TRAFİK IŞIK OPTİMİZASYON SİSTEMLERİNİN KARŞILAŞTIRILMASI

2016
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Alper Talha KARADENİZ

TRAFİK IŞIK OPTİMİZASYON SİSTEMLERİNİN KARŞILAŞTIRILMASI

Alper Talha KARADENİZ

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

KARABÜK
Ağustos 2016

Alper Talha KARADENİZ tarafından hazırlanan "TRAFİK IŞIK OPTİMİZASYON SİSTEMLERİNİN KARŞILAŞTIRILMASI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Yüksel ÇELİK .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 06/09/2016

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Doç. Dr. Oğuz FINDIK (KBÜ)	.....
Üye : Yrd.Doç. Dr. Yüksel ÇELİK (KBÜ)	.....
Üye : Yrd. Doç. Dr. Abdulkadir KARACI (KÜ)	.....

...../...../2016

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nevin AYTEMİZ .....
Fen Bilimleri Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Alper Talha KARADENİZ

ÖZET

Yüksek Lisans Tezi

TRAFİK IŞIK OPTİMİZASYON SİSTEMLERİNİN KARŞILAŞTIRILMASI

Alper Talha KARADENİZ

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Yrd. Doç. Dr. Yüksel ÇELİK
Ağustos 2016

Trafik sistemi çok sayıda akıllı bileşenin (sinyaller, araçlar, sensorlar ve yayalar)yerel düzeyde birbirleriyle iletişimi olan üst düzeylerde ortak davranışlar sergileyen, karmaşık bir sistemdir. Şehirlerde trafik kavşaklarında yetersiz trafik ışığı kontrol sistemlerinin doğal sonucu olarak gereksiz gecikmeler ve zaman kayıpları, ışıklarda rölantide çalışan motorun fazla yakıt yakması ve atmosfere salınan sera gazı emisyonlarının artmasına sebep olmaktadır. Yukarıda bahsedilen trafik problemlerini gidermek için birçok sistem geliştirilmiş ve geliştirilmeye devam etmektedir. Trafik optimizasyon sistemleri için geliştirilen yöntemlerden başlıcaları, önceden süreleri belirlenen sabit zamanlı ışık sistemleri, yeşil dalga ışık sistemleri ve gerçek zamanlı trafik ışık optimizasyon sistemleridir. Bizim yaptığımız çalışmada yukarıda bahsedilen bu sistemlerin, Karabük – Safranbolu güzergahında, farklı yoğunluklarda, gerçek veriler üzerinde testler yapılmıştır. Yapılan bu testler sonucunda gerçek

zamanlı trafik ışık optimizasyon sistemlerinin sabit zamanlı ve yeşil dalga ışık sistemlerinden daha iyi sonuçlar verdiği görülmektedir.

Anahtar Sözcükler : Trafik optimizasyonu, yeşil dalga trafik, trafik sinyalizasyon, sabit zamanlı trafik, şehir trafik simülasyonu(SUMO).

Bilim Kodu : 902.1.013

ABSTRACT

MASTER THESIS

COMPARISON OF TRAFFIC LİGTH SYSTEM OPTIMIZATION

Alper Talha KARADENİZ

Karabuk University
Institute of Science and Technology
The Department of Computer Engineering

Thesis Supervisor:
Yrd. Doç. Dr. Yüksel ÇELIK
August 2016

Traffic system is a complex system where a lot of smart components which include signals, vehicles, sensor and pedestrian have communication skills with together on local level and act in a particular manner on high level. Insufficient traffic light control system on intersections brings about unnecessary delays and waste of time, extremely oil firing of engine which run idle mode on lights and increasing greenhouse gas emission. Various systems have been developed in order to overcome these traffic problems. These are the primary developed methods for the traffic optimization systems: fixed time period systems of lighting where time is pre-determined, greenwave lighting system and real time optimization system of traffic light.Real data has been gathered on Karabuk-Safranbolu route to test above systems for different data density. The results of these tests on data show that real time traffic light optimization systems get better results than fixed time period and greenwave lighting systems.

Key Words : Traffic optimization, greenwave traffic, signalization, fixed time period system of lighting, Simulation of urban mobility (SUMO).

Science Code : 902.1.013

TEŞEKKÜR

Yüksek lisans tez çalışma konusunun planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren danışman hocam sayınYrd.Doç. Dr. Yüksel ÇELİK'e sonsuz şükranlarımı sunarım.

Tez çalışmalarım boyunca her fırsatta yardımlarını gördüğüm sayın hocam Yrd. Doç. Dr. Tuğba TUNACAN'a ayrıca teşekkür ediyorum.

Tez çalışmasında yapılması gerekenler ile ilgili bilgilerini benimle paylaşan sevgili babam Prof. Dr. Turan KARADENİZ'e, maddi ve manevi desteklerini benden esirgemeyen sevgili annem Zekiye KARADENİZ'e, meslektaşım olmak üzere olan kardeşim Muhammed Berker KARADENİZ'e ve sevgili eşim Esra KARADENİZ'e tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

SUMO : Simulation of Urban MObility (Şehir Trafik Simülasyon Programı)

SCOOT : Split Cycle Offset Optimization Technique (Boşluklu Döngüsel Mesafe İyileştirme Yöntemi)

SCATS : Sydney Coordinated Adaptive Traffic System (Sydney Koordine Uyarımlı Trafik Sistemi)

OPAC : Optimized Policies for Adaptive Control (Uyarlamalı Kontrol için İyileştirme Politikası)

RHODES : Real Time Hierarchical Optimizing Distributed Effective System (Dağıtılmış Etkili Sistemde Gerçek Zamanlı Hiyerarşik İyileştirme Yöntemi)

ACS : Adaptive Control System (Uyarlamalı Kontrol Sistemi)

FHWA : Federal Highway Administiration (Federal Karayolları İdaresi)

TRACI : Traffic Control Interface (Trafik Kontrol Arayüzü)

CA : Celler Automata (Hücresel Otomasyon)

BÖLÜM 1

GİRİŞ

Günümüzde şehirleşme hızla artmakta ve bu artış birçok avantaj ile birlikte birçok dezavantajı da beraberinde getirmektedir. Şehirleşmedeki en büyük dezavantajlardan birisi hiç şüphe yok ki trafik problemidir. Şehirlerdeki artan nüfus ile birlikte, toplu taşıma araçlarının yetersizliği veya cazip hale getirilememesi, toplu taşıma kültürünün vatandaşa aşılanamaması, gelir düzeyindeki artış, araç sahibi olmanın kolaylıkları gibi birçok faktör ile birlikte ülkemiz ve dünyada, trafikteki araç sayısı her geçen gün artmaktadır.

Türkiye İstatistik Kurumu raporlarına göre, trafiğe kayıtlı araç sayısı 2016 Aralık ayı sonunda 19.994.472, 2016 Ocak ayı sonunda ise 20.098.994 olarak hesaplanmıştır [1]. Bu raporlar bize her ay Türkiye'de 104 522 aracın trafiğe dahil olduğunu göstermektedir. Artan araç sayısı ile birlikte, şehirlerdeki yetersiz trafik ışığı kontrolünün doğal sonucu olarak, gecikmelerden oluşan zaman kayıpları, rölantide çalışan motorun fazla yakıt tüketmesi ile atmosfere saldığı sera gazı emisyonunun artması ve kavşaklarda meydana gelen kaza oranlarının artmasına neden olmaktadır.

Araç takip sistemleri üzerine faaliyet gösteren uluslararası bir firmanın 2015 Şubat ayında hazırladığı raporlara göre, İstanbul'da bir kişi günde en az 24 dakikasını trafikte herhangi bir yol kat etmeden geçirdiği görülmüştür. Araçların bekleme süresinde harcadığı günlük yakıtın ekonomik karşılığı ise 2 milyon 271 bin 103 TL olarak hesaplanmıştır [2]. Ayrıca Türkiye İstatistik Kurumu Karayolları Raporlarına göre, 2014 yılında Türkiye'deki kaza sayısı 1.199.010 olarak hesaplanmıştır [1]. Yapılan araştırmalarda trafik kazalarının %50'ye yakınının kavşaklarda meydana geldiği görülmektedir [3]. Yukarıda sözü edilen günümüz şehirlerinde yaşanan problemler ve yine atmosfere salınan sera gazı emisyonunun oluşturduğu olumsuz

gelişmeler göz önüne alındığında, trafikte yaşanan bu sorunlara acilen çözüm üretilmesi ve konunun önemi daha iyi şekilde anlaşılmaktadır.

Trafik çok sayıda akıllı bileşenin (trafik ışıkları, sensorlar ve yayalar) yerel düzeyde birbirleriyle iletişimi olan, üst düzeyde ortak davranışlar sergileyen, karmaşık bir sistemdir.

İlk 3 renkli trafik lambası 1920 yılında kullanılmaya başlamıştır[4]. Bu yıldan itibaren trafik ışık sistemleri konusunda, trafiğin verimliliğini arttırmak ve bahsedilen trafik sorunlarını gidermek için birçok çalışma yapılmış ve bu çalışmalar sürdürülmektedir.

1.1. TRAFİK IŞIK OPTİMİZASYON SİSTEMLERİ

Trafik ışık optimizasyon sistemleri genel olarak online ve offline olmak üzere 2'ye ayrılmıştır [5]. Offline trafik optimizasyon sistemleri, önceden belirlenen kurallara göre çalışmaktadır. Online trafik sistemleri ise kendi kendine organize olan, gerçek zamanlı olarak çalışan, anlık olarak en iyi sonucu üretmeye programlı sistemlerdir.

Offline trafik ışık sistemlerinden en yaygın olarak kullanılanlarının başında sabit zamanlı trafik ışık sistemleri gelmektedir. Sabit zamanlı kontrol tekniklerinde sinyal çevrim süresi ve yeşil ışık sürelerinin kavşağa yaklaşan akımların doygunluğuna göre önceden belirlendiği sabit sinyal planları kullanılmıştır [6]. Bu sistem içerisindeki bütün kavşaklar merkezde bulunan bir bilgisayar tarafından yönetilmektedir ve hesaplanan yeşil ışık süreleri tüm kavşaklara merkez tarafından iletilir. Sabit zamanlı sinyalizasyon sistemleri özellikle trafik hacimlerinin gün içinde değişken olmadığı durumlarda düzgün ve verimli şekilde kullanılabilmektedir. Bu çalışmaların geliştirilmesi ile trafiğin yoğun olduğu saatlerde kavşaktaki trafik sıkışıklığının giderilmesi için farklı zaman planları uygulanmaya başlanmıştır. Yürütülen çalışmalarda sabit zamanlı trafik sinyal sistemleri araştırılmış ve trafikte geniş bir uygulama alanı bulmuştur. Nitekim, Robetson (1983) TRANSTY olarak adlandırdığı çalışmalarında sabit zaman sinyalini uygulamış ve isteğe bağlı olarak günün farklı saatlerinde farklı sinyal planlarının uygulanabileceğini göstermiştir [7].

Trafiğin doğal karmaşıklığı göz önüne alındığında, taşıtlar, şehir ve trafik ışığı gibi parametreler sürekli olarak değişmektedir. Yani trafiğin canlı, yaşayan bir sistem olduğu açıkça görülmektedir. Yapılan çalışmalar ve testler sonucunda yaşayan ve karmaşık olan trafik probleminin durağan olan sabit zamanlı ışık sistemleriyle çözülemeyeceği artık anlaşılmaktadır.

Offline sistemlerden bir diğeri de yeşil dalga trafik sistemidir. Yeşil dalga trafik sistemi bir trafik rotası boyunca, tek düze trafik ışıkları uygulamasıdır. Kontrol mesafesindeki bütün trafik ışıkları merkezdeki bilgisayar sistemine kaydedilmektedir. Rota boyunca bütün kavşaklardaki trafik ışığının rengi merkez tarafından görülmekte, yeşil ışık ise yuvarlanarak ilerlemektedir. Bu geçişler bir dalgaya benzetilmiştir. Bu yüzden adına yeşil dalga trafik sistemi denilmiştir [3].

Yeşil dalganın amacı, yeşil ışıkta geçen araçların büyük bir çoğunluğunun bir sonraki kavşaktaki trafik ışığında yine yeşil ışık ile karşılaşarak, trafikteki sıkışıklığın ve buna bağlı olarak oluşan fazla yakıt tüketimini, ayın zamanda sera gazı emisyon salınımını en aza indirgemektir [8].

Yeşil dalga uygulamalarının verimli olarak çalışılabilmesi birçok parametrenin birlikte değerlendirilmesi ve uygulanması ile mümkündür. Göz ardı edilebilecek kimi parametrelerin eksikliğinde, problemler azalacakken artış söz konusu olabilmektedir. Yeşil dalganın verimli kullanılması için gereken etkenler şu şekilde sıralanmıştır.

1. Araçlar sabit ve istenilen hızda gitmelidir.
2. Trafik organizasyonu basit formda olmalıdır. Örneğin, tek yön trafik, 4 yönlü kavşak gibi.
3. Kavşaklar arasındaki mesafe iyi ayarlanmalıdır. Sabit hızda giden araçlar bir sonraki kavşağa ulaştığında tekrar yeşil ışık ile karşılaşabilmelidirler.
4. Sinyal fazı yeşil dalga trafik uygulamalarında önemli bir etkiye sahiptir. Sinyal fazının az olması yeşil dalganın bant genişliğini arttırmaktadır.
5. Bu faktörlerin tümü iyi bir şekilde uygulanırsa yeşil dalganın verimli bir şekilde kullanılabileceği görülmüştür. Aksi takdirde araçların büyük bir

çoğunluğu kırmızı ışığa yakalanacaklardır. Bu durum şüphesiz trafik sıkışıklığındaki sorunların artmasına sebep olacaktır [3].

Gelişen teknoloji ile birlikte, yüksek hızlı işlemciler, sensörlar, görüntü işleme teknikleri, trafik sinyalizasyonunun geliştirilebilmesi için olanak sağlamıştır. Trafik durağan değildir. Canlı, yaşayan ve parametreleri sürekli değişen bir sistemdir. Teknolojinin gelişmesi ile birlikte durağan olmayan trafik sistemine sabit yöntemlerle çözüm aranamayacağı anlaşılmış, dolayısıyla dinamik sistemleri devreye sokan projeler üretilmiştir. Online sistemler gerçek zamanlı olarak çalışan, trafiğin anlık durumuna cevap oluşturan sistemler olarak tanımlanmaktadır.

Gerçek zamanlı trafik modelinin kökeni 1958 yılında yürütülen Webster çalışmalarına dayandığı görülmüştür [9]. Bu çalışmalardan yola çıkarak (Miller, 1963) ve (Little, 1966) uygun parametreler geliştirmişler ve bu parametreler daha sonraki yürütülen çalışmalarda referans olarak değerlendirilmiştir.
Gerçek zamanlı sistemlerde rotadaki tüm kavşakların belirli bir merkez üzerinden iletişim kurmaları yerine, her bir kavşak komşusu olan kavşakla iletişim kurmakta ve bu yerel iletişim rota boyunca ilerleyerek genel bir senkronizasyon sağlamış olmaktadır. Merkezden yönetilmeyi ortadan kaldırdığımızda kazanılan zaman bile trafik için oldukça önem arz etmektedir.

Dünya üzerinde kabul görmüş ve uygulamalarda temel alınmış gerçek zamanlı trafik ışık optimizasyon sistemleri mevcuttur. Hunt ve arkadaşları çok yoğun olmayan şehir trafik sistemleri için trafik taleplerine uyarlamalı bir sistem geliştirmiş ve SCOOT adını vermişlerdir [10]. Yapılan ilk SCOOT (Split Cycle Offset Optimisation Technique) çalışması 1981 yılında İngiltere'de Ulaşım Araştırma Laboratuvarı tarafından geliştirilmiştir. 1970'li yılların sonunda Ulaşım Araştırma Laboratuvarı sabit zamanlı trafik sistemlerinin sorunlarını ortadan kaldırmak için bir yöntem geliştirmiştir. Yöntemde rota üzerindeki tüm trafik ağı bilgisayar sistemi tarafından izlenmiştir. Gecikmeleri azaltmak ve trafik ağını iyileştirmek için sinyal zaman planları sık sık değiştirilmiş ve bu yöntem SCOOT'a temel oluşturmuştur [7]. SCOOT yönteminin sabit zamanlı trafik sistemleri ile kıyaslandığında daha iyi sonuçlar verdiği görülmüştür. Yine, sabit zamanlı trafik sistemleri ile SCOOT trafik

sistemi karşılaştırıldığında araç gecikmelerinin Worcester'da %23, Southampton'da %30 ve Glasgow' da %27 azaldığı tespit edilmiştir [11]. Diğer yandan, 1993 yılında Toronto'da örnek bir SCOOT projesi gerçekleştirilmiş ve TRANSTY trafik sisteminden daha iyi sonuç verdiği belirlenmiştir. Kanada'da yapılan çalışmalarda ise %14'lük bir iyileşme meydana geldiği saptanmıştır [12].

SCOOT yöntemi trafikten istenilen verileri detektörler aracılığı ile almış ve sistem iyileştirmelerini bu veriler ışığında yapmıştır. SCOOT optimizasyon sistemi 3 prosedürden oluşmuştur. Bunlar, bölünmüş iyileştirme, uzaklık iyileştirme ve çevrim süresi olarak ifade edilmiştir. Algoritma tahminleri, araç gecikmeleri, her bir yön üzerindeki durmalar ve sistem performansı sürekli olarak hesaplanmış, ağın koordinasyonu sürekli olarak sağlanmıştır. Bölünmüş iyileştirme prosedürü 1 ila 4 saniyelik aralıklarla çalışmış, aşama değişim zamanında bir değişiklik yapılıp yapılmayacağı konusunda, kırmızı ışık ve yeşil ışık zamanları analiz edilerek hesaplanmıştır. Ofset iyileştirme prosedürü her kavşak için döngü başına 1 kez çalışmış, kavşaktaki her bir yön analiz edilerek değerlendirilmiş ve mevcut zaman için değişiklik yapılıp yapılmamasına karar verilmiştir. Çevrim süresi iyileştirme prosedüründe ise her bölge için 5 dakikada 1 kez çalışmıştır. Çevrim süresi prosedürü bölgedeki kritik düğümü belirlemiş ve her aşamasında bağlantı doygunluğunu %90'da tutmayı planlamıştır. Bu döngü süresinde bir değişiklik yapılması gerekiyorsa 4, 8 ve 16 saniyelik değişimler ile yapılmıştır [13].

SCOOT'ın temel felsefesi aynı kalmak ile beraber, birçok yenilikler eklenerek farklı sürümleri geliştirilmiştir. Bu gelişmeler kullanıcılardan gelen istekler üzerine sağlanmıştır. SCOOT 2.3. sürümde özel bağlantılara öncelik vermek için ağırlıklar tanımlanmıştır. SCOOT 2.4. sürümde 1990 yılının mart ayında kullanıcılara tanıtılmıştır [14]. SCOOT 2.5. sürüm Haziran 1994' de C programlama dili ile 120 saniyeden yüksek döngü zamanları sisteme eklenmiştir [15].

Dünya üzerinde 200'den fazla uygulaması olan SCOOT gerçek zamanlı, otoriteler tarafından kabul görmüş bir trafik sinyali sistemi olarak adlandırılmıştır [16].

Çalışmalardan bir diğeri de SCATS'dir. Bu yöntem TRANSTY offline metodunu temel almış ve üzerinde geliştirilmiştir. Bu sistem yol ağının verimini arttırmak için, araç duruşları ve gecikme miktarını minimize etmeyi amaçlamıştır [17]. SCATS kavşak grupları üzerinde hiyerarşik bir yapı oluşturmuş ve benzer sinyal zamanları kullanmıştır [18]. SCATS (Sydney Coordinated Adaptive Traffic System) 1970 yılında Avusturalya'da yollar ve trafik otoritesi tarafından geliştirilmiş ve ticari olarak temin edilebilen bir sistem olarak tanımlanmıştır. SCATS trafik akışı ve sistem kapasitesi içinde, trafik dalgalanmalarına tepki olarak, sinyal zamanlamasını ayarlayabilmiştir. SCATS, merkezi, bölgesel ve yerel olmak üzere üç kontrol seviyesi ile tasarlanmıştır. Bütün sistemler merkezden belli bir hiyerarşi içerisinde izlenmiştir. Bu sistem, geleneksel trafik kontrol sistemleri ile gerçek zamanlı trafik ışık kontrol sistemlerini birleştirmiştir. Gerçek zamanlı raporlama araçları sayesinde, sistemin izlenmesine olanak sağlamıştır. Kavşaklar izlenerek, olağandışı bir durum gerçekleştiğinde ya da kavşaklardaki donanım bir arıza ile karşılaştığında sistem mühendislerine haber verilmiştir [19].

Kabul görmüş ve referans alınmış bir başka çalışmada OPAC'tır. OPAC (Optimized Policies for Adaptive Control) 1980'li yılların başında Ulaştırma Bakanlığı sponsorluğunda Lowell' de Massachusetts Üniversitesin'de geliştirilmiştir. Kavşaklardaki gecikmeleri ve durmaları en aza indirmek için, sinyal zamanlarını hesaplamış, dinamik optimizasyon algoritması ile özellikleri dağıtılmış bir kontrol stratejisi olarak adlandırılmıştır. OPAC sinyal kontrol algoritması sabit zaman aralıklarının olduğu kararları değiştirmiştir. OPAC bir kararın uzatılması veya sonlandırılması ile ilgili karar verebilen bir sistem olarak tasarlanmıştır [20].

Bir başka sistem olan RHODES ise (Real time Hierarchical Optimizing Distributed Effective System), 1990 yılında Arizona Üniversitesi tarafından projelendirilmiştir. Hiyerarşik bir yapı ile gerçek zamanlı trafik uyarlamalı sistem olarak RHODES farklı türde girdi almak ve gelecekteki trafik koşullarını tahminde dayalı, optimize edilmiş sinyal kontrol planları oluşturmuştur. RHODES üç farklı sistem özelliğine sahip bir sistem olarak geliştirilmiştir.

1. Yeni teknolojiler ve metotlar, trafik akışı, trafik verileri ve sinyal kontrol tahminleri, performansın artması için kullanılmıştır.
2. Trafik akışının doğal değişimlerini göz önünde bulundurmuş ve her bir farklı talep bağlantısı ile her bir aşama için yeşil yanma süresini tahsis etmiştir.
3. Tek gelen araç, takım halinde gelen araçlar ve trafik akış oranlarının tahmini yapılmıştır[21].

Diğer bir sistem ise ACS (Adaptive Control System) olup, 1990'ların ortalarında Siemens, Arizona Üniversitesi ve Purdue Üniversitesi tarafından, Federal Karayolları İdaresi (FHWA) sponsorluğunda geliştirilmiştir [22]. Sistem küçük ve orta ölçekli topluluklarda, gerçek zamanlı olarak çalışan, düşük maliyetli trafik kontrol sistemi olarak tanımlanmıştır. ACS mevcut sistemleri güçlendirmiş ya da yeni sinyaller üreterek, kendi sistemini oluşturmuştur. Bu sistem Gahanna, Ohio, Houston, Teksas, Florida ve birçok daha yerde kullanılmıştır [23]. Dünya üzerinde kabul görmüş ve birçok çalışmada referans olarak alınmış gerçek zamanlı trafik ışık sistemlerinin maliyetleri ve verimliliklerinin karşılaştırılması Şekil 1.1'de sunulmuştur.

Yöntemler	Seyahat Süresi	Bekleme Süresi	Durma Süresi	Başlangıç Maliyeti
SCOOT	-29% to -5%	-28% to -2%	-32% to -17%	$30,000 to $60,000
SCATS	-20% to 0%	-19% to +3%	-24% to +5%	$20,000 to $30,000
OPAC	-26% to +10%	-	-55% to 0%	$20,000 to $50,000
RHODES	-7% to +4%	-19% to -2%	-	$30,000 to $50,000
ACS Lite	-12% to +7%	-38% to +2%	-35% to -28%	$6,000 to $10,000

Şekil 0.1. Optimizasyon sistemlerinin karşılaştırılması [24].

