

Pekiştirmeli Öğrenme Yöntemi ile Dinamik Kullanıcı Dengesi Altında Ulaşım Ağ Tasarımı

Cenk OZAN* Özgür BAŞKAN* Soner HALDENBİLEN* Halim CEYLAN*

**İnşaat Mühendisliği Bölümü
Mühendislik Fakültesi*

Pamukkale Üniversitesi, Kınıklı, Denizli

Email: cozan@pau.edu.tr obaskan@pau.edu.tr shaldenbilen@pau.edu.tr halimc@pau.edu.tr

Özet

Bu çalışmada dinamik kullanıcı dengesi altında Ulaşım Ağ Tasarımı (UAT) yapılmıştır. UAT problemini çözmek için iki seviyeli programlama tekniğinden yararlanılmıştır. Üst seviyede TRANSYT-7F yazılımı, alt seviyede ise simülasyon bazlı dinamik trafik atama yazılımı olan DynusT kullanılmıştır. Çalışmada, Q-öğrenme algoritması bazlı modifiye pekiştirmeli öğrenme algoritması kullanılarak UAT probleminin çözümü için yeni bir model geliştirilmiştir. Geliştirilen modelin örnek ulaşım ağı üzerinde sayısal uygulaması yapılmıştır. Elde edilen sonuçlara göre pekiştirmeli öğrenme algoritmalarının UAT probleminin çözümünde kullanılabileceği belirlenmiştir.

1. Giriş

Ulaşım ağının tasarım amacına göre en uygun performansı sağlayacak şekilde işletilmesi Ulaşım Ağ Tasarımı (UAT) olarak tanımlanmaktadır. UAT probleminin çözümü, konveks olmayan yapısından dolayı oldukça zor ve karmaşıktır. Problemin çözümü için iki seviyeli programlama tekniği son yıllarda literatürde oldukça fazla yer bulmaktadır. Bu yöntemde sinyal kontrolü ve Trafik Atama (TA) problemlerinin çözümü genellikle birbirinden ayrı olarak gerçekleştirilmektedir. İki problemin çözümü sırasında aslında girdi olarak kullanılan değerler diğer problemin çözümünden elde edilen çıktı değerleridir. UAT probleminde üst seviye sinyal kontrol problemini, alt seviye ise TA problemini temsil etmektedir [1].

Dinamik Trafik Atama (DTA) yöntemleri, statik atama yöntemlerinde olduğu gibi seyahat talebi sonucunda ağıdaki link trafik hacimlerini belirlemek için uygulanmaktadır. Fakat DTA'da seyahat talebi zamana bağlı olarak dikkate

alınmaktadır. Zaman parametresinin de probleme dahil olması karmaşıklığı artırmaktadır.

Bu çalışmada, UAT probleminin çözümü için iki seviyeli programlama tekniği kullanılmıştır. Üst seviye olan sinyal kontrol probleminde TRANSYT-7F yazılımı kullanılarak ulaşım ağının performans indeksi değeri elde edilmiştir. Alt seviye olan TA probleminde ise DynusT yazılımı kullanılarak link trafik hacimleri elde edilmiştir. İki seviyeli programlama modelinin çözümü için modifiye pekiştirmeli öğrenme algoritması kullanılarak Pekiştirmeli Öğrenme TRANSYT-7F DynusT (PÖTRAD) modeli geliştirilmiştir.

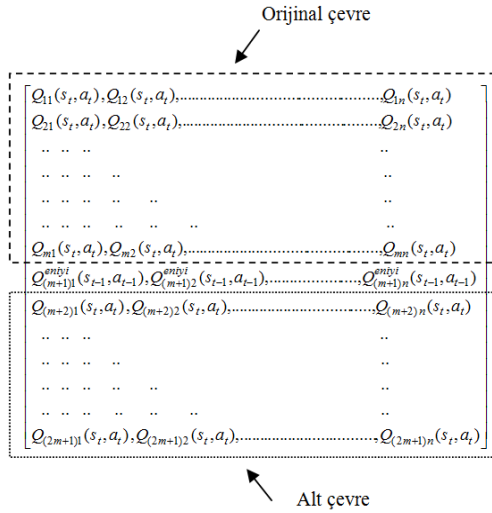
2. Pekiştirmeli Öğrenme Yöntemi

Pekiştirmeli Öğrenme (PÖ) yaklaşımlarının temel felsefesi gerçekleşen bir olgunun ödüllendirilmesi veya cezalandırılmasını bir mantık içerisinde düzenleyerek istenen bir sonucun ortaya çıkmasını sağlamaktır. PÖ'de karar verici ajan olarak adlandırılmakta ve çevresi ile etkileşim içinde olmaktadır. Bu etkileşim, ajanın çevre içinde uygulayacağı eylemi seçmesini sağlamaktadır.

