

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**SEZGİSEL ALGORİTMA TABANLI TRAFİK IŞIKLARI
OPTİMİZASYONU VE BENZETİMİ**

YÜKSEK LİSANS TEZİ

EMİN TOPAL

TEZ DANIŞMANI
PROF. DR. CİHAN KARAKUZU
İKİNCİ TEZ DANIŞMANI
DR. ÖĞR. ÜYESİ HASAN BOZKURT

BİLECİK, 2022

10442345

T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**SEZGİSEL ALGORİTMA TABANLI TRAFİK IŞIKLARI
OPTİMİZASYONU VE BENZETİMİ**

YÜKSEK LİSANS TEZİ

EMİN TOPAL

TEZ DANIŞMANI
PROF. DR. CİHAN KARAKUZU
İKİNCİ TEZ DANIŞMANI
DR. ÖĞR. ÜYESİ HASAN BOZKURT

BİLECİK, 2022

10442345

BEYAN

“Sezgisel Algoritma Tabanlı Trafik Işıkları Optimizasyonu ve Benzetimi” adlı yüksek lisans/doktora/sanatta yeterlik tezi/dönem projesinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
DESTEK ALINMIŞTIR		DESTEK ALINMAMIŞTIR	x
Destek alındı ise;			
Destekleyen kurum;			
Desteğin Türü		Proje Numarası	
1- BAP (Bilimsel Araştırma Projesi)			
2- TÜBİTAK			
Diğer;.....			
ETİK KURUL onayı var ise;			
ETİK KURUL karar tarih/sayı:		/.....	

Emin TOPAL

Tarih

.....

İmza

.....

ÖN SÖZ

Bu tez çalışmasının yazılmasında, çalışmamı sahiplenerek takip eden danışmanım Sayın Prof. Dr. Cihan KARAKUZU ve Dr. Öğr. Üyesi Hasan BOZKURT'a değerli katkı ve emekleri için teşekkürlerimi ve saygılarımı sunarım.

Savunma sınavı sunumu sırasında değerli jüri üyeleri Dr. Öğr. Üyesi Murtaza CİCİOĞLU, Prof Dr. Uğur YÜZGEÇ ve danışmanım Sayın Prof. Dr. Cihan KARAKUZU'ya çalışmamın son haline gelmesindeki değerli katkıları için teşekkürlerimi ve saygılarımı sunarım.

Kariyer ve başarı süreçlerimde her zaman arkamda durarak bana destek olan ve bu günlere ulaşmamda emeklerini hiçbir zaman unutmayacağım değerli aileme, tüm hocalarıma ve arkadaşlarıma teşekkür ederim.

Emin TOPAL

2022

ÖZET

SEZGİSEL ALGORİTMA TABANLI TRAFİK IŞIKLARI OPTİMİZASYONU VE BENZETİMİ

Bu çalışmada örnek bir yol ağı problem olarak belirlenerek trafik ağı içerisinde bulunan araçların seyahat sürelerini azaltmak amacıyla trafik sinyal sürelerinin eniyilemesi üzerine çalışılmıştır. Farksal Gelişim sezgisel yaklaşımı trafik yol bağlantısının modellenebileceği bir benzetim programı ile entegre edilerek trafik ışıklarının kullanıldığı bir trafik ağı modellenmesi yapılmıştır. Oluşturulan model kullanılarak belirlenen yol ağında bulunan trafik sinyal sürelerinin eniyilemesi yapılarak araçların seyahat süresi en aza indirilmeye çalışılmıştır. Farksal Gelişim sezgisel algoritması bu çalışmada iki farklı yöntem ile koşturulmuştur. Yapılan eniyileme işlemleri sonucu elde edilen seyahat süreleri ile sabit/standart trafik ışıklarının seyahat süreleri karşılaştırılmıştır. Geliştirilen akıllı trafik ağı ile modellenen benzetim ortamı yapılan eniyileme işlemlerinin görsel olarak analiz edilmesine olanak sağlamaktadır. Çalışılan trafik ağı için önceden belirlenmiş yol başlangıç-bitiş noktalarına göre araç seyahat sürelerinin karşılaştırması yapılmıştır. Çalışma öncesi ve kullanılan iki yöntem ile oluşturulmuş yol ağı görsel olarak belirlenen bir an için de karşılaştırılmıştır. Çalışma sonucunda sezgisel yaklaşım ile yapılan koşturmalar ile elde edilen trafik sinyal sürelerinin standart süreli trafik sinyallerine göre daha iyi sonuç verdiği görülmüştür. Trafik seyahat sürelerinin sezgisel yaklaşımlar ile düşürülebileceği gözlemlenmiştir.

Anahtar kelimeler: SUMO, Allsop ve Charlesworths Yol Ağı, Farksal Gelişim Algoritması, Yol Ağı Benzetimi, Trafik Işıkları, Sezgisel Yaklaşım.

ABSTRACT

HEURISTIC ALGORITHM BASED TRAFFIC LIGHTS OPTIMIZATION AND SIMULATION

In this study, a benchmark was determined as a road network problem and the optimization of traffic signal times was studied in order to reduce the travel times of vehicles in the traffic network. By integrating Differential Evolution Heuristic approach with a simulation program in which the traffic road connection can be modeled, a traffic network model using smart traffic lights has been made. By using the formed model, the traffic signal times in the determined road network were optimized and an effort was made to obtain a minimum travel time for the vehicles. Differential Evolution algorithm was run with two different methods. The travel times obtained as a result of the optimization processes and the travel times of the fixed/standard traffic lights were compared. The simulation environment modeled with the developed smart traffic network allows the optimization processes to be analyzed visually. For the traffic network studied, the vehicle travel times were compared according to the predetermined road start-end points. The road network formed by the two methods used and ones before the study was compared also visually at a moment. As a result of the study, it has been seen that the traffic signal times obtained with the heuristic approach gives better results than the standard time traffic signals. It has been observed that traffic travel times can be reduced using heuristic approaches.

Keywords: SUMO, Allsop and Charlesworths Road Network, Differential Evolution Algorithm, Road Network Simulation, Traffic Lights, Heuristic Approach.

İÇİNDEKİLER

	Sayfa
ÖNSÖZ.....	i
ÖZET.....	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
KISALTMALAR VE SİMGELER LİSTESİ	viii
1. GİRİŞ.....	1
2. KULLANILAN YAZILIM, YÖNTEMLER VE EK PROGRAMLAR.....	6
2.1. Kullanılan Programlama ve İşaretleme Dilleri.....	6
2.1.1. Python programlama dili	6
2.1.2. Genişletilebilir işaretleme dili (XML)	6
2.2. Çalışma İçin Gerekli İşletim Sistemleri.....	6
2.2.1. Windows işletim sistemi	6
2.2.2. Linux işletim sistemi	7
2.3. Çalışmada Kullanılan Kaynak Kod Düzenleyici	7
2.4. SUMO Benzetim Programı.....	7
2.4.1. Trafik yol ağı tanımlama işlemleri	7
2.4.2. Trafik ışıklarını tanımlama işlemleri.....	8
2.4.3. SUMO’da araç tanımlamaları.....	10
2.4.4. SUMO birleşim dosyası tanımlamaları.....	10
2.4.5. SUMO’da trafik senaryolarının koşturulması.....	11
2.4.6. SUMO’da analiz ve çıktı sonuçlarının elde edilmesi	12
2.5. Farksal Gelişim Algoritması.....	12
3. ÖRNEK (DENEKTAŞI) YOL AĞI İLE SUMO BENZETİMİ.....	14
3.1. Trafik Yol Ağının Modellenmesi.....	14
3.2. Trafik Sinyallerinin Faz Düzenlemeleri	15
4. KULLANILAN SEZGİSEL YÖNTEM	17
4.1. Değer Fonksiyonu	17
4.2. DE Algoritmasında Sürü Oluşturma Yöntemleri	18
4.2.1. Tek birey ile çözüm geliştirilmesi	18

4.2.2. Sürü ile çözüm geliştirme	20
4.3. Trafik Sinyallerinin Dengeleme İşlemleri	20
5. SUMO VE DE ALGORİTMASININ ENTEGRASYONU.....	23
6. ARAŞTIRMA VE BULGULAR.....	25
6.1. Eniyileme Öncesi Ön Çalışmaların Gerçekleştirilmesi.....	25
6.2. Tek Birey ile Yöntemin Uygulanması.....	30
6.3. Çok Bireyli Sürü Kullanılarak Yöntemin Uygulanması.....	32
7. SONUÇLAR VE TARTIŞMALAR.....	36
KAYNAKÇA	39



TABLÖLER LİSTESİ

Sayfa

Tablo 3.1. Trafik sinyalleri ve faz tablosu	15
Tablo 3.2. Fazlara göre belirlenmiş trafik sinyal süreleri	16
Tablo 6.1. Sezgisel yaklaşımda kullanılan başlangıç trafik sinyal süreleri	25
Tablo 6.2. Sezgisel parametrelerin eniyilemesi için kullanılan araç sayıları	26
Tablo 6.3. Tablo 6.1'deki sinyal süreleri ve Tablo 6.2'deki araç sayısı senaryosuna göre işletilen trafik ağ benzetim sonucu elde edilen başlangıç noktalarına göre araçların ortalama varış süreleri	26
Tablo 6.4. DE algoritmasının 100 iterasyon koşturulması sonucu bulunan ortalama seyahat süresi tablosu	27
Tablo 6.5. CR=0.85 F=0.80 ile bulunan yeşil sinyal süreleri	28
Tablo 6.6. CR=0.85 F=0.80 verileri ile bulunan ortalama araç seyahat süreleri	28
Tablo 6.7. Denektaşı yol ağının eniyilemesi için belirlenen araç sayıları	30
Tablo 6.8. Eniyileme öncesi yoğun araç sayılı benzetimde araçların ortalama seyahat süreleri	30
Tablo 6.9. Tek birey ile koşturulan algoritma ile belirlenen en iyi yeşil sinyal süreleri	31
Tablo 6.10. Tek birey ile koşturulan algoritma ile elde edilen ortalama seyahat süreleri	31
Tablo 6.11. Çok boyutlu sürü ile belirlenen en iyi yeşil sinyal süreleri	32
Tablo 6.12. Sürü ile koşturulan algoritmanın sunduğu çözüm ile elde edilen ortalama seyahat süreleri	33

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. <i>NetEdit</i> ile oluşturulmuş örnek yol tanımlarının yapıldığı ara yüz görseli.....	8
Şekil 2.2. <i>NetEdit</i> ile oluşturulmuş trafik denetleç görseli	10
Şekil 2.3. SUMO ara yüzü ve örnek trafik senaryosu görseli.....	11
Şekil 3.1. Allsop ve Charlesworth yol ağının SUMO ile benzetimi	14
Şekil 3.2. SUMO’da trafik sinyali ve yol bağlantıları görseli	15
Şekil 3.3. Trafik ağında bulunan 3 numaralı kavşak için SUMO faz benzetimi	16
Şekil 4.1. Trafik sinyali sezgisel optimizasyon ve dengeleme işlemleri akış şeması	22
Şekil 5.1. SUMO ve DE algoritması entegre işleyiş şeması.....	23
Şekil 5.2. SUMO ve DE algoritması entegrasyon ve etkileşim şeması	24
Şekil 6.1. 100 iterasyon koşturulan sezgisel yaklaşım sonucu “Ortalama Seyahat Süresi” değişim grafiği (CR=0.85, F=0.8).....	29
Şekil 6.2. Tek birey ile eniyileme sürecinde ölçüt değişimi	32
Şekil 6.3. Çok sayıda birey (sürü) ile eniyileme sürecinde ölçüt değişimi.....	34
Şekil 6.4. Sinyal süreleri eniyilenmemiş durumda trafik yol durumu (1.26 anı).....	34
Şekil 6.5. Tek birey kullanılarak eniyilenmiş sinyal süreleri ile trafik yol durumu (1.26 anı)	35
Şekil 6.6. Çok boyutlu bireyler ile optimizasyon sonucu trafik yol durumu (1.26 anı)	35

KISALTMALAR VE SİMGELER LİSTESİ

SUMO: Kentsel Hareketlilik Simülasyonu (Simulation of Urban Mobility)

DE: Farksal Gelişim (Differential Evolution)

PSO: Parçacıklı Sürü Optimizasyonu

GA: Genetik Algoritma

XML: Genişletilebilir İşaretleme Dili (Extensible Markup Language)

NP: Sürü Büyüklüğü

D: Sürünün Uzay Boyutu

CR: Çaprazlama Oranı

G: Jenerasyon Sayısı

F: Ölçekleme Faktörü

C: Toplam Döngü Süresi

Q: Yeşil Işık Süresi

Cmax: Üst Toplam Yeşil Sinyal Süresi

Cmin: Alt Toplam Yeşil Sinyal Süresi

Qmin: Alt Yeşil Sinyal Süresi

1. GİRİŞ

Trafik, günlük yařantımızda insanların seyahatlerini gerekleřtirmek amacıyla eřitli tařıtlar kullanılarak oluřan kavramlar bütünüdür. Bu kavram sık sık ierisinde bulunmak zorunda olduėumuz temel ihtiyalarımız arasındadır. Bu temel ihtiya insanların ve tařıtların yoğun hareketlilik taleplerine karřılık yol aėında doėru planlamalar yapılması gibi yeni kavramları beraberinde getirmektedir. Trafikte geirilen seyahat süresinin artmasının yol aėı ierisinde bulunan herkes iin olumsuz davranıřlara sebep olduėu ve trafik etkenin insan psikolojisini doėrudan etkilediėi yapılan arařtırmalar sonucu kanıtlanmıřtır (Groeger, J. A., & Rothengatter, J. A., 1998, s. 1-9). Tüm bu sistem ierisinde herkesin öncelikli beklentisi, seyahat sürelerinin en aza dıřürölmesi ve istenen varıř noktalarına en kısa sürede ulařılmasıdır. Bu karmařık trafik seyahat problemi günümüzde özölmesi beklenen sorunlar arasında bulunmaktadır. Trafik seyahat sürelerini etkileyen birden fazla etken bulunmaktadır. Bu etkenler arasında ara hızları, yol ara kapasitesi ve sabit trafik ıřıkları bulunmaktadır. Sabit süreli trafik ıřıklarının akıllı sistemler ile belirlenmesi bu bağlamda daha ok önem kazanmaktadır.

Trafik aėlarının neredeyse tamamında yetkili birimler tarafından istatistiksel olarak belirlenmiř sabit süreli trafik ıřıkları bulunmaktadır. Trafik ıřıklarını sabit süreli deėerlere bağlamak dinamik olarak iřleyen bir yol aėı iin özüm saėlayamamaktadır. Geliřen ölkeler bu sorunu yol kapasitelerinin arttırılması, trafik polisleri gibi üçüncü etmenler ile özmeye alıřmaktadır. Ancak tüm trafik yol aėları iin trafik polisi bulundurmak imkansızdır. Aynı řekilde yol kapasitelerini veya řerit sayılarını arttırmak ise yüksek maliyetli ve yüksek iř gücü gerektiren iřlemler arasında bulunduėundan yetkililer bu yöntemlere mesafeli yaklařarak farklı yöntemler bulmayı amalamıřlardır.

Farklı yollar arayan yetkililer ve arařtırmacılar bu bağlamda yol aėı ierisinde bulunan sabit süreli trafik ıřıkları veya eski aė türlerini, akıllı sistemler ile deėiřtiren alıřmaları gerekleřtirmiřlerdir (Khekare & Sakhare, 2013, s. 302-305) (Hirankitti & Krohkaew, 2007). Seyahat süresi problemini özmek iin trafik sinyallerinin akıllı sistemler ile deėiřtirilmesi üzerine olduka fazla arařtırmalar ve alıřmalar yapılmıřtır (Wei, Zheng , Yao, & Li, 2018). Trafik seyahat problemi özümünde kullanılan sinyal sürelerinin eniyilenmesi iin de birok yöntem ve algoritma kullanılarak iyi sonuçlar elde edilmiřtir. Yapılan tüm alıřmalarda eniyilenen akıllı sistemlerin sabit süreli trafik ıřıklarına göre daha iyi seyahat süresi elde edildiėi görölmüřtür. Trafikte bulunan anlık ara sayılarına göre, yol kapasitesine göre ve anlık

olarak deęiřen yol řartlarını baz alan ve bu verilere gre trafięi olumlu ynde yneten akıllı sistemler gelecekte kullanılması kaınılmaz olan sistemler arasında bulunmaktadır.