Şekil 1'den anlaşılacağı üzere, uygulanan bütün gerçek zamanlı trafik sistemlerinde, seyahat süreleri, bekleme süreleri ve durmalar belirli oranlarda azalmıştır. Ayrıca sistemin kurulum maliyetleri de birbirine yakın miktarlardadır. ACS lite trafik optimizasyon sistemi mevcut sistemi iyileştirdiği için, ACS lite sistemi diğer gerçek zamanlı trafik optimizasyon sistemine oranla daha düşük ücretlerle kurulmaktadır. Trafik ışık optimizasyon sistemleri ile ilgili olarak yukarıda bahsedilen çalışmaları referans alan veya yeni birçok gerçek zamanlı trafik ışık sistemleri projelendirme

çalışmaları yürütülerek tamamlanmış veya halen birçoğu devam etmektedir. Nitekim, Bando ve arkadaşları 1995 yılında her aracın hareket denklemine dayalı, trafik sıkışıklığını gidermek için dinamik bir model geliştirmiştir. Geliştirilen bu modelde, her bir aracın hareket denklemine, önündeki aracın yasal hızı referans alınarak bir hız limiti eklenmiştir. Böylelikle trafik kararlılığı sağlanmaya çalışılmış ve trafik sıkışıklığının zamana bağlı olarak gelişimi incelenmiştir [24].

Dağüstü (2010) yürüttüğü yüksek lisans tez çalışmasında kent içi trafiğinin kontrolünde önemli role sahip olan sinyalize kavşaklar için Webster metodu kullanmış ve sinyal zamanlama algoritması geliştirmiştir. Önerdiği algoritmayı İstanbul'daki bazı sinyalize kavşakların üzerinde uygulanmıştır. Araştırıcı, uygulamada taşıt başına ortalama gecikme süresi, ortalama duruş sayısı, ortalama durma süresi, toplam CO ve NOx emisyonu, yakıt tüketimi, gözlenen bir saatlik süre içinde kavşağı terk eden taşıt sayısı ve toplam seyahat süresi alınarak kıyaslamalar yapmış ve geliştirdiği algoritmanın kavşakların performanslarında iyileştirmeler yaptığını göstermiştir. [25].

Lämmer ve Helbing 2010 yılında yürüttüğü çalışmada sürekli yeşil ışık sağlamaya yönelik, merkezi olmayan dinamik zamanlı trafik ışık kontrolünü kendi kendine sabitleştirilmiş metot ile yerine getiren bir sistem önermişlerdir. Önerdikleri sistem sabit zamanlı trafik ışıkları ile kıyaslandığında geliştirdikleri sistemin daha iyi performans gösterdiğini belirlemişlerdir.

Yürütülen başka bir çalışmada, trafik tıkanıklığı tahmini, trafik tıkanıklığı maliyetleri ve potansiyel tıkanıklığı azaltma stratejileri değerlendirilmiş, kaleme alınan raporda trafik sıkışıklığı maliyet analizini etkileyen çeşitli faktörler açıklanmış ve bu faktörlerin planlama kararlarını nasıl etkilediğini kapsamlı ve çok yönlü değerlendirmeler ile olası en iyi uygulamalar ortaya koyulmuştur.

Teklu ve arkadaşları güzergâh seçimlerini dikkate alarak trafik sinyallerini optimize etmek için farklı bir sistem geliştirmişlerdir. Ancak bu çalışmada sadece statik olaylara odaklanılmıştır[26]. Bununla birlikte, Bifulco ve arkadaşları sinyal optimizasyonu için statik olaylara değil ATIS (Advanced Traveler Information

System- İleri Seyahat Bilgi Sistemi) verilerine göre tekrar şartlarına odaklanarak çalışmışlardır [27].

Sam Yagar ve Bin Han Toronto'nun Queen caddesinde gerçek verilerle test edilmiş bir uygulama hayata geçirmişlerdir. Bu uygulamada sistem, kısa süreli bir dizi karar üretmiştir. Bu kararlar geçiş öncelikleri için alternatif gerçek zamanlı kurallar uygulamıştır. Gerçek verilerle yapılmış olan simülasyon testlerinde sabit zamanlı ışık sistemlerinden daha iyi sonuç verdiği görülmüştür [28].

Gerçek zamanlı trafik sinyal optimizasyon sistemlerinde bulanık mantık tekniği de sıklıkla kullanılmıştır. Bulanık mantık belirsizliklerin modellenmesinde kullanılan bir yapay zeka yöntemi olarak tanımlanmıştır. Bulanık mantık tekniğinin trafik sinyal optimizasyonundaki ilk uygulaması, Pappis ve Mamdani tarafından 1997 yılında gerçekleşmiştir[29]. Oluşturulan sistem iki tek yönlü yolun kesiştiği kavşakta test edilmiştir. Kuyruk uzunluğu, araç sayısı ve zaman girdi verileri olarak, yeşil ışık süresinin uzatılma verisi ise çıktı olarak bulanık mantık denetleyicisine girilmiştir. Sistem 25 kuraldan oluşan bir denetim gerçekleştirmiştir[30]. Kurulan bulanık mantık sistemi, sabit zamanlı sistem ile karşılaştırılmış ve daha iyi sonuçlar verdiği görülmüştür[31]. Nakatsuyama ve arkadaşları yapılan ilk bulanık mantık çalışmasını referans alarak geliştirmiş ve farklı kurallar içeren bir trafik ışık optimizasyon sistemi oluşturmuşlardır. Bu çalışma trafik uyarmalı ışık sistemi ile karşılaştırılmış ve ortalama gecikme sürelerinin %20 oranında azaldığı görülmüştür [32]. Sabit faz düzenini ele alan en kapsamlı çalışmalardan birisi Helsing Teknoloji Üniversitesindeki [33]çalışma olarak görülmüştür[29]. Bu yöntemde bulanık mantık sistemi 2 aşamalı olarak çalışmıştır. Birisi trafik durumunu değerlendirmiş, diğeri yeşil ve kırmızı ışık süreleriyle ilgilenmiştir. Trafiğin durumu doygun, normal ve düşük olarak 3 gruba ayrılmıştır. Işık süreleri ise herhangi bir faz devam ederken kavşaktaki araç kuyruklarına bağlı olarak ayarlanmıştır[33].

Yapılan gerçek zamanlı trafik ışık sistemi çalışmalarında, seyahat süresi ve rota tahminlerinden de yararlanılmıştır. 1997 yılında Choi ve Lee tarafından geliştirilen sistemde, rotadaki detektörler tarafından elde edilen araç sayısı ve seyahat süresi bilgileri değerlendirilmiş ve tahmini seyahat süresi bilgisi oluşturulmuştur. Yapılan

çalışma simülasyon programlarında test edilmiş ve iyi sonuçlar verdiği görülmüştür[34]. Rota tahmini için yapılmış birçok çalışma geliştirilmiştir[35],[36].

Rota tahmini konusunda, sürücülerin o anki durumları, alışkanlıkları, gideceği yerler gibi birçok belirsizlik olduğu için kesin sonuçlara ulaşılamamıştır. 2007 yılında Ayhan Erdem bulanık mantık tekniğini kullanarak C++ programlama dili ile bir gerçek zamanlı kavşak tasarlamıştır. Tasarlanan bu sistemin geleneksel yöntem olan sabit zamanlı trafik ışık sistemine göre birçok avantaja sahip olduğu görülmüştür. Bulanık mantık sistemi ile kavşaktaki araçlar algılanmış, yoğunluklara bağlı olarak sinyaller değiştirilmiş ve trafik rahatlatılmaya çalışılmıştır. Sistem, sabit zamanlı sistemlerle karşılaştırıldığında çok daha iyi sonuçlar verdiği görülmüştür[37]. Tzes, McShane ve Kim tarafından 1995 yılında yapılan bulanık mantık tekniği ile birlikte hem koordineli hem de ayrık kavşakların optimizasyonu için bir yöntem geliştirilmiştir. Bu yöntemde girdiler her bir yönden gelen araç sayılarından oluşturulmuştur. Simülasyon programlarındaki testler sonucunda, kurulan sistemin, sabit zamanlı sistemden daha iyi sonuç verdiği görülmüştür[38].

Gerçek zamanlı trafik ışık sistemlerinde günümüzde sıklıkla çalışılmaya başlanmış olan yöntemlerden birisi de öz-örgütleme yöntemidir. Öz-örgütlenme sistemi kendi kendine organize olan ve tanımlanan kurallar çerçevesinde kendi başına karar veren parçalardan oluşan kompleks bir sistemi tanımlar. Merkezdeki bir bilgisayar tarafından yönetilmek yerine, her bir kavşak kendini yönetir ve komşusu olan kavşak ile iletişim kurar. Böylece genelde bir senkronizasyon sağlanmış olmaktadır. Öz-örgütlenme başka bir deyişle kendi kendine organize olma sistemi, öğrenebilen bir sistemin veriyi saklarken verileri dışarıdan müdahale edilmeden organize etmesi ve yeni gelen veriye göre oluşumu düzenleyebilmesi anlamına geldiği söylenmiştir [39]. Whitacre 2007'de evrimsel optimizasyon algoritmaları ile öz-örgütlenme sistemini adapte ettiği tez çalışması gerçekleştirmiştir. Önerdiği optimizasyon algoritmasını performansını görmek için bir dizi numerik sınırlamasız test ve mühendislik tasarım problemleri üzerinde uygulamıştır. Elde ettiği deneysel sonuçlar öz-örgütlenme sisteminin gelişim algoritmasına adaptasyonu ile daha iyi performans elde ettiğini göstermiştir. Carlos Gershenson ve arkadaşları 2014 yılında trafik ışıklarında öz-örgütlemenin karmaşıklığının ölçülmesi konusunda çalışmıştır. Trafiğin iyileşmesi

için 6 adımdan oluşan bir algoritma tasarlamıştır. Bu çalışmada her bir kavşak için sensorlar kullanılmış ve araç sayısı bu sensorlar ile sisteme kayıt edilmiştir. Tasarlanan bu yöntem simülasyon programında geleneksel trafik ışık sistemi olan yeşil dalga ışık sistemi ve sabit zamanlı ışık sistemi ile karşılaştırılmış, bu sistemin daha iyi sonuçlar verdiği gözlenmiştir [40].

2015 yılında Inchul Yang ve R. Jayakrishan gerçek zamanlı olarak trafik sinyal optimizasyonu için 2 seviyeli bir çalışma yapmışlar ve RAIN adını vermişlerdir. RAIN sistem yapısı olarak UTOPIA [41]ve RHODES [21]sistemlerinin yapısına benzemiştir. Bu modellerde trafik kontrol problemleri hiyerarşik açıdan birbirine bağlı alt problemlere bölünmüştür. UTOPIA'daki hiyerarşik sistem bölge seviyesi ve yerel seviye olmak üzere 2 seviyeden oluşmuştur. Uzun süreli trafik tahminleri ve sinyal optimizasyonu bölge seviyesinde yapılmış ve son bulan işlemler yerel seviyeye aktarılmış, ışık süreleri bu yöntemle sisteme verilmiştir. RAIN sisteminin çalışmasında ise, strateji seviyesi, uzun zaman periyotları için (15 dakika gibi) optimal durumları tanımlamış ve hesapladığı yeşil ve kırmızı ışık sürelerini kontrol seviyesine aktarmıştır. Kontrol seviyesi aldığı bu süreleri daha küçük periyotlara bölmüş (60 saniye gibi) ve uygun bir plan oluşturmuştur. Sensorlar ile kavşaklardaki kuyruk uzunluklarını hesaplamış ve strateji seviyesinden gelen süreler ile karşılaştırılmıştır. Gerekli formüller kullanılarak kuyruk ağırlıkları sürekli olarak güncellenmiş ve en optimum ışık süreleri hesaplanmıştır [17].

Bu yöntemleri referans alan, geliştiren veya birleştiren birçok farklı çalışma da yapılmıştır. Cai ve arkadaşları 2009'da dinamik programlama tekniklerini online öğrenme teknikleri ile kombine ederek iyileştirmeler gerçekleştirmiştir. Aboudolas ve arkadaşları 2009'da saklama ve ileri modelleme yaklaşımı ve yeşil zamanlarının uzunluk sıralarını, esnek ayarlamalar ile dengelemeye çalışmışlardır [42]. Başka bir yaklaşım da kavşakların özerk olarak iyileştirilmesi için çok etkenli (multi-agent) sistemlerde, etkenlerin birbirleriyle olan işbirliği kullanılmıştır [43]. Genetik algoritmalar da trafik optimizasyonuna gerçek zamanlı olarak çözüm arayan yöntemlerden bir tanesidir. Yakıt tüketimi ve gaz emisyonlarını aza indirgemek için çok amaçlı optimizasyon şeklinde [44] ve yeşil ışık dalgası programların dinamik olarak yeniden optimizasyonunda uygulanmışlardır. Kompleks karar ağaçlarının

değerlendirilmesi ve farklı anahtarlama sıralamalarında alternatif optimizasyon metotları önerilmiştir. Bunlardan birisi trafik akşında dinamik öncelik belirleme tabanlı olanıdır. Ganji ve arkadaşları, trafik tıkanıklığında, tıkanıklığın düzelmesi için diferansiyel dönüşüm metodunu temel alarak bir matematiksel model geliştirmişlerdir. Bu modeli birkaç geleneksel yöntem ile kıyaslayarak geliştirdikleri yöntemin başarı elde ettiklerini göstermişlerdir [45].

1983 yılında Gartner gerçek zamanlı trafik kontrol sistemi önermiştir. Fazlar arasında geçiş yapmak için, gerçek verilerden yararlanılarak, parametrik olmayan bir model önermiştir [46]. 2004 yılında Nigarnjanagool ve Dia bir trafik optimizasyon algoritması önermiş ve Brisbane batı koridorunu test bölgesi olarak seçmişlerdir. Her bölüm için, maksimum hız, kapasite, eğim, şerit değiştirme mesafesi gibi veriler simülasyon programına eklenmiştir. Yapılan çalışmada merkez-hedef çiftlerine birer numara verilmiş ve bu çiftler birbirinden ayrılmıştır. Her zaman dilimi için (örnek olarak 15 dakika), yeşil ışık süreleri, detektörler aracılığıyla doygunluk oranları, trafik hacimleri tanımlanmıştır. Bir sonra ki zaman dilimi için, sinyal zamanlarını optimize etmede tahmini trafik hacimleri gerekli olduğu görülmüştür. Sistem bu tahminlerde önceki 3 zaman dilimindeki gerçek trafik hacimlerini kullanmıştır. Yapılan simülasyon testlerinde sabit zamanlı trafik ışık sistemlerinden daha iyi sonuç verdiği görülmüştür [9].

1.2. TRAFİK SİMÜLASYON PROGRAMLARI

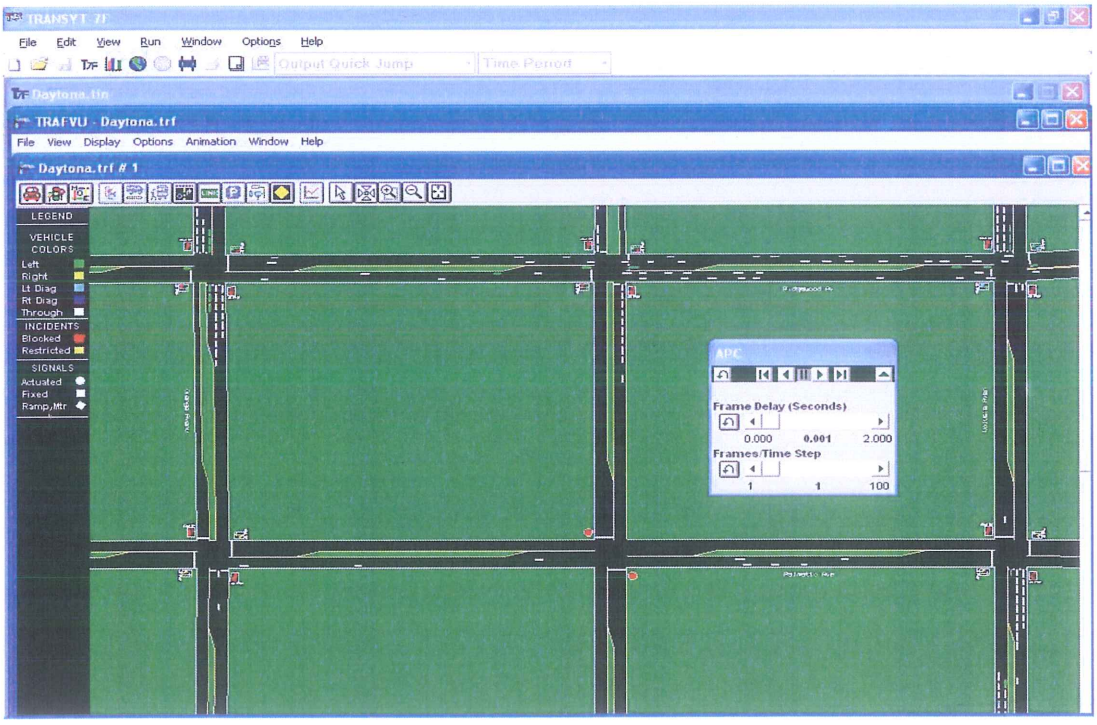
Şehirleşmenin artmasının trafik sıkışıklığı ve buna bağlı olarak birçok problemi meydana getirmesi ile birlikte, akıllı kavşak sistemlerinin artmasına neden olmuştur. Bu sistemlerin tasarlanması ve test edilmesi için en uygun maliyetli araçlardan birisinin simülasyon programları olduğu görülmüştür. Gerçek zamanlı trafik kontrol sistemleri içerisindeki, kontrol stratejileri, rota seçimi, çevre, güvenlik önlemleri, araç ve yol sayısı gibi birçok işlemi kullanıcılara sunan simülasyon programları geliştirilmeye başlanmış ve hala geliştirilmekte olan birçok çalışma vardır. Federal Karayolları İdaresi (FHWA) bu çalışmalara sponsor olmuştur. Simülasyon programları sadece planlamalar için değil, senaryo üretmek, ağ tahmini, optimizasyon sağlamak gibi alanlarda da geliştirilmiştir [47].

Bu çalışmalarda genel olarak mikroskobik ve makroskobik olarak 2 yaklaşım mevcuttur. Makroskobik yaklaşımlar için, trafik yoğunluğu ve trafik akışı çok önemli 2 parametre olarak tanımlanmıştır. Genelde otoban gibi geniş yolların simülasyonu için kullanılmıştır. Mikroskobik yaklaşımlar tek katılımcılar ve onlar arasındaki ilişkiyi tanımlamıştır. Bu yaklaşımlar trafik kuralları için temel oluşturmuştur [47]. Şehir trafik sisteminin optimizasyonunda sıklıkla mikroskobik yöntemler kullanılmıştır [48].

CORSIM 1997 yılında Federal Karayolları İdaresi tarafından Amerika Birleşik Devletleri'nde geliştirilmiş mikroskobik bir simülasyon programı olarak tanımlanmıştır. En yaygın olarak kullanılan mikroskobik simülasyon programlarından biri olarak kabul görmüştür. CORSIM iyi derecede araç takip sistemi, şerit değiştirme mantığına uygun, gerçek dünya çerçevesini kapsayan, ticari olarak temin edilebilen bir simülasyon programı olarak geliştirilmiştir [49]. CORSIM büyük ağlarda kodlamaları kolaylaştırmış, kullanıcı dostu bir ara yüze sahip olarak geliştirilmiştir. CORSIM simülasyon modeline dikkatli bir şekilde giriş ve çıkış parametreleri girildiğinde, gerçek hayattaki yol modeline uyumlu bir sisteme sahip olunur. Simülasyon çalıştırıldığında, optimum yollar, trafik koşulları ve ağ performansı simülasyon bitimine kadar sabit kalacak şekilde tasarlanmıştır. CORSIM içinde büyük kentsel ve bölgesel ağlarda oluşturulabilir. Akıllı kavşak bilgi sistemleri ile birlikte, sürücü davranışlarının modellenmesi sağlanmıştır. CORSIM için 1990 yıllarından itibaren birçok çalışma yapılmış ve farklı sürümleri çıkarılarak verimliliği arttırılmaya çalışılmıştır [47]. Owen ve arkadaşları 2000 yılında CORSIM modeli ve kullanımları ile ilgili bir çalışma yapmıştır. Bu çalışmada CORSIM' in gerçek zamanlı trafik sistemleri üzerindeki kullanımına odaklanmıştır [50]. Hansen ve arkadaşları 2000 yılında [51], Perrin ve arkadaşları 2002 yılında [52] gerçek zamanlı trafik kontrol sistemlerini analiz etmek için CORSIM kullanımını incelemiştir. Araştırıcılar CORSIM simülasyon programının kaliteli bir program olduğunu ve trafik mühendisliği alanında kullanışlı olduğunu kaydetmişlerdir [47]. Chien ve arkadaşaları 2002 yılında trafik kontrolünden kaynaklanan gecikmeleri simüle etmek için CORSIM simülasyon programını kullanmıştır [53]. 2004 yılında Yang ve Zahou geleneksel olarak sola dönüş sağlayan yollarda, sağa dönüşü ve u dönüşünü arttırmanın yararlarını incelemek için CORSIM simülasyon programı üzerinde

durmuştur [54]. Chien ve arkadaşları 2002 yılında geliştirdikleri iki yapay sinir ağı tarafından, otobüs varış saatlerini değerlendirmek için CORSIM simülasyon programını kullanmıştır [53]. CORSIM mikrosimilasyon modelini oluşturmak için veri gereksinimleri geniş yer almıştır. Her koridor için; tüm kavşak hareket sayıları, sinyal denetleyici ayarları, rota ve şerit geometrisi ve hız limitlerine ihtiyaç duyulmuştur. CORSIM' in güçlü yönleri; otoyollara açık modelleme, kolay ve çok yönlü kavşak tasarımı, çıkışların kapsamlı bir şekilde ayarlanması ve performans ölçütlerinin ayrıntılı analizi olarak gösterilmiştir [55].

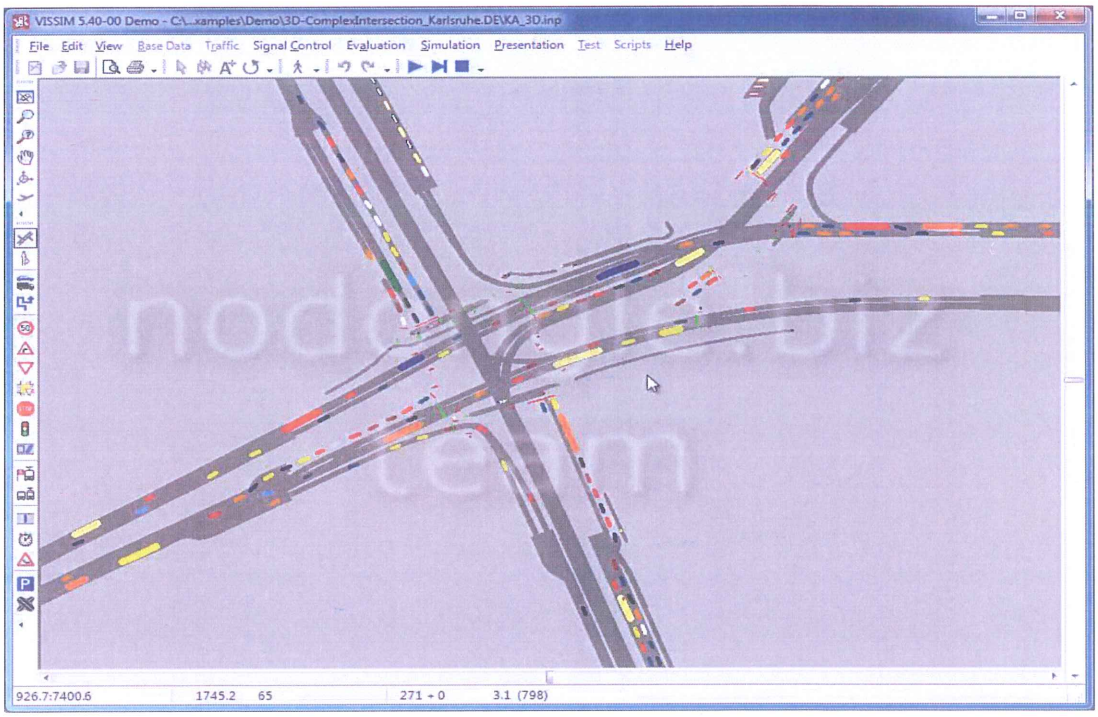
Bu sistemin zayıflıkları da incelenmiştir. Bir kavşak üzerindeki farklı yaklaşımlara ait veriler tek seferde görüntülenememiş, yeni pencere açmak gerekmiştir (Şekil 1.2) [56].



Şekil 0.2. CORSIM uygulama alan ekran görüntüsü.

