

Ersin KORKMAZ

Doktora Tezi

KÜ 2019

KIRIKKALE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

İNŞAAT ANABİLİM DALI

İZOLE SİNYALİZE KAVŞAKLARDA YAPAY ZEKÂ TEKNİKLERİ İLE TRAFİK  
SİNYAL KONTROLÜ VE OPTİMİZASYONU

Ersin KORKMAZ

EYLÜL 2019

**İnşaat Anabilim Dalında** Ersin KORKMAZ tarafından hazırlanan İZOLE SİNYALİZE KAVŞAKLARDA YAPAY ZEKÂ TEKNİKLERİ İLE TRAFİK SİNYAL KONTROLÜ VE OPTİMİZASYONU adlı Doktora Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Doç. Dr. Orhan DOĞAN  
Anabilim Dalı Başkanı

Bu tezi okuduğumu ve tezin **Doktora Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.

Prof. Dr. Ali Payıdar AKGÜNGÖR  
Danışman

Jüri Üyeleri

Başkan : Prof. Dr. Serhan TANYEL

Üye (Danışman) : Prof. Dr. Ali Payıdar AKGÜNGÖR

Üye : Prof. Dr. Yetiş Şazi MURAT

Üye : Doç. Dr. Özgür BAŞKAN

Üye : Dr. Öğr. Üy. Erdem DOĞAN

..... /..... /.....

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onaylamıştır.

Prof. Dr. Recep ÇALIN  
Fen Bilimleri Enstitüsü Müdürü

## ÖZET

### İZOLE SİNYALİZE KAVŞAKLARDA YAPAY ZEKÂ TEKNİKLERİ İLE TRAFİK SİNYAL KONTROLÜ VE OPTİMİZASYONU

KORKMAZ, Ersin

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

İnşaat Anabilim Dalı, Doktora Tezi

Danışman: Prof. Dr. Ali Payıdar AKGÜNGÖR

Eylül 2019, 117 sayfa

Bu çalışmada, izole sinyalize kavşaklar için faz planı ve sinyal süresinin optimizasyonuna dayalı Hibrid Trafik Kontrol Sistemi (HTKS) geliştirilmiştir. Geliştirilen sistemde Çiçek Tozlaşma Algoritması (ÇTA), Karga Arama Algoritması (KAA) ve Tip-2 Bulanık Mantık (Tip-2 BM) yaklaşımları kullanılmış olup, gecikmenin minimize edilmesi amaçlanmıştır. İki ana modülden oluşan sistemde, birinci modül ile ÇTA kullanılarak faz plan optimizasyonu yapılmakta, ikinci modülde ise KAA ile optimize edilen Tip-2 BM yaklaşımı ile süre optimizasyonu gerçekleştirilmektedir. Faz optimizasyonuna ilaveten faz seçimi de sağlanmış ve belirlenen faz düzeninde hangi faza öncelik verileceği tayin edilmiştir. Geliştirilen sistemlerin simülasyonları KU-Trsim mikroskobik simülasyon programında gerçekleştirilmiştir.

KU-Trsim simülasyon programı kuyruk uzunluklarını ve araç sayılarını verecek şekilde revize edilmiş, ayrıca farklı trafik kontrol sistemleri ile uyumlu hale getirilmiştir. Simülasyon programının kullanım kolaylığı ve çıktı parametrelerinin görselliği için girdi ve çıktı ara yüzleri kontrol sistemlerine uygun olarak oluşturulmuştur.

HTKS 4 farklı kavşak geometrisi ve 15 farklı trafik hacim senaryosuna göre farklı kontrol sistemleri ile karşılaştırılarak sistemin performansı ve uygulanabilirliği ortaya konulmuştur. Sabit zamanlı sistem, Diferansiyel Gelişim Algoritması (DGA) ile optimize edilen sabit zamanlı optimum devre süresi (DGA-ODS) sistemi, sadece faz

optimizasyon modülünün kullanıldığı ÇTA-TKS sistemi ve Tip-1 BM-TKS sistemi performans karşılaştırılmasında kullanılan farklı kontrol sistemleridir. Kontrol sistemlerinden elde edilen gecikme değerlerine göre, HTKS'nın sabit zamanlı sisteme göre % 17 ile % 33 arasında, DGA-ODS sistemine göre % 9 ile % 19, ÇTA-TKS sistemine göre % 1,5 ile % 5 ve Tip-1 BM-TKS sistemine göre % 3 ile % 6 arasında iyileştirme sağladığı görülmüştür.

### **Anahtar kelimeler**

Trafik Sinyal Kontrolü, Çiçek Tozlaşma Algoritması, Diferansiyel Gelişim Algoritması, Karga Arama Algoritması, Tip-2 Bulanık Mantık



## **ABSTRACT**

### **OPTIMIZATION AND CONTROL OF TRAFFIC SIGNAL WITH ARTIFICIAL INTELLIGENCE TECHNIQUES AT ISOLATED SIGNALIZED INTERSECTIONS**

**KORKMAZ, Ersin**

**Kırıkkale University**

**Graduate School of Natural and Applied Sciences**

**Department of Civil Engineering, Ph. D. Thesis**

**Supervisor: Prof. Dr. Ali Payıdar AKGÜNGÖR**

**September 2019, 117 pages**

In this study, a Hybrid Traffic Control System (HTCS) was developed for isolated signalized intersections based on optimization of phase plan and signal timing. In the developed system, Flower Pollination Algorithm (FPA), Crow Search Algorithm (CSA) and Type-2 Fuzzy Logic (Type-2 FL) approaches were used and it was aimed to minimize the delay. In the system consisting of two main modules, phase plan optimization is performed using FPA in the first module, while in the second module, signal timing optimization is performed with CSA optimized the Type-2 FL approach. In addition to phase optimization, phase selection has also been achieved and it has been determined which phase will be given priority in the determined phase sequence. Simulations of the developed systems were performed in KU-Trsim microscopic simulation program.

In the KU-Trsim simulation program, codes were revised to give queue lengths and the number of vehicles and were also made compatible with different traffic control systems. The input and output interfaces are built in accordance with the control systems in terms of the ease of use of the simulation program and the simple visibility of the output parameters decisively.

The performance and applicability of the HTCS was demonstrated by comparing it with different control systems based on 4 different intersection geometry and 15 different

traffic volume scenarios. The fixed time, the fixed time optimized via Differential Evolution (DE) algorithm (DE-OCL), the FPA-TCS where only phase optimization module is used and Type-1 FL-TCS were different control systems. According to the delay values obtained from control systems, the HTCS was able to improve between 17% and 33 % compared to the fixed-time system, between 9 % and 19 % compared to the DE-OCL system, between 1.5 % and 5 % compared to the FPA-TCS system and between 3 % and 6 % compared to the Type-1 FL-TCS system.

**Keywords**

Traffic Signal Control, Flower Pollination Algorithm, Differential Evolution Algorithm, Crow Search Algorithm, Type-2 Fuzzy Logic

## TEŞEKKÜR

Doktora tezimi hazırlarken yaptığım araştırmalar ve trafik kontrol sistemlerinin oluşturulması sırasında yardımlarını esirgemeyen, bilgi, tecrübe ve görüşlerinden yararlandığım değerli danışman hocam, Sayın Prof. Dr. Ali Payıdar AKGÜNGÖR' e, tez çalışmalarım esnasında, bilimsel konularda daima yardımını gördüğüm hocam Sayın Doç. Dr. Özgür BAŞKAN'a ve Sayın Dr. Öğr. Üy. Erdem DOĞAN' a, tezimin birçok aşamasında yardım gördüğüm Dr. Volkan ATEŞ ve Arş. Gör. Nilay AYTAŞ' a teşekkür ederim.



## İÇİNDEKİLER DİZİNİ

|                                                |      |
|------------------------------------------------|------|
| ÖZET.....                                      | iii  |
| ABSTRACT.....                                  | V    |
| TEŞEKKÜR .....                                 | vii  |
| İÇİNDEKİLER DİZİNİ.....                        | viii |
| ŞEKİLLER DİZİNİ.....                           | xi   |
| ÇİZELGELER DİZİNİ .....                        | xiii |
| SİMGELER VE KISALTMALAR DİZİNİ.....            | xiv  |
| 1. GİRİŞ.....                                  | 1    |
| 1.1. Problemin Tanımı.....                     | 1    |
| 1.2. Amaç.....                                 | 2    |
| 2. LİTERATÜR.....                              | 4    |
| 3. KAVŞAKLAR VE SİNYALİZASYON SİSTEMLERİ ..... | 10   |
| 3.1. Kavşaklar.....                            | 10   |
| 3.1.1. Eşdüzey Kavşaklar .....                 | 11   |
| 3.1.1.1. Denetimsiz Eşdüzey Kavşaklar.....     | 11   |
| 3.1.1.2. Denetimli Eşdüzey Kavşaklar.....      | 12   |
| 3.1.2. Katlı Kavşaklar .....                   | 13   |
| 3.2. Sinyalizasyon.....                        | 13   |
| 3.2.1. İzole Sinyalize Sistemler .....         | 14   |
| 3.2.1.1. Sabit Zamanlı Sinyalizasyon.....      | 15   |
| 3.2.1.2. Trafik Uyarmalı Sinyalizasyon .....   | 15   |
| 3.2.1.3. Yaya Uyarmalı Sinyalizasyon.....      | 16   |
| 3.2.1.4. El ile Kumandalı Sinyalizasyon.....   | 17   |
| 3.2.2. Koordine Sinyalize Sistemler.....       | 17   |

|           |                                                                 |           |
|-----------|-----------------------------------------------------------------|-----------|
| 3.2.3.    | Sinyalizasyon Temel Kavramları.....                             | 17        |
| 3.2.4.    | Sinyalizasyon Faz Düzenlemesi .....                             | 19        |
| 3.2.5.    | Sinyalizasyon Hesapları .....                                   | 20        |
| 3.2.5.1.  | Webster Yöntemi ile Devre Süresi Hesabı .....                   | 20        |
| 3.2.5.2.  | HCM Yöntemi ile Devre Süresi Hesabı .....                       | 21        |
| <b>4.</b> | <b>YAPAY ZEKÂ TEKNİKLERİ VE BULANIK MANTIK .....</b>            | <b>22</b> |
| 4.1.      | Çiçek Tozlaşma Algoritması .....                                | 22        |
| 4.2.      | Karga Arama Algoritması.....                                    | 27        |
| 4.3.      | Diferansiyel Gelişim Algoritması.....                           | 31        |
| 4.4.      | Tip-2 Bulanık Mantık Yöntemi .....                              | 35        |
| <b>5.</b> | <b>TRAFİK SİMÜLASYON PROGRAMI .....</b>                         | <b>42</b> |
| 5.1.      | Revize edilmiş KU-Trsim Programı.....                           | 44        |
| 5.1.1.    | Araç Üretim Modülü .....                                        | 44        |
| 5.1.2.    | Araç Dinamikleri Modülü .....                                   | 45        |
| 5.1.3.    | Sinyal Denetim Modülü .....                                     | 47        |
| 5.2.      | KU-Trsim Programı Kullanıcı Ara yüzü.....                       | 48        |
| 5.2.1.    | Sabit Zamanlı Kontrol Sistemi Ara Yüzleri .....                 | 48        |
| 5.2.2.    | Adaptif Kontrol Sistemi Ara Yüzleri.....                        | 51        |
| 5.3.      | KU-Trsim Programının Performansı .....                          | 54        |
| <b>6.</b> | <b>TRAFİK KONTROL SİSTEMLERİ.....</b>                           | <b>58</b> |
| 6.1.      | DGA Optimum Devre Süresi Arama ile Trafik Kontrol Sistemi ..... | 58        |
| 6.2.      | Çiçek Tozlaşma Algoritması ile Trafik Kontrol Sistemi .....     | 61        |
| 6.3.      | Hibrid Trafik Kontrol Sistemi.....                              | 65        |
| 6.4.      | Tip-1 BM Trafik Kontrol Sistemi.....                            | 76        |
| <b>7.</b> | <b>SİMÜLASYON SONUÇLARI VE PERFORMANS KARŞILAŞTIRMA....</b>     | <b>81</b> |

|                                                    |            |
|----------------------------------------------------|------------|
| <b>8. SONUÇLAR.....</b>                            | <b>93</b>  |
| <b>KAYNAKLAR .....</b>                             | <b>95</b>  |
| <b>9. EKLER.....</b>                               | <b>103</b> |
| <b>EK. 1. ARAÇ DİNAMİK MODÜLÜ MATLAB KODU.....</b> | <b>103</b> |
| <b>ÖZGEÇMİŞ.....</b>                               | <b>117</b> |



## ŞEKİLLER DİZİNİ

|                                                                                                    |    |
|----------------------------------------------------------------------------------------------------|----|
| Şekil 3-1 Faz diyagramı [43].....                                                                  | 19 |
| Şekil 3-2 NEMA faz numaralandırması [44].....                                                      | 19 |
| Şekil 4-1 Tozlaştırıcılar ve tozlaşma çeşitleri [57] .....                                         | 24 |
| Şekil 4-2 Karga arama algoritması akış diyagramı.....                                              | 30 |
| Şekil 4-3 Diferansiyel gelişim algoritması akış diyagramı [64] .....                               | 34 |
| Şekil 4-4 Üyelik fonksiyon şekilleri .....                                                         | 35 |
| Şekil 4-5 Tip-1 ve Tip-2 üyelik fonksiyonları [66] .....                                           | 36 |
| Şekil 4-6 Tip-2 bulanık kümenin belirsizlik taban alanı, alt ve üst üyelik fonksiyonları[67] ..... | 37 |
| Şekil 4-7 Tip-1 Bulanık mantık sistem yapısı [66].....                                             | 38 |
| Şekil 4-8 Tip-2 Bulanık mantık sistem yapısı [66].....                                             | 38 |
| Şekil 4-9 Genel Tip-2 üyelik fonksiyonu [66].....                                                  | 40 |
| Şekil 4-10 Aralıklı Tip-2 üyelik fonksiyonu [66].....                                              | 41 |
| Şekil 5-1 Sabit zamanlı kontrol sistemi giriş ara yüzü .....                                       | 49 |
| Şekil 5-2 Sabit zamanlı kontrol sistemi çıkış ara yüzü.....                                        | 50 |
| Şekil 5-3 Adaptif kontrol sistemi giriş ara yüzü .....                                             | 52 |
| Şekil 5-4 Adaptif kontrol sistemi çıkış ara yüzü.....                                              | 53 |
| Şekil 5-5 3 Kollu kavşak için simülasyon karşılaştırmaları .....                                   | 56 |
| Şekil 5-6 4 Kollu kavşak için simülasyon karşılaştırmaları .....                                   | 56 |
| Şekil 6-1 DGA-ODS akış diyagramı .....                                                             | 60 |
| Şekil 6-2 Faz optimizasyon modülü akış diyagramı .....                                             | 63 |
| Şekil 6-3 ÇTA-TKS sistemi genel yapısı.....                                                        | 64 |
| Şekil 6-4 HTKS sistemi genel yapısı .....                                                          | 66 |

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| <b>Şekil 6-5</b> Sinyal süresi optimizasyon modülü akış diyagramı .....                       | 67 |
| <b>Şekil 6-6</b> Araç sayısı üyelik fonksiyonları.....                                        | 68 |
| <b>Şekil 6-7</b> Kuyruk uzunluğu üyelik fonksiyonları .....                                   | 68 |
| <b>Şekil 6-8</b> Optimize edilmiş araç sayısı üyelik fonksiyonları.....                       | 69 |
| <b>Şekil 6-9</b> Optimize edilmiş kuyruk uzunluğu üyelik fonksiyonları .....                  | 69 |
| <b>Şekil 6-10</b> 3 kollu kavşak süre değişim aralıkları performans dağılımı .....            | 71 |
| <b>Şekil 6-11</b> 4 kollu kavşak süre değişim aralıkları performans dağılımı .....            | 71 |
| <b>Şekil 6-12</b> Dört farklı değişim süresi gecikme değerleri.....                           | 73 |
| <b>Şekil 6-13</b> YS-BMM üyelik fonksiyonları [25].....                                       | 77 |
| <b>Şekil 6-14</b> Yaklaşım kolu bölgeleri ve detektörler [25] .....                           | 78 |
| <b>Şekil 6-15</b> US-BMM üyelik fonksiyonları [25].....                                       | 79 |
| <b>Şekil 6-16</b> Bulanık mantık optimizasyonu [25].....                                      | 80 |
| <b>Şekil 7-1</b> 3 kollu kavşak geometrileri .....                                            | 82 |
| <b>Şekil 7-2</b> 4 kollu kavşak geometrileri .....                                            | 82 |
| <b>Şekil 7-3</b> 3 kollu kavşak için trafik senaryolarına göre gecikme dağılımı.....          | 86 |
| <b>Şekil 7-4</b> 3 kollu kavşak sol cep için trafik senaryolarına göre gecikme dağılımı ..... | 87 |
| <b>Şekil 7-5</b> 4 kollu kavşak için trafik senaryolarına göre gecikme dağılımı.....          | 88 |
| <b>Şekil 7-6</b> 4 kollu kavşak sol cep için trafik senaryolarına göre gecikme dağılımı ..... | 89 |

## ÇİZELGELER DİZİNİ

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Çizelge 5-1 Araç tipleri ve uzunlukları.....                                        | 45 |
| Çizelge 5-2 Hız aralıklarına bağlı ivmelenme değerleri [25].....                    | 46 |
| Çizelge 5-3 Simülasyon girdi parametreleri .....                                    | 55 |
| Çizelge 5-4 Simülasyon istatistiki göstergeleri .....                               | 55 |
| Çizelge 6-1 DGA-ODS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti..  | 59 |
| Çizelge 6-2 ÇTA-TKS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti... | 63 |
| Çizelge 6-3 HTKS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti ..... | 65 |
| Çizelge 6-4 Yeşil süre değişim aralıklarının performansı .....                      | 70 |
| Çizelge 6-5 Normallik testi sonuçları .....                                         | 72 |
| Çizelge 6-6 Levene ve ANOVA testi sonuçları .....                                   | 73 |
| Çizelge 6-7 Tip-2 BM çıkış üyelik fonksiyonu .....                                  | 74 |
| Çizelge 6-8 BM kural tabanı .....                                                   | 75 |
| Çizelge 6-9 YS-BMM kural tabanı [25] .....                                          | 77 |
| Çizelge 7-1 3 kollu kavşak için trafik senaryoları .....                            | 83 |
| Çizelge 7-2 4 kollu kavşak için trafik senaryoları .....                            | 83 |
| Çizelge 7-3 Diferansiyel gelişim algoritması kontrol parametreleri.....             | 84 |
| Çizelge 7-4 Çiçek tozlaşma algoritması kontrol parametreleri .....                  | 85 |
| Çizelge 7-5 Karga arama algoritması kontrol parametreleri .....                     | 85 |
| Çizelge 7-6 Farklı kavşak geometrilerine göre OYH.....                              | 90 |
| Çizelge 7-7 Doygunaltı ve doyunüstü trafik akımları göre OYH.....                   | 92 |

## SİMGELER VE KISALTMALAR DİZİNİ

### SİMGELER DİZİNİ

|           |                           |
|-----------|---------------------------|
| $d$       | Gecikme (sn)              |
| $C$       | Devre Süresi (sn)         |
| $g$       | Etkin Yeşil Süre (sn)     |
| $c$       | Kapasite (taşıt/saat)     |
| $v$       | Hacim (taşıt/saat)        |
| $T$       | Analiz Periyodu (sn)      |
| $x$       | Doygunluk Derecesi        |
| $q$       | Trafik Akımı (taşıt/saat) |
| $\lambda$ | Yeşiller Oranı            |
| $y$       | Akımlar Oranı             |
| $L$       | Kayıp Zaman (sn)          |
| $r$       | Kırmızı Süre (sn)         |
| $a$       | Sarı Süre (sn)            |
| $C_o$     | Optimum Devre Süresi (sn) |
| $u$       | Hız (km/saat)             |
| $k$       | Yoğunluk (taşıt/km)       |
| $s_v$     | Mesafe Ortalamalı Hız     |

## KISALTMALAR DİZİNİ

|       |                                                                                                        |
|-------|--------------------------------------------------------------------------------------------------------|
| AA    | Ateşböceği Algoritması (Firefly Algorithm)                                                             |
| AAA   | Armoni Arama Algoritması (Harmoni Search Algorithm)                                                    |
| ANBÇS | Adaptif Nöron Bulanık Çıkarım Sistemi (Adaptive Neuro Fuzzy Inference System)                          |
| BM    | Bulanık Mantık (Fuzzy Logic)                                                                           |
| BT    | Benzetimli Tavlama (Simulated Annealing) Algoritması                                                   |
| ÇOA   | Çekirge Optimizasyon Algoritması (Grasshopper Optimization Algorithm)                                  |
| ÇTA   | Çiçek Tozlaşma Algoritması (Flower Pollination Algorithm)                                              |
| DGA   | Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm)                                    |
| DGBYA | Diferansiyel Gelişim Bakteri Yiyecek Arama Algoritması (Differential Evolution and Bacterial Foraging) |
| EP    | Evrimsel Programlama (Evolutionary Programming)                                                        |
| GA    | Genetik Algoritma (Genetic Algorithm)                                                                  |
| GKA   | Guguk Kuşu Algoritması (Cuckoo Search Algorithm)                                                       |
| GKO   | Gri Kurt Algoritması (Grey Wolf Algorithm)                                                             |
| HCM   | Otoyol Kapasite Kullanımı (Highway Capacity Manual)                                                    |
| İKAA  | İyileştirilmiş Karga Arama Algoritması (Modified Crow Search Algorithm)                                |
| KAA   | Karga Arama Algoritması (Crow Search Algorithm)                                                        |
| KHA   | Kaotik Harmoni Algoritması (Chaotic Harmoni Algorithm)                                                 |
| KKOA  | Karınca Kolonisi Optimizasyon Algoritması (Ant Colony Optimization Algorithm)                          |
| KSA   | Klonal Seçim Algoritması (Clonal Selection Algorithm)                                                  |
| MA    | Memetik Algoritması (Memetic Algorithm)                                                                |
| MÇTA  | İyileştirilmiş Çiçek Tozlaşma Algoritması (Modified Flower Pollination Algorithm)                      |
| PSO   | Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) Algoritması                                  |

|      |                                                                  |
|------|------------------------------------------------------------------|
| TT   | Tepe Tırmanış Algoritması (Hill Climbing Algorithm)              |
| YA   | Yarasa Algoritması (Bat Algorithm)                               |
| YAKA | Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm) |
| YSA  | Yapay Sinir Ağları (Artificial Neural Network)                   |



## 1. GİRİŞ

Sinyalizasyon sistemleri, kullanılmaya başlanıldığından beri farklı yaklaşımların uygulanmasıyla birlikte sürekli gelişim içerisinde olmuştur. Böylece şehirlerin büyümesi ve trafik yoğunluğunun artması, mevcut kavşak kapasitelerinin yetersiz kalması gibi nedenlerden dolayı oluşan düşük kavşak performansı problemine daha etkin çözümler üretilebilinmiştir. Bu çalışma kapsamında da kavşak performansını artırmaya yönelik yeni bir trafik sinyal kontrol sistemi geliştirilmiş olup, mevcut sistemlerin eksiklikleri giderilmeye çalışılmıştır. Bu amaçla faz planının belirlenmesi, faz sırasının değiştirilmesi ve sinyal sürelerinin elde edilmesi yaklaşımları birlikte kullanılmış ve etkili güncel algoritmalarından yararlanılmıştır.

Doktora tezi çalışmasında ilgili literatüre 2. bölümde değinilmiştir. Kavşaklar, sinyalizasyon sistemleri 3. bölümde açıklanmış, 4. bölümde çiçek tozlaşma (ÇTA), karga arama (KAA) ve diferansiyel gelişim (DGA) algoritmaları ile Tip-2 bulanık mantık (Tip-2 BM) yaklaşımından bahsedilmiştir. 5. bölümde kullanılan Kırıkkale Üniversitesi Trafik Simülasyon (KU-Trsim) programının detayları verilmiştir. 6. bölümde geliştirilen trafik kontrol sistemine ait modüller ve çalışma yapısı verilmiş olup, 7. Bölümde performans karşılaştırmaları, bulgular ve değerlendirmeler sunulmuştur. Son bölümde ise elde edilen sonuçlar değerlendirilmiş ve bazı önerilerde bulunulmuştur.

### 1.1. Problemin Tanımı

Ülkemizde ve dünyada hem nüfus hem de motorlu taşıt sayısı artış eğilimindedir. Bu artış aynı zamanda ulaşım talebinin artmasına, şehir içi yol ağlarında yoğunluk oluşmasına ve dolayısıyla taşıt gecikmelerinin artmasına neden olmaktadır. Ayrıca kavşak geometrilerinin kolayca değiştirilememesi ve mevcut sistemlerin değişen taleplere yeterince adapte olamaması gibi etkenler de taşıt gecikmelerindeki bu artışlarda rol oynamaktadır. Özellikle yolların eş düzey olarak işletildiği kavşaklarda tıkanıklık ve gecikmeler daha belirgin olarak hissedilmektedir. Bu nedenle eş düzey kavşaklardaki trafik akım kontrolünün sağlanması ve güvenli bir şekilde yönetilmesi trafik mühendisliğinin önemli bir çalışma konusunu oluşturmaktadır. Bu hususta, taşıt

gecikmesi günlük hayatı etkileyen en önemli ulaşım problemlerinden birisi olarak değerlendirilmekte olup kavşak performansını yansıtan önemli bir parametredir. Tüm bu etkenler göz önünde bulundurulduğunda, kavşak performansı iyileştirilerek gecikmenin minimize edilmesi konusu, kavşak kontrolünde uygulanan birçok farklı yöntemin temel amacıdır.

Kavşakların kontrolünde kullanılan sinyal denetim sistemleri, ön zamanlı sistemler ve adaptif sistemler olmak üzere iki ana grupta sınıflandırılmaktadır. Özellikle ön zamanlı sinyal denetim sistemlerinin sabit sinyal süreleri ile çalışması, değişen trafik koşulları meydana geldiğinde sistemin etkisiz kalmasına ve sistemde yüksek gecikmelerin oluşmasına sebep olmaktadır. Adaptif denetim sistemlerinde ise değişen trafik hacimlerine göre sinyal sürelerinin belirlenmesi sistemin etkinliğini arttırmakta ve daha düşük gecikme değerleri ile kavşağın kontrol edilmesine olanak sağlamaktadır. Yapay zekâ tekniklerinin gelişmesiyle yüksek performans gösteren algoritmaların adaptif sistemlere dahil edilmesi kontrol sistemlerinin performanslarına önemli ölçüde katkı sağlamaktadır. Böylece optimum düzeyde kavşak kontrolü ile birim zamanda daha fazla taşıt geçişi sağlanmaktadır. Günümüzde yapılan çalışmalarla adaptif sistemlerin etkinliğinin ve performansının geliştirilmesine güncel kontrol yaklaşımları ile yeni metotlar uygulanılarak devam edilmektedir.

## **1.2. Amaç**

Bu tezde, değişen trafik koşullarına uyum sağlayabilecek özelliklere sahip dinamik trafik kontrol sisteminin kurulması amaçlanmaktadır. Ayrıca güncel ve etkili algoritmaların sistem içerisinde kullanımı sayesinde kontrol yaklaşım performansının artırılması da bu tezin diğer bir amacını oluşturmaktadır. Böylece trafikteki değişimlere hızlı tepki verebilen ve daha düşük gecikme değeriyle kavşak kontrolüne olanak sağlayan bir sistem geliştirilebilmektedir. Mevcut adaptif kontrol sistemlerinin çoğunda kullanılan süre optimizasyonuna dayalı yaklaşımlar, kavşak performansının artmasına etki göstermesine rağmen tek başına yeterli olmamaktadır. Aynı zamanda faz planının da kavşak performansında etkisi bulunmakta olup dinamik bir faz planının oluşturulması oldukça önemlidir. Bu hususta, tasarlanan dinamik trafik kontrol sisteminin mevcut sistemlere göre daha iyi bir performans ile çalışabilmesi için, faz

planı ve sinyal sürelerinin optimize edilmesinin yanı sıra faz sırasının değişken bir yapıda olmasına bağlı olarak geliştirilmesi amaçlanmıştır. Günümüze kadar yapılan çalışmalarda ÇTA ve KAA algoritmalarının trafik kontrol sistemlerinde tek başına ya da birlikte kullanılmamıştır. Bu tezin yenilikçi yönü ise, her iki algoritmanın da birlikte kullanılarak hem daha iyi performans sonuçlarının elde edilmesi hem de algoritmaların performanslarının ortaya konularak incelenmesidir.

Modüler bir yapıda sistemin geliştirilmesi amaçlanarak faz ve süre optimizasyonunun etkilerinin net bir şekilde gösterilmesi sağlanmaktadır. Bu tezde, faz optimizasyonu modülünde minimum gecikme değerini ortaya koyan faz planının belirlenmesi için ÇTA kullanılmıştır. Amaç fonksiyonu olarak, taşıt başına düşen gecikmenin minimize edilmesi hedeflenmiş olup, taşıt başına düşen gecikmenin hesabında HCM 2000 gecikme bağıntısı tercih edilmiştir. Aynı zamanda belirlenen faz düzenine ait optimum devre süresinin de tespiti bu modül içerisinde gerçekleştirilmektedir. Faz optimizasyonuna ek olarak faz seçimi de sağlanmış ve belirlenen faz düzeninde hangi faza öncelik verileceği tayin edilmiştir. Faz sırası, fazlara ait araç sayılarına göre belirlenmiş ve en yüksek taşıt sayısına sahip olan faza öncelik verilecek şekilde oluşturulmuştur. Süre optimizasyonu modülünde ise, kavşak geometrisi, faz planı ve optimum devre süresi belli olan bir izole kavşağın, Tip-2 bulanık mantık yaklaşımı ile sinyal sürelerinin artırılacağına ya da azaltılacağına karar verilmektedir. Farklı trafik durumlarına göre bulanık mantık yönteminin optimize edilmesi, daha düşük gecikme değerleri ile kavşak kontrolünün sağlanmasına olanak vermektedir. Bu nedenle KAA kullanılarak bulanık mantık yaklaşımının üyelik fonksiyonları optimize edilmiştir. Sadece faz optimizasyon modülünün kullanıldığı kontrol sistemi ve faz ile süre optimizasyon modüllerinin birlikte kullanıldığı kontrol sistemi oluşturularak bu modüllerde kullanılan yaklaşımların etkileri de incelenmiştir. Bütün modüllerin birlikte kullanımı ile değişen trafik durumlarına adapte olan dinamik trafik kontrol sistemi oluşturulmuştur. Geliştirilen sistemlerin yanı sıra DGA yaklaşımıyla optimum devre süresinin belirlenmesi ve aynı zamanda sabit zamanlı kontrol sisteminin performansının artırılması da amaçlanmıştır.

## 2. LİTERATÜR

Taşıt ve yaya akımlarında çakışmaların oluşmasıyla birlikte 20. yüzyılın başlarında sinyalize kavşaklar İngiltere’de kullanılmaya başlanmıştır. Bu durum birçok araştırmacının kavşak kontrolü üzerinde araştırmalar yapmasına neden olmuştur. Zaman içerisinde kavşak kontrolünün iyileştirilmesine yönelik farklı yaklaşımların kullanıldığı çalışmalar ortaya konulmuştur. Özellikle yapay zekâ yöntemlerine dayalı olarak yapılan çalışmaların, geleneksel kavşak kontrolünden daha iyi sonuçlar verdiği görülmüştür. Murat ve Gedizlioğlu [1] Bulanık Mantık (BM) yaklaşımı kullanarak kavşak denetim modeli geliştirmişlerdir. İki BM sisteminden oluşan modelde yeşil sürelerin belirlenmesi ve faz sırasının değiştirilmesi sağlanmaktadır. Geliştirilen denetim modelinin trafik uyarmalı denetleyici ile karşılaştırılması sonucunda yüksek trafik hacimlerinde ortalama gecikmeyi düşürdüğü görülmüştür. Trabia ve diğ. [2] yapmış oldukları çalışmada izole sinyalize kavşaklar için BM ile trafik sinyal kontrolünün tasarımını ve değerlendirmesini yapmışlardır. İki aşamalı olarak çalışan yaklaşımda, birinci aşamada trafik akımlarına bağlı olarak trafik yoğunluğu tahmin edilirken, ikinci aşamada trafik yoğunluğuna göre faz süresinin uzamasına veya aynı durumda kalmasına karar verilmiştir. Düşük, orta ve yüksek trafik için önerilen yöntemi ile trafik uyarmalı yöntemin sonuçları gecikme üzerinden karşılaştırılmış ve bulanık mantık yaklaşımının ortalama gecikmeyi % 9,5 azaltması ile kavşak kontrolünde etkin olabileceği görülmüştür.

Sarı ve diğ. [3] bulanık modelleme yaklaşımını çeşitli problemlere uygulamışlar ve bu problemlerden bir tanesini de trafik sinyal denetimi olarak seçmişlerdir. Yapılan çalışma da bulanık modellemenin klasik modellemeden daha iyi bir yaklaşım olduğunun ve daha yüksek bir performans gösterdiğinin altını çizmişlerdir. Murat ve Gedizlioğlu [4] izole sinyalize kavşaklar için BM çok fazlı sinyal kontrol modeli geliştirmişlerdir. Dört kollu izole kavşak üzerinden trafik uyarmalı ve SIDRA taşıt uyarmalı model sistemleri ile BM yaklaşımı karşılaştırmışlardır. BM yaklaşımı ile geliştirilen sistemin her iki sistemden de daha iyi performans sergilediğini ve özellikle yüksek trafik hacminde bu durumun daha belirgin olduğunu göstermişlerdir. Murat ve diğ. [5] BM trafik sinyal kontrol yaklaşımını iki şeritli rampa da kullanmışlar ve rampa ölçüm modeli geliştirmişlerdir. Giriş parametreleri olarak, ana yoldaki geliş aralıkları, rampadaki kuyruk uzunluğu ve

kırmızı sürenin kullanıldığı modelde kırmızı sinyal sürelerinin belirlenmesi sağlanmıştır. Microsoft Excel programı üzerinden geliştirdikleri simülasyon ile test ettikleri modelin gecikme, kuyruk uzunluğu, emisyon, yakıt tüketimi vb göstergeleri önemli ölçüde düşürebildiğini ortaya koymuşlardır.

Gacovski ve diğ [6] izole sinyalize kavşaklar için BM trafik kontrolünü uygulamışlardır. MATLAB simulink kullanarak gerçekleştirdikleri simülasyonu sabit zamanlı kontrol ile karşılaştırmışlardır. Kuyruk uzunluğu ve bekleme süresine göre çalışan kontrol sisteminde, minimum bekleme, gecikme ve kuyruk uzunluğuna bağlı olacak şekilde faz düzenini ve sinyal süresini optimize etmişlerdir. Sabit zamanlı sisteme göre daha etkili sonuçlar ortaya koyduğunu vurgulamışlardır. Mwangi ve diğ. [7] çalışmasında BM kontrol sistemine dayalı devre süresi ve faz düzeni optimizasyonu yapan bir kontrol sistemi geliştirmişlerdir. Normal trafik altında geliştirilen model sabit zamanlı kontrol sistemi ile yakın sonuçlar ortaya koyarken, yüksek trafik durumlarında önerilen model sabit zamanlı kontrolöre göre gecikme değerini %22 düşürebilmiştir. Yulianto [8] BM trafik sinyal kontrolünü gerçek zamanlı trafik taleplerine cevap verecek şekilde geliştirmiştir. VISSIM ile simülasyonu yapılan BM kontrol sistemi sabit zamanlı kontrol ile karşılaştırılarak gecikme, seyahat süresi ve hız bakımından önerilen yöntemin daha iyi olduğu gösterilmiştir.

Ceylan ve Bell [9] Genetik Algoritma (GA) yaklaşımını trafik sinyal süresi optimizasyonu için uygulamışlardır. TRANSYT yazılımı ile birlikte genetik algoritmayı kullanarak Allsop and Charlesworth'un test ağında süre optimizasyonu gerçekleştirmişlerdir. Böylece Performans İndeksi (PI), gecikme ve durma sayısına göre, stokastik kullanıcı denge akımlarının kullanıldığı yol ağ yapısında değerlendirilmiştir. Karşılaştırma sonuçları GA'nın diğer sezgisel yöntemlerden daha basit ve etkili olduğunu ortaya koymuşlardır. Shoufeng ve diğ. [10] gecikmeyi minimize etmek amacı ile Q-öğrenme (Q-learning) (QL) algoritmasının adaptif trafik sinyal kontrolü için performansını test etmişlerdir. İzole sinyalize kavşakta sabit zamanlı sinyal kontrol ile algoritmayı karşılaştırarak farklı trafik durumlarında daha düşük gecikme değerine ulaşabildiklerini göstermişlerdir. Zhao ve diğ. [11] doygünüstü kavşaklar için Açgözlü Arama Algoritmasını (Greedy Algorithm) devre süresi optimizasyonu için uygulamışlardır. Üç fazda çalıştırılan dört kollu bir kavşağa uygulanan algoritmanın,

diğer sezgisel algoritmalarla göre hesaplama karmaşıklığını azalttığını görmüşlerdir. Ayrıca algoritmanın etkinliğini de sayısal bir örnek ile göstermişlerdir.

Başkan ve Haldenbilen [12] Karınca Koloni Optimizasyon (KKO) yöntemini trafik sinyal süresini optimize etmek için uygulamışlardır. Allsop and Charlesworth'un test ağında önerilen yöntemi test etmişlerdir. Önerilen yaklaşımı Karşılıklı Uyumlu Çözüm (Mutually Consistent (MC) Solution) ile karşılaştırarak performansını ortaya koymuşlardır. Önerilen algoritmanın sinyal zamanlamaları ve doyumluk derecesi değerleri bakımından oldukça başarılı sonuçlar verdiğini ifade etmişlerdir. MC çözümünün benzer sonuçlar vermesine rağmen, daha fazla devre süresi belirlediği ve MC çözümünün ilk sinyal zamanlama grubuna bağlı olmasından dolayı ilk atamaya duyarlı olduğunu belirtmişlerdir. Putha ve diğ. [13] doyum üstü durumlardaki trafik sinyal optimizasyonu için KKO ve GA yaklaşımlarını karşılaştırmışlardır. Yirmi kavşaklı bir yol ağında iki farklı senaryoya göre denenilen yaklaşımlarda KKO'nun GA'dan daha iyi sonuçlar ortaya koyduğunu görmüşlerdir. He ve Hou [14] gecikme, durma sayısı ve kapasite değerlerinin performans indeksi olarak kullanıldığı çalışmalarında KKO ile sinyal sürelerini belirlemişlerdir. Webster modeli ve GA ile karşılaştırılan yaklaşımın Webster modeline üstünlük kurduğu ve GA'dan da daha iyi performans gösterdiğini ortaya koymuşlardır. Böylece daha düşük gecikme ve durma sayısı ile kavşak kontrolünü sağlayabilmişlerdir.

Liu ve Xu [15] izole kavşaklardaki trafik sinyal süresi optimizasyonunda Diferansiyel Gelişim Bakteri Yiyecek Arama (DGBYA) algoritmasını uygulamışlardır. Devre süresi içinde gecikmeyi minimize edecek şekilde amaç fonksiyonu oluşturulan yaklaşımda, düşük, orta ve yüksek talep içeren senaryolara göre sabit zamanlı ve GA'ya dayalı kontrol sistemleri ile karşılaştırma yapılmış ve önerilen yaklaşımın en düşük gecikme değerlerini verdiği gösterilmiştir. Dell'Orco ve diğ. [16,17] Armoni Arama Algoritmasını (AAA) ve Yapay Arı Koloni Algoritmasını (YAKA) denge ağı tasarım probleminde uygulamışlar ve trafik sinyal süresini optimize etmişlerdir. Sonuçları ise TRANSYT-7F trafik modelindeki GA ve Tepe Tırmanış (TT) algoritmaları ile karşılaştırmışlardır. AAA'nın daha basit ve etkili olduğunu, YAKA'nın ise PI değerini % 2,4 ile % 2,7 arasında iyileştirdiğini görmüşlerdir. Araghi ve diğ. [18] çok kavşaklı yol ağlarında kullanılan trafik sinyal kontrolörlerinin Guguk Kuşu Algoritmasıyla

(GKA) optimizasyonunu sağlamışlardır. Yapay Sinir Ağları (YSA) ve Ayarlanabilir Nöron Bulanık Çıkarım Sistemi (ANBÇS) ile geliştirilmiş olan trafik kontrol sistemleri GKA ile optimize edilerek dokuz kavşaklı bir yol ağında kullanılmış ve sabit zamanlı kontrol sistemine göre sırasıyla % 44 ve % 39 daha iyi performans göstermiştir. Zhao ve diğ. [19] Parçacık Sürü Optimizasyon (PSO) yaklaşımına dayalı kavşak kontrolünü Çin'deki bir kavşağa uygulamış ve taşıtların durma sayısını % 19,04 azaltmışlardır. Böylece toplam gecikme ve emisyonu, taşıtların durma sayısını minimize ederek düşürebilmişlerdir.

Sabar ve diğ. [20] güçlü bir yaklaşım olan GA'nın iyi sonuçlar vermesine rağmen yavaş yakınsama özelliğinden dolayı optimizasyon problemlerinde performans düşüklüğü gösterdiğini belirtmişlerdir. Adaptif Memetik Algoritma (MA) ile GA'yı birleştirerek geleneksel MA yaklaşımını iyileştiren araştırmacılar çok kavşaklı bir yol ağının trafik optimizasyonunda önerdikleri bu algoritmayı uygulamışlardır. Avustralya'daki iki farklı kavşak verilerine göre performans değerlendirmeleri yapmış, GA ve sabit zamanlı trafik kontrolünden daha iyi sonuçlar elde etmişlerdir. Erişkin ve diğ. [21] doygünüstü kavşaklarda eleminasyon eşleştirme sistemini (Elimination Pairing System) kullanarak devre süresini optimize etmişlerdir. Gecikme ve durma sayısına göre amaç fonksiyonu oluşturulmuş yöntemin sonuçları Webster ve TRANSYT 14 sinyal süresi yazılımı ile karşılaştırmışlardır. Önerilen yöntemin Webster'den daha iyi sonuçlar vererek trafik sinyal süresi optimizasyonunda kullanılabilir olduğunu göstermişlerdir.

