Yani ya hepsi yönlüdür ya da değildir (Gürpınar, 2005). Yol ağını temsil eden bir grafta trafiğin tek yada çift yönlü oluşu yönlü graflar için bir örnektir. **Döngü** ise, başladığı düğüme geri dönen ve aynı düğümden iki kez geçmeyen bir yolculuktur. Bir graftaki hat sayısı düğüm sayısına eşit yada fazlaysa, o graf en az bir döngü içeriyor demektir. İçinde döngü barındırmayan grafa **ağaç** adı verilir (Şekil 5). Ağaca bir hat eklendiğinde mutlaka bir döngü içerir. Bir ağaçtaki hat sayısı düğüm sayısının bir eksiği kadardır.



Şekil 3: Maliyetli graf



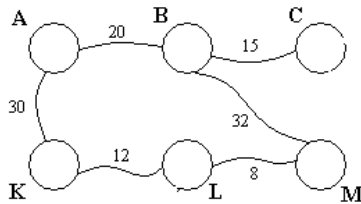
Şekil 4: Yönlü graf



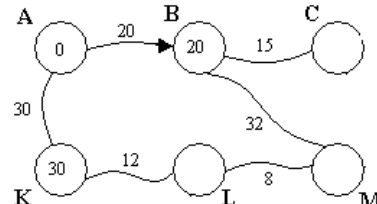
Şekil 5: Ağaç

En Az Dallanan Ağaç: Bir grafın herhangi bir noktasından başlayan en az dallanan ağacı bulmak için, o noktadan itibaren toplam düğüm sayısının bir eksiği kadar iterasyon yapılması gerekir. Bu işlem sırasında varılan her düğümün başlangıç düğüme (kök düğüm) uzaklığı belirlenir ve kayıt altına alınır. Ardından uzaklığı belirlenen bu düğümlerden en küçüğü işaretlenir. Bir sonraki aşamada aynı işlem işaretlenen düğümden itibaren gerçekleştirilir ve bu şekilde tüm düğümlere ulaşıncaya kadar devam edilir. En az dallanan ağaç bulunarak, kök düğüm ile diğer tüm düğümler arasındaki ulaşım maliyetinin en küçük olması sağlanır. Konu bir örnek üzerinde detaylandırılacak olursa, Şekil 6'daki graf 6 şehirden oluşan ve aralarındaki mesafeleri verilmiş bir yol ağı olsun. A noktası kök düğümü olarak seçildiğinde en az dallanan ağacın bulunması için adım adım şunlar yapılmalıdır:

1. A düğümünden yola çıkıldığında iki alternatif vardır; K ve B düğümleri. Bunların A'ya olan uzaklıkları belirlenir ve bu değerler kendi içlerine yazılır. Ardından değeri küçük olanı, yani B düğümü seçilir ve ok ile işaretlenir (Şekil 7).



Şekil 6: 6 düğümden oluşan bir maliyetli graf

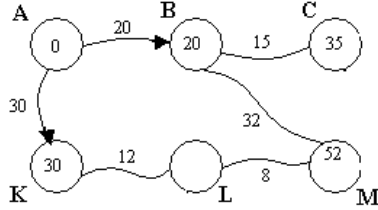


Şekil 7: En küçük değer B düğüme ait

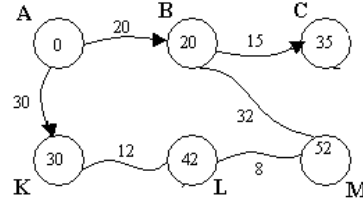
2. Önceki aşamada işaretlenen düğümden yani B'den sonra gelen düğümlere bakılır ve içlerine uzaklıkları yazılır. Buna göre C=35, M=52 değerlerini alır. Ardından kök düğümü (A) ve işaretlenmiş düğümler (B) hariç olmak üzere,

içine uzaklık bilgisi yazılmış olan tüm düğümlere bakılır ve en küçüğü işaretlenir. Dolayısı ile $C=35$, $M=52$ ve $K=30$ olduğundan K seçilir ve içine yazılı olan değeri veren yol esas olmak üzere, ok ile işaretlenir (Şekil 8).