Trafik optimizasyon sistemleri için tasarlanmış bir diğer trafik simülasyon uygulaması VISSIM'dir. VISSIM 1999 yılında Fransa' da PTV sistem yazılım ve danışmanlık hizmetleri tarafından geliştirilmiştir. Kentsel trafik sistemleri için geliştirilmiş, uygun simülasyon ekranları ile kullanıcılara seyahat süresi, gecikme

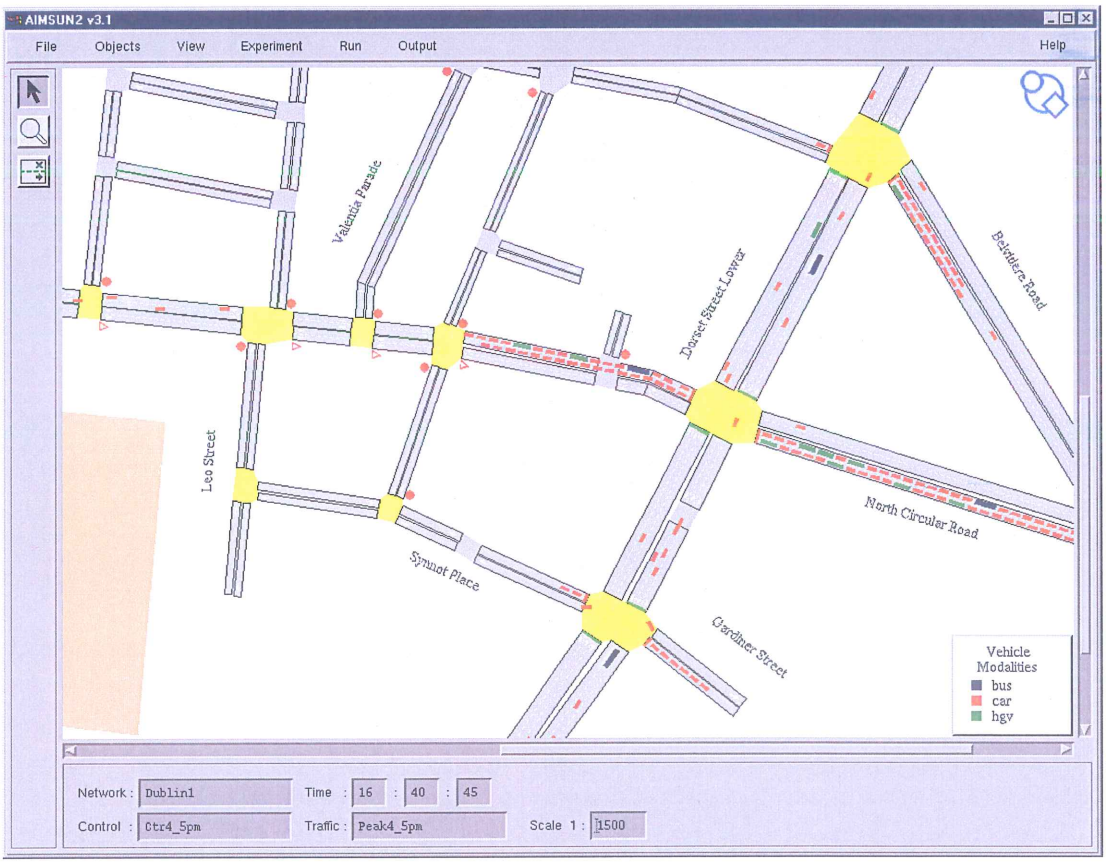
istatistikleri, kuyruk uzunluğu istatistikleri, detaylı sinyal zamanlama planları, geçiş öncelikleri ve hız profillerini sunabilen, mikroskobik bir simülasyon programıdır. Fizibilite testleri yapma, PC'de çalıştırılabilme, trafik destek hattı gibi güçlü yönleri görülmüştür. Bu özelliklere karşın, atama algoritması içermemesi ve giriş verilerinin kodlanmasının çok uzun zaman alması zayıf yönleri olarak görülmüştür [47]. Sistem yaya, bisiklet yolu, tekerlekli sandalye, demiryolu, motorlu taşıtlar, toplu taşıma, mal taşıma hatta uçak yolu dahil ve tüm trafik elemanlarını kapsayan bir simülasyon programıdır. Kentsel ve şehirlerarası alanlardaki analiz seçenekleri, programı trafik simülasyon programları arasında önemli bir yerde olmasına neden olmuştur. Farklı senaryolar ve farklı planlamalar kolaylıkla yapılabilir, ayrıntılı raporlar, grafikler ve 3D gösterimlerle analiz sunumları, çok daha net ve inandırıcı bir şekilde gerçekleşmektedir. Farklı kullanıcılar ve veri setlerini analiz etmek, güzergah seçimlerini görselleştirerek, kullanıcılara farklı seçenekler sunmuştur. Toplu taşıma yolları, durakları, bekleme süreleri, toplu taşıma türü gibi birçok konuda da kendisini geliştirmiş ve toplu taşıma konusunda kendini ispatlamıştır (Şekil 1.3) [57].



Şekil 0.3. VISSIM ekran görüntüsü.

Trafik simülasyon programlarından bir diğeri de AIMSUN2'dir. Bu sistem Barcelona Catalunya Politeknik Üniversitesi'nde J.Barcelo ve JL Ferrer tarafından

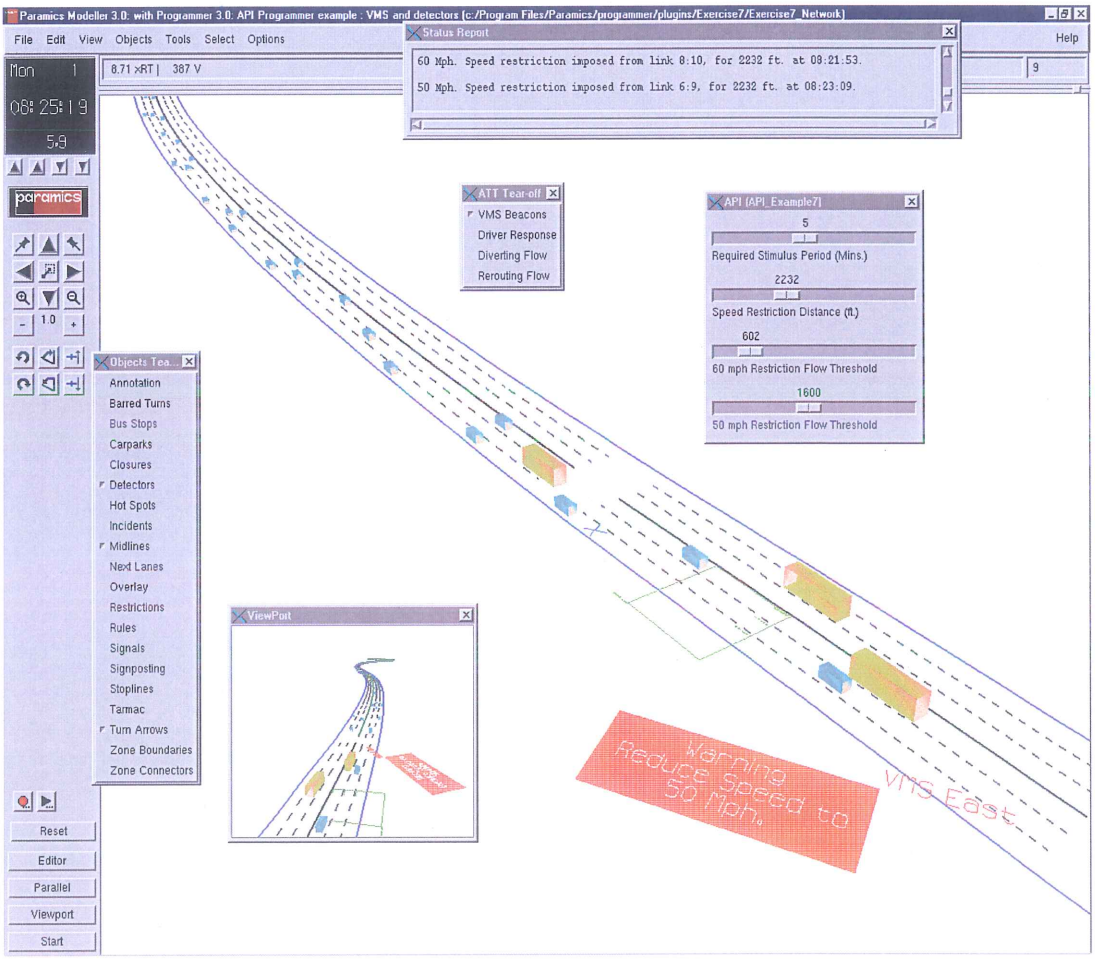
2005 yılında geliştirilmiştir [58]. Her aracın, şerit değiştirme, araç takip, araçlar arası mesafe gibi davranış özellikleri simülasyon boyunca güncellenir. AIMSUN2 ayrıntılı olarak, trafik akışı, araç hızları, seyahat süreleri gibi ayrıntılı istatistik çıktıları sağlamaktadır. AIMSUN2 gerçek dünya uygulamalarından elde edilen bilgiyi kullanmakta ve analiz etmektedir. AIMSUN2 bir araştırma ürünü olarak çalışılmaya başlanmış ve son zamanlarda ticari bir ürün olarak temin edilebilen bir trafik simülasyon programıdır [59]. PC'de çalıştırılabilir, kullanıcı dostu arayüzü, detaylı istatistik çıktıları, farklı trafik ağlarında kullanılabilen, trafik kontrol modelleri içerebilen bir simülasyon programıdır. Bu yönlerin yanı sıra ABD'de tercih edilmemiş, rota kılavuzu için verilerin harici olarak sisteme girilmesi gerektiğinden, bu durum sistemin zayıf yönleri olarak görülmüştür (Şekil 1.4) [9].



Şekil 0.4. AIMSUN2 simülasyon programı ekran görüntüsü.

Bir diğer trafik simülasyon programı PARAMICS'dir. PARAMICS İskoçya Edinburg Paralel Hesaplama Merkezi'nde UNIX iş istasyonu üzerinde geliştirilmiş bir mikroskobik simülasyon programıdır. Gelişmiş ve geniş donanımlara sahip olan

iş istasyonlarında kullanılarak, trafik sıkışıklığına çözüm aranmıştır. Simülasyonda, rota seçimi ve kirlilik izleme gibi konular için, bir sürücü ve araç tipi temsili olarak alınarak incelenmiştir. PAARAMICS sistemi, kapsamlı görüntüleme sistemi sağlayan, gerçek zamanlı trafik sistemlerinin simülasyonunda kullanılabilen, akıllı rota seçimi yeteneğine sahip bir trafik simülasyon sistemidir [56]. PARAMICS geliştirilerek 1996 yılında ABD de ve dünyada kullanılabilir ticari bir uygulama olarak piyasaya sürülmüştür (Şekil 1.5).



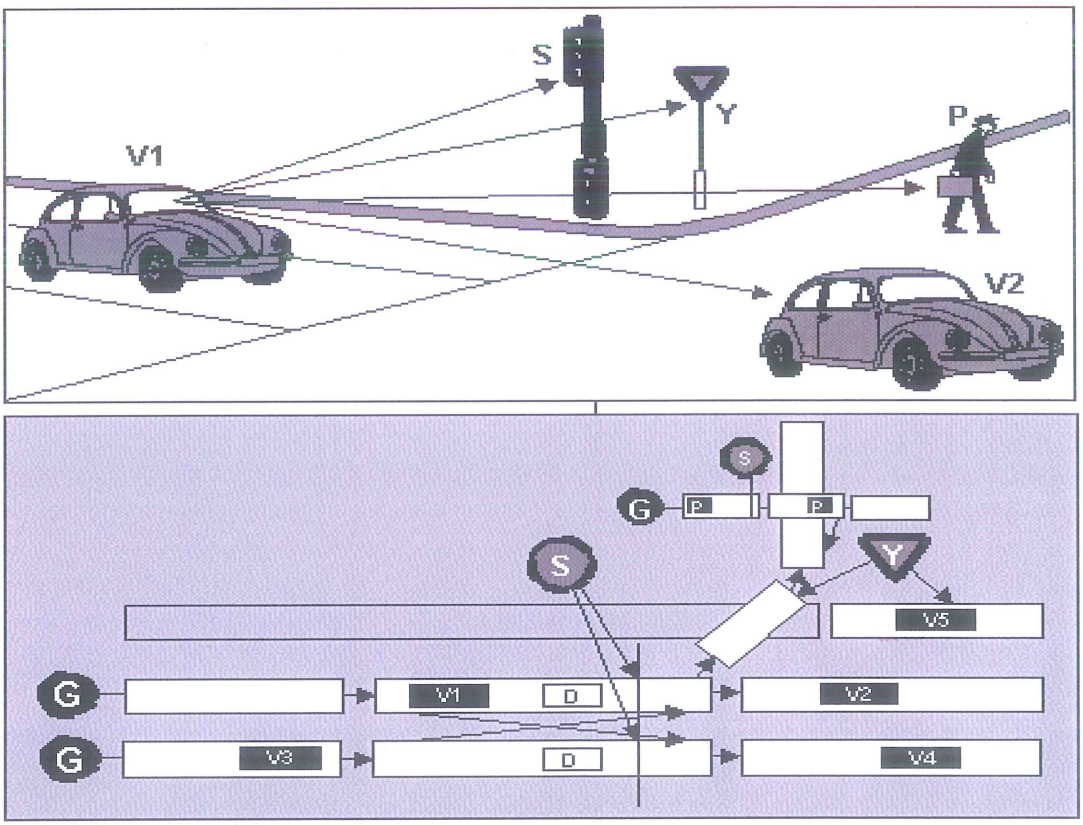
Şekil 0.5. PARAMICS trafik simülasyon programı ekran görüntüsü.

TRANSIM 1998 yılında New Mexico'daki Ulusal Los Alamos Laboratuvarı'nda geliştirilmiştir [60]. Ulaşım altyapısı ile sanal şehirler oluşturarak ağı bölgelere ayırmış ve modellemiştir. Oluşturulan bu sanal şehirde, bölgedeki insanların seyahat ve sürüş davranışları taklit edilmiştir. TRANSIM kullanılarak araç emisyon tüketimleri, araçların seyahat süreleri, ulaşım süreleri üretilen analizler tarafından

tahmin edilebilir ve sistemin genel performansını değerlendirebilirler. Modelde araçların takip ve şerit değiştirme mantığı hücresel otomat tekniğine bağlıdır. Simülasyon ortamı bireysel faaliyetleri planlar, bireysel seyahatleri ve trafik yüklerini inceler, bölgesel nüfus, çevre etkileri ve ulaşım sistemi üzerine etkileri simülasyon ortamında analiz edilir [56].

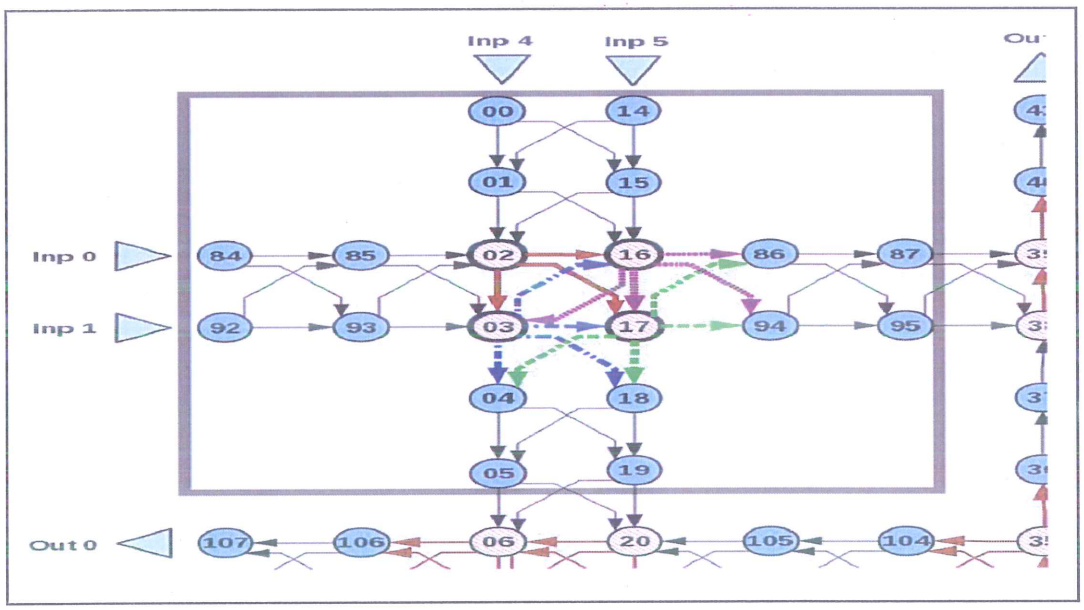
HUTSIM 1989 yılında Finlandiya'daki Helsinki Üniversitesi tarafından Ulaştırma Mühendisliği Laboratuvarı'nda geliştirilmiş trafik simülasyon programıdır. Karmaşık kavşak ve değişen trafik koşullarını analiz etmek için geliştirilmiştir. Gerçek sinyal kontrolleri simülasyona girilerek çalışmaktadır. Esnek, ayrıntılı modelleme, gerçek zamanlı animasyon ve ekran çıkışı sağlamaktadır [56].

HUTSIM, yol yönetici firmalar, şehir planlama ofisleri ve trafik danışmanlık şirketleri tarafından kullanılmakta olan ticari bir üründür (Şekil 1.6) [61].



Şekil 0.6. HUTSIM örnek kavşak simülasyon ekran görüntüsü.

Diğer bir sistem ise Celler Automata olup, karmaşık taşıma sistemleri için kullanılan basit bir simülasyon tekniğidir. Bu yöntem günümüze kadar biyoloji, tıp, matematik, fizik gibi birçok alanda kullanılmıştır [62]. Trafik simülasyon programı olarak ilk kullanımı 1990 yılında Nagel ve Schreckenberg tarafından gerçekleştirilmiştir [63]. Bu çalışma Almanya'daki motorlu taşıtlar ve şehir trafiğinin gerçek akışına yönelik bir çalışmadır. Bu sistem mikroskobik ve makroskobik simülasyon programı olarak kullanılabilmektedir. Sistem daha çok büyük ölçekli ağlarda kullanılabilmekte, hücre sistemi olarak uygulama bulmakta ve kural tabanlı bir sistem olarak bilinmektedir (Şekil 1.7) [64].



Şekil 0.7. Cellular Automata simülasyonu ekran görüntüsü.

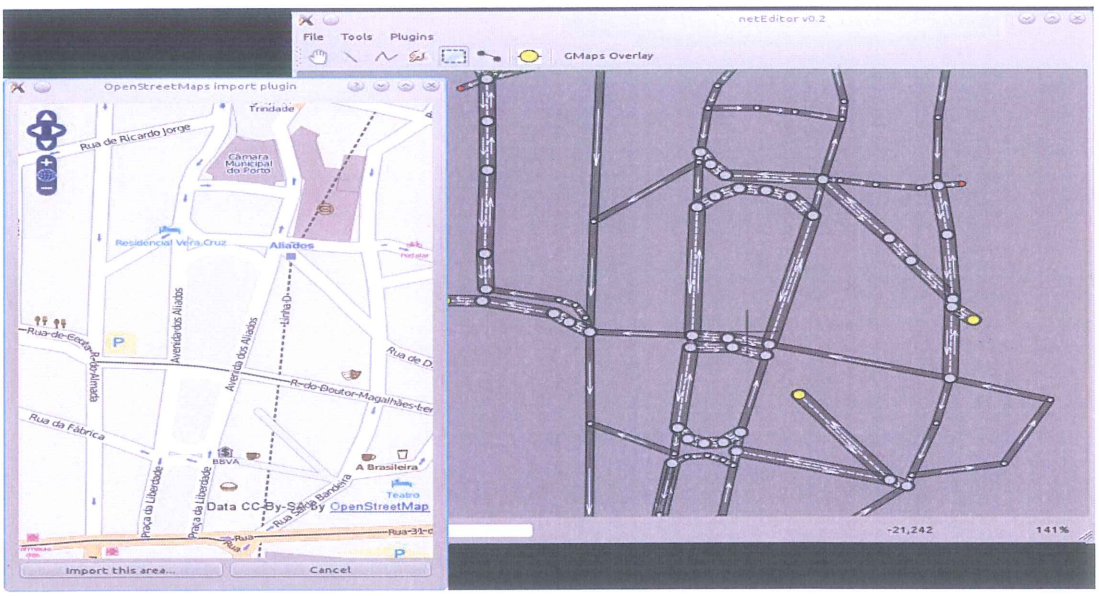
Dünya genelinde yaygın olarak kullanılan trafik simülasyon programlarından bir diğeri de SUMO (Simulation of Urban Mobility)'dur [65]. SUMO'nun ilk sürümü 2001 yılında Almanya Taşımacılık Sistemleri Enstitüsü tarafından geliştirilmiştir. SUMO açık kaynak kodlu olarak Python programlama dilinde yazılmış, mikroskobik bir trafik simülasyon programıdır. SUMO sistemi içerisinde trafiğe etki eden tüm elemanlar (araçlar, yollar, sinyalizasyon, yayalar, gaz emisyonları vb) mevcuttur. SUMO trafik simülasyon programı güncellenmiş, farklı sürümleri geliştirilmiş ve şehir trafiği için önemli bir araç haline gelmiştir [66]. SUMO'da trafik sahnelerinin işlemek için OpenGL programı kullanılmaktadır. Bu sayede sadece ögeyi seçerek ve fare tıklamasıyla çizimler kolay bir şekilde yapılabilmektedir. Ayrıca SUMO'da tüm

araçları almak için sensörlerin görüş açısının yeterli sonuçlar verdiği görülmektedir. SUMO trafik simülasyon programında her araç özerk olarak incelenmekte ve işlenmektedir. Veriler XML dosyasına yüklenir (Çizelge 1.1) ve simülasyon bu verileri referans alarak çalıştırır.

Çizelge 0.1. SUMO trafik simülasyon programı için XML verileri.

Vehicle id="veh1"	Route= "route01"	Type="carA"	Color="1,0,0"	Depart="0010"
Vehicle id="veh2"	Route= "route7890"	Type="autonomous"	Color="1,0,0"	Depart="0020"

Çizelge 1.1'de verilen Vehicle araca verilen kimlik, route aracın gideceği rota, type araç tipi, depart aracın hareket zamanıdır. Veriler hazırlandıktan sonra uygun formüller kullanılarak SUMO trafik simülasyon programı çalıştırılır [65]. SUMO trafik simülasyon programında giriş-çıkış verilerinin eklenmesini sağlamak ve elle haritanın oluşturulması için netEditor eklentisi kullanılmaktadır. netEditor programı açık kaynak kodludur ve ağı modellemek için kullanılmaktadır (Şekil 1.8) [67].



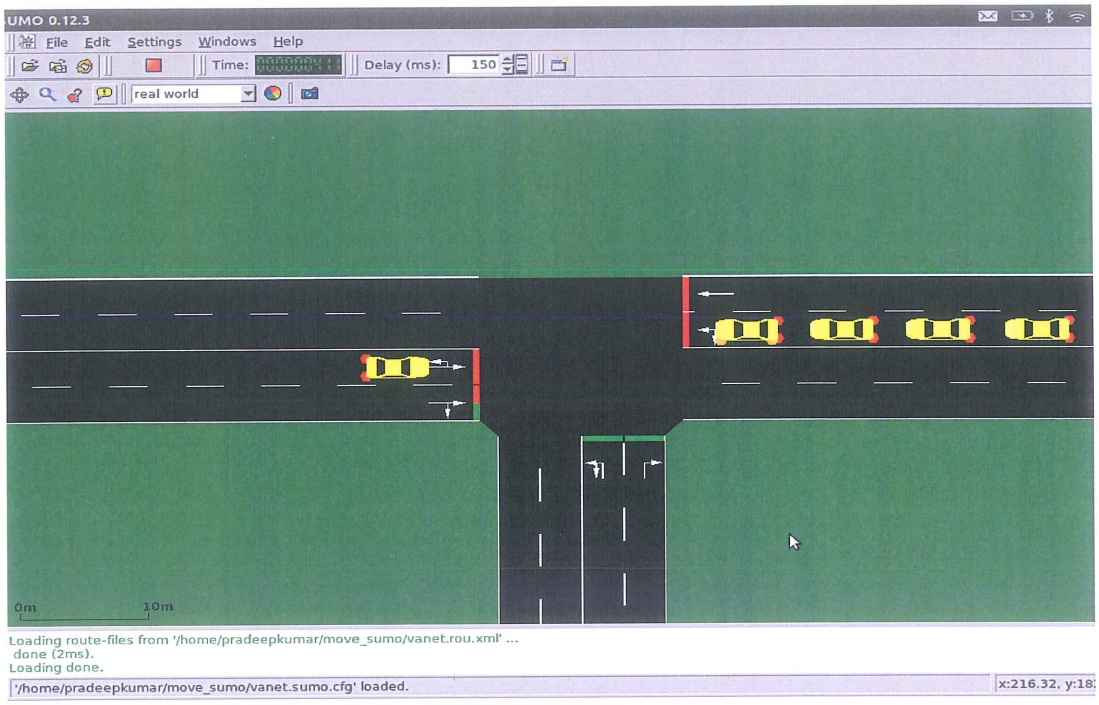
Şekil 0.8. netEditor v02 ekran görüntüsü.

