

Bir Trafik Ağı Probleminin Sürekli-Zamanlı Dağıtık Algoritma Kullanılarak Çözümü

Solution of a Traffic Network Problem by Using a Continuous-Time Distributed Algorithm

Oğuzhan Çiftçi¹, Onur Cihan²

¹Elektrik ve Elektronik Mühendisliği Bölümü
Boğaziçi Üniversitesi, Bebek, 34342, İstanbul

oguzhan.ciftci@boun.edu.tr

²Elektrik ve Elektronik Mühendisliği Bölümü
Marmara Üniversitesi, Göztepe, 34722, İstanbul

onur.cihan@marmara.edu.tr

Özetçe

Bu bildiri, bir trafik ağı probleminin, sürekli-zamanda tanımlı ve $Ax = b$ formundaki doğrusal denklem çözmek için geliştirilmiş bir dağıtık algoritma kullanılarak çözümü incelenmiştir. Söz konusu trafik ağı problemi, çok etmenli sistemlerde her etmenin denklem sisteminin yalnızca bir satırını bildiği ve denklem sisteminin eşsiz çözümüne ulaşmayı hedeflediği bir formülasyon olarak ifade edilmiştir. Trafik ağındaki sensörler önlerinden birim zamanda geçen araç sayısını ölçebilmekte ve kendilerine yakın olan komşu sensörlerle veri paylaşımı yapabilmekte olduğu varsayımıyla, her bir güzergahtan birim zamanda geçen araç sayısının belirlenebilmesine çalışmaktadır. Sensörler arası iletişimin zamanla değiştiği ancak çizge dizisinin tekrarlı birlikte kuvvetli bağlı olduğu bir durum için trafik ağı probleminin çözüldüğü benzetim çalışmaları ile gösterilmiştir.

Abstract

In this paper, we investigate the solution of a traffic network problem by using a continuous-time distributed algorithm designed for solution of linear equations in the form of $Ax = b$. We express the traffic network problem as the solution of a linear equation defined in a multi-agent system where each agent knows a single line of the equation system and aims to find the unique solution. In the traffic network, sensors are able to measure the number of vehicles passing in front of them in unit time, and try to determine the number of vehicles passing from each route in unit time, assuming that sensors can share data with neighboring sensors close to them. Simulation studies have shown that the traffic network problem can be solved when the underlying topology of the network is time varying and repeatedly jointly strongly connected.

1. Giriş

Doğrusal cebir denklem sistemlerinin çözümleri için çok çeşitli algoritmalar uzun yıllardan beri geliştirilmektedir. Denklem sistemlerini dağıtık olarak çözmekte kullanılacak algoritmaların tasarımı konusu ise son zamanlarda birçok araştırmacı tarafından ilgiyle araştırılmaktadır [1–6, 8]. Dağıtık algoritmaların ortaya çıkmasındaki en büyük motivasyon, tüm denklem sisteminin merkezi bir çözücü vasıtasıyla değil de birbirine belirli komşuluk ilişkileriyle bağlı $m > 1$ sayıda etmenin, sistemin sadece belirli parçalarını bilmesine rağmen komşuları ile veri alışverişi sayesinde tüm denklem sistemini çözebilmesidir. Doğrusal cebir denklem sistemlerinin eşsiz çözümleri olabileceği gibi bazı denklemlerin sonsuz sayıda çözümü olabilir veya hiç çözümü mevcut olmayabilir. Eşsiz çözümün olduğu durumlar için geliştirilmiş dağıtık algoritmalar olduğu gibi [1–5, 8], çözümün olmadığı durumlarda en küçük kareler çözümünü bulmayı sağlayan algoritmalar da literatürde bulunmaktadır [7].