3. Son işaretlenen düğüm K olduğu için onu izleyen düğüm(ler)e bakılır ve uzaklıklar yazılır. Buna göre K'nın ardından gelen düğüm olan L, 42 değerini alır. Ardından işaretlenmemiş fakat içlerine mesafeleri yazılmış düğümlere bakıldığında, C'nin 35, M'nin 52 ve L'nin 42 olduğu görülür. En küçüğü $C=35$ olduğundan ve bu değeri B üzerinden aldığından, BC hattına ok eklenerek işaretlenir (Şekil 9).



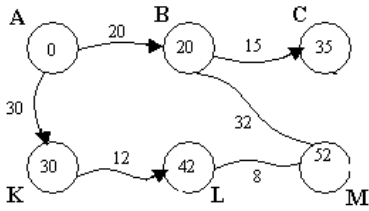
Şekil 8: En küçük değere sahip olan düğüm K



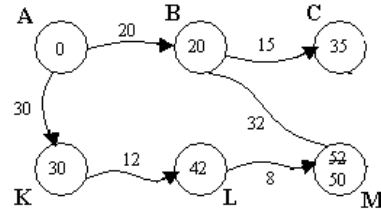
Şekil 9: En küçük değer C düğümüne ait

4. Daha sonra kalan iki düğümden değeri küçük olan belirlenir (L) ve işaretlenir (Şekil 10).

5. Son işaretlenen düğümün (L) devamındaki düğüm(ler)e bakılır ve değer atanır. Buna göre M $42+8=50$ olarak bulunur. Bu değer daha önce B üzerinden alınan 52 değerinden küçük olduğundan, 52 çizilir ve yerine 50 yazılır (Şekil) ve L'den gelen hat işaretlenir. (Şekil 11).

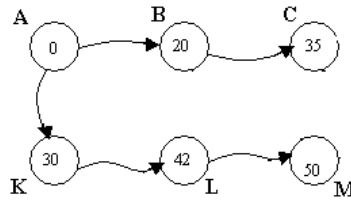


Şekil 10: Son iki düğümden küçük olanı L



Şekil 11: M düğümünün en düşük maliyeti

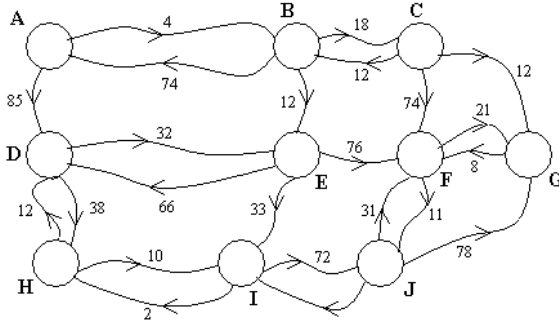
Tüm bu işlemlerin ardından A düğümü kök olarak seçilen yukarıdaki graf için en az dallanan ağaç Şekil 12'deki gibi elde edilmiş olur.



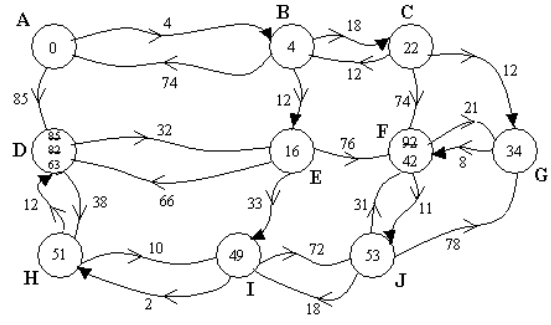
Şekil 12: Elde edilen en az dallanan ağaç