SUMO trafik simülasyon programında haritaları elle eklemek yerine, openstreetmap haritalama programını kullanarak da veriler eklenebilmektedir. Openstreetmap haritalama programı, içerisinde hem coğrafi bilgileri hem de trafik bilgilerini barındırmaktadır. SUMO trafik simülasyon programına dışardan erişim için, TraCI uygulama arayüz modülü geliştirilmiş ve SUMO'ya entegre edilmiştir. Bu sayede SUMO simülasyon programı ile Matlab yazılım programı arasında bağlantı kurulmuştur. SUMO programı kendi içerisinde trafik ışıklarına dayalı bir çözüm algoritmasına sahiptir. Simülasyon için kullanıcılar isterlerse SUMO içerisinde ki çözüm algoritmasını kullanabilmekte, isterlerse Matlab programı aracılığıyla kendi çözüm algoritmalarını üretebilmektedirler [66].

SUMO trafik simülasyon programının güçlü yönleri

1. Farklı araç türlerini yönetebilir.
2. Çok şeritli caddelerde çalışabilir ve şerit değiştirme mantığına sahiptir.
3. Trafik sinyalleri kullanılmaktadır.
4. Kullanıcı arayüzü için OpenGL grafik programı kullanılmaktadır.
5. Çok yüksek sayıda yönü olan sokaklarda çalışabilmektedir.
6. Hızlı işlem hızına sahiptir.
7. Farklı uygulamalar ile çalışabilmektedir.
8. Detektör tabanlıdır ve komut satırı işletilmektedir.
9. Açık kaynak kodludur (GPL)
10. Standart C++ kütüphaneleri kullanılabilmektedir.
11. Windows ve Linux işletim sistemlerinde kullanılabilmektedir (Luis & Pereira, 2011).

SUMO mikroskobik trafik simülasyon programı günümüz trafik sistemlerinde sıkça kullanılan bir programdır (Şekil 1.8) [68].



Şekil 0.9. SUMO trafik simülasyon programı ekran görüntüsü.

BÖLÜM 2

MATERYEL VE METOT

Çalışmamızda günümüze kadar çalışılmış ve dünya üzerinde en çok kullanılan trafik ışık optimizasyon sistemleri olan, sabit zamanlı trafik ışık sistemleri, yeşil dalga trafik ışık sistemi, gerçek zamanlı trafik ışık sistemleri hazırlanan çalışma ortamında incelenmekte ve bu sistemler karşılaştırılmaktadır. Bahsedilen trafik ışık optimizasyon sistemleri, 6 farklı yoğunlukta ve her bir yoğunluk için rastgele 5 farklı rota oluşturulmuştur. Oluşturulan her bir yoğunluk ve her bir rota SUMO trafik simülasyon programında test edilmiştir. Elde edilen test sonuçları BÖLÜM 3'deki BULGULAR başlığı altında verilmiştir.

2.1. KULLANILAN SABİT ZAMAN TRAFİK SİSTEMİ

Karabük-Safranbolu yolu arasında, karayollarının verdiği ışık düzeni bire bir uygulanmıştır. Kavşaklarda ki yeşil ışık ve kırmızı ışık süreleri, her bir kavşak 30'ar dakika incelenerek ve gerçek veriler kayıt altına alınarak test ortamına eklenmiştir. Karabük otogar kavşağına kavşak 1, Safranbolu kavşağına kavşak 7 ismini verdiğimizde, toplamda 7 adet kavşak mevcuttur. Kavşak 1 için ana yoldaki ışık süreleri 76 saniye kırmızı 27 saniye yeşil, tali yollar için ise 82 saniye kırmızı 33 saniye yeşildir. Kavşak 2 için ana yolda 60 saniye kırmızı 15 saniye yeşil, tali yolda 45 saniye kırmızı 15 saniye yeşil ışık yanmaktadır. Kavşak 3 için ana yolda 40 saniye kırmızı 15 saniye yeşil, tali yolda 26 saniye kırmızı, 33 saniye yeşil ışık yanmaktadır. Kavşak 4 için ana yolda 43 saniye kırmızı 60 saniye yeşil, tali yolda 70 saniye kırmızı, 36 saniye yeşil ışık yanmaktadır. Kavşak 5 için ana yolda 28 saniye kırmızı 47 saniye yeşil, tali yolda 60 saniye kırmızı, 17 saniye yeşil ışık yanmaktadır. Kavşak 6 için ana yolda 30 saniye kırmızı 30 saniye yeşil, tali yolda 30 saniye kırmızı, 30 saniye yeşil ışık yanmaktadır. Kavşak 7 için ana yolda 45 saniye kırmızı 59 saniye yeşil, tali yolda 47 saniye kırmızı, 27 saniye yeşil ışık yanmaktadır.

2.2. ÖNERİLEN YEŞİL DALGA TRAFİK SİSTEMİ

Karabük-Safranbolu ana yolunda bulunan ışıklar arasındaki mesafeler, yasal hız sınırı olan 70 km/saat hız ile gidildiğinde ışıklara varış süreleri hesaplanmıştır. Sistem çalışmaya başladığından kavşak 1 e varış süresi 60 saniyedir. Kavşak 2'ye varış süresi 86, kavşak 3'e varış süresi 174 saniye, kavşak 4'e varış süresi 241 saniye, kavşak 5'e varış süresi 313 saniye, kavşak 6'ya varış süresi 408 saniye ve kavşak 7'ye varış süresi 457 saniyedir. Bu süreler kullanılarak Karabük-Safranbolu yolu arasında 70 km'lik yasal hız sınırı kullanılarak yeşil dalga optimizasyon sistemi kullanılmıştır.

2.3. ÖNERİLEN GERÇEK ZAMANLI TRAFİK IŞIK MODELİ

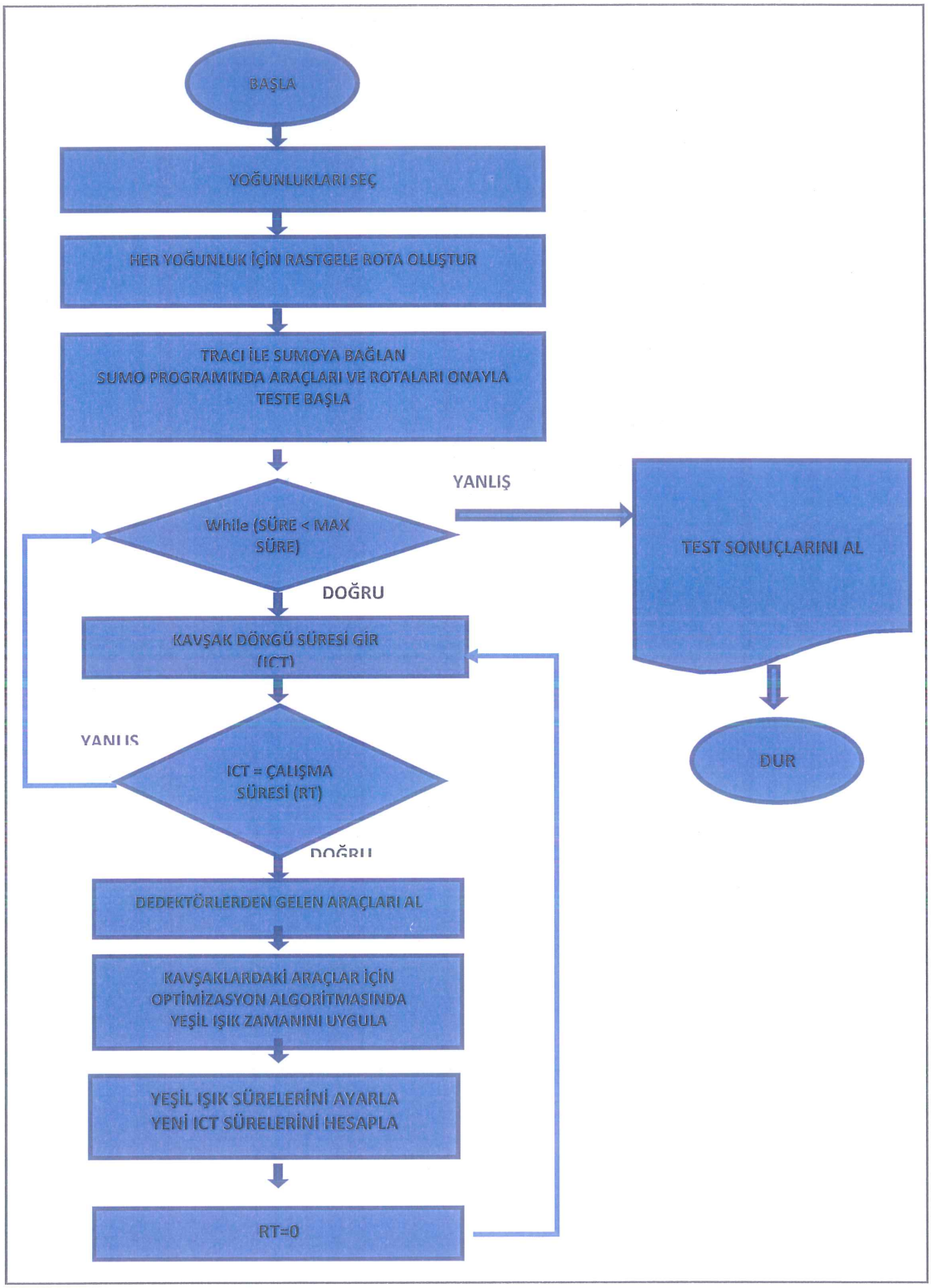
Bizim çalışmamızda her bir kavşak, trafik ışıklarını merkezi kontrol sisteminden bağımsız olarak ilgili kavşağın trafiğinin yoğunluğuna bağlı olarak her bir gelen yön üzerinde bulunan sensorlardan alınan araç sayılarını dikkate alarak uyguladığımız algoritmaya bağlı olarak kendi kendine karar verebilen akıllı bir trafik sistemi tasarladık.

Sistemde yer alan her bir araç yaklaşık olarak 7,5 m yer kaplamaktadır. Kavşaklarda ki uygun yerler de sensorlar mevcuttur ve bu sensorlar aracılığı ile her bir şeritte ki araç sayıları sisteme yüklenmektedir.

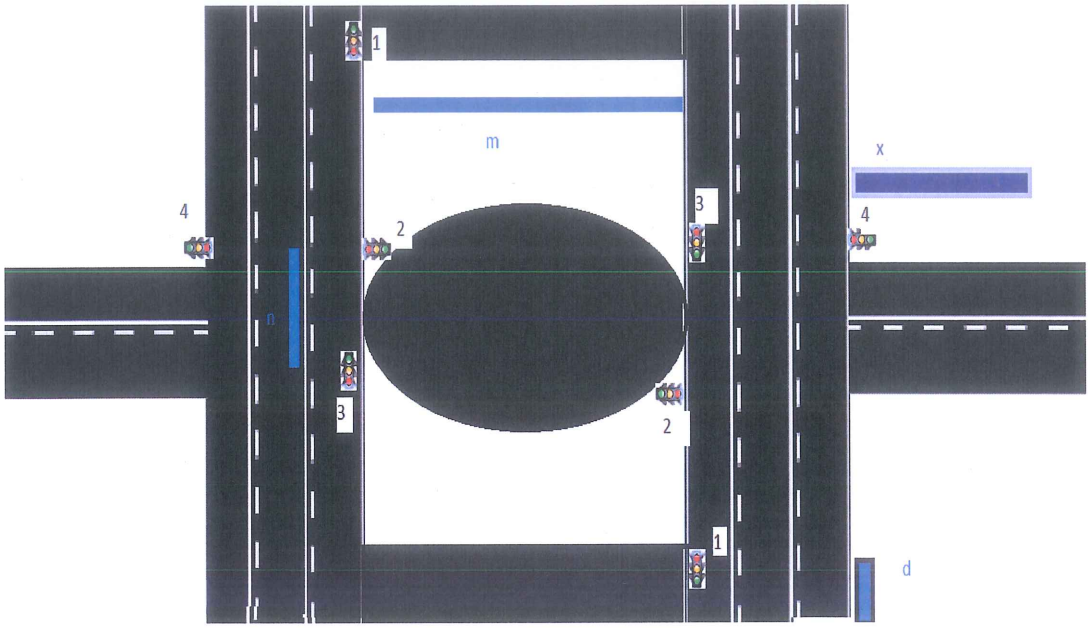
Her bir şeritteki kuyruk uzunluğu, araç sayısı ile 7,5 m'nin çarpılması ile hesaplanmaktadır. Ana yolda 3 gidiş ve 3 geliş olmak üzere 6 şerit, tali yolda ise 1 gidiş ve 1 geliş olmak üzere 2 şerit mevcuttur.

Sistemimiz her bir yönde ki bütün şeritleri incelemekte ve en fazla araç sayısının bulunduğu (en uzun kuyruk) şeritti baz alarak işlem yapmaktadır.

Sistemin akış diyagramı Şekil 2.1'de, tasarlanan kavşak sistemi Şekil 2.2'de gösterilmiştir.



Şekil 2.1. Gerçek zamanlı trafik ışık optimizasyon sistemi akış diyagramı.



Şekil 2.2. Tasarlanan akıllı kavşak sistemi.

Tasarladığımız gerçek zamanlı trafik ışık optimizasyon sisteminin 8 adımdan oluşan algoritması;

1. Tüm şeritlerdeki araç sayılarını incele, en uzun kuyruğu al
2. D eşik değerini geçerse 1, 3 yeşil ve 2, 4 kırmızı
3. M eşik değerini geçerse 1. Madde iptal olur ve 2,4 yeşil ve 1,3 kırmızı
4. N eşik değerini geçerse 1. Ve 2. Madde iptal olur ve 1,3 yeşil ve 2,4 kırmızı
5. İlk 3 maddedeki koşullar sağlanmıyorsa ve 2,4 numaralı ışıkta araç yoksa 1,3 yeşil
6. 1,3 numaralı ışıklar da d eşik değeri sağlanmamışsa ve 2 veya 4 numaralı ışığa araç geldiyse 2,4 yeşil ve 1,3 kırmızı olur
7. M veya X eşik değerleri aşılmazsa ve 1,3 ışıkları 90 saniyeden fazla yeşil yanarsa, 1,3 kırmızı, 2,4 yeşil
8. D ve N eşik değeri aşılmazsa ve 2,4 ışıkları 90 saniyeden fazla yanarsa, 1,3 yeşil, 2,4 kırmızı

2.4. YOĞUNLUĞUN HESAPLANMASI

Trafik testlerinde başarım ölçüleri farklı trafik yoğunlukları üzerinde test yapılması ile mümkün olmaktadır.Trafik yoğunluk oranları genellikle teste tabi tutulan alanın

maksimum araç sayısı üzerinde 0-1 arasında seçilen trafiği yoğunluğuna tabi tutularak test edilmektedir.

Maksimum araç sayısını elde edebilmek için test alanının uzunluğu, araç uzunluğu ve iki araç arasındaki güvenli mesafelerin hesaplanması ile elde edilebilir. Bu çalışmada teste tabi tutulan caddelerin toplam uzunluğu 8.350 Metredir. Bir binek aracın ortalama uzunluğu 5 metre olarak alınmıştır. İki araç arasındaki güvenli mesafe araç hızlarının (km/saat) ikiye bölünmesi ile elde mesafedir (m). Trafikte maksimum yoğunluk araçların hareket edemeyip durduğu ya da durmaya yakın olduğu andır. Dolayısıyla burada araçların maksimum yoğunlukta saatte 5 km/saat hızla gittiğinde iki araç arasındaki mesafe 2,5 m olarak elde edilir. Bu değerler alındığında bir her bir araç için yolda 7,5 metrelik bir alan gerekir. Bu da her bir şerit için 8350/7,5 = 1113 araca tekabül etmektedir. Karabük-Safranbolu yolu 6 şeritten oluşmaktadır. Bu durumda maksimum araç sayısı 1113×6= 6678 olarak hesaplanır. Bu hesaplamalar sonucunda yoğunluk oranlarına göre hesaplanan araç sayıları Çizelge 2.1'de verilmiştir.

Çizelge 2.1. Yoğunluklara göre araç sayıları.

Yoğunluk No	1	2	3	4	5	6
Oranı (0-1)	0,15	0,30	0,45	0,60	0,75	0,90
Araç Sayısı	1002	2003	3005	4007	5009	6010

2.5. ARAÇ GEÇİŞLERİNİN HESAPLANMASI

Yaptığımız bu çalışmada minimum fonksiyonun tespiti, bir kavşaktaki her bir şerit üzerinde bulunan araç sayıları, gerekli olan süre ve algoritmanın bulduğu sürenin birlikte hesaplanması ile yapılmaktadır. Bir şerit üzerinde, kuyrukta bekleyen araç sayısının ne kadar sürede kavşağı terk edeceğinin hesaplanması ve süreleri çizelge

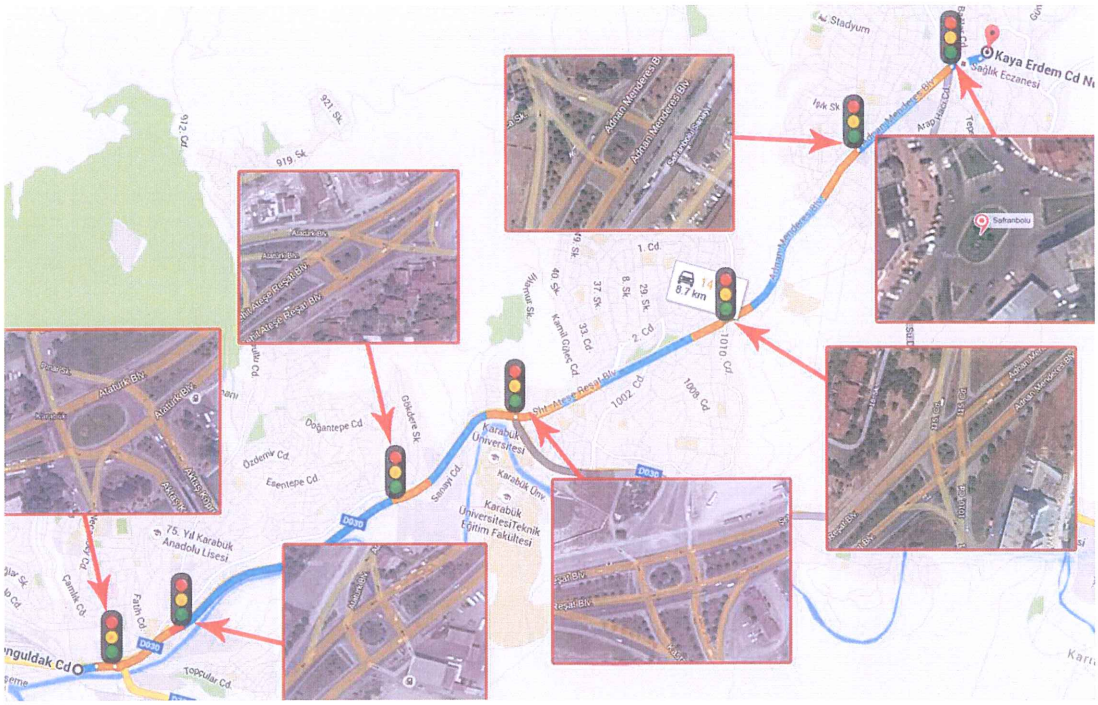
2.2'de verilmiştir.Verilen tabloda ki değerler göz önüne alındığında, 90 saniyede ortalama 35 aracın tahliye edilebileceği hesaplanmıştır.

Çizelge 2.2. Yeşil ışıkta duran araç sayısının ışığı terk etmesi için gereken süreleri (Anonymous,2015).

Şeritteki Sırası	Gözlemlenen Zaman Aralığı (sn)	Sabit Akışta Zaman Aralığı (sn)	Eklenen Başlangıç Zamanı (sn)
1	3.8	2.1	1.7
2	3.1	2.1	1
3	2.7	2.1	0.6
4	2.4	2.1	0.3
5	2.2	2.1	0.1
6 ve fazlası	2.1	2.1	0

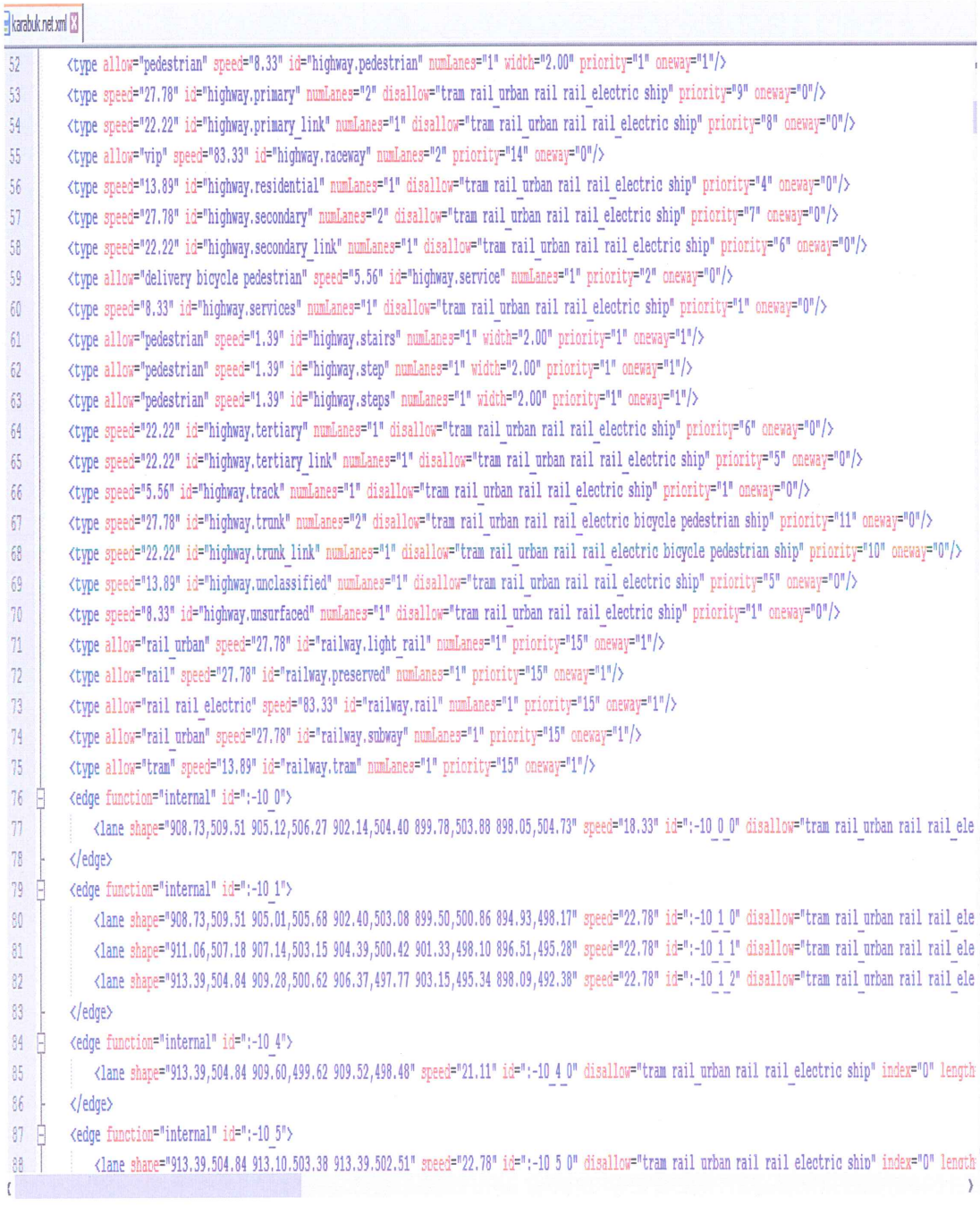
2.6. ÇALIŞMA ALANI

Çalışma alanı olarak yaklaşık 8.5 km'lik Karabük-Safranbolu yolu seçilmiştir. Bu çalışma alanında Şekil.2.2'de gösterildiği gibi 7 adet ışıklı kavşak bulunmaktadır.