Chen ve Xu [22] PSO algoritmasını trafik sinyal kontrolünde kullanılan BM yaklaşımının üyelik fonksiyonu ve kural tabanı optimizasyonuna uygulamışlardır. Gecikme ve durma sayısını minimize eden amaç fonksiyonuna göre çalışan algoritma ile iki fazda çalışan iki kavşağın optimizasyonunu gerçekleştirmişlerdir. Geliştirdikleri sistemi, sabit kontrollü sistemle karşılaştırmışlar ve gecikme değerinde % 26 oranında iyileşme elde etmişlerdir. Balaji and Srinivasan [23] Tip-2 BM mimarisi ile kentsel trafik sinyal kontrol sistemi geliştirmişlerdir. Singapur'daki 25 farklı kavşak verisine göre test ettikleri sistemi adaptif ve Tip-1 BM kontrol sistemleri ile karşılaştırmışlar ve gecikme değerini yaklaşık % 66 düşürmüşlerdir. Sabetghadam ve diğ. [24] Tip-2 BM trafik sinyal kontrol sistemi geliştirerek kuyruk uzunluğuna göre performans karşılaştırması

yapmışlardır. Sabit zamanlı kontrol ile yapılan karşılaştırmada kuyruk uzunluğunu yaklaşık % 40 azaltabildiklerini göstermişlerdir.

Doğan [25] doktora tezinde, izole sinyalize kavşaklar için BM ile sinyal denetim sistemi geliştirmiştir. İki modül olarak çalışan sistemde, ilk modül fazın süresini ayarlamakta, ikinci modül ise uzatma kararı verilen fazda yeşil sürenin ne kadar uzatılacağını belirlemektedir. Ayrıca modelin performans değerlendirmesi için mikroskobik trafik simülasyon programını (KU-Trsim) geliştirmiştir. BM üyelik fonksiyonları GA ile optimize edilmiş ve % 2,78 ile %32,48 arasında değişen iyileşmeler sağlamıştır. Bi ve diğ. [26] diferansiyel gelişim algoritması (DGA) ile optimize edilmiş Tip-2 BM çoklu kavşak trafik sinyal kontrolünü geliştirmişlerdir. On bir kavşaktan oluşan yol ağında kullanılan yöntem Tip-1 BM kontrol, sabit zamanlı kontrol gibi farklı yaklaşımlarla karşılaştırılmış ve yöntemin gecikme, kuyruk uzunluğu ve park oranını düşürdüğü görülmüştür. Odeh ve diğ. [27] BM ile GA'nın kombinasyonundan oluşan melez bir algoritma geliştirmişlerdir. Dört kavşaktan oluşan yol ağına uygulanan algoritmanın sabit zamanlı gibi geleneksel trafik kontrol sistemlerinden % 34 ve klasik BM kontrol sistemlerinden % 31 oranında daha iyi iyileşme gösterdiğini belirtmişlerdir.

Yu ve diğ. [28] BM yaklaşımına dayalı yeni bir metot ile kapasite, gecikme, durma sayısı ve egzoz emisyonuna bağlı çoklu amaç fonksiyonu ile optimizasyonu gerçekleştirmişlerdir. Ayrıca GA ile devre süresi ve etkili yeşil süreler elde edilip, VISSIM aracılığıyla geleneksel yöntemlerle karşılaştırmışlardır. Bu metot sayesinde gecikme ve durma sayısını azaltırken, kapasitenin artmasını sağlamışlardır. Doğan ve Akgüngör [29] BM trafik kontrolörü geliştirmişler ve BM sistemini DGA ile optimize etmişlerdir. İki seviyede gerçekleştirdikleri optimizasyonda ilk seviyede Üyelik Fonksiyonları (UF)'nın minimum ve maksimum noktalarını belirlemişler ve ikinci seviyede ise belirlenen aralıklarda en düşük gecikmeyi elde edecek şekilde UF'leri yeniden ayarlamışlardır. Bu yöntemle düşük, orta ve yüksek trafik durumlarına göre kavşak performansını sırasıyla % 52, % 48 ve % 14 iyileştirmişlerdir. Araghi ve diğ. [30] aralık Tip-2 BM trafik kontrolörünü Benzetimli Tavlama (BT), GA ve GKA algoritmaları ile optimize ederek kontrol sisteminin performansını iyileştirmişlerdir. Sabit zamanlı kontrole göre GKA yaklaşımı ile % 31, GA yaklaşımı ile % 17 ve BT yaklaşımı ile % 3 iyileştirme sağlamışlardır.

Khooban ve diğ. [31] Modifiye Edilmiş Geri İzleme Arama Algoritması (the Modified Backtracking Search Algorithm) (MBSA) ile Tip-2 BM sistemini optimize ederek geliştirdikleri kontrol sistemi ile sinyal sürelerini ve fazı kontrol etmişlerdir. Geliştirilen yeni sistemin sabit zamanlı ve Tip-1 BM kontrol yaklaşımlarına üstünlük kurarak kuyruk uzunluğunu ve ortalama gecikmeyi düşürdüğünü ortaya koymuşlardır. Chatterjee ve diğ. [32] dinamik trafik verilerini içeren büyük kavşaklardaki belirsizliklerin üstesinden gelmek için Shadowed Tip-2 BM yaklaşımını kullananmışlar ve yeni bir kontrol sistemi geliştirmişlerdir. Modelin üstünlüğünü altı kavşak üzerinde test etmişler ve toplam bekleme süresinde düşüş sağlamışlardır.

Farklı birçok algoritma ile BM ve YSA gibi uzman sistemlerin trafik sinyal kontrol sistemlerinde kullanılarak iyi sonuçlar elde edildiği görülmektedir. Her ne kadar sonuçlar iyi olsa da güncel ve daha üstün performans gösteren algoritmaların trafik sinyal kontrolüne uygulanmaları pozitif katkı sağlayacaktır. Bir diğer önemli hususta birçok kontrol sisteminde süre optimizasyonuna odaklanılmış olması ve faz optimizasyonunun göz ardı edilmesidir. Bir sinyalizasyonun performansında etkili olan süre, faz planı, faz sırası gibi etkenleri kontrol sisteminin aynı anda barındırması ve farklı algoritmaların birlikte çalıştırılması mevcut sistemlerden daha iyi performans sağlayacaktır.

### 3. KAVŞAKLAR VE SİNYALİZASYON SİSTEMLERİ

Uygun bir trafik kontrol sisteminin geliştirilmesinde kavşak yapılarının ve sinyalizasyon temel kavramlarının bilinmesi önem arz etmektedir. Bu nedenle kavşak çeşitleri, kavşakların kontrol şekilleri ve sinyalizasyon konusunun detaylarına bu bölümde yer verilmiştir.

#### 3.1. Kavşaklar

Farklı yönlerden gelen trafik akımlarının birbirleri ile kesiştiği, birleştiği veya ayrıldığı ortaklaşa kullanılan alanlar kavşak olarak adlandırılmaktadır. İki farklı kullanım şekli olan kavşaklar aynı kotta olan ve aynı kotta olmayan olarak birbirinden ayrılmaktadır. Aynı kotta olup zamana bağlı olarak geçiş sıralaması bulunan kavşaklar eş düzey olarak adlandırılırken, trafik akımların farklı kotlardan hareket etmesine olanak sağlayan kavşaklar ise katlı kavşak olarak isimlendirilir. Böylece katlı kavşaklarla aynı anda farklı akımların birbirleri ile çakışmadan geçişine olanak sağlanmakta ve herhangi bir sinyalizasyona ihtiyaç duyulmamaktadır [33].

Zamana bağlı olarak geçiş sıralaması bulunan eş düzey kavşaklarda trafik akımının kesintiye uğraması kaçınılmazdır. Trafik akımındaki bu kesinti ise taşıt gecikmesi, ekstra emisyon salınımı, fazla yakıt tüketimi gibi olumsuzluklara neden olmaktadır. Kavşak performansının bir göstergesi olan gecikme değerinin minimum seviyede olması, üzerinde çalışılan önemli bir konu olmakla birlikte kavşak analizinin de göz ardı edilmemesi gerekmektedir. Kavşak kontrolünün yanı sıra kavşak geometrisi, kavşağın çevresel koşulları, sürücü davranışları vb. gibi birçok farklı parametre de kavşak kapasitesine ve gecikme değerine etki etmektedir.

Bir kavşakta meydana gelen gecikmeyi etkileyen birçok parametre bulunur ancak bu parametrelerin içinden etkisi en fazla olanı, kavşak kontrolünde kullanılan sinyalizasyon sistemidir. Trafik akımına adapte olabilen bir sinyalizasyon sisteminin kullanılması gecikme, kapasite, emisyon vb. gibi performans göstergeleri üzerinde olumlu sonuçlar

verirken uygun olmayan kontrol yapısının kullanılması ise olumsuz etkiler göstererek kavşağın performansının düşmesine sebep olabilmektedir.

Sinyalizasyon sistemi kadar kavşak yapısı da kavşağın performansında etkin rol oynamaktadır. Bir sinyalizasyon sisteminin oluşturulması esnasında kavşak özelliklerinin de bilinmesi uygun çözümlerin üretilmesine katkı sağlamaktadır. Bu nedenle farklı tür kavşakların detaylı olarak incelenmesi önemlidir.

### **3.1.1. Eşdüzey Kavşaklar**

Güvenli bir ulaşımın ve trafik düzeninin sağlanması için trafik mühendislerinin üzerinde çalıştığı önemli konulardan birisi eşdüzey kavşaklar ve bunların kontrolüdür. Bir yol ağındaki özellikle eşdüzey kavşaklar, yol ağının kapasitesini etkilemekte ve bu noktalardaki işletim hataları ise yol ağındaki kapasiteyi de olumsuz olarak etkilemektedir. Aynı zamanda kapasite, gecikme gibi önemli olan hususların yanında trafik güvenliği de bir o kadar önem arz etmektedir. Kapasite ve trafik güvenliğinin birbirini ile ters ilişkili olması optimum kontrolün sağlanmasını kaçınılmaz hale getirmekte ve sürekli üzerinde çalışılarak geliştirilmesi gereken bir konu olmaya devam etmektedir. Her ne kadar uygun bir sinyalizasyon sistemi ile güvenlik ve kapasite arasında en iyi dengenin sağlanmasına çalışılsa da eşdüzey kavşakların geometrik yapıları ile hatalı sürücü davranışları güvenlik kriterini olumsuz etkilemektedir [34]. Güvenlik düzeyinin düşük olması ve kazaların %50'sinden fazlasının eş düzey kavşaklarda meydana gelmesi de bu durumu doğrulayan bir göstergedir [35].

Eşdüzey kavşaklar kontrol şekillerine göre denetimli ve denetimsiz olmak üzere ikiye ayrılmaktadır.

#### **3.1.1.1. Denetimsiz Eşdüzey Kavşaklar**

Herhangi bir sinyalizasyon sisteminin kullanılmadığı sadece işaret levhaları bulunan ya da işaret levhaları bulunmaksızın trafik akımının devamlılığının sağlandığı kavşak türüdür. Bu kavşak türünde geçiş önceliğinin belirlenmesi için ana yol ve tali yol olmak üzere iki kavram bulunmaktadır. Böylece ana yoldan gelen taşıtlara geçiş önceliği

tanınmakta ve tali yoldan gelen sürücülerin buna riayet etmesi ile trafik güvenliği sağlanmaya çalışılmaktadır. Özellikle tali yoldan gelen sürücülerin “YOL VER” ve “DUR” levhaları ile uyarılmaları ile de kazaların azaltılmasına çalışılmaktadır. Ayrıca trafik akımlarının birbirine yakın olduğu denetimsiz kavşaklarda sağdan gelen sürücünün geçiş üstünlüğü ilkesi uygulanarak kaza riskinin azaltılması sağlanmaktadır[36].

#### **3.1.1.2. Denetimli Eşdüzey Kavşaklar**

Sinyalizasyon sistemi ile kontrol edilerek trafik güvenliğinin ve kavşak kapasitesinin artırılmış olduğu kavşaktır. Denetimli eşdüzey kavşaklar dönel kavşak ve sinyalize kavşak olmak üzere iki farklı geometriye sahiptir.

Dönel kavşaklar, taşıt trafiğinin dairesel biçimde hareket etmesini ve sürücülerin isteklerine göre çeşitli doğrultularda dağılmalarına olanak sağlamaktadır. Düşük trafik hızlarında hareketin gerçekleştiği bu kavşak türünde kazalar çoğunlukla maddi hasarlı olup güvenliğin riskli olmadığı durumlarda sinyalizasyon kullanılmamaktadır. Trafik hacmine bağlı olarak merkezinde bulunun yuvarlak adanın haricinde adasız ya da sadece merkezi boya ile belirlenmiş dönel kavşaklar da mevcuttur [33].

Trafik ışıklarına göre geçişin sağlandığı sinyalize kavşaklar trafik akımlarında düzeninin oluşturulmasıyla kapasitenin ve güvenliğin daha artırılmasına olanak sağlamaktadır. Sinyalize kavşaklarda kaza oranının düşmesinde etkili olan unsur hem yayaların hem de taşıtların belirli bir düzen içinde hareket etmeleridir. Ayrıca sinyalizasyon sisteminden dolayı taşıtların hızını azaltması hem taşıt hem de yayaların karıştığı kazalarda ölüm riskini azaltmaktadır[35].

Sinyalizasyon sisteminin temel amaçları arasında bulunan güvenliğin artırılmasının yanı sıra, taşıt ve yayalar için gecikmenin minimum seviye de tutulmaya çalışılması bir diğer temel amacını oluşturmaktadır. Sinyalize kavşakların avantajlarının yanında, devre süresinde yetersizlik, uygun olmayan faz düzeni ve kavşak geometrisinin yeterince analiz edilmemesi gibi sorunlara bağlı olarak ortaya çıkan olumsuz etkileri de mevcuttur. Bu durum, gecikmenin artmasına sebep olmakla beraber, sürücülerin

davranışlarında ve farklı reaksiyon etkilerine bağlı olarak özellikle arkadan çarpma türünde kazaların da artmasına neden olabilmektedir [37].

### **3.1.2. Katlı Kavşaklar**

Genellikle trafik hacminin yoğun olduğu yollarda, trafik akımının kesintiye uğramadan ve mevcut hızın mümkün olduğunca korunarak trafik akışına olanak sağlayan kavşak türüdür. Bu tip kavşaklarda güvenlik ön planda olmakla beraber trafiğin hızında değişme olmaksızın trafik akımının devamlılığını sağlayacak şekilde bir tasarım söz konusudur. Eş düzey kavşaklarda olduğu gibi katlı kavşakların kullanımının da avantaj ve dezavantajları bulunmaktadır. Bahsi geçen kavşakta maliyetin yüksek olmasına rağmen talebin yüksek olduğu, kaza ve gecikmelerin fazla olduğu yüksek seviyeli yollarda etkin bir çözüm olarak kullanılmaktadır [38].

### **3.2. Sinyalizasyon**

Kavşaklarda güvenlik ve kapasitenin artırılması, kaza sayısı ve gecikmenin azaltılması için belli bir düzen dâhilinde çalışan ışıklı sistemler sinyalizasyon olarak adlandırılmaktadır. Sinyalizasyon sistemlerinin temeli trenlerin hareketleri kontrol edilmesine dayanmaktadır. 1800'lü yılların başında itibaren demiryolu kontrolünde kullanılan sinyalizasyonda, geçme işareti olarak beyaz renk, durma işareti olarak kırmızı renk ve uyarı işareti olarak yeşil renk kullanılmaktadır. Kavşaklarda kullanılan sinyalizasyonda ise yeşil renk geç, sarı renk harekete hazırlık ve kırmızı renk dur anlamında kullanılmaktadır.

Karayolundaki sinyalizasyon sisteminin ilk uygulaması 1868 yılına dayanmaktadır ve at arabalarının kavşak içinde oluşturduğu tıkanıklığı düzenlemek için kullanılmıştır. 1914 yılında iki renkli (kırmızı ve yeşil) ışıktan oluşan ilk sinyalizasyon Cleveland/A.B.D.'de uygulanmaya konmuş olup daha sonrasında 1920 yılında sarı ışıklar da kullanıma dahil edilmiştir. 1950 yılından sonra gelişmeye devam eden ışıklı sinyaller birçok ülkede yaygınlaşmıştır [37].

Garrett Augustus Morgan tarafından patenti alınmış olan ve günümüzde halen kullanılan trafik lambaları etkin bir şekilde işlevselliğini sürdürmekte ve yeni teknoloji ve taleplere bağlı olarak gelişmeye devam etmektedir.

Sinyalizasyon sisteminde temel ilkelerin bekleme sürelerinin azaltılması, güvenliğin ve kapasitenin artırılması olmasından dolayı trafik akımının devamlılığının sağlanması, konforun artması, yakıt tüketiminin ve emisyonun azalması gibi çevresel ve ekonomik etkileri söz konusudur. Kavşak kapasitesinin yetersiz kalması, güvenliğin zayıflaması, sıkışıklık, taşıtların katılım ya da geçiş için yeterli boşluğu bulamaması, yayaların kaza risklerinin artması vb. gibi birçok neden sinyalizasyonun gerekliliğini ortaya koyan etmenlerdir. Ancak uygun bir şekilde yapılmayan sinyalizasyon sistemi olumsuz etkiler yaratmakta olup, kavşak performansını daha da kötüye götürmektedir. Aynı zamanda uygun olmayan sinyalizasyon sistemleri sürücülerin davranışını olumsuz etkileyerek kural ihlallerine neden olmaktadır ki bu durum kazaların artmasına sebep olmaktadır.

Sinyalizasyon sistemleri temelde izole ve koordineli sistemler olmak üzere iki farklı çalışma şekline sahiptir. İzole sistemlerin diğer kavşaklardan bağımsız olması ve o kavşağa ait olan trafik hareketlerini içermesi sistemin çalışmasını takip etmeyi ve sinyalizasyonu optimize etmeyi daha basit hale getirmektedir ve böylece doğru bir şekilde sonuç alınabilmesine imkân sağlamaktadır.

### **3.2.1. İzole Sinyalize Sistemler**

Yaygın bir kullanımı bulunan bu sinyalizasyon türünde, diğer kavşaklarda bulunan sinyalizasyonlardan bağımsız olarak kontrol yapısı bulunmaktadır. Bu kontrol yapısı çevrimiçi ve çevrimdışı çalışma olarak iki ana kategoriye ayrılmaktadır. Trafik akımı gibi kavşağa ait herhangi bir verinin sürekli olarak izlenmeden çalışması sağlanan sinyalizasyon sistemi çevrimdışı kategorisine girmekte olup, sabit zamanlı kontrol sistemleri buna en iyi örnektir. Detektör gibi farklı cihazlarla trafik akımı başta olmak üzere kavşak verisinin sürekli izlenmesi ve buna bağlı olarak sinyalizasyona ait devre süresi, yeşil süre ve faz sayısı ve hareketleri gibi operasyonel bileşenlerinin ayarlanması ile çalışan sistemler çevrimiçi olarak adlandırılır ve uyarmalı kontrol sistemleri bu çalışma mantığına göre oluşturulmuş kontrol sistemleridir. İzole sinyalize sistemler dört

başlık altında toplanmakta olup, bunlar; sabit zamanlı, trafik uyarmalı, yaya uyarmalı, el ile kumandalı sinyalizasyon sistemidir.

#### **3.2.1.1. Sabit Zamanlı Sinyalizasyon**

Kavşak kollarındaki trafik akımlarına ve pik saatlerinde oluşan trafik durumlarına göre faz düzeni ve sinyal süreleri (kırmızı, sarı ve yeşil) önceden belirlenmiş olan kavşak işletim sistemidir. Farklı zaman dilimlerine ait birden fazla faz düzeni ve sinyal sürelerinin belirlenebilmesi sisteme iyileştirici bir katkı sağlamaktadır. Bu sistemdeki en önemli nokta ise, kavşağa ait veri toplama işidir. Çünkü bu sinyalizasyon sisteminde devre süresi, faz düzeni gibi sinyalizasyonun parametrelerinin belirlenmesi kollarda bulunan trafik durumuyla doğrudan ilişkili olduğundan dolayı trafik durumunun doğru bir şekilde belirlenmesi sistemin performansı açısından önem arz etmektedir.

Her ne kadar yaygın şekilde kullanılsa da bu sistemin sabit zamanlı olarak çalışması kavşak performansını olumsuz olarak etkilemektedir. Bu olumsuz etkinin temel nedeni trafik akımların stokastik yapısından dolayı değişkenlik göstermesi ve sistemin buna adapte olabilecek bir çalışma mantığına sahip olmamasıdır. Mevsim özellikleri ve kavşağın bulunduğu bölgenin fiziki yapısı gibi trafik durumlarını etkileyen faktörler de kavşak performansını olumsuz etkilemekte ve özellikle gecikmenin artmasına neden olmaktadır.

#### **3.2.1.2. Trafik Uyarmalı Sinyalizasyon**

Sabit zamanlı sistemin aksine taşıtların geçiş hakkı sırası ve süreleri kolların üzerinde bulunan trafik yüküne ve taleplerine bağlı olarak değişkenlik gösteren bir çalışma prensibine sahiptir. Detektörlerin kullanılması ile kavşak kollarındaki trafik akımı bilinmekte ve sinyal süreleri buna göre tayin edilmektedir. Trafik uyarmalı sistemlerin gecikmeyi azaltma, kapasiteyi arttırma, trafik akımındaki değişikliklere adapte olabilme gibi avantajları bulunurken, yüksek kurulum maliyeti ve bakım giderleri gibi dezavantajları da bulunmaktadır.

Detektörlerin kavşak kollarında yerleştirilme durumlarına göre trafik uyarmalı sinyalizasyon sistemi kendi içinde iki kısma ayrılarak, yarı trafik uyarmalı ve tam trafik uyarmalı olarak adlandırılır.

Yarı trafik uyarmalı sistemler kavşak kollarında bulunan trafik akımları arasında farkın fazla olması durumunda uygulanmaktadır. Trafik hacminin fazla olduğu kol veya kollar anayol olarak nitelendirilmekte ve bu kola sürekli yeşil ışık yakarak trafik hareketliliğinin durmasının ve gecikme oluşmasının önüne geçilmesi amaçlanmaktadır. Tali yola ise detektörler konulması suretiyle herhangi bir trafik uyarısı alınması neticesinde bu kola geçiş hakkı verilmesi sağlanarak sistemin düzgün bir işleyişi amaçlanmaktadır [39].

Tam trafik uyarmalı sistemlerde ise, bütün kollara detektörlerin konulması ile trafik durumları sürekli izlenmekte ve bu verilere göre geçiş hakkı sıra ve süresinin değiştirilmesi söz konusudur. Kavşak üzerinde bulunan mevcut taleplerin anlık olarak algılanması ve buna göre sinyalizasyondaki geçiş sürelerinin otomatik olarak tayin edilmesi gecikmenin minimum değerlerde olmasını sağlayan bir etkidir. Tam uyarmalı sistemlerde gelişen teknolojilerin yardımıyla farklı çalışma prensipleri uygulanabilmekte ve böylece faz süresi uzatımı, faz atlama gibi kavşak performansını artırıcı uygulamalara yer verilebilmektedir [39].

### **3.2.1.3. Yaya Uyarmalı Sinyalizasyon**

Sinyalizasyon sistemlerinin görevi trafikteki taşıtların güvenliğini ve düzenini sağlamak olmakla birlikte ulaşımın bir diğer aktörü olan yayalar içinde aynı şekilde geçerlidir. Yaya uyarmalı sistemler kavşaklarda bulunan yaya geçitlerinde, yayaların taleplerine göre yayalara güvenli geçiş hakkı sağlamak için kullanılan sistemlerdir.

Sabit zamanlı ve trafik uyarmalı sistemlere entegrasyonu ile kullanılabilen bu sistemler, yayadan gelen talep doğrultusunda yayaya geçiş hakkı verip, diğer durumlarda anayola devamlı yeşil ışık yakar. Böylece zaman kayıplarını ve gecikmeleri azaltarak hem taşıt hem de yayalar için güvenlik konusunda ve kullanıcıların memnuniyetinde etkili olabilmektedir.

#### 3.2.1.4. El ile Kumandalı Sinyalizasyon

Trafik ve yaya uyarmalı sistemlere benzer bir çalışma mantığı bulunan bu sinyalizasyonda, dışarıdan gözlemler sonucunda taleplere göre manuel olarak bir kumanda yardımıyla sinyallerin değiştirilmesi prensibi uygulanmaktadır [40].

#### 3.2.2. Koordine Sinyalize Sistemler

Trafik hacminin fazla olduğu ana arter üzerindeki kavşakların meydana getirdiği duruş-kalkıştan dolayı oluşan trafik akımındaki düzensizlik ve tıkanıklığın azaltılması için sinyalizasyon sistemlerinin birbiriyle koordineli olarak çalışmasına ve maksimum sayıda taşıdın durmadan geçirilmesine olanak sağlayan sistemlerdir.

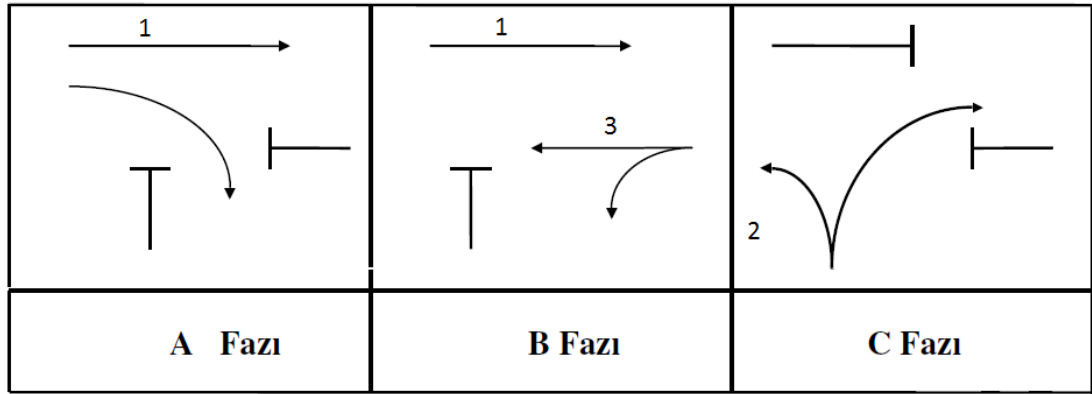
Koordine sistemlerin ana amaçlarından bir diğeri de birbirine yakın olan ve yoğun akımların bulunduğu kavşaklarda biriken taşıt kuyruklarının diğeri kavşak alanına taşmasının engellenmesidir. Bu bağlamda, koordine sistemlerin ana arterler üzerinde uygulanmasının yanı sıra, diğeri yönlerden gelen taleplerin de fazlalığı sonucu bütün kollara uygulanmasıyla minimum gecikme sağlanabilmektedir. Koordine sistemler dört farklı şekilde işletilebilmekte olup; çalışma prensibine göre senkronize, alternatif, progresif ve arazi trafik kontrolü sistemleri olarak ayrılmaktadır.

#### 3.2.3. Sinyalizasyon Temel Kavramları

Sinyalizasyon içerisinde farklı parametreler ve kavramlar bulunmaktadır. Bunların işlevlerinin ve birimlerinin bilinmesi uygun bir sinyalizasyon sisteminin oluşturulmasında önem arz etmektedir. Sinyalizasyonda kullanılan temel kavramlar [41] ve [42] numaralı kaynaklardan elde edilmiştir.

- **Faz:** Kavşaktaki kollarda bulunan akımlardan en az birinin geçiş hakkı alması ve geçiş süresi bitiminde geçiş hakkının sona ermesidir.
- **Devre süresi (C):** Kavşakta bulunun bütün fazlara ait geçiş haklarının tamamlanana kadar geçen toplam süredir.

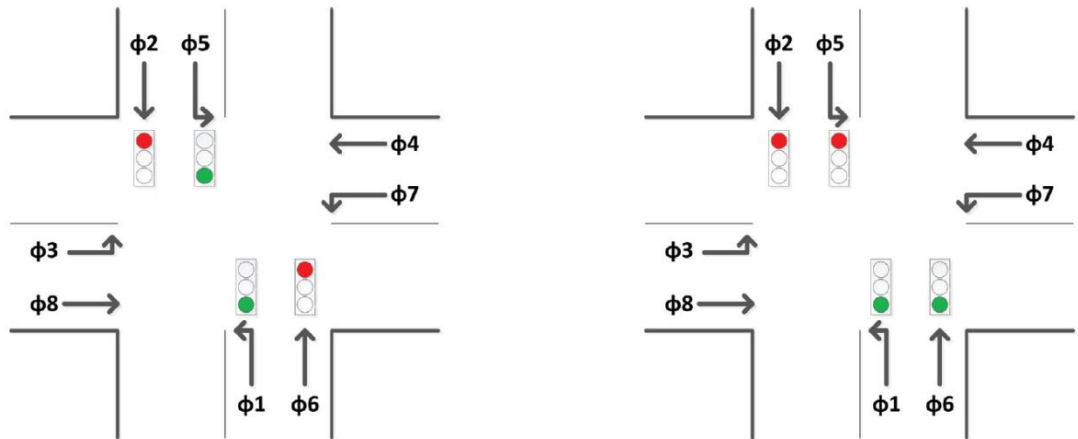
- **Kayıp zaman ( $L$ ):** Kavşakta hiçbir kola hareketin verilmeyip sarı ve her yöne kırmızı süreler boyunca etkisizce geçen süredir.
- **Etkili yeşil süre ( $g$ ):** Geçiş hakkı verilen fazda taşıtların hareket etmesine izin verilen süredir.
- **Yeşil oranı ( $g/C$ ):** Etkili yeşil sürenin devre süresine oranıdır.
- **Yeşiller arası süre ( $I$ ):** Bu zaman periyodu geçiş hakkı bulanan faza ait yeşil sürenin bitip bir sonraki faza ait yeşil sürenin başlaması arasındaki geçen süreye kapsamaktadır.
- **Kırmızı süre ( $r$ ):** Geçiş hakkı verilen faz dışındaki fazlardaki araçların durması için geçen süredir.
- **Sarı süre ( $a$ ):** Kırmızı ile yeşil arasındaki harekete hazırlık için olan süredir.
- **Doygun akım ( $s$ ):** 1 saatlik süre boyunca yeşil sürenin yanmasıyla kavşaktan geçen maksimum araç sayısını içeren akım değeridir. Saatte geçen taşıt sayısı ile ifade edilir.
- **Analiz periyodu ( $T$ ):** Kavşaktaki analizin yapıldığı zaman dilimidir.
- **Kapasite ( $c$ ):** Geçiş hakkı verilen süre boyunca geçebilecek maksimum araç sayısıdır.
- **Doygunluk derecesi ( $x$ ):** Trafik hacminin kapasiteye oranıdır.
- **Faz Düzeni:** Kavşakta mevcut olan birden fazla akımın hangi düzende ve sıra ile geçeceğini gösteren faz diyagramıdır. Birden fazla fazda trafiğin hareket etmesi durumunda tekrarlı akımlar olarak ifade edilir. Şekil 3-1 ile örnek bir faz diyagramı gösterilmiştir.



**Şekil 3-1** Faz diyagramı [43]

### 3.2.4. Sinyalizasyon Faz Düzenlemesi

Faz düzenlemesi, trafik güvenliğini arttırmaya yönelik kesişim sayısının azaltılmasıdır. Ulusal Elektrik Üreticileri Birliği (National Electrical Manufacturers Association – NEMA) bir kavşağın sekiz standart araç fazının numaralandırılması için bir prosedür geliştirmişlerdir ve bu prosedür genellikle NEMA olarak adlandırılmaktadır. NEMA araç faz sistemi kullanılırken, sol dönüş evrelerini belirlemek için tek sayılar kullanılırken, geçiş evresini belirlemek için çift sayılar kullanılır. NEMA gösterimi Şekil 3-2’de verilmiştir.



**Şekil 3-2** NEMA faz numaralandırması [44]

NEMA numaralandırmasında sola dönüş şeridi için 1 sayısı atanmakta ve saat yönünde diğer şeritlere tek sayılar olan 3-5-7 sayıları atanmaktadır. 1 numaralı sola dönüş cebi

ile kesişen doğru akımın bulunduğu şeride 2 sayısı atanmakta ve saat yönünde diğer şeritlere çift sayılar olan 4-6-8 sayıları atanmaktadır. Böylece 1-2, 3-4 gibi kesişim olan hareketlerin ayırt edilebilmesi kolaylaşmaktadır. Burada önemli olan nokta, sola dönüş yapacak taşıtların ayrı bir fazda çalıştırılıp çalıştırılmamasıdır. Eğer ki sola dönüş yapacak trafik hacmi 100 taşıt/saat 'den fazla ise ya da sola dönüş yapacak trafik hacmi ile sola dönen trafik hacmini kesen şeritteki trafik hacminin çarpımı 50.000 araçtan az ise ayrı bir fazda çalıştırılması gerekmektedir [42].

### 3.2.5. Sinyalizasyon Hesapları

Sinyalizasyon hesaplarında iki temel nokta bulunmaktadır. Birincisi, verilerin elde edilmesi, ikincisi ise, devre süresinin hesaplanmasıdır. Geometrik özellikler, trafik hacmi, yaya hacmi, doygun akım gibi devre süresinin belirlenmesinde önemli rol oynayan bu verilerin düzgün bir şekilde toplanması gerekmektedir. Aksi halde yanlış verilere göre devre sürelerinin hesaplanması, kavşak performansını olumsuz etkileyecek ve kapasitenin düşmesi, gecikme, emisyon, yakıt tüketimi değerlerinin artması gibi etkilere yol açacaktır. Devre süresinin hesaplanmasında farklı yöntemler olmakla birlikte, Webster ve Highway Capacity Manual (HCM) yöntemleri literatürde kabul görmüş en yaygın iki yöntemdir.

#### 3.2.5.1. Webster Yöntemi ile Devre Süresi Hesabı

Sinyalize kavşaklar da sinyal sürelerinin hesaplanmasına yönelik öncü çalışma 1958 yılında Webster tarafından geliştirilen ve kendi adı ile anılan yöntemle olmuştur [45]. Webster, faz sayısına, toplam kayıp zamana ( $L$ ) ve kritik akım değerlerine ( $Y$ ) bağlı olarak optimum devre süresinin hesaplanabileceğini ortaya koymuş olup, geliştirdiği yöntemin ifadesi Eşitlik 3.1'de verilmiştir.

$$C_o = \frac{1.5L+5}{1-\sum Y_i} \quad (3.1)$$

Burada,  $C_o$  optimum devre süresi (sn),  $L$  toplam kayıp zaman (sn) ve  $Y_i$  i fazındaki kritik akım oranıdır.

Yeşil sürelerin dağıtımı ise, kritik akımların toplam kritik akımlara oranına göre yapılmaktadır. Matematiksel ifadesi Eşitlik 3.2’de verilmiştir.

$$S_{yi} = \frac{Y_i}{Y} (C_o - L) \quad (3.2)$$

Burada,  $S_{yi}$  i fazına ait yeşil süre (sn),  $Y$  toplam kritik akım oranıdır.

### 3.2.5.2. HCM Yöntemi ile Devre Süresi Hesabı

Bu yöntemde devre süresi kritik doyumluk derecesine ( $x_c$ ) göre belirlenmektedir [41]. Doyumluk derecesi herhangi bir koldaki trafik hacminin kapasiteye oranı ile elde edilebilmektedir. Bu nokta da kapasitenin bulunması gerekmekte ve Eşitlik 3.3 ile hesaplanmaktadır. Doyumluk derecesinin ifadesi ise Eşitlik 3.4’de verilmiştir.

$$c = s \left( \frac{g}{C} \right) \quad (3.3)$$

Burada,  $c$  kapasite (taşıt/saat),  $s$  doyum akım (taşıt/saat),  $g$  yeşil süre (sn),  $C$  devre süresi (sn)’dir.

$$x = \frac{q}{c} \quad (3.4)$$

Burada,  $q$  trafik akımı (taşıt/saat)’tir.

Kavşak kolları bir arada incelenerek kritik doyumluk derecesi ( $x_c$ ) belirlenmektedir. Fazlara ait kollar arasından en yüksek doyumluk derecesi,  $i$  fazının kritik doyumluk derecesi olarak tayin edilmektedir. Eşitlik 3.5’ de bütün kritik hacimler bir arada değerlendirilerek, kavşağa ait  $x_c$  belirlenmektedir. Eşitlik 3.5’ de yerine konan değişkenler yardımıyla devre süresi belirlenmektedir.

$$x_c = \sum_i \left( \frac{v}{s} \right)_{ci} \frac{c}{C-L} \quad (3.5)$$

Burada,  $x_c$  kritik doyumluk derecesi ve  $\sum_i \left( \frac{v}{s} \right)_{ci}$  kritik akım/doyum akım oranıdır.

## 4. YAPAY ZEKÂ TEKNİKLERİ VE BULANIK MANTIK

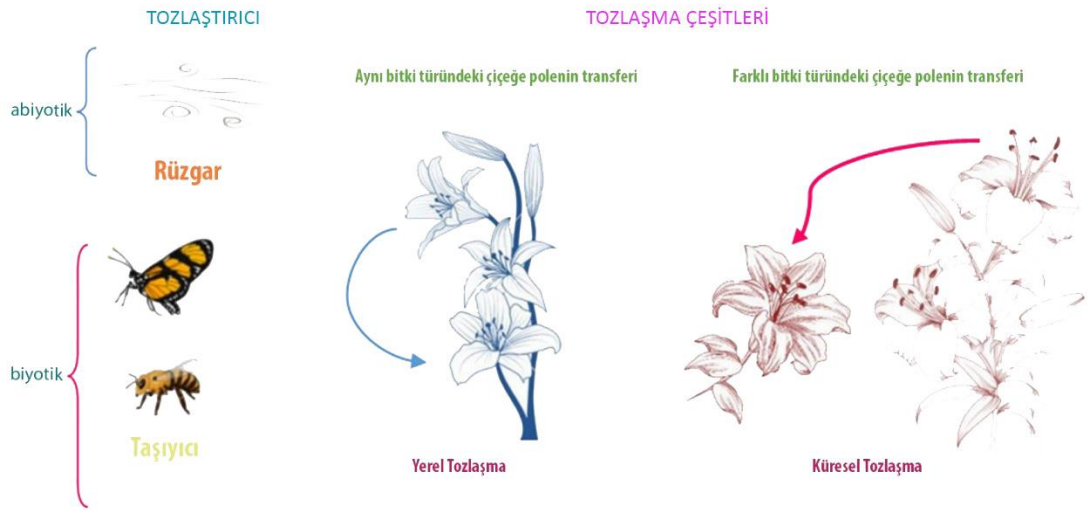
### 4.1. Çiçek Tozlaşma Algoritması

Evrimin biyolojik penceresinden bakıldığında çiçeklerin tozlaşmasındaki amaç, türün hayata devam etmesi, sayı ve nitelik bakımından en uygun popülasyonun üretilmesidir. Bu durum çiçekler tarafından gerçekleştirilen bir çeşit optimizasyon sürecidir. Polenle üreme, doğal dünyanın en ilginç süreçlerinden birisidir ve bu sürecin evrimsel karakteristiği 2012 yılında Yang tarafından, Çiçek Tozlaşma Algoritması (ÇTA) adı ile sunulmuştur [46]. Ortaya konulduğu günden beri hızlı bir şekilde popülaritesini arttıran bir algoritma olmuş ve farklı alanlardaki birçok araştırmacının dikkatini çekmiştir. Farklı alanlardaki uygulamaları ve mevcut yapay zekâ yaklaşımları ile performans kıyaslamalarında rekabet edici özelliklere sahip olduğu ve üstünlük gösteren sonuçları ortaya koyduğu belirtilmiştir. Pant ve diğ. algoritmanın basit ve esnek bir çalışma mantığına sahip olmasının yanı sıra çoklu amaç fonksiyonlarını içeren problemlerin çözümünde de etkin olmasının fazla talep görmesinde önemli etkenler olduğunu vurgulamışlardır. Ayrıca medikal, enerji, görüntü işleme, yapısal dizayn gibi farklı alanlarda etkin kullanımları olduğunu ortaya koymuşlardır [47]. Kayabekir ve diğ. diğer algoritmalarla birleştirerek performansı iyileştirilmiş olan ÇTA yaklaşımının kimya mühendisliği, inşaat mühendisliği, enerji ve güç sistemleri, makine mühendisliği, elektronik ve haberleşme mühendisliği, bilgisayar bilimleri gibi alanlarda iyi sonuçlar verdiğini belirtmişlerdir [48]. Literatürde geniş bir uygulama alanı bulunan ÇTA algoritmasının geliştirilme sürecinin devam ettiği ve daha iyi performanslar ortaya koyabildiği bilinmektedir. Özellikle farklı alanlardaki uygulamaların olması ve melez formlarının da etkin sonuçlar göstermesi algoritmanın iyi bir performansa sahip olduğunu ortaya koymaktadır. Sharawi ve arkadaşları kablosuz sensör ağlarının kümelenmesinde ÇTA yaklaşımını kullanmışlardır. Düşük enerji uyarlamalı kümeleme hiyerarşi yöntemi ile sonuçları karşılaştırarak ÇTA yönteminin üstünlüğünü vurgulamışlardır [49]. Sakib ve diğ. sürekli optimizasyon problemlerinin çözümünde ÇTA ve Yarasa Algoritması (YA) yöntemlerini uygulamışlardır. İki algoritmanın performanslarını karşılaştırarak ÇTA yaklaşımının daha geniş çözüm uzayında arama yapmasından dolayı nihai çözüm kalitesi açısından YA'ya göre daha iyi performans

gösterdiğini belirtmişlerdir [50]. Lukasik ve Kowalski, ÇTA yaklaşımını etkinliği bilinen bir algoritma olan PSO ile karşılaştırmışlardır. ÇTA'nın sürekli optimizasyon problemlerindeki deneysel çalışma sonuçlarına dayanarak oldukça rekabetçi olduğunu kanıtlamışlardır. Yerel ve küresel arama mekanizmalarının algoritmayı daha etkin hale getirdiğini belirtmişlerdir. Ayrıca, az sayıda parametre ile kontrol edilebilmesinin algoritmanın tercih edilen bir yöntem olmasında etkili olduğunu belirtmişlerdir [51]. El-Henawy ve Ismail, ÇTA ve PSO yaklaşımlarını birleştirerek melez bir yaklaşım geliştirmişlerdir. Büyük tam sayılı programlama sorunlarının çözümünde kullandıkları yeni yaklaşımın arama uzayında doğru çözüm noktasını bulmada çok faydalı olduğunu ifade etmişlerdir [52]. Abdel-Raouf ve arkadaşları formüle edilmiş sudoku bulmacalarını çözmek için ÇTA ve Kaotik Harmoni Aramasını (KHA) kullanarak yeni bir melez optimizasyon yöntemi geliştirmişlerdir. Geliştirilen yöntemi kullanarak en uygun çözümü elde ettiklerini araştırma sonuçlarında göstermişlerdir. Önerilen yöntemin üstünlüğünü kanıtlamışlar ve bu yöntemin daha başarılı sonuçlara ulaşmasının nedenini ÇTA yaklaşımının arama gücü olarak belirtmişlerdir [53]. Nabil, ÇTA ve Klonal Seçim Algoritması (KSA) kullanılarak modifiye edilmiş ÇTA (MÇTA) yaklaşımını geliştirerek 32 farklı fonksiyonda önerilen yöntemi test etmiştir ve elde ettiği sonuçları GA, YA ve BT gibi farklı optimizasyon teknikleriyle karşılaştırmıştır. Karşılaştırma sonuçları, geliştirilen yönteminin diğer algoritmalara göre daha umut verici ve doğru olduğunu ortaya koymuştur [54]. Kalra ve Arora, Ateşböceği Algoritması (AA) ve ÇTA'nın bileşimi olan melez bir algoritma geliştirmişlerdir. Önerilen algoritmayı matematiksel fonksiyonlar üzerinde test ederek AA ve ÇTA'dan daha umut verici sonuçlar verdiğini göstermişlerdir [55]. Arora ve Anand sürekli optimizasyon problemlerini çözmek için kaos ile geliştirilmiş MÇTA algoritmasını önermişlerdir. Arama alanı aralığının tanımlanmasında adım boyutunu tanımlamak için klasik ÇTA yaklaşımı rasgele sayılarına ihtiyaç duyulmaktadır. ÇTA yaklaşımında kaotik sayı üretme tekniğini kullanmışlar ve önerilen yöntemi matematiksel fonksiyonlara göre test etmişlerdir. Simülasyon sonuçları, kaotik sayı oluşturma yaklaşımının ÇTA performansını artırmak için yararlı olduğunu göstermiştir [56]. Abdel-Basset ve Shawky, ÇTA'yı GA, PSO, KKO, GKA, Çekirge Optimizasyonu Algoritması (ÇOA), Gri Kurt Optimizasyonu (GKO) olan altı farklı meta-sezgisel yaklaşımla karşılaştırmıştır. Karşılaştırma sonuçlarına göre, ÇTA'nın diğer meta-

sezgisel yaklaşımdan daha hızlı yakınsamaya sahip olduğunu belirtmişlerdir. Ayrıca ÇTA'nın daha düşük hata ile algoritmalarından daha iyi performansa sahip olduğunu göstermişlerdir [57].

