

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

**ANFİS VE POPÜLASYONA DAYALI OPTİMİZASYON ALGORİTMASI
KULLANILARAK İZOLE BİR KAVŞAĞIN SİNYAL ZAMANININ
TAHMİNİ**

DOKTORA TEZİ

İnş. Yük. Müh. Amir SHAHKAR

EYLÜL 2020
TRABZON



**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

**ANFİS VE POPÜLASYONA DAYALI OPTİMİZASYON ALGORİTMASI
KULLANILARAK İZOLE BİR KAVŞAĞIN SİNYAL ZAMANININ TAHMİNİ**

**İnş. Yük. Müh. Amir SHAHKAR
ORCID: 0000 - 0001 - 6466 - 6993**

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
“DOKTOR (İNŞAAT MÜHENDİSLİĞİ)”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 18/08/2020
Tezin Savunma Tarihi : 07.09.2020**

**Tez Danışmanı : Prof. Dr. Şeref ORUÇ
ORCID : 0000 - 0001 - 5788 - 890X**

Trabzon 2020

ÖNSÖZ

Tez çalışması sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana destek olan değerli danışman hocam sayın Prof. Dr. Şeref ORUÇ'a teşekkürü bir borç biliyor ve şükranlarımı sunuyorum.

Önerilerini benimle paylaşan değerli arkadaşım Dr. Aref YELEGİ'ye, lisans ve yüksek lisans eğitimim boyunca sabır ve ilgiyle bana destek olan tüm hocalarıma teşekkür ederim.

Çalışmalarım boyunca maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan sevgili babam Mohammed SHAHKAR'a, baba özlemini bana artmayan amcam Mustafa SHAHKAR'a sonsuz teşekkürlerimi sunuyorum.

Bu çalışmadaki azim ve isteğimin başlıca kaynağı başaracağıma her zaman inanan, beni bu günlere sevgi ve saygı kelimelerinin anlamlarını bilecek şekilde yetiştirerek getiren, benden hiçbir zaman desteğini esirgemeyen, hayattaki en büyük şansım, rahmetli Annem Feride AGAHVAZİFE'nin anısına ve alanımızda bilgi ışığını paylaşıp geliştireceğine inandığım insanlığa takdim ediyorum.

Amir SHAHKAR
Trabzon, 2020

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduğum “ANFİS VE POPÜLASYONA DAYALI OPTİMİZASYON ALGORİTMASI KULLANILARAK İZOLE BİR KAVŞAĞIN SİNYAL ZAMANININ TAHMİNİ” başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. Şeref ORUÇ’un sorumluluğunda tamamladığımı, verileri kendim topladığımı, analizleri ilgili laboratuarlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 07/09/2020

Amir SHAHKAR

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	X
SUMMARY	XI
ŞEKİLLER DİZİNİ	XII
TABLolar DİZİNİ.....	XVI
SEMBOLLER DİZİNİ	XVII
1. GENEL BİLGİLER.....	1
1.1. Giriş	1
1.2. Çalışma Amacı ve İzlenen Yol	4
1.3. Veri Toplama	6
1.3.1. Veri Toplama Yöntemleri.....	6
1.4. Lipanov'un Üstü	7
1.4.1. Liapunov Görünümünü Kullanarak kaotik Zaman Serileri Davranışını Tanımlamak	8
1.5. VİSSİM.....	9
1.5.1. VİSSİM Yapısı	11
1.6. Bulanık Mantık	12
1.6.1. Dilsel Değişkenler	13
1.6.2. Üyelik Fonsiyon Tanımları.....	14
1.6.2.1. Üyelik Fonsiyonlarının Türleri	14
1.6.2.1.1. Üçgen Üyelik Fonsiyonu	15
1.6.2.1.2. Yamuk Üyelik Fonsiyonu.....	15
1.6.2.1.3. Gauss Üyelik Fonsiyonu.....	16
1.6.3. Bulanık Kümeler İşlemleri	17
1.7. Belirsizlik Türleri.....	18
1.7.1. Olasılığa Karşı Bulanıklık	18
1.7.2. Belirsizlik.....	19
1.8. Bulanık Kontrol	21
1.8.1. Bulanık Çıkarımda Adımlar	23