Şekil 2.3. Karabük-Safranbolu yolu çalışma alanı.

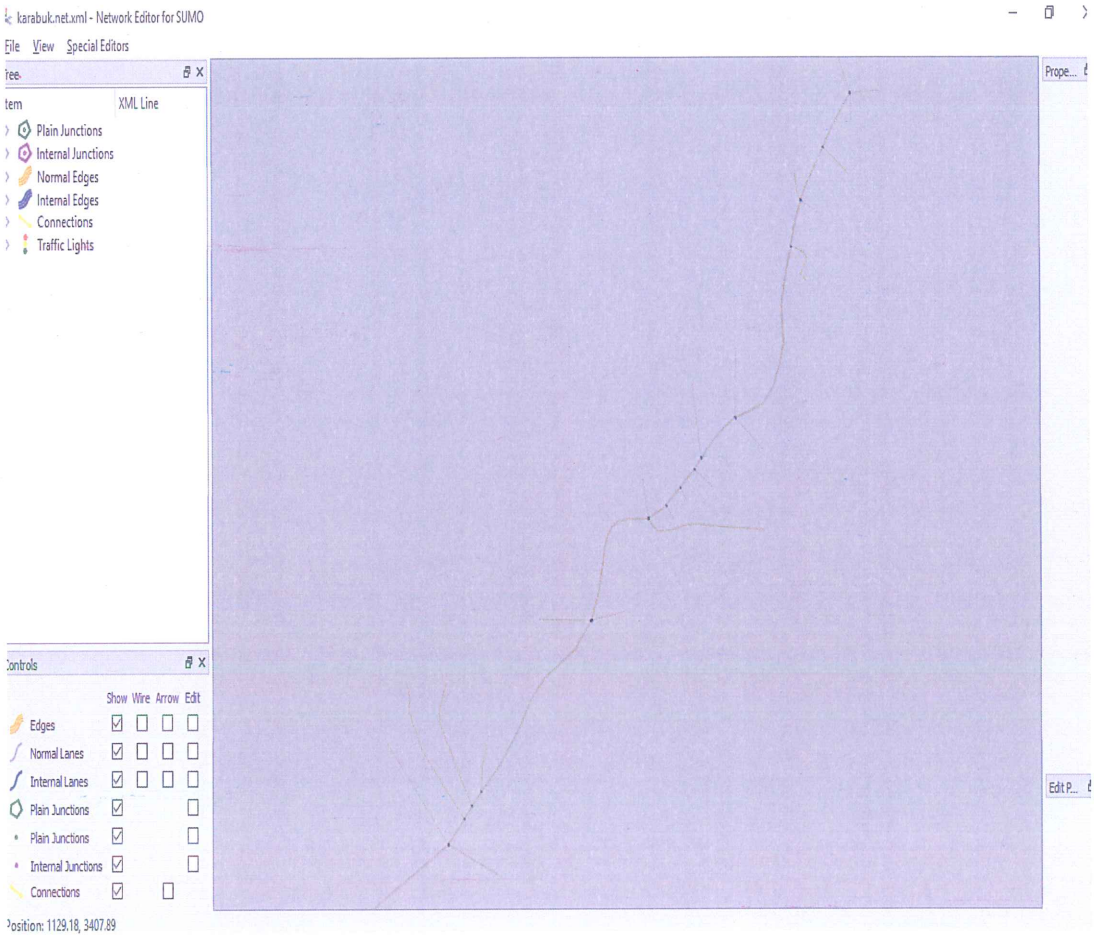
Seçilen çalışma ortamı için Openstreetmap haritalama programından Karabük-Safranbolu yolu indirilmiş ve haritadaki veriler Şekil 2.3'de gösterildiği gibi karabuk.net.xml dosyasına kaydedilmiştir.



Şekil 2.4. karabuk.net.xml dosyası ekran görüntüsü.

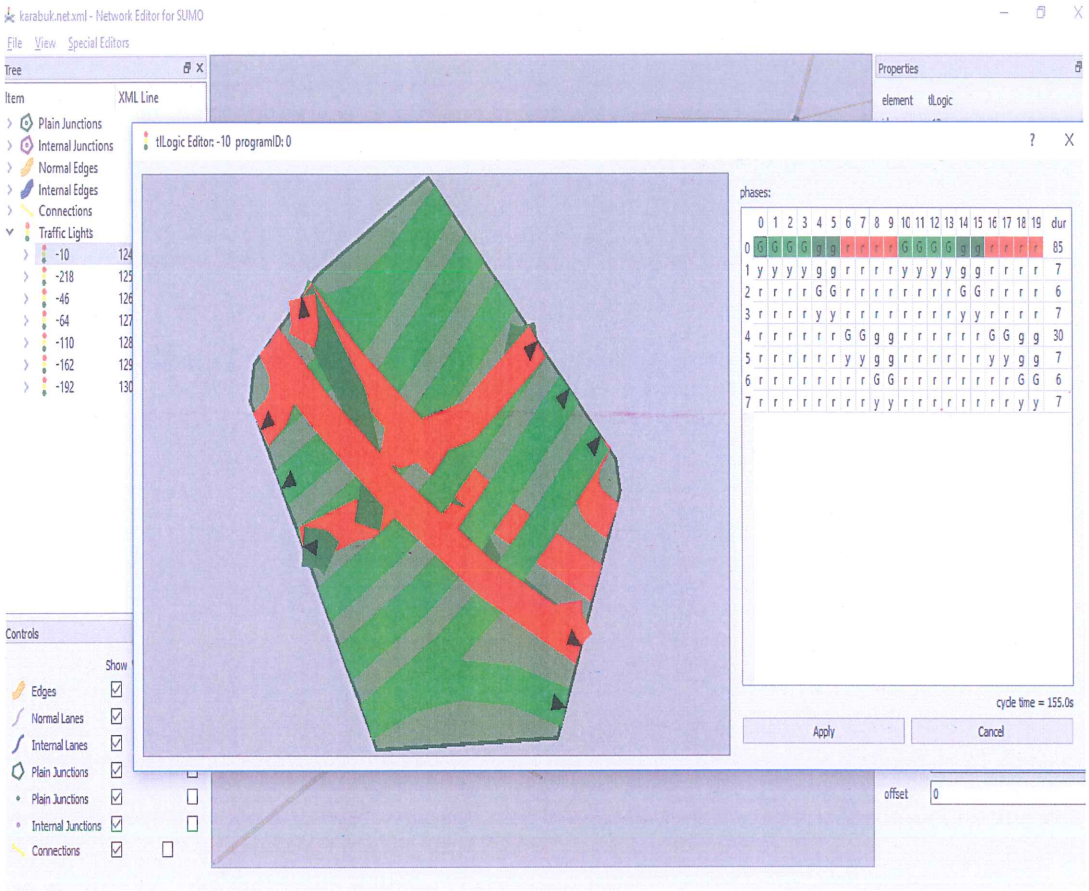
OpenStreetMap harita programı tarafından indirilen, Karabük-Safranbolu haritası tarafından alınan bilgilerin bulunduğu karabuk.net.xml dosyası Net Editor For

SUMO programı kullanılarak açılır ve güzergah üzerinde belirlenen gerçek veriler (yollar, ışık süreleri, fazlar) dosyaya yüklenir (Şekil 2.4).



Şekil 2.5. NetEditor4Sumo Karabük-Safranbolu yolu ekran görüntüsü.

Oluşturulan çalışma alanında, ışıkların süreleri 8 fazlı bir biçimde, kavşaklardaki ışıkların gerçek süreleri kullanılarak ayarlanmıştır (Şekil 2.5).



Şekil 2.6. NetEditor4SUMO programı ışık süreleri ve faz ayarları ekran görüntüsü.

Şekil 2.5'de verilen ekran görüntüsünde yukarıdan aşağı doğru ve 0'dan 7'ye kadar olan rakamlar kavşaktaki faz numaralarını, şeklin sonunda ki 85'den başlayıp 7'ye kadar olan rakamlar ise fazın periyot süresini göstermektedir. Yani gösterilen kavşaktaki ilk faz 85 saniye boyunca uygulanmaktadır. Periyot süresi boyunca gösterilen renkli kısımlardaki G ışığın tam yeşil, g eğer uygunsa yeşil, y sarı ve r kırmızı ışık olduğunu göstermektedir.

2.7. ROTALARIN OLUŞTURULMASI

6 farklı yoğunlukta, her yoğunluk için farklı 5 rota rastgele olarak Şekil 2.6'de gösterilen kodları kullanarak üretilmiştir.

```
netconvert --osm-files karabuk.osm -o karabuk.net.xml

randomTrips.py -n karabuk.net.xml -r karabuk1.rou.xml -e 1002 -l
randomTrips.py -n karabuk.net.xml -r karabuk2.rou.xml -e 2003 -l
randomTrips.py -n karabuk.net.xml -r karabuk3.rou.xml -e 3005 -l
randomTrips.py -n karabuk.net.xml -r karabuk4.rou.xml -e 4007 -l
randomTrips.py -n karabuk.net.xml -r karabuk5.rou.xml -e 5009 -l
randomTrips.py -n karabuk.net.xml -r karabuk6.rou.xml -e 6010 -l

DUAROUTER -n karabuk.net.xml -o karabukx.rou.xml

duarouter --trip-files=trips.trips.xml --net-file=karabuk.net.xml  --output-file=MySUMORoutes.rou.xml

randomTrips.py -n karabuk.net.xml  -r karabukx.rou.xml --trip-attributes="departLane=\"best\" departSpeed=\"max\" departPos=\"random\""

py output\xml2csv.py outputkarabuk10bin.rou.xml
```

Şekil 2.7. Farklı yoğunluklarda araç üreten XML kodları.

1 yoğunluk için 1002 araç, 2. yoğunluk için 2003 araç, 3. yoğunluk için 3005 araç, 4. yoğunluk için 4007 araç, 5. yoğunluk için 5009 araç ve 6. yoğunluk için 6010 araç üretilmiştir. Verilen bu rakamlardan anlaşılacağı üzere, rota oluşturma işlemi sonucunda, birbirinden farklı 30 adet rota oluşturulmuştur. Oluşturulan rotalardan ilki olan karabuk1.1.xml dosyasının ekran görüntüsü Şekil 2.7'de verilmiştir.

karabuk11.rou.xml

```
</configuration>
-->

<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes_file.xsd">
    <vehicle id="0" depart="0.00">
        <route edges="-15281#3 -15285#0 -15285#1 -15285#2 -15285#3 -15285#4 -15285#5 -15301#0"/>
    </vehicle>
    <vehicle id="1" depart="1.00">
        <route edges="-15285#0 -15287"/>
    </vehicle>
    <vehicle id="2" depart="2.00">
        <route edges="--15295 -15285#5 -15301#0"/>
    </vehicle>
    <vehicle id="3" depart="3.00">
        <route edges="-15301#0 --15301#0 --15285#5 --15285#4 --15285#3 --15285#2 --15285#1 --15285#0 --15281#3"/>
    </vehicle>
    <vehicle id="4" depart="4.00">
        <route edges="-15279 -15281#1 -15281#2 -15281#3 -15285#0 --15289"/>
    </vehicle>
    <vehicle id="5" depart="5.00">
        <route edges="-15301#0 --15301#0 --15285#5 --15285#4 --15285#3 --15285#2 --15285#1 --15285#0 --15281#3 --15281#2 --15281#1 --15279"/>
    </vehicle>
    <vehicle id="6" depart="6.00">
        <route edges="-15285#0 --15289"/>
    </vehicle>
    <vehicle id="7" depart="7.00">
        <route edges="--15283 -15281#2 -15281#3 -15285#0 -15285#1 -15285#2 -15285#3 --15285#3"/>
    </vehicle>
    <vehicle id="8" depart="8.00">
        <route edges="--15319 --15301#3 --15301#2 --15301#1 --15301#0"/>
    </vehicle>
    <vehicle id="9" depart="9.00">
        <route edges="-15301#0 -15301#1 -15301#2 -15301#3"/>
    </vehicle>
    <vehicle id="10" depart="10.00">
```

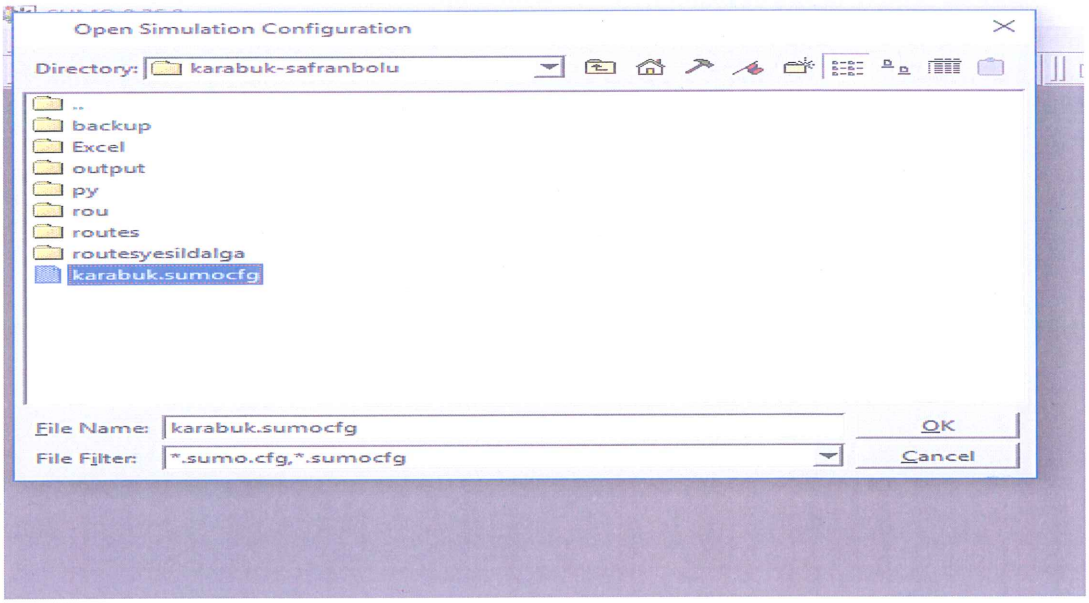
Şekil 2.8. karabuk1.1.xml dosyasının görüntüsü.

2.8. SUMO TRAFİK SİMÜLASYON PROGRAMININ KULLANILMASI

Oluşturulan rotalar Şekil 2.8'de gösterildiği gibi karabuk.sumocfg isimli ana dosyaya yazılır. Oluşturulan karabuk.sumocfg isimli dosya SUMO trafik simülasyon programı tarafından çalıştırılmaktadır(Şekil2.9).

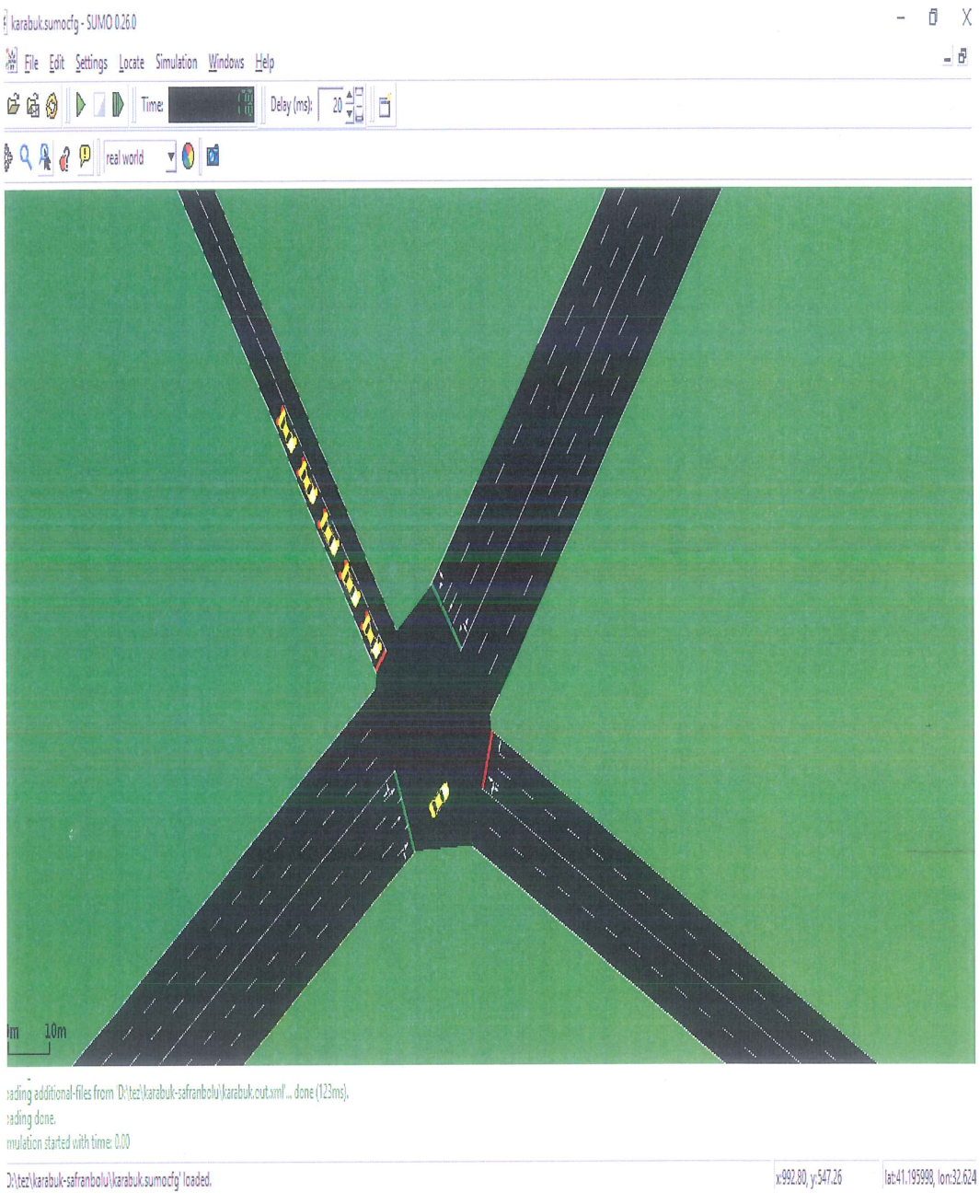
```
karabuk11rou.xml | karabuk.sumocfg
1  <?xml version="1.0" encoding="iso-8859-1"?>
2  <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">
3
4      <input>
5          <net-file value="karabuk.net.xml"/>
6      <!--
7
8
9      -->
10         <route-files value="routes/karabuk65.rou.xml"/>
11         <additional-files value="karabuk.out.xml"/>
12     </input>
13     <time>
14         <begin value="0"/>
15         <end value="6000"/>
16     </time>
17     <output>
18         <summary value="output/summary.xml"/>
19         <netstate-dump value="output/dump.xml" />
20     </output>
21     <report>
22         <no-warnings value="true"/>
23     </report>
24     <!-- Matlabtan yönetmek istediğimiz zaman  bu port açılacak
25         <traci_server>
26         <remote-port value="8813" />
27     </traci_server>
28      -->
29     <time-to-teleport value="-1" />
30 </configuration>
```

Şekil 2.9. karabuk.sumocfg dosyası.



Şekil 2.10. karabuk.sumocfg dosyasının SUMO programında çalıştırılması.

Uygun rotalar ve harita SUMO trafik simülasyon programına yüklendikten sonra gerçek dünya ortamında istenilen hızda simüle edilerek Şekil 2.10'da gösterildiği gibi test edilebilmektedir.

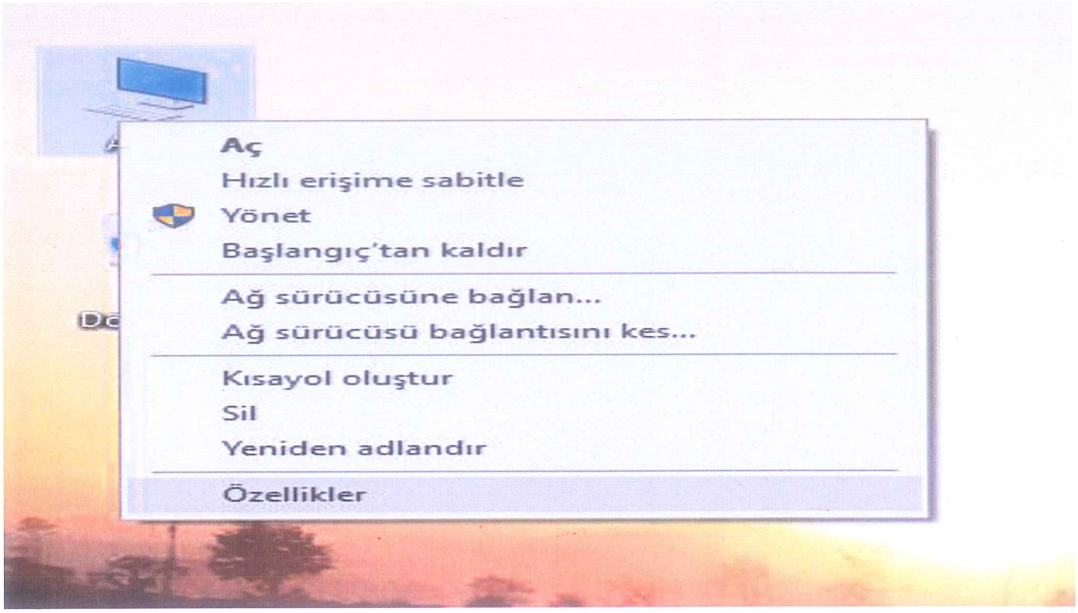


Şekil 2.11. SUMO trafik simülasyon programı çalışma ortamı.

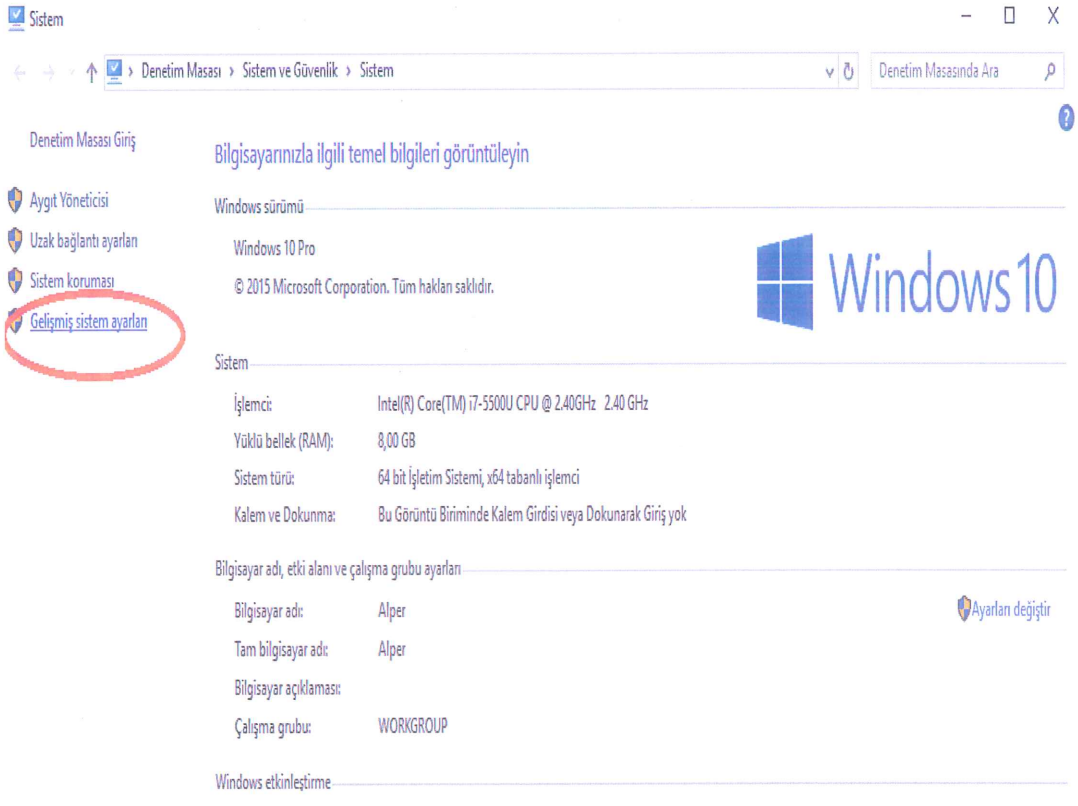
2.9. SUMO VE MATLAB PROGRAMLARI ARASINDAKİ İLETİŞİMİN SAĞLANMASI

SUMO trafik simülasyon programının MATLAB yazılım dili ile iletişim kurması, kodlama yapılarak istenilen trafik ışık optimizasyon metodunun kullanması ve test edilmesi sağlanabilmektedir. Bahsedilen iletişimin sağlanması için TraCIforMatlab programının kullanılması gerekmektedir. TraCI programı, Matlab yazılım dili ile SUMO şehirsel trafik simülasyon programı arasında ki iletişimi sağlayan bir trafik kontrol ara yüzüdür. TraCI, trafik ışıkları, araçlar, kavşaklar gibi SUMO nesnelerinin kontrolünü sağlamaktadır. Kurulumu için ilk olarak TraCI4Matlab programı indirilmelidir. İndirme işlemi tamamlandıktan sonra bilgisayarım linkine farenin sağ tuşu ile tıklanılmalı ve özellikler seçilmelidir (Şekil 2.11). Özellikler kısımda gelişmiş sistem ayarları linkine girilmeli (Şekil 2.12) ve ortam değişkeni alanına tıklanılmalıdır (Şekil 2.13). Açılan pencerede yeni butonu seçilerek, değişken adı yerine SUMO_HOME ve değişken değeri yerine SUMO trafik simülasyon programının bilgisayara kurulduğu yol yazılmalıdır (Şekil 2.14). Bahsedilen bu işlemler tamamlandığında TraCIforMatlab programının kurulum işlemi tamamlanmış olmaktadır.