1.2 En Düşük Maliyetli Yolun Bulunması (Dijkstra Algoritması)

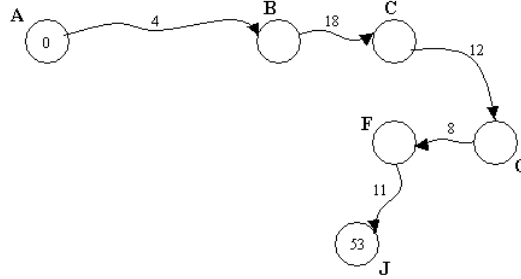
İki nokta arasındaki en düşük maliyetli (en kısa yol, en kısa süre vb.) yolun bulunması işleminde en az dallanan ağaç probleminin çözümünden faydalanılır. Bölüm 1.1'de de ayrıntılarıyla açıklandığı üzere en az dallanan ağaç bulunarak, kök düğüm ile diğer tüm düğümler arasındaki maliyetin en küçük olması sağlanır. Dolayısı ile başlangıç noktası kök düğüm olarak seçildiğinde, elde edilen en az dallanan ağaçtaki diğer noktaya giden yol, iki nokta arasındaki en düşük maliyetli yolu verecektir. Edsger W. Dijkstra tarafından 1959 yılında geliştirilen (Dijkstra, 1959) ve yönlü graflara da uygulanabilen en düşük maliyetli yol çözümü, **Dijkstra Algoritması** olarak da anılır. Şekil 13'da görülen yönlü ve maliyetli graf bir yol ağını ve düğümler de şehirleri temsil etsinler. Şehirler arasındaki gidiş ve dönüş güzergahları ve bu güzergahların uzunlukları birbirinden farklı olsun. Bu aşda A ile J şehri arasındaki en kısa yol bulunmak istendiğinde, öncelikle A noktasının kök düğüm olarak kabul edildiği en az dallanan ağaç elde edilmelidir. Dolayısı ile Bölüm 1.1'de açıklanan adımlar uygulanarak elde edilen en az dallanan ağaç Şekil 14'deki gibi olacaktır. Bu ağaçtaki A'dan J'ye giden toplam 53 birimlik yol, iki nokta arasındaki en kısa mesafeyi verecektir (Şekil 15).



Şekil 13: Graf gösterimiyle şehirleri ve yollar



Şekil 14: En az dallanan ağaç.



Şekil 15: A şehrinden J şehrine giden en kısa yol.

2. 3B AĞ ANALİZİ UYGULAMASI

Bir Coğrafi Bilgi Sistemi (CBS), veri entegrasyonu, yapılandırma, manipülasyon, analiz, karar destek ve sunum gibi tüm fonksiyonları sağlayabilmelidir. Bugün yaygın olarak kullanılan CBS yazılımları, 2B verinin yönetimi anlamında bu fonksiyonları başarıyla gerçekleştirebilmektedirler. Fakat böyle bir sistemden, ileri düzeyde 3B uygulamalar talep edildiğinde ise istenen sonuçlar elde edilememektedir (Raper ve Kelk, 1991; Rongxing Li, 1994; Stoter ve Zlatanova, 2003). İdeal anlamda 3B CBS'lerden bahsetmek için bir çok alanda çözülmeyi bekleyen altyapı sorunları mevcuttur. Bu alanlardan birisi de çok katlı ve karmaşık yapılardaki 3B Ağ Analizleridir. Aşağıda, üç boyutlu bir ağda otomatik olarak en kısa yolların hesaplanması ve coğrafi veritabanında bir araya getirilmesine yönelik olarak geliştirilen uygulama detaylandırılacaktır.