Doğadaki bitkilerin yaklaşık %80'inin çiçekli bitkiler olduğu tahmin edilmektedir. Bu bitki türlerinin günümüzden 125 milyon yıl önce Cretaceous döneminde ortaya çıkıp dünyaya nasıl hâkim olduğu hala gizemini korumaktadır. Çiçekli bitkilerin üreme davranışının temelinde en uygun biyolojik canlılığın devam etmesi yatmakta ve bunun için tozlaşma yöntemi kullanılmaktadır. Bu nedenle bitkiler iki farklı tozlaşma yöntemi kullanılmaktadırlar. Kelebek, kuş, böcek gibi herhangi bir canlı tozlaştırıcı ile sağlanan tozlaşma yöntemi biyotik tozlaşma olarak adlandırılır ve bitkilerdeki tozlaşmanın %90'ı bu yöntemle gerçekleşmektedir. Su ya da rüzgâr gibi canlı olmayan bir taşıyıcı ile polenlerin taşınması durumu ise abiyotik olarak adlandırılmakta ve %10 oranında tozlaşmada kullanılmaktadır. Yerel tozlaşma ve küresel tozlaşma olarak da ifade edilen tozlaşma yöntemlerinde, küresel tozlaşma ile farklı çiçeklerden gelen polenler ve yerel tozlaşma ile aynı çiçeğin polenleri kullanılmaktadır. Tozlaştırıcılar ve tozlaşma çeşitleri Şekil 4-1'de gösterilmektedir.



**Şekil 4-1** Tozlaştırıcılar ve tozlaşma çeşitleri [57]

Belli kabullere göre en uygun şekilde çözümü arayan bu algoritmanın dört temel kuralı vardır.

Kural 1: Levy uçuşuna göre taşıyıcıların tozlaşmayı sağlaması küresel tozlaşma işlemidir ve biyotik tozlaşma olarak adlandırılır.

Kural 2: Abiyotik tozlaşma işlemi yerel tozlaşma işlemidir.

Kural 3: Üretim olasılığı çiçek sabiti adı verilen bir katsayı ile sağlanmakta ve çiçek türlerinin benzerliğiyle orantılı değişmektedir.

Kural 4: Yerel ve global tozlaşma işlemi [0-1] aralığında geçiş olasılığı değişkeni ile kontrol edilir.

Gerçekte bitkiler birden fazla çiçeğe ve milyonlarca polene sahiptir. Bu algortmada ise basitlik açısından her bir bitkinin bir çiçeğe sahip olduğu ve bir polen üretebildiği varsayılmıştır. Böylece bir çiçek ya da polen gameti çözüm noktasını ifade edebilmektedir. Global aramada Levy dağılımına göre böceklerin uzun süre uçabilmesi sağlanmakta ve durum polenlerin uzun mesafede taşınabilmesine olanak sağlamaktadır. Böylece en iyi üreme garanti edilmektedir. Global aramanın matematiksel ifadesi Eşitlik 4.1’de verilmiştir.

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) \quad (4.1)$$

Burada,  $x_i^t$  t. döngüdeki çözüm vektörü ve  $g_*$  en iyi sonuçtur.  $\gamma$  adım boyutunu ayarlayan kontrol parametresidir. Eşitlik 4.2’de gösterilen Levy dağılımı ( $L$ ) ise tozlaşmanın gücüne tekabül eden Levy uçuş temelli adım büyüklüğüdür.

$$L \sim \frac{\lambda \Gamma(\lambda) \sin \frac{\pi \lambda}{2}}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_o > 0) \quad (4.2)$$

Burada  $\Gamma(\lambda)$  standart gama fonksiyonudur ve  $s$  Levy dağılımının adım büyüklüğündedir. Bu dağılım  $s > 0$  büyük adımlar için geçerlidir. Teoride,  $s_o \gg 0$  gereklidir, ancak uygulamada  $s_o$  0.1 kadar küçük olabilir.

Levy dağılımındaki rastgele adım büyüklüğün üretilmesi önemlidir. Bunun için birkaç farklı yöntem bulunmakta olup bunların en etkin olanı Mantegna algoritmasıdır. Mantegna algoritması  $U$  ve  $V$  ile gösterilen iki Gauss dağılımı kullanmaktadır. Mantegna algoritmasına göre adım boyutunun ifadesi Eşitlik 4.3’de verilmiştir.

$$s = \frac{U}{|V|^{1/\lambda}} \quad U \sim N(0, \sigma^2), V \sim N(0,1) \quad (4.3)$$

Burada,  $U \sim (0, \sigma^2)$  ortalaması 0 ve varyansı  $\sigma^2$  olan örneklemdir. Varyansın ifadesi Eşitlik 4.4’de verilmiştir.

$$\sigma^2 = \left[ \frac{\Gamma(1+\lambda)}{\lambda \Gamma(\frac{1+\lambda}{2})} \frac{\sin(\frac{\pi\lambda}{2})}{2^{\frac{\lambda-1}{2}}} \right]^{1/\lambda} \quad (4.4)$$

Lokal tozlaşma çözümlerin yakınsanmasında etkilidir. Böylece çözüm noktasının küçük bir komşuluğunda arama süresini kısaltarak en optimum noktayı bulabilir. Lokal tozlaşmanın ifadesi Eşitlik 4.5’te verilmiştir.

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t) \quad (4.5)$$

Burada  $x_j^t$  ve  $x_k^t$  rastgele seçilen ve aynı bitki türlerinin farklı çiçeklerinden gelen polen türüdür.

ÇTA yaklaşımının çalışma mantığı aşağıdaki şekilde psedo kodu ile verilmiştir.

Amaç Fonksiyonu:  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)$

Başlangıç popülasyonunu rastgele oluştur. (n adet Çiçek/Polen gameti)

Başlangıç popülasyonundaki en iyi çözümü hesapla  $g^*$

Geçiş olasılığı katsayısını  $p \in [0, 1]$  aralığında belirle

*İken*( $t < \text{Maksimum iterasyon}$

*döngü*  $i = 1:n$

*Eğer* *rastgele*  $< p$  (Üniform dağılım)

*Levy dağılımı*  $L$  (Parametre sayısı kadar)

*Biyotik üreme*  $x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t)$

*ise*

*Düzenle  $\varepsilon$  (üniforma dağılıma göre [0,1])*

*Rastgele  $j$  ve  $k$  çözümlerini seç*

*Abiyotik üreme  $x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t)$*

*eğer bitir*

*Yeni çözümü al ve kontrol et*

*Yeni çözüm daha iyi ise popülasyonu güncelle*

*döngü bitir*

*En iyi çözümü seç  $g_*$*

***İken bitir***

#### **4.2. Karga Arama Algoritması**

Optimizasyon yöntemlerinde temel iki amaç bulunmaktadır. Bunlardan birincisi elde edilen sonuç ile beklenen sonuç arasındaki hatayı minimize etmek, ikincisi ise çözüm arama süresini düşürmektir. Bu doğrultuda birçok farklı optimizasyon algoritması geliştirilmiş olup Karga Arama Algoritması (KAA)'da bu optimizasyon yöntemlerinden birisidir. Askarzadeh [58] karga sürülerinin akıllı davranışından ilham alarak popülasyon tabanlı sezgisel algoritmayı geliştirmiştir. 2 parametre ile kontrol edilebilen bir algoritma olduğundan dolayı farklı alanlara kolaylıkla uygulanabilmektedir. GA, PSO ve AAA gibi farklı algoritmalarla karşılaştırıldığında daha üstün bir performans göstererek çözüme hızlıca ulaşabilmektedir. Zolghadr-Asli ve diğ. uçuş uzunluğu ve farkındalık olasılığı ile kontrol edilebilen KAA yaklaşımının hem basit hem de kompleks mühendislik problemlerinde başarılı bir şekilde uygulanabilir olduğunu belirtmişlerdir. Farklı alanlara uygulamalarının ve modifiye edilmiş versiyonlarının olması, yeni bir algoritma olmasına rağmen fazla talep görmesine sebep olmaktadır [59]. Abdelaziz ve Fathy, KAA yöntemini radyal dağıtım şebekelerinde iletken boyutunun optimum seçimi için uygulamışlardır. Güç kaybını ve toplam yıllık sistem işletme maliyetini azaltma için optimum iletken boyutu belirlemişler ve yaklaşımın performansını ortaya koymak için Evrimsel Programlama (EP) ve AAA yöntemleriyle karşılaştırmışlardır. EP ve AAA'dan daha iyi performans gösteren KAA yaklaşımı ile 16-hatlı ağda güç kaybında % 33,32 ve yıllık maliyette % 24,48, 85-hatlı ağda ise güç

kaybında % 19,53 ve yıllık maliyette % 16,17'lik tasarruf sağlanabilmişlerdir [60]. Mohammadi ve Abdi, KAA yöntemini modifeye etmişler ve ekonomik yük dağıtım probleminde uygulanabilir olduğunu göstermişlerdir. YAKA, PSO, GA gibi diğer yaklaşımlarla karşılaştırarak çözüm kalitesi, sağlamlığı ve hesaplama süresi açısından, diğer tekniklerden daha üstün bir performansa sahip olduğunu ortaya koymuşlardır [61]. Javidi ve arkadaşları optimum yapı dizayn probleminde daha iyi performansa sahip olan İyileştirilmiş Karga Arama Algoritması (iKAA) yaklaşımının uygulanabilirliğini ortaya koymuşlardır. Minimum ağırlık, yakınsama hızı ve güvenilirlik açısından standart KAA'dan daha iyi olan İKAA yöntemi diğer sezgisel yöntemlerden de daha iyi ya da rekabetçi sonuçlar elde etmiştir [62].

KAA algoritması çalışmasında dört temel prensip bulunmakta olup bunlar;

- Kargalar sürü şeklinde yaşar.
- Kargalar saklanma yerlerinin konumunu ezberler.
- Kargalar hırsızlığı yapmak için birbirini takip eder.
- Kargalar, hırsızlıklara karşı tedbirlidirler ve yakalanma olasılıklarına karşı kendilerini korurlar.

KAA yöntemi kargaların zekalarını kullanarak yiyeceklerini saklama, sakladıkları yerden geri getirme ve başka kargalara ait yiyecekleri çalabilme kabiliyetlerini modelleyerek çalışmaktadır. Bu nedenle KAA yönteminde kargaların yiyecek arama davranışları iki durumda modellenmektedir. Birinci durum sakladığı yiyecek kaynağına giden karganın başka bir karga tarafından takip edildiğini bilmemesidir. Böylece takip eden karga kolaylıkla yiyecek kaynağına ulaşabilmektedir. KAA yönteminde takip eden karganın yeni pozisyonunu Eşitlik 4.6'da verilen matematiksel ifadeye göre belirlenmektedir.

$$x^{i,iter+1} = x^{i,iter} + r_i x f^{l,iter} x (m^{j,iter} - x^{i,iter}) \quad (4.6)$$

Burada  $r_i$  0 ile 1 arasında üniform dağılımlı rastgele bir sayı ve  $f^{l,iter}$  i. döngüdeki uçuş uzunluğunu,  $x^{i,iter}$  i. karganın pozisyonunu ve  $m^{j,iter}$  j. karga tarafından saklanan yiyecek kaynağının yerini ifade eder.  $Iter$  ise iterasyon sayısının kısaltmasıdır.

Algoritmanın parametrelerinden birisi olan  $fl$  algoritmanın arama yeteneğini etkilemektedir. Daha küçük  $fl$  değerlerinin yerel aramaya yol açmakta, daha büyük  $fl$  değerleri ise arama alanını genişletmektedir. Optimizasyon açısından, daha küçük  $fl$  değerleri sonuçları yoğunlaştırmaya yardımcı olurken, daha büyük  $fl$  değerleri sonuçları çeşitlendirmektedir.

İkinci durum ise yiyecek kaynağına giden karganın başka bir karga tarafından takip edildiğini bilmesidir. Bu durumda takip edilen karga zekasını kullanarak takip eden kargayı yanıltmakta ve başka bir yöne götürmektedir. Böylece takip eden karga arama uzayının herhangi bir pozisyonuna giderek çözüm noktasından uzaklaşmaktadır. Takip eden karganın yeni pozisyonu Eşitlik 4.7’de verilen ifadeye göre belirlenmektedir.

$$x^{i,iter+1} = \begin{cases} x^{i,iter} + r_i x fl^{i,iter} x(m^{j,iter} - x^{i,iter}) & r_j \geq AP^{j,iter} \\ rastgele pozisyon & \text{diğer} \end{cases} \quad (4.7)$$

Burada  $r_i$  0 ile 1 arasında uniform dağılımlı rastgele sayı ve  $AP^{j,iter}$  j. döngüdeki farkındalık olasılığını ifade eder.

Sezgisel algoritmalarda performansı etkileyen iki önemli kavram vardır ve bunlar çeşitlendirme ve yoğunlaşmadır. Bunların arasında iyi bir denge kurulması algoritma performansında etkin bir rol oynaktadır. Farkındalık olasılığı (FO) parametresi KAA yaklaşımında bu amaçla kullanılmaktadır. Çeşitliliğin artırılmasında FO değeri artırılarak arama uzayı global ölçeğe genişletilmekte,  $AP$  değeri azaltılarak ise yerel ölçekte arama yapılarak yoğunluk artırılmaktadır.

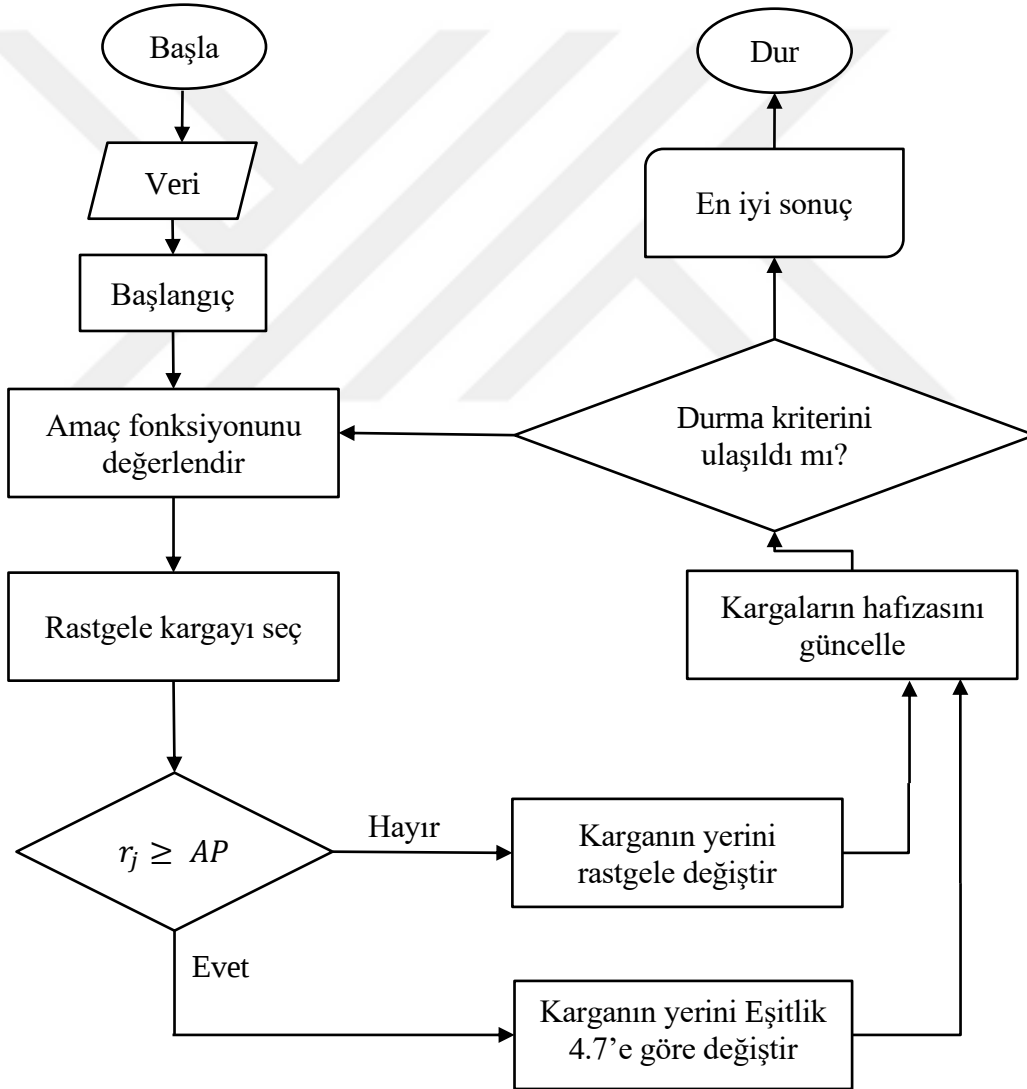
Meta-sezgisel bir algoritmanın verimli bir şekilde uygulanabilmesi için algoritmanın parametrelerinin ayarlanması gerekir. Bununla birlikte, parametre ayarı zaman alan bir işlemdir. Böylece, sınırlı sayıda parametreye sahip algoritmaların farklı optimizasyon problemlerinde uygulanması daha kolaydır. Yukarıda bahsedilenler doğrultusunda ayarlama gerektiren sadece iki ana parametreye ( $fl$  ve  $AP$ ) sahip olması KAA'yı birçok geleneksel meta-sezgisel algoritmalara göre avantajlı hale getirmektedir. Algoritmaların en uygun çözümü bulmasının yanı sıra en hızlı şekilde bunu gerçekleştirebilmesi de algoritmayı avantajlı hale getirmekte ve talep görmesini sağlamaktadır. Bunun için uygun olmayan cevaplardan kaçınmak adına, KAA’da yeni yerin uygunluğunu kontrol

etmek önerilir. İkinci işlemde uygun olmayan bir yer oluşturulursa, karga sabit kalmalıdır. Böyle bir prosedür için mümkün olmayan cevaplar olduğunda bir ceza fonksiyonunun uygulanmasıdır. Ceza fonksiyonunun ifadesi Eşitlik 4.8’de verilmiştir.

$$m_{(i,t)} = \begin{cases} x_{(i,t+1)} & \text{if } f[x_{(i,t+1)}] \text{ daha iyi ise } f[x_{(i,t)}] \\ m_{(i,t)} & \text{Diğer} \end{cases} \quad (4.8)$$

Burada  $f$  amaç fonksiyonudur.

KAA yaklaşımının akış diyagramı ise Şekil 4-2’de gösterilmektedir.



Şekil 4-2 Karga arama algoritması akış diyagramı

### 4.3. Diferansiyel Gelişim Algoritması

Gerçek değerli ve sürekli verilerin olduğu problemlerin çözümünü sağlayabilen bir algoritma olan Diferansiyel Gelişim Algoritması (DGA) popülasyon tabanlı olup, GA ile benzerlik göstermektedir [63]. DGA temelde 4 operatör ile çalışmakta olup, bunlar başlangıç popülasyonu, mutasyon, çaprazlama ve seçimdir. DGA'yı GA'dan ayıran en önemli fark ise mutasyon operatöründe olmaktadır. DGA'nın rastgele seçilen vektörlerin farkına göre mutasyon işlemi gerçekleştirmesi bu operatörü adaptif hale getirmektedir. Bu farklar uygun bir adım boyutunun belirlenmesinde önemlidir. Çünkü, algoritma olarak optimum değişimlere doğru atılacak adım boyutu belirlenmesi önemlidir ve mutasyon operatörünün adaptif olmasından dolayı da algoritmanın performansını geliştirmekte ve daha güçlü yapmaktadır. Bu sayede popülasyon üyeleri arasındaki varyans arttığında ya da azaldığında DGA'daki adım boyutu da buna göre değişmektedir. Ayrıca DGA yaklaşımında operatörlerin bütün popülasyona uygulanmaması, az satır ile kodlanması gibi avantajları algoritmanın tercih edilirliliğini arttırmaktadır. DGA'nın diğer sezgisel yöntemlere göre diğer bir üstünlüğü de çok az parametre ile kontrol edilebilmesidir. Sadece 3 tane kontrol parametresi vardır ki bunlar mutasyon katsayısı F, çaprazlama katsayısı CR ve popülasyon büyüklüğüdür NP'dir. Algoritmaya ait 4 operatör aşağıda detaylı olarak verilmiştir.

- **Başlangıç popülasyonu:**

Herhangi bir sezgisel arama yönteminde başlangıç popülasyonunu önem arz etmektedir. Çünkü uygun bir şekilde oluşturulduğunda en iyi sonuç daha kolay bulunabilir ve hızlı bir şekilde yakınsama olabilmektedir. Başlangıç popülasyonu en az 3 kromozomdan oluşmakta olup, popülasyon boyutu girdi değişken sayısına (D) göre belirlenmektedir. Başlangıç popülasyonu 0 ile 1 arasında değişen rastgele dağılıma göre belirlenmektedir. Başlangıç popülasyonu üretmenin ifadesi Eşitlik 4.9'da verilmiştir.

$$x_{j,i} = x_j^{min} + rand_j(0,1)(x_j^{max} - x_j^{min}) \quad (4.9)$$

Buradaki  $x_j^{max}$  ve  $x_j^{min}$  her bir parametrenin alt ve üst sınırı olup  $i = 1, \dots, NP$  (Popülasyonun sayısı) ve  $j = 1, \dots, D$  (parametre sayısı) 'dır.

- **Mutasyon:**

Mutasyon işlemi kromozomdaki genler üzerinde yapılmış olan bazı değişiklikler olarak ifade edilmektedir. Böylece alternatif bir çözüm olan kromozom değerleri uzayda hareket ederek farklı bir noktada çözümü temsil etmektedir. Mutasyon yapıldığında iki şeye dikkat edilmesi gerekmektedir. Bunlar mutasyonun yönü ve değişim miktarıdır. Mutasyon işleminde genellikle bir ya da iki tane fark vektörü kullanılır. Price'e göre iki tane fark vektörü kullanmak uygun mutasyon katsayısını bulmayı kolaylaştırır[63]. Mutasyon işleminde rastgele seçilen 3 kromozoma ihtiyaç duymaktadır. Mutasyon işleminin ifadesi Eşitlik 4.10'da verilmektedir.

$$x_i'^{(G)} = x_a^{(G)} + F(x_b^{(G)} - x_c^{(G)}) \quad i = 1 \dots N_p \quad (4.10)$$

Burada  $x_i'^{(G)}$  mutant vektörü,  $x_a^{(G)}$  baz vektör, G nesil numarası ve F mutasyon katsayısıdır.  $x_b^{(G)}$  ve  $x_c^{(G)}$  fark vektörünün oluşturmak için kullanılan rastgele seçilmiş vektörlerdir.

Mutasyonda DGA/x/y/z şeklinde ifade edilen birçok strateji mevcuttur. Burada mutant vektörün oluşturulmasında kullanılan baz vektörü x, mutasyon işlemindeki fark vektörünün sayısı y ve çaprazlama işlemindeki çaprazlama şeması z ile ifade edilmektedir. X ile ifade edilen baz vektörünü seçerken “rand, best, rand to best, current ve current to best and current to rand” olarak adlandırılan yöntemler kullanılmaktadır. Mutasyonda farklı yöntemler olup literatürde en çok tercih edilen 4 yöntemin matematiksel ifadeleri Eşitlik 4.11-14'de verilmiştir.

$$\text{DGA/rastgele/1/ikili} : V_{i,g} = X_{ro,g} + F(X_{r1,g} - X_{r2,g}) \quad (4.11)$$

$$\text{DGA/eniyi/1/ikili} : V_{i,g} = X_{best,g} + F(X_{r1,g} - X_{r2,g}) \quad (4.12)$$

$$\text{DGA/eniyi/2/ikili} : V_{i,g} = X_{best,g} + F(X_{r1,g} - X_{r2,g} + X_{r3,g} - X_{r4,g}) \quad (4.13)$$

$$DGA/mevcuttan eniyiye/1/ikili : V_{i,g} = X_{i,g} + F(X_{best,g} - X_{i,g}) + F(X_{r1,g} - X_{r2,g}) \quad (4.14)$$

- **Çaprazlama:**

Mutasyon işleminden sonra oluşan mutant vektörü ile hedef vektörünün çaprazlamasıyla deneme vektörü elde edilmektedir. Bu işlem sırasında üç farklı çaprazlama metodundan birisi ile çaprazlama katsayısı kullanılır. Çaprazlamada kullanılan metotlar Binomial, exponential, aritmetiktir. En yaygın olarak kullanılan binomial çaprazlama metodunun ifadesi Eşitlik 4.15 ile verilmektedir.

$$x_i^{''(G)} = \begin{cases} x_i^{'(G)} & \text{if } rastgele_j(0,1) \leq C_r \text{ or } j = j_{rnbr(i)} \\ x_i^{(G)} & \text{diğer} \end{cases} \quad (4.15)$$

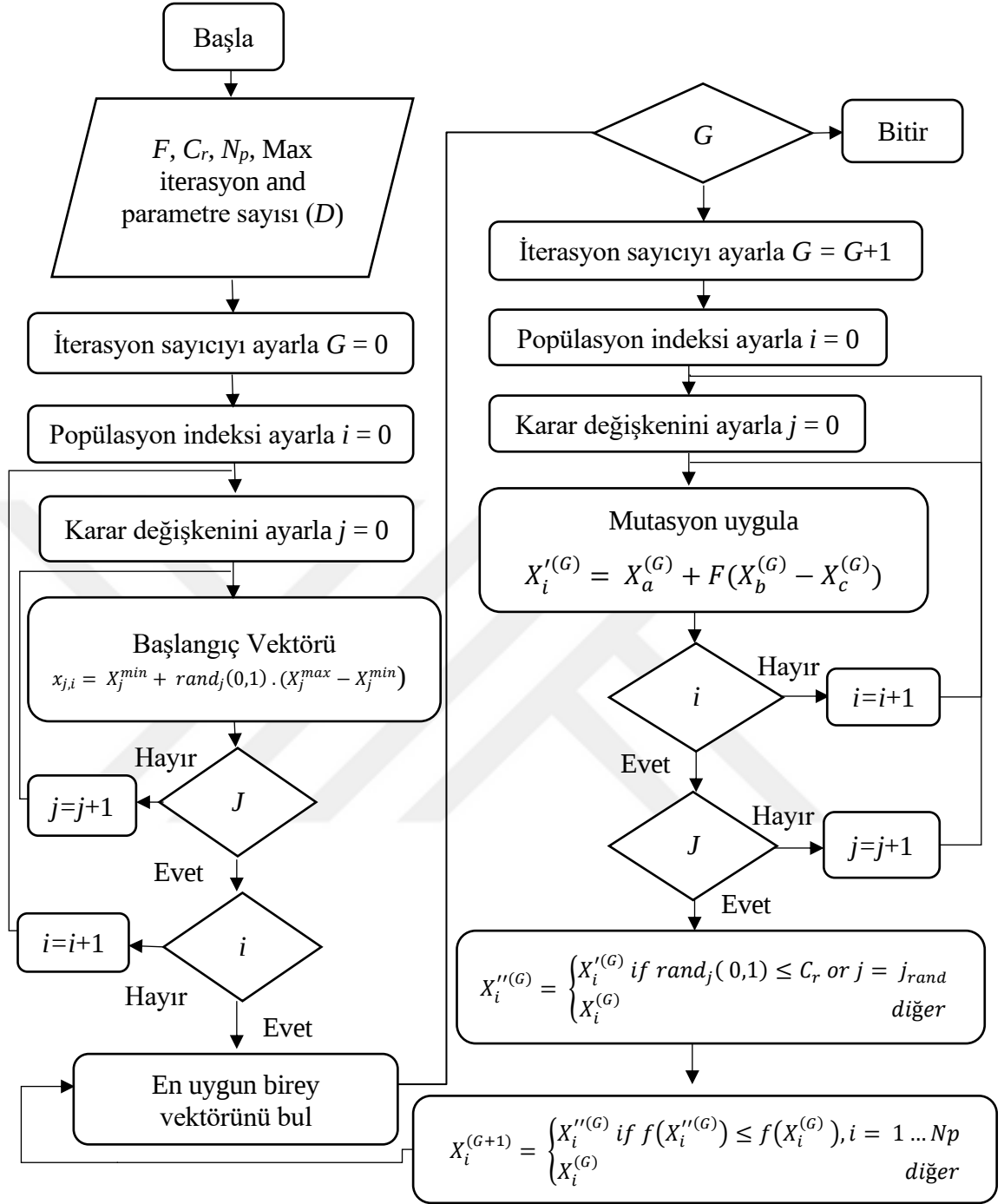
Burada  $x_i^{''(G)}$  deneme vektörü olup  $rastgele_j$  üniform dağılıma göre 0 ile 1 arasında değişir.  $rnbr(i)$  1 ile D arasında değişir.

- **Seçim:**

DGA'da seçim iki adaylı bir turnuva olarak da adlandırılabilir. Uygunluk fonksiyonunu minimize edecek şekilde deneme vektörü ile hedef vektörü arasında en iyi seçimi yaparak yeni jenerasyonu oluşturmaktır. Seçim operatörünün ifadesi Eşitlik 4.16'da verilmektedir.

$$x_i^{(G+1)} = \begin{cases} x_i^{''(G)} & \text{if } f(x_i^{''(G)}) \leq f(x_i^{(G)}), i = 1 \dots Np \\ x_i^{(G)} & \text{diğer} \end{cases} \quad (4.16)$$

DGA yaklaşımının akış diyagramı Şekil 4-3'de verilmiştir.



Şekil 4-3 Diferansiyel gelişim algoritması akış diyagramı [64]

#### 4.4. Tip-2 Bulanık Mantık Yöntemi

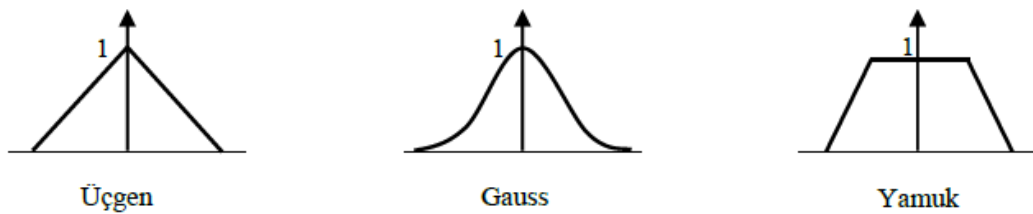
Bulanık Mantık (BM) matematiksel ifadelerle tanımlanması ve çözülmesi zor olan problemlerde alternatif bir yaklaşım olarak ortaya çıkmıştır. Klasik küme teorisinde bir nesnenin kümenin elemanı (1) ya da değil (0) mantığı her problemin tanımlanmasında kullanılamazken, BM yaklaşımında nesnelerin  $[0,1]$  arasında bir üyelik derecesine sahip olması ile belirsizliklerin daha kolay ifade edilmesine olanak sağlamaktadır. Bir diğer husus ise bilgisayarların karar verebilme sürecinde sadece kesin değerlere göre değil aynı zamanda insan beyninde olduğu gibi sözel ifadelerin ve düşünmenin de karar verme sürecine dahil edilebilmesidir. Böylece insanın düşünme mantığına daha yakın bir sistemle bilgisayarlar tam ya da kesin olmayan bilgilere göre daha doğru kararlar verebilmektedir.

BM yaklaşımı Zadeh tarafından 1975 yılında ortaya konulmuştur [65]. Mutlak kesin diye bir şeyin olmadığı bu yaklaşımda her şey 0 ile 1 arasında ifade edilebilmektedir. Diğer bir deyişle her şey bulanık küme kuramıyla ifade edilmektedir ki bu küme de elemanlar arası kesin bir ayrım yoktur ve sınırları açıkça belli değildir. Bulanık kümenin ifadesi Eşitlik 4.17’de gösterilmektedir.

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (4.17)$$

Burada, A bulanık küme,  $\mu_A(x)$  üyelik fonksiyonu ve x nesne koleksiyonudur.

Üyelik fonksiyonları üçgen, yamuk, gauss, sigmoid vb. gibi farklı şekillerde ifade edilebilmekte olup, bazı üyelik fonksiyon şekilleri Şekil 4-4’de verilmiştir.



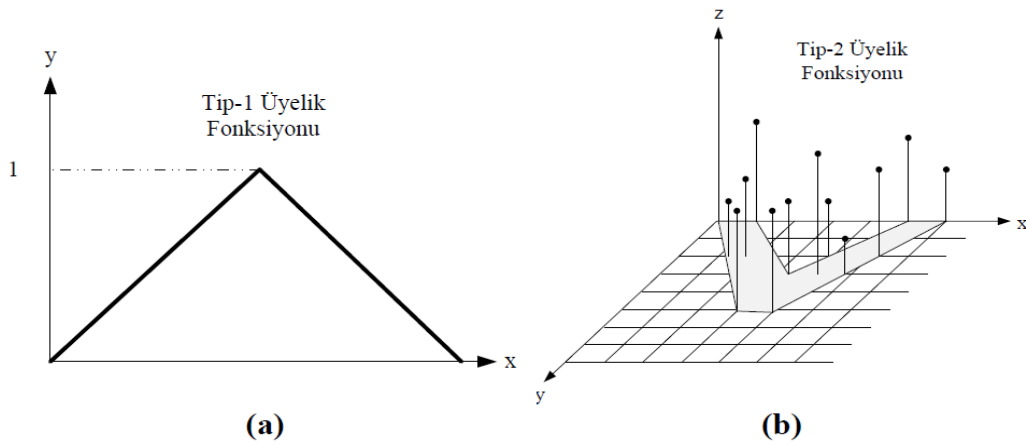
**Şekil 4-4** Üyelik fonksiyon şekilleri

Şekil 4.4’de gösterilen üyelik fonksiyonu şekillerinde y eksenı üyelik derecesini ifade etmekte olup 0 ile 1 arasında bir deęer almaktadır. Eęer üyelik derecesi deęeri 0-1

arasında kesin bir değer ise Tip-1 BM olarak ifade edilmektedir. Tip-1 BM yaklaşımı yüksek dereceli belirsizliklerin olduğu problemlerde yetersiz kalmaktadır. Aynı zamanda üyelik fonksiyonları, sınır değerleri ve kural tabanı yeni belirsizliklerin ortaya çıkmasına neden olmaktadır. Tip-1 BM sisteminde diğer nedenlerden dolayı meydana gelen belirsizlikler aşağıda sıralanmıştır.

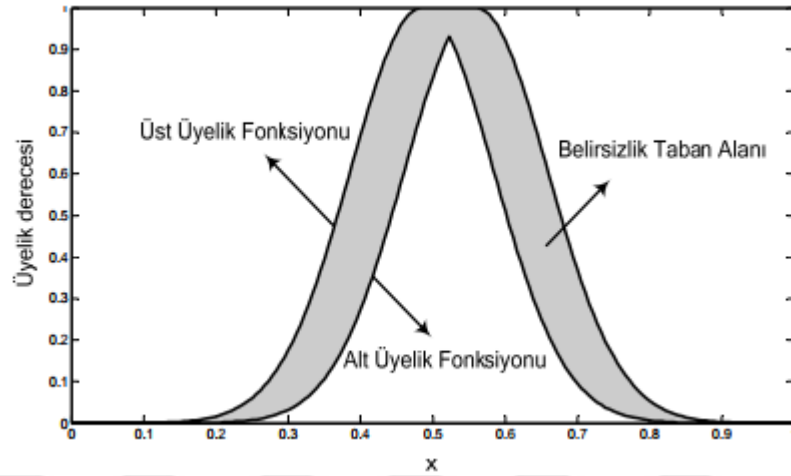
- Kuralların sözel ifadesinde kullanılan kelimelerde anlamsal belirsizlikler oluşturabilir.
- Farklı uzmanlardan alınan bilgide düşünce farklılıkları olabilir.
- Tip-1 bulanık mantığın çalışmasında kullanılan ölçüm verileri gürültü içerebilir.

Tip-1 BM’de üyelik derecelerinin kesin değer olmasından dolayı bu belirsizlikler direkt olarak modellenemezler. Bu nedenle Tip-2 BM kavramı ortaya çıkmıştır. Tip-2 BM’de ise her bir parametrenin üyelik dereceleri bulanık küme içerdiğinden dolayı üyelik fonksiyonları ile tanımlanmaktadır. Bu durum serbestlik derecesinin artmasını sağlamakta ve fazladan serbestlik derecesi ile Tip-2 BM’de belirsizliklerin daha iyi ifade edilebilmesi mümkün olmaktadır. Tip-2 bulanık kümelerin kullanımı belirsizliklerin olmasına göre sıklık göstermekte olup bir kümeye ait üyelik fonksiyonunu tanımlanırken üyelik fonksiyonu net olarak belirlenemiyorsa tercih edilmektedir. Tip-1 ve Tip-2 bulanık kümelerde mantık birbirine benzerken temel farklılık boyutsal olarak ortaya çıkmaktadır. Yani, Tip-1 bulanık kümelerde iki boyutlu üyelik fonksiyonları kullanılırken, Tip-2 bulanık kümelerin üyelik fonksiyonları üç boyutludur. Tip-1 ve Tip-2 üyelik fonksiyonları Şekil 4-5’de gösterilmiştir.



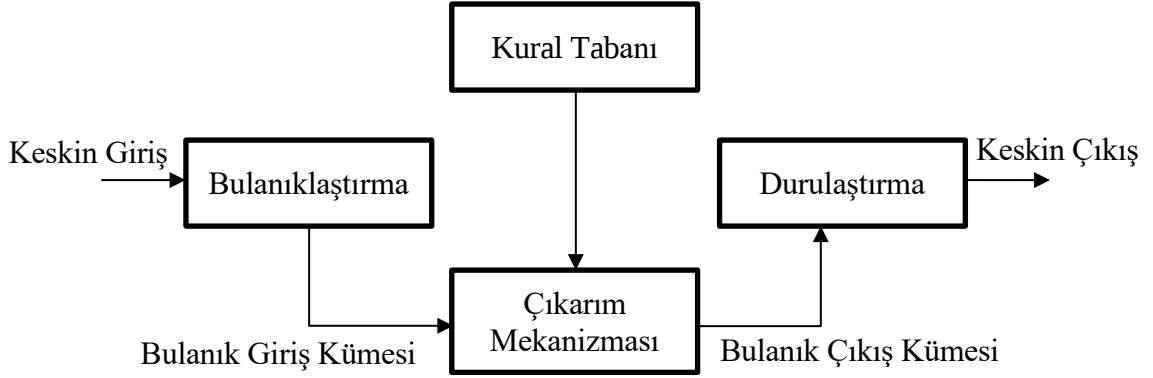
**Şekil 4-5** Tip-1 ve Tip-2 üyelik fonksiyonları [66]

Tip-2’de üyelik fonksiyonu iki adet Tip- 1 üyelik fonksiyonu ile sınırlandırılmış olup bulanık kümesinin belirsizlik taban alanı bu iki fonksiyonla belirlenen sınırlar arasındadır. Bir diğer ifade ile üç boyutlu görüntüsü olan Tip-2 bulanık kümenin iki boyuta iz düşürülmüş hali belirsizlik alanı olarak ifade edilmekte ve bir Tip-2 bulanık kümelerin belirsizlik tabanı alanı ve alt sınır üyelik fonksiyonları Şekil 4-6’de verilmektedir.

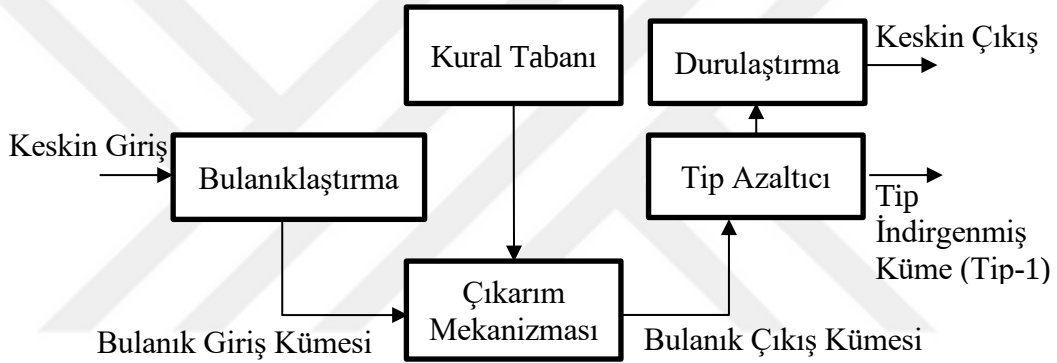


**Şekil 4-6** Tip-2 bulanık kümenin belirsizlik taban alanı, alt ve üst üyelik fonksiyonları[67]

Tip-2 BM’de önemli bir diğer nokta ise Tip-1 bulanık kümelerini barındırmasıdır. Çünkü Tip-2 bulanık kümelerde bulunan belirsizlikler ortadan kaldırıldığında Tip-2 bulanık küme yaklaşımı Tip-1 bulanık kümeye dönüşmektedir. Bu durum rastgeleliklerin ortadan kalkması durumunda olasılık kuramının deterministik kurama dönüşmesine benzetilebilir. Tip-2 BM sisteminin yapısı Tip-1 BM sistemine çok benzemektedir. Tek farklılık Tip-2 BM işlem bloğunda tip azaltıcı işlemin bulunmasıdır. Böylece son işlem olan durulaştırma ile keskin çıkış değerinin basit şekilde elde edilmesine olanak sağlamaktadır. Tip-1 BM ve Tip-2 BM sistem yapısı Şekil 4-7 ve 4-8’de verilmiştir.