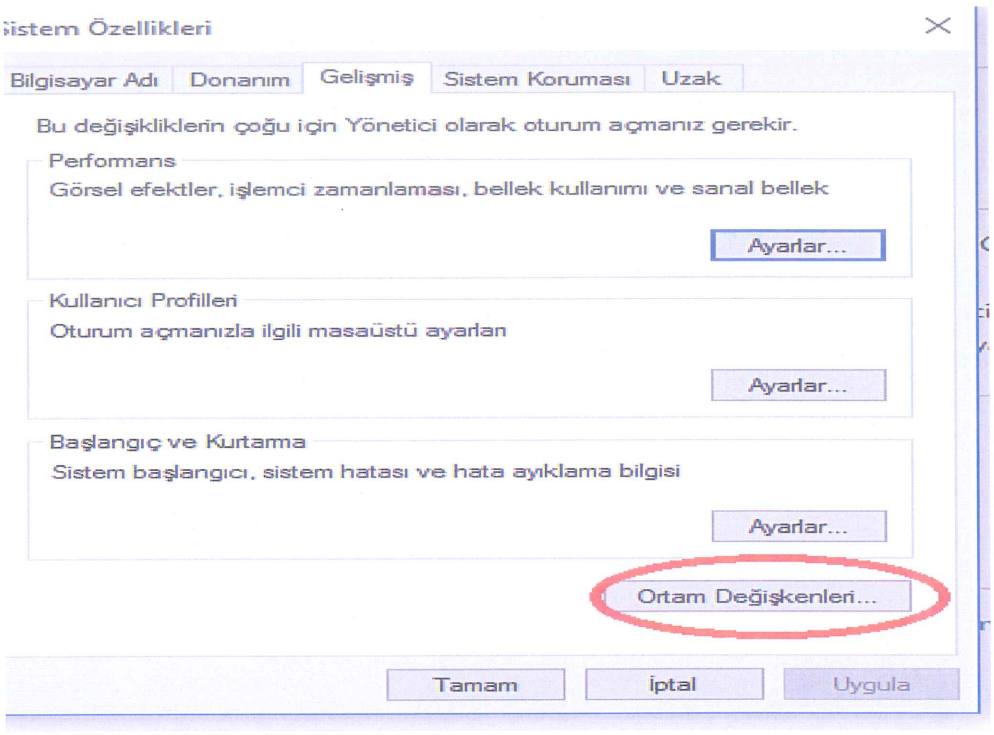
Traci ile Matlab iletişimin sağlanması için Matlab program ayarlarının da yapılması gerekmektedir. İlk olarak Matlab programı açılır, özellikler sekmesinden Add with subfolders alanı seçilerek TraCIforMatlab işaretlenir ve kaydedilir (Şekil 2.15). Son olarak ise SUMO trafik simülasyon programı ile Matlab arasında ki iletişimin sağlanması için karabuk.sumocfg isimli ayar dosyası içinde ki 8813 numaralı port açılmalıdır (Şekil 2.16).



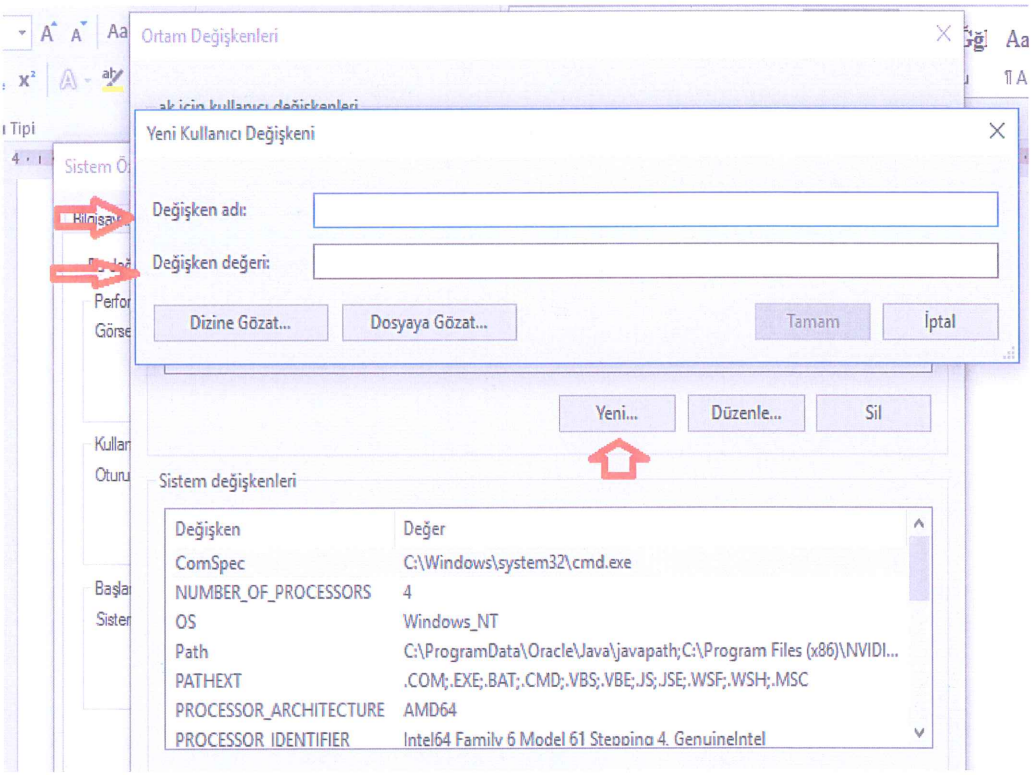
Şekil 2.12. Bilgisayarım'dan özelliklerin seçilmesi.



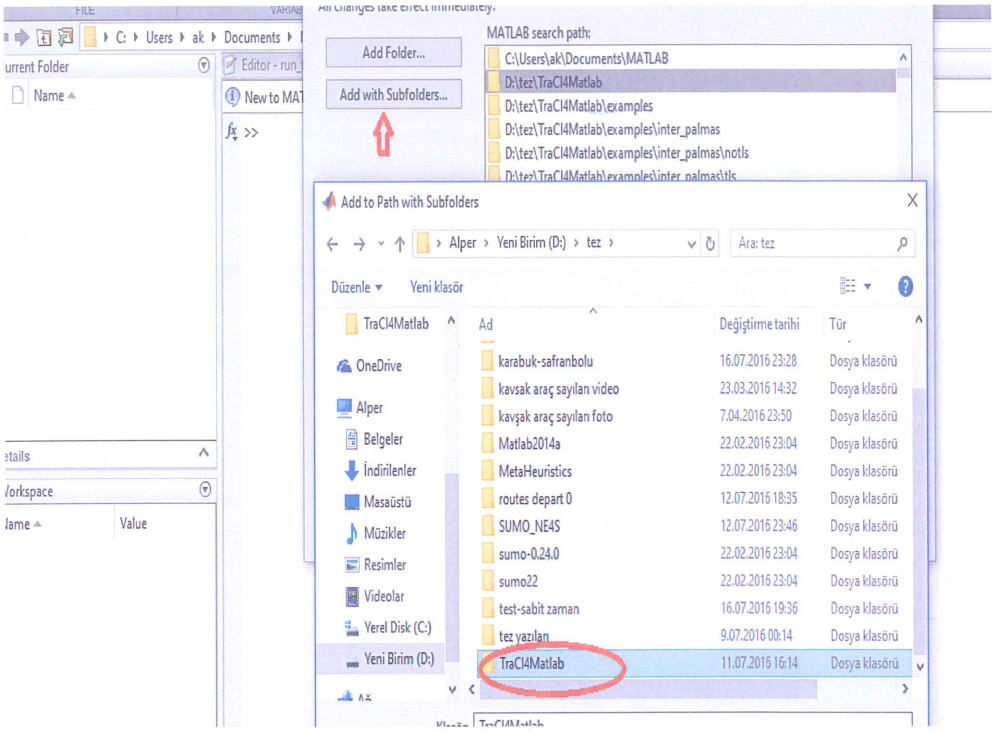
Şekil 2.13. Gelişmiş sistem ayarlarının seçilmesi.



Şekil 2.14. Ortam değişkenleri alanının seçilmesi.



Şekil 2.15. SUMO trafik simülasyon programının adının ve yolunun tanıtılması.



Şekil 2.16. Matlab programı içerisinde TraCI4Matlab programının tanıtılması.

karabuk.sumocfg

```
<?xml version="1.0" encoding="iso-8859-1"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">

    <input>
      <net-file value="karabuk.net.xml"/>
    <!--


    -->
      <route-files value="routes/karabuk65.rou.xml"/>
      <additional-files value="karabuk.out.xml"/>
  </input>
  <time>
    <begin value="0"/>
    <end value="6000"/>
  </time>
  <output>
    <summary value="output/summary.xml"/>
    <netstate-dump value="output/dump.xml" />
  </output>
  <report>
    <no-warnings value="true"/>
  </report>
  <!-- Matlabtan yönetmek istediğimiz zaman  bu port açılacak
    <traci_server>
    <remote-port value="8813" />
  </traci_server>
    -->
```

Şekil 2.17. SUMO trafik simülasyon programının config ayar dosyasında iletişim portunun açılması.

BÖLÜM 3

BULGULAR

Hazırlanan çalışma ortamında, ihtiyaç duyulan verilerin yüklenmesi ve programların kurulmasının ardından sabit zamanlı trafik ışık sistemleri, yeşil dalga trafik ışık sistemleri ve gerçek zamanlı trafik ışık sistemleri Çizelge 2.2'de gösterildiği gibi farklı yoğunluk ve farklı araç sayıları ile SUMO trafik simülasyon programında gerçek dünya ortamında ayrı ayrı test edilmiş ve elde edilen test sonuçları ayrıntılı olarak çizelgelerde sunulmuştur.

3.1. HESAPLAMA METRİKLERİ

Sistemde her bir araç için ortalama hız, ortalama bekleme süreleri ve ortalama seyahat süreleri hesaplanmıştır.

Ortalama hız ağdaki her bir aracın seyahat süresi boyunca, ortalama hızlarını ve verilerini referans almaktadır. Simülasyon sırasında her araca her saniye için bir hız değeri atanmaktadır. Seyahat süresine araçların her durması eklenmez. Araçların durakların da harcanan her saniye için 0 değeri atanmaktadır. Ortalama hız, ağdaki her bir aracın ortalama hızına, ortalama hızın, ağda seyahat eden araçların toplamının bölünmesi eklenerek bulunur. Ortalama hız değeri ne kadar yüksekse sistemin verimliliği o kadar iyidir. Ortalama araç hızı $\sum$ ($\sum$ (araç hızı) / araç seyahat süresi) / toplam araç sayısı formülü ile bulunabilir.

Araç seyahat süresi SUMO trafik simülasyon programında her bir aracın rotasını tamamladığı saniyeyi ifade etmektedir. Ortalama seyahat süresi her bir aracın seyahat süreleri toplamının, toplam araç sayısına bölünmesi ile bulunur. Ortalama seyahat süresi $\sum$ (araç seyahat süresi) / $\sum$araç sayısı formülü ile bulunabilmektedir.

Bekleme süresi SUMO trafik simülasyon programında her bir aracın kavşaktaki kırmızı ışık kuyruğunda hareketsiz olarak geçirdiği zamandır. Ortalama bekleme süresi her bir aracın bekleme sürelerinin toplamının, araç sayısına bölünmesi ile bulunur. Ortalama bekleme süresi $\sum$ (araç bekleme süresi) / $\sum$araç sayısı formülü ile bulunabilmektedir.

3.2. SABİT ZAMANLI TRAFİK IŞIK SİSTEMİ TEST SONUÇLARI

Sabit zamanlı trafik ışık sisteminde 6 farklı yoğunluk ve her bir yoğunluk için 5 adet birbirinden farklı rastgele oluşturulan rotlardan elde edilen test sonuçları, Çizelge 3.1, Çizelge 3.2, Çizelge 3.3, Çizelge 3.4, Çizelge 3.5, Çizelge 3.6'da verilmiştir. Elde edilen test sonuçlarının yoğunluklara göre ortalama değerleri Çizelge 3.7'de verilmiştir.

Çizelge 3.1. Yoğunluk 1 için test sonuçları.

Yoğunluk 1	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
Rota1	1002	1000	652	348	0,7	293,9
Rota2	1002	1000	655	345	0,71	303,87
Rota3	1002	1000	631	369	0,67	299,14
Rota4	1002	1000	656	344	0,74	298,67
Rota5	1002	1000	652	348	0,57	291,61
Ortalama	**1002**	**1000**	**649,2**	**350,8**	**0,678**	**297,438**

Çizelge 3.2. Yoğunluk 2 için test sonuçları.

Yoğunluk 2	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	2003	2000	1655	345	0,8	326,57
Rota2	2003	2000	1660	340	0,76	324,48
Rota3	2003	1999	1644	355	0,86	326,02
Rota4	2003	2000	1656	344	0,76	320,95
Rota5	2003	1997	1646	351	0,81	324,42
Ortalama	**2003**	**1999,2**	**1652,2**	**347**	**0,798**	**324,488**

Çizelge 3.3. Yoğunluk 3 için test sonuçları.

Yoğunluk 3	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	3005	3000	2621	379	0,81	335,84
Rota2	3005	2999	2664	335	0,9	333,35
Rota3	3005	3000	2644	356	0,79	329,76
Rota4	3005	3000	2655	345	0,83	332,84
Rota5	3005	3000	2622	378	0,87	335,54
Ortalama	**3005**	**2999,8**	**2641,2**	**358,6**	**0,84**	**333,466**

Çizelge 3.4. Yoğunluk 4 test sonuçları.

Yoğunluk 4	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	4007	4000	3662	338	0,78	335,15
Rota2	4007	4000	3670	330	0,9	337,59
Rota3	4007	4000	3646	354	0,88	340,67
Rota4	4007	4000	3654	346	0,93	337,95
Rota5	4007	3999	3652	347	0,83	339,8
Ortalama	**4007**	**3999,8**	**3656,8**	**343**	**0,864**	**338,232**

Çizelge 3.5. Yoğunluk 5 test sonuçları.

Yoğunluk 5	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	5009	4999	4652	347	0,89	336,64
Rota2	5009	4999	4664	335	0,82	334,88
Rota3	5009	4999	4656	343	0,92	338,52
Rota4	5009	4999	4660	339	0,84	336,07
Rota5	5009	5000	4655	345	0,85	340,76
Ortalama	**5009**	**4999,2**	**4657,4**	**341,8**	**0,864**	**337,374**

Çizelge 3.6. Yoğunluk 6 test sonuçları.

Yoğunluk 6	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	6010	6000	5637	363	0,88	339,85
Rota2	6010	5999	5636	363	0,9	344,86
Rota3	6010	5999	5659	340	0,86	338,9
Rota4	6010	6000	5648	352	0,89	341,83
Rota5	6010	5999	5649	350	0,88	334,74
Ortalama	**6010**	**5999,4**	**5645,8**	**353,6**	**0,882**	**340,036**

Çizelge 3.7. Yoğunluklara göre test sonuçlarının ortalamaları.

Yoğunluk No	Yoğunluk Oranı (0-1)	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
1	0,15	1002	1000	649,2	350,8	0,678	297,438
2	0,30	2003	1999,2	1652,2	347	0,798	324,488
3	0,45	3005	2999,8	2641,2	358,6	0,84	333,466
4	0,60	4007	3999,8	3656,8	343	0,864	338,232
5	0,75	5009	4999,2	4657,4	341,8	0,864	337,374
6	0,90	6010	5999,4	5645,8	353,6	0,882	340,036

3.3. YEŞİL DALGA TRAFİK IŞIK SİSTEMİ TEST SONUÇLARI

Yeşil dalga trafik ışık sisteminde 6 farklı yoğunluk ve her bir yoğunluk için 5 adet birbirinden farklı rastgele oluşturulan rotlardan elde edilen test sonuçları, Çizelge 3.8, Çizelge 3.9, Çizelge 3.10, Çizelge 3.11, Çizelge 3.12, Çizelge 3.13'de

verilmiştir. Elde edilen test sonuçlarının yoğunluklara göre ortalama değerleri Çizelge 3.14'de verilmiştir.

Çizelge 3.8. Yoğunluk 1 için test sonuçları.

Yoğunluk 1	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
Rota1	1002	1000	730	270	0,5	249,01
Rota2	1002	1000	732	268	0,6	252,256
Rota3	1002	1000	699	301	0,529	250,865
Rota4	1002	1000	722	278	0,601	249,458
Rota5	1002	1000	715	285	0,4125	244,256
Ortalama	**1002**	**1000**	**701,82**	**298,18**	**0,5763**	**252,8223**

Çizelge 3.9. Yoğunluk 2 için test sonuçları.

Yoğunluk 2	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	2003	2000	1700	285	0,61	273,456
Rota2	2003	2000	1698	281	0,623	272,4521
Rota3	2003	1999	1714	285	0,7	272,456
Rota4	2003	2000	1714	286	0,635	270,8965
Rota5	2003	1997	1706	291	0,6589	273,021
Ortalama	**2003**	**1999,2**	**1704,25**	**294,95**	**0,6783**	**275,8148**

Çizelge 3.10. Yoğunluk 3 için test sonuçları.

Yoğunluk 3	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	3005	3000	2720	280	0,595	254,26
Rota2	3005	2999	2748	251	0,7124	251,155
Rota3	3005	3000	2732	268	0,612	248,521
Rota4	3005	3000	2752	248	0,6325	250,58
Rota5	3005	3000	2707	293	0,6578	235,57
Ortalama	**3005**	**2999,8**	**2731,8**	**268**	**0,64194**	**248,0172**

Çizelge 3.11. Yoğunluk 4 için test sonuçları.

Yoğunluk 4	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	4007	4000	3745	255	0,589	249,45
Rota2	4007	4000	3765	235	0,6985	251,985
Rota3	4007	4000	3751	249	0,6895	257,985
Rota4	4007	4000	3787	213	0,7198	253,489
Rota5	4007	3999	3723	276	0,6385	251,589
Ortalama	**4007**	**3999,8**	**3754,2**	**245,6**	**0,66706**	**252,8996**

Çizelge 3.12. Yoğunluk 5 için test sonuçları.

Yoğunluk 5	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	5009	4999	4704	295	0,7565	286,144
Rota2	5009	4999	4714	285	0,697	284,648
Rota3	5009	4999	4707	292	0,782	287,742
Rota4	5009	4999	4711	288	0,714	285,6595
Rota5	5009	5000	4707	293	0,7225	289,646
Ortalama	**5009**	**4999,2**	**4708,67**	**290,53**	**0,7344**	**286,7679**

Çizelge 3.13. Yoğunluk 6 için test sonuçları.

Yoğunluk 6	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	6010	6000	5691	309	0,748	288,8725
Rota2	6010	5999	5690	309	0,765	293,131
Rota3	6010	5999	5710	289	0,731	288,065
Rota4	6010	6000	5701	299	0,7565	290,5555
Rota5	6010	5999	5702	298	0,748	284,529
Ortalama	**6010**	**5999,4**	**5698,84**	**300,56**	**0,7497**	**289,0306**

Çizelge 3.14. Yoğunluklara göre test sonuçlarının ortalamaları.

Yoğunluk No	Yoğunluk Oranı (0-1)	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
1	0,15	1002	1000	719,6	280,4	0,5285	249,169
2	0,30	2003	1999,2	1706,4	285,6	0,64538	272,45632
3	0,45	3005	2999,8	2731,8	268	0,64194	248,0172
4	0,60	4007	3999,8	3754,2	245,6	0,66706	252,8996
5	0,75	5009	4999,2	4708,67	290,53	0,7344	286,7679
6	0,90	6010	5999,4	5698,84	300,56	0,7497	289,0306

3.4. GERÇEK ZAMANLI TRAFİK IŞIK SİSTEMİ TEST SONUÇLARI

Gerçek zamanlı trafik ışık sisteminde 6 farklı yoğunluk ve her bir yoğunluk için 5 adet birbirinden farklı rastgele oluşturulan rotlardan elde edilen test sonuçları, Çizelge 3.15, Çizelge 3.16, Çizelge 3.17, Çizelge 3.18, Çizelge 3.19, Çizelge 3.20'de verilmiştir. Elde edilen test sonuçlarının yoğunluklara göre ortalama değerleri Çizelge 3.21'de verilmiştir.

Çizelge 3.15. Yoğunluk 1 için test sonuçları.

Yoğunluk 1	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
Rota1	1002	1000	756	244	0,49	205,73
Rota2	1002	1000	759,00	241,00	0,497	212,709
Rota3	1002	1000	742	258	0,469	209,398
Rota4	1002	1000	759	241	0,518	209,069
Rota5	1002	1000	756	244	0,399	204,127
Ortalama	**1002**	**1000**	**754,54**	**245,46**	**0,4746**	**208,2066**

Çizelge 3.16. Yoğunluk 2 için test sonuçları.

Yoğunluk 2	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	2003	2000	1759,00	241,00	0,56	228,599
Rota2	2003	2000	1762	238	0,532	227,136
Rota3	2003	1999	1751	249	0,602	228,214
Rota4	2003	2000	1759	241	0,532	224,665
Rota5	2003	1997	1751	246	0,567	227,094
Ortalama	**2003**	**1999,2**	**1756,4**	**242,8**	**0,5586**	**227,1416**

Çizelge 3.17. Yoğunluk 3 için test sonuçları.

Yoğunluk 3	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	3005	3000	2755	245	0,521	228,78
Rota2	3005	2999	2776	223	0,534	224,785
Rota3	3005	3000	2768	232	0,4985	219,875
Rota4	3005	3000	2758,00	231,00	0,5068	224,563
Rota5	3005	3000	2749	251	0,534	219,78
Ortalama	**3005**	**2999,8**	**2761,2**	**236,4**	**0,51886**	**223,5566**

Çizelge 3.18. Yoğunluk 4 için test sonuçları.

Yoğunluk 4	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	4007	4000	3787	213	0,41	222,456
Rota2	4007	4000	3786	214	0,4895	219,456
Rota3	4007	4000	3779	221	0,4576	213,458
Rota4	4007	4000	3774	226	0,512	212,856
Rota5	4007	3999	3769	230	0,423	211,458
Ortalama	**4007**	**3999,8**	**3779**	**220,8**	**0,45842**	**215,9368**

Çizelge 3.19. Yoğunluk 5 için test sonuçları.

Yoğunluk 5	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	5009	4999	4756	243	0,623	235,648
Rota2	5009	4999	4765,00	234,00	0,574	234,416
Rota3	5009	4999	4759	240	0,644	236,964
Rota4	5009	4999	4762	237	0,588	235,249
Rota5	5009	5000	4758,50	241,50	0,595	238,532
Ortalama	**5009**	**4999,2**	**4760,04**	**239,16**	**0,6048**	**236,1618**

Çizelge 3.20. Yoğunluk 6 için test sonuçları.

Yoğunluk 6	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi
Rota1	6010	6000	5746	254	0,616	237,895
Rota2	6010	5999	5745	254	0,63	241,402
Rota3	6010	5999	5761	238	0,602	237,23
Rota4	6010	6000	5754	246	0,623	239,281
Rota5	6010	5999	5754	245	0,616	234,318
Ortalama	**6010**	**5999,4**	**5751,88**	**247,52**	**0,6174**	**238,0252**

Çizelge 3.21. Yoğunluklara göre test sonuçlarının ortalamaları.