Yapılan literatr alıřmalarında genel olarak seyahat sresi problemini, daha az maliyet harcayarak zmek ve seyahat srelerini azaltmak iin trafik sinyal srelerini dzenlemenin bu baęlamda zm olabileceęi belirtilmiřtir (Liu, Liu, & Chen, 2017, s. 1-8). Maliyetli ve iř gc gerektiren iřlemlerin aksine akıllı trafik sinyallerinin dzenlenmesi kolay ve dięer yntemlere gre uygulanabilirlięi bakımından arařtırmacılar ve yazarlar iin ilgi ekici bulunmuřtur (Hu & Wrang, 2010, s. 372-376). Trafikte yařanan seyahat sresi problemi zm iin arařtırmacılar ve yazarlar farklı algoritmalar ve yntemler ile sorunu zmeyi amalamıřlardır (Chao, Lee, & Wang, 2008). Modern řehirler iin trafik sorunlarının zlmesinin nemini vurgulayan yazarlar bu baęlamda alıřmalarını ynlendirmiřlerdir (Araghi, Khosravi, Jhonstone, & Creighton, 2013).

Akıllı trafik sistemlerinin, geleneksel sistemlere gre daha iyi sonu verdięini TRACOM adlı iki katmanlı bir model kullanan Bařkan ve arkadařları (Bařkan vd., 2020, s. 572)'de trafikte geirilen sreleri en aza indirebilmek iin Allsop ve Charlesworths trafik yol aęı zerinde trafik sinyal srelerini belirleyen eniyileme alıřması gerekleřtirmiřlerdir. İki katmandan oluřan TRACOM modeli yksek ve alak seviyeli katman yardımıyla trafik sinyal srelerini dengeleyerek alıřmalarını gerekleřtirmiřlerdir. Arařtırmacılar Farksal Geliřim algoritmasının trafik sinyallerinin eniyilemesi ve seyahat sresini azaltarak problem iin olumlu veriler elde etmiřlerdir.

(Karakuzu & Demirci, 2010, s. 66-73)'de akıllı trafik sinyallerinin kullanıldıęı, drt koldan oluřan bir yol-kavřakta trafik akıřını iyileřtirilmesi zerine alıřmalar gerekleřtirmiřlerdir. Yol aęı zerine uygulanan bulanık denetle yeřil sinyal sreleri ile trafik seyahat srelerinin eniyilemesi zerinde durulmuřtur. Bulanık denetle yardımıyla oluřturulan akıllı trafik sinyal sistemlerinin sabit sreli trafik sinyal sistemlerine gre daha olumlu sonular verdięi raporlanmıřtır.

Trafik aęlarının ve sistemlerinin seyahat srelerine olan etkisi gz nne alınarak alıřmaları yapılan (Karakuzu & Topal, 2019, s. 255-260)'te yazarlar drt farklı senaryo iin bulanık mantık tabanlı akıllı trafik aęı modellemesi yapmıřlardır. Tenha, yoęun, normal ve az yoęun olmak zere farklı senaryolar zerinde Bilecik řehrinin yol aęı baz alınarak trafik sinyal srelerinin eniyilemesi gerekleřtirilmiřtir. SUMO programı yardımıyla trafik senaryoları benzetimi gerekleřtirilmiř olup sabit sreli trafik sinyallerine gre daha iyi sonular elde

edilmiştir. SUMO programı ile gerçeğe en yakın benzetim yapılmış olup şehir özelinde uygulanabilirliği maliyetli işlemlere göre daha yüksek olan bir yol ağı modellenmesi yapılmıştır.

Simulation of Urban Mobility (SUMO) (Krajzewicz, Erdmann, Behrisch, & Bieker, 2012)'te belirtildiği üzere kentsel trafik ağlarını modellemek üzere geliştirilmiş açık kaynak kodlu geliştirilebilirliği yüksek bir benzetim programıdır. SUMO ülke ve şehir ağlarının haritalarını kullanarak yapılabilecek benzetimlerin yanı sıra küçük trafik ağlarının da kolaylıkla modellenebileceği yazılımsal bir ortamdır. Farklı kodlama dilleri ile ve farklı platformlar ile kolayca entegre olabilmektedir. SUMO, seyahat süreleri, araç hız değerleri, seyahat süresi boyunca yaşanan gecikmeler ve araç seyahat bilgileri gibi birçok sonucu gerçek dünyaya uygun şekilde çıktı olarak verebilmektedir. Bu çalışmada açık kaynak kodlu olması, birden fazla trafik senaryosunun entegre uygulamalar ile kolayca yapılabileceğinden SUMO benzetim programı yöntemlerin uygulanacağı platform olarak seçilmiştir.

Farksal Gelişim Algoritması (DE) (Storn & Price, 1997, s. 341-359)'te de belirtildiği üzere operatörleri yönünden genetik algoritmaya dayanan sürü mantığı ile geliştirilmiş bir sezgisel eniyileme yöntemidir. Sistemlere ve programlara kolaylıkla entegre edilebilirliği ve kodlama yöntemiyle diğer algoritmalara göre kolaylık bakımından öne çıkan bir algoritmadır. İşleyiş yapısına göre sürüsünü geliştiren DE algoritması kodlanabilirliğinin kolay olması ve her koşma sonucunda birbirine çok yakın kararlı sonuçlar üretmesi nedeniyle bu çalışmada eniyileme işlemlerinin gerçekleştirilmesinde kullanılmıştır.

(Wiering, Vreeken, Veenen, & Koopman, 2004, s. 453-458)'da Pekiştirme Algoritması'nı kullanarak trafik ağında bulunan araçların ağ içerisindeki bekleme zamanlarını kullanarak yeşil ışık sürelerini eniyilemeyi amaçlamışlardır. Yazarlar aynı şekilde seyahat sürelerini azaltmak amacıyla sezgisel algoritmaları kullanarak (Sanchez, Galan, & Rubio, 2004, s. 1668-1674)'de Genetik Algoritma'nın trafik sinyal döngülerinin iyileştirilmesini sağlamışlardır. Modern şehirler için trafik sorunlarının çözülmesini amaçlayan Gupta ve arkadaşları ise (Gupta, Kumar, Reddy, & Panigrahi, 2017) yapay sinir ağları ile trafik ışıklarını düzenleyerek eniyileme çalışmaları gerçekleştirmişlerdir.

Derin öğrenme yöntemlerinin de seyahat süresi probleminin çözümü için oldukça iyi sonuçlar verdiğini belirten bir diğer çalışmada ise yazarlar (Coşkun, Baggag, & Chawla, 2018, s. 564-571)'de pekiştirmeli derin öğrenme yöntemini trafik sinyalleri üzerinde kullanarak seyahat süresinden kazanç elde etmişlerdir. Tajalli ve arkadaşları ise (Tajalli, Mehrabipour, & Hajbabaie, 2020, s. 6748-6759)'da trafik sinyal sürelerini, trafik ağında bulunan araçların

seyahat sürecinde elde ettikleri hız değerlerine göre düzenlemişlerdir. Tajalli ve arkadaşları birbirlerine bir sistem üzerinden bağlı araçların hız verilerini yol üzerinde bulunan trafik denetleçlerine iletmışlerdir. Araç hız değerlerini elde eden trafik denetleçleri ise bu değerlere göre hızların eniyilemesi ile seyahat süresinden kazanç sağlanmak istenmiştir. Trafik denetleci doğrusal olmayan bir denklem yardımıyla araçlardan gelen hız verilerini işleyerek eniyilenmiş hız verilerini tekrar araçlara göndermektedir. Yeni hız verilerini trafik denetlecinden alan araçlar ise hız değerlerini gelen veriye göre azaltıp-arttırarak seyahatin daha olumlu sürelerde geçmesini sağlamıştır.

Statik bir eniyileme yöntemi olan “Yeşil Dalga” ve “Adaptive Self-Organizing” yöntemlerini kullanan Gershenson ve arkadaşları (Gershenson & Rosenblueth, 2012, s. 386-403)’da yöntemleri denektaşı bir ağ üzerinde test ederek gerçek dünyada en iyi yöntemi bulmayı amaçlayan çalışmalar gerçekleştirmişlerdir.

Garcia-Nicto ve arkadaşları (García-Nieto, Olivera, & Alba, 2013, s. 823-839)’de bir başka sezgisel algoritma olan parçacık sürü algoritmasının (PSO), trafik sinyallerinin eniyilemesi çalışmalarına yer vermiştir. SUMO benzetim programı ile PSO algoritmasını entegre ederek benzetimini gerçekleştirmişlerdir. Araçlar için trafikte geçirilen seyahat sürelerini en aza indirerek eniyileme çalışmasını tamamlamayı amaçlamışlardır. Arjantin ve İspanya’nın belirli denektaşı şehirleri için trafik benzetimi işlemini ve PSO algoritmasının seyahat sürelerine etkisini incelemişlerdir. Yapılan çalışma ile hedefine ulaşan araç sayılarında belirlenen süre içerisinde iyileşme olduğu tespit edilmiştir. Sabit süreli trafik denetleçlerine göre daha akıllı trafik ağı sistemi oluşturularak işlem başarıyla sonuçlanmıştır.

İki farklı sezgisel yaklaşımın kullanıldığı (Abushehab, Abdalhaq, & Sartawi, 2014, s. 27-35)’de yazarlar Genetik Algoritma (GA) ve Parçacıklı Sürü Optimizasyonu (PSO) algoritmalarını kullanarak SUMO kentsel benzetim programı yardımıyla simülasyon tabanlı trafik seyahat süresi eniyileme çalışmaları yapmıştır. Denektaşı olarak belirlenen bir trafik ağı içerisindeki 13 adet trafik denetleci üzerinde uygulanan eniyileme işlemlerini gerçekleştiren yazarlar araçlar için seyahat süresini azaltarak bu süreden kazanç elde etmişlerdir.

(Yetis & Cakici, 2015, s. 11-13)’de yazarlar seyahat sürelerine etkisi üzerinde Webster hesaplama yöntemi ve sezgisel yaklaşımları karşılaştırmışlardır. Kullanılan sezgisel yaklaşımların optimum sinyal süreleri, Webster yöntemine göre daha iyi seyahat süresi sağladığı görülmüştür. Yazarlar 14 farklı senaryo üzerinde sezgisel yaklaşımları uygulayarak seyahat süresi eniyileme çalışması gerçekleştirmişlerdir.

Yazarlar (Akgüngör, Yılmaz, Korkmaz, & Doğan, 2019, s. 259-269)'te ise Webster hesaplama yöntemini kalibre ederek seyahat problemlerinde daha verimli sonuçlar elde etmek için çalışmalarını gerçekleştirmişlerdir. Trafik seyahat sürelerinin eniyilemesini kalibre edilmiş Webster üzerinden gerçekleştiren yazarlar Yapay Arı Kolonisi Algoritması ve Farksal Gelişim Algoritması kullanarak model iyileştirmesi gerçekleştirmişlerdir.

(Oztemiz & Yeroglu, 2018, s. 1-6)'te Karınca Koloni Algoritması (KKA) sezgisel yaklaşımını kullanarak Malatya ili özelinde merkez noktasında oluşan trafik yoğunluğuna çözüm bulunması amaçlanmıştır. Çalışmada sezgisel yaklaşımdan çıkan verilere göre araçlara en uygun rota bilgileri iletilerek seyahat süresinden kazanç elde edilmiştir.

Ö. Kart ve arkadaşları (Kart, Genç, & Basciftci, 2021, s. 411 - 416)'da trafik senaryosu üzerinde, yeşil dalga koridoru yöntemini kullanarak araçların hız değerlerini, kavşaklara yerleştirilen denetleçler yardımı ile anlık olarak değiştirerek dinamik bir akıllı ağ yapısı oluşturmayı amaçlamışlardır. Nesnelerin interneti ve hız verilerini baz alan çalışma ile dinamik olmayan denetleçlere göre daha iyi seyahat süreleri elde edilmiştir.

Y. Çelik ve H. Soy ise (Çelik & Soy, 2014, s. 467-474)'de seyahat süresinin eniyilemesinin yapılabilmesi için öz-örgütlenme tabanlı akıllı trafik ağı elde etmişlerdir. Öz-örgütlenme her bir aracın basit kuralları takip ettiği ve yakınında bulunan diğer araçlardan elde ettiği yerel bilgilere göre kararlar alabilme yöntemidir.

İncelenen literatür çalışmalarına göre, trafik seyahat sürelerinin eniyileme çalışmalarında akıllı sistemlerin sabit süreli trafik denetleçlerine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Sabit süreli trafik denetleçlerin akan trafik içerisinde değişken yapıya ayak uyduramadığı ve karmaşık ağlarda başarısızlığı dikkat çekmiştir. Çalışmalarda sezgisel algoritmaların, derin öğrenme algoritmalarının, bulanık sistemlerin ve kullanılan benzetim programlarının seyahat problemini çözmek için oldukça başarılı sonuçlar verdiği bildirilmiştir.

Bu çalışmada, Allsop ve Charlesworth'in (Allsop & Charlesworth, 1977, s. 262-264)'de tanımladığı trafik yol ağı problemi ele alınmış olup DE sezgisel yaklaşımı ile araçların seyahat süresini azaltmak amaçlanmaktadır. Allsop ve Charlesworth trafik ağının SUMO benzetim programı kullanılarak modellenmesi yapılmıştır. Yol ağında bulunan altı kavşak için bu kavşakları denetleyen 6 farklı trafik denetleci model içerisinde kullanılmıştır. Benzetim programı yardımıyla belirlenen trafik senaryosu için gerçek dünyaya uygun çıktılar elde edilerek çıktı sonuçlarına göre seyahat süresi karşılaştırmaları yapılmıştır. Eniyileme öncesi ve sonrası olarak trafik senaryoları için görsel olarak da karşılaştırma yapılmıştır.

2. KULLANILAN YAZILIM, YÖNTEMLER VE EK PROGRAMLAR

Bu bölümde çalışmada kullanılan işletim sistemlerine, yazılımların içeriğine, benzetim programının işleyiş yapısına, sezgisel yaklaşım ile benzetim programının tümleşik hale getirilmesi işlemleri ve kullanılan ek programlar anlatılmaktadır.

2.1. Kullanılan Programlama ve İşaretleme Dilleri

Bu alt bölümde çalışma içerisinde sezgisel yaklaşımların gerçekleştirilmesi için kullanılan programlama dili ve benzetimin gerçekleştirilmesi için kullanılan işaretleme dili anlatılmıştır.

2.1.1. Python programlama dili

Python, Guido V. R. isimli Hollandalı yazılımcı tarafından geliştirilmiş programlama dilidir. Platformlara entegre edilebilirliği ve popülerliği yüksek olan bu programlama dili sezgisel yaklaşımlarımızı kodlamak ve SUMO programının altyapısında kullanılabilirliği yönünden temel programlama dili olarak belirlenmiştir.

SUMO çıktı verilerinin elde edilmesinde ve XML dosyalarının birleştirilmesinde Python dilini desteklediğinden temel programlama dili olarak belirlenmiştir. Linux ve Windows işletim sistemlerinin SUMO ile uyumlu olması amacıyla Python uygulamalarının en son sürümlerinin kurulu olması gerekmektedir.

2.1.2. Genişletilebilir işaretleme dili (XML)

XML yapısı gereği SUMO programının temelini oluşturan internet üzerinden veri alışverişini sağlayan bir işaretleme dilidir. SUMO programı XML yapısı ile oluşturulan araç, yol ve trafik ışıklarını interneti kullanarak ana veri tabanında işleyerek benzetimi gerçekleştirmektedir. Entegre edilebilirliği açısından çalışma içerisinde bu dosyalar ve hiyerarşik yapıları işaretleme dili kullanılacaktır.

2.2. Çalışma İçin Gerekli İşletim Sistemleri

Bu alt bölümde, çalışmanın gerçekleştirilmesi için hem kodlama çalışmalarının hem de benzetim programlarını kullanabilmek için gerekli işletim sistemlerine yer verilmiştir.

2.2.1. Windows işletim sistemi

Günlük bilgisayarlarımızda genel olarak kullandığımız Windows işletim sistemi çalışmada kullanılan SUMO programının görsel olarak analiz edilebileceği ve kodlama derleyicilerinin kurulması adına ana işletim sistemi olarak kullanılmıştır.

2.2.2. Linux işletim sistemi

Öncelikli işletim sistemi olarak Windows kullanılmaktadır ancak SUMO programı içerisinde sonuç alma, yol haritası oluşturma gibi işlemlerin Linux işletim sisteminde daha efektif bir şekilde yapıldığı görülmüştür. Sanal bir makine yardımıyla Linux sistemi içerisinde SUMO kodlarını kullanmak ve sonuç verilerini daha hızlı bir şekilde almak için ücretsiz olarak dağıtımı gerçekleştirilen Linux işletim sistemi de çalışma içerisinde kullanılmıştır.