Yang ve Tang tarafından, yönsüz çizgelerde çözümü sağlayan ve sürekli zamanda bir algoritma tanımlanmıştır [9]. Bu algoritma birinci dereceden dinamiklere sahip olarak tasarlanmış olup, üstel hızla çözümü sağlamaktadır. Liu ve ark. tarafından önerilen ve sürekli zamanda tanımlanmış bir başka algoritma ise birinci dereceden integratör dinamiklerine sahiptir [4]. Bu algoritma, ağ çizgesinin tekrarlı birlikte kuvvetli bağlı olması durumunda yakınsamaktadır. Anderson ve ark. tarafından sürekli zamanda önerilen algoritma ile keyfi başlangıç vektörleri seçilerek denklem sisteminin eşsiz çözümünün dağıtık olarak bulunabilirdiği gösterilmiştir [8]. Her etmenin denklemin bir satırını bildiği durumda, sürekli zamanda denklem çözümünü sağlayan bir algoritma [10] ve [11]'de önerilmiş olup, kazanç katsayısı doğru şekilde seçilerek, denklemin çözümünün olmadığı durumlarda en küçük kareler çözümüne keyfi bir öklid uzaklığında bir çözüme ulaşabilmektedir. Zeng ve Cao tarafından yakın zamanda geliştirilen bir algoritma ise, etmenlerin A matrisinin bazı sü-

tunlarını bildiği durumu ele almıştır [12]. Bu algoritma kullanılarak denklemin en küçük kareler çözümüne ulaşmak mümkün hale gelmektedir.

Bu çalışmada, bir trafik ağında sadece belirli noktalardan birim zamanda geçen araç sayısını ölçen sensörler kullanılarak, ağda tanımlı tüm güzergahlardan birim zamanda geçen araç sayısının sensörler tarafından dağıtık olarak belirlenebilmesi konusu, doğrusal bir denklem sisteminin dağıtık olarak çözülebilmesi ile ilişkilendirilerek incelenmiştir.

Bildirinin kalan kısmı şu şekilde düzenlenmiştir. Bölüm 2’de çok etmenli ağların haberleşme düzenini açıklamakta kullandığımız çizge kuramıyla ilgili temel bilgiler verilmiş ve bu ağlarda doğrusal denklem sistemlerini çözmekte kullanılan dağıtık bir algoritma tanıtılmıştır. Bölüm 3’te trafik ağı problemi açıklanarak, problemin $Ax = b$ biçiminde bir doğrusal denklem sisteminin çözümü problemi ile nasıl ilişkili olduğu anlatılmıştır. Bölüm 4’te ele alınan örnek bir trafik ağı için, sensörlerin çözüm tahminlerinin problemin eşsiz çözümüne yakınsadığı benzetim çalışmaları ile gösterilmiştir. Elde edilen sonuçların uygulama alanları ve gelecek çalışmalar hakkında bilgiler ise Bölüm 5’te verilmiştir.

2. Çizge Kuramı ve Algoritma

Bu bölümde, trafik ağı probleminin modellenmesinde kullanılan çizge kuramı ve bu problemin çözülmesinde kullanılacak olan sürekli zamanda tanımlı dağıtık bir algoritma hakkında bilgiler verilmiştir.

2.1. Çizge Kuramı

Çok etmenli bir sistemde $m > 1$ sayıda etmen olduğu ve bu etmenlerden her birinin diğer bir etmene aralarındaki iletişim ağları sayesinde bilgi gönderebildiği varsayalım. Bu durumda $G = (V, E)$ çizgesi bu etmen sistemini temsil etmektedir. $V = \{1, 2, \dots, m\}$ bu sistemdeki etmenleri ve $E \subset V \times V$ etmenler arası bağlantıları ifade eden kümeler olarak tanımlanmaktadır. Eğer i . etmen j . etmene bilgisini gönderebiliyor ise, i . etmen j . etmenin komşusu olarak nitelendirilir ve bu $(i, j) \in E$ ile gösterilir. i . etmenin komşuluk kümesi N_i ile ve i . etmenin komşu sayısı $|N_i|$ ile ifade edilebilir. Bu bildiride her etmenin kendi bilgisini kullanabildiği, dolayısı ile kendisinin komşusu olduğu varsayılmıştır. Komşuluk ilişkilerini gösteren $C = [c_{ij}]$ bitişiklik matrisi

$$c_{ij} = \begin{cases} 1, & \text{eğer } (i, j) \in E \text{ ise,} \\ 0, & \text{eğer } (i, j) \notin E \text{ ise.} \end{cases} \quad (1)$$