Q-öğrenme algoritması, PÖ yöntemleri içinde modele ihtiyaç duymayan bir yaklaşım olup bu algorithmada çevrenin nasıl çalıştığı hakkında ajana herhangi bir bilgi verilmemektedir. Bunun yerine ajan eylemleri deneyerek en iyi ödülü veren eylemi seçmektedir. Q-öğrenme algoritması, durum-eylem çiftinin sahip olduğu değerlerin tahmin edilmesine bağlı olarak çalışmaktadır. Q değerleri olarak adlandırılan bu değerler verilen bir durum-eylem çifti için sayısal tahminler olarak nitelendirilmiştir [2]. Algorithmada, Q tablosu olarak adlandırılan tablonun elemanları her durum değişiminde güncellenmektedir [3]. Bu tabloda her s durumu ve a eylemi çifti için $Q(s,a)$ değeri bulunmaktadır. Ajan, çevrenin s_t durumundan, s_{t+1} durumuna geçişinde yapılan a_t eyleminin

ardından r_{t+1} olarak belirtilen ödülü almaktadır [4].

PÖ ile diğer sezgisel metotların hibrit edilmesi sonucu elde edilen algoritmaların, farklı alanlardaki optimizasyon problemlerinin çözümünde kullanılması oldukça etkin yollardan biridir. Q-öğrenme algoritması tabanlı modifiye PÖ algoritmasında her öğrenme evresinde önceki en iyi çözüm bilgisinin yardımıyla orijinal çevre boyutunda bir alt çevre Şekil 1’de görüldüğü gibi oluşturulmaktadır.



Şekil 1. Orijinal-alt çevre ilişkisi

Burada, m çevrenin boyutunu, n karar değişkenlerinin sayısını ve t öğrenme evresinin sayısını ifade etmektedir. Şekil 1’den görüldüğü üzere, önceki öğrenme evresinden elde edilen en iyi $Q(s, a)$ değeri, yerel optimuma takılmayı önlemek için çevrenin $(m+1)$ 'inci satırında saklanmaktadır. Ayrıca alt çevre, önceki öğrenme evresinde elde edilen en iyi değer ve β vektörü kullanılarak oluşturulmaktadır. Böylelikle, global optimum, algoritma süreci boyunca alt çevrede β değeri ile sınırlandırılan alt çevrede araştırılmaktadır. β değerinin aralığı verilen probleme bağlı olarak seçilebilmektedir [5]. Sonrasında orijinal çevre ve alt çevre iyiden kötüye doğru sıralanmaktadır. Bu şekilde, bir önceki öğrenme evresinden elde edilen en iyi çözüm ve alt çevre, orijinal çevre ile karşılaştırılmaktadır. Çözümlerden biri kötü olanlardan daha iyi değer sağlıyorsa, yeni değerler orijinal çevrenin içine dahil edilirken kötü değerler çevreden çıkarılmaktadır. Algoritmanın çözüm adımları Şekil 2’de gösterilmektedir.

Modifiye PÖ algoritmasında $r_t(s, a)$ olarak belirtilen ödül fonksiyonunun değeri en iyi Q

değeri ile öğrenme evresindeki Q değeri arasındaki farkın Q değerine bölümü olarak belirlenmektedir. Ödül değerleri, fonksiyonunun yapısından dolayı “0” değerine yaklaşmaktadır. Diğer bir deyişle modifiye PÖ algoritmasında toplam ödül değerinin, verilen optimizasyon probleminin global optimum değerinin bulunabilmesi için minimize edilmesi gerekmektedir.