Öz konusu uygulamada üç boyutlu ağ modeli iki farklı tablodan oluşmaktadır: Düğüm (Nodes) ve Hatlar (Arcs) (Şekil 16). Ağ Modelinde her bir düğüm tektir (unique) ve her bir düğümün koordinatı "Nodes" tablosunda bir kez saklanmaktadır. Dolayısı ile tablolar hat-düğüm yapısını esas alan topolojik bir yaklaşımla oluşturulmaktadır. Koordinat bilgileri sayesinde ağ görselleştirilmekte ve düğümler arasındaki mesafeler hesaplanabilmektedir. Arcs tablosu ise graf yapısında, düğümler arasındaki bağlantıları (hatları) içermektedir ve bu bağlantılar başlangıç ve bitiş düğümleri ile tanımlıdır. Bunun dışında iki düğüm arasındaki maliyet de Arcs tablosuna kayıtlıdır. Maliyet düğüm koordinatlarına bağlı olarak hesaplanan uzaklık bilgisi olabileceği gibi, zaman vb. gibi ölçütler de kullanılabilir.

Nodes : Tablo					Arcs : Tablo			
	Node_ID	x	y	z		FromNode	ToNode	Distance
▶	1	92.001347826	145	5	▶	1	2	8
	2	286	145	5		3	4	6
	3	92.001	293	5		1	5	6
	4	91.999870229	145	5		6	3	3
	5	92	83	5		7	8	4
	6	62	319	5		8	3	8
	7	50	247	5		4	9	1
	8	134	253	5				

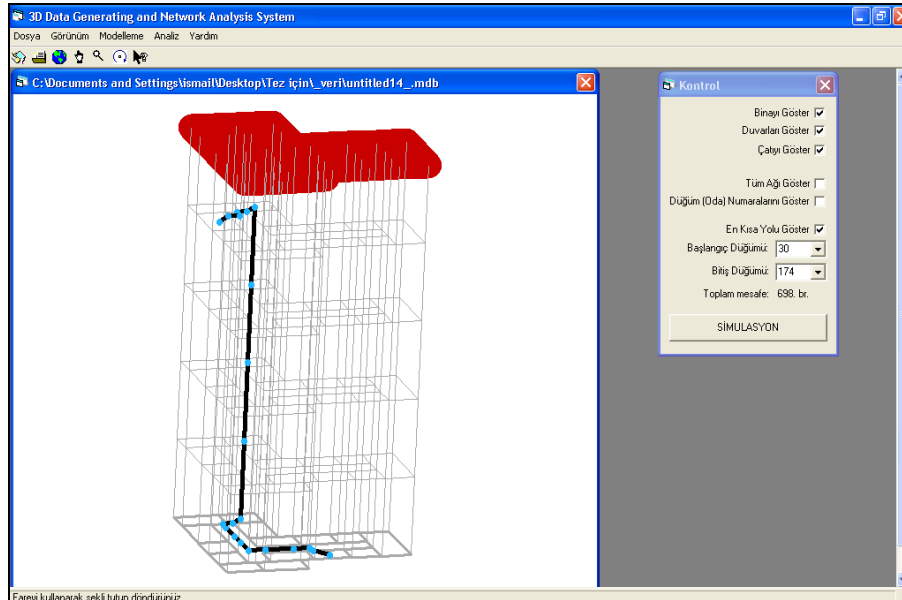
Şekil 16: Konumsal veritabanında ağ modeli