olarak tanımlanır. Bir ağdaki etmenler arası bağlantılar zaman geçtikçe değişkenlik gösteriyorsa çizge değişken ilingeli, eğer değişmiyorsa sabit ilingeli çizge olarak isimlendirilir. Ayrıca çizgeler etmenler arasındaki bilgi akışının yönüne göre yönlü ve yönsüz olarak iki kısımda incelenir. Eğer i . etmen ile j . etmen arasındaki iletişim çift yönlü ise $((i, j) \in E, \iff (j, i) \in E, \forall i, j \in V)$ ise çizge yönsüz olarak adlandırılır. Bu durumun sağlanmadığı çizgeler ise yönlü olarak tanımlanır. Bir çizgede her etmen çifti arasında (bir ya da daha fazla adımda) bir iletişim mevcut ise, bu çizgeye kuvvetli bağlı çizge denir. Kuvvetli bağlılık sabit ilingeli çizgeler için tanımlanmış bir kavram olmasına karşın, değişken ilingeli çizgelerde tekrarlı birlikte kuv-

vetli bağlı çizge dizisi tanımı büyük önem taşır. Bu tanımın yapılabilmesi için gerekli olan çizge bileşimi tanımı aşağıda verilmiştir.

Tanım 1 (Çizge bileşimi) $G_1 = (V, E_1)$ ve $G_2 = (V, E_2)$ yönlü iki çizgeyi ifade ediyor olsun. Bu iki çizgenin bileşimi $G = (V, E) = G_1 \circ G_2$ olarak gösterilir ve her $(i, j) \in E$ için öyle bir k vardır ki $(i, k) \in E_1$ ve $(k, j) \in E_2$ sağlanır.

Çizge bileşimi tanımı kullanarak, değişken ilingeli çizgelerde tekrarlı birlikte kuvvetli bağlı çizge aşağıdaki gibi tanımlanabilir.

Tanım 2 (Tekrarlı birlikte kuvvetli bağlı çizge dizisi [1])

Sonsuz elemanlı G_1, G_2, \dots dizisi, her $k > 1$ değeri için $G_{kl+\tau_0-1} \circ G_{kl+\tau_0-2} \circ \dots \circ G_{(k-1)l+\tau_0}$ çizgelerini kuvvetli bağlı yapan l ve τ_0 tamsayıları varsa, tekrarlı birlikte kuvvetli bağlı çizge dizisi olarak tanımlanır.

2.2. Doğrusal Denklem Çözme Algoritması

Bu bölümde bir trafik ağında farklı güzergahları kullanan araç sayısının belirlenmesini sağlayacak ve sürekli zamanda tanımlı bir dağıtık algoritma tanıtılacaktır. Ele alınacak olan trafik ağı sistemi, komşuluk ilişkileri G yönsüz çizgesiyle ifade edilen m etmenli (sensörlü) bir ağ olarak tanımlanmaktadır. Her i etmeni, $Ax = b$ formundaki doğrusal denklem sisteminin yalnızca bir satırını bilmektedir. Yani $[A_i \ b_i]$ olarak gösterilebilen, $[A \ b]$ matrisinin i . satırını bilmektedir. Burada A_i vektörü A 'nın i . satırını ve b_i değeri b 'nin i . elemanını ifade etmektedir. Her etmen denklemin yalnızca bir satırını bilmenin yanı sıra, komşuları ile bildiği denklem sistemini çözen bir çözüm vektörü paylaşabilmektedir. Bu çözüm vektörlerini kullanarak kendi çözümünü bir algoritma aracılığıyla güncellemekte ve $Ax = b$ denklem sisteminin eşsiz çözümünü bulmayı hedeflemektedir. Sürekli zamanda doğrusal denklem sistemlerinin dağıtık olarak çözümünde kullanılabilecek bir algoritma [4]'te

$$\dot{x}_i(t) = -P_i \sum_{j \in N_i(t)} (x_i(t) - x_j(t)), \quad (2)$$

olarak önerilmiştir. Bu denklemdeki $x_j(t)$ vektörü j . etmenin t anındaki çözüm tahminini ifade ederken, $P_i \in R^{n \times n}$ matrisi A_i satır vektörünün sıfır uzayına dik izdüşüm matrisidir ve