1.8.2.	Durulařtırma	28
1.9.	Sinir Ağları	34
1.9.1.	İleri Beslemeli Sinir Ağları.....	36
1.9.2.	Sinir Ağlarında Yüzeysel Uyumu (Montajı)	39
1.10.	Neuro Bulanık Sistemler (Neuro Fuzzy Systems).....	40
1.10.1.	Neuro Fuzzy Sistemlerin Avantajları	41
1.10.2.	Yapay Sinir Ağında Bulanık Bir Sistemin Yapısı	42
1.10.3.	Neuro Fuzzy Sistemlerine Örnekler	44
1.11.	Adaptif Neuro-Bulanık Çıkarım Sistemi (ANFIS).....	46
1.11.1.	ANFIS Tip-1 Mimarisi	48
1.11.2.	ANFİS Tip-2 Mimarisi	51
1.12.	Takviye Öğrenim	52
1.12.1.	Denetlenmeyen Öğrenme	53
1.13.	Üstsezgisel Algoritmalar (Tek-Amaçlı)	54
1.14.	Genetik.....	55
1.15.	Genetic Algoritmalar (GA).....	57
1.15.1.	GA'nın Diğer Yöntemlerden Farkı.....	58
1.15.2.	Basit Genetik Algoritma	59
1.15.3.	Çözümlerin (Kromozomun) Kodlanması	59
1.15.4.	İlk Populasyonun Oluřturulması	60
1.15.5.	Uygunluk Deęerinin Hesaplanması	60
1.15.6.	Çoęalma İşleminin Uygulanması	60
1.15.7.	Çaprazlama İşleminin Uygulanması.....	61
1.15.8.	Mutasyon İşleminin Uygulanması.....	61
1.15.9.	Yeni Kuşanın Oluřması ve Döngünün Durdurulması	62
1.15.10.	Genetik Algoritmalarda Parametre Seçimi	62
1.16.	Parçacık Sürü Optimizasyon Algoritmaları (PSO).....	64
1.16.1.	PSO Algoritmasının Farklı Topolojileri	66
1.16.1.1.	Kare Topolojisi (Van Neuman)	66
1.16.1.2.	Halka Topolojisi (Yerel Optimum)	66
1.16.1.3.	Yıldız Topolojisi (Global Optimum).....	67
1.16.2.	Temel PSO Parametrelerinin Denklemleri (Hız ve Hareket)	67
1.16.3.	Parçacık Sürü Optimizasyon (PSO) Algoritmanın Uygulama Adımları.....	69
1.17.	Harmoni Arama Algoritmaları (Harmony Search HS).....	71