2.1 En Düşük Maliyetli Yolların Hesaplanması ve Coğrafi Veritabanına Kaydedilmesi

Uygulama kapsamında en düşük maliyetli yollar hesaplanmakta ve coğrafi veri tabanında bir araya getirilmektedir. Bölüm 1.2’de detaylı olarak açıklanan Dijkstra algoritması (Dijkstra, 1959) kullanılarak gerçekleştirilen işlem ile, Ağ Modelindeki her bir düğüm için, “en az dallanan ağaç” bulunmakta, dolayısı ile diğer tüm düğümlere ulaşmak için gerekli en kısa yollar hesaplanmaktadır. Her bir düğüm için elde edilen “en az dallanan ağaç”, tablo yapısında gösterilmekte ve otomatik olarak konumsal veritabanında kayıt altına alınmaktadır. Dolayısı ile düğüm sayısı kadar tablo veritabanında otomatik olarak oluşturulmakta ve her bir tablo da, o düğümünden diğer düğümlere en kısa yoldan nasıl gidileceği bilgisini içermektedir. Şekil 17’de 44 numaralı düğüme ait tablo görünmektedir. Bu tablodan 44 numaralı düğümden diğer tüm düğümlere en kısa yoldan nasıl ulaşılacağı bilgisini edinmek mümkündür. Buna göre mesela 29 numaralı düğümden 44 numaralı düğüme gitmek için tabloyu incelediğimizde: Node: 29 - Path: 23, Node: 23 - Path: 26, Node: 26 - Path: 46, Node: 46 - Path: 43, Node: 43 - Path: 44 sırasını elde edilmektedir. Dolayısı ile 29 numaralı düğümden 44 numaralı düğüme en uygun yoldan gitmek için 29-23-26-46-43-44 numaralı düğümleri takip edilmelidir. En kısa yolların bu şekilde hesaplanmasının ve bir coğrafi veri tabanında bir araya getirilmesinin ardından, geliştirilen 3B Ağ Analizi Sistemi ile karmaşık ve çok katlı binalarda optimum yol analizi yapmak artık mümkün hale gelmektedir (Şekil 18).

44 : Tablo				
	Node	Known	Cost	Path
	44	-1	0	
	43	-1	6	44
	46	-1	9	43
	47	-1	10	43
	48	-1	14	43
	49	-1	7	43
	26	-1	109	46
	66	-1	109	46
	23	-1	112	26
	6	-1	209	26
	63	-1	112	66
	86	-1	209	66
	24	-1	118	23
	27	-1	116	23
	28	-1	120	23
	29	-1	113	23

Şekil 17: 44 numaralı düğüm için en az dallanan ağacın konumsal veri tabanında gösterimi



Şekil 18: 3B Ağ analizi sisteminde iki nokta arasında en kısa yol analizi.

3. SONUÇLAR

Klasik ağ analizi uygulamalarının ötesinde, geliştirilen uygulama ile, 3B bir ağ için graf yapısındaki verilerin coğrafi veri tabanında bir araya getirilmesi ve en kısa yolların otomatik olarak hesaplanması sağlanmıştır. Böylece bu veri alt yapısının kullanılmasıyla, büyük ve karmaşık yapılardaki 3B ağ analizlerinin etkin ve verimli bir şekilde yapılabileceği gösterilmiştir.

KAYNAKLAR

Çölkese, R., 2002, “*Veri Yapıları ve Algoritmalar*”, Papatya Yayınevi, İstanbul.

Dijkstra, E. W., 1959, “*A note on two problems in connexion with graphs*”, *Numerische Mathematik*, 1, 269-271.

ESRI, 1996, *Using The ArcView Network Analyst GIS*, Environmental Systems Research Institute Inc., Redlans.

Güngör, V., 1999, *Coğrafi Bilgi Sistemlerinde Ağ Analizi*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Gürpınar, B., Alp, R., Yakın, Y., 2005, “*Grafler*”, Veri Yapıları ve Algoritmalar Dersi Semineri, Gebze Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Bölümü.

Küçükçifçi, S., Sural, H., Nesin, A., 2003, “*Kapak Konusu: Çizgeler*”, *Matematik Dünyası*, Sayı: 2003/3, Sayfa: 9-46.

Raper, J. ve Kelk, B., 1991, *Three-dimensional GIS. In: Geographical information systems: principles and applications*, D. J. Maguire, M. Goodchild and D. W. Rhind (eds.), Longman Geoinformation, pp. 299-317

Rongxing, L., 1994, “*Data structures and application issues in 3-D geographic information Systems*”, *Geomatica*, Vol. 48, No. 3, pp. 209-224

Stoter, J. ve Zlatanova, S., 2003, “*3D GIS where are we standing?*”, Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, 2-3 Ekim, Quebec city, Canada, 6p.

Worboys, M., 1995, *GIS: a computing perspective*. Taylor & Francis publication, 376 p.