$$P_i = I - A_i^T (A_i A_i^T)^{-1} A_i \quad (3)$$

olarak hesaplanabilir. Algoritmanın başlangıç değeri, etmenin kendi denklem sistemini çözen $(A_i x_i(0) = b_i)$ herhangi bir çözüm olarak seçilebilmektedir. Verilen bir $x_i(0)$ için $A_i x_i(0) = b_i$ olduğu garanti edildiğinden, (2)'ye göre $A_i x_i(t) = b_i$ denklemini her t için sağlanacaktır. Denklem dağıtık olarak çözülebilmesi için her etmenin çözümünün, denklemin eşsiz çözümünü olan x^* vektörüne yakınsaması gerekir, yani her i etmeni için $\lim_{t \rightarrow \infty} \|x_i^* - x_i(t)\| = 0$ olmalıdır. Bu algoritmanın denklem sistemini dağıtık olarak çözebilmesi için gerek ve yeter koşul, ağın tekrarlı birlikte kuvvetli bağlı çizge dizisine sahip olması olarak verilmiştir [4].

Bununla beraber, sürekli zamanlı doğrusal denklem sistemlerinin dağıtık olarak çözümünde kullanılan bir diğer algoritma [8]'de,

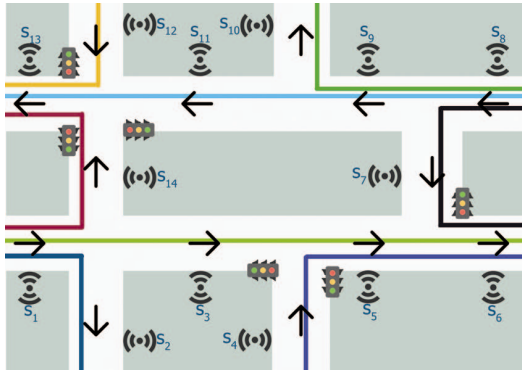
$$\dot{x}_i(t) = -P_i \sum_{j \in N_i(t)} (x_i(t) - x_j(t)) - A_i^T (A_i A_i^T)^{-1} (A_i x_i - b_i) \quad (4)$$

olarak önerilmiştir. Bu algoritmanın [4]'te önerilenden farkı, başlangıç değeri olarak verilen $x_i(0)$ 'ın tamamen keyfi bir şekilde seçilebiliyor olmasıdır. Bu sayede, denklem sistemini çözmeye başlarken seçilen $x_i(0)$ vektörlerinin, $A_i x_i(0) = b_i$ denklemini sağlanması gerekmemektedir.

3. Trafik Ağı Problemi ve Formülasyonu

Günümüzde özellikle büyük şehirlerdeki trafik yoğunluğu problemi, hem sosyal hem de ekonomik olarak ciddi bir sorun olarak karşımıza çıkmaktadır. Gün içerisinde özellikle belirli vakitlerde yaşanan yoğunluklar, hem kişilerin vaktinin yolda boşa geçmesine ve böylece aktif olabilecekleri sürenin kısıtlanmasına hem de araçların dur kalk yaparak harcadıkları yakıttan dolayı enerji kaynağının boşa harcanmasına sebep olabilmektedir. Bunun için trafik yoğunluğunu azaltacak akıllı trafik ağlarının kurulması büyük önem arz etmektedir.

Bu çalışmada önerdiğimiz trafik ağı modelinde, şehirden küçük bir trafik ağı kesiti için mümkün olan tüm güzergahlardan birim zamanda geçen araç sayılarının, sadece o yoldan birim zamanda geçen toplam araç sayısını ölçen sensörlerin (etmenler) veri iletişimleri aracılığıyla dağıtık olarak belirlenmesi amaçlanmaktadır. Bu problemdeki motivasyon, tüm güzergahlardaki araç sayılarının belirlenerek yol üzerindeki trafik ışıklarının yanma sürelerinin rotalardaki araç yoğunluğuna göre ayarlanması ve yoğun rotalardaki bekleme sürelerinin mümkün mertebe azaltılmasıdır.