**Şekil 4-7** Tip-1 Bulanık mantık sistem yapısı [66]



**Şekil 4-8** Tip-2 Bulanık mantık sistem yapısı [66]

Şekil 4-7 ve 4-8’de gösterilen BM sistem yapılarının işlem bloklarında bulunan bulanıklaştırma işlemi ile keskin giriş değerlerinin bulanık kümelerle dönüştürülmesi sağlanmaktadır. Bulanık kümeler arasındaki değişim aralığı, net bir kıstas olmamakla birlikte tecrübeye dayalı olarak belirlenmektedir. Bu değişim aralığının ve bulanık mantık kümesindeki maksimum ve minimum değerlerinin uygun bir şekilde belirlenebilmesi farklı bir problem konusunu da ortaya çıkarmaktadır. Kural tablosu sistemin çalışmasında önemli bir yere sahiptir. Giriş çıkış değişkenleri arasındaki mantıksal bağlantıyı oluşturduğu için sistemin karar vermesinde doğrudan etkilidir. Bu nedenle giriş çıkış değişkenleri arasındaki bütün durumların göz önüne alınarak kuralların oluşturulması ve oluşturulan kuralların birbiriyle tutarlı olması gerekmektedir. Mantıksal bağlantının kurulmasında Eğer-İse yapısı kullanılmaktadır. Çıkarım mekanizması işlem bloğu sistemin karar merkezidir. Oluşturulan kurallara ve bulanık

kümeyle baęlı olarak ıkış kümesini oluřturmaktadır. Bir veya birden fazla kurala göre ıkarım yapılmasına göre iki farklı yaklařım bulunmaktadır. Bir kurala göre ıkarım yapılmasında “ve” ya da “veya” baęlalarına göre ıkarım yapılmaktadır. ıkış kümelerinin “ve” ya da “veya” baęlalarına göre üyelik fonksiyonu ifadeleri sırasıyla Eřitlik 4.18 ve 4.19’da gösterilmiřtir.

$$\mu_C = EB\{EK(\mu_A(x), \mu_B(y))\} \quad (4.18)$$

$$\mu_C = EB\{EB(\mu_A(x), \mu_B(y))\} \quad (4.19)$$

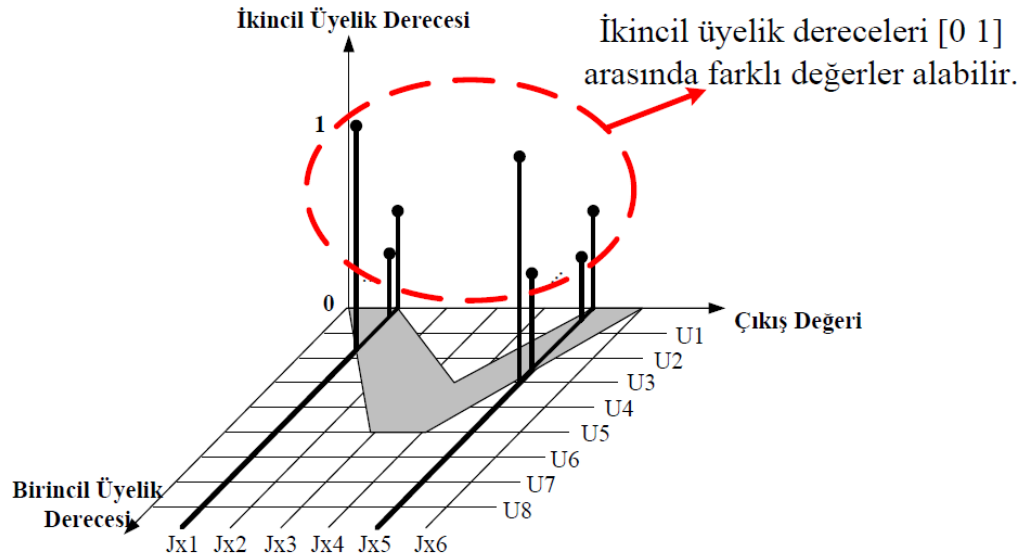
Burada,  $A$  ve  $B$  girdi kümeleri  $C$  ise ıktı kümesidir.  $EK$  ve  $EB$  en küçük ve en büyük deęerleri ifade etmektedir.

Birden fazla kurala göre yapılan ıkarımlarda ise her kurala göre elde edilen sonuların en büyük noktalarından birleřtirilmesi ile sonu kümesi elde edilmektedir. Son iřlem olan durulařtırma da ise bulanık ıkış kümesinden keskin ıkış deęerine dönüřtürölür. Durulařtırma iřleminde birok farklı yöntem bulunmakta olup aęırlık merkezi ya da en büyük alanın ilk, orta ve son deęerleri gibi yöntemler en sık tercih edilen yöntemlerdendir. Tip-2 BM sisteminde önemli bir yere sahip olan tip azaltıcı iřleminde birok farklı teknik geliřtirilmiř olup bunlardan en ok kullanılanı Karnik-Mendel (KM) algoritmasıdır [68]. Dięer tip azaltıcı yöntemleri ise ařağıda sıralanmıřtır.

- İyileřtirilmiř KM Algoritması (İKM)
- Durma Kriterli İteratif Algoritma (DKİA)
- İyileřtirilmiř Durma Kriterli İteratif Algoritma (İDKİA)
- İyileřtirilmiř Karřı Yön Arama Algoirtması (İKYAA)
- Wu-Mendel Belirsizlik Limit Metodu (WM)
- Nie-Tan Metot (NT)
- Begian-Melek-Mendel Metot (BMM)

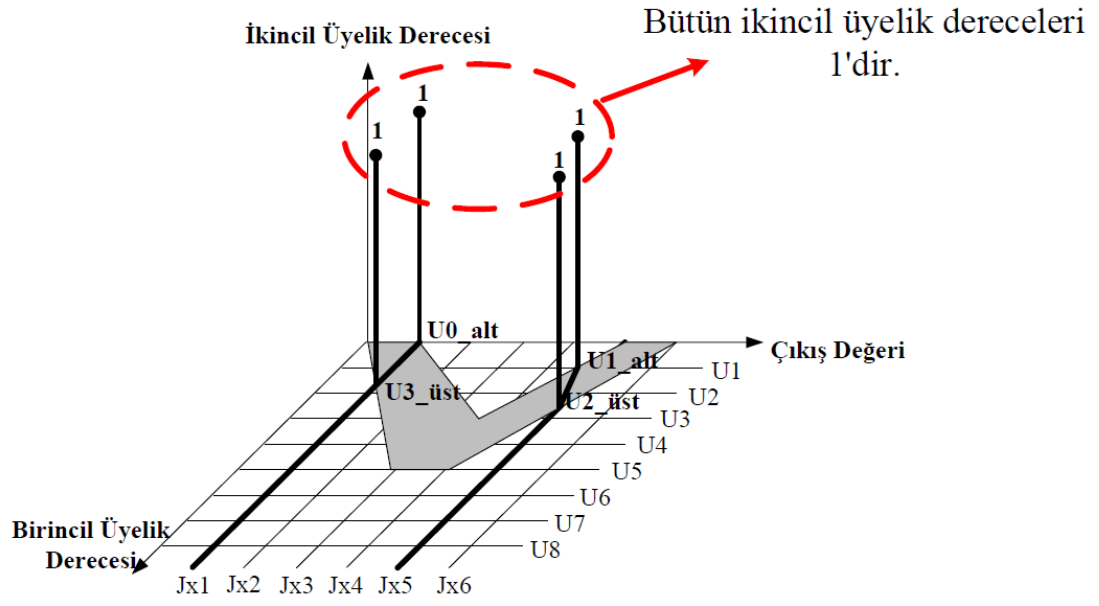
Taskın ve Kumbasar [68] tip azaltmada kullanılan teknikleri minimum ve maksimum hesaplama sürelerine göre karşılaştırarak performanslarını ortaya koymuştur. Bu çalışmaya göre WM tekniği en düşük hesaplama süresine sahip oldurken karar verebilmesi hızlı bir şekilde gerçekleşmektedir.

Tip-2 BM’da 2 farklı sistem bulunmaktadır. Bunlar genel Tip-2 BM ve aralıklı Tip-2 BM’dir. Genel Tip-2 BM’de üyelik fonksiyonun ikincil üyelik derecesi  $[0, 1]$  aralığındadır. Bu nedenle ikincil üyelik dereceleri 0-1 arasında farklı değerler alabildiği için sistem modellemesinde üçüncü boyutunda etkisi bulunmaktadır. Buna bağlı olarak genel Tip-2 BM sistem modellerinde işlem karmaşası daha fazla olmaktadır. Genel Tip-2 BM üyelik fonksiyonu Şekil 4-9’da gösterilmiştir.



**Şekil 4-9** Genel Tip-2 üyelik fonksiyonu [66]

Aralıklı Tip-2 BM sisteminde ise üçüncü boyutun etkisini ortadan kaldırarak işlem karmaşasını azaltmak için ikincil üyelik dereceleri 1 olarak kabul edilmektedir. Böylece alt ve üst üyelik derecesi değerlerine göre işlem yapılmaktadır. Aralıklı Tip-2 BM üyelik fonksiyonu Şekil 4-10’da gösterilmektedir.



Şekil 4-10 Aralıklı Tip-2 üyelik fonksiyonu [66].

## 5. TRAFİK SİMÜLASYON PROGRAMI

Trafiğin stokastik yapısı, farklı trafik durumları, yeni kontrol sistemlerinin arazi uygulamaları ile test edilebilmesinde yaşanan sıkıntılar (yasal izin, yeni sistemin oluşturabileceği olumsuzluklar vs) ve yüksek maliyet gibi birçok etken arazi uygulamalarında zorluklar yaratmaktadır. Bu nedenle araştırmacılar bilgisayar teknolojileri kullanarak bir sistemin benzetimine odaklanmışlardır. Bu doğrultuda CORSIM, VISSIM, AIMSUN, PARAMICS gibi birçok farklı simülasyon programı geliştirilmiş, kullanılmış ve geliştirilmeye de devam edilmektedir. Bu benzetim programları kavşak modellenmesi, kontrol sistemlerinin test edilmesi, yol ağlarında performans artırıcı tekniklerin uygulanabilirliği gibi karayolu trafiğiyle alakalı her türlü çalışmanın kolay bir şekilde uygulanabilmesine ve değerlendirilebilmesine olanak sağlamaktadır. Trafik mühendisliğinde kullanılan simülasyon programları kendi içerisinde makroskobik, mikroskobik ve mezoskobik olarak üç ana kategoriye ayrılmaktadır.

Simülasyonların geliştirilmesinde kurgulanan yaklaşımlarda makroskobik yaklaşım en basit tekniktir. Sistem için kurulan tek bir matematiksel model korunurken, sistemin tamamını ilgilendirmeyen çeşitli hesaplamalar ihmal edilir. Hız ( $u$ ), yoğunluk ( $k$ ) ve akıma ( $q$ ) bağlı olarak trafiğin karakterize edildiği bu yaklaşımda sistemin tümünün davranış modellemesi yapılarak hızlı çözümler üretilebilir ama hassaslık daha azdır. Mikroskobik yaklaşım daha karmaşık yapıya sahip olup detaylı olarak fiziksel özellikler ve performans özellikleri dikkate alınmaktadır. Bir diğer ifadeyle problemin parçaları ve ayırık elemanların tüm özellikleri hesaba katılmaktadır. Makroskobik simülasyondaki kullanılan sistemin tümünün basitçe modellenmesi yaklaşımı yerine, mikroskobik simülasyonda trafiğin karakterize edilmesi için her bir taşıtın bireysel olarak değerlendirilmesi yaklaşımı kullanılmaktadır. Böylece her bir taşıtın hız, ivme, araç tipi gibi spesifik özellikleri ele alınarak trafik akımının daha detaylı modellenmesi sağlanmaktadır. Mikroskobik yaklaşım, daha hassas ve güvenilir sonuçlar verebilirken, bazı durumlarda daha fazla maliyet ve zaman gerektirebilmektedir. Bu yaklaşımda her bir araç bireysel olarak ele alındığı için diğer araçlarla olan ilişkileri de hız, ivme gibi karakteristiklere doğrudan etki etmektedir. Mikroskobik yaklaşımda hız, ivme gibi taşıt

parametrelerinin belirlenmesinde farklı modeller kullanılmaktadır. Bunlardan en yaygın olanları aşağıda sıralanmıştır.

- Gazis-Herman-Rothery (GHR) modeli
- Güvenli mesafe modeli
- Psikolojik fiziksel araç takip modeli

Her üç modelde de araçlar arası mesafe hız ve ivme gibi parametrelerin belirlenmesinde güvenlik temel kriterdir. Mezoskopik yaklaşım gerçeğe en uygun model olup, Mikroskopik ve makroskopik yaklaşımların üstün yönlerini birleştirilmesine dayanmaktadır.

Mevcut simülasyon programlarının birçok avantajının olmasının yanı sıra bazı olumsuz yönleri de bulunmaktadır. Özellikle Matlab, C gibi programlama dilleri ile entegrasyonunun olmaması ya da bazı kısıtların bulunması bu simülasyon programlarının her koşulda kullanılmasına imkân vermemektedir. Herhangi bir programlama dili aracılığıyla geliştirilen trafik kontrol sistemlerinin mevcut simülasyonlarla birleştirilmesinde yaşanan güçlükler, trafik mühendislerini yeni simülasyon programları geliştirmeye yönlendirmiştir. Doğan [25] yapmış olduğu doktora tezinde bu problemle karşılaşarak Kırıkkale Üniversitesi Trafik ve Simülasyon (KU-Trsim) ismini verdiği bir mikroskobik simülasyon programı geliştirmiştir. Geliştirdiği simülasyon programını mevcut bir kavşak verisi ile karşılaştırarak istatistiki olarak doğrulamış ve programlama dilleri ile birlikte etkin bir şekilde çalışabilir olduğunu ortaya koymuştur. Bu çalışmada, KU-Trsim programı geliştirilen kontrol sistemlerinin test edilmesinde kullanılmak üzere uygun bulunmuş olup sistemin ihtiyaçlarına göre revize edilmiştir. Bu tez çalışmasında mevcut programın araç üretim modülünde herhangi bir değişiklik yapılmamakla birlikte araç dinamikleri modülü kuyruk uzunluğunun ve şerit bazlı araç sayılarının elde edilmesi için tekrar düzenlenmiştir. Sinyal denetim modülü ise çalışma kapsamında geliştirilen kontrol sistemlerine göre oluşturulmuştur. Ayrıca programın farklı kontrol sistemlerinde kullanım kolaylığı sağlaması için girdi ara yüzleri ve kontrol sisteminin yapısına uygun

çıktı ara yüzleri de Matlabda bulunan grafiksel kullanıcı ara yüzü (Graphical User Interface-GUI) ile oluşturulmuştur.

### **5.1. Revize edilmiş KU-Trsim Programı**

Matlab programı kullanılarak geliştirilen KU-Trsim programında araç üretimi, araç dinamikleri ve sinyal denetimi olmak üzere üç ana modül bulunmaktadır. Araç üretimi modülü araçların sisteme giriş yapmalarını sağlamakta olup araçlar arası giriş aralıklarını ve hız, araç tipi gibi özelliklerini belirlemektedir. Bu modülde araçlar arası geliş aralıkları üstel dağılıma göre belirlenmektedir. Araç dinamikleri modülünde sisteme giren araçların hareketleri kontrol edilmektedir. Bu modülde araçların güvenli mesafede birbirini takip etmesi, yavaşlama, hızlanma ve durma durumlarının kontrol edilmesi, ivme, hız ve koordinat gibi dinamik parametrelerin ayarlanması sağlanmaktadır. Ayrıca detektörler ile şeritlerdeki araç sayıları ve kuyruk uzunlukları bilgileri de bu modül içinde takip edilmektedir. Sinyal denetim modülü de simülasyon programının sinyalizasyon kısmını oluşturmaktadır.

#### **5.1.1. Araç Üretim Modülü**

Trafiğin stokastik yapısından dolayı araçların gelişleri rastgele oluşmaktadır. Bu nedenle araç üretim modülünde araç gelişlerinin gerçekte olan yapıya uygun olabilmesi için rastgele sayıların üretilmesi gerekmektedir. Matlab’da bulunan “rand” fonksiyonu rastgele sayıların üretilmesini sağlamakta olup bir başlangıç sayısına bağlıdır. Bu nedenle farklı trafik durumlarını oluşturan farklı gelişlerin elde edilebilmesi farklı “çekirdek (seed)” numaralarının kullanılması ile sağlanabilmektedir. Eğer, çekirdek numarası aynı tutulursa aynı geliş aralıkları, değiştirilirse farklı geliş aralıkları oluşturulabilmektedir. Bu durum gerçekte olması çok zor olan aynı trafik senaryolarının gerçekleştirilmesini sağlayarak farklı kontrol yaklaşımlarının performansını ortaya koyabilmek için kullanılmaktadır. Araç üretim modülünde araç gelişlerinin üstel dağılıma uygun olduğu ki kare testi ile kontrol edilmiş ve araç üretim modülünün ürettiği araç gelişlerinin kullanılabilir olduğu görülmüştür [25]. Aynı zamanda bu modülde araç geliş aralıklarının yanı sıra araç dağılımı ve giriş hızları da belirlenmektedir. 5 farklı araç tipinin kullanıldığı bu simülasyon programında her araç tipinin boyut, hızlanma ve

yavaşlama ivmeleri gibi farklı özellikleri bulunmaktadır. Otomobil, otobüs, minibüs, kamyon ve tır olarak 5 farklı aracın simülasyonda kullanılan uzunlukları CORSIM [69] simülasyon programı ve Doğan'ın [25] geliştirdiği KU-Trsim programının mevcut hali ile uyumlu olacak şekilde belirlenmiş ve Çizelge 5-1'de verilmiştir.

**Çizelge 5-1** Araç tipleri ve uzunlukları

| Araç Tipi | Uzunluk (m) |
|-----------|-------------|
| Otomobil  | 4,50        |
| Minibüs   | 6,00        |
| Kamyon    | 8,50        |
| Otobüs    | 12,00       |
| Tır       | 13,50       |

Araçların giriş hızlarının belirlenmesinde Greenshields'in hız ve yoğunluk ilişkisi kullanılmaktadır. Böylece yoğunluğa bağlı olarak sisteme giriş hızı belirlenmektedir. En yüksek hız olarak ise sistemde tanımlanan serbest akım hızı kullanılmaktadır. Greenshields hız yoğunluk ilişkisinin ifadesi Eşitlik 5.1'de verilmiştir.

$$v_s = v_f - \frac{v_f}{k_j} k \quad (5.1)$$

Burada;  $v_s$  mesafe anlamlı hız,  $v_f$  serbest akım hızı ve  $k_j$  tıkanıklık yoğunluğudur.

### 5.1.2. Araç Dinamikleri Modülü

Simülasyon programı içerisindeki araçların hareketlerinin kontrol edildiği, güvenlik mesafesi, hız, ivme, konum gibi dinamik parametrelerinin belirlendiği ve tutulduğu modüldür. Bu modül sayesinde araçların simülasyon içerisinde yer değiştirmeleri ve yavaşlamaları, durmaları, hızlanmaları gibi davranışların modellenmesi sağlanmaktadır. Dinamik parametrelerin kontrol edildiği bu modülde aynı zamanda detektörler vasıtasıyla istenen aralıklarda araç sayımı yapılabilmekte ve kuyruk uzunlukları da ölçülebilmektedir. Bu modüle ait Matlab kodu Ek 1'de verilmiştir. Verilen kod üzerinde yapılan değişiklikler koyu yazım ile gösterilmiştir.

Simülasyon programında araç hareketlerinin gerçekçi bir şekilde benzetilebilmesi için her bir araç tipine ait ve hızlara bağlı olarak değişen ivmelenme değerlerinin uygun şekilde belirlenmesi gerekmektedir. Bu doğrultuda Synchro 7 simülasyon programının kullandığı ivmelenme değerleri kullanılmış olup bu değerler Çizelge 5-2’de verilmektedir.

**Çizelge 5-2** Hız aralıklarına bağlı ivmelenme değerleri [25]

| Taşıt Tipi      | Hız Aralıkları |       |       |       |       |       |       |       |       |        |
|-----------------|----------------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
|                 | 0-10           | 10-20 | 20-30 | 30-40 | 40-50 | 50-60 | 60-70 | 70-80 | 80-90 | 90-100 |
| <b>Otomobil</b> | 2,44           | 2,74  | 1,83  | 1,52  | 1,52  | 1,52  | 1,22  | 0,91  | 0,61  | 0,61   |
| <b>Minibüs</b>  | 1,43           | 1,63  | 1,51  | 1,06  | 0,94  | 0,80  | 0,65  | 0,52  | 0,39  | 0,26   |
| <b>Kamyon</b>   | 0,86           | 0,74  | 0,66  | 0,62  | 0,53  | 0,43  | 0,34  | 0,25  | 0,17  | 0,09   |
| <b>Otobüs</b>   | 2,28           | 1,62  | 0,97  | 0,81  | 0,70  | 0,50  | 0,43  | 0,29  | 0,23  | 0,15   |
| <b>Tır</b>      | 0,75           | 0,65  | 0,43  | 0,34  | 0,26  | 0,19  | 0,13  | 0,09  | 0,04  | 0,00   |

Araçların aynı şeritte ilerlemelerinde güvenlik kriteri temel unsurdur ve birbirleri ile çarpışmamaları gerekmektedir. Bu nedenle araçların yavaşlama ve hızlanma durumlarındaki hızlarının belirlenmesinde ivmeleri ile birlikte güvenlik mesafesi de önem arz etmektedir. Bir aracın belli bir hızdan duruncaya kadarki ya da belli bir hıza ulaşincaya kadarki ivmeye bağlı olarak kat ettiği yol  $s(m)$  ile ifade edilmektedir.  $S$ ’in hıza ve ivmeye bağlı ifadesi Eşitlik 5.2’de yer almaktadır.

$$s = \frac{v^2}{2a} \quad (5.2)$$

Burada  $v$  (m/sn) aracın hızı,  $a$  (m/sn<sup>2</sup>) aracın ivmesidir.

Aynı şeritte ilerleyen araçların arasında güvenlik mesafesinin belirlenmesi için ampirik bir formül kullanılmıştır. Bu formül araçların arasında mesafenin yani arkadaki taşıtın ön tamponu ile öndeki taşıtın arka tamponu arasında en az 1,5 m olmasına göre ayarlanmış olup, Eşitlik 5.3’de verilmektedir.

$$s_{güv} = \frac{v(\frac{km}{saat})}{2,5} + 1,5 \quad (5.3)$$

Bu modül ile istenen aralıktaki araç sayısı ve kuyruk uzunluğu ölçülebilmektedir. Giriş ve çıkış detektörlerine göre ölçüm yapılmakta ve iki detektör arasında kalan taşıtlar istenen aralıktaki taşıt sayılarını vermektedir. Çıkış detektörü dur çizgisinde konumlandırılmakta ve giriş detektörü ise ne kadar mesafede ölçüm yapılmak isteniyorsa ona göre dur çizgisinin gerisine konumlandırılmaktadır. Bu çalışmada taşıt sayılarının ve kuyruk uzunluklarının ölçüm aralığı 175 m olarak belirlenmiştir. Giriş detektörü dur çizgisinden 175 m geriye konumlandırılmış bu aralıktaki ölçümlere göre kontrol sistemleri oluşturulmuştur. 175 m ölçüm aralığı Doğanın [25] çalışması referans alınarak belirlenmiştir. Kuyruk uzunluğunun ölçülmesinde ise bazı kabuller yapılmıştır ve CORSIM simülasyon programı [69] referans alınmıştır. Kuyruk uzunluğu belirlenirken sadece duran araçlar ( $v=0$  km/saat) değil aynı zamanda hızı ( $v<10$  km/saat) olan araçlar da dahil edilmiştir. Böylece belli bir hızın altında kuyruk içerisinde hareket eden araçlar da ihmal edilmemiştir. Kuyruk uzunluklarının ölçülmesinde taşıtların durma esnasındaki ara mesafeleri kuyruk uzunluğuna dahil edilmemiştir. Bu mesafeler taşıt cinslerine göre 0,1 m ile 0,5 m arasında değişkenlik göstermekte ve kuyruk uzunluğunun ölçülmesinde ciddi bir farklılık oluşturmamakta olup, %1'in altında bir hata oranına neden olmaktadır. Bu nedenle kuyruk uzunlukları, kuyrukta bulunan taşıt sayılarına ve taşıt uzunluklarına göre belirlenmiştir. Simülasyon sisteminde kuyruk uzunlukları şerit bazlı olarak ölçüldüğü için her bir taşıtın kuyrukta olup olmama durumu 0 ve 1 değişkenleri ile kontrol edilmektedir. Özellikle sola dönüş cebi olan kavşak kollarında, sol cebe girişten önce kuyrukta bulunan araçların izlenmesi hassas bir kuyruk uzunluğu ölçümü için gerekmektedir. Böylece kuyruk içerisinde olan aracın şerit değiştirmesi ve kuyruktan çıkması söz konusu olduğunda kolaylıkla izlenebilmekte ve doğru şekilde şeritteki kuyruk uzunlukları ölçülebilmektedir.

### 5.1.3. Sinyal Denetim Modülü

Bu modül simülasyondaki sinyalizasyonu kontrol eden kısımdır. Kontrol yaklaşımına göre belirlenen faz planına ve kırmızı, sarı ve yeşil sürelerine göre sinyalizasyonu

çalıştırmaktadır. Bu çalışma kapsamında sabit zamanlı trafik kontrol sistemi ve farklı yaklaşımlara göre oluşturulan adaptif trafik kontrol sistemleri kullanıldığından dolayı sinyal denetim modülü bu sistemlerle birlikte çalışabilecek şekilde oluşturulmuştur. Kontrol sistemleri hakkındaki detaylı bilgi bir sonraki bölüm olan 6. bölümde Trafik Kontrol Sistemleri başlığı altında detaylı olarak verilecektir.

## **5.2. KU-Trsim Programı Kullanıcı Ara yüzü**

Simülasyon programında verilerin kolaylıkla sisteme girilebilmesi ve çıktı değerlerinin izlenebilmesi için GUI üzerinden ara yüzler oluşturulmuştur. Simülasyon üzerinde farklı kontrol sistemlerinin çalışabilmesi mümkün olduğu için farklı girdi ve çıktı değişkenlerine ihtiyaç duyulmaktadır. Bu nedenle ara yüzlerin oluşturulmasında bu durum göz önüne alınmış ve sabit zamanlı kontrol sistemi ve adaptif kontrol sistemleri için farklı giriş çıkış ara yüzleri oluşturulmuştur.

### **5.2.1. Sabit Zamanlı Kontrol Sistemi Ara Yüzleri**

Veri giriş ara yüzü üzerinden kavşağın geometrik özellikleri, faz ve sinyal süresi değişkenleri girilmektedir. Kaç kollu bir kavşak tasarlanacağı, her bir kolda kaç şerit olacağı, her bir şeride ait trafik hacmi, taşıt dağılımı, dönüş oranları, detektör uzaklığı gibi parametreler kavşak değişkenleri kısmından girilebilmektedir. Faz ve sinyal süresi değişkenleri kısmında ise faz düzeni, devre süresi, simülasyon süresi gibi farklı değerler sisteme girilebilmektedir. Ayrıca kavşak geometrisinin görseli de ara yüz üzerinden gösterilmektedir. Hazırlanan giriş ara yüzü Şekil 5-1’de gösterilmiştir. Aynı zamanda, giriş ara yüzü ile verilerin kaydedilmesi ve simülasyonun çalıştırılması da sağlanmaktadır. Çıktı ara yüzünde ise simülasyonu yapılan kavşağa ait araç sayıları ve kuyruk uzunlukları ile kollardaki değişimler gösterilmekte olup, çıkış ara yüzü Şekil 5-2’de gösterildiği gibidir. Çıktı ara yüzündeki süreler kısmında ise devre süreleri gösterilmektedir.

**Kayıtlık Değişkenleri**

Kol Sayısı Giriniz

Kol Sayısını Kaydet / Şerit Sayılarını Gir

Şerit Sayısı Giriniz

K1 Doğu  ☒ 1. Kolda Sol Cep Varmı ?

K2 Batı  ☐ 2. Kolda Sol Cep Varmı ?

K3 Kuzey  ☐ 3. Kolda Sol Cep Varmı ?

K4 Güney  ☐ 4. Kolda Sol Cep Varmı ?

Kayıtlık Şeklini Göster ve Parametre Tablosunu Aç

**Parametre Tablosu**

| Trafik Akım | Ortomobil Yüzdesi (%) | Minibüs Yüzdesi (%) | Kamyon Yüzdesi (%) | Otobüs Yüzdesi (%) | Tır Yüzdesi (%) | Sol Cep Logic | Sol Cep Uzunluk (m) | Sol Cebeye Dönüş Oranı (%) | Sola Dönüş Logic | Sola Dönüş Oranı (%) | Sağa Dönüş Logic | Sağa Dönüş Oranı (%) | Detektör Uzunluğu (m) | Serbest Hız Değeri (km/saat) | Kol Uzunluğu (m) |
|-------------|-----------------------|---------------------|--------------------|--------------------|-----------------|---------------|---------------------|----------------------------|------------------|----------------------|------------------|----------------------|-----------------------|------------------------------|------------------|
| K1-S1 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K1-S2 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K2-S1 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K2-S2 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K3-S1 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K3-S2 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K4-S1 Düz > | 0                     | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                          | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |

**Faz ve Sinyal Süresi Değişkenleri**

Faz Sayısı  Faz Düzeni Aç

Simulasyon Süresi (s)

Besleme Süresi (s)

Devre süresi (s)

Sani Süresi (s)

Her Yöne Kırmızı Süresi (s)

Seed Numarası

**Faz Tablosu**

|       | F-1                                 | F-2                                 | F-3                                 | F-4                                 | Kontrol Sürelini Gir | Faz 1 Yeşil Süre     | Faz 2 Yeşil Süre     | Faz 3 Yeşil Süre     | Faz 4 Yeşil Süre     |
|-------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| K1-S1 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K1-S2 | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K2-S1 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K2-S2 | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K3-S1 | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K3-S2 | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| K4-S1 | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |                      | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Verileri Kaydet

SİMULASYONU BAŞLAT

Kuzey

Güney

Doğu

Batı

Şekil 5-1 Sabit zamanlı kontrol sistemi giriş ara yüzü



### 5.2.2. Adaptif Kontrol Sistemi Ara Yüzleri

Veri giriş ara yüzü üzerinden kavşağın geometrik özellikleri ve simülasyon değişkenleri girilmektedir. Kaç kollu bir kavşak tasarlanacağı, her bir kolda kaç şerit olacağı, her bir şeride ait trafik hacmi, taşıt dağılımı, dönüş oranları, detektör uzaklığı gibi parametrelerin kavşak değişkenleri kısmından sisteme girilmesi mümkündür. Simülasyon süresi, besleme süresi, çekirdek numarası gibi simülasyon değişkenleri de girilebilmektedir. Faz planı ve sinyal sürelerinin girilmesine adaptif sistemlerde gerek duyulmamaktadır. Adaptif sistemlerin çalışma prensibinde faz ve süre optimizasyonu bulunduğu için kontrol sistemi tarafından faz planı ve sinyal süreleri belirlenmektedir. Kontrol sistemleri ile ilgili detaylı bilgi 6. bölümde verilmiştir. Sistemde faz ve süre optimizasyonu ile birlikte faz sırası değişimi de yapıldığı için çıktı ara yüzünde faz değişimleri ve fazlardaki yeşil sürelerinin de gösterimi sağlanmıştır. Ayrıca kavşak geometrisinin görseli de ara yüz üzerinden gösterilmektedir. Adaptif bir sisteme ait giriş ara yüzü Şekil 5-3’de gösterilmiştir. Giriş ara yüzü ile verilerin kaydedilmesi ve simülasyonun çalıştırılması da sağlanmaktadır. Şekil 5-4’de gösterilen çıkış ara yüzü ile devre süreleri, araç sayıları, kuyruk uzunlukları, faz değişimleri ve kollardaki değişimin grafiksel gösterimi ve gecikme değerleri aynı anda görülebilmektedir. Çıktı ara yüzünde ki süreler kısmında ise devre süreleri gösterilmektedir.

**Kayıtlık Değişkenleri**

Kol Sayısı Girmiz

Kol Sayısını Kaydet / Şartı Saylamı Gır

Şartı Sayısı Girmiz

K1 Doğru  ☐ 1. Kolda Sol Cep Var mı ?

K2 Batı

K3 Kuzey

**Parametre Tablosu**

| Tür         | Trafik Akım | Otomobil Yüzdesi (%) | Minibus Yüzdesi (%) | Kamyon Yüzdesi (%) | Otobüs Yüzdesi (%) | Tir Yüzdesi (%) | Sol Cep Logic | Sol Cep Uzunluk (m) | Sol Cebde Uzunluk (m) | Sola Dönüş Logic | Sola Dönüş Oranı (%) | Saga Dönüş Logic | Saga Dönüş Oranı (%) | Detektor Uzunluğu (m) | Serbest Hız Degeri (km/saat) | Kol Uzunluğu (m) |
|-------------|-------------|----------------------|---------------------|--------------------|--------------------|-----------------|---------------|---------------------|-----------------------|------------------|----------------------|------------------|----------------------|-----------------------|------------------------------|------------------|
| K1-S1 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K1-S2 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K2-S1 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K2-S2 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K3-S1 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |
| K3-S2 Düz ✓ | 0           | 0                    | 0                   | 0                  | 0                  | 0               | 0             | 0                   | 0                     | 0                | 0                    | 0                | 0                    | 0                     | 0                            | 0                |

**Kayıtlık Şeklini Göster ve Parametre Tablosunu Aç**

**Simülasyon Değişkenleri**

Simülasyon Süresi (s)

Başlama Süresi (s)

Batı

Sarı Süresi (s)

Her Yöne Kırmızı Süresi (s)

Seed Numarası

Simülasyon Süresi (s)

Başlama Süresi (s)

Batı

Sarı Süresi (s)

Her Yöne Kırmızı Süresi (s)

Seed Numarası

Verileri Kaydet

SİMÜLASYONU BAŞLAT

Şekil 5-3 Adaptif kontrol sistemi giriş ara yüzü

Araç Sayıları

| Sıra | Kol 1 | Kol 2 | Kol 3 | Kol 4 |
|------|-------|-------|-------|-------|
| 1    | 182   | 20    | 19    | 24    |
| 2    | 362   | 30    | 27    | 37    |
| 3    | 540   | 30    | 0     | 11    |
| 4    | 710   | 37    | 0     | 10    |
| 5    | 878   | 52    | 17    | 20    |
| 6    | 1046  | 51    | 28    | 40    |
| 7    | 1208  | 63    | 25    | 50    |
| 8    | 1372  | 82    | 17    | 64    |
| 9    | 1540  | 54    | 12    | 64    |
| 10   | 1649  | 66    | 6     | 63    |
| 11   | 1700  | 64    | 2     | 63    |

Kuyruk Uzunluğu

| Kol 1 | Kol 2 | Kol 3 | Kol 4 |
|-------|-------|-------|-------|
| 1     | 60.5  | 13.5  | 4.5   |
| 2     | 86    | 0     | 65.5  |
| 3     | 128   | 0     | 37.5  |
| 4     | 168.5 | 0     | 57.5  |
| 5     | 187.5 | 69    | 77    |
| 6     | 185   | 115.5 | 169.5 |
| 7     | 172   | 121.5 | 162   |
| 8     | 167.5 | 96    | 165   |
| 9     | 165   | 62.5  | 165   |
| 10    | 165   | 13.5  | 162   |
| 11    | 164.5 | 0     | 162   |

Faz Değişimi

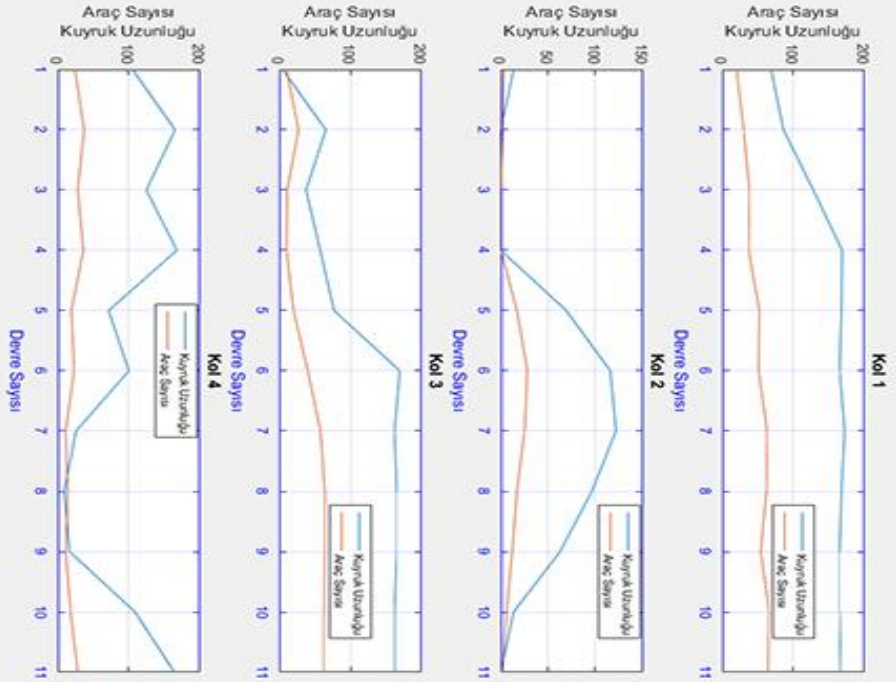
| Sıra | Faz Sırası | 1.faz | 2.faz | 3.faz | 4.faz |
|------|------------|-------|-------|-------|-------|
| 1    | 182        | 1     | 2     | 3     | 4     |
| 2    | 182        | 2     | 3     | 4     | 1     |
| 3    | 182        | 3     | 4     | 1     | 2     |
| 4    | 182        | 4     | 1     | 2     | 3     |
| 5    | 182        | 1     | 2     | 3     | 4     |
| 6    | 182        | 2     | 3     | 4     | 1     |
| 7    | 182        | 3     | 4     | 1     | 2     |
| 8    | 182        | 4     | 1     | 2     | 3     |
| 9    | 182        | 1     | 2     | 3     | 4     |
| 10   | 182        | 2     | 3     | 4     | 1     |
| 11   | 182        | 3     | 4     | 1     | 2     |
| 12   | 170        | 4     | 1     | 2     | 3     |
| 13   | 170        | 1     | 2     | 3     | 4     |
| 14   | 170        | 2     | 3     | 4     | 1     |
| 15   | 170        | 3     | 4     | 1     | 2     |
| 16   | 170        | 4     | 1     | 2     | 3     |
| 17   | 168        | 1     | 2     | 3     | 4     |
| 18   | 168        | 2     | 3     | 4     | 1     |
| 19   | 168        | 3     | 4     | 1     | 2     |
| 20   | 168        | 4     | 1     | 2     | 3     |
| 21   | 168        | 1     | 2     | 3     | 4     |
| 22   | 168        | 2     | 3     | 4     | 1     |
| 23   | 168        | 3     | 4     | 1     | 2     |
| 24   | 168        | 4     | 1     | 2     | 3     |
| 25   | 162        | 1     | 2     | 3     | 4     |
| 26   | 162        | 2     | 3     | 4     | 1     |
| 27   | 162        | 3     | 4     | 1     | 2     |
| 28   | 162        | 4     | 1     | 2     | 3     |
| 29   | 164        | 1     | 2     | 3     | 4     |
| 30   | 164        | 2     | 3     | 4     | 1     |
| 31   | 164        | 3     | 4     | 1     | 2     |
| 32   | 164        | 4     | 1     | 2     | 3     |
| 33   | 166        | 1     | 2     | 3     | 4     |
| 34   | 168        | 2     | 3     | 4     | 1     |
| 35   | 168        | 3     | 4     | 1     | 2     |
| 36   | 168        | 4     | 1     | 2     | 3     |
| 37   | 169        | 1     | 2     | 3     | 4     |
| 38   | 169        | 2     | 3     | 4     | 1     |
| 39   | 169        | 3     | 4     | 1     | 2     |
| 40   | 169        | 4     | 1     | 2     | 3     |

Ortalama Geçiş

163.159

Dönme Geçirimi

126.076



Şekil 5-4 Adaptif kontrol sistemi çıkış ara yüzü

### 5.3. KU-Trsim Programının Performansı

Mikroskobik trafik simülasyonu KU-Trsim'in performansını ortaya koymak için, bilinen simülasyon programlarından VISSIM ve CORSIM simülasyon programları ile karşılaştırılması yapılmıştır. 3 kollu ve 4 kollu bir kavşak için yapılan simülasyonlarda 200, 400 ve 600 taşıt/saat olan 3 farklı trafik senaryosu göz önüne alınmıştır. Kavşaklardaki kol sayısının yanı sıra şerit sayıları da farklı belirlenmiş ve şerit sayılarındaki değişimde simülasyon programlarının davranışı izlenmiştir. Trafik senaryolarında %90 otomobil ve %10 diğer taşıtlar kullanılmıştır. Diğer taşıtlar kendi içinde % 5 minibüs, % 2 otobüs, % 2 kamyon ve % 1 tır olarak belirlenmiştir. Böylece farklı araç tiplerinin de etkisi göz ardı edilmemiş, gerçek hayatta olduğu gibi simülasyon ortamına dahil edilmiştir. Sinyalizasyonda gecikmeye etki eden faktörlerin birisi de araçların dönüşteki manevra hareketleridir. Bu nedenle sağ ve sola dönüşlerden kaynaklı gecikmelerin de etkisinin görülebilmesi için kollardan gelen araçların %25'nin sola ve %15'nin sağ dönüş yapacakları şekilde hareket planı oluşturulmuştur. Tek bir çekirdek numarasına göre simülasyonları yapmak yanıltıcı sonuçlar ortaya çıkartacağı için 10 farklı çekirdek numarası kullanılarak simülasyonlar yapılmış ve 10 gecikme değerinin ortalaması alınarak her bir trafik durumu için gecikme değerleri elde edilmiştir. Ayrıca farklı periyotlarda sistem performansını görebilmek için 15 dakika ve 30 dakika olarak 2 farklı periyot süresi kullanılmıştır. 3 kollu kavşakta tek şeritli olarak kurulan kavşak geometrisinde 1800 taşıt/saat doymun akım değeri olarak kabul edilmiş ve sabit zamanlı kontrol ile sinyalizasyonu kurulmuştur. 4 kollu kavşak ise 2 şeritli olarak kurulmuş olup 3600 taşıt/saat doymun akım değeri kullanılmıştır. Sinyalizasyonda kullanılan devre süreleri en yüksek trafik senaryosu olan 600 taşıt/saat'e göre belirlenmiş ve diğer trafik durumlarında kullanılmıştır. 3 kollu kavşak için 51 saniye ve 4 kollu kavşak için 52 saniye devre süreleri bu simülasyonlarda kullanılmıştır. Simülasyonlara ait girdi parametreleri Çizelge 5-3'de verilmiştir. Çizelge 5-3'de  $T$  (dakika) değişkeni simülasyon periyodunu,  $C$  (saniye) değişkeni devre süresini,  $s$  (araç/saat) değişkeni doymun akımı,  $y$  (saniye) değişkeni yeşil süreleri ve  $v$  (araç/saat) değişkeni trafik hacmini ifade etmektedir.