Adım 0: $t=0$. Algoritma parametrelerini(β, a, γ) ayarla
Adım 1: If $t=0$, başlangıç $Q(s, a)$ değerlerini oluştur
Amaç fonksiyonunun değerlerini hesapla
Else
En iyi $Q_{t-1}^{en\ iyi}(s, a)$ sakla
$\beta_t = \beta_{t-1} * 0.99$
$rastgele(Q_{t-1}^{en\ iyi}(s, a) - \beta : Q_{t-1}^{en\ iyi}(s, a) + \beta)$ denklemi ile alt çevreyi oluştur
end if
Adım 2: Çevre ve alt çevreyi iyiden kötüye doğru sırala
En iyi $Q_t^{en\ iyi}(s, a)$ bul
$r_t(s, a)$ değerini $\lim_{r_t(s, a) \rightarrow 0} \frac{(Q_t^{en\ iyi}(s, a) - Q_t(s, a))}{Q_t(s, a)} = 0$ denklemi ile hesapla
Adım 3: $Q_t(s, a) = (1 - \alpha) \times Q_{t-1}(s, a) + \alpha \times [r(s, a) + \gamma \times \max_{a'} Q_{t-1}(s', a')]$ denklemi ile Q değerlerini güncelle
Adım 4: t_{max} (maximum öğrenme evresi sayısı)’a ulaşıldıysa algoritmayı durdur. Aksi takdirde $t=t+1$ yap Adım 1’e git.

Şekil 2. Modifiye PÖ algoritması.

3. Ulaşım Ağ Tasarımı

UAT, TA ve sinyal kontrol aşamalarından oluşmaktadır. TA probleminin amacı ulaşım ağı topolojisi, link maliyet fonksiyonu ve Başlangıç-Varış (B-V) seyahat miktarlarına göre link akımlarının bulunmasıdır. Problemin çözümü her bir ağ kullanıcısının B-V çiftleri arasındaki en düşük maliyetli rota üzerinde seyahat ettiği kabulüne dayanmaktadır [6]. Denge durumunda herhangi bir B-V çiftini arasındaki rotalardaki seyahat maliyetlerinin eşit olacağı ve ayrıca bu rotalardaki seyahat maliyetlerinin kullanılmayan rotalardaki seyahat maliyetine eşit veya ondan daha az olacağı kabulü yapılmaktadır. Bu durumda ulaşım ağının Kullanıcı Dengesi (KD) altında hizmet verdiği kabulü yapılmakta ve hiçbir sürücü tek taraflı olarak rotasını değiştirerek kendi seyahat maliyetini azaltamamaktadır [7].

KD yaklaşımının DTA’da kullanılabilmesi için rota seçiminde ağ üzerindeki linklerin seyahat sürelerinin zamanla değiştiği kabulü yapılmalıdır. Statik atama yöntemlerinde ağ kullanıcısının seyahati boyunca geçmişteki deneyimlerinden yararlanarak trafik koşullarını tahmin ettiği ve bir B-V çifti arasındaki rota seçimini seyahat süresini

minimize edecek şekilde yaptığı varsayımı bulunmaktadır. Bu durum DTA'da farklıdır. En kısa seyahat süreli rota, ağ kullanıcısının rota üzerindeki linklere giriş yaptığı zamana ve rota üzerinde ilerlerken linkler üzerinde geçerli olan seyahat süresine bağlı olarak değişmektedir. Diğer taraftan herhangi bir B-V çifti için farklı zamanlarda hareket eden kullanıcılar farklı seyahat süreleri ile karşı karşıya kalacakları için KD altındaki rotalar üzerinde seyahat süresinin eşitliği sadece aynı B-V çifti arasında ve aynı anda hareket ettiği kabulü yapılan kullanıcılar için geçerli olacaktır. KD yaklaşımının DTA'da kullanılabilmesi amacıyla, zamanla değişen ulaşım ağı trafik koşullarını tanımlayan ve zamanla değişen link seyahat sürelerini dikkate alarak en kısa rotayı bulmayı sağlayan zamana bağlı en kısa rota algoritması geliştirilmiştir [8].

Çalışmada TA probleminin çözümünde simülasyon bazlı dinamik trafik atama programı DynusT kullanılarak link hacimleri bulunmuştur. DynusT (Dynamic Urban Systems for Transportation) yazılımı Arizona Üniversitesi ve DynusT laboratuvarı tarafından geliştirilen simülasyon bazlı DTA yazılımıdır. Program zamana bağlı en kısa rota algoritmasını kullanarak sistem optimumu bulabilmekte ya da KD altında dinamik trafik ataması yapabilmektedir.