Şekil 1: Bir şehirdeki trafik ağından bir kesit ve mümkün olan güzergahlar

Şekil 1'de bir şehirdeki trafik ağından bir kesit gösterilmiştir. Caddelerin kenarlarına yerleştirilen 14 sensör (s_1, \dots, s_{14}) ile o noktadan geçen araç sayıları ölçülebilmektedir. Bu sensörler sadece önünden geçen araçların sayısını ölçebilir ve kablosuz veri iletişimi kabiliyetine sahip basit sensörlerdir. Şekil 1'de bu ağdaki mümkün olan tüm güzergahlar farklı renklerle işaretlenmiş olup, belirtilen yönlerin aksine yönlerde araç geçme-

diği ve tüm trafik kurallarına uyulduğu varsayımıyla, her bir güzergahı takip eden araçların önünden geçtiği sensörlerin listesi Tablo 1'de listelenmiştir.

Tablo 1: Her bir güzergahı kullanan araçların önünden geçtiği sensörler

Güzergah rengi	Güzergah adı	Önünden geçtiği sensörler
Koyu Mavi	g_1	$s_1 - s_2$
Açık Yeşil	g_2	$s_1 - s_3 - s_5 - s_6$
Kırmızı	g_3	$s_1 - s_{14} - s_{13}$
Mor	g_4	$s_4 - s_5 - s_6$
Siyah	g_5	$s_8 - s_7 - s_6$
Turkuaz	g_6	$s_8 - s_9 - s_{11} - s_{13}$
Koyu Yeşil	g_7	$s_8 - s_9 - s_{10}$
Turuncu	g_8	$s_{12} - s_{13}$

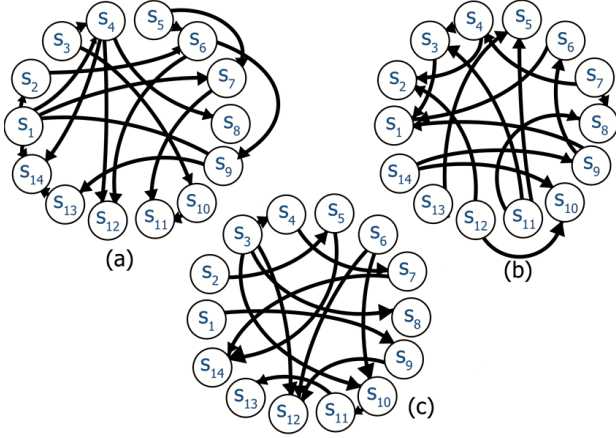
Sensörlerin önünden geçen araç sayısı güzergahları o sensörlerin önünden geçen tüm araçların sayısı olacağı için, her bir sensörün ölçümü bir doğrusal denkleme karşılık gelmektedir. Örneğin s_1 sensörünün önünden geçen araçlar g_1, g_2 ve g_3 güzergahlarını kullanıyor olacağı için, bu durum $s_1 = g_1 + g_2 + g_3$ olarak ifade edilebilmektedir. Tüm sensörler için benzer şekilde denklemler yazılarak, tüm trafik ağı için sensörün ölçtüğü araç sayısı değerleri ile güzergahları kullanan araç sayıları arasındaki ilişki, $x = [g_1, \dots, g_8]^T$ ve $b = [s_1, \dots, s_{14}]^T$ olmak üzere $Ax = b$ formunda aşağıdaki gibi ifade edilebilir.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \end{bmatrix} \quad (5)$$

Her bir sensörün, (5) denkleminin yalnızca kendine ait satırını biliyor olması ve komşu sensörler ile kablosuz iletişim içerisinde olması nedeniyle önerilen bu problem Bölüm 2.2'de verilen algoritma ile dağıtık olarak çözülebilecektir.

4. Benzetim Çalışmaları

Şekil 1'de verilen trafik ağında, güzergahlardan geçen araçların belirlenmesi problemini ele alalım. Sensörler birim zamanda önlerinden geçen araç sayısını ölçebilmektedir. Trafik ağı problemini temsil eden doğrusal denklem sisteminin zamanla değişen olmaması için, bu çalışmada birim zamanda bir güzergahı kullanan araç sayısının değişmediği varsayılacaktır. Buna karşın, sensörler arası iletişimin sabit olmadığı ve ağ ilingisinin 0.1 saniye aralıklarla Şekil 2'de verilen ilingeler arasında sürekli değiştiği durum ele alınacaktır.