1.17.1.	Uyum Arama Parametreleri.....	73
1.17.1.1.	Uyum Arama Adımları.....	74
1.18.	Çok Amaçlı Optimizasyon (MOP).....	78
1.18.1.	Klasik Yöntemler.....	78
1.18.2.	Evrimsel Yöntemler.....	79
1.19.	Domine Olmayan Genetic Algorithm (NSGA II).....	80
1.19.1.	Algoritmasının Avantajları (NSGA II).....	81
1.20.	Parçacık Sürüşü Optimizasyonu ve Evrimsel Algoritmalar.....	82
1.21.	Çok Amaçlı Parçacık Sürüşü Optimizasyonu (MOPSO).....	83
1.21.1.	Çok Amaçlı Parçacık Sürüşü Optimizasyonunun Gelişimi.....	84
1.21.2.	MOPSO'nun En Çok Uygulandığı Alanlar.....	86
2.	YAPILAN ÇALIŞMALAR.....	88
2.1.	Çalışma Adımları ve Veri Toplama Yöntemi.....	88
2.2.	Test Ortamı.....	88
2.3.	Mikroskobik Simülasyon (VİSSİM).....	90
2.4.	ANFİS Tasarımı.....	90
2.5.	Üyelik İşlevlerinin Tasarımı.....	91
2.6.	Tek-Amaçlı Algoritmalar İçin Kullanılan Hata Fonksiyonları-Maliyet Fonksiyonları.....	92
3.	BULGULAR.....	95
3.1.	VİSSİM ve Verilerin Girişi.....	95
3.2.	Kavşağın VİSSİM Ortamında Tasarlanması.....	95
3.3.	Tasarım Sonrası Optimum Yeşil Süreyi Kullanarak Faz kuruluşu.....	96
3.4.	Sabah 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme.....	98
3.4.1.	Sabah Her Yön İçin Kuyruk Uzunluğu Karşılaştırma sonuçları.....	98
3.4.2.	Sabah Her Yön İçin Gecikme Sonuçları.....	99
3.5.	Sabah 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme.....	99
3.5.1.	Sabah Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları.....	99
3.5.2.	Sabah Toplam Gecikme Karşılaştırma Sonuçları.....	100
3.6.	Öğlen 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme.....	100
3.6.1.	Öğlen Her Yön İçin Kuyruk Uzunluğu Karşılaştırma Sonuçları.....	100
3.6.2.	Öğlen Her Yön İçin Gecikme Karşılaştırma Sonuçları.....	101
3.7.	Öğlen 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme.....	101

3.7.1.	Öğlen Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları	101
3.7.2.	Öğlen Toplam Gecikme Karşılaştırma Sonuçları.....	102
3.8.	Akşam 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme.....	102
3.8.1.	Akşam Her Yön İçin Kuyruk Uzunluğu Karşılaştırma Sonuçları	102
3.8.1.	Akşam Her Yön İçin Gecikme Karşılaştırma Sonuçları.....	103
3.9.	Akşam 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme	103
3.9.1.	Akşam Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları.....	103
3.9.2.	Akşam Toplam Gecikme Karşılaştırma Sonuçları	104
3.10.	Tasarlanmış ANFIS Yapısı.....	104
3.10.1.	Tek Amaçlı Optimizasyon Algoritmalarına Dayalı ANFİS Uygulanması.....	106
3.10.1.1.	GA'nın Test Sonu En İyi Maliyeti	108
3.10.1.2.	GA'nın Eğitim Sonu En İyi Maliyeti	108
3.10.1.3.	PSO'nun Test Sonu En İyi Maliyeti	109
3.10.1.4.	PSO'nun Eğitim Sonu En İyi Maliyeti	109
3.10.1.5.	HS'in Test Sonu En İyi Maliyeti	110
3.10.1.6.	HS'in Eğitim Sonu En İyi Maliyeti	110
3.11.	Tek Amaçlı Optimizasyon Algoritmalarının Karşılaştırması.....	113
3.12.	Çok Amaçlı Optimizasyon Algoritmalarına Dayalı ANFİS Uygulanması.....	113
3.12.1.	NSGAII Algoritması F1 ve F2 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	114
3.12.1.1.	NSGAII'nın F1ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu	114
3.12.1.2.	NSGAII'nın F1ve F2 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu	115
3.12.2.	NSGAII Algoritması F1 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	115
3.12.2.1.	NSGAII'nın F1ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu	115
3.12.2.2.	NSGAII'nın F1ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu	116
3.12.3.	NSGAII Algoritması F2 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	116
3.12.3.1.	NSGAII'nın F2ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu	116