Yoğunluk No	Yoğunluk Oranı (0-1)	Sisteme Yüklenen Araç Sayısı	Sisteme Giren Araç Sayısı	Sistemden Çıkan Araç Sayısı	Sistemde Kalan Araç Sayısı	Ortalama Bekleme Süresi	Ortalama Seyahat Süresi (Saniye)
1	0,15	1002	1000	754,54	245,46	0,4746	208,2066
2	0,30	2003	1999,2	1756,4	242,8	0,5586	227,1416
3	0,45	3005	2999,8	2761,2	236,4	0,51886	223,5566
4	0,60	4007	3999,8	3779	220,8	0,45842	215,9368
5	0,75	5009	4999,2	4760,04	239,16	0,6048	236,1618
6	0,90	6010	5999,4	5751,88	247,52	0,6174	238,0252

3.5. TRAFİK IŞIK SİNYAL OPTİMİZASYON SİSTEMLERİNİN KARŞILAŞTIRILMASI

Önceki bölümlerde verilen çizelgelerde ki test sonuçlarına göre sabit zamanlı trafik ışık sistemi, yeşil dalga trafik ışık sistemi ve gerçek zamanlı trafik ışık sistemi için

her bir yoğunluktaki karşılaştırmalar Çizelge 3.22, Çizelge 3.23, Çizelge 3.24, Çizelge 3.25 ve Şekil 3.1, Şekil 3.2, Şekil 3.3 ve Şekil 3.4'de verilmiştir.

Çizelge 3.22. Sistemden çıkan araç sayıları karşılaştırılması.

Yoğunluk No	Yoğunluk Oranı	Sabit Zaman	Yeşil Dalga	Gerçek Zaman
1	0,15	649,2	719,6	754,44
2	0,3	1652,2	1706,4	1756,3
3	0,45	2641,2	2731,8	2748,78
4	0,6	3656,8	3754,2	3759,7
5	0,75	4657,4	4708,67	4759,94
6	0,9	5645,8	5698,84	5751,88

Çizelge 3.23. Sistemde kalan araç sayıları karşılaştırılması.

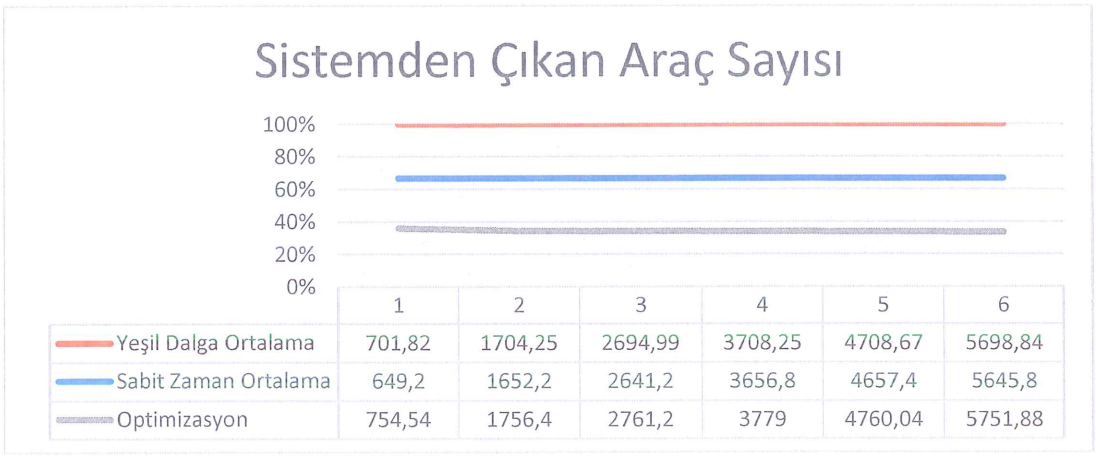
Yoğunluk No	Yoğunluk Oranı	Sabit Zaman	Yeşil Dalga	Gerçek Zaman
1	0,15	350,8	280,4	245,56
2	0,3	347	285,6	242,9
3	0,45	358,6	268	251,02
4	0,6	343	245,6	240,1
5	0,75	341,8	290,53	239,26
6	0,9	353,6	300,56	247,52

Çizelge 3.24. Ortalama bekleme süreleri karşılaştırılması.

Yoğunluk No	Yoğunluk Oranı	Sabit Zaman	Yeşil Dalga	Gerçek Zaman
1	0,15	0,68	0,5763	0,4746
2	0,3	0,798	0,6783	0,5586
3	0,45	0,84	0,714	0,51886
4	0,6	0,864	0,7344	0,45842
5	0,75	0,864	0,7344	0,6048
6	0,9	0,882	0,7497	0,6174

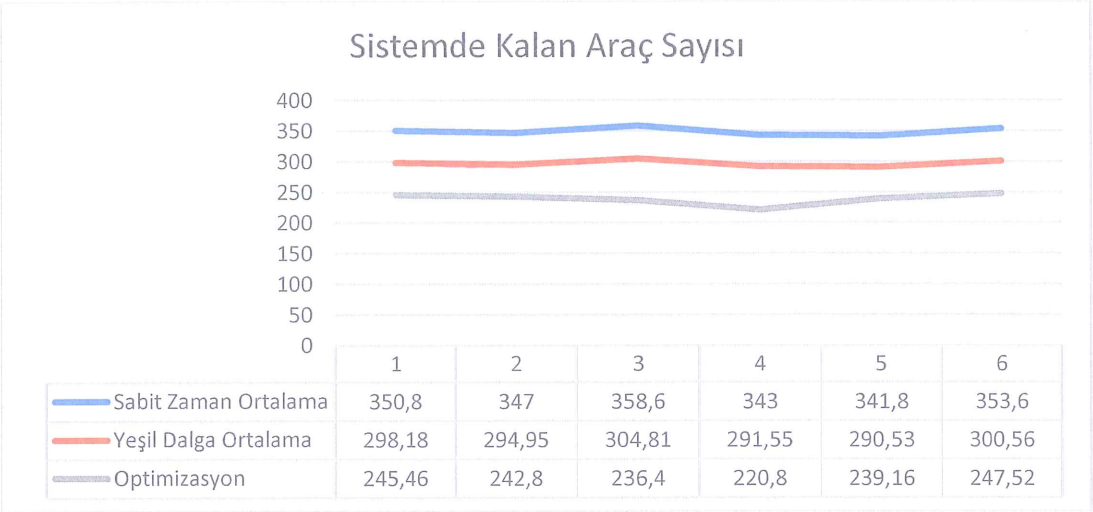
Çizelge 3.25. Ortalama seyahat süresi karşılaştırılması.

Yoğunluk No	Yoğunluk Oranı	Sabit Zaman	Yeşil Dalga	Gerçek Zaman
1	0,15	297,44	252,8223	208,2066
2	0,3	324,488	275,8148	227,1416
3	0,45	333,466	283,4461	223,5566
4	0,6	338,232	287,4972	221,2172
5	0,75	337,374	286,7679	236,1618
6	0,9	340,036	289,0306	238,0252



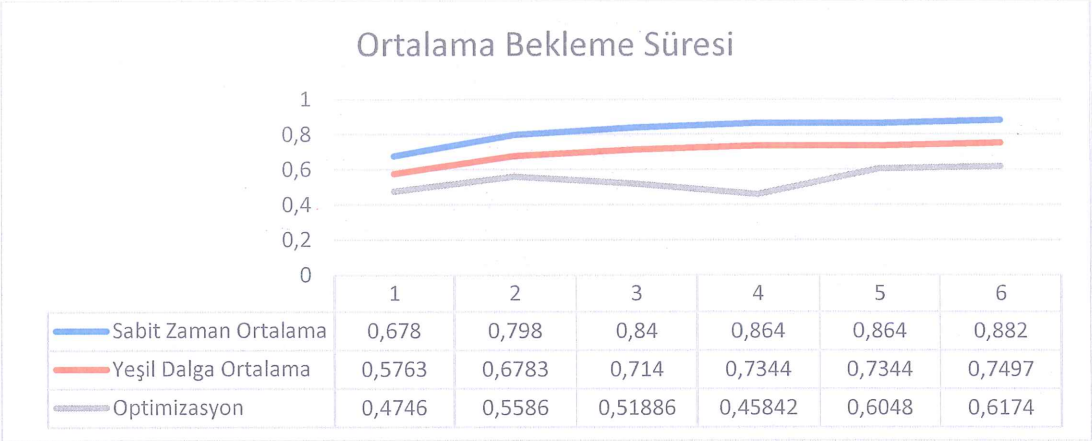
	1	2	3	4	5	6
Yeşil Dalga Ortalama	701,82	1704,25	2694,99	3708,25	4708,67	5698,84
Sabit Zaman Ortalama	649,2	1652,2	2641,2	3656,8	4657,4	5645,8
Optimizasyon	754,54	1756,4	2761,2	3779	4760,04	5751,88

Şekil 3.1. Sistemden çıkan araç sayılarının karşılaştırılması.



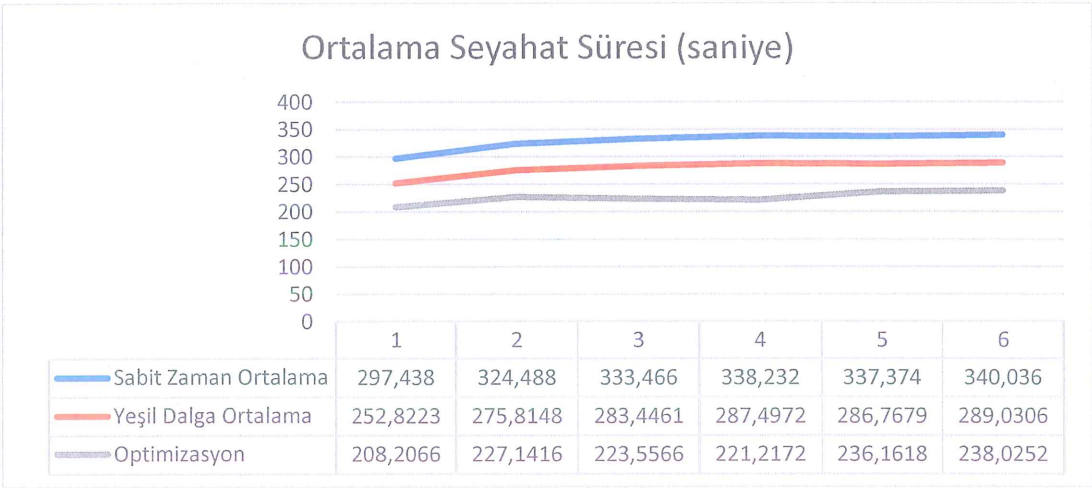
	1	2	3	4	5	6
Sabit Zaman Ortalama	350,8	347	358,6	343	341,8	353,6
Yeşil Dalga Ortalama	298,18	294,95	304,81	291,55	290,53	300,56
Optimizasyon	245,46	242,8	236,4	220,8	239,16	247,52

Şekil 3.2. Sistemde kalan araç sayılarının karşılaştırılması.



	1	2	3	4	5	6
Sabit Zaman Ortalama	0,678	0,798	0,84	0,864	0,864	0,882
Yeşil Dalga Ortalama	0,5763	0,6783	0,714	0,7344	0,7344	0,7497
Optimizasyon	0,4746	0,5586	0,51886	0,45842	0,6048	0,6174

Şekil 3.3. Ortalama bekleme sürelerinin karşılaştırılması.



	1	2	3	4	5	6
Sabit Zaman Ortalama	297,438	324,488	333,466	338,232	337,374	340,036
Yeşil Dalga Ortalama	252,8223	275,8148	283,4461	287,4972	286,7679	289,0306
Optimizasyon	208,2066	227,1416	223,5566	221,2172	236,1618	238,0252

Şekil 3.4. Ortalama seyahat sürelerinin karşılaştırılması.

BÖLÜM 4

SONUÇLAR VE ÖNERİLER

4.1. SONUÇLAR

Seçilen trafik ışık sinyalizasyon sistemlerinin trafik verimi üzerinde oldukça önemli bir etkisi bulunmaktadır. Gerçekleştirdiğimiz simülasyon testlerinde gözlemlediğimiz sonuçlara göre her yoğunluk için trafik ağlarındaki sinyalize kavşaklarda gerçek zamanlı trafik ışık sistemlerinin sabit zamanlı trafik ışık sistemleri ve yeşil dalga trafik ışık sistemlerinden daha iyi sonuç verdiği görülmektedir. Yeşil dalga trafik ışık sisteminin ise sabit zamanlı trafik ışık sisteminden daha iyi sonuç verdiği görülmektedir. Ancak yoğunluğun az olduğu yada çok fazla olduğu durumlarda trafikte kullanılan optimizasyon sisteminin başarısı çok fazla görülmemektedir. Optimizasyon sisteminin başarısı trafiğin yoğun olduğu ancak akışın sağlanabildiği durumlarda oldukça fazla görülmektedir. Bu sonuçlar ışığında gerçek zamanlı trafik ışık optimizasyon sistemlerinin trafik ağlarında kullanılması, trafik sıkışıklığının diğer sistemlere oranla daha az olacaktır.

Gerçek zamanlı trafik ışık optimizasyon sisteminin önemli avantajlarından birisi, kavşaklarda ki her bir yönde ki ışıkların süreleri araç yoğunluklarına göre değiştirilebilmektedir. Bu uygulama sayesinde kavşaklardaki araç yoğunluğunun en fazla olduğu yöndeki trafik akışına daha fazla süre verilerek, kavşaktaki trafik sıkışıklarının azalması sağlanmış olmaktadır.

Ayrıca gerçek zamanlı trafik optimizasyon sistemleri kullanıldığında kullanılan bütün yoğunluklarda araçların hız ortalamaları daha yüksek olmakta, sistemde ki bekleme süreleri azalmakta, sistemden çıkan araç sayısı artmakta ve sistemde kalan araç sayısı azalmaktadır. Tüm bu değişkenler göz önünde bulundurulduğunda harcanan yakıt miktarının, sera gazı etkisinin ve trafikte harcanan zamanın da önemli

ölçüde azalacağı söylenebilmektedir. Yeşil dalga trafik ışık sistemi, sabit zamanlı trafik ışık sistemine oranla trafik sıkışıklığını azaltabilecek bir çözümdür ancak kavşaklardaki yönlerin artması durumunda yan yollardan ana yola katılacak olan araçların bekleme sürelerinin artmasına sebep olabilmektedir.

Gerçekleştirilen simülasyon ortamında ki testlerde başlangıç noktasından sisteme katılacak olan araçların sayısının gerçek zamanlı trafik ışık sistemlerinde diğer sistemlere oranla oldukça fazla olduğu görülmektedir.

Bu başarının sebebi trafiğin anlık durumuna bağlı olarak sistemin güncellenmesidir. Anlık durum göz önüne alınarak yapılan sistemlerde, trafik kavşaklarındaki ışıklarda bekleyen araç sayısının azaldığı ve trafik sıkışıklığının azaldığı gözlemlenmektedir.

4.2. ÖNERİLER

Oluşturulacak trafik sistemlerinde gerçek zamanlı trafik ışık optimizasyon sistemlerinin kullanılması, trafik sıkışıklığını en aza indirebilmektedir. Maliyetleri de göz önüne alarak, kavşakta bulunan her bir bağlantı noktası için uygun konumlara yerleştirilen sensorlar aracılığı ile,araç sayıları veya kuyruk uzunlukları hesaplanarak, trafiğin yoğun olduğu hat üzerinden anlık olarak sinyal zaman planı yapmak trafik akışını önemli ölçüde düzeltecektir.

Ana hat üzerinde trafik akışının yoğun olduğu ve yan yol sayısının az olduğu bölgelerde, trafik akışının kesintisiz sürmesi için yeşil dalga trafik ışık sistemi kullanılabilir. Gerçek zamanlı trafik ışık optimizasyon sistemi kadar olmasa da yeşil dalga yöntemi de bahsedilen koşullarda trafik akışını olumlu yönde etkileyebilmektedir.

KAYNAKLAR

[1] İnternet: Türkiye İstatistik Kurumu, "Karayolları Raporları", **http://www.tuik.gov.tr/KarayluBultenleri.ci?id=116**(2016).

[2] İnternet:, "Habertürk",**http://www.haberturk.com/ekonomi/ekonomi/haber/1044155-istanbullular-her-gun-trafikte-bekleyerek-23-milyon-lira-harciyor** (2015).

[3] W. Xiaoping, D. Shuai, D. Xiaohong ve M. Jing, "Green-Wave Traffic Theory Optimization",***Scientific Research,*** 14-19(2014).

[4] Bono ve D. Edward, "İllustrated History of İnventions from the wheel to the computer", London: **Thomes ve Hudson ltd**(1974).

[5] P. M, X. Z., Pengfei Li a, "Solving simultaneous route guidance and traffic signal",***Transportation Research,*** *(*2015).

[6] A. Akbaş ve E. Akdoğan, "İstanbul Kent İç Trafik Kontrol Sistemi Üzerine Bir Durum Değerlendirmesi",**%1 içinde *Türkiye Makine Mühendisleri Odası "İstanbul'da Kent İçi Ulaşım Sempozyumu***, İstanbul(2001).

[7] D. Robertson ve P. Hunt, "Estimating the benefits of signal co-ordination using TRANSYT or SCOOT", ***ITE Conference***, London(1983).

[8] K. Tobita, T. Nagatani, "Green-Wave Control of an Unbalanced Two-Route Traffic System with Signals",***Physica A,*** 5422–5430(2013).

[9] S. Nigarnjanagool ve H. Dia, "Evaluation Of Dynamic Signal Optimization Control Model UsingTraffic Simulation"***The Computerizaiton Of Tranportation*** *(*2004).

[10] P. Hunt, D. Robertson, R. Bretherton ve R. Winton, "SCOOT: a traffic responsive method of coordinating signals"***Transport and Road Research Laboratory***, Crowthorne(1981).

[11] M. Bretherton, C. Bell ve R. D, "Ageing of Fixed Time Traffic Signal Plans",***Proceedings IEE Second International***, London (1986).

[12] Bretherton C, "SCOOT Demonstration Projects",***Municipality of Metropolitan Toronto***, Toronto(1993).

[13] R. Bretherton ve G. Bowen, "Recent Enhancements to SCOOT - SCOOT Version 2.4 Proceedings",***International Conference on Road Traffic Control,***

London(1990).

[14] K. Wood ve R. Baker, "Using SCOOT Weightings to Benefit Strategic Routes",***Traffic Engineering and Control***(1992).

[15] K. Wood, R. Baker, "User Guide to the Gating Method of Reducing Congestion in Traffic Networks",***Traffic Engineering and Control***(1995).

[16] Z. T. Yi Zhao, "An Overview of the Usage of Adaptive Signal Control System in the United States of America"***Trans Tech Publications***, 2591-2598(2012).

[17] I. Yang ve R. Jayakrishnan, "Real-Time Network-Wide Traffic Signal Optimization Considering Long-Term Green Ratios Based on Expected Route Flows",***Transportation Research Part C***, 241-257(2015).

[18] S. Sen, L. Head, "Controlled Optimization of Phases at an Intersection",***Transportation Science***(1997).

[19] P. Lowrie, "SCATS: The Sydney Co-ordinated Adaptive Traffic System",***IEE International Conference on Road Traffic Signaling***, London(1982).

[20] L.C, Liao, "A Review of the Optimization Policies for Adaptive Control Strategy (OPAC)",***California Path Working Paper University of Colifornia, Berkeley***(1998).

[21] P. Mirchandani, L. Head, "RHODES: A Real-Time Traffic Signal Control System", ***Architecture,Algoritms and Analysis. Transportation Research Part C***, 415-432(2001).

[22] R. Ghaman, "ACS lite. A Signal Timing Strategy for Closed Loop System",***Preseted at the 2007 ITE Annual Meeting***, Pittsburg,PA(2007).

[23] İnternet, Siemens "Research Project ITS Consulting Group Siemens Intelligent Transportation System"**http://www.mobility.siemens.com/mobility/global/en/urban-mobility/road-solutions/infrastructure-and-urban-traffic-control/pages/infrastructure-and-urban-traffic-control.aspx**(2000).

[24] M. Bando, K. Hasebe, A. Nakayama, A. Shibata ve Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation",***Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics,*** 1035-1042 (1995).

[25] Y. Ş. Murat, "Ulaştırma -Trafik Mühendisliğinde Yeni Yöntemler; Bulanık Mantık Tekniği Uygulamaları",***Türkiye Mühendislik Haberleri 439. sayı***(2004).

[26] F. Teklu, A. Sumalee, "A Genetic Algorithm Approach for Optimizing Traffic Control Signals Considering Routing",***Computer-Aided Civil and Infrastructure Engineering***, 31-43(2007).

[27] R. Bifulco, J. Rückert, M. Rizwan-Ul-Haq, H.-J. Kolbe ve D. Hausheer, "Flexible traffic management in broadband access networks using Software Defined Networking",***Network Operations and Management Symposium (NOMS)***(2014).

[28] S. Yagar, B. Han, "A Procedure For Real-Time Signal Control That Considers Transit Interference and Priorty",***Transpotation Research Part B***, 315-331 (1994).

[29] Y. Ş. MURAT, "Ulaştırma- Trafik Mühendisliğin de yeni yöntemler: Bulanık Mantık Tekniği Uygulamaları", ***TMH***(2004).

[30] A. T. Mohammad, I. Laheeb, "Traffic Simulation System based on Fuzzy Logic",***Procedia Computer Science***, 356-360(2012).

[31] C. Pappis, E. Mamdani, "A Fuzzy Logic Controller for a Traffic Junction Systemscs"***Man and Cyberneti***, 707-717 (1977).

[32] N. Nakatsuyama, M. Nagahashi, H. Nishizura, "Fuzzy Logic Controller for a Traflc Junction in the One-way Arterial Road",***9th PAC-World Congress***(1984).

[33] J. Niittymaki, "Installation and Experiences of Field Testing a Fuzzy Signal Controller", ***European Journal of Operational Research***, 273-281(2001).

[34] J.-i. Choi, S. Y. Lee, "Process Analysis and Economic Evaluation for Poly(3-hydroxybutyrate) Production by fermentation",***Bioprocess Engineering***, 335-342(1997).

[35] P. H. L. Bovy, E. Stern, "Route Choice: Wayfinding in Transport Networks",***Studies in Operational Regional Science***(1990).

[36] M. Emmering, H. Richard, P. Nijkamp, P. Rietveld, "Variable Message Sign and Radio Traffic Information: An Integrated Empirical Analysis of Drivers Route Choice Behavior",***Transportion Research Part A***, 135-153 (1996).

[37] O. A. Erdem, "Kavşak Trafik Sinaylizasyon Sistemi içinBulanık Mantık Tabanlı Gerçek Zamanlı Denetleyici Tasarımı ve Uygulanması",***e-journal of New World Sciences Academy***, 1306-3111(2007).

[38] A. Tzes, W. McShane, S. Kim, «Expert Fuzzy Logic Traffic Signal Control for Transportation Networks",***In Compendium of Technical Papers of the 65th Annual Meeting of the Institute of Transportation Engineers***, Denver(1995).

[39] X. R. W. Zheng, "An adaptive control algorithm for traffic-actuated signals",***Transport. Res. Part C***(2013).

[40] D. Zubillaga, L. D. A. J. Z. Geovany Cruz, "Measuring the Complexity of Self-Organizing Traffic Lights", ***Entropy*** (2014).

[41] V. Mauro, D. Taranto, "UTOPIA", ***Proceedings of the 6th IFAC/IFIP/IFORS Symposium on Control and Communication,*** Paris (1990).

[42] K. P. M. , K. E. Aboudolas, "Store-and-Forward Based Methods for the Signal Control Problem in Large-Scale Congested Urban Road Networks",***Transportation Research Part C,*** 163-174(2009).

[43] A. L. Bazzan, "Opportunities for Multiagent Systems and Multiagent Reinforcement Learning in Traffic Control",***Autonomous Agents and Multi-Agent Systems,*** 342-375(2009).

[44] B. Y. D, A. K. Park, "Stochastic Optimization for Sustainable Traffic Signal Control", ***International Journal of Sustainable Transportation,*** 263-284 (2009).

[45] D. G. Ganji, "Differential Transform Method for Mathematical Modeling of Jamming Transition Problem in Traffic Congestion Flow",***Central European Journal of Operations Research,*** 87-100(2012).

[46] H. Gardner, "Frames of Mind", ***New York: Basic Book Inc***(1983).

[47] L. S. Jones, M. D. Anderson, D. Malave, A. J. Sullivan, "Traffic Simulation Software Comparison Study",***The University of Alabama, The University of Alabama at Birmingham, and The University of Alabama***, Huntsville(2004).