Şekil 2: Bileşimi tekrarlı birlikte kuvvetli bağlı ilingeler

Bu durumda, ağı temsil eden çizge dizisi tekrarlı birlikte kuvvetli bağlı olmaktadır ve (2) algoritması ile denklem sisteminin dağıtık olarak çözümü için gerekli koşullar sağlanmıştır. Şekil 1’de verilen trafik ağı için her bir güzergahtan birim zamanda geçen araç sayısı Tablo 2’de verilmiştir.

Tablo 2: Rotalar ve geçen araç sayıları

Rota kodu	Geçen araç sayısı
g_1	3
g_2	2
g_3	10
g_4	10
g_5	9
g_6	8
g_7	2
g_8	4

Bu durumda, sensörlerin birim zamanda ölçtükları araç sayıları vektörü

$$b = [15, 3, 2, 10, 12, 20, 8, 14, 6, 6, 4, 9, 23, 10]^T$$

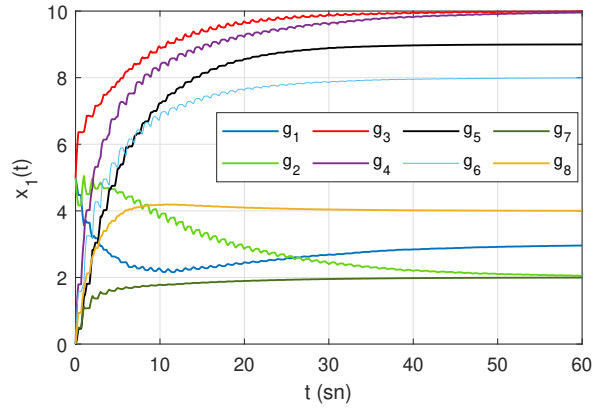
olacaktır. Trafik ağı probleminin dağıtık olarak çözülmesi, $Ax = b$ denklem sisteminin eşsiz çözümü olan $x = [3, 2, 10, 10, 9, 8, 2, 4]^T$ ’nin her bir sensör tarafından belirlenebilmesi olarak tanımlanmaktadır. Bölüm 2.2’de bahsedildiği üzere, algoritmanın çalışabilmesi için her bir etmenin kendi denklemini sağlayan bir çözüm vektörü oluşturması gerekmektedir. Bu çözümlerin sistematik bir şekilde oluşturulabilmesi amacıyla,

$$x_i(0) = A_i^T (A_i A_i^T)^{-1} b_i \quad (6)$$

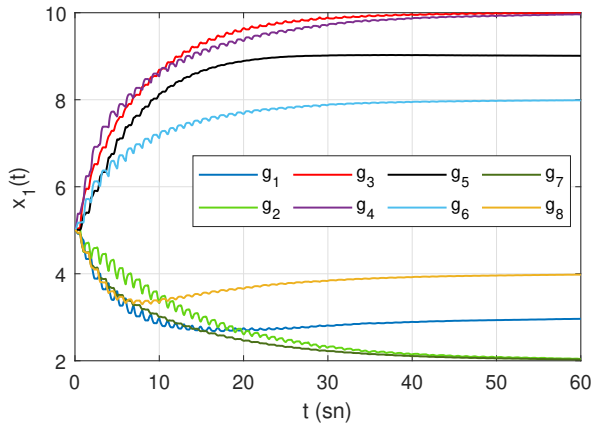
ile ifade edilen en küçük norm çözümleri kullanılabilir. Benzetim çalışması yapılacak olan ağı için etmenlerin başlangıç çözümleri (en küçük norm çözümleri) Tablo 3’te verilmiştir. Bu çözümlerden başlayarak (2) algoritması kullanıldığında 1. etmenin çözümünün zamanla değişimi Şekil 3’teki gibi olmaktadır.