3.12.3.2.	NSGAIİ'nin F2 ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu	117
3.12.4.	MOPSO Algoritmasının F1 ve F2 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	117
3.12.4.1.	MOPSO Algoritmasının F1 ve F2 Maliyet Fonksiyonu ile Birlikte Dağılımı	117
3.12.4.2.	MOPSO Algoritmasının F1ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu.....	118
3.12.4.3.	MOPSO Algoritmasının F1ve F2 Fonksiyonları ile Eğitim Sonu En İyi Maliyet sonucu.....	118
3.12.5.	MOPSO Algoritmasının F1 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	119
3.12.5.1.	MOPSO Algoritmasının F1 ve F3 Maliyet Fonksiyonu ile Birlikte Dağılımı	119
3.12.5.2.	MOPSO Algoritmasının F1ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu.....	119
3.12.5.3.	MOPSO Algoritmasının F1ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu.....	120
3.12.6.	MOPSO Algoritmasının F2 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları.....	120
3.12.6.1.	MOPSO Algoritmasının F2 ve F3 Maliyet Fonksiyonu ile Birlikte Dağılımı	120
3.12.6.2.	MOPSO Algoritmasının F2 ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet sonucu.....	121
3.12.6.3.	MOPSO Algoritmasının F2 ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu	121
4.	SONUÇLAR.....	124
5.	ÖNERİLER.....	127
6.	KAYNAKLAR.....	129
7.	EKLER	145
ÖZGEÇMİŞ		

Doktora Tezi

ÖZET

ANFİS VE POPÜLASYONA DAYALI OPTİMİZYASON ALGORİTMASI KULLANILARAK İZOLE BİR KAVŞAĞIN SİNYAL ZAMANININ TAHMİNİ

Amir SHAHKAR

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Şeref ORUÇ
2020, 144 Sayfa, 14 Sayfa Ek

Trafik sıkışıklığı, hızla artan araç ve yaya nüfusu nedeniyle özellikle metropolitan alanlarda ele alınması gereken meselelerden biri haline gelmektedir. Kaza sayısındaki artış, hava kirliliği, trafik kuyruklarındaki zaman ve enerji kaybı, bu sıkışıklığın neden olduğu en önemli sorunlardandır. Kentsel trafiğin doğası dinamiktir. Trafik sinyallerinin amacı; güvenli seyahat sağlarken, yoğun kavşaklarda yol kapasitesini arttırarak gecikmeleri azaltmaktır. Günümüzde trafik sinyal zamanlamasını optimize eden yöntemler, kavşaktaki trafik dinamiklerini yeterince yakalaymayan matematiksel modellere dayanmaktadır. Bu nedenle trafiğin yoğun olduğu saatlerde kavşaklardaki tıkanıklık sorununu aşabilmek için düşük performanslı matematiksel modeller yerine uyarlamalı trafik kontrol modellerinin kullanımı kaçınılmaz hale gelmiştir. İstanbul ,Türkiye'nin en büyük şehridir. Şehir; uluslararası, ticari, siyasi, akademik, sosyal, sanatsal ve kültürel anlamda her yıl çok büyük sayıda turist ağırlamaktadır. İstanbul, Avrupa yakasında 76,5 milyon metrekare civarında bir alan kaplamaktadır. 200 milyonluk yolcu kapasitesi ve 350'den fazla uçuş noktası ile birlikte dünya listelerinde çok büyük avantaja sahiptir. Dolayısıyla bahsedilen bütün faaliyetler, şehrin ne denli bir trafik yüküne maruz kaldığını görme noktasında yeterlidir. Bu tez çalışmasındaki veriler, video kameralar vasıtasıyla Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul üzerinde dört kollu ve üç fazlı kavşaktan toplanmış ve Excel ortamında kullanabilecek veri formatına dönüştürülmüştür. Kavşağa ait hacim sayımı tamamlandıktan sonra zirve saat standardı elde edilmiştir. Sabah, öğlen ve akşam için belirli saatlerde yüksek araç hacmi (aşırı doymun durumlar) zirve saatin ne zaman gerçekleştiğini göstermektedir. Bu çalışmada, ANFİS mimarisine uygun, kaliteli ve hatasız girdiler elde edebilmek için VİSSİM simülasyon modeli kullanılmıştır. VİSSİM trafik simülasyon ortamında kavşağın modellenmesi için trafik hacim verileri ve diğer bağımlı parametreler girilmiş, programın tasarım özelliği kullanılarak optimum yeşil süre, kuyruk uzunluğu ve gecikme değerleri elde edilmiştir. Tasarım sonrası elde edilen değerlerde iyileşmeler olduğu, kavşaktaki doymunluk oranının kritik değerlerin altında kaldığı gözlenmiştir. Tasarladığımız ANFİS-Type1 mimarisi üç girdi ve bir çıktıdan oluşmaktadır. ANFİS, sistemin kurallarını uyarlamalı olarak öğrenen ve değiştiren birinci sıra Sugeno-FIS'e dayanmaktadır. ANFİS eğitimi, bir optimizasyon algoritması kullanarak öncül ve sonuç parametrelerinin belirlenmesi anlamına gelmektedir. ANFİS ile başarılı sonuçlara ulaşmak doğrudan eğitildiği optimizasyon algoritmalarla bağlıdır. Bu çalışmada eğitim süresinde parametreleri ayarlamak ve hataları minimize edebilmek için Tek-amaçlı (GA, PSO, HS) ve Çok-amaçlı (MOPSO, NSGAI) algoritmalar kullanılmıştır. Tek-amaçlı algoritmalar ile yapılan simülasyon sonuçları incelendiğinde, ANFİS-GA'in ANFİS-PSO ve ANFİS-HS'den daha iyi tahminler gösterdiği gözlenmiştir. Çok-amaçlı algoritmalar ile yapılan simülasyon sonuçları incelendiğinde, ANFİS-MOPSO'nun ANFİS- NSGAI'dan daha iyi tahminler gösterdiği gözlenmiştir.