**Çizelge 5-3** Simülasyon girdi parametreleri

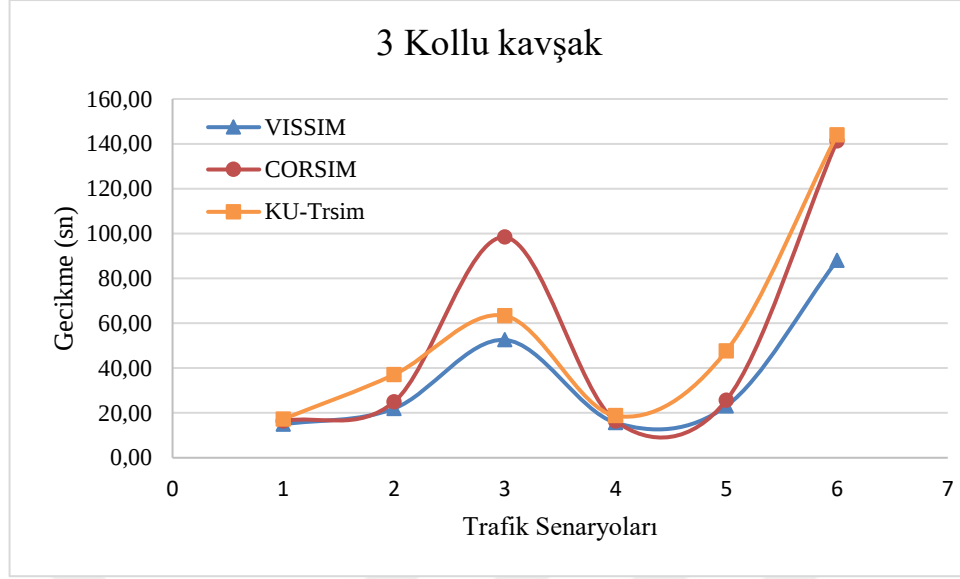
| 3 KOLLU KAVŞAK |          |          |                       |                       |                       |                    |                       |                       |                       |  |
|----------------|----------|----------|-----------------------|-----------------------|-----------------------|--------------------|-----------------------|-----------------------|-----------------------|--|
| <i>T</i>       | <i>C</i> | <i>S</i> | <i>y</i> <sub>1</sub> | <i>y</i> <sub>2</sub> | <i>y</i> <sub>3</sub> | Hep kırmızı + Sarı | <i>V</i> <sub>1</sub> | <i>V</i> <sub>2</sub> | <i>V</i> <sub>3</sub> |  |
| 15             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 200                   | 200                   | 200                   |  |
| 15             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 400                   | 400                   | 400                   |  |
| 15             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 600                   | 600                   | 600                   |  |
| 30             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 200                   | 200                   | 200                   |  |
| 30             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 400                   | 400                   | 400                   |  |
| 30             | 51       | 1800     | 14                    | 14                    | 14                    | 3                  | 600                   | 600                   | 600                   |  |

| 4 KOLLU KAVŞAK |          |          |                       |                       |                       |                       |                    |                       |                       |                       |                       |
|----------------|----------|----------|-----------------------|-----------------------|-----------------------|-----------------------|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <i>T</i>       | <i>C</i> | <i>S</i> | <i>y</i> <sub>1</sub> | <i>y</i> <sub>2</sub> | <i>y</i> <sub>3</sub> | <i>y</i> <sub>4</sub> | Hep kırmızı + Sarı | <i>V</i> <sub>1</sub> | <i>V</i> <sub>2</sub> | <i>V</i> <sub>3</sub> | <i>V</i> <sub>4</sub> |
| 15             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 200                   | 200                   | 200                   | 200                   |
| 15             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 400                   | 400                   | 400                   | 400                   |
| 15             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 600                   | 600                   | 600                   | 600                   |
| 30             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 200                   | 200                   | 200                   | 200                   |
| 30             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 400                   | 400                   | 400                   | 400                   |
| 30             | 52       | 3600     | 10                    | 10                    | 10                    | 10                    | 3                  | 600                   | 600                   | 600                   | 600                   |

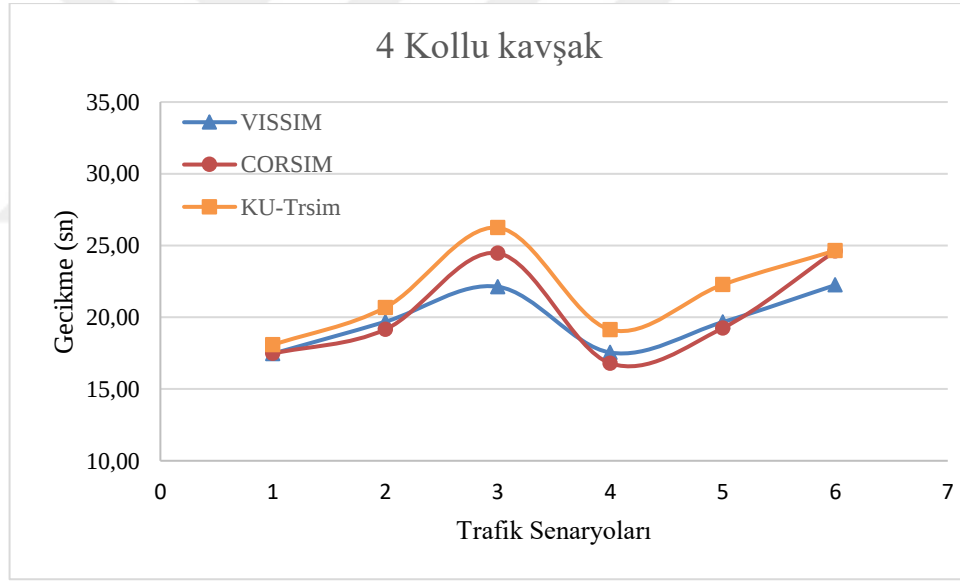
Ortalama Mutlak Hatalar (OMH) ve Ortalama Karesel Hataların Karekökü (OKHK) performans karşılaştırmasında kullanılan istatistiki göstergelerdir. KU-Trsim mikroskobik simülasyon programının diğer iki simülasyon programı ile olan OMH ve OKHK istatistiki değerleri Çizelge 5-4’de verilmektedir. Yapılan simülasyonlar sonucunda, 3 ve 4 kollu kavşaklara ait simülasyon programların performansları Şekil 5-5 ve 5-6’da gösterilmiştir.

**Çizelge 5-4** Simülasyon istatistiki göstergeleri

| Kavşak  | VISSIM |       | CORSIM |       |
|---------|--------|-------|--------|-------|
|         | OMH    | OKHK  | OMH    | OKHK  |
| 3 Kollu | 18,64  | 26,13 | 12,50  | 17,66 |
| 4 Kollu | 2,05   | 2,36  | 1,55   | 1,85  |



**Şekil 5-5** 3 Kollu kavşak için simülasyon karşılaştırmaları



**Şekil 5-6** 4 Kollu kavşak için simülasyon karşılaştırmaları

Simülasyon programlarının karşılaştırmalarına göre 3 kollu ve 4 kollu kavşak geometrilerinde her 3 simülasyonda aynı trendi yakalayarak birbirine yakın gecikme değerleri tahmin etmiştir. Simülasyon programlarının farklı gecikme değerleri vermesi ise her programdaki araç karakteristiklerinden ve çekirdek numaralarındaki sürücü davranışlarının farklı olmasından kaynaklanmaktadır. İstatistiki sonuçlara göre de KU-Trsim programı VISSIM ve CORSIM simülasyonuna yakın bir performans

göstermektedir. 3 kollu ve 4 kollu kavşak geometrilerinde VISSIM ve CORSIM programları ile KU-Trsim programının OMH ve OKHK değerleri birbirine yakındır. Ayrıca OMH ve OKHK değerlerine göre, KU-Trsim programı CORSIM programına daha yakın sonuçlar ortaya koymuş ve hata oranları düşük çıkmıştır. 3 kollu kavşak geometrisinde ise OHM ve OKHK değerleri VISSIM ve CORSIM programları için 4 kollu kavşağa göre sırasıyla yaklaşık olarak 8 kat ve 11 kat fazladır. 3 kollu kavşak türünde simülasyonların gecikme değerleri arasında farkın ve istatistiki gösterge değerlerinin fazla olmasında şerit sayısı en önemli etkidir. Tek şeritli olarak oluşturulan 3 kollu kavşakta araçların dönüş hareketleri, kuyruk oluşması, simülasyon programların tek şerit üzerindeki hareketleri modelleyebilme kabiliyetleri gibi nedenlere bağlı olarak gecikme değerlerinde farklılık oluşmakta ve bu durum OHM ile OKHK değerlerine yansımaktadır. Böylece geliştirilen simülasyon programının uygun ve kullanılabilir olduğu da ortaya konulmuştur.

## 6. TRAFİK KONTROL SİSTEMLERİ

Tez kapsamında farklı trafik kontrol sistemleri geliştirilmiş ve mevcut trafik kontrol sistemleri ile karşılaştırılmıştır. Sabit zamanlı kontrol sistemi ve Doğan [29] tarafından geliştirilen optimize edilmiş BM kontrol sistemi, geliştirilen sistemlerin performans karşılaştırmasında kullanılacak trafik kontrol sistemleridir. Bu çalışmada ilk olarak DGA kullanılarak optimum devre süresi arama algoritması geliştirilmiş ve sabit zamanlı kontrol sisteminin iyileştirilmesi sağlanmıştır. Daha sonra ÇTA yaklaşımıyla faz optimizasyonu yapılarak yeni bir kontrol sistemi geliştirilmiştir. Son olarak, ÇTA yaklaşımı ile faz optimizasyonu ve KAA algoritmasıyla optimize edilmiş Tip-2 BM yaklaşımı ile süre optimizasyonunun birlikte çalıştırıldığı hibrid bir trafik kontrol sistemi geliştirilmiştir. Tez kapsamında geliştirilen kontrol sistemlerinin detayları aşağıdaki alt başlıklarda verilmiştir.

### 6.1. DGA Optimum Devre Süresi Arama ile Trafik Kontrol Sistemi

Optimum Devre Süresinin (ODS) en düşük gecikmeyi verecek şekilde aranması ile sabit zamanlı kontrolün performansının iyileştirilmesi amaçlanmaktadır. Amaç fonksiyonu olarak faz sayısına göre toplam gecikmenin minimum olması hedeflenmiş ve bunun için HCM 2000 [41] gecikme bağıntısı kullanılmıştır. Burada DGA yaklaşımının, kısa sürede optimum sonuca ulaşabilmesi ve basit şekilde kodlanabilmesi gibi avantajlarından yararlanarak bir arama algoritması geliştirilmiştir. Kullanılan optimizasyon yönteminde kavşağın kaç fazlı olarak çalışacağına göre yeşil süre değişkenleri belirlenmekte ve kayıp zamanların ilave edilmesi ile optimum devre süresi tayin edilebilmektedir. Amaç fonksiyonunun oluşturulmasında bazı kıstaslar göz önüne alınmıştır. Her bir kolun şeridindeki kritik trafik akımı, yani kola ait şeritlerdeki maksimum akım girdi parametresi olarak tayin edilmiştir. Eğer bir fazda, birden fazla kolu aynı anda çalıştırılıyor ise o durumda kollar içerisindeki en yüksek şerit akımı girdi parametresi olarak belirlenmektedir. Şerit bazlı amaç fonksiyonu oluşturulduğundan dolayı doymuş akım değeri 1800 (taşıt/saat/şerit) olarak kullanılmış olup minimum yeşil ışık süresi 8 sn. olarak ayarlanmıştır. Doymuşluk derecesi ise trafik senaryosuna göre değişiklik göstermekte olup, 1,4'ün üstüne çıkmayacak şekilde sınırlandırılmıştır.

Diferansiyel Gelişim Algoritması Optimum Devre Süresi (DGA-ODS) sisteminin amaç fonksiyonu Eşitlik 6.1’de verilmektedir.

$$\text{Min } F(x) = \sum_{n=1}^m \frac{d_n}{q_n} \quad (6.1)$$

Burada,  $d_n$  fazdaki gecikme değeri,  $q_n$  fazdaki kritik trafik akımı ve  $m$  kavşaktaki toplam faz sayısıdır.

Bir optimizasyon problemi, amaç fonksiyonu, karar değişkenleri ve kısıtlar setinden oluşmaktadır. DGA-ODS sistemi kapsamında, oluşturulan amaç fonksiyonu, karar değişkenleri ve kısıtlar seti Çizelge 6-1’ de sunulmaktadır.

**Çizelge 6-1** DGA-ODS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti

|                                                          |                                                          |                                                             |
|----------------------------------------------------------|----------------------------------------------------------|-------------------------------------------------------------|
| <b>Amaç Fonksiyonu</b><br>( $m$ , kavşaktaki faz sayısı) | $f = \min \left\{ \sum_{n=1}^m \frac{d_n}{q_n} \right\}$ |                                                             |
| <b>Karar Değişkenleri</b>                                | $g_i$                                                    | $g$ faza ait yeşil ışık süresi<br>$i$ kavşaktaki faz sayısı |
| <b>Kısıtlar</b>                                          | $8 \leq g_i$                                             |                                                             |
|                                                          | $0 \leq \frac{q_i \times C}{s \times g_i} \leq 1.4$      |                                                             |

Gecikme sürelerinin belirlenmesinde farklı zaman kayıpları göz önüne alınmakta olup bunlar; yeşil sürenin bitmesi ve durma komutu ile kendisine yeşil süre başlayana kadar beklerken geçen süre, bir aracın önünde bulunan araçla arasındaki güvenli bir mesafenin korunması amacıyla hızında yaptığı ayarlama kaybedtiği süre, kavşakta ağır hareket eden ve kavşağa park ederek geçiş zorluğu yaratan araçlar nedeniyle kaybedilen sürelerdir. Amaç fonksiyonunda kullanılan HCM 2000 gecikme bağıntısı Eşitlik 6-2’de verilmektedir.

$$d = d_1 * (PF) + d_2 + d_3 \quad (6.2)$$

Burada,  $d$  taşıt başına gecikme (sn./taşıt),  $d_1$  üniform gecikme (sn./taşıt),  $PF$  üniform gecikme ayarlama faktörü,  $d_2$  eklenik gecikme (sn./taşıt),  $d_3$  başlangıç kuyruk gecikmesi (sn./taşıt)'dir.

HCM 2000 gecikme bağıntısındaki üniform gecikme ve eklenik gecikmenin matematiksel ifadeleri ise Eşitlik 6.3 ve 6.4'te gösterildiği gibidir.

$$d_1 = \frac{0,5 * c * \left(1 - \frac{g}{c}\right)^2}{1 - \left[\min(1, X) * \frac{g}{c}\right]} \quad (6.3)$$

$$d_2 = 900 * T * \left[ (X - 1) + \sqrt{(X - 1)^2 + \frac{8 * k * I * X}{c * T}} \right] \quad (6.4)$$

Burada,  $d_2$  eklenik gecikme (sn./taşıt),  $T$  analiz periyodu (sn.),  $k$  gecikme faktörü,  $I$  ayarlama faktörü,  $c$  kapasite (taşıt/saat)'dir.

DGA-ODS sisteminin akış diyagramı Şekil 6-1'de verilmiştir.



Şekil 6-1 DGA-ODS akış diyagramı

## 6.2. Çiçek Tozlaşma Algoritması ile Trafik Kontrol Sistemi

Önerilen yeni Trafik Kontrol Sisteminde (TKS) ÇTA yaklaşımı kullanılarak faz optimizasyonu yapan bir modül geliştirilmiştir. Bu modülde kavşağın kaç fazda çalışacağı ve hangi şeritlerin hangi fazlarda geçiş hakkı alacağına karar verilmektedir. Kavşak geometrisi ve trafik hacimlerine göre faz planı belirlenirken HCM gecikme bağıntısına göre hesaplanan gecikmenin minimize edilmesi amaçlanmış ve bu bağıntı amaç fonksiyonu olarak kullanılmıştır. Her bir şeritteki gecikme değeri hesaplanarak fazlarda hareket eden şeritlere göre faz başına düşen gecikmeler elde edilebilmektedir. Böylece bütün faz kombinasyonları arasından en düşük gecikmeyi veren faz planının belirlenebilmesi sağlanmaktadır. Bir kavşağın uygun faz planı ile işletiminde gecikmenin minimize edilmesini sağlamak kadar trafik güvenliğinin de maksimize edilmesini de sağlamak gereklidir. Bu modülde güvenlik kriteri de göz önüne alındığı için faz planı belirlenirken trafik akımlarının kesişmesine izin verilmezken sadece sola ve sağa dönüşlerde katımlara izin verilmektedir. Özellikle sola dönen araçların ayrı bir fazda çalışıp çalışmayacağı hem gecikme hem de trafik güvenliği açısından önem arz etmektedir. Bu durumda modül, güvenlik kriterine göre çalışmaktadır. Bu kriter gere göre sola dönen trafik hacmi ile sola dönen trafik hacmini kesen hacmin çarpımı 50.000 araçtan küçük ise katılmaya izin verilirken, küçük olmaması durumunda ise sola dönecek araçlar için ayrı bir faz oluşturulmaktadır [42]. Aynı zamanda belirlenen faz planına göre optimum devre süresi de elde edilebilmektedir. Optimum devre süresinin belirlenmesinde minimum ve maksimum yeşil süreleri belirlemek için kısıtlar konulmuştur. Bu kısıtlarda minimum yeşil süre 8 sn. ve maksimum yeşil süre 60 sn. olarak belirlenmiştir. Böylece devre süreleri 22 sn. ile 252 sn. arasında değişim göstermektedir. Faz optimizasyon modülünün kaç devrede bir çalışacağının belirlenmesi önemli bir husustur. Bu modülün etkinliğini arttırabilmek için sistem içerisinde daha fazla görev almasının sağlanması gerekmektedir. Bu nedenle kaç devrede bir faz kontrolünün yapılacağına karar verirken ne çok kısa tutularak bir sonraki sistemde detayı verilecek olan süre optimizasyonunun etkisiz hale getirilmesinden, ne de uzun tutup faz modülünün kullanımının azaltılmasına sebep olmaktan kaçınılmıştır. Faz optimizasyon modülünün kaç kez çalışacağının belirlenmesinde belirleyici unsur simülasyon süresidir. Simülasyon süresi olarak 900 sn. kullanılmış olup bu sürenin kullanılmasında temel neden karşılaştırma sistemi olarak kullanılan Tip-1 BM, Dogan

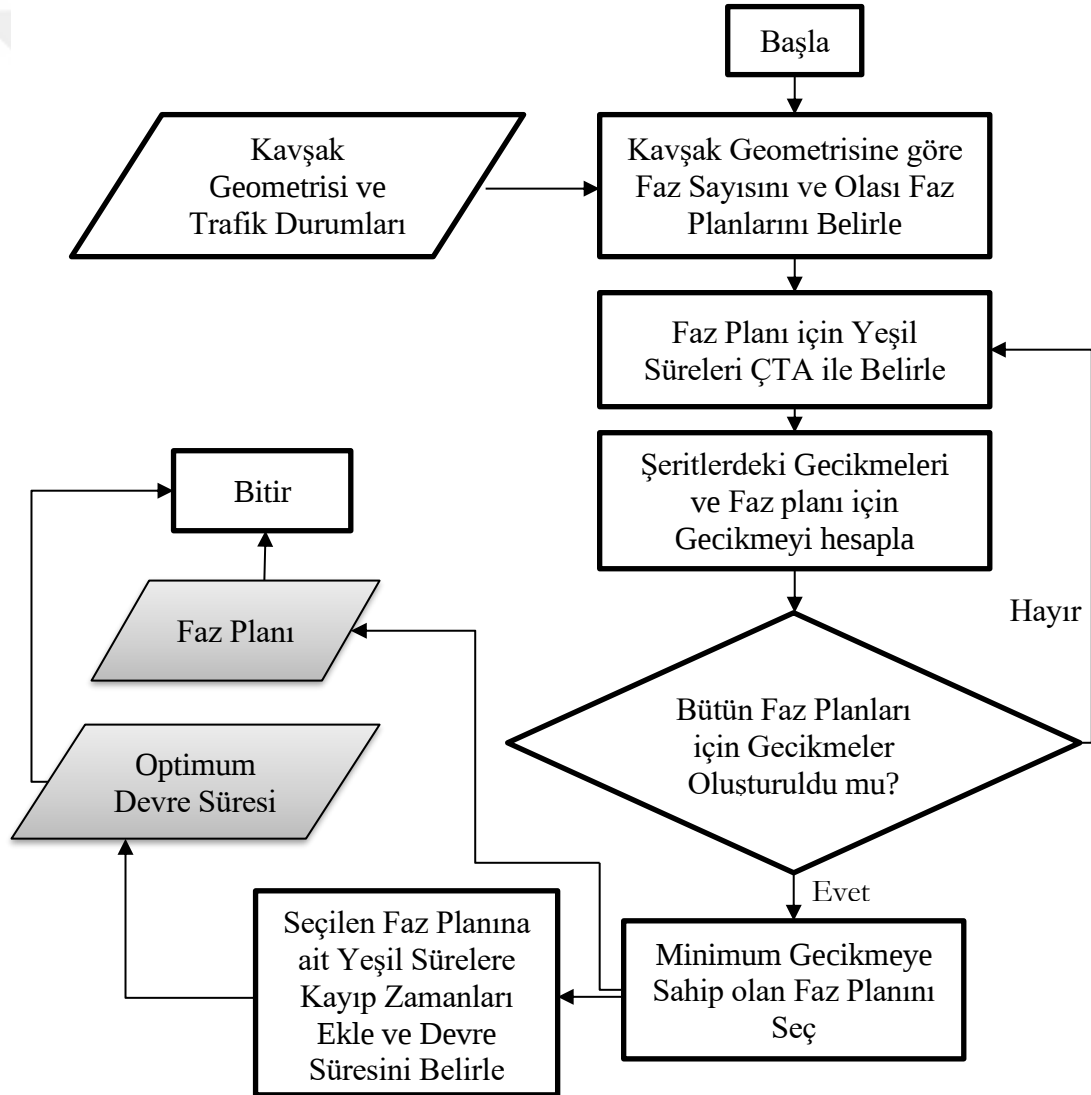
[25] tarafından geliştirilen, yönteminin çok uzun çalışma süresine ihtiyaç duymasındır. Ayrıca 900 sn. simülasyon süresinin bir kavşağın performansını göstermek için yeterli bir zaman dilimi olmasından dolayı bu süre tercih edilmiştir. Kaç devrede bir kontrol edilmesinin belirlenmesinde simülasyon süresi kadar devre süreleri de önem teşkil etmektedir. Geliştirilen sistemin düşük, orta ve yüksek trafik akımlarına cevap verebilecek şekilde çalışabilmesi gerekmektedir. Bu nedenle özellikle yüksek trafik akımlarında devre süresi fazla olacaktır. 900 sn. simülasyon süresi içerisinde faz optimizasyon modülünün en az iki kez sistemde çalışması planladığı için 140 sn. devre süresi kritik eşiştir. Böylece simülasyon süresi 420. sn. ve 840. sn.'de faz modülü devreye girebilecektir. Daha yüksek devre sürelerinin olduğu trafik durumlarında ise mecburen bir kez devreye girecektir. Bu nedenle maksimum faz modülü kontrol devre sayısı 3 olmaktadır. 3 devreden daha fazla seçilmesi durumunda ise kritik devre süresi eşiği daha düşük olacağından yüksek trafik akımlarında performans yetersizliğine sebep olacaktır. 1 ve 2 devrede bir kontrol edilme durumları ise süre optimizasyon modülü göz önüne alınarak uygun bulunmamıştır. Her devrede bir faz optimizasyon modülünün çalışması durumunda süre optimizasyon modülünün çalışmaması ve 2 devrede bir olması durumunda ise süre optimizasyon modülünün sadece bir kez devreye girmesi tercih edilmeme sebepleridir. Böylece 3 devrede bir kontrol edilmesiyle 900 sn. simülasyon süresi içerisinde hem faz hem de süre optimizasyon modüllerinin en az iki kez çalışmasına olanak sağlanmıştır. Bu aşamada bir diğer önemli husus ise 3 devre süresi bittiğinde simülasyonda kesinti olmaksızın faz optimizasyonun yapılabilmiş olmasıdır. Bu nedenle 3. devre süresinde son faz çalışırken faz optimizasyonu modülü paralel olarak çalıştırılmakta ve 3. devre süresinin bitiminde yeni faz düzeni hazır hale getirilmektedir. Böylece simülasyon ortamında herhangi bir kesinti olmaksızın optimizasyon ve simülasyonlar devam ettirilmektedir. Faz optimizasyonunda kullanılan amaç fonksiyonun ifadesi Eşitlik 6.5'te gösterilmiştir. ÇTA-TKS sistemi kapsamında, oluşturulan amaç fonksiyonu, karar değişkenleri ve kısıtlar seti Çizelge 6-2' de sunulmaktadır. Faz optimizasyon modülün akış diyagramı Şekil 6-2'de verilmiştir.

$$\text{Min } F(x) = \sum_{l=1}^k \frac{d_l}{q_l} \quad (6.5)$$

Burada,  $d_l$  şeritteki gecikme değeri,  $q_l$  şeritteki trafik akımı ve  $k$  kavşaktaki toplam şerit sayısıdır.

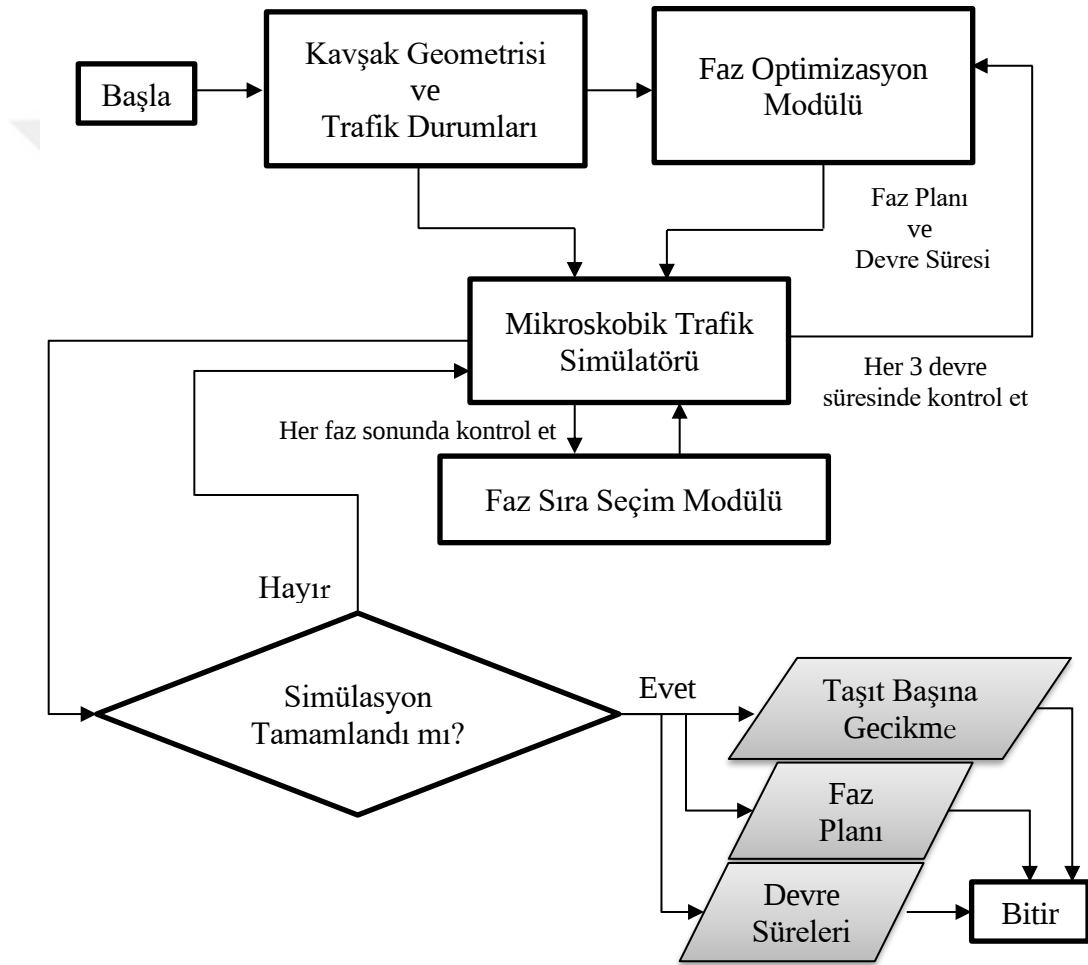
**Çizelge 6-2** ÇTA-TKS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti

|                                                       |                                                                       |                                                             |
|-------------------------------------------------------|-----------------------------------------------------------------------|-------------------------------------------------------------|
| <b>Amaç Fonksiyonu</b><br>(k kavşaktaki şerit sayısı) | $f = \min \left\{ \frac{\sum_{l=1}^k d_l}{\sum_{l=1}^k q_l} \right\}$ |                                                             |
| <b>Karar Değişkenleri</b>                             | $g_i$                                                                 | $g$ faza ait yeşil ışık süresi<br>$i$ kavşaktaki faz sayısı |
| <b>Kısıtlar</b>                                       | $8 \leq g_i \leq 60$                                                  |                                                             |
|                                                       | $0 \leq \frac{q_i \times C}{s \times g_i} \leq 1.4$                   |                                                             |



**Şekil 6-2** Faz optimizasyon modülü akış diyagramı

Faz planı ve sinyal sürelerinin optimize edilmesine ek olarak kontrol sistemi faz sırasını da değiştirebilmektedir. Burada amaç, belirlenmiş faz planında hangi fazın öncelik alacağını seçilmesidir. Bu aşamada faz sıra seçim modülü oluşturulmuş ve her bir fazın bitiminde diğer fazlara ait şeritlerdeki araç sayılarına bakılarak en yüksek araç sayısına sahip olan faza öncelik verilmiştir. Böylece sabit faz sırası yerine dinamik faz sırasının oluşturulması gecikmenin azalmasına olanak sağlayacaktır. ÇTA ile geliştirilen faz optimizasyonuna dayalı kontrol sisteminin genel yapısı Şekil 6.3’de verilmektedir.



Şekil 6-3 ÇTA-TKS sistemi genel yapısı

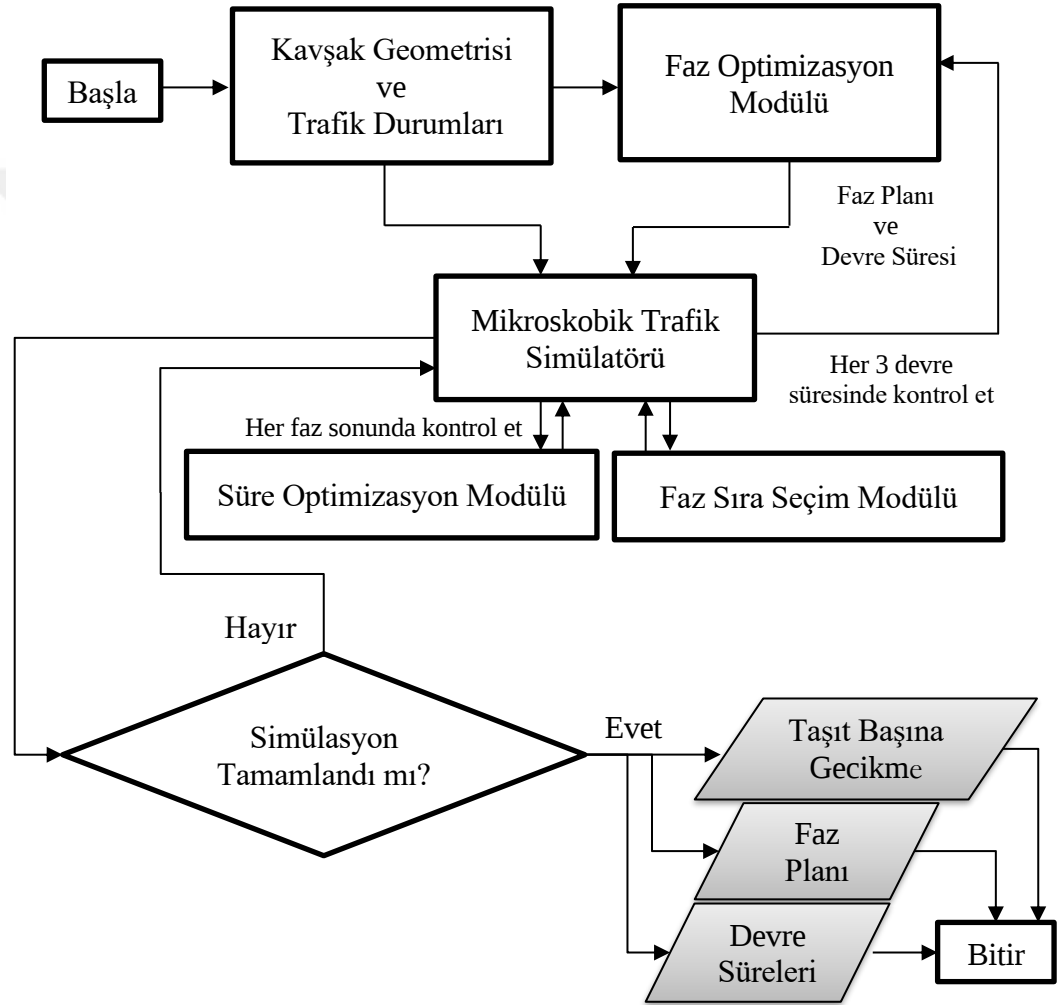
### 6.3. Hibrid Trafik Kontrol Sistemi

Önerilen Hibrid Trafik Kontrol Sisteminde (HTKS) KAA ile optimize edilen Tip-2 BM yaklaşımı ve ÇTA yaklaşımı birlikte kullanılmaktadır. Önceki bölümde detayı verilen faz optimizasyonu modülüyle birlikte çalışacak süre optimizasyonu modülü geliştirilmiştir. Faz optimizasyon modülünde elde edilen optimum devre süresinin kavşak kollarındaki trafik bilgisine göre optimize edilmesi kontrol sisteminin dinamikliğini sağlamaktadır. Süre optimizasyonu modülünde ise her bir fazın bitiminde bir sonraki faz için yeşil sürelerinin araç sayısı ve kuyruk uzunluğuna göre artırılması ya da azaltılması gerektiğine karar verilmektedir. Araç sayısı ve kuyruk uzunluğu araç gelişlerindeki rastsallığa göre değişim gösterdiği için belirsizlikler ortaya çıkmaktadır. Bu noktada BM yaklaşımının kullanılması ile bu belirsizlikler ifade edilebilmekte ve uygun çözümler sağlanabilmektedir. İki giriş değişkeni ve tek çıkış değişkeni olacak şekilde oluşturulan sugeno Tip-2 BM süre optimizasyonu modülüyle -6 sn ile +6 sn. arasında yeşil süreler değiştirilmektedir. Böylece faz optimizasyon modülü içerisindeki optimum devre süresinin belirlenmesi esnasında yeşil süreler uygulanan 8 sn. ile 60 sn. kısıtlarının altında veya üstünde süre ihtiyacı olduğunda bu talep karşılanabilmektedir. Dolayısı ile, yeşil süre kısıtlarından kaynaklanan gecikmeler ortadan kaldırılabilecektir. HTKS sistemi kapsamında, oluşturulan amaç fonksiyonu, BM karar değişkeni ve KAA karar değişkenleri ile kısıtlar seti Çizelge 6-3’de sunulmaktadır. HTKS sisteminin genel çalışma yapısı Şekil 6.4’de verilmiştir.

**Çizelge 6-3** HTKS sistemi amaç fonksiyonu, karar değişkenleri ve kısıtlar seti

| Amaç Fonksiyonu           | $f = \min \left\{ \sum_{l=1}^k d_n \right\}$ |                                                    |
|---------------------------|----------------------------------------------|----------------------------------------------------|
| <b>Karar Değişkenleri</b> | $g_{değişim}$                                | $g_{değişim}$ yeşil ışık süresi değişim miktarı    |
|                           | $m_i$                                        | $m_i$ BM üst üyelik fonksiyonları merkezleri       |
|                           | $b_i$                                        | $b_i$ BM üst üyelik fonksiyonları sol bacakları    |
|                           | $b_j$                                        | $b_j$ BM üst üyelik fonksiyonları sağ bacakları    |
|                           | $m_{ii}$                                     | $m_{ii}$ BM alt üyelik fonksiyonları merkezleri    |
|                           | $b_{ii}$                                     | $b_{ii}$ BM alt üyelik fonksiyonları sol bacakları |
|                           | $b_{jj}$                                     | $b_{jj}$ BM alt üyelik fonksiyonları sağ bacakları |

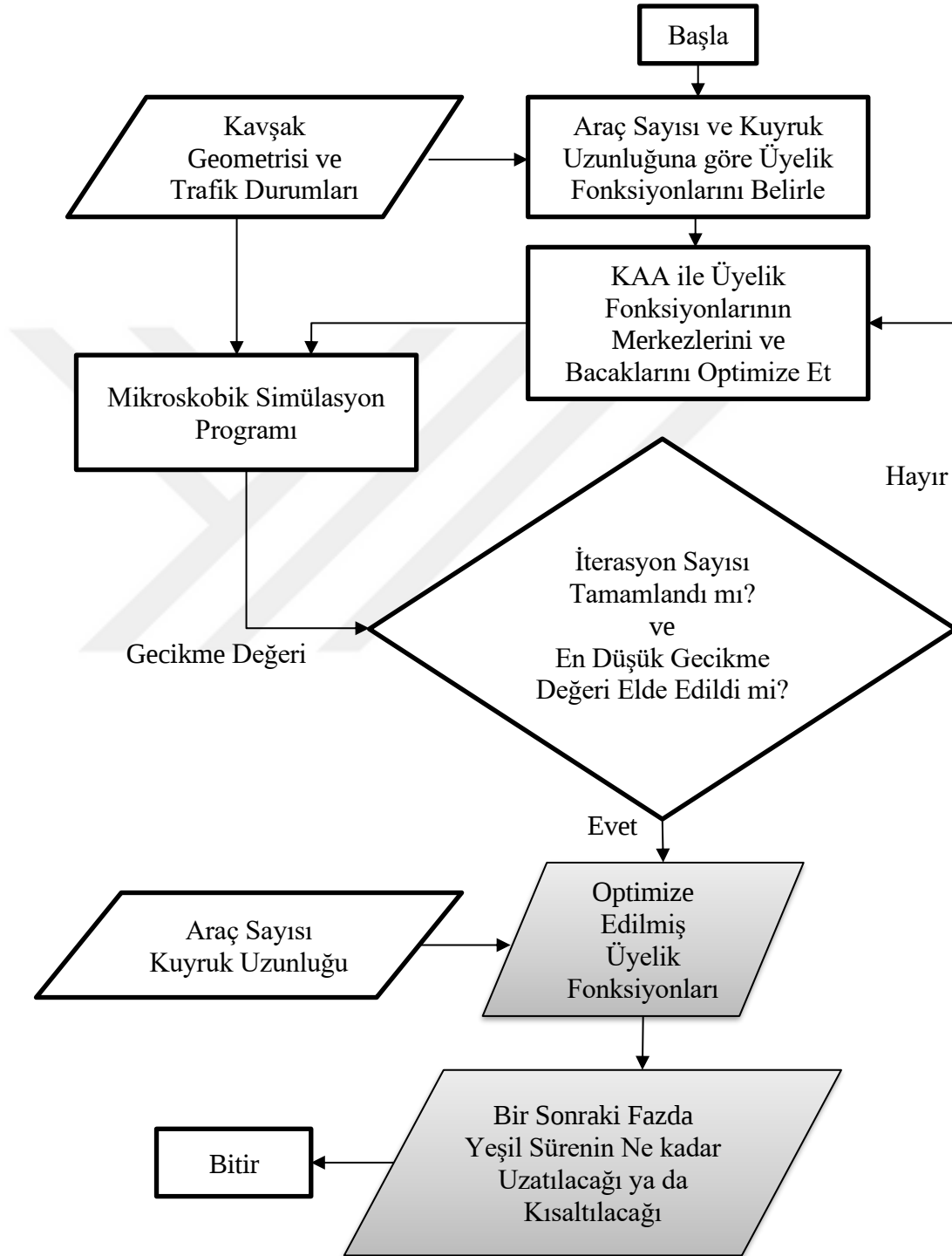
|                 |                               |
|-----------------|-------------------------------|
| <b>Kısıtlar</b> | $-6 \leq g_{değişim} \leq +6$ |
|                 | $m_i \leq m_{i+1}$            |
|                 | $m_i = m_{ii}$                |
|                 | $b_i \leq m_i$                |
|                 | $b_j \geq m_i$                |
|                 | $b_i \leq b_{ii} \leq m_i$    |
|                 | $m_i \leq b_{jj} \leq b_j$    |



**Şekil 6-4** HTKS sistemi genel yapısı

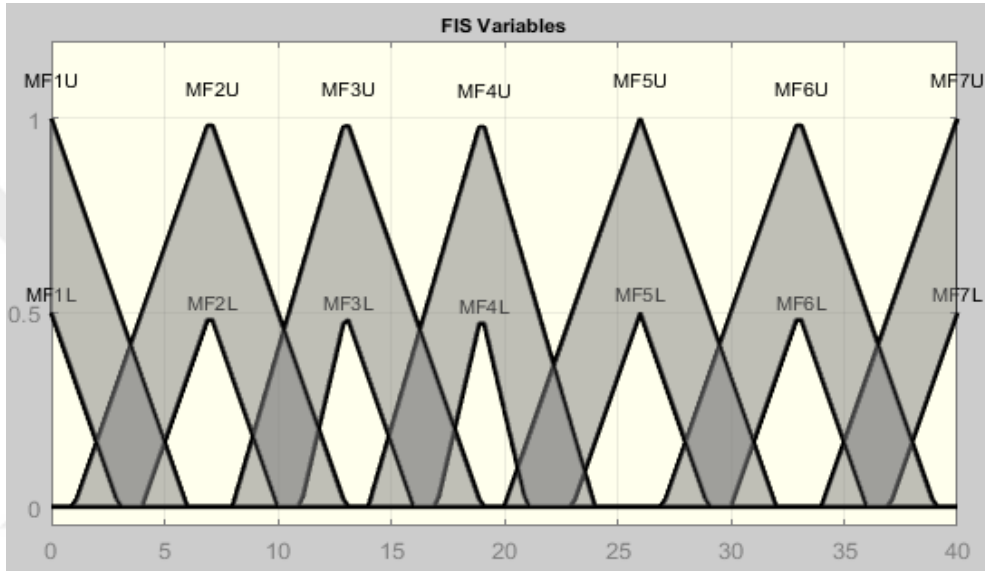
Bu yaklaşımın tek bir üyelik fonksiyonuna göre çalıştırılması farklı trafik durumlarında yetersizliklere sebep olmaktadır. Bu nedenle her bir trafik durumuna göre üyelik fonksiyonlarının optimize edilmesi kaçınılmazdır. Her bir trafik durumu için üyelik fonksiyonlarının optimizasyonu KAA yaklaşımı ile sağlanmıştır. En düşük gecikme

değerinin elde edilmesi amaçlanan KAA yöntemine göre her bir üyelik kümesinin merkezi ve bacaları optimize edilmiştir. Sinyal süresi optimizasyon modülünün akış diyagramı Şekil 6-5’de verilmiştir.