2.3. Çalışmada Kullanılan Kaynak Kod Düzenleyici

Sezgisel yaklaşımların kodlanabilmesi, SUMO programının çalışabilmesi için gerekli olan XML dosyalarının oluşturulması ve sonuçların elde edilebilmesi için kullanılan Python ve diğer kodlama dillerini derleyebilen kod derleyicisine ihtiyaç duyulmaktadır. Bu çalışmada kod ve işlemler için ara yüzü sade, derleme işlemleri için hızlı çalışan Microsoft'un geliştirdiği “Visual Studio Code” programı kullanılmıştır.

2.4. SUMO Benzetim Programı

SUMO Alman Havacılık ve Uzay Merkezi ve toplulukta bulunan kullanıcılar tarafından geliştirilmiş ve geliştirilmeye devam eden bir benzetim ortamıdır. İlk sürümü 2001 yılında kullanıcılara sunulan SUMO programı “<https://www.eclipse.org/sumo/>” web adresinden indirilebilmektedir. Çeşitli trafik senaryolarının oluşturulabileceği ve görsel olarak benzetiminin gerçekleştirilebileceği açık kaynak kodlu ücretsiz olarak sunulan bir yazılım ortamıdır. Büyük trafik senaryolarının yanı sıra, küçük trafik ağlarının modellenebileceği ve modellenen senaryolar için sonuç çıktısı veren SUMO XML işaretleme dilini kullanmaktadır. Başlangıç ve bitiş noktaları verilen araçlar için gerçek zamanlı sonuç çıktısı veren SUMO benzetim programı programlama dillerine kolayca uyarlanabilmesi sebebiyle diğer benzetim programlarının önüne geçmektedir. Bu sebeple bu çalışmada benzetim ve trafik ağı modellemesinde kullanılması tercih edilmiştir. Aşağıdaki alt başlıklarda SUMO hakkında daha kapsamlı bir tanıtım sunulmaktadır.

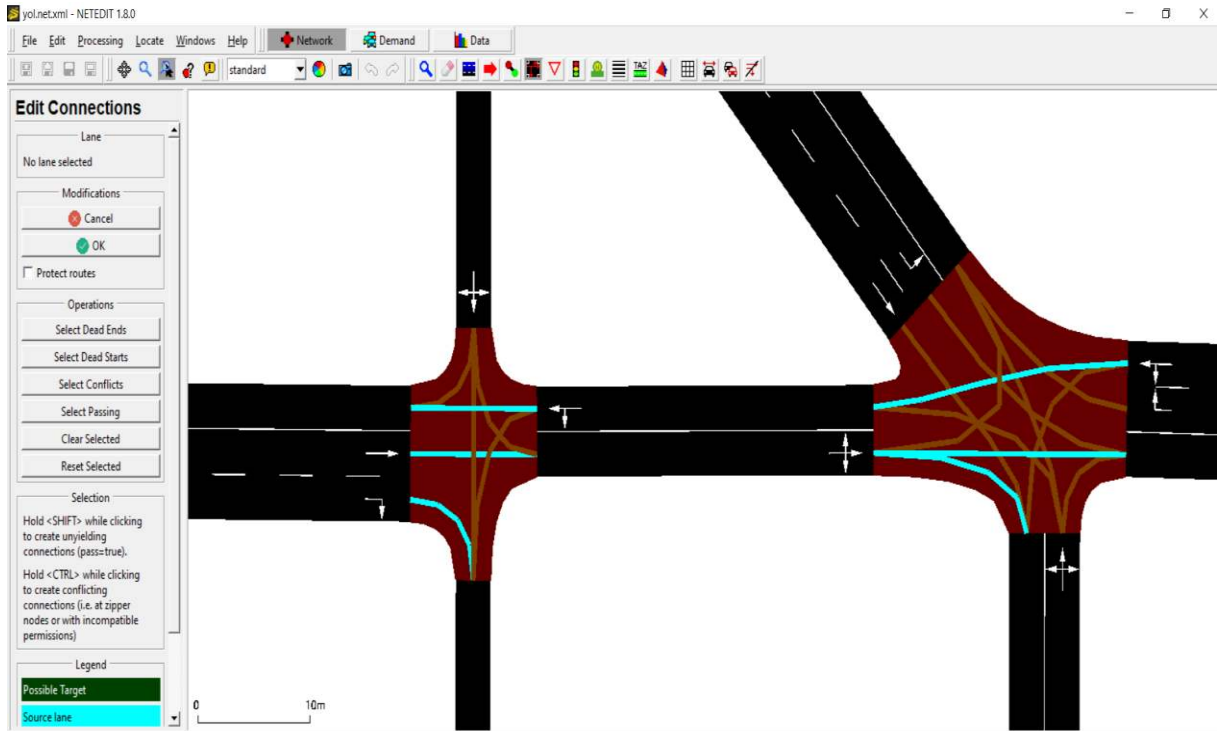
2.4.1. Trafik yol ağı tanımlama işlemleri

XML dosyalarını kullanan SUMO trafik ağı için yol noktalarını tek tek “*yol.net.xml*” isimli dosya içerisinde tutmaktadır. Bu dosya tüm yol bağlantı noktaları için yolun id değerini, yol uzunluğu, hangi şeritlerin yol bağlantısı içerisinde bulunduğunu ve bu yol içerisinde araçların hız değerlerini gösteren dosyadır. Kod 2.1’de “*yol.net.xml*” dosyası içerisinde bulunan “n1112” id değerine sahip yol için örnek yol bağlantı kodu verilmiştir.

```
<edge from="gneJ64" id="n1112" length="1000.00" priority="-1" to="4-light">
<lane id="n1112_0" index="0" length="1000.00" shape="-10.21,76.21 -48.70,77.18" speed="13.89"/>
<lane id="n1112_1" index="1" length="1000.00" shape="-10.29,73.01 -48.79,73.98" speed="13.89"/>
</edge>
```

Kod 2.1. SUMO’da örnek yol bağlantı kodlaması (“n1112” yolu)

Yol bağlantılarının tanımlanması SUMO programının eklentilerinden olan “*NetEdit*” uygulaması ile görsel olarak da yapılabilmektedir. Şekil 2.1’de örnek bir yol tanımlaması için *NetEdit* ara yüz görselini görebilirsiniz.



Şekil 2.1. *NetEdit* ile oluşturulmuş örnek yol tanımlarının yapıldığı ara yüz görseli

2.4.2. Trafik ışıklarını tanımlama işlemleri

Trafik sinyallerini tanımlama işlemleri de “*yol.net.xml*” içerisinde yolların tanımlanmasından sonra id değerleri baz alınarak yapılmaktadır. Kod 2.2’de “*6-light*” id numarasına sahip trafik denetleci için tanımlamaları görebiliriz. Koda göre örnek olarak “*duration*” alanı ilk fazda 50 saniye yeşil ışık süresi yandığını göstermektedir. “*State*” alanı faz düzenlemelerinin ve yol bağlantılarında bağlantılı yeşil sürelerine denk gelmektedir.

```

<tlLogic id="6-light" offset="0" programID="0" type="static">
  <phase duration="50" state="GGrrrrGGG"/>
  <phase duration="50" state="rrGGGGrrr"/>
</tlLogic>

```

Kod 2.2. SUMO’da örnek trafik denetleç tanımlaması (“6-light” trafik denetleci)

Kod 2.3’te ise örnek olarak tanımlanan 6 numaralı trafik ışığı için bağlantı noktaları ve hangi yollara, hangi şeritlere etki edeceğini belirten XML dili ile yazılmış hiyerarşik yapı verilmiştir.

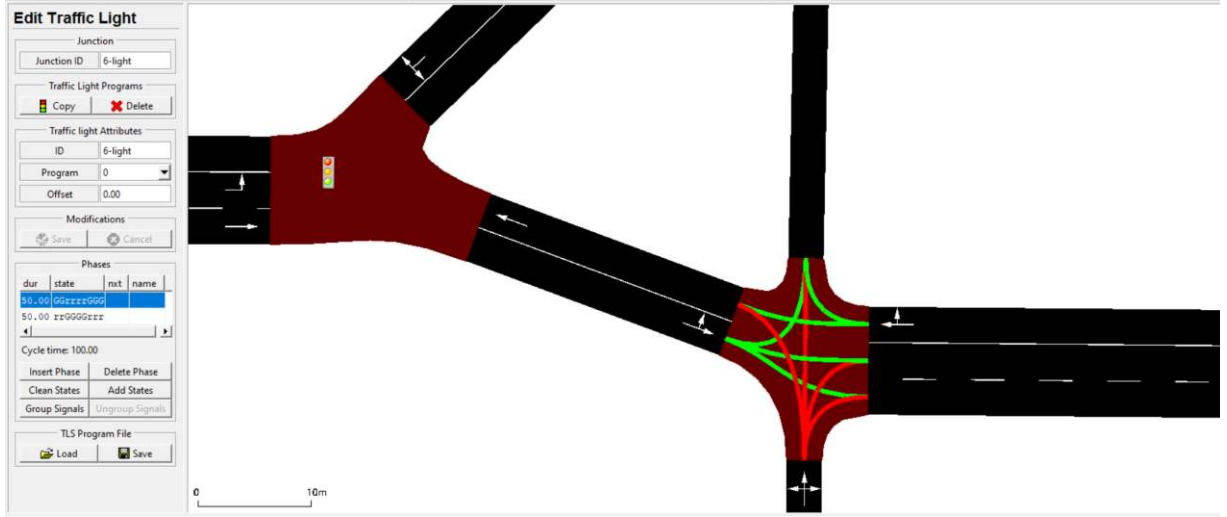
```

<junction id="6-light" incLanes="n18_0 n22_0 n7_0"
intLanes=":6-light_0_0 :6-light_1_0 :6-light_2_0 :6-light_3_0 :
6-light_4_0 :6-light_5_0 :6-light_6_0 :6-light_6_1 :6-light_8_0"
shape="-176.36,80.42 -173.16,80.36 -172.75,78.04 -172.21,77.23
-171.44,76.65 -170.44,76.29 -169.21,76.16 -169.30,66.56 -171.52,66.14
-172.30,65.59 -172.86,64.82 -173.19,63.82 -173.30,62.60 -176.50,62.60
-176.66,64.81 -177.16,66.74 -177.98,68.41 -179.13,69.81 -180.61,70.95
-182.41,71.81 -180.18,77.81 -178.09,77.48 -177.35,77.77 -176.82,78.35
type="traffic_light" x="-176.50" y="73.03">
  <request cont="0" foes="100010000" index="0" response="000000000"/>
  <request cont="0" foes="100110000" index="1" response="000000000"/>
  <request cont="0" foes="011000000" index="2" response="011000000"/>
  <request cont="0" foes="011000000" index="3" response="011000000"/>
  <request cont="0" foes="111000011" index="4" response="111000011"/>
  <request cont="0" foes="111000010" index="5" response="111000010"/>
  <request cont="0" foes="000111100" index="6" response="000000000"/>
  <request cont="0" foes="000111100" index="7" response="000000000"/>
  <request cont="0" foes="000110011" index="8" response="000000011"/>
</junction>

```

Kod 2.3. SUMO’da trafik denetleçlerinin yol bağlantı tanımlarının XML içeriği

Trafik sinyallerinin kodlama ve işaretleme dili ile yapılabileceği gibi Şekil 2.2’de *NetEdit* aracı ile oluşturulabilir. “6-light” numaralı trafik sinyali için tanımlamalarına görsel üzerinden ulaşılabilir. Sol tarafında bulunan trafik ışık düzenleyicisi ile faz düzenlemeleri, sinyal süresi düzenlemeleri ve şeritler için öncelik verilen yol tanımlamaları yapılabilmektedir.



Şekil 2.2. NetEdit ile oluşturulmuş trafik denetleç görseli

2.4.3. SUMO’da araç tanımlamaları

SUMO benzetim programı içerisinde araçların hangi yönde ilerleyeceklerini, yolları ve id değerlerinin belirlendiği XML dosyası “*yol.rou.xml*” dosyadır. Kod 2.4’te id değeri “1-2-1” olan aracın hangi yollardan geçeceğini “*edges*” alanından ve başlama anını belirten “*depart*” özelliğini görebiliriz.

```
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes_file.xsd">
  <vehicle depart="0" id="1-2-1"><route edges="n12 n3 n222"/>
</vehicle>
```

Kod 2.4. SUMO’da örnek araç tanımlama kodu (“1-2-1” aracı için)

2.4.4. SUMO birleşim dosyası tanımlamaları

SUMO, oluşturulan araç, yol ve trafik ışıklarının tanımlamaları ve XML dosyalarını birleştirmek ve yapıyı tanımlamak için “*sumoconfig.sumo.cfg*” dosyasına ihtiyaç duymaktadır. Kod 2.5’te bu çalışmada kullanılan ayarları barındıran XML dosyasının içeriğine yer verilmiştir. Giriş değeri olarak yol bağlantılarının ve trafik sinyallerinin değerlerinin bulunduğu “*yol.net.xml*” dosyasını almaktadır. Araçların rota bilgilerini barındıran “*yol.rou.xml*” dosyasının yolu veya isimlendirmesi yazılmaktadır. XML içerisinde bulunan “*time*” değeri araçların hangi zamanda yola girebileceklerini, görünüm ayarlarının yapıldığı “*viewsettings*” alanı ise programın görsel olarak analizinin yapılacağı ekran için genel ara yüz ayarlarını belirtmektedir.

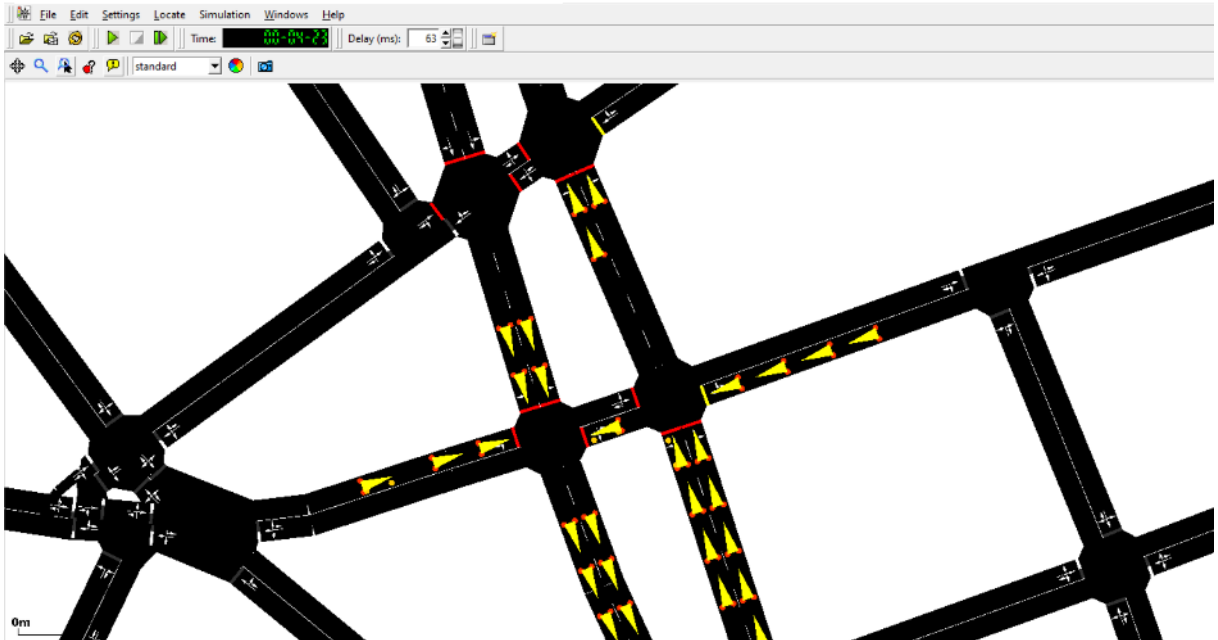
```
<configuration>
<input>
<net-file value="yol.net.xml"/>
<route-files value="yol.rou.xml"/>
</input>
<time>
<begin value="0"/>

</time>
<viewsettings>
<viewport y="0" x="250" zoom="100"/>
</viewsettings>
</configuration>
```

Kod 2.5. “Sumoconfig.sumo.cfg” dosyası XML içerik görseli

2.4.5. SUMO’da trafik senaryolarının kořturulması

Oluřturulan XML dosyaları ve ayar dosyası “*cfg*” uzantılı dosya ile aılarak SUMO’nun grsel olarak bizlere sunduėu kendi ara yz ile oluřturulan trafik senaryosunu grebiliriz. “*SUMO-GUP*” ierisinde bulunan “*play*” butonu ile araların yollar zerinde ilerlediėini ve trafik ıřıklarının belirlenen deėerlere gre alıřtıėını gzlemleyebiliriz. řekil 2.3’te SUMO-GUI rnek senaryo kořturulması sırasında anlık bir grnm rneėi verilmiřtir.