Anahtar Kelimeler: Trafik Modelleme, Optimizasyon, ANFİS, GA, PSO, MOPSO, NSGAI

PhD. Thesis

SUMMARY

THE PREDICTION OF THE SIGNAL TIMING OF AN ISOLATED INTERSECTION USING A COMBINATION OF ANFIS AND POPULATION-BASED OPTIMIZATION ALGORITHM

Amir SHAHKAR

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Civil Engineering Graduate Program
Supervisor: Prof. Dr. Şeref ORUÇ
2020, 144 Pages, 14 Pages Appendix

Due to the rapidly increasing number of vehicles and pedestrian population, traffic congestion has become one of the issues to be addressed specifically in metropolitan areas. An increase in the number of accidents, air pollution, and loss of energy and time due to the queue of traffics are among the most important problems caused by traffic congestion. Urban traffic is dynamic in nature. The purpose of traffic signals is to provide safe travel and decrease the time of traffic delays by increasing the road capacity at crowded intersections. Nowadays, methods to optimize traffic signal timing are based on mathematical models that don't adequately capture traffic dynamics at an intersection. Therefore, it has become inevitable to use adaptive traffic control models as a replacement for low-performance mathematical models in order to overcome the traffic congestion problem at the intersections during peak hours. Istanbul is the largest city in Turkey. In international, commercial, political, academic, social, artistic, and cultural terms, a city hosts numerous tourists annually. Istanbul covers an area of approximately 76,5 million square meters on the European side. It has a 200 million seating capacity together with more than 350 flight points, which are considered to be great advantages on the world lists. Therefore, all the above-mentioned activities are sufficient to show the level of traffic the city is exposed to. The data of the current thesis study was collected from three-phase and four-way intersections in Cumhuriyet road, Yarımburgaz 34303, Kuchukchekmece, Istanbul and converted into a data format that can be used in excel environment. After the volume count related to an intersection was completed, the peak hour standard obtained. High vehicle volume (super-saturated situations) at certain times of the day (morning, noon, and evening) indicates the times of peak hour occurrence. In the current study, the VISSIM simulation model was utilized to obtain high quality, and error-free inputs suitable for the ANFIS architecture. In the VISSIM traffic simulation environment, traffic volume data and other related parameters were entered to model the intersection, and optimum time of the green signal, length of queue of traffic, and traffic delay values were obtained by utilizing the design feature of the program. After the design was completed, we observed that there were improvements in the values obtained, and the saturation degree at the intersection remained below the critical values. The ANFIS-Type1 architecture we designed incorporates three inputs and one output. ANFIS is based on the first order SUGENO-FIS which learns and changes the rules of the system, adaptively. ANFIS training means the calculation of the precursor and resultant parameters using an optimization algorithm. Achieving successful results with ANFIS depends directly on the optimization algorithms that are trained for. In the current study, single-objective (GA, PSO, HS) and multi-objective (MOPSO, NSGAI) algorithms were used to regulate the parameters and minimize errors during the training period. By analyzing simulations completed by the single-objective algorithms, we discovered that ANFIS-GA resulted in better estimations than ANFIS-PSO and ANFIS-HS. In addition, By analyzing simulations completed by the multi-objective algorithms, it was found out that ANFIS-MOPSO resulted in better estimations than ANFIS-NSGAI.