Tablo 3: Sensörlerin başlangıç çözümleri (en küçük norm çözümleri)

Sensör	En küçük norm çözümü
s_1	$[5, 5, 5, 0, 0, 0, 0, 0]^T$
s_2	$[3, 0, 0, 0, 0, 0, 0, 0]^T$
s_3	$[0, 2, 0, 0, 0, 0, 0, 0]^T$
s_4	$[0, 0, 0, 10, 0, 0, 0, 0]^T$
s_5	$[0, 6, 0, 6, 0, 0, 0, 0]^T$
s_6	$[0, 20/3, 0, 20/3, 0, 20/3, 0, 0]^T$
s_7	$[0, 0, 0, 0, 0, 8, 0, 0]^T$
s_8	$[0, 0, 0, 0, 0, 14/3, 14/3, 14/3]^T$
s_9	$[0, 0, 0, 0, 0, 0, 3, 3]^T$
s_{10}	$[0, 0, 0, 0, 0, 0, 3, 3]^T$
s_{11}	$[0, 0, 0, 0, 0, 0, 0, 4]^T$
s_{12}	$[0, 0, 0, 0, 9, 0, 0, 0]^T$
s_{13}	$[0, 0, 23/3, 0, 23/3, 0, 0, 23/3]^T$
s_{14}	$[0, 0, 10, 0, 0, 0, 0, 0]^T$



Şekil 3: (2)’de verilen dağıtık algoritma kullanıldığında 1. etmenin çözümünün zamana bağlı değişimi



Şekil 4: (4)’te verilen dağıtık algoritma kullanıldığında 1. etmenin çözümünün zamana bağlı değişimi

Giriş kısmında da bahsedildiği üzere, [8]'de önerilen algoritma için başlangıç vektörü keyfi olarak seçilebilmektedir. Bu örnek için kullanılan başlangıç vektörü, tüm etmenler için

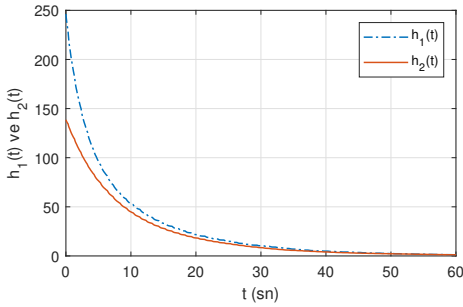
$$x_i(0) = [5, 5, 5, 5, 5, 5, 5, 5]^T$$

olarak verilmiştir. [8]'de önerilen sürekli-zamanlı dağıtık algoritma ile elde edilen sonuçlar Şekil 4'de verilmiştir.

Tüm etmenlerin trafik ağı problemini çözdüğünü göstermek amacıyla, çözüm hatalarının normlarının toplamı olarak tanımlanan

$$h(t) = \sum_{i=1}^m \|x^* - x_i(t)\| \quad (7)$$

fonksiyonunun algoritma (2) ve (4) kullanıldığında zamanla değişimi Şekil 5'te verilmiştir. Bu değer in sıfıra gitmesi, tüm hataların sıfırlandığını ve bütün sensörlerin ağıdaki tüm güzergahlardan birim zamanda geçen araç sayılarını belirleyebildiklerini göstermektedir.



Şekil 5: Hataların zamana bağlı değişimi ($h_1(t)$: Algoritma (2) kullanıldığında, $h_2(t)$: Algoritma (4) kullanıldığında)

Sensörlerin ağıdaki tüm olası güzergahlardan birim zamanda geçen araç sayısını çözebilmesi, çeşitli kavşak noktalarındaki trafik ışıklarında bekleyen araçların ne kadarının hangi güzergahı takip etmekte olduğunun belirlenebilmesini sağlamaktadır. Bu bilgi kullanılarak, trafik ışıklarının yanma süresi ayarlanması ve yoğun araç geçişinin olduğu güzergahlardaki trafiğin azaltılması mümkün olabilir.