[48] C. Ruedy, "The Smartest Project, of Review of Micro-Simulation Models Appendix", Leeds University, 1997.

[49] A. Catarella, B. Wolshon, N. Juneau, "A Method for the Simulation of Coordinated Leadlag Left-turn Phasing in CORSIM",***ITE Journal, Institute of Transportation Engineers***, Washington, DC(2001).

[50] L. E. Owen, Y. Zhang, L. Rao, G. McHale, "Traffic Follw Simulations Using CORSIM",***Proceedings of the 2000 Winter Simulation Conference,*** Orlando(2000).

[51] J. Hansen, M. Sato, R. Ruedy, A. Lacis, V. Oinas, "Global Warming in the Twenty-First Century: An Alternative Scenario",***Center for Climate Systems Research***(2000).

[52] J. Perrin, P. Martin, C. Jhaveri, "Interfacings SCOOT and CORSIM: Theree Salt Lake City Networks Modeled",**ITE 2002 Annual Meeting and Exhibit,**

Philadelphia(2002).

[53] S. I.-J. Chien, D. G. Goulias, S. Yahalom, M. Chowdhury, "Simulation-Based Estimates of Delays at Freeway Work Zones",***Journal of Advanced Transportation***, 131-156(2002).

[54] H. Zhou, J. Lu, X. Yang, S. Dissanayake, K. Williams, "Operational Effects of U-turns As Alternatives to Direct Left Turns From Driveways",***Transportation Research Record, Journal of Transportation Research***, 72-79, (2002).

[55] İnternet: M. Trans, "Corsim Product Information Available from McTrans",**http:// mctrans.ce.ufl.edu/store/description.asp?itemID=450** (2003).

[56] S. A. Boxill, Y. Lei, "An Evaluation of Traffic Simulation Models for Supporting ITS Development",***Center for Transportation Training and Research***(2000).

[57] İnternet:Siemens "Microskobik traffic simülation with Vissim"**http://www.mobility.siemens.com/mobility/global/en/urban-mobility/road-solutions/infrastructure-and-urban-traffic-control/pages/infrastructure-and-urban-traffic-control.aspx**(2012).

[58] J. Barceló, E. Codina, J. Casas, J. Ferrer, D. García, "Microscopic Traffic Simulation: A Tool for the Design, Analysis and Evaluation of Intelligent Transport Systems", ***Journal of Intelligent and Robotic Systems***, 173-203 (2005).

[59] A. Staffan, B. Eric, B. Marco, B. Laurent, T. C. Di, D. Mark, F. Ken, G. Jean-François, "Review of Micro-Simulation Models",***Simulation Modelling. Applied-to-Road-Trasnport European***(1997).

[60] İnternet: "lanl",**www.transims.tsasa.lanl.gov/documents-research98.html.**(1998).

[61] İnternet : "Hutsim Traffic Simulation Model"**www.hutif/units/Transportation/Research/hutsim.html**. (2000)

[62] S. Maerivoet, B. D. Moor, "Cellular Automata Models of Road Traffic", ***Siciense Direct Physics Reports***, 1-64(2005).

[63] G. H. Bham, R. F. Benekohal, "A High Fidelity Traffic Simulation Model Based on Cellular Automata and Car-Following Concepts",***Transportation Research Part C***, 1-32 (2004).

[64] J. Dijkstra, J. Jessurun, H. Timmermans, "A Multi-Agent Cellular Automata Model of Pedestrian Movement",***Pedestrian and Evacuation Dynamics***, 173-

181(2001).

[65] J. Luis, F. Pereira, "An Integrated Architecture for Autonomus Vehicles Simulation",***Faculdade de Engenharia da Universidada do Porto***(2011).

[66] İnternet : "SUMO User Documentation" SourceForge.net. (2010)

[67] J. L. Pereira, R. j. Rosetti, "Toward a cooperative traffic network editor",***In Proceeding of the 6th International Conference on Cooperative Design, Visualization and Engineering***, Berlin (2009).

[68] J. Macedo, G. Soares, Z. Kokkinogenis, D. Perrotta, J. F. Rosaldo, "Framework for Electric Bus Powertrain Simulation in Urban Mobility Settings: coupling SUMO with a Matlab/Simulink nanoscopic model", ***The first SUMO User Conference***, Berlin-Adlershof(2013).

EKLER ve MATLAB KODLARI

```
clear all
close all
clc

import traci.constants % Traci sabitlerini çağırma
system(['sumo-gui -c ' 'karabuk.sumocfg --start&']); % Grafik ekranda uyfulamayı
açma
traci.init(); % Traciyi başlatma
% SUMO Görünüm ayarları
traci.gui.setZoom('View #0', 1000);
traci.gui.setOffset('View #0', 1959, 1670);
traci.gui.setSchema('View #0', 'real world');

% ******** Başlangıç değerleri
Intersect_greentime_a=[35 35 35 35 35 35 35]; % Anayol ilk ışık değerlerini ata
Intersect_greentime_b=[0 0 0 0 0 0 0]; % Tali yollar ilk ışık değerlerini ata
light_turn=['a' 'a' 'a' 'a' 'a' 'a' 'a'] ; % Anayol öncelik sırası ver
count_vehcle_control_a =[1 1 1 1 1 1 1]; % ana yol Araç sayılırını kontrol et
paramatresi
count_vehcle_control_b =[1 1 1 1 1 1 1]; % ana yol Araç sayılırını kontrol et
parametresi

traci.trafficlights.setRedYellowGreenState('-10', 'GGGGggrrrrGGGGggrrrr'); %1
Anayol otogar
traci.trafficlights.setRedYellowGreenState('-218', 'rrrrGGGGgGGGgg'); %2 Petrol
traci.trafficlights.setRedYellowGreenState('-46', 'rrrrrrGGggrrrrrGGgg'); %3 Sanayi
traci.trafficlights.setRedYellowGreenState('-64', 'GGGGggrrrrGGGGggrrrr'); % 4
Üniversite
```

```
traci.trafficlights.setRedYellowGreenState('-110', 'rrrrGGGggrrrrGGGGgg'); % 5
Yapı market
traci.trafficlights.setRedYellowGreenState('-162', 'GGGGggrrrrGGGGggrrrr'); % 6
safranbolu otogar
traci.trafficlights.setRedYellowGreenState('-192', 'GGggrrrrGGggrrrr'); % 7
safranbolu merkez
Intersect_greentime_a_count=[0 0 0 0 0 0 0];
Intersect_greentime_b_count=[0 0 0 0 0 0 0];

for i = 1:1000 % Sumo adımları zaman döngüsü
   traci.simulationStep(); % Simulasyonu her adımda çalıştırma
   % -10 Intersection---------- 1. Otogar kavşağı
   %****************** 1 ANA YOL ********************

   Intersect_greentime_a_count(1)=Intersect_greentime_a_count(1)+1;

   if Intersect_greentime_a_count(1) == Intersect_greentime_a(1)
      light_turn(1)='b';
      Intersect_greentime_a_count(1)=0;
      count_vehcle_control_b(1)=1;
   end
   a=0;
   b=0;
   if light_turn(1)=='a' %&& count_vehcle_control_a(1)==1
      a(1) = traci.lane.getLastStepVehicleNumber('-15273_0');
      a(2) = traci.lane.getLastStepVehicleNumber('-15273_1');
      a(3) = traci.lane.getLastStepVehicleNumber('-15273_2');
      a(4) = traci.lane.getLastStepVehicleNumber('--15281#0_0');
      a(5) = traci.lane.getLastStepVehicleNumber('--15281#0_1');
      a(6) = traci.lane.getLastStepVehicleNumber('--15281#0_2');

      vehicle_count= max(a); % ana yol en uzun şerit
      if vehicle_count>0
```

```
        Intersect_greentime_a(1)=gettime(vehicle_count);
        traci.trafficlights.setRedYellowGreenState('-10', 'GGGGggrrrrGGGGggrrrr');
% Anayol
        count_vehcle_control_a(1)=0;
    else
        light_turn(1)='b';
        Intersect_greentime_a_count(1)=0;
        count_vehcle_control_b(1)=1;
    end
end

%****************** 1 TALİ YOL ********************
Intersect_greentime_b_count(1)=Intersect_greentime_b_count(1)+1;
if Intersect_greentime_b_count(1) == Intersect_greentime_b(1)
    light_turn(1)='a';
    Intersect_greentime_b_count(1)=0;
    count_vehcle_control_a(1)=1;
end

if light_turn(1)=='b' %&& count_vehcle_control_b(1)==1
    b(1)= traci.lane.getLastStepVehicleNumber('-15275_0');
    b(2)= traci.lane.getLastStepVehicleNumber('-15275_1');
    b(3)= traci.lane.getLastStepVehicleNumber('-15277_0');

    vehicle_count= max(b); % tali yol
    if vehicle_count>0
        Intersect_greentime_b(1)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-10', 'rrrrrrGGggrrrrrrGGgg'); %
Taliyol
        count_vehcle_control_b(1)=0;
    else
        light_turn(1)='a';
```

```
        Intersect_greentime_b_count(1)=0;
        count_vehcle_control_a(1)=1;
    end
end

%% -218 Intersection---------- 2. PETROL kavşağı**********
%****************** 2 ANA YOL ********************
Intersect_greentime_a_count(2)=Intersect_greentime_a_count(2)+1;
if Intersect_greentime_a_count(2) == Intersect_greentime_a(2)
    light_turn(2)='b';
    Intersect_greentime_a_count(2)=0;
    count_vehcle_control_b(2)=1;
end

if light_turn(2)=='a'  %&& count_vehcle_control_a(2)==1
    a(1) = traci.lane.getLastStepVehicleNumber('-15281#2_0');
    a(2) = traci.lane.getLastStepVehicleNumber('-15281#2_1');
    a(3) = traci.lane.getLastStepVehicleNumber('-15281#2_2');
    a(4) = traci.lane.getLastStepVehicleNumber('--15281#3_0');
    a(5) = traci.lane.getLastStepVehicleNumber('--15281#3_1');
    a(6) = traci.lane.getLastStepVehicleNumber('--15281#3_2');

    vehicle_count= max(a); % ana yol en uzun şerit
    if vehicle_count>0
        Intersect_greentime_a(2)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-218', 'rrrrGGGGgGGGgg'); % 2
ANAYOL
        count_vehcle_control_a(2)=0;
    else
        light_turn(2)='b';
        Intersect_greentime_a_count(2)=0;
        count_vehcle_control_b(2)=1;
```

```
    end
  end

  %****************** 2 TALİ YOL ********************
  Intersect_greentime_b_count(2)=Intersect_greentime_b_count(2)+1;
  if Intersect_greentime_b_count(2) == Intersect_greentime_b(2)
    light_turn(2)='a';
    Intersect_greentime_b_count(2)=0;
    count_vehcle_control_a(2)=1;
  end

  if light_turn(2)=='b'%&& count_vehcle_control_b(2)==1
    b(1)= traci.lane.getLastStepVehicleNumber('--15321_0');

    vehicle_count= max(b); % tali yol
    if vehicle_count>0
      Intersect_greentime_b(2)=gettime(vehicle_count);

      traci.trafficlights.setRedYellowGreenState('-218',  'GGGgGrrrrrrrrr');  %  2
Taliyol
      count_vehcle_control_b(2)=0;
    else
      light_turn(2)='a';
      Intersect_greentime_b_count(2)=0;
      count_vehcle_control_a(2)=1;
    end
  end

  %% -46 Intersection----------------  3. Sanayi kavşağı**************
  %****************** 3 ANA YOL ********************
  Intersect_greentime_a_count(3)=Intersect_greentime_a_count(3)+1;
  if Intersect_greentime_a_count(3) == Intersect_greentime_a(3)
    light_turn(3)='b';
```

```
    Intersect_greentime_a_count(3)=0;
    count_vehcle_control_b(3)=1;
  end

  if light_turn(3)=='a' %&& count_vehcle_control_a(3)==1
    a(1) = traci.lane.getLastStepVehicleNumber('-15281#3_0');
    a(2) = traci.lane.getLastStepVehicleNumber('-15281#3_1');
    a(3) = traci.lane.getLastStepVehicleNumber('-15281#3_2');
    a(4) = traci.lane.getLastStepVehicleNumber('--15285#0_0');
    a(5) = traci.lane.getLastStepVehicleNumber('--15285#0_1');
    a(6) = traci.lane.getLastStepVehicleNumber('--15285#0_2');

    vehicle_count= max(a); % ana yol en uzun şerit
    if vehicle_count>0
      Intersect_greentime_a(3)=gettime(vehicle_count);

      traci.trafficlights.setRedYellowGreenState('-46',    'GGGGggrrrrGGGggrrrr');
% 3 ANAYOL
      count_vehcle_control_a(3)=0;
    else
      light_turn(3)='b';
      Intersect_greentime_a_count(3)=0;
      count_vehcle_control_b(3)=1;
    end
  end

  %****************** 3 TALİ YOL ********************
  Intersect_greentime_b_count(3)=Intersect_greentime_b_count(3)+1;
  if Intersect_greentime_b_count(3) == Intersect_greentime_b(3)
    light_turn(3)='a';
    Intersect_greentime_b_count(3)=0;
    count_vehcle_control_a(3)=1;
  end
```

```
if light_turn(3)=='b' %&& count_vehcle_control_b(3)==1
    b(1)= traci.lane.getLastStepVehicleNumber('--15315_0');
    b(2)= traci.lane.getLastStepVehicleNumber('-15317_0');

    vehicle_count= max(b); % tali yol
    if vehicle_count>0
        Intersect_greentime_b(3)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-46', 'rrrrrrGGggrrrrrGGgg'); % 3
Taliyol
        count_vehcle_control_b(3)=0;
    else
        light_turn(3)='a';
        Intersect_greentime_b_count(3)=0;
        count_vehcle_control_a(3)=1;
    end
end
%% -64 Intersection----------------- 4. Üniversite kavşağı**************
%****************** 4 ANA YOL ********************
Intersect_greentime_a_count(4)=Intersect_greentime_a_count(4)+1;
if Intersect_greentime_a_count(4) == Intersect_greentime_a(4)
    light_turn(4)='b';
    Intersect_greentime_a_count(4)=0;
    count_vehcle_control_b(4)=1;
end

if light_turn(4)=='a' % && count_vehcle_control_a(4)==1

    a(1) = traci.lane.getLastStepVehicleNumber('-15285#0_0');
    a(2) = traci.lane.getLastStepVehicleNumber('-15285#0_1');
    a(3) = traci.lane.getLastStepVehicleNumber('-15285#0_2');
    a(4) = traci.lane.getLastStepVehicleNumber('--15285#1_2');
```

```
    a(5) = traci.lane.getLastStepVehicleNumber('--15285#1_1');
    a(6) = traci.lane.getLastStepVehicleNumber('--15285#1_0');

    vehicle_count= max(a); % ana yol en uzun şerit
    if vehicle_count>0
      Intersect_greentime_a(4)=gettime(vehicle_count);

      traci.trafficlights.setRedYellowGreenState('-64',  'GGGGggrrrrGGGGggrrrr');
% 4 ANAYOL
      count_vehcle_control_a(4)=0;
    else
      light_turn(4)='b';
      Intersect_greentime_a_count(4)=0;
      count_vehcle_control_b(4)=1;
    end

  end

  %****************** 4 TALİ YOL ********************
  Intersect_greentime_b_count(4)=Intersect_greentime_b_count(4)+1;
  if Intersect_greentime_b_count(4) == Intersect_greentime_b(4)
    light_turn(4)='a';
    Intersect_greentime_b_count(4)=0;
    count_vehcle_control_a(4)=1;
  end

  if light_turn(4)=='b'%&& count_vehcle_control_b(4)==1

    b(1) = traci.lane.getLastStepVehicleNumber('--15287_0');
    b(2) = traci.lane.getLastStepVehicleNumber('--15287_1');
    b(3) = traci.lane.getLastStepVehicleNumber('-15289_0');

    vehicle_count= max(b); % tali yol
```

```
        if vehicle_count>0
            Intersect_greentime_b(4)=gettime(vehicle_count);

            traci.trafficlights.setRedYellowGreenState('-64',  'rrrrrrGGggrrrrrrGGgg');  %
4 Taliyol
            count_vehcle_control_b(4)=0;
        else
            light_turn(4)='a';
            Intersect_greentime_b_count(4)=0;
            count_vehcle_control_a(4)=1;
        end
    end
    %% -110 Intersection----------  5. Yapı market kavşağı**********
    %****************** 5 ANA YOL ********************
    Intersect_greentime_a_count(5)=Intersect_greentime_a_count(5)+1;
    if Intersect_greentime_a_count(5) == Intersect_greentime_a(5)
        light_turn(5)='b';
        Intersect_greentime_a_count(5)=0;
        count_vehcle_control_b(5)=1;
    end

    if light_turn(5)=='a' %&& count_vehcle_control_a(5)==1

        a(1) = traci.lane.getLastStepVehicleNumber('-15285#5_0');
        a(2) = traci.lane.getLastStepVehicleNumber('-15285#5_1');
        a(3) = traci.lane.getLastStepVehicleNumber('-15285#5_2');
        a(4) = traci.lane.getLastStepVehicleNumber('--15301#0_0');
        a(5) = traci.lane.getLastStepVehicleNumber('--15301#0_1');
        a(6) = traci.lane.getLastStepVehicleNumber('--15301#0_2');

        vehicle_count= max(a); % ana yol en uzun şerit
        if vehicle_count>0
            Intersect_greentime_a(5)=gettime(vehicle_count);
```

```
        traci.trafficlights.setRedYellowGreenState('-110',  'rrrrGGGggrrrrGGGGgg');
% 5 ANAYOL
        count_vehcle_control_a(5)=0;
    else
        light_turn(5)='b';
        Intersect_greentime_a_count(5)=0;
        count_vehcle_control_b(5)=1;
    end
end

%****************** 5 TALİ YOL ********************
Intersect_greentime_b_count(5)=Intersect_greentime_b_count(5)+1;
if Intersect_greentime_b_count(5) == Intersect_greentime_b(5)
    light_turn(5)='a';
    Intersect_greentime_b_count(5)=0;
    count_vehcle_control_a(5)=1;
end

if light_turn(5)=='b' %&& count_vehcle_control_b(5)==1

    b(1)= traci.lane.getLastStepVehicleNumber('-15299_0');
    b(2)= traci.lane.getLastStepVehicleNumber('-15297_0');

    vehicle_count= max(b); % tali yol
    if vehicle_count>0
        Intersect_greentime_b(5)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-110',  'GGggrrrrrGGggrrrrrr');  %
5 Taliyol
        count_vehcle_control_b(5)=0;
    else
        light_turn(5)='a';
        Intersect_greentime_b_count(5)=0;
```

```
        count_vehcle_control_a(5)=1;
      end
    end
    %%    -192    Intersection----------    6.    Safranbolu    otogar
kavşağı**********
    %****************** 7 ANA YOL ********************
    Intersect_greentime_a_count(6)=Intersect_greentime_a_count(6)+1;
    if Intersect_greentime_a_count(6) == Intersect_greentime_a(6)
      light_turn(6)='b';
      Intersect_greentime_a_count(6)=0;
      count_vehcle_control_b(6)=1;
    end

    if light_turn(6)=='a' %&& count_vehcle_control_a(6)==1

      a(1) = traci.lane.getLastStepVehicleNumber('-15301#1_2');
      a(2) = traci.lane.getLastStepVehicleNumber('-15301#1_1');
      a(3) = traci.lane.getLastStepVehicleNumber('-15301#1_0');
      a(4) = traci.lane.getLastStepVehicleNumber('--15301#2_0');
      a(5) = traci.lane.getLastStepVehicleNumber('--15301#2_1');
      a(6) = traci.lane.getLastStepVehicleNumber('--15301#2_2');

      vehicle_count= max(a); % ana yol en uzun şerit
      if vehicle_count>0
        Intersect_greentime_a(6)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-162',
'GGGGggrrrrGGGGggrrrr'); % 7 ANAYOL
        count_vehcle_control_a(6)=0;
      else
        light_turn(6)='b';
        Intersect_greentime_a_count(6)=0;
        count_vehcle_control_b(6)=1;
```

```
    end
end

%****************** 6 TALİ YOL ********************
Intersect_greentime_b_count(6)=Intersect_greentime_b_count(6)+1;
if Intersect_greentime_b_count(6) == Intersect_greentime_b(6)
    light_turn(6)='a';
    Intersect_greentime_b_count(6)=0;
    count_vehcle_control_a(6)=1;
end

if light_turn(6)=='b'%&& count_vehcle_control_b(6)==1

    b(1) = traci.lane.getLastStepVehicleNumber('--15307_0');
    b(2) = traci.lane.getLastStepVehicleNumber('--15305_0');

    vehicle_count= max(b); % tali yol
    if vehicle_count>0
        Intersect_greentime_b(6)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-162', 'rrrrrrGGggrrrrrrGGgg'); %
6 Taliyol
        count_vehcle_control_b(6)=0;
    else
        light_turn(6)='a';
        Intersect_greentime_b_count(6)=0;
        count_vehcle_control_a(6)=1;
    end
end

%% -192 Intersection---------- 7. Safranbolu kavşağı**********
%****************** 7 ANA YOL ********************
Intersect_greentime_a_count(7)=Intersect_greentime_a_count(7)+1;
```

```
if Intersect_greentime_a_count(7) == Intersect_greentime_a(7)
    light_turn(7)='b';
    Intersect_greentime_a_count(7)=0;
    count_vehcle_control_b(7)=1;
end

if light_turn(7)=='a' %&& count_vehcle_control_a(7)==1

    a(1) = traci.lane.getLastStepVehicleNumber('-15301#3_2');
    a(2) = traci.lane.getLastStepVehicleNumber('-15301#3_1');
    a(3) = traci.lane.getLastStepVehicleNumber('-15301#3_0');
    a(4) = traci.lane.getLastStepVehicleNumber('--15313_0');

    vehicle_count= max(a); % ana yol en uzun şerit
    if vehicle_count>0
        Intersect_greentime_a(7)=gettime(vehicle_count);

        traci.trafficlights.setRedYellowGreenState('-192', 'GGggrrrrGGggrrrr'); % 7
ANAYOL
        count_vehcle_control_a(7)=0;
    else
        light_turn(7)='b';
        Intersect_greentime_a_count(7)=0;
        count_vehcle_control_b(7)=1;
    end
end

%****************** 6 TALİ YOL ********************
Intersect_greentime_b_count(7)=Intersect_greentime_b_count(7)+1;
if Intersect_greentime_b_count(7) == Intersect_greentime_b(7)
    light_turn(7)='a';
    Intersect_greentime_b_count(7)=0;
    count_vehcle_control_a(7)=1;
```

```
    end

    if light_turn(7)=='b'%&& count_vehcle_control_b(7)==1

        b(1) = traci.lane.getLastStepVehicleNumber('--15311_0');
        b(2) = traci.lane.getLastStepVehicleNumber('--15319_0');

        vehicle_count= max(b); % tali yol
        if vehicle_count>0
            Intersect_greentime_b(7)=gettime(vehicle_count);

            traci.trafficlights.setRedYellowGreenState('-192', 'rrrrGGggrrrrGGgg'); % 6
Taliyol
            count_vehcle_control_b(7)=0;
        else
            light_turn(7)='a';
            Intersect_greentime_b_count(7)=0;
            count_vehcle_control_a(7)=1;
        end
    end

end
traci.close()
```

ÖZGEÇMİŞ

Alper Talha KARADENİZ 1989'da Trabzon'da doğdu. İlk ve orta öğrenimini Ordu şehrinde tamamladı. Özel Seçkin Kolej Lisesi, Fen Lisesi Bölümü'nden mezun olduktan sonra 2007 yılında Sakarya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne girdi. 2011'de "iyi" derece ile mezun olduktan sonra LS-Pro Yazılım'da Yazılım Uzmanı olarak çalışmaya başladı. 2013'de Yerinde Bilgisayar Yazılım ve Danışmanlık Hizmetlerinde proje yöneticisi olarak, 2014 yılından bu zamana kadarda Ordu Üniversitesi Mesudiye Meslek Yüksek Okulu Bilgisayar Programcılığı Bölümü'nde öğretim görevlisi olarak çalışmaya devam etmektedir. 2013 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda başlamış olduğu yüksek lisans programını sürdürmektedir. Evli ve İngilizce bilmektedir.

ADRES BİLGİLERİ

Adres: Ordu Üniversitesi Mesudiye Meslek Yüksek Okulu
Mesudiye / ORDU

Tel: 0533 696 21 42

Faks: 0452 761 33 50

E-posta: alperkaradeniz5@gmail.com