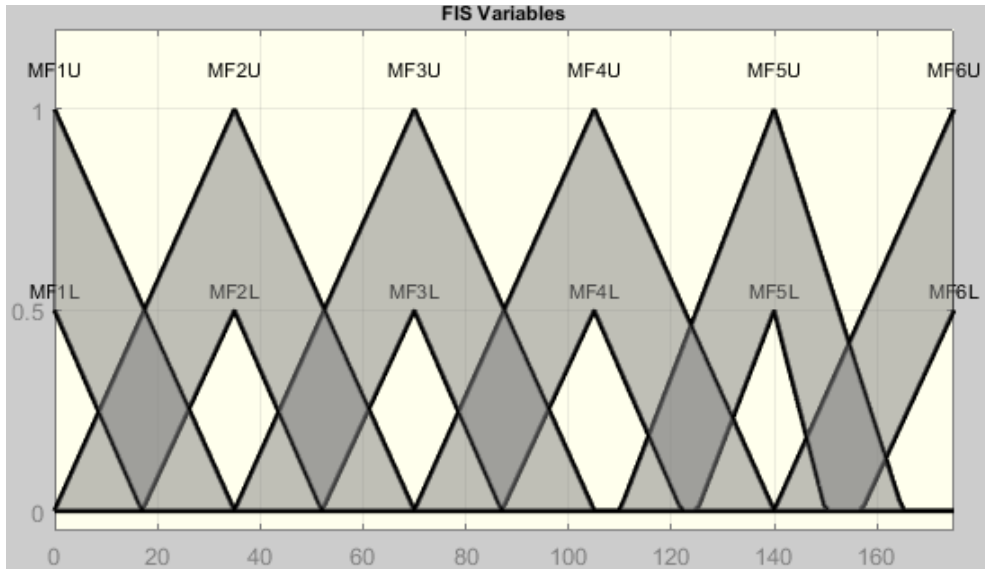


**Şekil 6-5** Sinyal süresi optimizasyon modülü akış diyagramı

Sinyal süresi optimizasyon modülünün başta oluşturduğu taşıt sayısı ve kuyruk uzunluğu üyelik fonksiyonları sırasıyla 7 ve 6 kümeye ayrılmış ve üyelik dereceleri belirlenmiştir. Taşıt sayısı üyelik fonksiyonunun 7 kümeye ayrılması, ölçüm aralığı olan 175 m’de bulunan maksimum 35 otomobil sayısına göre belirlenmiştir. Kuyruk uzunluğu üyelik fonksiyonu ise her 30 m’de kümelere ayrılmıştır. Araç sayısı ve kuyruk uzunluğu için üçgen tipte oluşturulan üyelik fonksiyonları Şekil 6-6 ve 6-7’de gösterilmiştir.

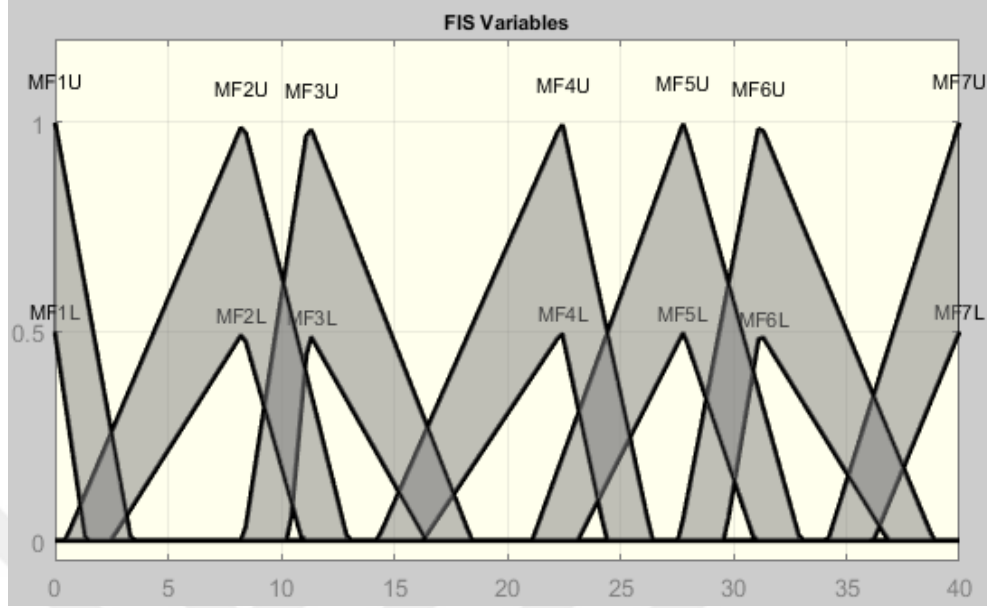


**Şekil 6-6** Araç sayısı üyelik fonksiyonları

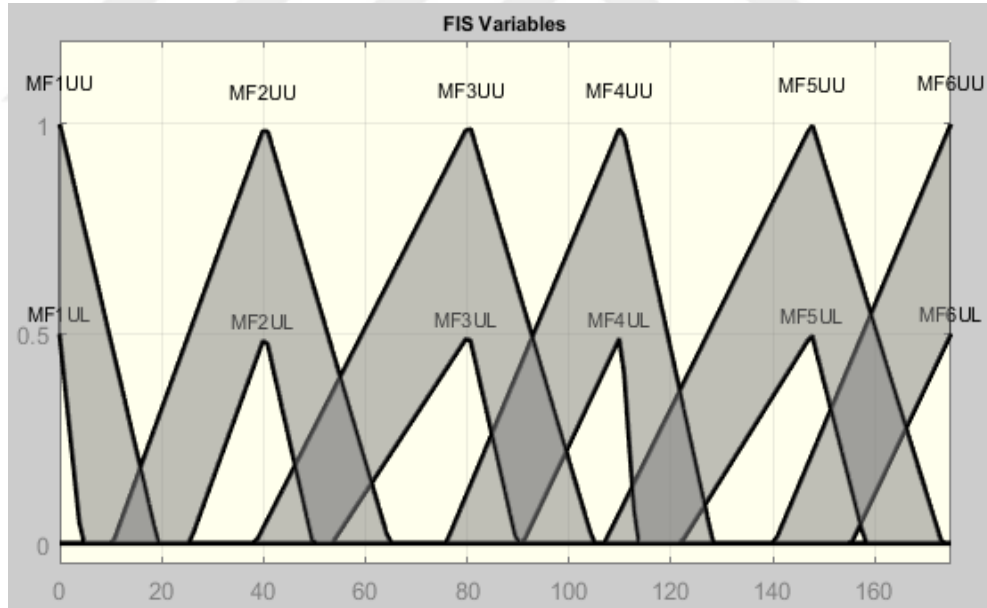


**Şekil 6-7** Kuyruk uzunluğu üyelik fonksiyonları

KAA ile optimize edilmiş üyelik fonksiyonları ise Şekil 6-8 ve 6-9’da verilmiştir.



**Şekil 6-8** Optimize edilmiş araç sayısı üyelik fonksiyonları



**Şekil 6-9** Optimize edilmiş kuyruk uzunluğu üyelik fonksiyonları

Üyelik fonksiyonlarında MFU üst üyelik fonksiyonunu ve MFL ise alt üyelik fonksiyonunu ifade etmektedir. Üyelik fonksiyonlarının optimizasyonunda Tip-2 BM yaklaşımının çalışma mantığıyla çakışmasına yol açacak şekilde düzenlemelerin

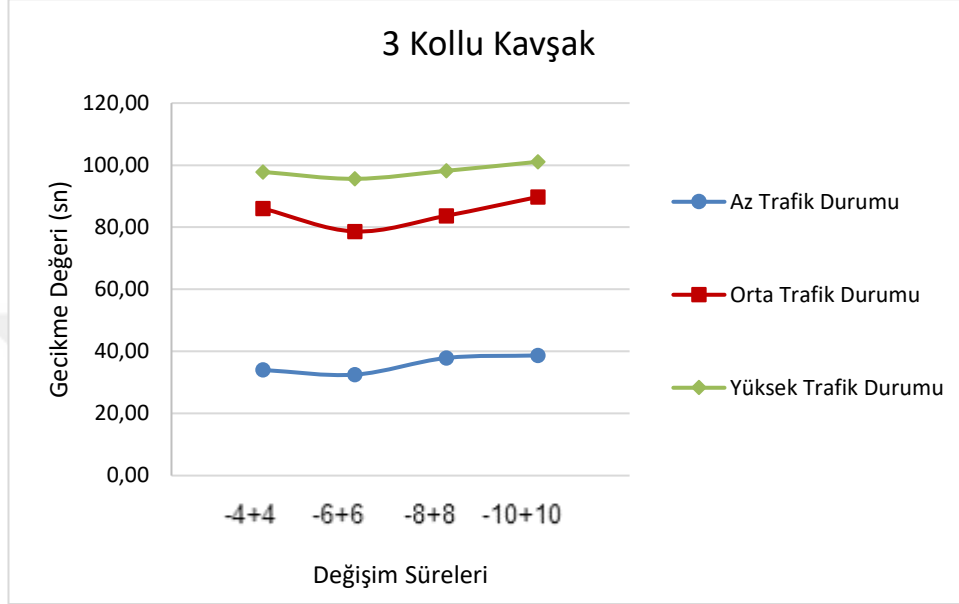
önlenmesi için kısıtlar konulmuştur. Üst üyelik fonksiyonlarının merkezlerinin yerinin belirlenmesinde kümelerin diğer kümelere göre konumlarının değişmesi engellenmiştir. Diğer bir deyişle, örneğin Şekil 5-8’de görünen MF3UU kümesinin MF2UU ve MF4UU kümesi arasında olacak şekilde konumu belirlenmiş ve bu kümelerin dışına çıkması engellenmiştir. Bir diğer önemli kısıt alt üyelik fonksiyonları ile alakalıdır. Tip-2 BM yaklaşımında alt üyelik fonksiyonlarının merkezlerinin üst üyelik fonksiyonları ile aynı konumda olması gerekmektedir. Merkezlerde olduğu gibi üyelik fonksiyonlarının bacalarının optimize edilmesinde de kısıtlar söz konusudur. Üst üyelik fonksiyonlarında bir kümenin bacalarının konumu komşu kümelerinin sınırlarını geçmemektedir. Alt üyelik fonksiyonunda ise bacaların konumu bulunduğu üst üyelik fonksiyonunun sınır değerleri içerisinde kalmalıdır.

BM yaklaşımında çıkış kümesinin belirlenmesi için hangi aralıklarda yeşil sürenin arttırılıp azaltılacağı belirlenmesi gerekmektedir. Bu nedenle farklı süre değişim aralıkları incelenmiştir. -4 ile +4, -6 ile +6, -8 ile +8 ve -10 ile +10 saniye aralıklarının gecikme üzerindeki etkileri incelenerek hangi aralıkta süre değişimi yapılacağına karar verilmiştir. Az, orta ve yüksek trafik hacimleri üzerinden 3 ve 4 kollu kavşak geometrilerine göre 15 dakikalık simülasyon periyodunda her bir aralığın performansı elde edilmiş olup Çizelge 6-4’de gösterilmiştir.

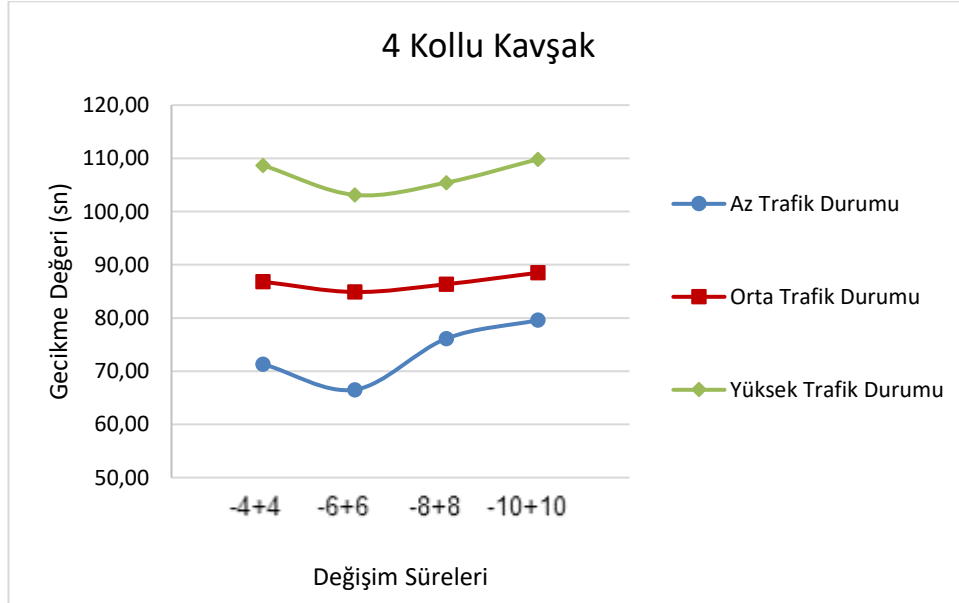
**Çizelge 6-4** Yeşil süre değişim aralıklarının performansı

| Trafik Durumları | <i>T</i> | Trafik Hacimleri     |                      |                      |                      | -4 +4   | -6 +6   | -8 +8   | -10 +10 |
|------------------|----------|----------------------|----------------------|----------------------|----------------------|---------|---------|---------|---------|
|                  |          | <i>V<sub>1</sub></i> | <i>V<sub>2</sub></i> | <i>V<sub>3</sub></i> |                      | Gecikme | Gecikme | Gecikme | Gecikme |
| Az               | 15       | 400                  | 400                  | 400                  |                      | 34,01   | 32,51   | 37,88   | 38,70   |
| Orta             | 15       | 800                  | 800                  | 800                  |                      | 86,00   | 78,65   | 83,70   | 89,77   |
| Yüksek           | 15       | 1200                 | 1200                 | 1200                 |                      | 97,82   | 95,60   | 98,18   | 101,09  |
| Trafik Durumları | <i>T</i> | <i>V<sub>1</sub></i> | <i>V<sub>2</sub></i> | <i>V<sub>3</sub></i> | <i>V<sub>4</sub></i> | Gecikme | Gecikme | Gecikme | Gecikme |
|                  |          |                      |                      |                      |                      |         |         |         |         |
| Az               | 15       | 400                  | 400                  | 400                  | 400                  | 71,33   | 66,53   | 76,13   | 79,59   |
| Orta             | 15       | 800                  | 800                  | 800                  | 800                  | 86,81   | 84,88   | 86,35   | 88,50   |
| Yüksek           | 15       | 1200                 | 1200                 | 1200                 | 1200                 | 108,68  | 103,13  | 105,44  | 109,84  |

3 kollu ve 4 kollu kavşak geometrilerinde trafik durumlarına göre süre değişim aralıklarının performans dağılımı Şekil 6-10 ve 6-11’de verilmiştir. Performans grafiklerindeki verilere göre, hem 3 kollu hem de 4 kollu kavşaklarda -6 ile +6 sn arasında süre değişimi yapılması ile en düşük gecikme değerleri elde edilebilmektedir.



**Şekil 6-10** 3 kollu kavşak süre değişim aralıkları performans dağılımı



**Şekil 6-11** 4 kollu kavşak süre değişim aralıkları performans dağılımı

Farklı deęişim süreleri kullanımının gecikme üzerindeki etkisine varyans analiz ile bakılmıştır. Varyans analizinde  $h_0$  ve  $h_1$  hipotezleri kurulmuştur.

- $h_0$  : Ortalamalar arasında fark yok.
- $h_1$  : En az bir ortalama dięerlerinden farklı.

Dört farklı deęişim süresine gecikme deęerlerinde anlamlı bir farkın olup olmadığına bakılırken tek yönlü varyans analizi (ANOVA) kullanılmış olup % 95 güven aralığına göre sonuçlar deęerlendirilmiştir. Varyans analizinin uygulanabilmesi için farklı deęişim sürelerine ait gecikme deęerlerinin normal dağılımı uyması gerekmektedir. Farklı normallik testleri bulunmakta olup, örnek sayısının 30'dan az olmasından dolayı Shapiro-Wilk kullanılmıştır. Dört farklı deęişim süresine ait Shapiro-Wilk normallik testi sonuçları Çizelge 6-5'de verilmiştir.

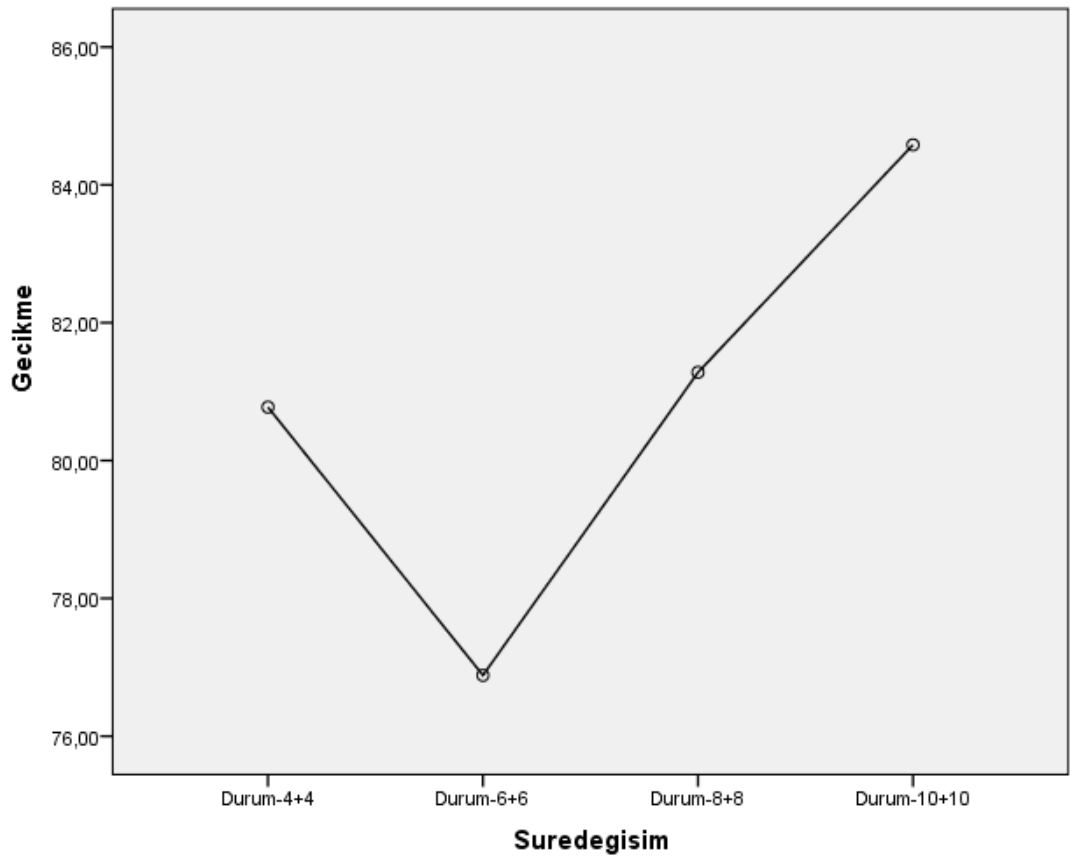
**Çizelge 6-5** Normallik testi sonuçları

|         | Süre deęişim | Shapiro-Wilk |              |       |
|---------|--------------|--------------|--------------|-------|
|         |              | İstatistik   | Örnek Sayısı | Sigma |
| Gecikme | Durum-4+4    | 0,904        | 6            | 0,399 |
|         | Durum-6+6    | 0,920        | 6            | 0,507 |
|         | Durum-8+8    | 0,884        | 6            | 0,290 |
|         | Durum-10+10  | 0,873        | 6            | 0,240 |

Shapiro-Wilk testi sonuçlarında anlamlılık deęeri olan sigmalara bakıldığında eşik anlamlılık deęeri olan 0,05'den büyük olduęu için deęişim sürelerine ait gecikme deęerleri normal dağılıma uygundur. Varyans homojenlik testi olan Levene testi ile ANOVA testi sonuçları Çizelge 6-6'da verilmiştir. Homojenlik testine göre hesaplanan anlamlılık deęeri olan 0,992 eşik anlamlılık deęeri olan 0,05'den büyük için olduęu dört grup arasında anlamlı bir farklılık yoktur, öyleyse varyanslar homojenliği sağlanmıştır. ANOVA testinde anlamlılık deęeri olan 0,962 eşik anlamlılık deęeri olan 0,05'den büyük olduęu için  $h_0$  hipotezi kabul edilir. Dięer bir deyişle dört grup arasında gecikme deęeri bakımından anlamlı bir fark yoktur. Dört farklı deęişim süresine ait ortalama gecikme deęerleri Şekil 6-12'de verilmiştir.

**Çizelge 6-6** Levene ve ANOVA testi sonuçları

|             | Örnek Sayısı | Ortalama | Std. Sapma | Levene Homojenlik Testi | ANOVA    |          |
|-------------|--------------|----------|------------|-------------------------|----------|----------|
|             |              |          |            |                         | <i>F</i> | <i>P</i> |
| Durum-4+4   | 6            | 80,77    | 26,11      | 0,992                   | 0,096    | 0,962    |
| Durum-6+6   | 6            | 76,88    | 25,22      |                         |          |          |
| Durum-8+8   | 6            | 81,28    | 23,71      |                         |          |          |
| Durum-10+10 | 6            | 84,58    | 24,83      |                         |          |          |
| Total       | 24           | 80,88    | 23,46      |                         |          |          |



**Şekil 6-12** Dört farklı değişim süresi gecikme değerleri

Varyans analizine göre değişim süreleri aralarında anlamlı bir farklılık olmadığı için herhangi birisi kullanılabilir. Şekil 6-12 incelendiğinde En düşük gecikme değerini veren -6 ile +6 sürelerinin kullanılması uygun görülmüştür.

-6 ile +6 saniye arasında deęişim sürelerine Tip-2 BM yaklaşımının çıkış üyelik fonksiyonları oluşturulmuş ve 7 kümeye ayrılmıştır. Çıkış kümesinin dağılımı ve sözel ifadeleri Çizelge 5-7’de verilmiştir.

**Çizelge 6-7** Tip-2 BM çıkış üyelik fonksiyonu

| Üyelik Fonksiyonları | Sözel İfade      | Deęer |
|----------------------|------------------|-------|
| MF1                  | Aşırı Az (AA)    | -6    |
| MF2                  | Çok Az (ÇA)      | -4    |
| MF3                  | Az (A)           | -2    |
| MF4                  | Normal (N)       | 0     |
| MF5                  | Fazla (F)        | 2     |
| MF6                  | Çok Fazla (ÇF)   | 4     |
| MF7                  | Aşırı Fazla (AF) | 6     |

BM yaklaşımında karar verme sürecinde etkili olan kural tabanının oluşturulması gereklidir. “Eğer-İse” şeklindeki sözel anlatım yoluyla kural tabanının oluşturulması sağlanmış olup Eşitlik 6.6’da verilmiştir. Giriş deęişkenlerinin 6 ve 7 kümeye ayrılmasından dolayı 42 farklı kural durumu oluşmaktadır ve Çizelge 6-8’de bu kurallar verilmiştir.

$$\text{Eğer } \{AS = MF1\} \text{ ve } \{KU = MF1\} \text{ İse } \{YD AA\} \quad (6.6)$$

Burada AS araç sayısını, KU kuyruk uzunluğunu, YD yeşil süre deęişimini ifade etmekte olup AA ise sürenin azaltılma miktarını göstermektedir.

Çizelge 6-8 BM kural tabanı

|    |      |         |    |        |     |        |
|----|------|---------|----|--------|-----|--------|
| 1  | Eğer | AS= MF1 | ve | KU=MF1 | İse | YD=ÇÇA |
| 2  | Eğer | AS= MF1 | ve | KU=MF2 | İse | YD=ÇA  |
| 3  | Eğer | AS= MF1 | ve | KU=MF3 | İse | YD=A   |
| 4  | Eğer | AS= MF1 | ve | KU=MF4 | İse | YD=N   |
| 5  | Eğer | AS= MF1 | ve | KU=MF5 | İse | YD=F   |
| 6  | Eğer | AS= MF1 | ve | KU=MF6 | İse | YD=ÇF  |
| 7  | Eğer | AS= MF2 | ve | KU=MF1 | İse | YD=A   |
| 8  | Eğer | AS= MF2 | ve | KU=MF2 | İse | YD=A   |
| 9  | Eğer | AS= MF2 | ve | KU=MF3 | İse | YD=N   |
| 10 | Eğer | AS= MF2 | ve | KU=MF4 | İse | YD=N   |
| 11 | Eğer | AS= MF2 | ve | KU=MF5 | İse | YD=F   |
| 12 | Eğer | AS= MF2 | ve | KU=MF6 | İse | YD=ÇF  |
| 13 | Eğer | AS= MF3 | ve | KU=MF1 | İse | YD=A   |
| 14 | Eğer | AS= MF3 | ve | KU=MF2 | İse | YD=N   |
| 15 | Eğer | AS= MF3 | ve | KU=MF3 | İse | YD=F   |
| 16 | Eğer | AS= MF3 | ve | KU=MF4 | İse | YD=F   |
| 17 | Eğer | AS= MF3 | ve | KU=MF5 | İse | YD=ÇF  |
| 18 | Eğer | AS= MF3 | ve | KU=MF6 | İse | YD=ÇÇF |
| 19 | Eğer | AS= MF4 | ve | KU=MF1 | İse | YD=N   |
| 20 | Eğer | AS= MF4 | ve | KU=MF2 | İse | YD=F   |
| 21 | Eğer | AS= MF4 | ve | KU=MF3 | İse | YD=F   |
| 22 | Eğer | AS= MF4 | ve | KU=MF4 | İse | YD=F   |
| 23 | Eğer | AS= MF4 | ve | KU=MF5 | İse | YD=ÇF  |
| 24 | Eğer | AS= MF4 | ve | KU=MF6 | İse | YD=ÇÇF |
| 25 | Eğer | AS= MF5 | ve | KU=MF1 | İse | YD=N   |
| 26 | Eğer | AS= MF5 | ve | KU=MF2 | İse | YD=F   |
| 27 | Eğer | AS= MF5 | ve | KU=MF3 | İse | YD=F   |
| 28 | Eğer | AS= MF5 | ve | KU=MF4 | İse | YD=ÇF  |
| 29 | Eğer | AS= MF5 | ve | KU=MF5 | İse | YD=ÇF  |
| 30 | Eğer | AS= MF5 | ve | KU=MF6 | İse | YD=ÇÇF |
| 31 | Eğer | AS= MF6 | ve | KU=MF1 | İse | YD=F   |
| 32 | Eğer | AS= MF6 | ve | KU=MF2 | İse | YD=F   |
| 33 | Eğer | AS= MF6 | ve | KU=MF3 | İse | YD=ÇF  |
| 34 | Eğer | AS= MF6 | ve | KU=MF4 | İse | YD=ÇF  |
| 35 | Eğer | AS= MF6 | ve | KU=MF5 | İse | YD=ÇF  |
| 36 | Eğer | AS= MF6 | ve | KU=MF6 | İse | YD=ÇÇF |
| 37 | Eğer | AS= MF7 | ve | KU=MF1 | İse | YD=F   |
| 38 | Eğer | AS= MF7 | ve | KU=MF2 | İse | YD=ÇF  |
| 39 | Eğer | AS= MF7 | ve | KU=MF3 | İse | YD=ÇF  |
| 40 | Eğer | AS= MF7 | ve | KU=MF4 | İse | YD=ÇF  |
| 41 | Eğer | AS= MF7 | ve | KU=MF5 | İse | YD=ÇÇF |
| 42 | Eğer | AS= MF7 | ve | KU=MF6 | İse | YD=ÇÇF |

#### 6.4. Tip-1 BM Trafik Kontrol Sistemi

Doğan, doktora tezinde izole sinyalizasyon kavşakları için bulanık mantık tabanlı sinyal kontrol sistemi (Tip-1 BM-TKS) geliştirmiştir [25]. İki farklı bulanık mantık modülüne sahip olan sistemde birinci BM modülü ile yeşil faz süreleri belirlenmiş ve ikinci BM modülü ile de yeşil sürenin sonunda uzatılma kararı verilmiş fazın ne kadar süre uzatılacağını belirlenmiştir. Geliştirilen Tip-1 BM-TKS 5 ana modülden oluşmaktadır.

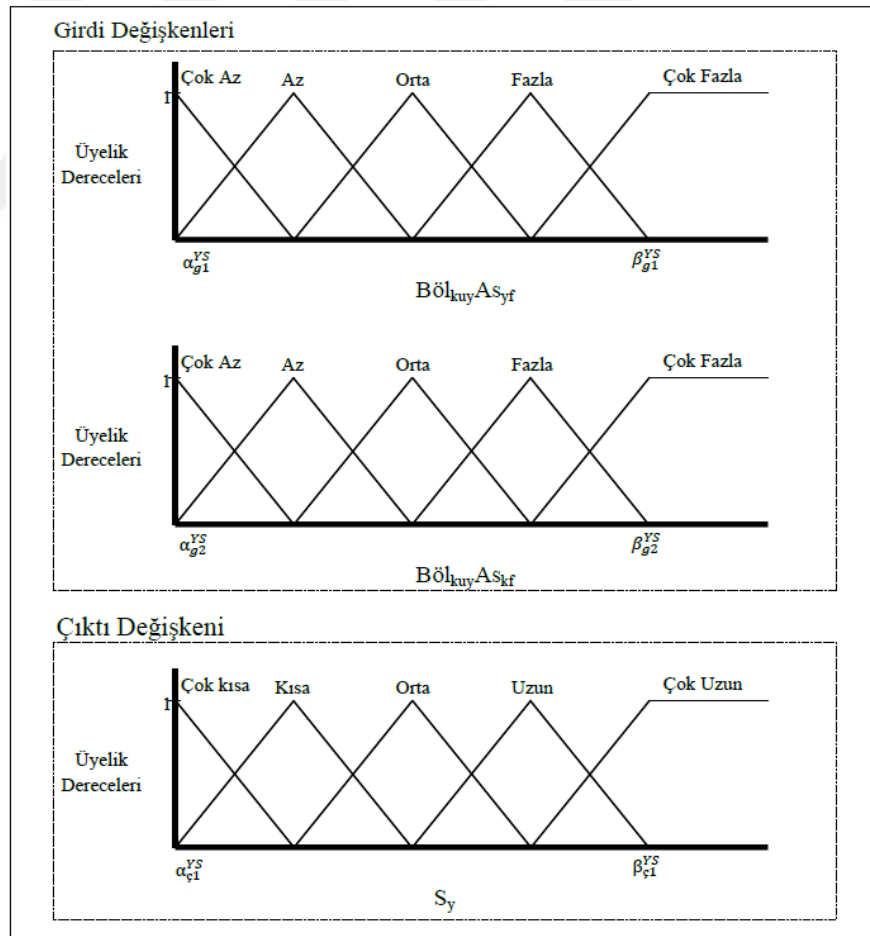
- Yeşil Faz Seçim Modülü (YF-SM)
- Yeşil Süre Bulanık Mantık Modülü (YS-BMM)
- Yeşil Süre Uzatma Karar Modülü (YS-UKM)
- Uzatma Süresi Bulanık Mantık Modülü (US-BMM)
- Bekleme Süresi Kontrol Modülü (BS-KM)

YF-SM’de hangi faza geçiş hakkı verileceği belirlenmektedir. Araç sayısına bağlı olarak yeşil ışık yakılacak fazın belirlendiği bu modülde; en yüksek araç sayısına sahip olan faza geçiş hakkı verilmiştir. Araç sayılarının eşit olması durumunda ise şerit sayısı en az olan faza geçiş hakkı verilmiştir. YS-BMM’de ise geçiş hakkı verilen faza yakılacak yeşil süre belirlenmektedir. Bu modülde yeşil sürenin uzunluğuna BM’nin iki giriş parametresine göre karar verilmiştir. Bu giriş parametreleri yeşil ışık alacak faza ait kuyruk bölgesinde bulunan araç sayısı olan (BölkuyAsyf) ve diğer fazlara ait kuyruk bölgelerindeki toplam araç sayısı (BölkuyAskf) Girdi parametreleri Çok Az, Az, Orta, Fazla, Çok Fazla olarak 5 kümeye ayrılmış ve 25 farklı kural durumu bu kümelere göre oluşturulmuştur. Doğan tarafından kullanılan kural tabanı Çizelge 6-9’de verilmiştir.

Çizelge 6-9 YS-BMM kural tabanı [25]

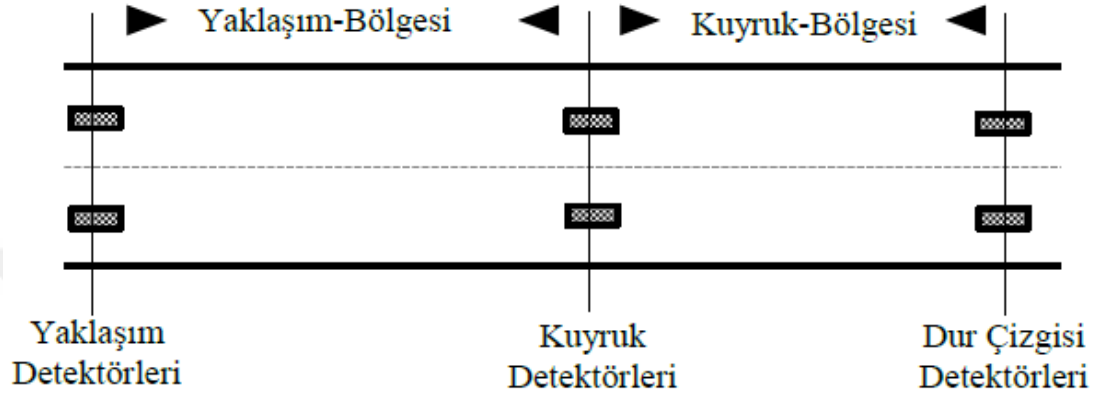
|                    |           | $Böl_{kuy}As_{kf}$ |    |      |       |           |
|--------------------|-----------|--------------------|----|------|-------|-----------|
|                    |           | Çok Az             | Az | Orta | Fazla | Çok Fazla |
| $Böl_{kuy}As_{yf}$ | Çok Az    | ÇK                 | ÇK | ÇK   | ÇK    | ÇK        |
|                    | Az        | A                  | A  | A    | A     | ÇK        |
|                    | Orta      | O                  | O  | O    | O     | A         |
|                    | Fazla     | F                  | F  | F    | O     | O         |
|                    | Çok Fazla | ÇF                 | ÇF | F    | F     | O         |

Oluşturulan 25 kural durumuna göre çıktı parametresi olan yeşil süre miktarı ( $S_y$ ) belirlenmiştir. Çıktı parametresi de 5 kümeye ayrılmış ve sözel ifadeleri Çok Kısa, Kısa, Orta, Uzun ve Çok Uzun olarak belirlenmiştir. Üçgen tipte oluşturulan girdi ve çıktı üyelik fonksiyonları Şekil 6-13’de gösterilmiştir.



Şekil 6-13 YS-BMM üyelik fonksiyonları [25]

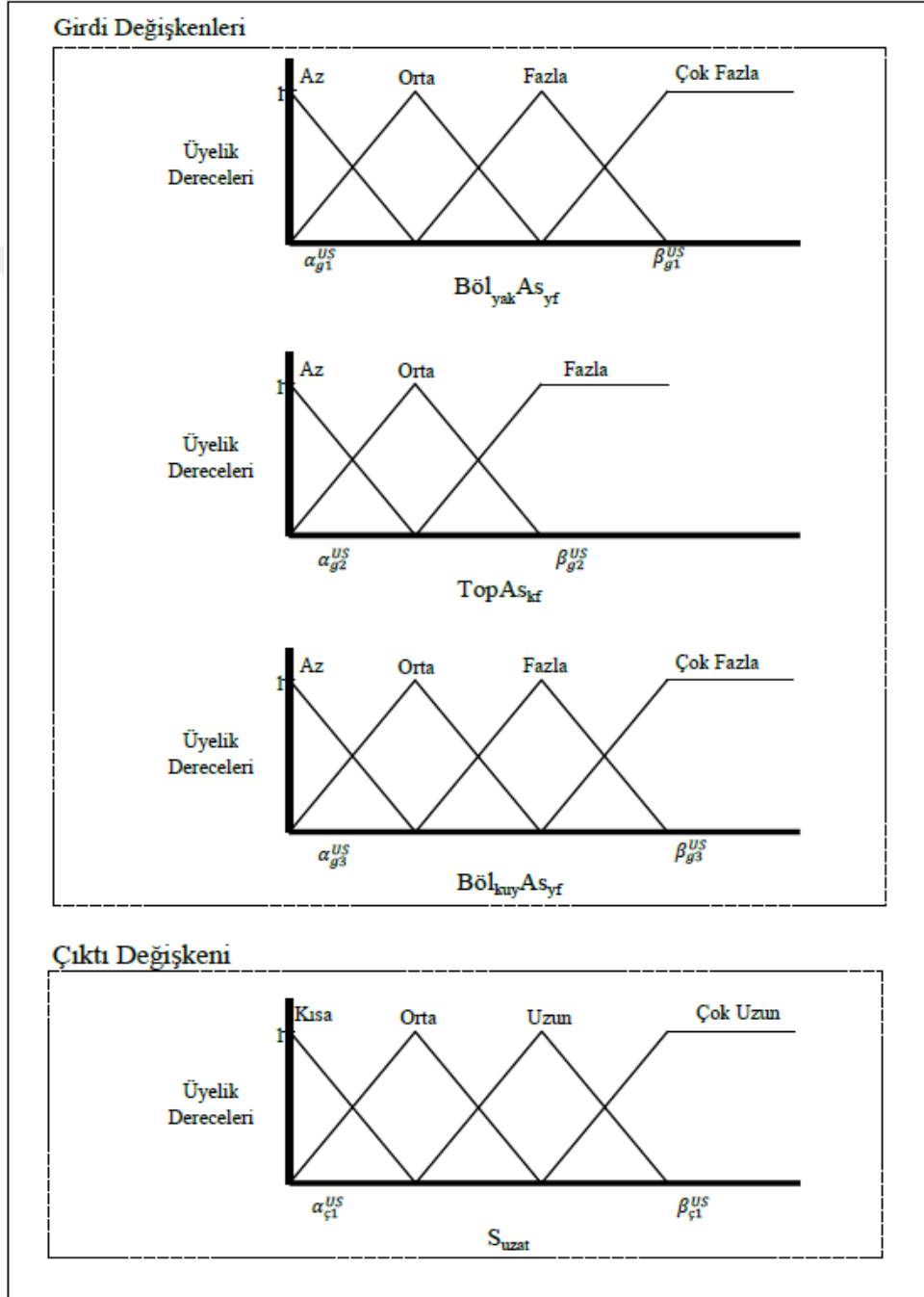
YS-UKM’de geiş hakkı verilen fazda yeşil süreninin uzatılıp uzatılmayacağına karar verilmektedir. Karar verilme işleminde Kuyruk Bölgesi (Bölkuy) ve Yaklaşım Bölgelerine (Bölyak) ait araç sayıları kullanılmaktadır. Bölgeler ve detektör yerleşimleri Şekil 6-14’de gösterilmiştir. Kuyruk bölgesi 75 m ve yaklaşım bölgesi 100 m olarak seçilmiş ve detektörler buna göre konumlandırılmıştır.