TRANSYT (TRAffic Network StudY Tool) İngiltere Ulaşım ve Yol Araştırma Laboratuvarı (TRRL) tarafından geliştirilmiştir [9]. TRANSYT-7F faz bazlı sinyal optimizasyon yazılımı olup trafik modeli ve sinyal süresi optimize edici olmak üzere iki ana kısımdan oluşmaktadır. TRANSYT-7F trafik modeli sinyalizasyon kavşaklı bir ağda girilen sinyal süre ve faz planlarına bağlı olarak ağı Performans İndeksi (PI) değerini hesaplamak için trafiği simüle etmektedir. PI değeri, ağ üzerinde birim zamandaki duruş sayısı ile ağ üzerinde elde edilen gecikmenin ağırlıklandırılmış toplamıdır ve Denklem (1)'de verildiği gibi ifade edilmektedir.

$$PI = \sum_{a=1}^L (w_{d_a} d_a + K w_{s_a} S_a) \quad (1)$$

Burada, L link sayısı, d_a ta-sa/sa cinsinden a linki üzerindeki gecikme değeri, K duruş cezası faktörü, S_a bir saniyede a linki üzerindeki taşıt-duruş sayısı, w_{d_a} ve w_{s_a} sırasıyla gecikme değeri ve duruş sayısının ağırlık katsayılarıdır [10].

3.1. PÖTRAD Modeli

Bu çalışmada Dinamik Kullanıcı Dengesi (DKD) altında UAT problemini iki seviyeli programlama tekniği ile çözmek amacıyla

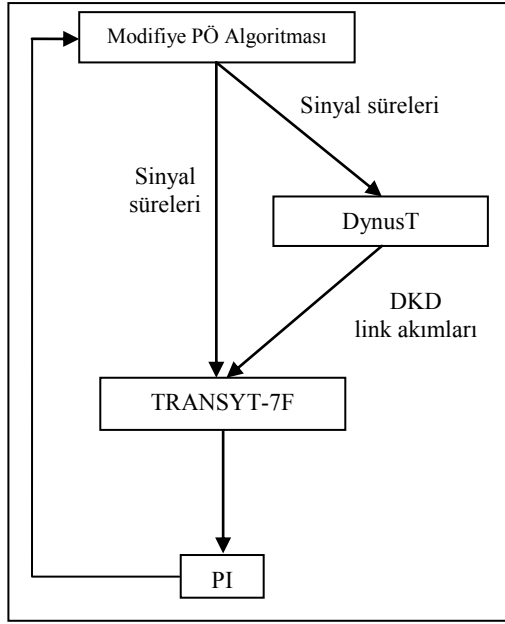
PÖTRAD modeli geliştirilmiştir. Modelin üst seviyesi olan sinyal kontrol aşamasında TRANSYT-7F yazılımı kullanılarak ulaşım ağının PI değeri, alt seviyesi olan TA aşamasında ise DynusT yazılımı ile DKD altında link trafik hacimleri elde edilmektedir.

Modifiye PÖ algoritmasının sinyal kontrol probleminde uygulanmasında her bir $Q(s,a)$ değeri sinyal kontrol karar değişkenlerini temsil etmektedir. Her öğrenme evresinde, sinyal süre kısıtlarına bağlı olarak karar değişkenlerinin sayısı kadar sinyal süreleri üretilmektedir. Sinyal süreleri ve TRANSYT-7F yazılımı kullanılarak $f(Q(s,a))$ olarak adlandırılan modelin amaç fonksiyonu diğer bir deyişle ağın PI değeri hesaplanmaktadır. Geliştirilen algoritmada, $Q_i^{en iyi}(s,a)$ değeri i 'inci öğrenme evresinde en iyi PI değerini veren sinyal süreleri olarak tanımlanmaktadır.

Geliştirilen PÖTRAD modelinin amaç fonksiyonu ve kısıtları Denklem (2)'de verilmektedir.

$$\begin{aligned} \min_{\Psi} PI(\mathbf{q}^*(\Psi), \Psi) \\ \Psi(c, \theta, \phi) \in \Omega_0 \end{aligned} \quad (2)$$

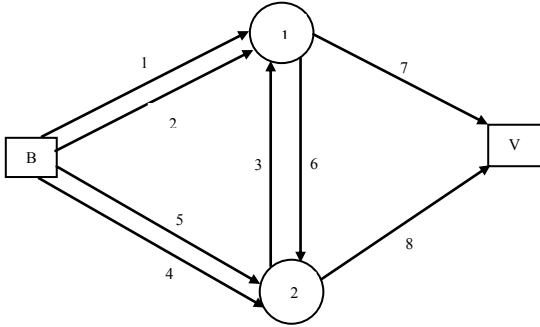
Burada, Ψ sinyal kontrol parametreleri (c devre süresi, θ her kavşak için ofset süresi, ϕ her fazın yeşil süresi), Ω_0 sinyal kontrol parametreleri için çözüm uzayı, $\mathbf{q}^*(\Psi)$ ise DKD link trafik hacimleridir. DKD link trafik hacimleri PÖTRAD modelinin alt seviyesinde DynusT yazılımı ile edilmektedir. DKD altındaki link hacim değerleri ile sinyal kontrol parametreleri modelin üst seviyesinde TRANSYT-7F yazılımına girdi olmakta ve ağın PI değeri hesaplanmaktadır. Çalışmada modifiye PÖ algoritması ile bu işlemler her öğrenme evresinde tekrar edilerek PI değerini minimize eden en uygun sinyal kontrol parametrelerinin bulunması amaçlanmaktadır. Şekil 3'de PÖTRAD modelinin genel çerçevesi görülebilmektedir.