řekil 2.3. SUMO ara yz ve rnek trafik senaryosu grseli

2.4.6. SUMO’da analiz ve çıktı sonuçlarının elde edilmesi

SUMO, Python komutlarının ve komut istemini kullanarak belirli fonksiyonlar yardımıyla XML dosyalarını işleyerek anlamlı sonuç verileri sağlamaktadır. Bu çalışmada kullanılan “*tripinfo*” SUMO fonksiyonu bizlere aracın toplam seyahat süresini, aracın senaryo içerisinde bulunduğu giriş-çıkış bilgilerini, seyahat süresi boyunca bekleme zamanları gibi birçok veriyi sunmaktadır. SUMO’nun sağlamış olduğu çıktıyı almak için Python kodlama çalışması içerisinde yazılmış “*simulate*” fonksiyonuna Kod 2.6’da verilmiştir.

```
def simulate():
    cfg = os.path.join(root, "sumoconfig.sumo.cfg")

    output_xml = os.path.join(root, "sonuc.xml")

    subprocess.call("sumo -c {} --tripinfo-output {}".format(cfg, output_xml))
```

Kod 2.6. SUMO’da “*tripinfo*” komutu ile analiz sonuçlarının elde edilmesi

Kod 2.7.’de ise “*tripinfo*” komutunun çalışması sonucu elde edilen XML formatında elde edilen çıktı örneği gösterilmiştir. Şekle göre id değeri “1-2-1” numaralı araç için “n12_0” yolundan harekete başladığını ve “n222_0” yolundan seyahatini tamamladığını görebiliriz. Aynı şekilde yol uzunluğu olarak 776.16, 5.61 dakika zaman kaybı yaşadığı ve 59 dakikada yolculuğunu tamamladığı bilgilerine ulaşabiliriz.

```
<tripinfos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation
<tripinfo id="1-2-1" depart="0.00" departLane="n12_0" departPos="5.10"
departSpeed="0.00" departDelay="0.00" arrival="59.00" arrivalLane="n222_0"
arrivalPos="250.00" arrivalSpeed="13.66" duration="59.00" routeLength="776.16"
waitingTime="0.00" waitingCount="0" stopTime="0.00" timeLoss="5.61" rerouteNo="0"
devices="tripinfo_1-2-1" vType="DEFAULT_VEHTYPE" speedFactor="1.06" vaporized=""/>
```

Kod 2.7. SUMO analiz veri örnek görseli (“1-2-1” numaralı araç için)

2.5. Farksal Gelişim Algoritması

Farksal Gelişim Algoritması (DE), 1996 yılında Storn ve Price tarafından keşfedilen popüler ve yaygın olarak kullanılan bir metasezgisel algoritmadır. DE algoritmasının en önemli özelliği, GA (Abidin, 2016, s. 11-22) gibi işlemleri kullanarak deneme parametre vektörleri oluşturmastır. DE, üçüncü bir birey eklemek için bir popülasyondan rastgele seçilen iki birey arasındaki farkı kullanır.

Sürü yapısı sezgisel modelini kullanan DE, içerdiği parametreler ve amaç fonksiyonu ile problemin eniyilemesi için kullanılmıştır. Doğrusal olmayan problemlerin çözümünde kullanılan DE, kodlanabilirliği kolay ve problemlere entegre edilebilirliği oldukça başarılıdır.

Çalışmada kullanılan sezgisel algoritma kararlı olması yönüyle diğer algoritmaların önüne geçmektedir. Koşturma sayısı değişse de genel olarak en iyi çözüm kümesine yaklaşık sonuçlar elde edilmektedir. Bu çalışmada temel olarak ele alınan sezgisel yaklaşım olan DE algoritmasının kaba kodu Algoritma 2.1'de verilmiştir.

Algoritma 2.1 Diferansiyel Gelişim Algoritması Kaba Kodu

Oluştur Başlangıç popülasyonu ($N \geq 4$)

Ayarla Ölçekleme faktörü $F \in [0, 2]$, çaprazlama oranı $CR \in (0, 1)$

Do (Ana Döngü)

For popülasyonun her bir x_i bireyi için

Seç rastgele üç sayı ($r_1, r_2, r_3 \in [1, D]$ ve $r_1 \neq r_2 \neq r_3 \neq i$)

Oluştur rastgele bir tam sayı $j_{rand} \in [1, D]$

For her x_i bireyi için

$$u_{i,j} = \begin{cases} x_{r1,j} + F(x_{r2,j} - x_{r3,j}) & \text{if } (rand \leq CR \vee j = j_{rand}) \\ x_{i,j} & \text{Değilse} \end{cases}$$

End For

Değiştir aday u_i , x_i ' den daha iyi ise

End For

Sonlandır koşul sağlanana kadar

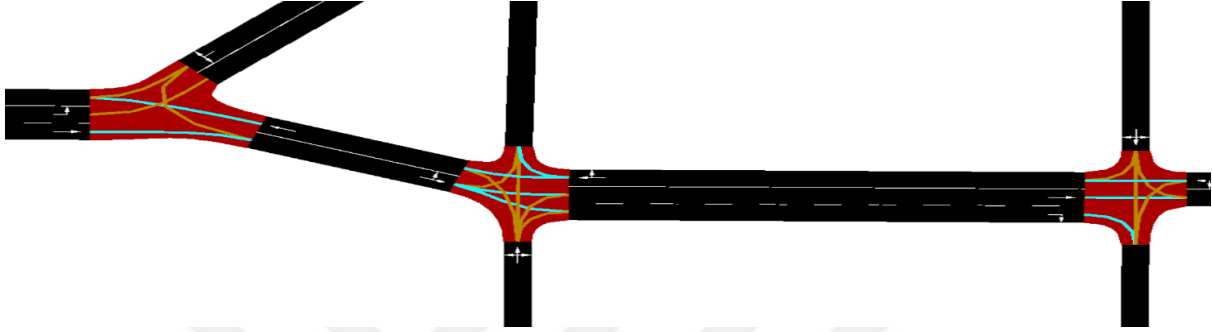
Çözüm: En iyi cost değerine sahip olan birey problemin çözümü olacaktır.

Algoritma 2.1. DE algoritması kaba kodu (Karakuzu C. , 2017, s. 4706 - 4721)

Sürü büyüklüğü (NP), arama uzay boyutu (D), çaprazlama oranı (CR), nesil sayısı (G) ve ölçekleme faktörü (F) gibi parametreleri kullanan DE bu çalışmada iki farklı şekilde kullanılmaktadır. Tek bireyli ve çok bireyli sürü olarak DE koşturulmuştur. Koşturma sonrasında iki farklı yöntem sonuçları itibariyle hem birbirleri arasında karşılaştırılmıştır hem de koşturma öncesi veriler ile karşılaştırılmıştır. Belirlenen amaç fonksiyonuna göre aç gözlü seçme işlemi kullanılan DE mutasyon ve rekombinasyon işlemleri ile yeni bireyler üzerinden eniyileme yapan sezgisel optimizasyon yöntemidir.

3.2. Trafik Sinyallerinin Faz Düzenlemeleri

Trafikte bir kavşakta, en az bir yol için geçiş hakkı verilen toplam süre “faz” olarak adlandırılmaktadır. Yol ağında bulunan trafik sinyallerinin faz düzenlemeleri ve yol yapısı [1]’de belirtildiği gibi 1-2-3 ve 6 bağlantı noktalarında bulunan kavşaklar için 2 faz, 4. ve 5. kavşaklar için ise 3 fazdan oluşan trafik sinyal tipleri kullanılmıştır. Şekil 3.2’de örnek olarak verilen 6 numaralı kavşak için trafik sinyallerinin yol ağına etkisi ve Şekil 3.1’de belirtilen oklar ile araç yönlerinin bağlantılarının görseline yer verilmiştir.



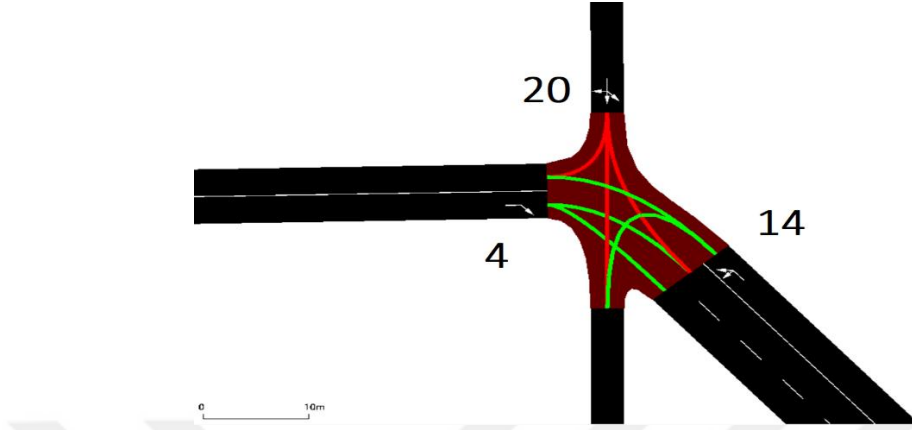
Şekil 3.2. SUMO’da trafik sinyali ve yol bağlantıları görseli

Tablo 3.1’de trafik sinyallerinin faz değerleri verilmiştir. Modellenen trafik ağında trafik denetleçleri 2 ve 3 fazlı tipte her kavşak için belirli anlarda hangi yollar için yeşil ışık veya kırmızı ışık verileceği gösterilmiştir. Tabloya göre 2 numaralı kavşakta ilk fazda 3 ve 15 numaralı yollar için yeşil ışık verilmekte olup ikinci fazda ise 23 numaralı yol için geçiş yapılacağı belirtilmiştir. Yine aynı şekilde 3 numaralı bağlantı noktası için ilk durumda 20 numaralı yola geçiş önceliği verilmekte olup ikinci fazda 4 ve 14 yollarına geçiş hakkı verilmektedir. 4 numaralı kavşak için ise ilk fazda 5,11 ve 12 numaralı yol için, ikinci fazda 10 ve 12, üçüncü fazda ise 5, 6 ve 13 numaralı yollar için geçiş hakkı verilmektedir. 5 numaralı kavşak için ise ilk fazda 8 ve 9 numaralı yol için, ikinci fazda 8 ve 17, üçüncü fazda ise 23 numaralı yol için geçiş hakkı verilmektedir.

Tablo 3.1. Trafik sinyalleri ve faz tablosu

FAZ \ KAVŞAK	1	2	3	4	5	6
Durum 1	1, 16	3,15	20	5,11,12	8,9	7,18
Durum 2	1,2,19	23	4,14	10,12	8,17	22
Durum 3	-	-	-	5,6,13	21	-

Şekil 3.3'te tabloda verilen 3 numaralı kavşakta 4 ve 14 yolları için yeşil faz süresi verilirken 20 numaralı yol için kırmızı ışık süresi verilen SUMO benzetim görseline yer verilmiştir.



Şekil 3.3. Trafik ağında bulunan 3 numaralı kavşak için SUMO faz benzetimi

Benzetim için oluşturulan 3 fazlı, 6 adet kavşak için trafik sinyali benzetim gerçekleştirilmesi için yeşil sinyal zamanları Tablo 3.2’de verilmiştir. Tabloyu inceleyecek olursak 2 numaralı kavşak için ilk fazda 20 saniye yeşil sinyal süresi verilmekte ve 2. fazda ise 50 saniye yeşil sinyal süresi verilmektedir. 4 ve 5 numaralı kavşaklar için 3 fazdan oluştuğundan dolayı Yeşil-3 faz süresi belirtilmiştir. Tabloda C(s) ile belirtilen çevrim süresi yeşil sinyal sürelerinin toplamını vermektedir. Örneğin 6 numaralı trafik denetleci için faz yeşil sinyal değerleri 30, 40 saniye olduğundan C(s) çevrim süresi bu iki sürenin toplamı olan 70 saniye olarak belirtilmiştir

Tablo 3.2. Fazlara göre belirlenmiş trafik sinyal süreleri

Toplam Işık Süresi C(s)	Trafik Işığı (Kavşak) Numarası (i)	Faz-1 Yeşil-1	Faz-2 Yeşil-2	Faz-3 Yeşil-3
50	1	30	20	-
70	2	20	50	-
80	3	30	50	-
120	4	40	40	40
110	5	20	40	50
70	6	30	40	-

4. KULLANILAN SEZGİSEL YÖNTEM

Bu bölüm, 2.5. nolu alt bölümde tanımlanan DE algoritması ile yol-kavşak sinyal süresi eniyileme problemini çözmek amacıyla geliştirilmiş değer fonksiyonunun, sürü geliştirme işlemlerinin ve trafik sinyalleri için dengeleme işlemlerinin hangi yöntemler ve işlevler kullanılarak yapıldığının anlatıldığı bölümdür.

4.1. Değer Fonksiyonu

DE algoritmasında belirlenen değer fonksiyonun amacı sürüdeki bireylerin çözüm kalitesinin sayısallaştırılarak algoritmanın işleyişi sonucu seyahat süreleri en az olacak şekilde çözüm sunan geliştirilmiş bir çözüm elde etmektir. Bu çalışma için belirlenen değer fonksiyonu Denklem 4.1’de verilmiştir.

Denklem T konumundan seyahate başlayan tüm araçların seyahat sürelerinin varış yerlerine ulaşan araç sayılarının toplam araç sayısına göre ağırlıklandırılmış toplamından oluşturulmuştur. Araçların ortalama seyahat sürelerini baz alan değer fonksiyonu her bağlantı noktasında bulunan kavşaklar için tek tek hesaplanarak belirtilen J değer fonksiyonunun sonuç değerine ulaşmaktadır.

$$J(T) = \sum_i^{\text{Çıkış}} (T' \text{ den } i' \text{ ye ulaşma süresi}) * \frac{(i' \text{ ye giden araç sayısı})}{(T' \text{ den çıkan araç sayısı})} \quad (4.1)$$

Denklem 4.1’de i değeri T konumundan başlayan tüm araçların çıkış olarak kullanabileceği konumları belirtmektedir. T konumu başlangıç noktası olarak belirtilmiş olup, J(T) değeri ise değer fonksiyonunun çıktı sonucunu temsil etmektedir. Örnek olarak C konumundan çıkan araçların değer fonksiyonundan dönecek sonucu bulmak için; C konumundan giriş yapan araçların B, A, F, E ve D noktalarından çıkış yapacak araçların sürelerinin tek tek bu noktalara giden araçlar ile çarpılarak C konumundan çıkan tüm araçların sayısına bölünmesiyle bulunabilmektedir.

Kod 4.1.’de Python kodlama dili kullanılan “sonuc.xml” dosyası içerisinde SUMO’dan alınan analiz verileri kullanarak değer fonksiyonu için sonuç elde edilmesi için yazılan kodlama görseli verilmiştir. “Sonuc.xml” dosyası içerisinde tüm araçların seyahat sürelerini alarak bu değerlerin id değerlerine göre ayrımı anlamlı veri oluşturmaktadır.

```

def analyze():
    totalduration = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]

    output_xml = os.path.join(root, "sonuc.xml")
    mytree = ET.parse(output_xml)
    myroot = mytree.getroot()
    #print(myroot)
    for child in myroot:
        id = child.attrib['id'].split('-')
        duration = child.attrib['duration']
        #print(id[0],id[1],id[2],duration )
        id1=int(id[0])-1
        id2=int(id[1])-1
        duration = float(duration)
        #print(duration)
        totalduration[id1][id2]=totalduration[id1][id2] + duration

    for i in range(0,5):
        for j in range(0,5):
            if arac[i][j] !=0:
                totalduration[i][j] =round(totalduration[i][j]/arac[i][j]/60,2)
    return totalduration

```

Kod 4.1. Değer fonksiyonun Python ile oluşturulması kod kümesi

4.2. DE Algoritmasında Sürü Oluşturma Yöntemleri

Sürü yapısı kullanılan DE algoritması için bu çalışmada iki farklı yöntem kullanılmıştır. Bu bölümde tek bireyli ve çok bireyli sürüler için geliştirme yöntemleri anlatılmıştır.

4.2.1. Tek birey ile çözüm geliştirilmesi

Tek birey ile çözüm geliştirme yöntemi ile 2 fazdan oluşan 4 kavşak için 8 birey elemanı ve 3 fazdan oluşan 2 kavşak için 6 birey elemanı olmak üzere toplamda 14 boyutlu tek bireyden oluşan DE ile sezgisel bir yaklaşım bu çalışmaya özgü tanımlanmıştır. Bu yaklaşımda, trafik denetleçlerinde bulunan tüm yeşil sinyal süreleri bireyde birer eleman olarak belirlenmiştir.