**Şekil 6-14** Yaklaşım kolu bölgeleri ve detektörler [25]

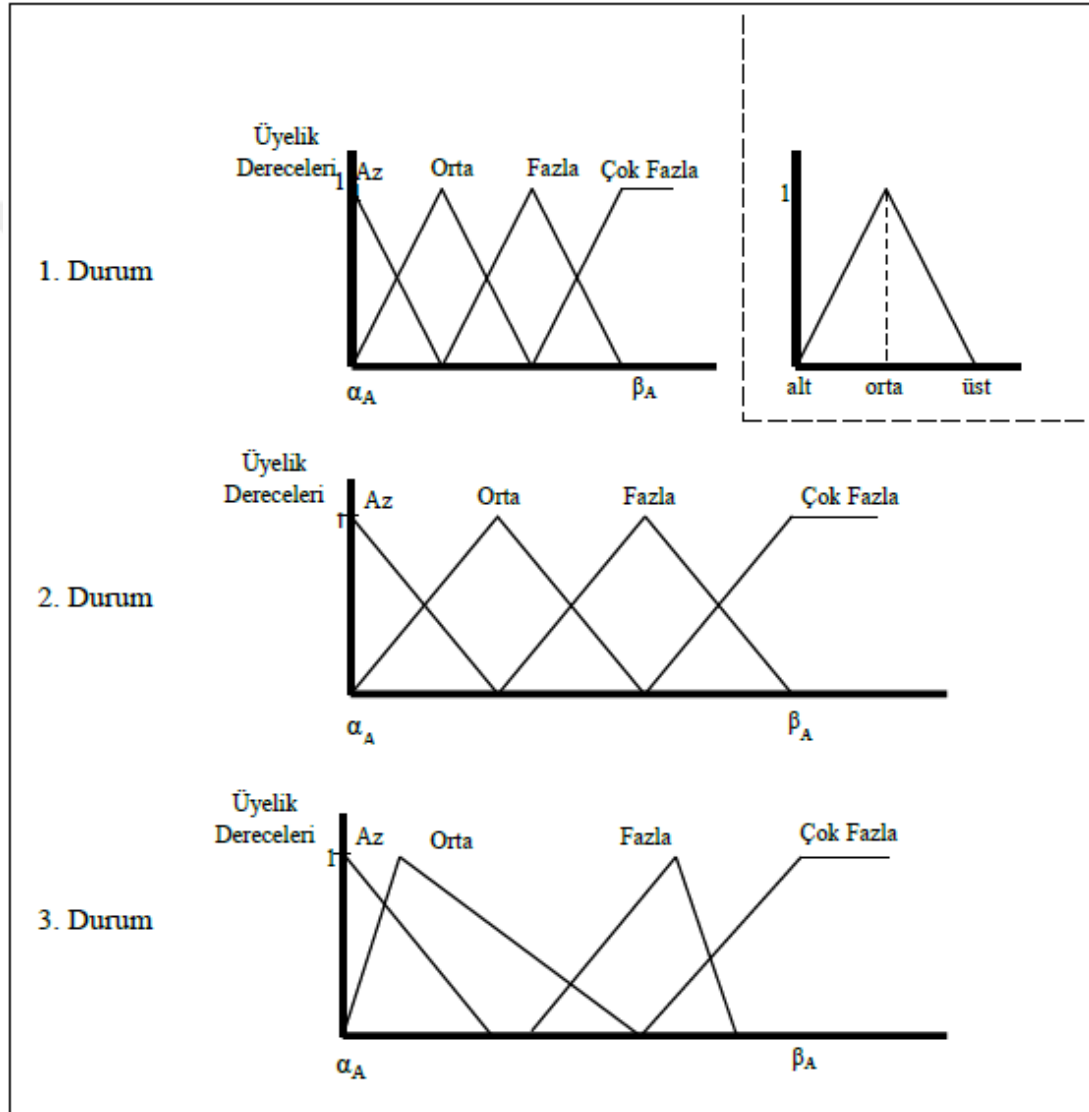
Uzatılma kararının verilmesi esnasında yeşil ışık yanan faza ait kuyruk ve yaklaşım bölgelerindeki toplam araç sayısı, diğer fazlara ait toplam araç sayılarıyla karşılaştırılır. Eğer yeşil ışık yanan faza ait araç sayıları diğer fazlardaki araç sayılarına eşit ya da daha fazla ise uzatma kararı verilir, aksi halde faz değiştirilir. Doğan’ın US-BMM modülünde ise uzatma kararı verilen faza ait uzatma süresinin miktarı belirlenmektedir. Diğer fazlardaki toplam araç sayısı (TopAskf), yeşil faza ait yaklaşım bölgesi araç sayısı (BölyakAsyf) ve yeşil faza ait kuyruk bölgesi araç sayısı (BölkuyAsyf) olmak üzere 3 adet girdi değişkenine göre uzatma süresine karar verilmiştir. Yaklaşım bölgesi ve kuyruk bölgesi araç sayısı değişkenleri Az, Orta, Fazla ve Çok Fazla olarak 4 kümeye, diğer fazlardaki araç sayıları değişkeni ise Az, Orta ve Fazla olmak üzere 3 kümeye ayrılmıştır. US-BMM’ye ait üçgen tipte oluşturulan girdi ve çıktı üyelik fonksiyonları Şekil 6-15’de gösterilmiştir. Girdi değişkenlerine göre oluşturulan 48 farklı kural durumu kullanılarak uzatma süresinin miktarına karar verilmiştir. Geçiş hakkı verilecek fazın belirlenmesinde fazlara ait araç sayılarının kullanılmasından dolayı düşük araç sayısına sahip olan fazların geçiş hakkı alamaması durumu oluşmaktadır. Bu nedenle BS-KM oluşturulmuş ve araçların

bekleme süreleri kontrol edilmiştir. Eğer belirlenen maksimum bekleme süresinden daha fazla bekleyen faz olması durumunda o faza bir sonraki geçiş hakkı verilmiştir. Birden fazla maksimum bekleme süresini aşan faz olduğunda ise en fazla araç sayısına sahip olan faza geçiş hakkı tanınmaktadır. Böylece sistem içerisinde uzun bekleme sürelerinin ve güvenlik ihlallerinin önüne geçilmiştir.



Şekil 6-15 US-BMM üyelik fonksiyonları [25]

Tip-1BM-TKS yaklaşımının farklı trafik akımlarına daha iyi adapte olabilmesi için Doğan BM modülünü iki aşamalı olarak optimize etmiştir. Birinci aşamada her bir bulanık kümeye ait sınır değerlerini belirleyerek optimize etmiş ve ikinci aşamada ise belirlenen sınır değerlerine göre üyelik fonksiyonlarının ayarlanmasını sağlamıştır. Üyelik fonksiyonlarının optimizasyon seviyeleri sonrasındaki durumuna ait örnek Şekil 6-16’da gösterilmiştir.



**Şekil 6-16** Bulanık mantık optimizasyonu [25]

## 7. SİMÜLASYON SONUÇLARI VE PERFORMANS KARŞILAŞTIRMA

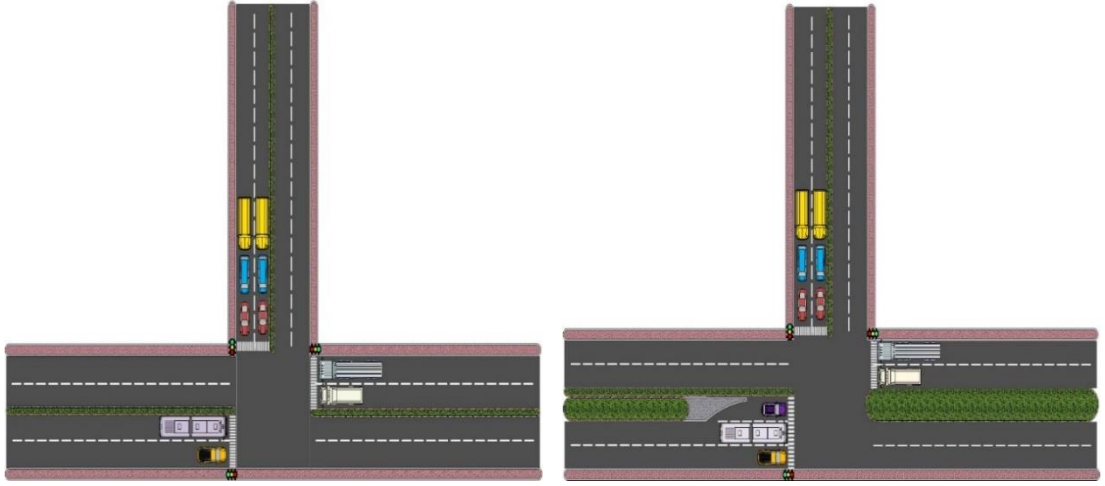
Geliştirilen kontrol sistemlerinin, birçok trafik durumuna ve farklı kavşak geometrilerine uygun olması ve etkili sonuçlar ortaya koyabilmesi gerekmektedir. Yeni önerilen kontrol sistemlerinin uygunluğu istatistiksel olarak doğrulanmalıdır. Bu doğrultuda mevcut sistemlerle performans karşılaştırması yapılarak istatistiksel olarak performans sonuçları ortaya konulmuştur. Ortalama Yüzdesel Hatalar (OYH) karşılaştırmada kullanılmış olup OYH ifadesi Eşitlik 7.1’de verilmiştir.

$$OYH = 1/n \sum_{i=1}^n \frac{Gecikme_r - Gecikme_h}{Gecikme_r} * 100 \quad (7.1)$$

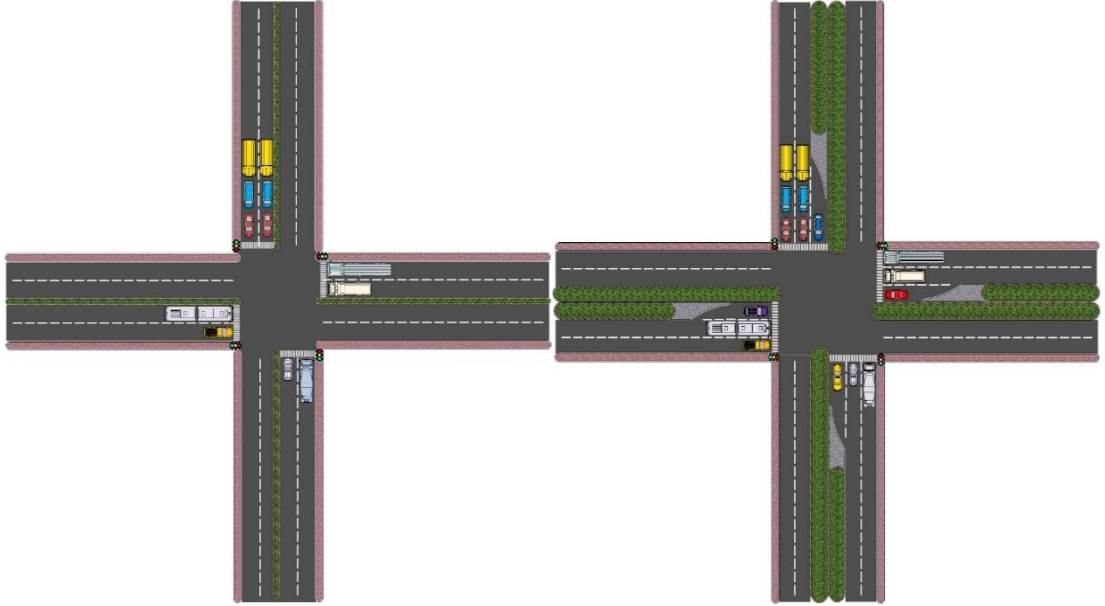
Burada n örnek sayısı,  $Gecikme_r$  referans alınacak kontrol sisteminin gecikme değerini ve  $Gecikme_h$  yüzdesel hatası hesaplanacak kontrol sisteminin gecikme değerini ifade etmektedir.

Önerilen kontrol yaklaşımın performans değerlendirmesinde farklı trafik durumları ve kavşak geometrileri göz önüne alınmıştır. Düşük, orta ve yüksek olmak üzere trafik hacimleri içeren 15 farklı senaryo oluşturulmuştur. Ayrıca 3 kollu ve 4 kollu olmak üzere 4 farklı kavşak geometrisine göre simülasyonlar gerçekleştirilmiştir. Kavşak geometrileri Şekil 7-1 ve 7-2’de verilmiş olup simülasyon için yapılan kabuller ise aşağıdaki gibidir.

- Her kol 2 şeritli, şerit genişlikleri 3.6m ve eğim 0’dır.
- Başlangıç kaybı 3.6 sn’dir.
- Serbest akım hızı 50 km/saat’tir.
- Kayıp zamanlar; sarı ışık 2 sn ve her yöne kırmızı 1 sn olarak alınmıştır.
- Sağa dönüşler % 10 ve sola dönüler % 20 olarak belirlenmiştir.
- Doygun akım 1800 araç/saat/şerit olarak ayarlanmıştır.
- Simülasyon süresi 15 dakikadır.
- Yayalar fazlar arası geçiş yapmaktadır.
- Sürücüler herhangi bir ihlal yapmamaktadır.



**Şekil 7-1** 3 kollu kavşak geometrileri



**Şekil 7-2** 4 kollu kavşak geometrileri

Farklı araç tiplerinin gecikme üzerindeki etkisini de sisteme dahil etmek için 5 farklı araç tipinin sistemde simülasyonunun yapılması sağlanmıştır. Otomobil, minibüs, otobüs, kamyon ve tır olarak farklı uzunluklara ve hızlanma-yavaşlama ivmelerine sahip araçlar kullanılmıştır. Senaryolarda oluşturulan trafik durumlarında % 80 otomobil, % 10 minibüs, % 5 otobüs, % 3 kamyon ve % 2 tır olarak araç tipi dağılımı oluşturulmuştur. Simülasyonda kullanılan trafik senaryoları 3 ve 4 kollu kavşaklar için Çizelge 7-1 ve 7-2’de verilmiştir. Her koldan gelen trafik hacimlerinin aynı olması

durumu eşit olarak, kollarda farklı trafik hacimlerinin olması durumu da karışık olarak ifade edilmiştir.

**Çizelge 7-1** 3 kollu kavşak için trafik senaryoları

| Durum  | Trafik Şablonları | Doğu (araç/s) | Batı (araç/s) | Kuzey (araç/s) |
|--------|-------------------|---------------|---------------|----------------|
| Düşük  | Eşit              | 100           | 100           | 100            |
|        | Eşit              | 250           | 250           | 250            |
|        | Eşit              | 400           | 400           | 400            |
|        | Karışık           | 100           | 250           | 400            |
|        | Karışık           | 400           | 250           | 250            |
| Orta   | Eşit              | 500           | 500           | 500            |
|        | Eşit              | 650           | 650           | 650            |
|        | Eşit              | 800           | 800           | 800            |
|        | Karışık           | 500           | 650           | 650            |
|        | Karışık           | 650           | 800           | 800            |
| Yüksek | Eşit              | 900           | 900           | 900            |
|        | Eşit              | 1050          | 1050          | 1050           |
|        | Eşit              | 1200          | 1200          | 1200           |
|        | Karışık           | 900           | 1050          | 1050           |
|        | Karışık           | 1200          | 1200          | 900            |

**Çizelge 7-2** 4 kollu kavşak için trafik senaryoları

| Durum  | Trafik Şablonları | Doğu (araç/s) | Batı (araç/s) | Kuzey (araç/s) | Güney (araç/s) |
|--------|-------------------|---------------|---------------|----------------|----------------|
| Düşük  | Eşit              | 100           | 100           | 100            | 100            |
|        | Eşit              | 250           | 250           | 250            | 250            |
|        | Eşit              | 400           | 400           | 400            | 400            |
|        | Karışık           | 100           | 250           | 400            | 400            |
|        | Karışık           | 400           | 250           | 250            | 100            |
| Orta   | Eşit              | 500           | 500           | 500            | 500            |
|        | Eşit              | 650           | 650           | 650            | 650            |
|        | Eşit              | 800           | 800           | 800            | 800            |
|        | Karışık           | 500           | 650           | 650            | 800            |
|        | Karışık           | 650           | 800           | 800            | 500            |
| Yüksek | Eşit              | 900           | 900           | 900            | 900            |
|        | Eşit              | 1050          | 1050          | 1050           | 1050           |
|        | Eşit              | 1200          | 1200          | 1200           | 1200           |
|        | Karışık           | 900           | 1050          | 1050           | 1200           |
|        | Karışık           | 1200          | 1200          | 900            | 1050           |

4 farklı kavşak geometrisinde oluşturulan trafik durumlarına göre sabit zamanlı, optimize edilen sabit zamanlı DGA-ODS, sadece faz optimizasyon modülünün kullanıldığı ÇTA-TKS, Tip-1 BM-TKS ve HTKS yaklaşımlarının simülasyonları gerçekleştirilmiştir. Simülasyonların gerçekleştirilmesinde sistemlerle birlikte çalışan optimizasyon yöntemlerinin kontrol parametrelerinin uygun bir şekilde belirlenmesi gerekmektedir. Algoritmaların iterasyon sayısında 2000 olarak sabit bir değer kullanılmış olup bütün algoritmalar buna göre çalıştırılmıştır. DGA-ODS kontrol sisteminde popülasyon boyutu ( $Pb$ ), çaprazlama oranı ( $CR$ ) ve mutasyon katsayısı ( $F$ ) olarak 3 kontrol parametresine ihtiyaç duyulmaktadır. Mallipeddi ve diğ. [70] yapmış olduğu çalışma sonucunda DGA'nın parametreleri olan NP, CR ve F için optimum aralık ortaya koymuşlardır. Popülasyon boyutu için  $4D$  ile  $10D$  arasında, çaprazlama oranı için 0,9 ile 1 arasında ve mutasyon katsayısı için 0,4 ile 0,95 arasında olması gerektiğini belirtmişlerdir.  $D$ , değişken sayısını ifade etmekte olup, sistem 2, 3 ve 4 fazlı kavşaklarda optimum devre süresini belirleyebilmektedir. Bu nedenle popülasyon boyutu 30 olarak belirlenmiştir. DGA-ODS sisteminde kullanılan kontrol parametreleri Çizelge 7-3'de verilmiştir.

**Çizelge 7-3** Diferansiyel gelişim algoritması kontrol parametreleri

|                            |      |
|----------------------------|------|
| Popülasyon boyutu ( $Pb$ ) | 30   |
| Çaprazlama oranı ( $CR$ )  | 0.90 |
| Mutasyon katsayısı ( $F$ ) | 0.95 |

ÇTA-TKS'de popülasyon boyutu ( $N$ ) ve olasılık anahtarı ( $P$ ) parametreleri olmak üzere 2 farklı kontrol parametresi kullanılmaktadır. Yang 0,05 ile 0,95 arasında olasılık anahtarının ve 10 ile 50 arasında popülasyon boyutunun kullanılabilir olduğunu göstermiş ve farklı değerlerinin performansını incelemiştir.  $N$  için 25 ve  $P$  için 0,8 değerlerinin en uygun çalışma durumları olduğunu ifade etmiştir [46]. ÇTA-TKS'de kullanılan kontrol parametreleri Çizelge 7-4'de verilmiştir.

**Çizelge 7-4** Çiçek tozlaşma algoritması kontrol parametreleri

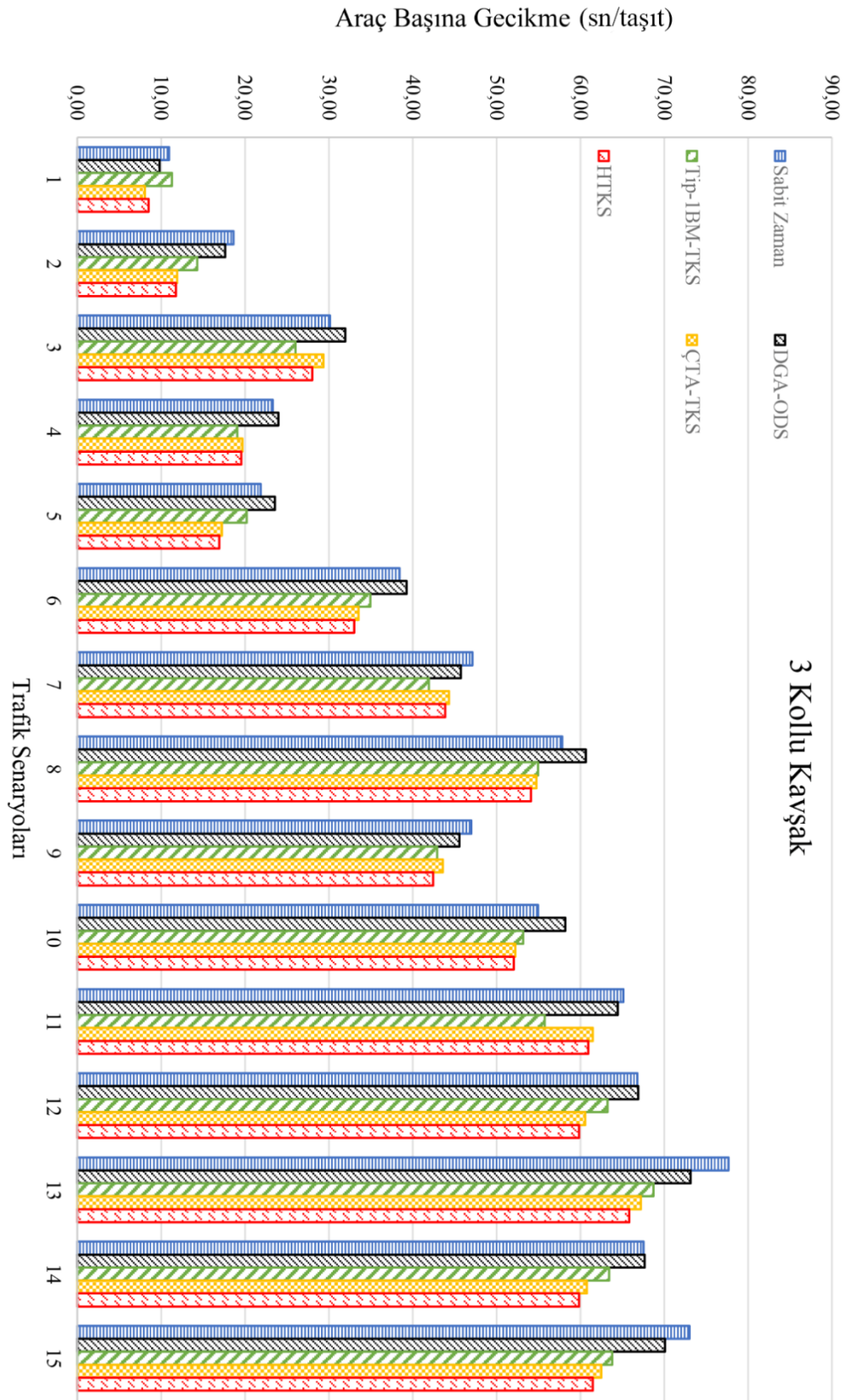
|                            |     |
|----------------------------|-----|
| Popülasyon boyutu ( $N$ )  | 25  |
| Mutasyon katsayısı ( $F$ ) | 0.8 |

HTKS’de popülasyon boyutu ( $N$ ) ve farkındalık olasılığı ( $AP$ ) ve uçuş uzunluğu ( $fl$ ) olmak üzere 3 kontrol parametresi bulunmaktadır. Askarzadeh,  $AP$  parametresinin 0,05 ile 0,3 arasında ve  $fl$  parametresinin 1,5 ile 2,5 arasında olması gerektiğini belirtmiştir. Farklı değerlerin performanslarını karşılaştırarak  $AP$  için 0.05 ve  $fl$  için 2 değerinin en iyi sonuçları ortaya koyduğunu göstermiştir [58]. HTKS sisteminde kullanılan kontrol parametreleri Çizelge 7-5’de verilmiştir.

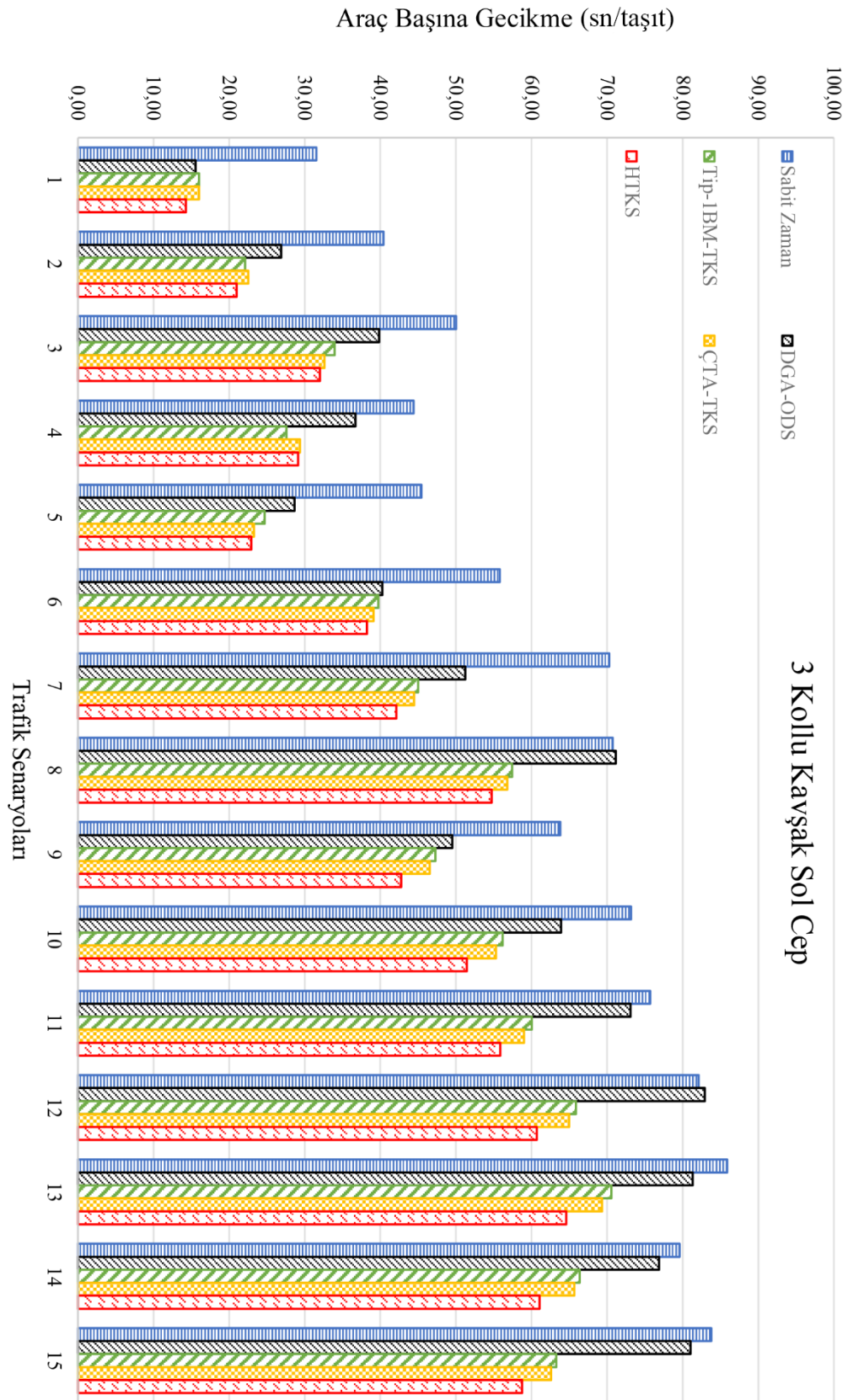
**Çizelge 7-5** Karga arama algoritması kontrol parametreleri

|                                |      |
|--------------------------------|------|
| Popülasyon Boyutu ( $N$ )      | 30   |
| Farkındalık olasılığı ( $AP$ ) | 0.05 |
| Uçuş uzunluğu ( $fl$ )         | 2    |

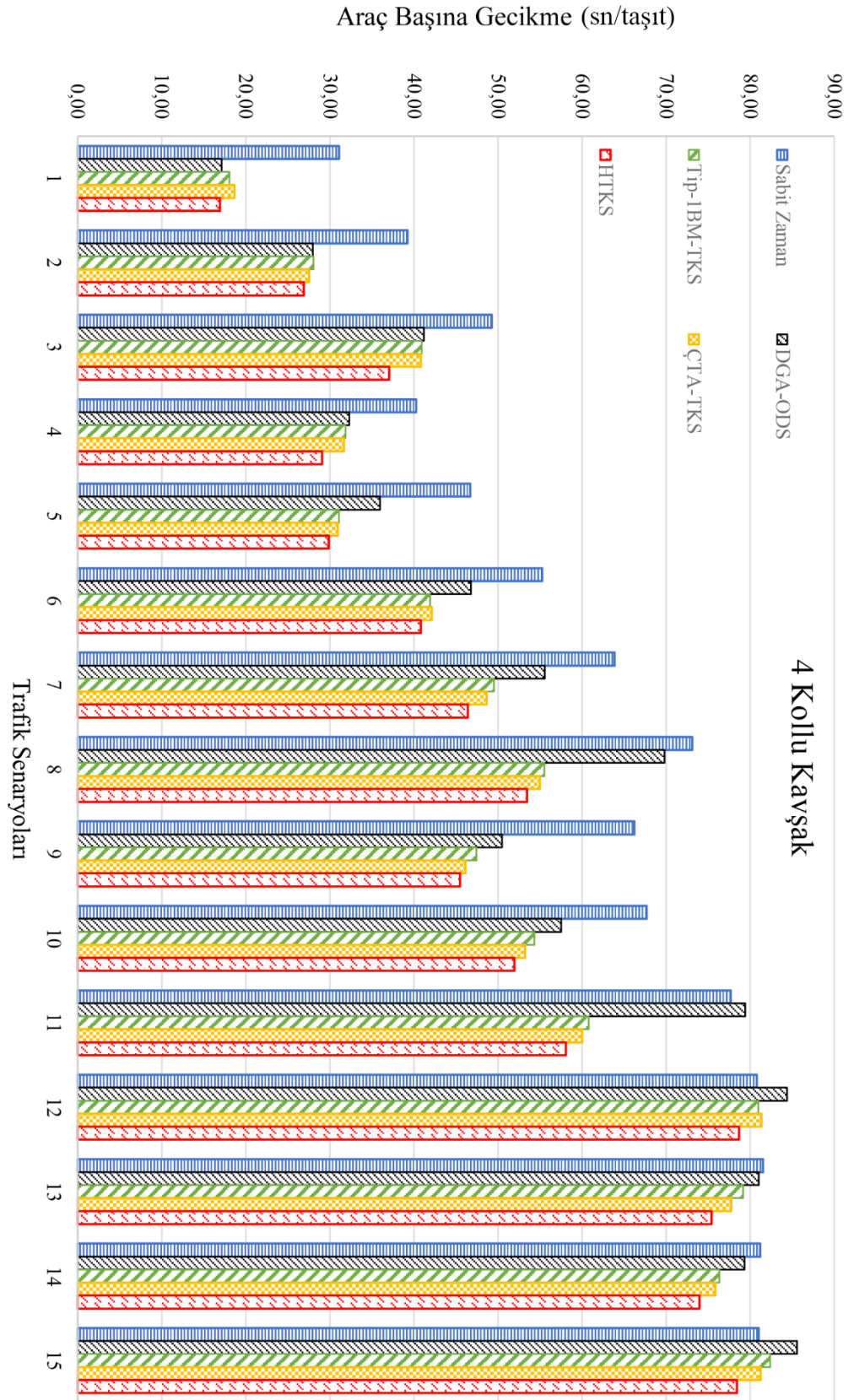
Simülasyon ortamında araç gelişleri çekirdek numarasına göre değişkenlik gösterdiği için gecikmenin belirlenmesinde tek bir çekirdek numarasının kullanılması sonuçların yanıltıcı olmasına neden olabilmektedir. Bu nedenle çalışma kapsamında 20 farklı çekirdek numarası kullanılarak her bir trafik senaryosunun simülasyonu yapılmış ve 20 farklı gecikme değerinin ortalaması alınarak her bir trafik senaryosuna ait gecikme değeri elde edilmiştir. Her bir kavşak geometrisinde yapılan simülasyonlara göre trafik senaryolarına ait ortalama gecikme değerlerinin grafiksel gösterimi Şekil 7-3, 7-4, 7-5 ve 7-6’da verilmiştir.



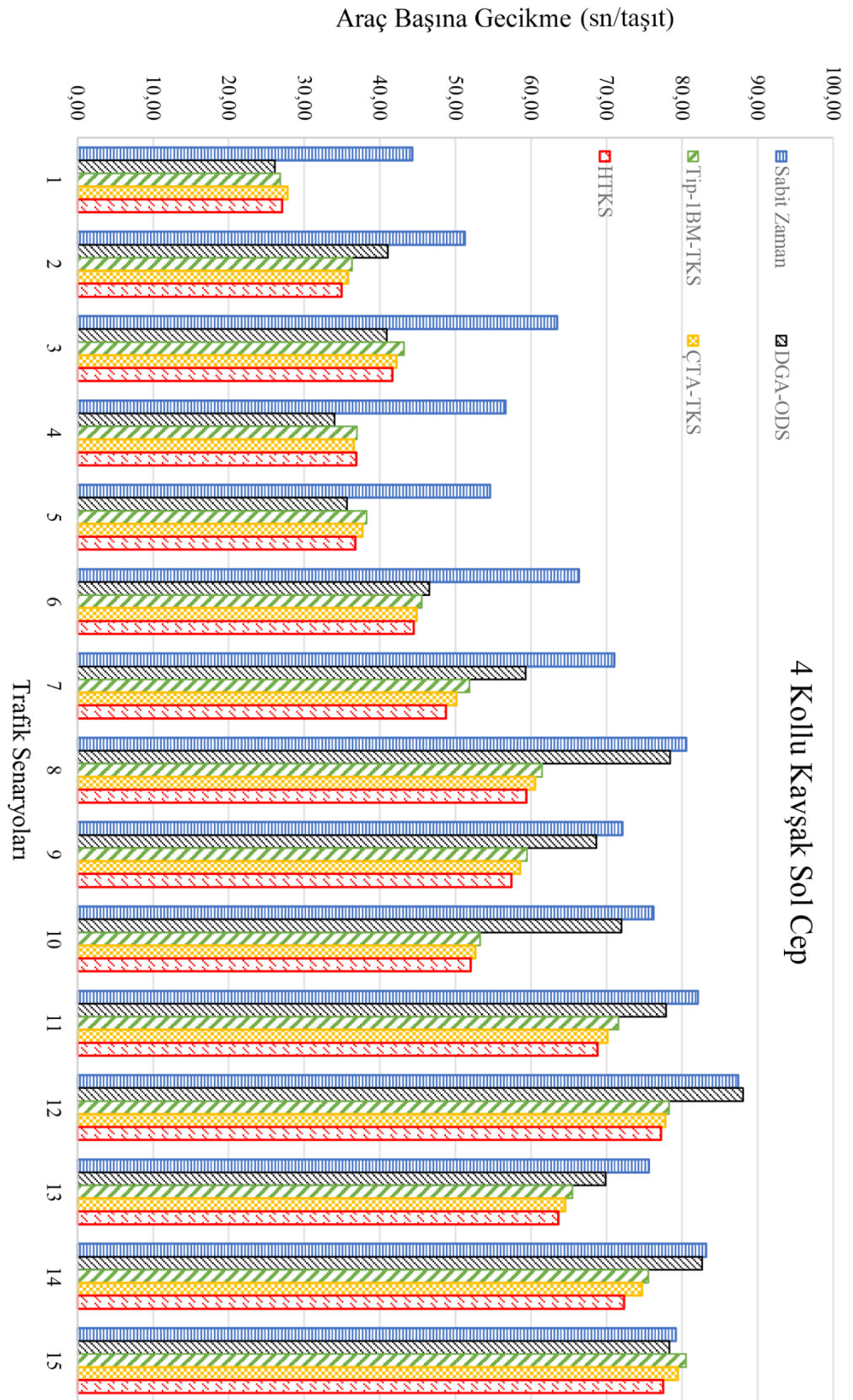
**Şekil 7-3** 3 kollu kavşak için trafik senaryolarına göre gecikme dağılımı



Şekil 7-4 3 kollu kavşak sol cep için trafik senaryolarına göre gecikme dağılımı



Şekil 7-5 4 kollu kavşak için trafik senaryolarına göre gecikme dağılımı



Şekil 7-6 4 kollu kavşak sol cep için trafik senaryolarına göre gecikme dağılımı

Şekil 7-3 -Şekil 7-6'dan de görüldüğü gibi sabit zamanlı kontrol yaklaşımı en yüksek gecikme değerleri ile kavşak kontrolünü gerçekleştirmiştir. DGA-ODS kontrol yaklaşımı önemli ölçüde iyileştirmeye sahip olmuş ve daha düşük gecikme değerleri ile trafik kontrolü sağlamıştır. Tip-1 BM-TKS yaklaşımı ise sabit zamanlı kontrollerden daha iyi performans göstermiş, fakat ÇTA-TKS ve HTKS yaklaşımlarına yakın gecikme değerleri ortaya koymasına rağmen bunların gerisinde kalmıştır. ÇTA-TKS ve HTKS yaklaşımları birbirine çok yakın bir performans göstermekle birlikte HTSK en düşük gecikme değerleri ile kavşak kontrolünü gerçekleştirmiştir. 15 farklı trafik durumuna göre kontrol sistemlerinin birbirleri ile performans karşılaştırılmasında kullanılan OYH değerleri dört farklı kavşak geometrisi için Çizelge 7-6'da verilmiştir.

**Çizelge 7-6** Farklı kavşak geometrilerine göre OYH

| Kavşaklar        | OYH         |             |             |             | DGA-ODS     |
|------------------|-------------|-------------|-------------|-------------|-------------|
|                  | Sabit Zaman | Sabit Zaman | Sabit Zaman | Sabit Zaman |             |
|                  | -----       | -----       | -----       | -----       |             |
|                  | DGA-ODS     | Tip-1BM-TKS | ÇTA-TKS     | HTKS        | Tip-1BM-TKS |
| 3 Kollu          | -0,17       | -9,75       | -12,71      | -17,76      | -9,36       |
| 3 Kollu- Sol Cep | -17,50      | -29,48      | -30,25      | -33,98      | -13,67      |
| 4 Kollu          | -13,00      | -19,28      | -19,95      | -23,19      | -6,70       |
| 4 Kollu- Sol Cep | -16,27      | -22,6       | -23,59      | -24,94      | -6,29       |

| Kavşaklar        | OYH     |         |             |             |         |
|------------------|---------|---------|-------------|-------------|---------|
|                  | DGA-ODS | DGA-ODS | Tip-1BM-TKS | Tip-1BM-TKS | ÇTA-TKS |
|                  | -----   | -----   | -----       | -----       | -----   |
|                  | ÇTA-TKS | HTKS    | ÇTA-TKS     | HTKS        | HTKS    |
| 3 Kollu          | -12,56  | -13,57  | -3,02       | -4,24       | -1,11   |
| 3 Kollu- Sol Cep | -14,6   | -19,23  | -1,03       | -6,29       | -5,32   |
| 4 Kollu          | -7,39   | -11,32  | -0,77       | -4,88       | -4,13   |
| 4 Kollu- Sol Cep | -7,40   | -9,00   | -1,22       | -2,93       | -1,74   |

Şekil 7-3 ve Çizelge 7-6'dan anlaşıldığı gibi üç kollu kavşakta DGA-ODS yaklaşımı sabit zamanlı sistem ile çok yakın bir performans göstermiş olup, gecikme değerini

ancak % 0,17 iyileştirebilmiştir. Tip-1BM-TKS, ÇTA-TKS ve HTKS yaklaşımları ise sırasıyla % 9,75, % 12,71 ve % 17,76 iyileştirme sağlamışlardır. ÇTA-TKS yaklaşımı Tip-1 BM-TKS'ye göre % 3,02 iyileştirme sağlamış olup en iyi performansı gösteren HTKS sırasıyla Tip-1BM-TKS ve ÇTA-TKS'ye göre % 4,24 ve % 1,11 iyileşme sağlayarak kavşak performansını arttırabilmiştir. 3 kollu kavşakta sola dönüş cebinin olması kontrol sistemlerinin performansını etkilemektedir. DGA-ODS, Tip-1 BM-TKS, ÇTA-TKS ve HTKS yaklaşımları sabit zamanlı sisteme göre % 17,5 ile % 33,98 arasında iyileştirme sağlayabilmiştir. En iyi performansı gösteren HTKS yaklaşımı Tip-1 BM-TKS, ÇTA-TKS yaklaşımlarına göre gecikme değerini sırasıyla % 6,29 ve % 5,32 oranında düşürebilmiştir. Tip-1BM-TKS, ÇTA-TKS yaklaşımları birbirine yakın performans göstermekle birlikte % 1,03 iyileştirme sağlaması faz optimizasyonun kavşak performansı üzerindeki etkisini de ortaya koymaktadır.

Dört kollu kavşak geometrilerinde düşük ve orta trafik durumlarında DGA-ODS, Tip-1 BM-TKS, ÇTA-TKS ve HTKS yaklaşımları sabit zamana göre daha düşük gecikme ile kavşağı kontrol ederken yüksek trafik durumlarında performansları ve gecikme değerleri sabit zamanlı kontrol sistemine yaklaşmıştır. Bu durumun temel nedeni ise yüksek trafik durumundan dolayı kavşağın doyuma ulaşması ve kontrol sistemlerinin etkisinin azalmasıdır. Dört kollu kavşaklarda da en iyi performansı HTKS yaklaşımı göstermiş olup kendisine en yakın performansta olan ÇTA-TKS yaklaşımından kavşak geometrilerine % 4,13 ve % 1,74 daha iyi sonuçlar vermiştir. DGA-ODS yaklaşımının sabit zaman'a göre % 13-16 arasında kavşak performansını iyileştirirken, diğer sistemlere göre performansı yaklaşık % 10 daha düşük kalmıştır. Bu çalışma ile farklı yapay zekâ yöntemlerinin ve kontrol yaklaşımlarının uygulanmasıyla elde edilen istatistiksel sonuçlarla sabit zamanlı kontrol yaklaşımına göre daha iyi kavşak performansının sağlanabildiği ortaya konulmuştur. Optimum devre süresinin belirlenmesi performans iyileştirmesi sağlarken, tek başına yeterli düzeye ulaşmamıştır. Süre optimizasyonu ile birlikte faz düzeni ve sırasının optimize edilmesinin kavşak performansı üzerinde önemli bir etkiye sahip olduğu görülmüş ve kavşak performansında yaklaşık % 9 ile % 19 arasında iyileştirme sağlamıştır. Ayrıca etkin bir performans gösteren Tip-1 BM-TKS yaklaşımına göre Tip-2 BM ve algoritmaların kullanılarak geliştirilen hibrid sistemin kavşak performansı üzerinde % 3 ile % 6 arasında iyileşme sağladığı görülmüştür.

ÇTA-TKS ve HTKS sistemlerinin doygunaltı ve doygunüstü trafik akımlarındaki performansları dört farklı kavşak geometrisindeki OYM değerlerine göre incelenmiş ve Çizelge 7-7’de verilmiştir.

**Çizelge 7-7** Doygunaltı ve doygunüstü trafik akımları göre OYH

|                                   | ÇTA-TKS     |         |              | HTKS        |         |              |
|-----------------------------------|-------------|---------|--------------|-------------|---------|--------------|
|                                   | Sabit Zaman | DGA-ODS | Tip-1 BM TKS | Sabit Zaman | DGA-ODS | Tip-1 BM TKS |
| <b>Doygunaltı Trafik Akımları</b> | -24,21      | -10,81  | -1,47        | -26,41      | -13,53  | -4,47        |
| <b>Doygunüstü Trafik Akımları</b> | -11,27      | -9,18   | -1,36        | -14,24      | -12,24  | -4,04        |

Her iki kontrol sisteminin de doygunaltı trafik akımlarında daha iyi performans gösterdiği görülmüştür. ÇTA-TKS ile % 24,21, % 10,81 ve % 1,47 iyileştirmeler sırasıyla sabit zamanlı, DGA-ODS ve Tip-1 BM TKS sistemlerine göre elde edilmişken, HTKS’de bu iyileştirmeler % 26,41, % 13,53 ve % 4,47 olmuştur. Doygunüstü trafik akımlarında sabit zamanlı kontrol sistemine göre kontrol sistemlerinin performansları % 50’ye yakın düşmekte olup, ÇTA-TKS’de % 11.27 ve HTKS’de % 14,24 olmuştur. DGA-ODS ve Tip-1 BM TKS sistemlerine göre ise % 10 civarı bir performans kaybı ile ÇTA-TKS’de % 9,18 ve % 1,36 iyileştirme sağlanmışken, HTKS’de % 12,24 ve % 4,04 iyileştirme sağlanmıştır.