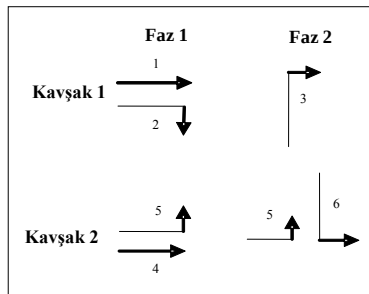
Şekil 3. PÖTRAD modelinin genel çerçevesi.

4. Sayısal Uygulama

PÖTRAD modelinin sayısal uygulaması iki kavşaklı bir test ağı üzerinde yapılmıştır. Örnek ulaşım ağında 1 adet B-V çifti, 2 adet kavşak ve 8 adet link bulunmaktadır. Ulaşım ağı ve kavşaklara ait sinyal faz düzenleri sırasıyla Şekil 4a ve 4b'de görülebilmektedir. Ulaşım ağında tüm linklerin doygun akım değerleri 1800 ta/sa, 7 ve 8 nolu linklerin serbest akım seyahat süreleri 10 sn, diğer tüm linklerin 20 sn'dir. B-V talebi 1500 ta/sa'dır.



Şekil 4a. Örnek ulaşım ağı [6].



Şekil 4b. Sinyal faz planları [6].

Örnek ulaşım ağının çözümünde kullanılacak sinyal parametrelerinin maksimum ve minimum değerleri aşağıda verilmektedir.

$$36 \leq c \leq 90$$

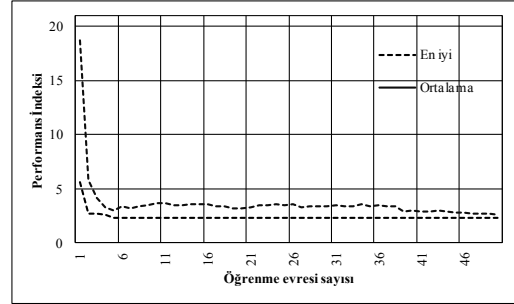
$$0 \leq \theta \leq c$$

$$7 \leq \phi \leq c$$

$$I_{1-2} = I_{2-1} = 5 \text{ sn}$$

Burada c devre süresi (sn) ve I yeşillerarası süre (sn) olarak ifade edilmiştir. PÖTRAD modelinin ulaşım ağına uygulanması aşamasında PÖ algoritması parametreleri; öğrenme oranı (α) 0.8, azaltma faktörü (γ) 0.2, çevre boyutu (m) 20 ve maksimum öğrenme evresi sayısı (t_{max}) 80 olarak seçilmiştir. Çözüm prosedürü verilen durma kriteri sağlanıncaya kadar devam ettirilmiştir. t 'inci öğrenme evresinde ulaşım ağının en iyi PI değeri ile ortalama PI değeri arasındaki fark % 10'dan az olduğunda PÖTRAD modeli sonlanmaktadır.

1500 ta/sa'lık ulaşım talebi, ulaşım ağına her 10 dk'lık zaman dilimlerinde 250 ta/sa'lık talep olarak yüklenmiştir. Bir başka deyişle, 1500 ta/sa'lık talep analiz süresi boyunca ulaşım ağına düzgün dağılımlı olarak atanmıştır. PÖTRAD modelinin çalıştırılması sonucu ağı PI değerinin en iyi ve ortalama değerlerinin yakınsama grafiği Şekil 5'de verilmektedir.



Şekil 5. PÖTRAD modelinin yakınsaması.

Şekil 5'den görülebildiği gibi 50'inci öğrenme evresinde durma kriteri sağlandığı için model sonlanmıştır. Örnek ulaşım ağında en iyi PI değeri 2.33 olarak bulunmuştur. Elde edilen en iyi sinyal parametreleri ve DKD link akımları sırasıyla Tablo 1 ve 2'de verilmektedir.