Kod 4.2’de Python ile örnek olarak oluşturulmuş bir birey örneği gösterilmiştir. Matrisin ilk kolonunda bulunan değerler C(s) değerine karşılık gelmekte ve trafik sinyal sürelerinin dengeleme işlemlerinde kullanılmaktadır. 2, 3 ve 4. kolonlar ise ilgili denetleçte yeşil sinyal süresini temsil etmektedir. Matristeki her bir satırın bir kavşak için olduğu anlaşılmaktadır.

```

light=[[100,50,50,0.],
        [100,50,50,0.],
        [100,50,50,0.],
        [120,40,40,40],
        [120,40,40,40],
        [100,50,50,0]]

```

Kod 4.2. Tek birey ile çözüm geliştirme yönteminde bireyin yapısı

DE algoritmasında alışagelmış sürü güncelleştirme işlemlerinde kullanılan matematiksel işlemler tek birey ile çözüm geliştirmesinde belirlenen bireyin uzay boyutu olarak tanımlanan elemanları içerisinde gerçekleştirilmiştir. Her bir iterasyonda güncelleme yapılacak yeni bireyin elemanları rastgele seçilen ve birbirlerine eşit olmayan üç farklı sayıya göre (r_1 , r_2 , r_3) belirlenmiştir. Rastgele seçilen “J” değeri 1 ile uzay boyutu arasında atanmıştır. Aday birey elemanı fazda karşılık gelen durumuna göre dengeleme işlemleri uygulanarak sınırlandırmalar yapılır. Daha sonra bir döngü yardımı ile belirlenen yeni aday eleman değeri bulunmaktadır. Bulunan aday eleman değeri, ana eleman değerinden daha iyi bir seyahat süresi sağlaması durumunda yeni birey elemanı olarak kabul edilerek güncelleme işlemleri gerçekleştirilmektedir.

Algoritma 4.1’de yukarıda belirtilen tek birey ile çözüm geliştirmesi yönteminin kaba koduna yer verilmiştir. Şekle göre aday birey elemanı her iterasyonda dengeleme işlemlerine tabi tutulmakta ve orantısal olarak düzenlenmektedir.

Algoritma 4.1 Tek bireyli çözüm algoritması kaba kodu

Oluştur Başlangıç bireyi ($N = 1$)

Dengele Rastgele başlangıç bireyi C_{max} , Q_{max} ve Q_{min} değerlerine göre dengelenir.

Ayarla Ölçekleme faktörü $F \in [0, 2]$, çaprazlama oranı $CR \in (0, 1)$

Do (Ana Döngü)

Eşitle Aday birey U_{light} ve ana birey X_{light}

For tek bireyin her bir x_i elemanı için ($x_i \in X_{light}$)

Seç rastgele üç sayı ($r_1, r_2, r_3 \in [1, D]$ ve $r_1 \neq r_2 \neq r_3 \neq i$)

Oluştur rastgele bir tam sayı $j_{rand} \in [1, D]$

For tek bireyin her bir x_i elemanı için

$$u_i = \begin{cases} x_{r1} + F(x_{r2} - x_{r3}) & \text{if}(rand \leq CR \vee i = j_{rand}) \\ x_i & \text{Değilse} \end{cases}$$

End For

For her bir u_i için ($u_i \in U_{light}$)

$$u_i = \begin{cases} C_{max} \leq 120 \\ C_{min} \geq 25 \\ Q_{min} \geq 8 \\ Q_{max} < C_{max} \\ Q_{min} < u_{i,j} < C_{max} \end{cases}$$

End For

Hesapla: Dengelenmiş U_{light} değer fonksiyonuna gönderilir, seyahat süresi bulunur.

If aday U_{light} , X_{light} ’ den daha iyi ise

$$X_{light} = U_{light}$$

End If

End For

Sonlandır: koşul sağlanana kadar

Çözüm: Bulunan son birey problemin çözümü olacaktır.

Algoritma 4.1. Tek bireyli çözüm algoritması kaba kodu

4.2.2. Sürü ile çözüm geliştirme

Bu çalışmada kullanılan ikinci çözüm geliştirme yaklaşımında ise alışagelmış sürü tabanlı sezgisel yaklaşımlarda olduğu gibi tüm trafik sinyal fazları bireyin bir uzay boyutu olarak alınarak 14 boyutlu 10 adet bireyli bir sürü kullanılmıştır. Her bir birey içerisinde trafik sinyalinin fazlarını barındırmaktadır.

Kod 4.3'te 18 boyutlu 2 bireyin örnek olarak değerleri verilmiştir. Bireyin ilk elamanları 1 nolu kavşak için C(s) değerini temsil etmekte ve takip eden her 3 birimde bir sırasıyla diğer kavşaklar için C(s) değerleri de birey içinde yer almaktadır. Diğer elemanlar ise kavşaktaki fazlar için yeşil sinyal sürelerine karşılık gelmektedir.

```
Pop =  
[  
[100 50 50 0 100 50 50 0 100 50 50 0 120 40 40 40 120 40 40 40 100 50 50 0 ]  
[120 60 60 0 120 60 60 0 120 60 60 0 90 30 30 30 90 30 30 30 120 60 60 0 ]  
]
```

Kod 4.3. Sürü ile çözüm geliştirmede iki bireyli sürü örneği

4.3. Trafik Sinyallerinin Dengeleme İşlemleri

DE algoritmasının işletilmesi ile birlikte elde edilen yeni aday bireyler ve aday trafik sinyal süreleri için kısıtlayıcı koşullar problemin doğası gereği belirlenmiştir. Bu koşullar bir trafik ağında yeşil sinyal süresinin belirli bir süreden düşük veya belirli bir süreden yüksek olamayacağını belirten kısıtlayıcı etkenlerdir.

Denklem 4.2'de verilen kısıtlamalar bu çalışma için temel alınan koşullara karşılık gelmektedir. Örnek yol ağı ve sinyalizasyonu için uygun olmayan değerler dikkate alınmamaktadır. Dengeleme işlemleri ile, alt ve üst sınırlar dışında kalan sürelerin gerçek hayata uygun olmayan değerler alınmasının önüne geçilmiştir.

$$\begin{aligned} C_{\max} &\leq 120 \\ C_{\min} &\geq 25 \\ Q_{\min} &\geq 8 \\ Q_{\max} &< C_{\max} \\ Q_{\min} &< Q < C_{\max} \end{aligned} \quad (4.2)$$

Denklem 4.2'de C_{max} değeri toplam döngü değerinin üst limit tanımına, C_{min} değeri toplam döngü değerinin alt limit tanımlamasına, Q değeri trafik denetlecinde yeşil sinyal süresini temsil etmektedir. Q_{min} değeri yeşil sinyal süresinin alt sınırına Q_{max} ise yeşil sinyal süresinin üst sınırını temsil etmektedir.

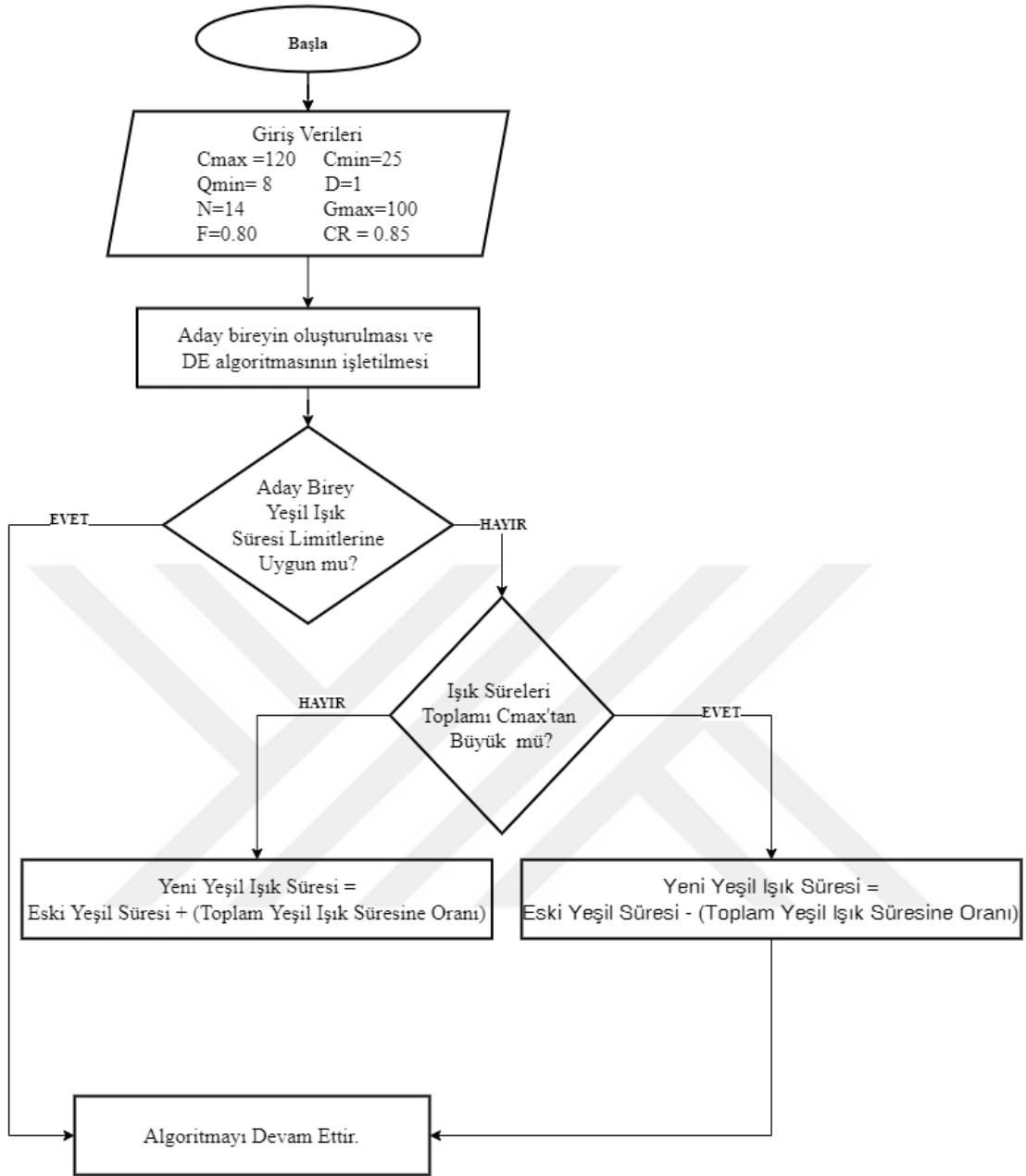
Kod 4.4.'te dengeleme işlemlerinin Python ile yapılan kodlamaya yer verilmiştir. Verilen kod kümesine göre DE algoritmasının koşması sonucu çıktı matris değeri döngüler yardımı ile her biri tek tek dengeleme işlemlerinden geçmektedir. Daha sonra çıktı sonuçları için C(s) değerleri kontrol edilerek dengeleme işlemi gerçekleştirilmektedir.

```
while Ulight[iy][0]>Cmax or Ulight[iy][0]<Cmin:
    Ulight[iy][0]=0
    for k in range(1,len(Ulight[iy])):
        Ulight[iy][0]+=Ulight[iy][k]

    if Ulight[iy][0]< Cmax and Ulight[iy][0] > Cmin:
        for k in range(1,len(Ulight[iy])):
            if k <len(Ulight[iy]) and Ulight[iy][k] !=0 :
                Ulight[iy][k] = max(Ulight[iy][k],Qmin)
                Ulight[iy][k] = min(Ulight[iy][k],Cmax)
    if Ulight[iy][0]> Cmax:
        for t in range(1,4):
            if t <len(Ulight[iy]) and Ulight[iy][t] !=0 :
                Ulight[iy][t]=Ulight[iy][t] - (((Ulight[iy][0] - Cmax) * Ulight[iy][t])/Ulight[iy][0])
                Ulight[iy][t] = max(Ulight[iy][t],Qmin)
                Ulight[iy][t] = min(Ulight[iy][t],Cmax)
    if Ulight[iy][0] < Cmin:
        for t in range(1,4):
            if t <len(Ulight[iy]) and Ulight[iy][t] !=0 :
                Ulight[iy][t]= Ulight[iy][t] + (((Cmin-Ulight[iy][0])* Ulight[iy][t])/Ulight[iy][0])
                Ulight[iy][t] = max(Ulight[iy][t],Qmin)
                Ulight[iy][t] = min(Ulight[iy][t],Cmax)
```

Kod 4.4. Trafik sinyali dengeleme kod kümesi

Şekil 4.1’de ise işletilen trafik sinyali dengeleme işlemlerinin akış şeması ile şematize edilmiş görseline yer verilmiştir. Görsele göre aday bireylerin her adımda güncellendiği ve dengeleme işlemlerinden geçtiği görülmüştür. Dengeleme işlemleri ilk olarak giriş verilerine göre bir aday birey oluşturmakta ve bu rastgele oluşturulan aday birey Denklem 4.2’de verilen koşullara uygun olarak sınırlandırma işlemlerinden geçmektedir. Elde edilen aday birey Cmax sınırlandırmasından geçirilerek orantısal olarak diğer yeşil sürelerine eklenmekte veya çıkartılmaktadır. Daha sonra dengelenen aday birey algoritmada kullanılmak üzere sürecine devam etmektedir.



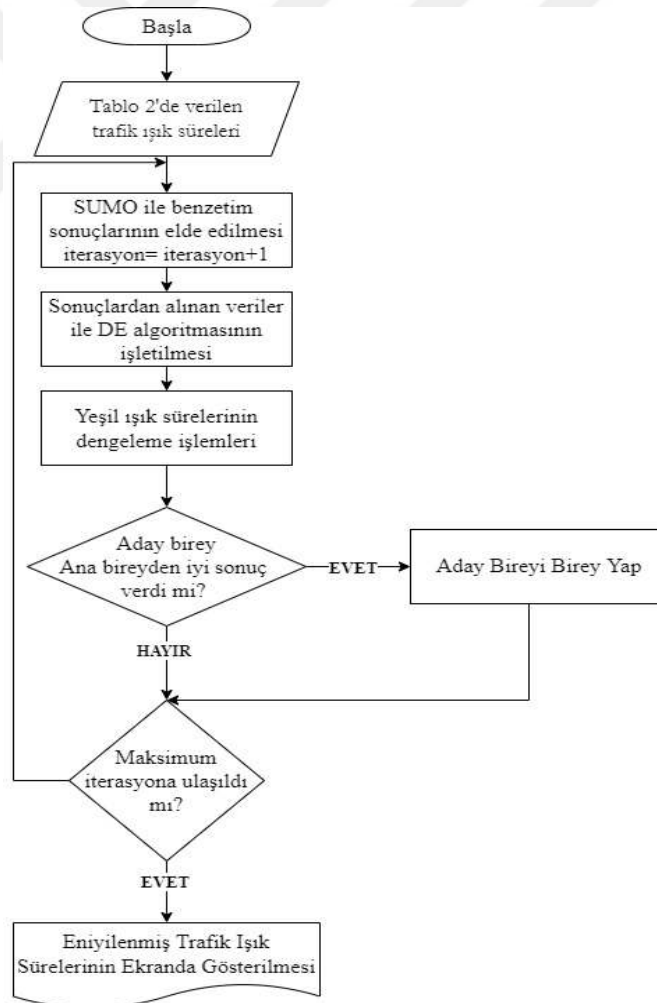
Şekil 4.1. Trafik sinyali sezgisel optimizasyon ve dengeleme işlemleri akış şeması

5. SUMO VE DE ALGORİTMASININ ENTEGRASYONU

Bu bölümde oluşturulmuş SUMO trafik ağı ve sürü tanımlamaları yapılmış DE algoritması arasında gerçekleştirilen entegrasyon çalışmalarına yer verilmiştir.