5. Tartışma ve Sonuçlar

Bu çalışmada çok etmenli sistemlerde, doğrusal denklem sistemlerinin dağıtık olarak çözümünde kullanılan ve sürekli zamanda tanımlı iki farklı algoritma, bir trafik ağı probleminin çözümünün hesaplanmasında uygulanmıştır. Bu iki farklı algoritmanın da, ağ ilingesinin zamanla değişen ve tekrarlı birlikte kuvvetli bağlı olması durumunda yalnızca önünden geçen araç sayısını ölçen sensörlerin komşularıyla bilgi paylaşımında bulunarak, trafikteki tüm güzergahlardan geçen araç sayılarını çözebildikleri gösterilmiştir. Her güzergahtan geçen araç sayısının tüm sensörler tarafından bulunması sayesinde, kavşaklarda bulunan trafik ışıklarının süreleri uygun şekilde ayarlanabilir ve trafik sıklığı azaltılabilir.

Trafik akışının çift yönlü (gidiş-dönüş) olduğu bazı tasarımlarda, sensörlerin belirledikleri araç sayıları kullanılarak her bir güzergahı kullanan araç sayısının belirlenmesi mümkün olmayabilmektedir. Bunun nedeni, bu tasarımda $Ax = b$ denklem

sisteminin sonsuz çözüme sahip olmasıdır (A matrisinin tam sütun kerteli olmadığı durum). Gelecek çalışmalarda, denklem sisteminin eşsiz çözüme sahip olabilmesi için trafik ağı üzerinde gerekli koşulların belirlenmesi ve güzergahtan geçen araç sayılarının daha hızlı belirlenmesini sağlayan algoritma iyileştirmeleri yapılması planlanmaktadır.

6. Teşekkür

Bu çalışma TÜBİTAK'ın 117E204 numaralı projesi tarafından desteklenmiştir.

7. Kaynakça

- [1] S. Mou , J. Liu ve A. S. Morse, "A Distributed Algorithm for Solving a Linear Algebraic Equation." *IEEE Trans. Auto Control* , vol. 60, no. 11, pp. 2863-2878, Nov. 2015.
- [2] S. Mou ve A. S. Morse, "A Fixed-Neighbor, Distributed Algorithm for Solving a Linear Algebraic Equation." *European Control Conference*, pp. 22669-2273, Jul. 2013.
- [3] Gao, X. , Liu, J. ve Başar, T. Stochastic communication-efficient distributed algorithms for solving linear algebraic equations. *IEEE conference on control applications* pp. 380-385, 2016.
- [4] J. Liu, X. Chen, T. Başar, ve A. Nedic, "A Continuous-Time Distributed Algorithm for Solving Linear Equations." *American Control Conference*, pp. 5551-5556 Boston, USA, 2016.
- [5] X. Wang ve S. Mou, "Improvement of A Distributed Algorithm for Solving a Linear Algebraic Equation." *IEEE Trans. on Industrial Electronics*, vol. 64, no. 4 pp. 3113-3117, April 2017.
- [6] P. Wang, S. Mou and J. Lian et al., "Solving a system of linear equations: From centralized to distributed algorithms" , *Annual Reviews in Control*, May 2019
- [7] X. Wang, , J. Zhou, S. Mou, ve M.J. Corless. "A distributed linear equation solver for least square solutions." *IEEE 56th annual conference on decision and control*, pp. 5955-5960, 2017.
- [8] Anderson, B. D. O. , Mou, S. , Morse, A. S. ve Helmke, U. "Decentralized gradient algorithm for solution of a linear equation." *Numerical Algebra, Control ve Optimization*, 6 , p. 319 .(2016)
- [9] M. Yang ve C. Y. Tang. A distributed algorithm for solving general linear equations over networks, *In Proceedings of the 2015 IEEE Conference on Decision and Control*, pp. 3580-3585, 2015.
- [10] G. Shi ve B. D. O. Anderson. Distributed network flows solving linear algebraic equations, *In Proceedings of the 2016 American Control Conference*, pp. 2864-2869, 2016.
- [11] G. Shi, B.D. O. Anderson ve U. Helmke. Network flows that solve linear equations, *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659-2674, 2017.
- [12] X. Zeng ve K. Cao. Computation of linear algebraic equations with solvability verification over multi-agent networks, *Kybernetika*, vol. 53, no. 5, pp. 803-819, 2017.