Tablo 1. En iyi sinyal parametreleri.

PI	Devre süresi c (sn)	Kavşak no	Faz yeşil süreleri (sn)		Ofset (sn) θ_i
			Faz 1 $\phi_{i,1}$	Faz 2 $\phi_{i,2}$	
2.33	90	1	80	10	26
		2	80	10	0

Tablo 2. DKD link akımları (ta/sa).

q_1	q_2	q_3	q_4	q_5	q_6
365	382	54	10	54	367

Geliştirilen PÖTRAD modeli ağın PI değerini minimize edebilmektedir. Ayrıca Tablo 3’de ulaşım ağı için elde edilen istatistikler verilmektedir.

Tablo 3. Ulaşım ağından elde edilen istatistikler.

Toplam seyahat süresi (sa)	Ortalama seyahat süresi (dk)	Toplam seyahat mesafesi (ta-km)	Ortalama seyahat mesafesi (ta-km)	Ağda kalan taşıt sayısı
56.77	2.28	378.32	0.25	337

Tablo 3’den görüldüğü gibi simülasyon süresi sonunda ağı verilen ulaşım talebinin tamamının varış noktasına ulaşamadığı ve 337 taşıtın ağ içinde kaldığı belirlenmiştir. Ayrıca ulaşım ağındaki ortalama seyahat süresi yaklaşık 2.30 dk olarak bulunmuştur. Linklerin serbest akım seyahat süreleri dikkate alındığında bu sonuca göre ağın sıklık etki altında çalıştığı söylenebilir.

5. Sonuçlar

Bu çalışmada iki seviyeli programlama tekniğinden yararlanarak UAT problemi çözülmüştür. Bu amaçla PÖTRAD modeli geliştirilmiştir. Üst seviyede ağın PI değeri TRANSYT-7F yazılımı kullanılarak elde edilmiştir. Alt seviyede ise DynusT yazılımı kullanılarak DKD altındaki link trafik hacimleri bulunmuştur. Modifiye PÖ algoritması üst seviyede ağın PI değerinin minimize edilmesi amacıyla kullanılmıştır. Geliştirilen modelin sayısal uygulaması örnek ulaşım ağı üzerinde yapılmış ve elde edilen sonuçlar modifiye PÖ algoritmasının DKD altında UAT probleminin çözümünde başarılı olduğunu göstermiştir. Gelecek çalışmalarda PÖTRAD modelinin gerçek ulaşım ağları üzerindeki performansının test edilmesi amaçlanmaktadır.

6. Teşekkür

Bu çalışma Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Biriminin desteklediği olduğu BAP-FBE-063 nolu proje kapsamında gerçekleştirilmiştir.

7. Kaynaklar

- [1] Ö. Başkan, Karınca Kolonisi Optimizasyonu ile Ulaşım Ağ Tasarımı, Doktora Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü/İnşaat Mühendisliği Anabilim Dalı, Ekim, 2009.
- [2] A.L.C. Bazzan, D. Oliveira ve B.C. Silva, “Learning in Groups of Traffic Signals”, Engineering Applications of Artificial Engineering, Elsevier, 2010, 23, p. 560-568.
- [3] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, The MIT Press, Cambridge, Massachusetts, USA/London, England, 1998.
- [4] M. Vanhulsel, D. Janssens, G. Wets ve K. Vanhoof, “Simulation of Sequential Data: An Enhanced Reinforcement Learning Approach”, Expert Systems with Applications, 2009, 36, p. 8032-8039.
- [5] O. Baskan, S. Haldenbilen, H. Ceylan ve H. Ceylan, “A New Solution Algorithm for Improving Performance of Ant Colony Optimization”, Applied Mathematics and Computation, 211, 2009, s.75-84.
- [6] H. Ceylan, A genetic algorithm approach to the equilibrium network design problem, Ph.D. Thesis, University of Newcastle upon Tyne, 2002.
- [7] J. Wardrop, “Some Theoretical Aspect of Road Traffic Research”, Proceedings Institute Civil Engineers Part II, 1952, s. 325-378.
- [8] ADB30 Transportation Network Modelling Committee, A Primer for Dynamic Traffic Assignment, Transportation Research Board, 2010.
- [9] D.I. Robertson, TRANSYT: a traffic network study tool, RRL Report, LR 253, Transport and Road Research Laboratory, Crowthorne, UK, 1969.
- [10] McTrans Center, TRANSYT-7F Release 11.3 Users Guide, University of Florida, Gainesville, Florida, 2008.