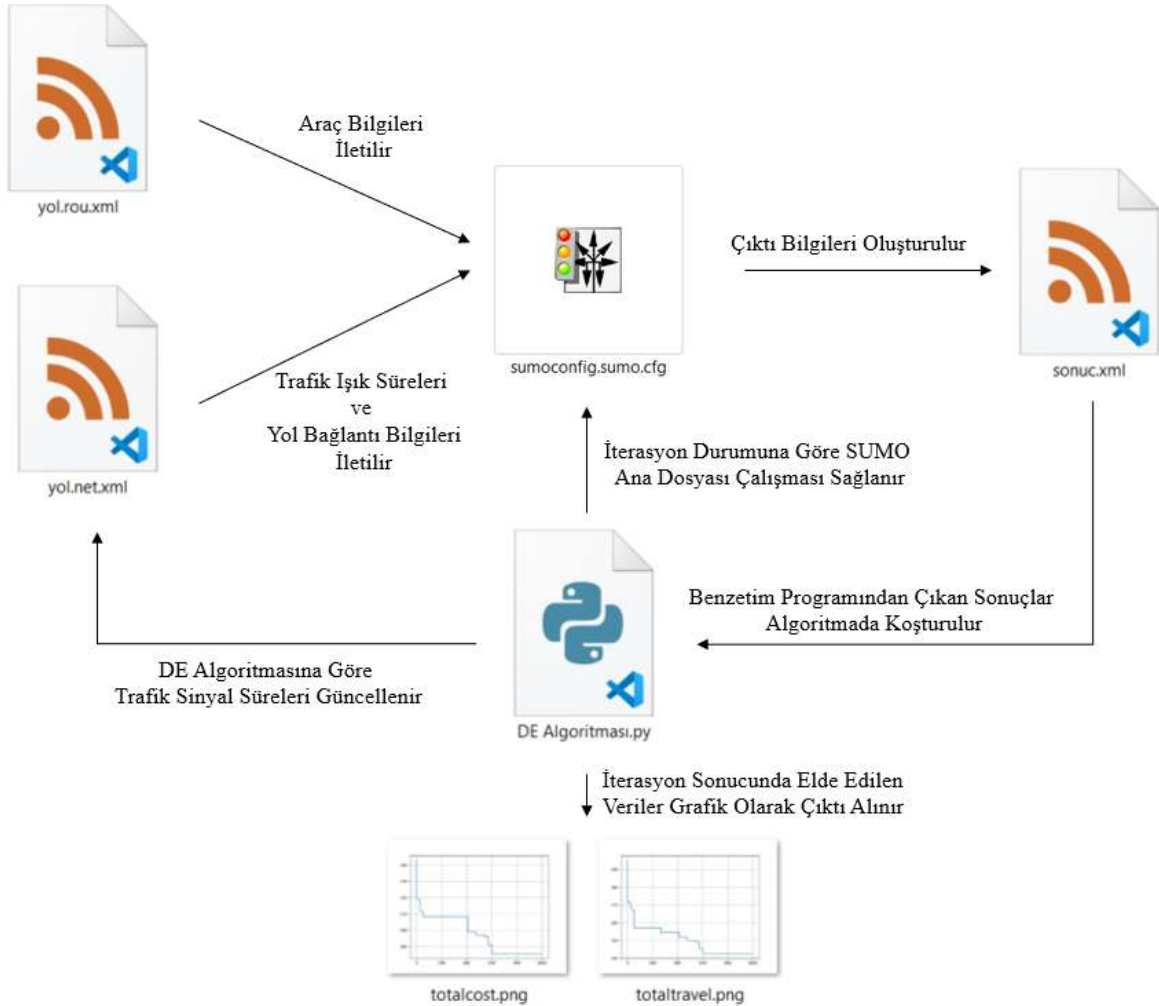
Entegrasyon işlemlerinde iki ayrı platformun birbirleriyle uyumlu şekilde çalışabilmesi için hem Python programa dilinin hem de SUMO benzetim programının ortak paydası olabilecek XML dosyalarına etki edilerek sistem bütünleştirilmesi amaçlanmıştır. XML işaretleme dili değişik uygulamalar/platformlar arasında veri alışverişi yapılabilmeye imkân vermektedir.

Şekil 5.1.'de SUMO ve DE algoritmasının bütünleşik bir şekilde işleyişinin akış şeması verilmiştir. SUMO ile başlayan süreç her iterasyonda DE algoritması ile haberleşerek 1000 iterasyon devam etmiştir. Süreç 2.40 GHz işlemcili, 16 GB RAM, Intel(R) Core(TM) i7-4700HQ CPU özellikli bilgisayarda işletilmiştir.



Şekil 5.1. SUMO ve DE algoritması entegre işleyiş şeması

Optimizasyon süresinin kullanılan birey sayısı, sürü boyutu büyüklüğü gibi etkenlere bağlı olarak artacağı ya da azalacağı Şekil 5.2.'de SUMO ve Python ile oluşturulmuş DE algoritması arasındaki etkileşim yapısı verilmiştir.



Şekil 5.2. SUMO ve DE algoritması entegrasyon ve etkileşim şeması

Şemaya göre SUMO ana dosyası olan “*sumoconfig.sumo.cfg*” dosyası tüm dosyalar ile entegre olarak XML dosyaları yardımıyla her güncelleme işlemi sonrasında çıktı üretmek için veri seti sağlamaktadır. DE algoritması Python dosyası ise iterasyon bazlı “*cfg*” dosyasının çalıştırmakta ve işlemleri gerçekleştirmektedir. SUMO tarafından üretilen çıktı dosya ise yine aynı şekilde Python dosyası ile işlenerek veri seti grafiğe dökülmekte ve trafik sinyal süreleri yeni değerler ile güncellenmektedir. Python dosyası içerisinde dengeleme işlemleri ve DE algoritmasının koşturulması gerçekleştirilmektedir.

6. ARAŞTIRMA VE BULGULAR

Bu bölümde SUMO programı ile modellenen trafik ağı için iki farklı birey yapısı ile koşturulan sezgisel yaklaşımın sunduğu çözümlerin trafikte bulunan araçların seyahat sürelerine olan etkisi incelenmiştir. Sezgisel yaklaşımda kullanılan yöntemlerin birbirleri ile ve çalışma öncesi kullanılan eniyilenmiş sinyal süreleri ile elde edilen verilerin karşılaştırmalarına yer verilmiştir.

6.1. Eniyileme Öncesi Ön Çalışmaların Gerçekleştirilmesi

Bu bölümde DE algoritmasının temel parametrelerinden olan CR ve F parametre değerlerinin üzerinde çalışılan problem için en uygun değerlerini bulmak için koşturulan sezgisel yaklaşım deney sonuçlarına yer verilmiştir.

Tablo 6.1’de en uygun CR ve F parametrelerini bulmak için yol ağında önceden belirlenmiş sinyal sürelerine yer verilmiştir. Tabloya göre 1, 2, 3 ve 6 numaralı trafik denetleçleri 2 fazdan oluşmaktadır. 4 ve 5 numaralı trafik denetleçleri ise 3 fazdan oluşmaktadır. Tabloda verilen ilk kolon ise C(s) yeşil sinyal döngü sürelerini temsil etmektedir. Örnek olarak 4 numaralı trafik denetleci için ilk fazda 40 saniye, ikinci ve üçüncü fazda da 40’ar saniye yeşil sinyal süresi verilmektedir.

Tablo 6.1. Sezgisel yaklaşımda kullanılan başlangıç trafik sinyal süreleri

Toplam Işık Süresi C(s)	Trafik Işığı Numarası (i)	Yeşil-1	Yeşil-2	Yeşil-3
100	1	50	50	-
100	2	50	50	-
100	3	50	50	-
120	4	40	40	40
120	5	40	40	40
100	6	50	50	-

Tablo 6.1’de verilen yeşil sinyal süreleri sezgisel yaklaşımın başlangıç trafik sinyal süreleri olup SUMO ortamında el ile (manuel) olarak ayarlanmıştır. Bu verilere göre işletilen trafik ağında yollar ve bağlantılar Şekil 3.3.’te verildiği gibi ve Tablo 3.2’de belirtilen yol faz matris tablosuna göre oluşturulmuştur. Örnek olarak 5 numaralı kavşak ve trafik denetleci için

ilk fazda 8 ve 9 numaralı yollar için 40 saniye yeşil sinyal süresi, ikinci fazda 8 ve 17 numaralı yollar için 40 saniye yeşil sinyal süresi ve 3. fazda ise sadece 21 numaralı yol için yeşil sinyal süresi verilmektedir. Diğer tüm durumlarda belirtilmeyen tüm yollar için kırmızı sinyal süresi verilmektedir. Buna göre örnek olarak 5 numaralı yolda 8 numaralı yol için toplam 80 saniye yeşil sinyal süresi ve 4 saniye kırmızı sinyal süresi belirlenirken 21 numaralı yol için toplamda 80 saniye kırmızı sinyal süresi ve 40 saniye yeşil sinyal süresi verilmektedir.

Tablo 6.2’te CR ve F parametre değerlerini bulmak için oluşturulan sezgisel yaklaşım ve işletilen SUMO benzetim programı içerisinde kullanılan araç sayıları ve bu araçların başlangıç-varış matrisine yer verilmiştir. Tabloya göre örnek olarak E noktasından seyahatine başlayan araçlar için A, B ve F noktalarına giden toplam 30 araç bulunmakta ve her bir çıkış noktasından 10’ar araç geçiş yapmaktadır. E noktasından başlayan araçlar için D ve E noktalarına giden araç bulunmamaktadır.

Tablo 6.2. Sezgisel parametrelerin eniyilemesi için kullanılan araç sayıları

Varış Başlangıç	A	B	D	E	F
A	0	10	10	10	10
C	10	10	10	10	10
D	10	10	0	0	10
E	10	10	0	0	10
G	10	10	10	10	10

Tablo 6.2’de verilen trafik sinyal süreleri ile işletilen SUMO trafik modeli sezgisel yaklaşım uygulanmaksızın işletilmiş ve 63.25 dakikalık ortalama seyahat süresi elde edilmiştir. Tablo 6.3’te, Tablo 6.2’deki araç sayılarına uygun olarak işletilen benzetim ile elde edilen ortalama seyahat süreleri araçların seyahate başladıkları nokta baz alınarak verilmiştir. Tabloya göre; örneğin A noktasından başlayan (B, D, E ve F’ye varan 10’ar adet) araçların varış noktalarına ulaşma süresi ortalama 13.69 dakika olarak tespit edilmiştir. Aynı şekilde C, D, E ve G başlangıç noktaları için sırasıyla 13.97 dakika, 8.89 dakika, 11.03 dakika ve 15.67 dakika ortalama seyahat süresi benzetim sonucu elde edilmiştir. Bu benzetim ile elde edilen sonuçlar aşağıda verilecek deneysel çalışmalar için baz alınmıştır.

Tablo 6.3. Tablo 6.1’deki sinyal süreleri ve Tablo 6.2’deki araç sayısı senaryosuna göre işletilen trafik ağ benzetim sonucu elde edilen başlangıç noktalarına göre araçların ortalama varış süreleri

Başlangıç Noktaları	A	C	D	E	G
Ortalama Varış Süresi (Dakika)	13.69	13.97	8.89	11.03	15.67

Çalışmanın bu aşamasında, Tablo 6.1’de verilen başlangıç trafik sinyal süreleri ile işletilen SUMO trafik modeli, farklı C ve FR parametreleri ile sezgisel yaklaşım ile etkileşimli olarak 100’er iterasyon işletilmiştir. Bu deneysel koşturmalar sonucu elde edilen sonuçlar Tablo 6.4’te verilmiştir. Eniyileme öncesi yani sezgisel yaklaşım işletmeksizin 63.25 dakika olan ortalama seyahat zamanının tüm deneysel koşturmalar sonucunda azaldığı gözlemlenmiştir. Tabloya göre en iyi ortalama seyahat süresi CR parametresinin 0.85, F değerinin ise 0.80 olduğu koşturma sonucu elde edilmiştir. Tabloya göre en kötü sonuç ise 56.87 dakika ile CR 1 ve F değerinin 0.80 olduğu durumda yaşanmıştır.

Tablo 6.4. DE algoritmasının 100 iterasyon koşturulması sonucu bulunan ortalama seyahat süresi tablosu

CR \ F	0.40	0.50	0.60	0.70	0.80	0.90	1.00
0.70	49.92	48.19	46.14	45.97	47.41	45.14	47.55
0.80	50.59	48.72	45.08	47.16	47.81	50.84	45.77
0.85	51.97	46.87	52.09	53.5	43.05	44.12	45.82
0.90	48.84	50.55	47.83	52.98	49.67	44.44	46.23
0.95	52.61	50.08	51.05	50.98	44.84	47.62	52.83
1.00	54.68	48.26	47.75	48.92	56.87	49.19	47.67

Çalışmanın sonraki aşamalarında kullanılmak üzere belirlenen CR = 0.85 ve F= 0.80 parametreleri ile koşturulan algoritmanın 100 iterasyon sonucunda probleme çözüm olarak sunduğu yeşil sinyal süreleri Tablo 6.5’te verildiği gibidir. Tabloya göre, örnek olarak 3 numaralı trafik denetleci için başlangıçta 50 saniye olarak belirlen faz-1 değeri 21.35 saniye olarak eniyilenmiş, faz-2’de 50 saniyelik başlangıç süresinin 8 saniye olarak eniyilendiği görülmüştür. Toplam döngü süresi ise 3 numaralı trafik denetleci için 29.35 saniye olarak elde edilmiştir.

Tablo 6.5. CR=0.85 F=0.80 ile bulunan yeşil sinyal süreleri

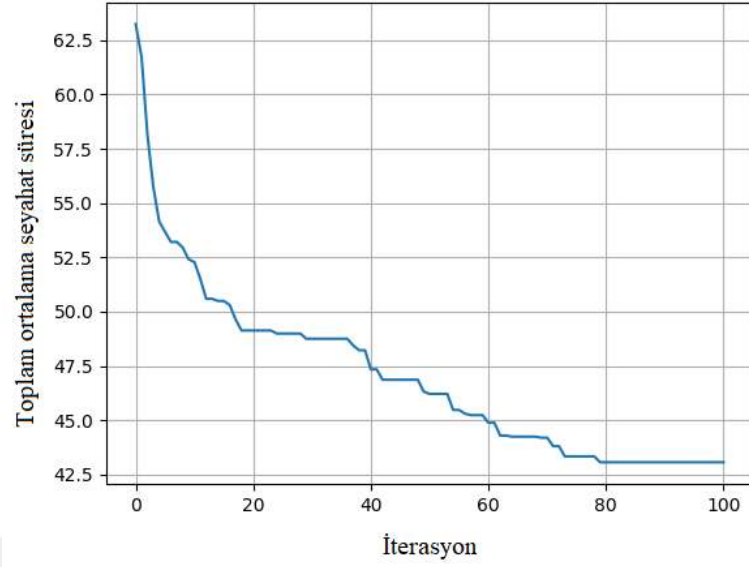
Toplam Işık Süresi C(s)	Trafik Işığı Numarası (i)	Faz-1 Yeşil-1	Faz-2 Yeşil-2	Faz-3 Yeşil-3
32.130	1	14.01	18.12	-
42.620	2	31.44	11.18	-
29.350	3	21.35	8.00	-
68.640	4	12.15	24.49	32.0
92.670	5	22.28	44.79	25.60
61.380	6	26.62	34.76	-

Tablo 6.6’da ise 100 iterasyonluk belirlenen en iyi CR ve F parametreleri ile koşturulan çözümün araçların ortalama seyahat süreleri başlangıç noktaları baz alınarak verilmiştir. Tabloya göre başlangıç noktası olarak belirlenen A konumundan başlayan araçların varış konumlarına ortalama seyahat süresi 9.78 dakika olarak tespit edilmiştir. C, D, E ve G konumları için ortalama seyahat süreleri sırasıyla 10.11 dakika, 5.439 dakika 7.430 dakika ve 10.299 dakika olarak elde edilmiştir. Bu değerlerin sezgisel eniyileme olmadan elde edilen (bkz. Tablo 6.3) değerlere göre çok daha iyi olduğu açıktır.

Tablo 6.6. CR=0.85 F=0.80 verileri ile bulunan ortalama araç seyahat süreleri

Başlangıç Noktaları	A	C	D	E	G
Ortalama Varış Süresi (Dakika)	9.780	10.110	5.439	7.430	10.299

Tablo 6.5’de belirtilen trafik ışıkları ve Tablo 6.6’da seyahat süresi sonuçları verilen sezgisel yaklaşımın koşturma boyunca değişen toplam ortalama seyahat süresine grafiği Şekil 6.1’de verilmiştir. Toplam ortalama seyahat süresinin 62.5 dakika ile başlayarak sezgisel yaklaşımın işletilmesi boyunca ilk 80 iterasyonda istikrarlı bir şekilde azaldığı grafikten açıkça görülmektedir. 80. iterasyondan sonra ise, çözüm doygunluğa ulaştığı için seyahat süresinde iyileştirme gerçekleşmemiştir. Sezgisel yaklaşım sonunda 63.25 dakikalık seyahat süresi 43.05 dakikaya gerilemiştir.