## 8. SONUÇLAR

Trafik mühendisliğinde önemli bir konu olan kavşak kontrolünde temel husus kavşağın hızlı, etkili ve güvenli bir şekilde işletilebilmesidir. Bu nedenle birçok araştırmacının temel amaçları arasında herhangi bir trafik durumunda minimum gecikme, minimum duruş oranı, maksimum kapasite gibi kavşak performans göstergelerinin iyileştirilmesi yer almaktadır. Yapay zekâ teknolojisinin gelişmesi ve yeni yaklaşımların daha iyi performanslar ortaya koyması bu amaca ulaşılabilmede önemli katkılar sağlamaktadır. Sabit zamanlı ya da yapay zekâ yöntemlerinin öncü yaklaşımlarına dayalı kavşak kontrol sistemlerinin yetersiz kaldığı noktalarda yeni yaklaşımların kavşak kontrolüne uygulanması ile mevcut sistemlerin eksiklikleri bertaraf edilebilmekte ve performans iyileştirmesi sağlanmaktadır. Bu çalışmanın temel amacı da bu doğrultuda olup güncel ve etkili algoritmalarından ÇTA ve KAA yaklaşımlarını ve Tip-2 BM yaklaşımını kullanarak kavşak kontrolünde efektif sonuçlar alabilmektir. ÇTA ve KAA yaklaşımlarının ve Tip-2 BM yaklaşımının optimizasyon tekniklerinin performans avantajlarından yararlanılarak geliştirilen kontrol sistemlerinin uygulanabilirliği ortaya konulmuş olup etkin bir performansa sahip olan Tip-1BM-TKS ile karşılaştırılması yapılmıştır. Bu çalışmadan aşağıdaki sonuçlar elde edilmiştir.

1. Farklı trafik durumlarında sabit zamanlı kontrol sistemi yetersiz kalmakta ve her trafik durumuna göre kavşağı optimum şekilde kontrol edememektedir. Bu bağlamda kavşak kontrolünde minimum gecikmeyi amaçlayan bir arama algoritması geliştirilerek her bir trafik durumunun optimum devre süresi tayini sağlanmıştır. Böylece farklı kavşak geometrilerinde önerilen yaklaşımla sabit zamanlı sisteme göre yaklaşık %18 iyileştirme sağlanabilmektedir.
2. Bir kavşağın performansında sinyalizasyondaki devre süresi kadar faz planı ve sırası da önem arz etmektedir. Bu durumda ÇTA yaklaşımı kullanılarak devre süresinin, faz planının ve faz sırasının optimizasyonuna dayalı yeni bir kontrol sistemi geliştirilmiş olup bu sistemin etkinliği gösterilmiştir. Ortaya konulan bu sistemle sabit zamanlı kontrol sistemine göre %30'a kadar iyileştirme sağlanmıştır. Ayrıca, bu sistem etkin bir performansa sahip olan Tip-1BM-TKS

yaklaşımından da % 1 ile % 3 arasında daha iyi sonuçlar elde etmiştir. Böylece ÇTA yaklaşımının kavşak kontrolünde etkin bir sistem olduğu görülmüştür. Bir diğer husus ise değişen trafik durumlarına olabildiğince çabuk adapte olan sistemlerin geliştirilebilmesidir. Tip-1BM-TKS yaklaşımı bu duruma cevap verebilmekte fakat uzun bir optimize süresine ihtiyaç duymaktadır. Önerilen ÇTA-TKS yaklaşımın performansının yanında bir diğer avantajı da çok kısa bir sürede trafik durumlarına adapte olabilmesi ve kavşak kontrolünde etkin kararları hızlıca verebilmesidir.

3. Sinyal sürelerinin her bir faz sonunda tekrar optimize edilerek dinamik hale getirilmesi gecikmenin iyileştirilmesinde katkı sağlamaktadır. KAA ile optimize edilen Tip-2BM yaklaşımı ile geliştirilen sinyal süresi optimizasyon modülü gecikmenin düşürülmesinde etkin bir performans göstermiştir. ÇTA, KAA ve Tip-2BM yaklaşımların birlikte kullanılmasıyla oluşturulan HTKS sistemi en iyi performansı göstermiş olup en düşük gecikme değerleri ile bütün kavşak geometrilerinde kavşak işletimini sağlamıştır. Ayrıca, Tip-1BM-TKS yaklaşımına göre %3 ile %6 arasında iyileştirme sağlanması gerek emisyon ve yakıt tüketiminin azaltılmasına gerekse sinyalizasyondan kaynaklı sürücü memnuniyetsizliklerin minimum seviyeye indirilmesinde etkili olmuştur.
4. Türkiye'deki araç sahipliği nüfusla birlikte her geçen gün artış göstermektedir. Bu durum seyahat taleplerinin artışına ve trafik sıkışıklıklarına neden olmaktadır. Özellikle kavşak geometrilerinin değiştirilmesinin mümkün olmadığı durumlarda etkinliği daha fazla olan trafik kontrol sisteminin kullanımı gerekmektedir. Geliştirilen kontrol yaklaşımının hızlı ve daha yüksek performansa sahip olması sistemin uygulanabilirliğini arttırmaktadır.
5. İzole kavşaklar için geliştirilen bu sistem sonraki çalışmalarda koordine kavşaklara da uygulanarak bir yol ağındaki bütün kavşakların daha yüksek hizmet seviyesinde işletilmesine katkı sağlayacaktır.

## KAYNAKLAR

- [1] Y. Ş. Murat and E. Gedizlioğlu, “Ayrık sinyalizasyon kavşakları için bulanık mantık denetim modeli,” *Tek. Dergi*, vol. 14, no. 67, 2003.
- [2] M. B. Trabia, M. S. Kaseko, and M. Ande, “A two-stage fuzzy logic controller for traffic signals,” *Transp. Res. Part C Emerg. Technol.*, vol. 7, no. 6, pp. 353–367, 1999.
- [3] M. Sarı, Y. Ş. Murat, and M. Kırabalı, “Bulanık modelleme yaklaşımı ve uygulamaları,” *Dumlupınar Üniversitesi Fen Bilim. Enstitüsü Derg.*, no. 009, pp. 77–92, 2005.
- [4] Y. S. Murat and E. Gedizlioglu, “A fuzzy logic multi-phased signal control model for isolated junctions,” *Transp. Res. Part C Emerg. Technol.*, vol. 13, no. 1, pp. 19–36, 2005.
- [5] Y. S. Murat, Z. Cakici, and G. Yaslan, “Use of fuzzy logic traffic signal control approach as dual lane ramp metering model for freeways,” in *Soft Computing in Industrial Applications*, Springer, 2014, pp. 339–349.
- [6] Z. Gacovski, S. Deskovski, and K. Veljanovska, “Fuzzy traffic controller for urban intersection developed in Simulink/SimEvents,” *IFAC Proc. Vol.*, vol. 45, no. 24, pp. 122–125, 2012.
- [7] C. M. Mwangi, S. M. Kang’ethe, and G. N. Nyakoe, “Design and simulation of a fuzzy logic traffic signal controller for a signalized intersection,” in *Scientific Conference Proceedings*, 2012.
- [8] B. Yulianto, “Adaptive traffic signal control for mixed traffic conditions,” *ICETIA*, pp.111-116, 2014.
- [9] H. Ceylan and M. G. H. Bell, “Traffic signal timing optimisation based on genetic algorithm approach, including drivers’ routing,” *Transp. Res. Part B Methodol.*,

vol. 38, no. 4, pp. 329–342, 2004.

- [10] L. Shoufeng, L. Ximin, and D. Shiqiang, “Q-Learning for adaptive traffic signal control based on delay minimization strategy,” in *2008 IEEE International Conference on Networking, Sensing and Control*, 2008, pp. 687–691.
- [11] L. Zhao, X. Peng, L. Li, and Z. Li, “A fast signal timing algorithm for individual oversaturated intersections,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 280–283, 2010.
- [12] O. Baskan and S. Haldenbilen, “Ant colony optimization approach for optimizing traffic signal timings,” in *Ant colony optimization-methods and applications*, IntechOpen, 2011.
- [13] R. Putha, L. Quadrifoglio, and E. Zechman, “Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions,” *Comput. Civ. Infrastruct. Eng.*, vol. 27, no. 1, pp. 14–28, 2012.
- [14] J. He and Z. Hou, “Ant colony algorithm for traffic signal timing optimization,” *Adv. Eng. Softw.*, vol. 43, no. 1, pp. 14–18, 2012.
- [15] Q. Liu and J. Xu, “Traffic signal timing optimization for isolated intersections based on differential evolution bacteria foraging algorithm,” *Procedia-Social Behav. Sci.*, vol. 43, pp. 210–215, 2012.
- [16] M. Dell’Orco, O. Baskan, and M. Marinelli, “A harmony search algorithm approach for optimizing traffic signal timings,” *Promet-Traffic&Transportation*, vol. 25, no. 4, pp. 349–358, 2013.
- [17] M. Dell’Orco, O. Başkan, and M. Marinelli, “Artificial bee colony-based algorithm for optimising traffic signal timings,” in *Soft Computing in Industrial Applications*, Springer, 2014, pp. 327–337.
- [18] S. Araghi, A. Khosravi, and D. Creighton, “Intelligent cuckoo search optimized

- traffic signal controllers for multi-intersection network,” *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4422–4431, 2015.
- [19] Y. Zhao, W. Fang, and C. Qing, “A PSO based signal timing optimization approach of phase combination,” in *2016 International Conference on Civil, Transportation and Environment*, 2016.
  - [20] N. R. Sabar, E. Chung, T. Tsubota, and P. E. M. de Almeida, “A memetic algorithm for real world multi-intersection traffic signal optimisation problems,” *Eng. Appl. Artif. Intell.*, vol. 63, pp. 45–53, 2017.
  - [21] E. Eriskin, S. Karahancer, S. Terzi, and M. Saltan, “Optimization of traffic signal timing at oversaturated intersections using elimination pairing system,” *Procedia Eng.*, vol. 187, pp. 295–300, 2017.
  - [22] J. Chen and L. Xu, “Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm,” in *2006 9th International Conference on Control, Automation, Robotics and Vision*, 2006, pp. 1–7.
  - [23] P. G. Balaji and D. Srinivasan, “Distributed multi-agent type-2 fuzzy architecture for urban traffic signal control,” in *2009 IEEE International Conference on Fuzzy Systems*, 2009, pp. 1627–1632.
  - [24] B. Sabetghadam, F. Shabaninia, M. Vaziri, and S. Vadhava, “Type-2 fuzzy multiagent traffic signal control,” in *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, 2012, pp. 509–512.
  - [25] E. Dogan, “Optimize edilmiş bulanık mantık yöntemi ile izol sinyalize kavşak kontrolü,” *Kırıkkale Üniversitesi, Fen Bilim. Enstitüsü, Kırıkkale*, 2014.
  - [26] Y. Bi, D. Srinivasan, X. Lu, Z. Sun, and W. Zeng, “Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization,” *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7338–7349, 2014.
  - [27] S. M. Odeh, A. M. Mora, M. N. Moreno, and J. J. Merelo, “A hybrid fuzzy genetic

- algorithm for an adaptive traffic signal system,” *Adv. Fuzzy Syst.*, vol. 2015, p. 11, 2015.
- [28] D. Yu, X. Tian, X. Xing, and S. Gao, “Signal timing optimization based on fuzzy compromise programming for isolated signalized intersection,” *Math. Probl. Eng.*, vol. 2016, 2016.
- [29] E. Doğan and A. P. Akgüngör, “Optimizing a fuzzy logic traffic signal controller via the differential evolution algorithm under different traffic scenarios,” *Simulation*, vol. 92, no. 11, pp. 1013–1023, 2016.
- [30] S. Araghi, A. Khosravi, D. Creighton, and S. Nahavandi, “Influence of meta-heuristic optimization on the performance of adaptive interval type2-fuzzy traffic signal controllers,” *Expert Syst. Appl.*, vol. 71, pp. 493–503, 2017.
- [31] M. H. Khooban, N. Vafamand, A. Liaghat, and T. Dragicevic, “An optimal general type-2 fuzzy controller for Urban Traffic Network,” *ISA Trans.*, vol. 66, pp. 335–343, 2017.
- [32] K. Chatterjee, A. De, and F. T. S. Chan, “Real time traffic delay optimization using shadowed type-2 fuzzy rule base,” *Appl. Soft Comput.*, vol. 74, pp. 226–241, 2019.
- [33] N. Yayla, *Karayolu mühendisliği*. Birsen Yayınevi, 2011.
- [34] M. Saplıoğlu and M. Karaşahin, “Şehiriçi kontrolsüz eşdüzey kavşak kazalarını etkileyen unsurların değerlendirilmesi,” *Uluslararası Teknol. Bilim. Derg.*, vol. 2, no. 2, pp. 26–49.
- [35] E. Gedizlioğlu, “Trafik yönetimi ders notları.” İstanbul, 2002.
- [36] KGM (Karayolları Genel Müdürlüğü), *Trafik İşaretleri El kitabı*. 2005.
- [37] Y. Uysal, “Kavşak ve Kavşaklarda Sinyalizasyon,” *Süleyman Demirel Üniversitesi, Fen Bilim. Enstitüsü, Isparta*, 2001.

- [38] M. K. Çubuk, “Köprülü kavşak tiplerinin avantaj ve dezavantajları”, 2. *Ulaşım ve Trafik Kongresi*, Ankara-Türkiye, 1999.
- [39] Y. Ş. Murat, “Kentsel ulaşım çerçevesinde sinyalize kavşak etüdleri,” in 2. *Ulusal Kentsel Altyapı Sempozyumu*, 1999, pp. 75–86.
- [40] Y. Ş. Murat, “Denizli şehiriçi kavşaklarındaki trafik akımlarının bilgisayarla incelenmesi.” *Yüksek Lisans Tezi. Denizli: Pamukkale Üniversitesi*, 1996.
- [41] TRB, *Highway capacity manual 2000*. 2000.
- [42] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic engineering*. Pearson/Prentice Hall, 2004.
- [43] Y. S. Murat, “Trafik mühendisliği ders notları,” Denizli, 2012.
- [44] B. J. Kemp, C. Kim, J. J. Williams, A. Ganin, and V. J. Lowe, “National electrical manufacturers association (NEMA),” *NEMA NU*, vol. 2, pp. 1960–1967, 2001.
- [45] F. V. Webster, “Traffic signal settings”, London : H.M.S.O., 1958.
- [46] X.-S. Yang, “Flower pollination algorithm for global optimization,” in *International conference on unconventional computing and natural computation*, 2012, pp. 240–249.
- [47] S. Pant, A. Kumar, and M. Ram, “Flower pollination algorithm development: a state of art review,” *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. 2, pp. 1858–1866, 2017.
- [48] A. E. Kayabekir, G. Bekdaş, S. M. Nigdeli, and X.-S. Yang, “A comprehensive review of the flower pollination algorithm for solving engineering problems,” in *Nature-Inspired Algorithms and Applied Optimization*, Springer, 2018, pp. 171–188.
- [49] M. Sharawi, E. Emary, I. A. Saroit, and H. El-Mahdy, “Flower pollination optimization algorithm for wireless sensor network lifetime global optimization,”

*Int. J. Soft Comput. Eng.*, vol. 4, no. 3, pp. 54–59, 2014.

- [50] N. Sakib, M. W. U. Kabir, M. Subbir, and S. Alam, “A comparative study of flower pollination algorithm and bat algorithm on continuous optimization problems,” *Int. J. Soft Comput. Eng.*, vol. 4, no. 3, pp. 13–19, 2014.
- [51] S. Łukasik and P. A. Kowalski, “Study of flower pollination algorithm for continuous optimization,” in *Intelligent Systems’ 2014*, Springer, 2015, pp. 451–459.
- [52] I. El-henawy and M. Ismail, “An improved chaotic flower pollination algorithm for solving large integer programming problems,” *Int. J. Digit. Content Technol. its Appl.*, vol. 8, no. 3, pp. 72–81, 2014.
- [53] O. Abdel-Raouf, I. El-Henawy, and M. Abdel-Baset, “A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 3, p. 38, 2014.
- [54] E. Nabil, “A modified flower pollination algorithm for global optimization,” *Expert Syst. Appl.*, vol. 57, pp. 192–203, 2016.
- [55] S. Kalra and S. Arora, “Firefly algorithm hybridized with flower pollination algorithm for multimodal functions,” in *Proceedings of the international congress on information and communication technology*, 2016, pp. 207–219.
- [56] S. Arora and P. Anand, “Binary butterfly optimization approaches for feature selection,” *Expert Syst. Appl.*, vol. 116, pp. 147–160, 2019.
- [57] M. Abdel-Basset and L. A. Shawky, “Flower pollination algorithm: a comprehensive review,” *Artif. Intell. Rev.*, pp. 1–25, 2018.
- [58] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm,” *Comput. Struct.*, vol. 169, pp. 1–12, 2016.

- [59] B. Zolghadr-Asli, O. Bozorg-Haddad, and X. Chu, "Crow search algorithm (CSA)," in *Advanced Optimization by Nature-Inspired Algorithms*, Springer, 2018, pp. 143–149.
- [60] A. Y. Abdelaziz and A. Fathy, "A novel approach based on crow search algorithm for optimal selection of conductor size in radial distribution networks," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 2, pp. 391–402, Apr. 2017.
- [61] F. Mohammadi and H. Abdi, "A modified crow search algorithm (MCSA) for solving economic load dispatch problem," *Appl. Soft Comput.*, vol. 71, pp. 51–65, 2018.
- [62] A. Javidi, E. Salajegheh, and J. Salajegheh, "Enhanced crow search algorithm for optimum design of structures," *Appl. Soft Comput.*, 2019.
- [63] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [64] E. Korkmaz and A. P. Akgüngör, "Delay estimation models for signalized intersections using differential evolution algorithm," *J. Eng. Res.*, vol. 5, no. 3, 2017.
- [65] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci. (Ny)*, vol. 8, no. 3, pp. 199–249, 1975.
- [66] M. Karakose and S. Makinist, "İnsan kaldırma hareketinin analizi için tip- 2 bulanık sistem yaklaşımı," *Türkiye Bilişim Vakfı Bilgi. Bilim. ve Mühendisliği Derg.*, vol. 6, no. 1, 2013.
- [67] C. Ulu, "Granüler tip-2 bulanık yapılar kullanılarak sistemlerin modellenmesi ve kontrolü." Fen Bilimleri Enstitüsü, 2013.
- [68] A. Taskin and T. Kumbasar, "An open source Matlab/Simulink toolbox for interval type-2 fuzzy logic systems," in *2015 IEEE Symposium Series on*

*Computational Intelligence*, 2015, pp. 1561–1568.

- [69] A. Halati, H. Lieu, and S. Walker, “CORSIM-corridor traffic simulation model,” in *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities* Urban Transportation Division, ASCE; Highway Division, ASCE; Federal Highway Administration, USDOT; and National Highway Traffic Safety Administration, US, 1997.
- [70] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, “Differential evolution algorithm with ensemble of parameters and mutation strategies,” *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, 2011.

## 9. EKLER

### EK. 1. ARAÇ DİNAMİK MODÜLÜ MATLAB KODU

```
if time > 0
    for k=1:nl
        if sol_cep_logic(1,k) == 1
            s(1:end,end,k) = 20;
        end

        n(k)=length(find(s(:,1,k)));
        for i1=1:n(k)

            if g(i1,5,k) == 3 && i1 > 1
                g(i1,1:4,k) = g(i1-1,1:4,k);

            elseif g(i1,5,k) == 3 && i1==1
                g(i1,1:4,k) = [999 999 999 999];
                d(i1,1:4,k) = [0 0 0 0];

            elseif g(i1,5,k) == 4
                g(i1,2,k) = 0;

            else

                enter1=0;
                if i1==1 && g(1,1,k)<0
                    enter1=1;

                elseif g(i1,1,k)<0 && g((i1-1),1,k) >= 0
                    enter1=1;

                elseif i1>=2
                    if g((i1-1),1,k)<0 && g(i1,1,k)<0
                        enter1=2;

                    end
                end

            if enter1==1 % Sisteme girişte önünde araç yoksa veya tek kendisi varsa

                if yesil(k)==1&&g(i1,1,k)<0

                    if g(i1,2,k)>=Vf(k)
                        g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik; %yeni1 koorninat hesabı
```

```

elseif g(i1,2,k)<Vf(k)
    g = acceleration_func (g,s,i1,k,acceleration_file);
    ah = g(i1,3,k);
    g(i1,2,k) = g(i1,2,k) + ah*aralik;
    g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
    d(i1,3,k)=d(i1,3,k)+aralik*(1-(g(i1,2,k)/Vf(k))); %hızlanma
                                                    gecikmesi

elseif g(i1,2,k) == 0
    zc(k) = zc(k) + aralik;

    if zc(k) >= kalkis_kaybi
        g = acceleration_func (g,s,i1,k,acceleration_file);
        ah = g(i1,3,k);
        g(i1,2,k) = g(i1,2,k) + ah*aralik;
        g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
        d(i1,3,k)=d(i1,3,k)+aralik*(1-(g(i1,2,k)/Vf(k))); %hızlanma
                                                    gecikmesi
    end
end

elseif (sari(k)==1 && (g(i1,2,k)*sari_t) >= (abs(g(i1,1,k))+30)) &&
    g(i1,1,k) < 0
    g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;

else %kirmizi ise...

    if abs(g(i1,1,k)) < 45
        g(i1,3,k) = -((g(i1,2,k)^2)/(2*(abs(g(i1,1,k))))); % yavasla
                                                    ivmesi
        g(i1,2,k) = g(i1,2,k) + g(i1,3,k) * aralik;
        if (g(i1,2,k)>=0&&g(i1,2,k)<0.5); %arac durduysa..
            (g(i1,2,k)>=0&&g(i1,2,k)<1);
            g(i1,2,k) = 0;
            d(i1,2,k)=d(i1,2,k)+ aralik; %durma gecikmesi
            zc(k) = 0;

        else

            d(i1,1,k)=d(i1,1,k)+aralik*(1-(g(i1,2,k)/Vf(k))); %yavaşlama
                                                    gecikmesi
        end
    end

    if g(i1,2,k)<0
        g(i1,2,k)=0;
        zc(k) = 0;

    end
end

```

```

        g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;

    end

elseif enter1==2

    % onunde arac varsa

    s(i1,3,k)=((g(i1,2,k)*3.6)/2.5)+1.5;                % 'u_guv' hesaplandi

    g(i1,4,k)=(abs(g(i1,1,k)-g(i1-1,1,k)))-((s(i1,2,k)+s(i1-1,2,k))/2);
                                                    % mevcut U hesaplandi

    if g(i1,4,k) >= s(i1,3,k) % U mesafesi yeterli.
        if yesil(k) == 1

            if g(i1,2,k)<Vf(k) % serit hiz limitinden düşükse

                if g(i1,2,k) == 0
                    if g(i1-1,2,k) >= 2.8 % 10 km/saat den büyük mu?
                        g = acceleration_func (g,s,i1,k,acceleration_file);
                        ah = g(i1,3,k);
                        g(i1,2,k) = g(i1,2,k) + ah*aralik;
                        g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
                        d(i1,3,k)=d(i1,3,k)+aralik*(1-(g(i1,2,k)/Vf(k)));
                                                                %hızlanma gecikmesi

                    elseif g(i1-1,2,k) < 2.8
                        g(i1,2,k) = 0;
                        d(i1,2,k)=d(i1,2,k)+aralik; %durma gecikmesi
                    end

                else
                    g = acceleration_func (g,s,i1,k,acceleration_file);
                    ah = g(i1,3,k);
                    g(i1,2,k) = g(i1,2,k) + ah*aralik;
                end

                g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
                d(i1,3,k)=d(i1,3,k)+aralik*(1-(g(i1,2,k)/Vf(k)));    %hızlanma
                                                                gecikmesi

            else
                g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;

            end

        elseif (sari(k)==1 && (g(i1,2,k)*sari_t)>=(abs(g(i1,1,k)+30)))

```

```

g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;

else    %kirimizi ise...
    if g(i1,2,k) < Vf(k) && g(i1,4,k) > s(i1,3,k)
        g = acceleration_func (g,s,i1,k,acceleration_file);
        ah = g(i1,3,k);
        g(i1,2,k) = g(i1,2,k) + ah*aralik;
        g(i1,2,k) = g(i1,2,k) + ah*aralik;
    end

    d(i1,3,k)=d(i1,3,k)+aralik*(1-(g(i1,2,k)/Vf(k)));    %hızlanma
                                                         gecikmesi

    g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
end

elseif g(i1,4,k) <= s(i1,3,k)    % U mesafesi asılmış hiz azaltılacak
    if g(i1,2,k)>0

        if g(i1,4,k) > 3
            g(i1,3,k)=-(abs((g(i1,2,k)^2)-(g(i1,2,k)^2))/(abs(2*((g(i1,4,k)-3))))));

        else
            g(i1,3,k)=-(abs((g(i1,2,k)^2)-(g(i1,2,k)^2))/(abs(2*((g(i1,4,k))))));

        end

    elseif g(i1,2,k) == 0
        g(i1,3,k) = 0;

    end

    g(i1,2,k)=g(i1,2,k)+g(i1,3,k)*aralik;

    if g(i1,2,k)<0
        g(i1,2,k)=0;
    end

    if (g(i1,2,k)>=0)&&(g(i1,2,k)<0.5);
        g(i1,2,k) = 0;
        d(i1,2,k)=d(i1,2,k)+aralik*(1-(g(i1,2,k)/Vf(k)));    %durma
                                                         gecikmesi

    else
        d(i1,1,k)=d(i1,1,k)+aralik*(1-(g(i1,2,k)/Vf(k)));    %yavaşlama
                                                         gecikmesi

    end

    g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;

end

```

```

end

%% Cebe Giriş Hazırlık
if s(i1,4,k) == 1 && g(i1,2,k) > 0 && g(i1,1,k) >=
    -(sol_cep_uzunluk(1,k)+50) && g(i1,1,k) <
    -(sol_cep_uzunluk(1,k))
% cebe girmek için yaklaşan araç cep başlangıcına 40 m mesafede karar verecek

    if g(i1,2,k) > 5
        g(i1,3,k) = (5 - g(i1,2,k))/80; % yavaşlama ivmesi % 50 m
                                         içerisinde 5 m/sn hız düşecek

        g(i1,2,k) = g(i1,2,k) + g(i1,3,k)*aralik;
        g(i1,1,k) = g(i1,1,k) + g(i1,2,k)*aralik;

    else
        g(i1,1,k) = g(i1,1,k) + g(i1,2,k)*aralik;
    end
end

%% Cebe giriş
if s(i1,4,k) == 1
    if n(k+1) > 0 && g(i1,1,k) >= -(sol_cep_uzunluk(1,k)+10) &&
        -(sol_cep_uzunluk(1,k)) > (g(n(k+1),1,k+1)-s(i1,2,k)-10)
        % cepte yeterli yer yoksa

        g(i1,3,k) = -(g(i1,2,k)/20);
        % 10 m içerisinde "0 m/sn" ye düşecek ivme

        g(i1,2,k) = g(i1,2,k) + g(i1,3,k)*aralik;

        if g(i1,2,k) < 0
            g(i1,2,k) = 0;
        end

        g(i1,1,k) = g(i1,1,k) + g(i1,2,k)*aralik;

    elseif (n(k+1)==0 && g(i1,1,k) > -(sol_cep_uzunluk(1,k))) || ((g(i1,1,k) >=
        -(sol_cep_uzunluk(1,k))&& n(k+1)>0 && -
        (sol_cep_uzunluk(1,k)) < (g(n(k+1),1,k+1)-s(i1,2,k)-10)))
        % cepte yeterli yer varsa

        AracGiren(k+1) = AracGiren(k+1) + 1;

        g(n(k+1)+1,1,k+1) = g(i1,1,k); % Koordonat Aktarimi.
        g(n(k+1)+1,2,k+1) = g(i1,2,k); % Hiz Aktarimi.

```

```

g(n(k+1)+1,3,k+1) = g(i1,3,k); % Ivme aktarimi.
g(n(k+1)+1,6,k+1) = g(i1,6,k); % Uzunluk Aktarimi.
s(n(k+1)+1,1,k+1) = s(i1,1,k); % İlk hiz Aktarimi.
s(n(k+1)+1,2,k+1) = s(i1,2,k); % Uzunluk Aktarimi.
d(n(k+1)+1,1:3,k+1) = d(i1,1:3,k); % Gecikmelerin Aktarimi.

if g(i1,7,k)==1
    Dec_K(k)= Dec_K(k)-g(i1,6,k);
    g(i1,7,k)=0;

end

g(n(k+1)+1,7,k+1) = g(i1,7,k);
    % sol cebe giren aracın kuyrukta olmadığını
gösterir.

%% Güvenli duruş mesafesi yeniden hesabi.

s(n(k+1)+1,3,k+1) = ((g(n(k+1)+1,2,k+1)*3.6)/2.5)+1.5;

%% iki arac arasi mesafe degistiginden yeniden hesabi.

if n(k+1) > 1
g(n(k+1)+1,4,k+1) = abs((g(n(k+1)+1,1,k+1)-g(n(k+1)+1,1,k+1)))-
((s(n(k+1)+1,2,k)+s(n(k+1)+1,2,k))/2);
end

%% Diger seritten silinmeleri.

if i1 == 1
    g(i1,1,k) = 100;
    g(i1,2,k) = 200;
    d(i1,1:end,k) = zeros(1,5);

elseif i1 > 1
    if g(i1,1,k) < 0
        g(i1,1:end,k) = g(i1-1,1:end,k);
        g(i1,5,k) = 3;
        s(i1,1:end,k) = s(i1-1,1:end,k);
        s(i1,4,k) = 3;
        d(i1,1:end,k) = zeros(1,5);

    end
end
end
end

%%% SL gecilmis

```

```

if g(i1,1,k)>=0

    if i1==1 && g(i1,2,k) < Vf(k)
        g = acceleration_func (g,s,i1,k,acceleration_file);

        if s(i1,end,k) == 10 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
            ah = g(i1,3,k)/1.2;

        elseif s(i1,end,k) == 20 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
            ah = g(i1,3,k)/1.1;

        else
            ah = g(i1,3,k);
        end

        g(i1,2,k) = g(i1,2,k) + ah*aralik;
        g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
        d(i1,4,k)=d(i1,4,k)+aralik*(1-(g(i1,2,k)/Vf(k)));
        %Hızlanma gecikmesi (çizgiden sonra)

    elseif g(i1,2,k) < Vf(k) && i1>1
        s(i1,3,k)=((g(i1,2,k)*3.6)/2.5)+1.5;          % 'u_guv' hesaplandı
        g(i1,4,k)=(abs(g(i1,1,k)-g(i1-1,1,k)))-((s(i1,2,k)+s(i1-1,2,k))/2);

        if g(i1,4,k)>=s(i1,3,k)          %% Guvenik mesafesi yeterli ise
            if g(i1,2,k)==0&&g(i1-1,2,k)>0 %
                g = acceleration_func (g,s,i1,k,acceleration_file);

                if s(i1,end,k) == 10 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
                    ah = g(i1,3,k)/1.2;
                elseif s(i1,end,k) == 20 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
                    ah = g(i1,3,k)/1.1;
                else
                    ah = g(i1,3,k);
                end

                g(i1,2,k) = g(i1,2,k) + ah*aralik;
                g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
                d(i1,4,k)=d(i1,4,k)+aralik*(1-(g(i1,2,k)/Vf(k)));
                %Hızlanma gecikmesi (çizgiden sonra)

            elseif g(i1,2,k)<Vf(k)          % serit hiz limitinden dusukse
                g = acceleration_func (g,s,i1,k,acceleration_file);

                if s(i1,end,k) == 10 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
                    ah = g(i1,3,k)/1.2;

```

```

elseif s(i1,end,k) == 20 && (g(i1,1,k) < 30 && g(i1,1,k) > 1 )
    ah = g(i1,3,k)/1.1;
else
    ah = g(i1,3,k);
end
g(i1,2,k) = g(i1,2,k) + ah*aralik;

if g(i1,2,k)>Vf(k)
    g(i1,2,k)=Vf(k);
end

g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
end

elseif g(i1,4,k)<s(i1,3,k)          % U mesafesi asilmis hiz azaltılacak

    if g(i1,2,k)>0
        if g(i1,4,k)>3
            g(i1,3,k)=-((abs((g(i1,2,k)^2)-(g(i1-1,2,k)^2)))/(abs(2*((g(i1,4,k)-3)))));
        else
            g(i1,3,k)=-((abs((g(i1,2,k)^2)-(g(i1-1,2,k)^2)))/(abs(2*((g(i1,4,k))))));
        end
    end

    g(i1,2,k)=g(i1,2,k)+g(i1,3,k)*aralik;

    if g(i1,2,k)<0
        g(i1,2,k)=0;
    end

    if (g(i1,2,k)>=0)&&(g(i1,2,k)<0.5);
        g(i1,2,k) = 0;
        d(i1,2,k)=d(i1,2,k)+aralik*(1-(g(i1,2,k)/Vf(k)));
        %durma gecikmesi
    else

        d(i1,1,k)=d(i1,1,k)+aralik*(1-(g(i1,2,k)/Vf(k))); %yavaşlama gecikmesi

    end

    g(i1,1,k)=g(i1,1,k)+g(i1,2,k)*aralik;
end
end
end
end
end
end

```

```

%% Silinme islemi

if g(1,1,k)>0 && g(1,2,k)>=(Vf(k));

    Dy(k)=Dy(k)+(d(1,1,k)); %toplam yavaşlama Gecikmesi
    Ds(k)=Ds(k)+(d(1,2,k)); %toplam durma gecikmesi
    Da1(k)=Da1(k)+(d(1,3,k)); % toplam çizgiden önceki hızlanma gecikmesi
    Da2(k)=Da2(k)+(d(1,4,k)); % toplam çizgiden sonraki hızlanma gecikmesi

    g(1:end,1:end,k) = [g(2:end,1:end,k);0 0 0 0 0 0];
    s(1:end,1:end,k) = [s(2:end,1:end,k);0 0 0 0];
    d(1:end,1:end,k) = [d(2:end,1:end,k);0 0 0 0 0 ];
end

inx_cep = (sol_cep_logic(k)==1);
inx_komsu = (sol_cep_logic(k)==2);

if inx_cep ~= 1 && inx_komsu ~= 1

    Dec_N1(k)=length(find(g(:,1,k)>=Dec_yak_L(k)));
    zero(k)=length(find(g(:,1,k)==0));
    SL1(k)=length(find(g(:,1,k)>0));
    % stop çizgisini gecmis olan arac
sayisi.

    Dec1_N(k)=Dec_N1(k)-SL1(k)-zero(k);

    % detektor 1 ile stop çizgisi arasindaki arac
sayisi.

    %kuyruk uzunluğu (sağ şerit veya sola dönüş olmayan şeritler)

    kuyruktaki_NV =find(g(:,1,k)>=Dec_yak_L(k) & g(:,1,k)<0 &
g(:,2,k)<=2.78);
    klgt=length(kuyruktaki_NV);

    if klgt>0
    for u=1:klgt
        if g(kuyruktaki_NV(u),7,k)==0
            Dec_K(k)= Dec_K(k)+g(kuyruktaki_NV(u),6,k);
            g(kuyruktaki_NV(u),7,k)=1;
        end
    end
end
end

```

```

kuyruktaki_NV = find(g(:,1,k) >= Dec_yak_L(k) & g(:,1,k) < 0 & g(:,2,k) >= 2.78
& g(:,7,k) == 1);
klgth = length(kuyruktaki_NV);

if klgth > 0
for u = 1:klgth
Dec_K(k) = Dec_K(k) - g(kuyruktaki_NV(u), 6, k);
g(kuyruktaki_NV(u), 7, k) = 0;
if Dec_K(k) < 0.2
Dec_K(k) = 0;
end
end
end

kuyruktaki_NV = find(g(:,1,k) > 0 & g(:,7,k) == 1);
klgth = length(kuyruktaki_NV);

if klgth > 0
for u = 1:klgth
Dec_K(k) = Dec_K(k) - g(kuyruktaki_NV(u), 6, k);
g(kuyruktaki_NV(u), 7, k) = 2;
end
end

if nf == 4
if time == queuelenght(1, end)
satir = find(queuelenght(1, :) == Qlenght);
kuyrukuzunlugu(satir, k) = Dec_K(1, k);

end

elseif nf == 3
if time == queuelenght(1, end)
satir = find(queuelenght(1, :) == Qlenght);
kuyrukuzunlugu(satir, k) = Dec_K(1, k);

end

else
if time == queuelenght(1, end)
satir = find(queuelenght(1, :) == Qlenght);
kuyrukuzunlugu(satir, k) = Dec_K(1, k);
end
end

elseif inx_cep == 1

```

```

Dec_N1(k)=length(find(g(:,1,k)>=Dec_yak_L(k)));
zero(k)=length(find(g(:,1,k)==0));
SL1(k)=length(find(g(:,1,k)>0));
                                % stop çizgisini gecmis olan arac
sayisi.

Dec1_N(k)=Dec_N1(k)-SL1(k)-zero(k);
                                % detektor 1 ile stop çizgisi arasindaki arac
sayisi.

%kuyruk uzunluğu (sol cep)

kuyruktaki_NV =find(g(:,1,k)>= -sol_cep_uzunluk(k-1) & g(:,1,k)<0 &
                                g(:,2,k)<=2.78);

klgth=length(kuyruktaki_NV);

if klgth>0
for u=1:klgth
    if g(kuyruktaki_NV(u),7,k)==0
        Dec_K(k)= Dec_K(k)+g(kuyruktaki_NV(u),6,k);
        g(kuyruktaki_NV(u),7,k)=1;
    end
end
end

kuyruktaki_NV =find(g(:,1,k)>= -sol_cep_uzunluk(k-1)& g(:,1,k)<0 &
                                g(:,2,k)>=2.78 & g(:,7,k)==1);

klgth=length(kuyruktaki_NV);

if klgth>0
for u=1:klgth
    Dec_K(k)= Dec_K(k)-g(kuyruktaki_NV(u),6,k);
    g(kuyruktaki_NV(u),7,k)=0;
    if Dec_K(k)<0.2
        Dec_K(k)= 0;
    end
end
end

kuyruktaki_NV =find(g(:,1,k)>0 & g(:,7,k)==1);

klgth=length(kuyruktaki_NV);
if klgth>0
for u=1:klgth
    Dec_K(k)= Dec_K(k)-g(kuyruktaki_NV(u),6,k);
    g(kuyruktaki_NV(u),7,k)=2;

```

```

end
end

if nf==4
if time==queuelength(1,end)
    satir=find(queuelength(1,:)==Qlenght);
    kuyrukuzunlugu(satir,k)=Dec_K(1,k);
end
elseif nf==3
    if time==queuelength(1,end)
        satir=find(queuelength(1,:)==Qlenght);
        kuyrukuzunlugu(satir,k)=Dec_K(1,k);
    end
end

else

    if time==queuelength(1,end)
        satir=find(queuelength(1,:)==Qlenght);
        kuyrukuzunlugu(satir,k)=Dec_K(1,k);
    end
end

% orta şerit

elseif inx_komsu == 1
    Dec_N1(k)=length(find(g(:,1,k)>=Dec_yak_L(k))); %
    zero(k)=length(find(g(:,1,k)==0));

    SL1(k)=length(find(g(:,1,k)>0)); % stop çizgisini gecmis olan arac
sayisi.
    m11 = (g(:,1,k) < 0);
    p11 = (g(:,1,k)>=-sol_cep_uzunluk(k));
    q11 = (g(:,5,k) == 3);
    s111 = p11.* q11;
    s11 = s111 .* m11;
    n_ghost_d2(k) = sum(s11);

    Dec1_N(k)=Dec_N1(k)-SL1(k)-zero(k)- n_ghost_d2(k);
    n_ghost_d2 = zeros(1,nl);

%kuyruk uzunluğu (orta şerit sol cebe giren araç var)

    kuyruktaki_NV =find(g(:,1,k)>=Dec_yak_L(k) & g(:,1,k)<0 & g(:,5,k)~=3
        & g(:,2,k)<=2.78);

    klgth=length(kuyruktaki_NV);

```

```

if klgt>0
for u=1:klgt
    if g(kuyruktaki_NV(u),7,k)==0
        Dec_K(k)= Dec_K(k)+g(kuyruktaki_NV(u),6,k);
        g(kuyruktaki_NV(u),7,k)=1;
    end
end
end

kuyruktaki_NV =find(g(:,1,k)>=Dec_yak_L(k)&    g(:,1,k)<0    &
                    g(:,2,k)>=2.78 & g(:,5,k)~=3 & g(:,7,k)==1);
klgt=length(kuyruktaki_NV);

if klgt>0
for u=1:klgt
    Dec_K(k)= Dec_K(k)-g(kuyruktaki_NV(u),6,k);
    g(kuyruktaki_NV(u),7,k)=0;
    if Dec_K(k)<0.2
        Dec_K(k)= 0;
    end
end
end

kuyruktaki_NV =find(g(:,1,k)>0 & g(:,5,k)~=3 & g(:,7,k)==1);

klgt=length(kuyruktaki_NV);

if klgt>0
for u=1:klgt
    Dec_K(k)= Dec_K(k)-g(kuyruktaki_NV(u),6,k);
    g(kuyruktaki_NV(u),7,k)=2;
end
end

if nf==4
if time==queuelengt(1,end)
    satir=find(queuelengt(1,:)==Qlengt);
    kuyrukuzunlugu(satir,k)=Dec_K(1,k);
end

elseif nf==3

    if time==queuelengt(1,end)
        satir=find(queuelengt(1,:)==Qlengt);
        kuyrukuzunlugu(satir,k)=Dec_K(1,k);
    end

else

```

```
        if time==queuelenght(1,end)
            satir=find(queuelenght(1,:)==Qlenght);
            kuyrukuzunlugu(satir,k)=Dec_K(1,k);
        end
    end
end
end
end
```



## ÖZGEÇMİŞ

Ersin KORKMAZ, 1987 yılında Kırıkkale’de doğmuştur. İlköğretim ve lise öğrenimini Kırıkkale’de tamamlamıştır. 2011 yılında Erciyes Üniversitesi Elektrik Elektronik Mühendisliği bölümünü, 2012 yılında Erciyes Üniversitesi İnşaat Mühendisliği bölümünü bitirmiştir. 2013 yılında Kırıkkale Üniversitesi İnşaat Mühendisliği Bölümünde Ulaştırma Anabilim Dalında Araştırma Görevlisi olarak çalışmaya başlamış ve 2016 yılında yüksek lisansını tamamlamıştır. Ulusal ve Uluslararası alanda birçok makale ve bildirisi bulunmaktadır. Gecikme analizi, sinyalizasyon, optimizasyon teknikleri ve trafik kazaları ile ilgilenmektedir.