Şekil 6.1. 100 iterasyon koşturulan sezgisel yaklaşım sonucu “Ortalama Seyahat Süresi” değişim grafiği (CR=0.85, F=0.8)

100 iterasyon koşturulan sezgisel yaklaşımda, kullanılan az araç sayılı trafik senaryosunda %31.93'lük seyahat süresinden kazanç elde edilmiştir. Bu çalışmalar sonucu, sonraki aşamalarda kullanmak üzere en iyi sonucu veren CR ve F parametreleri sırasıyla 0.85 ve 0.80 olarak belirlenmiştir.

DE algoritmasının parametrelerinin az yoğun trafik ağında belirlenmesinden sonra daha yoğun araç sayılı bir senaryoda 1000 iterasyon ile eniyileme işlemleri gerçekleştirilmiştir. Tablo 6.7’de bu çalışmalarda kullanılan araç sayıları başlangıç-varış tablosunun 1000 iterasyonlu koşmada kullanılan değerlerine yer verilmiştir. Tabloya göre örnek olarak, D konumundan başlayan araçlardan 400 adet araç A konumuna, 250 araç B noktasına ve F noktasına 100 araç seyahat etmektedir. D noktası için toplam 750 araçlık bir senaryo düşünülmüştür. D ve E noktalarına bu konumdan araç seyahati olmamıştır.

Tablo 6.7. Denektaşı yol ağının eniyilemesi için belirlenen araç sayıları
(Başkan vd., 2020, s. 572)

O-D	A	B	D	E	F	Toplam
A	0	250	700	30	200	1180
C	40	20	200	130	900	1290
D	400	250	0	0	100	750
E	300	130	0	0	20	450
G	550	450	170	60	20	1250

Tablo 6.1’de verilen trafik ışık süreleri ve Tablo 6.7’de verilen araç matrisine göre eniyileme yapılmaksızın işletilmesi sonucu elde edilen ortalama varış süreleri Tablo 6.8’de belirtilmiştir. Araçların benzetim işlemi boyunca ortalama varış süresi toplamı 198.85 dakika olarak bulunmuştur. Tabloya göre A noktasından başlayan bir aracın eniyilenmemiş süreli trafik denetleci ile 80.22 dakikada ortalama seyahat edeceği bilgisi elde edilmiştir. Aynı şekilde C, D, E ve G noktalarından çıkan bir aracın için ise sırasıyla ortalama seyahat süresi 44.45, 18.13, 30.34 ve 25.81 dakika olarak bulunmuştur. En yüksek seyahat süresi A noktası için elde edilirken en düşük seyahat süresi ise D noktası için elde edilmiştir. Bu çalışmada sezgisel yaklaşım sonrası bulunan sonuçlar Tablo 6.8’deki veriler ile karşılaştırılacaktır.

Tablo 6.8. Eniyileme öncesi yoğun araç sayılı benzetimde araçların ortalama seyahat süreleri

Başlama Noktaları	A	C	D	E	G
Ortalama Varış Süresi (Dakika)	80.22	44.35	18.13	30.34	25.81

6.2. Tek Birey ile Yöntemin Uygulanması

Bu bölümde sezgisel yaklaşım ile koşturulan benzetim senaryosunda 4.2.1 bölümünde belirtilen DE algoritması için tek birey kullanılarak trafik sinyal süresi eniyilemesi yapılmıştır. 1000 iterasyon koşturulan DE algoritmasının her adımında oluşan bireyin elemanları sinyal dengeleme işlemlerine tabi tutulmuştur yani önceden belirlenen kısıtlar içinde kalmaya zorlanmıştır.

CR= 0.85 ve F=0.80 parametreleri kullanılarak 1000 iterasyon devam ettirilen 14 boyutlu/elemanlı tek birey ile kořturulan algoritmanın sunduđu çözümler toplam seyahat süresi 135.31 dakika olarak elde edilmiştir. Eniyileme sonucu elde edilen en iyi trafik sinyal süreleri Tablo 6.9’de verilmiştir. Tabloya göre örnek olarak 2 numaralı trafik denetleci için ilk fazda 48.82 saniye yeşil sinyal süresi verilirken, ikinci fazda ise 23.06 saniye yeşil sinyal süresi verilerek araçların seyahat sürelerinin daha iyi olacağı belirlenmiştir.

Tablo 6.9. Tek birey ile kořturulan algoritma ile belirlenen en iyi yeşil sinyal süreleri

Toplam Işık Süresi C(s)	Trafik Sinyal Numarası (i)	Faz-1 Yeşil-1	Faz-2 Yeşil-2	Faz-3 Yeşil-3
67.510	1	33.51	34.00	-
71.880	2	48.82	23.06	-
67.670	3	35.71	31.96	-
114.810	4	40.00	34.81	40.00
116.970	5	42.54	45.63	28.80
67.560	6	46.07	21.49	-

Tablo 6.10’de ise araçların seyahate başladığı noktalara göre ortalama seyahat süreleri verilmiştir. Tablodaki bu sürelerin toplamı 135.31 dakikadır. Toplam seyahat süresi eniyileme öncesi süreli trafik sinyal durumuna göre %31.95’lik bir kazanç elde etmiştir.

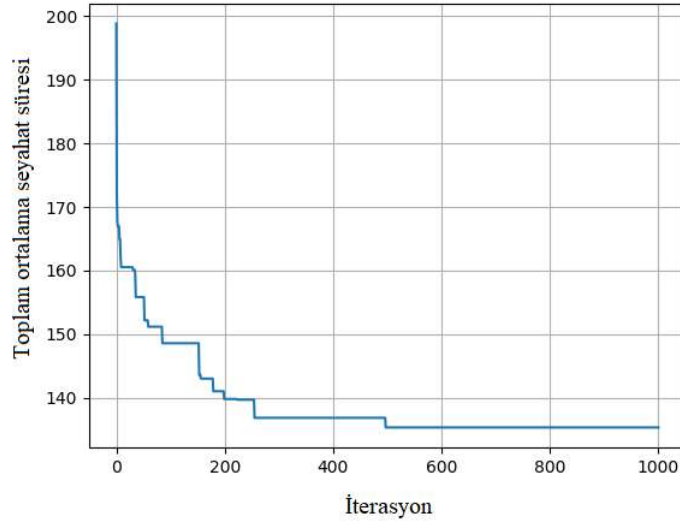
Tablo 6.10. Tek birey ile kořturulan algoritma ile elde edilen ortalama seyahat süreleri

Başlangıç Noktaları	A	C	D	E	G
Ortalama Varış Süresi (Dakika)	48.84	30.52	14.64	17.34	23.97

A başlangıç konumundan seyahate başlayan araçlar sezgisel yaklaşım ile eniyileme öncesinde elde edilen verilere göre %39.11, C konumundan benzetime başlayan araçlar %31.18, D konumu için %19.24, E konumu için %42,84 ve G konumundan başlayan araçlar için ise %7.12’lik ortalama seyahat süresinden kazanç elde edilmiştir.

Şekil 6.2’de 14 boyutlu/elemanlı tek birey ile koşturulan sezgisel yaklaşımın eğitim seyri verilmiştir. 1000 iterasyon koşturulan algoritma eğitim ölçüt değeri olarak 198.85 dakika ile başlamış ve eniyileme sonrası bu ölçüt değeri 135.31 dakikaya düşmüştür.

Şekilden de görüleceği üzere, eğitim seyri 550. iterasyona kadar seyahat süresinde belirgin azalmayla oluşurken takip eden iterasyonlarda seyahat süresinde herhangi bir iyileşme olmamıştır.



Şekil 6.2. Tek birey ile eniyileme sürecinde ölçüt değişimi

6.3. Çok Bireyli Sürü Kullanılarak Yöntemin Uygulanması

Bu alt bölümde sezgisel yaklaşım ile koşturulan benzetim senaryosunda 4.2.2 bölümünde tanımlanan DE algoritması ile sürü kullanılarak trafik sinyal süresi eniyilemesi yapılmıştır. Tablo 6.7’de belirtilen araç sayıları kullanılarak 1000 iterasyon koşturulan DE algoritması her adımda sinyal dengeleme işlemlerine tabi tutulmuştur. Tablo 6.11’de sezgisel yaklaşım sonucu elde edilmiş en iyi yeşil sinyal süreleri verilmiştir.

Tablo 6.11. Çok boyutlu sürü ile belirlenen en iyi yeşil sinyal süreleri

Başlangıç Noktaları	A	C	D	E	G
Ortalama Varış Süresi (Dakika)	57.48	43.47	14.46	17.58	31.21

Tablo 6.12.’de ise sürü ile koşturulan algoritmanın 1000 iterasyon sonucu başlangıç konumlarına göre elde edilen ortalama araç seyahat sürelerine yer verilmiştir. Tabloya göre tüm

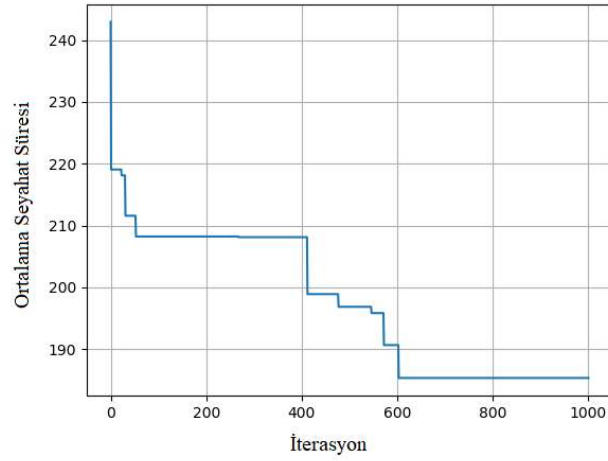
konumlar için ortalama seyahat süresi toplamı 164.20 dakika olarak elde edilmiştir ve sabit süreli trafik sinyallerine göre %17.425’lik seyahat süresinden kazanç elde edilmiştir.

Tablo 6.12. Sürü ile koşturulan algoritmanın sunduğu çözüm ile elde edilen ortalama seyahat süreleri

Toplam Işık Süresi C(s)	Trafik Işığı Numarası (i)	Yeşil-1	Yeşil-2	Yeşil-3
116.906	1	16.73	100.17	-
108.885	2	50.33	58.54	-
87.877	3	45.70	42.17	-
109.260	4	29.92	47.93	31.39
75.428	5	43.09	22.04	10.29
68.000	6	40.13	27.86	-

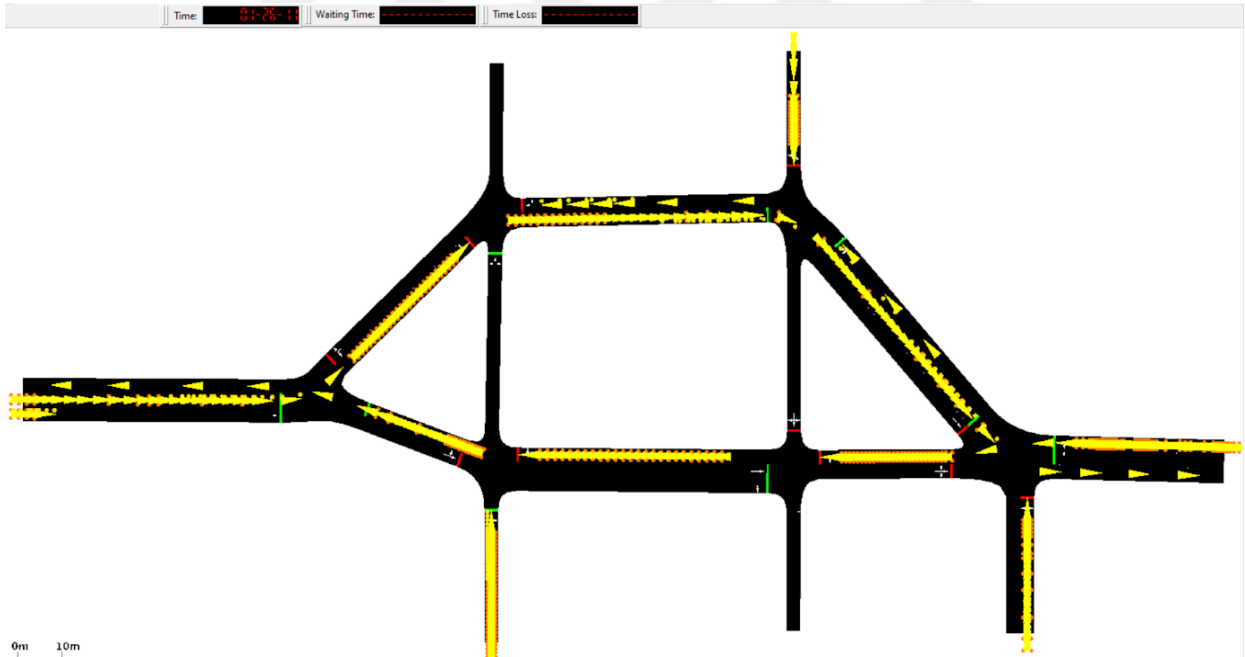
A başlangıç konumundan benzetime başlayan araçlar sezgisel yaklaşım öncesinde elde edilen verilere göre %28.34, C konumundan benzetime başlayan araçlar %1.98 , D konumu için %20.24, E konumu için %42,05 ortalama seyahat süresinden kazanç elde edilmiştir. G konumundan benzetime başlayan araçlar için ise %20.92’lik ortalama seyahat süresinden kayıp elde edilmiştir.

Şekil 6.3’te sürü kullanılan sezgisel yaklaşımın 1000 iterasyon boyunca sergilediği eğitim grafiği verilmiştir. Şekle göre 276.85 dakika ile başlayan toplam seyahat süresi ölçüt değeri 164.20 dakikaya kadar gerilemiştir. Şekle göre yüksek bir seyahat süresi ile başlayan algoritma 600’üncü iterasyona kadar sürekli olarak ölçüt değerini azaltarak olumlu şekilde eniyileme işlemini gerçekleştirmiştir.



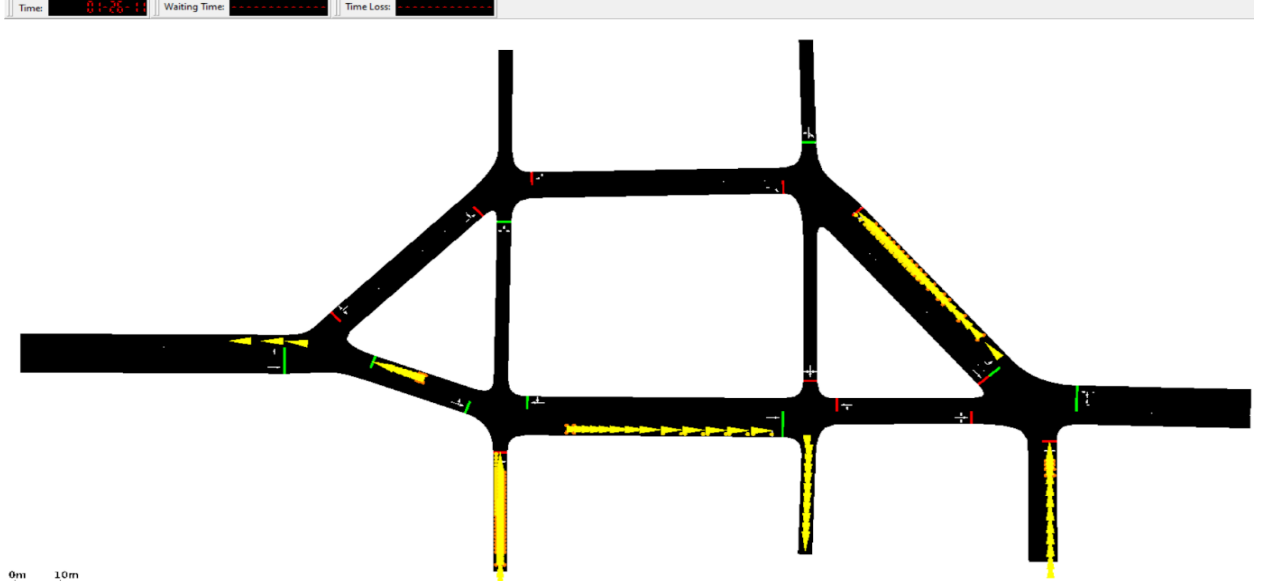
Şekil 6.3. Çok sayıda birey (sürü) ile eniyileme sürecinde ölçüt değişimi

Şekil 6.4'te görsel olarak karşılaştırma yapılabilmesi için önerilen uygulamalar öncesi yani eniyilenmemiş sinyal süreleri ile yapılan benzetimin sonuna yakın olan 1.26 anı baz alınarak trafik yol durumu görseli verilmiştir. Şekilden de görüleceği üzere tüm yollar için yoğunluk devam etmekte olup ana yollarda büyük araç kuyrukları oluştuğu tespit edilmiştir.



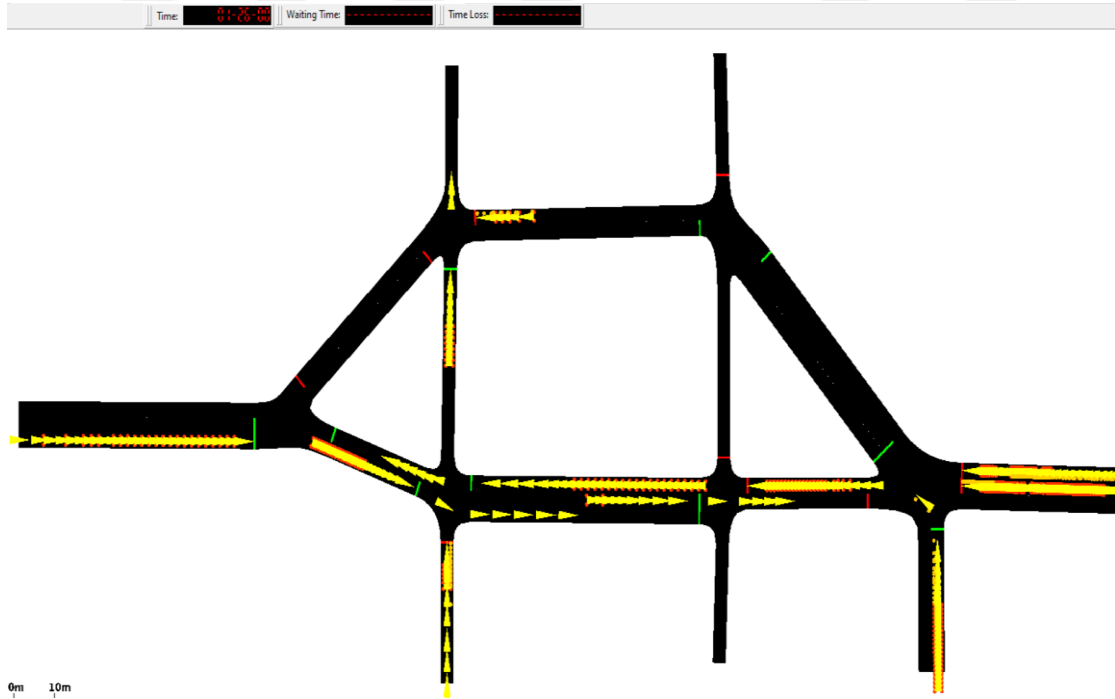
Şekil 6.4. Sinyal süreleri eniyilenmemiş durumda trafik yol durumu (1.26 anı)

Şekil 6.5'te ise, tek birey ile koşturulan sezgisel yaklaşım ile elde edilen sinyal süreleri ile yapılan benzetim sonuna yakın olan 1.26 anı baz alınarak trafik yol durumu görseli verilmiştir. Tüm yollar için araç sayıları yani yoğunluk azalmış olduğu Şekil 6.4 ile kıyaslama yapılarak net bir şekilde görülebilir.



Şekil 6.5. Tek birey kullanılarak eniyilenmiş sinyal süreleri ile trafik yol durumu (1.26 anı)

Şekil 6.6’da da çok birey yani sürü kullanılarak eniyilenmiş sinyal süreleri ile benzetim sonuna yakın olan yine 1.26 anındaki trafik yol durumu görseli verilmiştir. Bu görselden de durumun Şekil 6.4’teki eniyilenmemiş duruma göre trafik yoğunluğunun azaldığı ancak Şekil 6.5’te verilen tek birey ile eniyilenmiş trafik durumuna göre ise daha yoğun bir trafik durumunun olduğu gözlenmiştir.



Şekil 6.6. Çok boyutlu bireyler ile optimizasyon sonucu trafik yol durumu (1.26 anı)

7. SONUÇLAR VE TARTIŞMALAR

Bu çalışmada belirlenen sezgisel yaklaşım için iki farklı sürü oluşturma yöntemi kullanılarak denektaşı bir yol ağı için sinyal süresi dolayısıyla araç seyahat süresi eniyilemesi yapılmıştır. 14 elemanlı tek bireyden oluşan birinci yöntem ve 10 adet bireyden oluşan sürü yöntemi ile elde edilen sonuçların karşılaştırmaları yapılmıştır. İki yöntem için de sezgisel yaklaşım ile belirlenen sürelerde oluşturulan akıllı trafik denetleçlerin, geleneksel süreli yani eniyilenmemiş süreler ile oluşturulan trafik denetleçlerine göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Yeşil sinyal sürelerinin eniyilemesi için belirlenen sezgisel yaklaşım olan DE algoritması ile entegre edilen SUMO trafik yol benzetim programı süreç boyunca birlikte çalıştırılmıştır. Koşturulan tüm iterasyonlarda sistemler birbirleri ile haberleşerek veri alış-verişinde bulunmuştur. DE algoritması Python kolama dili ile yazılmış olup, SUMO programı için XML işaretleme dili kullanılmıştır. Benzetimi yapılan denektaşı yol ağı için sabit süreli yol ağı, tek birey ile çözüm geliştirilerek oluşturulan sezgisel yöntem ve çok bireyli/sürü ile oluşturulan sezgisel yöntem sonuçları itibarıyla hem sayısal metrikler hem de aynı andaki görsel trafik yoğunluğu ile karşılaştırılmıştır.

Tablo 6.8’de verilen eniyilenmemiş geleneksel süreli trafik denetleç sonuçları ve Tablo 6.10’de verilen sezgisel yaklaşımın tek birey ile koşturulması sonucu oluşan sinyal süreleri ile oluşturulan trafik denetleç sonuçları kıyaslandığında, tek bireyli sezgisel yöntemin ortalama seyahat süresi 135.31 dakika olduğu bu değer eniyilenmemiş yani geleneksel sabit süreli trafik sinyalleri durumuna göre 63.54 dakika daha kısa olduğu sonucuna ulaşılmıştır. Trafik seyahat süresinden kazanç ortalama yüzdeler bazda %31.953 olarak belirlenmiştir. Tablolara göre A, C, D, E ve G noktaları için seyahat sürelerinin oldukça azaldığı ve çalışmanın başarı ile sonuçlandığı söylenebilir. A noktasından benzetime giren araçlar seyahat süresi 80.22 dakika iken sezgisel yöntem sonrası bu değer 48.84 dakika olarak bulunmuştur. A noktası için ortalama yüzdeler bazda %39.11’lik başarı sağlanmıştır. C noktası için 44.35 dakikalık seyahat süresi 30.52 dakikaya düşerek %31.18, D noktası için 18.13 dakikalık süre 14.64 dakikaya düşerek %19.24, E noktası için 30.34 dakikalık süre 17.34 dakikaya düşerek %42.84 ve son olarak G noktası için 25.81 dakikalık değer 23.97 dakikaya düşerek %7.12’lik seyahat süresinden kazanç elde edilmiştir.

18 boyutlu 10 adet bireyden oluşan sürü ile işletilen ikinci yönteme göre ise toplam ortalama seyahat süresi 164.20 dakika olarak elde edilmiştir. Bu değer ise sabit süreli geleneksel

trafik sinyal süreli senaryoya göre 57.48 dakika daha kısadır. Tablo 6.12 ve Tablo 6.8 karşılaştırıldığında ise A noktası için ortalama seyahat süresi 80.22 dakikalık sabit süreli senaryo verisi 57.48 dakikaya düşerek %28,34 başarı sağlanmıştır. C noktası için 44.35 dakikalık seyahat süresi 43.47 dakikaya düşerek %1.98, D noktası için 18.13 dakikalık süre 14.46 dakikaya düşerek %20.24, E noktası için 30.34 dakikalık süre 17.58 dakikaya düşerek %42.05'lik başarı elde edilmiştir. G noktası için ise 25.81 dakikalık değer 31.21 dakikaya çıkarak %20.92 seyahat süresinden kayıp elde edilmiştir.

İki yöntem ile her ne kadar trafik sinyal süreleri dinamik olarak uyarlanmasa bile geleneksel sabit süreli trafik sinyallerine göre olumlu bir şekilde seyahat sürelerini etkilediği gözlemlenmiştir. Geleneksel eniyilenmemiş sabit süreli denetleç ile belirlenen yol ağı için bulunan 198.85 dakikalık toplam ortalama seyahat süresi her iki yöntem ile düşürülerek seyahat süresinden kazanç elde edilmiştir. Çalışmada elde edilen verilere göre eniyilenmiş trafik sinyal süreli denetleçlerin geleneksel sabit trafik sinyal süreli denetleçlere göre daha iyi sonuçlar verdiği sonucuna ulaşılmıştır.

Bu çalışmada kullanılan iki farklı sezgisel eniyileme yöntemleri arasında karşılaştırma yapıldığında ise tek bireyli yaklaşımın, çok bireyli yani sürü yaklaşımından daha iyi sonuç verdiği gözlemlenmiştir. Tek birey ile eniyileme sonucunda çoklu bireylere göre %17,59'luk daha iyi toplam ortalama seyahat süresi elde edilmiştir. Tek bireyli eniyilemede tüm başlangıç noktaları için olumlu sonuç elde edilirken, çok bireyli/sürü sürü eniyilemesinde ise beş başlangıç noktasından dört tanesinde olumlu sonuç elde edilmiştir.

Şekil 6.4'de eniyileme öncesi ve Şekil 6.5 ve Şekil 6.6'da eniyileme sonrası görsel olarak karşılaştırma yapılacak olur ise 1.26 anı için eniyileme öncesi tüm yollar için yoğunluk devam ederken tek bireyli eniyileme yöntemi ile işletilen algoritma ile aynı an için yollarda sadece şerit sayısı az olan ağlar için nispeten daha az bir yoğunluk olduğu görülmektedir. Çok birey/sürü ile işletilen algoritma için ise eniyileme öncesi duruma göre daha iyi yol durumu olduğu tespit edilmiştir.

Bu çalışmada araç rota bilgileri sistem üzerinde önceden tanımlanmış olarak sezgisel yaklaşımda kullanılmıştır. Gelecek çalışmalarda araç rotaları rastgele şekilde oluşturularak sezgisel yaklaşım için test edilebilir. Yollara belirli önem katsayıları verilerek bu araçların yollara stokastik şekilde atanması ve sezgisel yaklaşımın bu atamalar ile koşturulması sağlanabilir. Trafik sinyal süreleri belirlenen farklı stokastik atamalara göre eniyilemesi yapılarak seyahat süresi eniyilemesi yapılabilir. Birey sayıları, yol uzunluk değerleri ve araç

sayıları arttırılarak işlem gücü yüksek bilgisayarlarda gerçek dünyaya en yakın sonuçların elde edilmesi sağlanabilir. Sezgisel yaklaşımda daha hassas işlemler yapılması ve en iyi parametrelerin bulunması için CR ve FR parametreleri daha çok test verisinden geçirilerek iyileştirilebilir. Trafik denetleçlerinde sinyal süresi eniyilemesi ile birlikte yol doluluk oranına göre araç rotaları yeniden oluşturularak akıllı sistemler yardımı ile araç rotaları güncellenebilir ve çift taraflı veri aktarımı ile seyahat süresi eniyilemesi yapılabilir.



KAYNAKÇA

- Abushehab, R. K., Abdalhaq, B. K., & Sartawi, B.** (2014, March). Genetic vs. particle swarm optimization techniques for traffic light signals timing. In 2014 6th International Conference on Computer Science and Information Technology (CSIT) (pp. 27-35). IEEE.
- Akgüngör, A. P., YILMAZ, Ö., Korkmaz, E., & Doğan, E.** (2019). Meta-Sezgisel Yöntemlerle Sabit Zamanlı Sinyalize Kavşaklar için Optimum Devre Süresi Modeli. *El-Cezeri Journal of Science and Engineering*, 6(2), 259-269.
- Allsop, R. E., & Charlesworth, J. A.** (1977). Traffic in a signal-controlled road network: An example of different signal timings including different routeing. *Traffic Engineering & Control*, 18(Analytic).
- Araghi, S., Khosravi, A., Johnstone, M., & Creighton, D.** (2013, October). Intelligent traffic light control of isolated intersections using machine learning methods. In 2013 IEEE International Conference on Systems, Man, and Cybernetics (pp. 3621-3626). IEEE.
- Başkan, Ö. , Ceylan, H. & Ozan, C.** (2020). Investigating Acceptable Level of Travel Demand Before Capacity Enhancement for Signalized Urban Road Networks . *Teknik Dergi* , 31 (2) , 9897-9917 . DOI: 10.18400/tekderg.464260
- Chao, K. H., Lee, R. H., & Wang, M. H.** (2008, September). An intelligent traffic light control based on extension neural network. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 17-24). Springer, Berlin, Heidelberg.
- Coşkun, M., Baggag, A., & Chawla, S.** (2018, November). Deep reinforcement learning for traffic light optimization. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 564-571). IEEE.
- Çelik, Y., & Soy, H.** Trafik Tıkanıklığının Azaltılması İçin Öz-Örgütlenme Tabanlı Akıllı Trafik Optimizasyonu Yaklaşımı.
- Garcia-Nieto, J., Olivera, A. C., & Alba, E.** (2013). Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 17(6), 823-839.
- Gershenson, C., & Rosenblueth, D. A.** (2012). Adaptive self-organization vs static optimization: A qualitative comparison in traffic light coordination. *Kybernetes*.

Groeger, J. A., & Rothengatter, J. A. (1998). Traffic psychology and behaviour. *Transportation research part F: traffic psychology and behaviour*, 1(1), 1-9.

Gupta, V., Kumar, R., Reddy, K. S., & Panigrahi, B. K. (2017, July). Intelligent traffic light control for congestion management for smart city development. In *2017 IEEE region 10 symposium (TENSYP)* (pp. 1-5). IEEE.

Hirankitti, V., & Krohkaew, J. (2007, March). An agent approach for intelligent traffic-light control. In *First Asia International Conference on Modelling & Simulation (AMS'07)* (pp. 496-501). IEEE.

Hu, C., & Wang, Y. (2010, November). A novel intelligent traffic light control scheme. In *2010 Ninth International Conference on Grid and Cloud Computing* (pp. 372-376). IEEE.

Karakuzu, C. (2017). On the performance of newsworthy meta-heuristic algorithms based on point of view fuzzy modelling. *Turkish Journal of Electrical Engineering & Computer Sciences*, 25(6), 4706-4721.

Karakuzu, C. (2017). On the performance of newsworthy meta-heuristic algorithms based on point of view fuzzy modelling. *Turkish Journal of Electrical Engineering & Computer Sciences*, 25(6), 4706-4721.

Karakuzu, C., & Demirci, O. (2010). Fuzzy logic based smart traffic light simulator design and hardware implementation. *Applied Soft Computing*, 10(1), 66-73.

Karakuzu, C., & Topal, E. (2019, September). Fuzzy Logic Based Traffic Control System in Bilecik Province. In *2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 255-260). IEEE.

Khekare, G. S., & Sakhare, A. V. (2013, March). A smart city framework for intelligent traffic system using VANET. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)* (pp. 302-305). IEEE.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of SUMO-Simulation of Urban MObility. *International journal on advances in systems and measurements*, 5(3&4).

Liu, Y., Liu, L., & Chen, W. P. (2017, October). Intelligent traffic light control using distributed multi-agent Q learning. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)* (pp. 1-8). IEEE.

Murat, Y. Ş., & Çakıcı, Z. (2015). Sezgisel optimizasyon algoritmalarının taşıt gecikmesi problemi üzerine uygulaması, 7. Altyapı Sempozyumu, Trabzon, 615-625.

Sanchez, J. J., Galán, M., & Rubio, E. (2004, June). Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) (Vol. 2, pp. 1668-1674). IEEE.

Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.

Oztemiz, F., & Yeroğlu, C. (2018, September). Malatya public transportation route optimization via ant colony algorithm. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-6). IEEE.

Özgür, K. A. R. T., Genç, O. Ç., & Basciftci, F. Speed Compatible Green Wave Corridor with The Internet of Things. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 411-416.

Tajalli, M., Mehrabipour, M., & Hajbabaie, A. (2020). Network-level coordinated speed optimization and traffic light control for connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*.

Wei, H., Zheng, G., Yao, H., & Li, Z. (2018, July). Intellilight: A reinforcement learning approach for intelligent traffic light control. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2496-2505).

Wiering, M., Vreeken, J., Van Veenen, J., & Koopman, A. (2004, June). Simulation and optimization of traffic in a city. In IEEE Intelligent Vehicles Symposium, 2004 (pp. 453-458). IEEE.