Keywords: Traffic Modeling, Optimization, ANFIS, GA, PSO, MOPSO, NSGAI

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. (a,b) üçgen ve ymuk üyelik fonksiyonları ve parametreleri.....	16
Şekil 2. Gauss üyelik fonksiyonu	16
Şekil 3. Bulanık kümelerin birleşimi.....	17
Şekil 4. Bulanık kümelerin kesişimi.....	17
Şekil 5. Bulanık kümelerin tümleyeni	18
Şekil 6. Kanıt ve önermeler arasındaki belirsizlik durumun'daki ilişkiler.....	20
Şekil 7. Basit bir bulanık kontrolör	21
Şekil 8. İki kuralın bir kural tabanında bulanık çıkarım.....	22
Şekil 9. μ Üyelik fonksiyonu ve çeşitleri. a) Kırpılmış çeşiti b) ölçeklendirilmiş çeşidi ($w = 0: 8$).....	25
Şekil 10. (a) Üyelik fonksiyonları: $\mu_{T_1}, \mu_{T_2}, \mu_{T_3}$ ve kural ateşleme gücü $w_1 = 0.6, w_2 = 0.7, w_3 = 0.9$	26
Şekil 11. (b) Min-maks ayrışımı.....	27
Şekil 12. (c) işlenmiş (ürün)-toplam ayrışımı	27
Şekil 13. (d) Üst eğri: Sınırlı toplamı kullanarak ürün-toplam ayrışım. Düşük eğri: olasılıksal toplamı kullanarak ürün- toplam ayrışımın göstermektedir.	27
Şekil 14. (a) μ_1, μ_2, μ_3 Üyelik fonksiyonlarının Durulaştırması (bk. Şekil 10).....	29
Şekil 15. (b) Ağırlık olarak w_i ve V_i 'nin kullanıldığı LCOA yöntemi	29
Şekil 16. (c) Ağırlık olarak w_i 'nin kullanıldığı LMOM yöntemi.....	30
Şekil 17. LMOM yöntemi	33
Şekil 18. Bir gizli tabakaya sahip bir ileriye besleme sinir ağıdır. a_{ij} ve c_{jk} , ağ ağırlık parametreleridir.....	36
Şekil 19. (a) Hiperbolik aktivasyon fonksiyonu -1'den 1'e sıkıştır.....	37
Şekil 20. (b) Sigmoidal (veya lojistik) aktivasyon fonksiyonu 0'dan 1'e kadar sıkıştır	38
Şekil 21. Sinir ağı biçimindeki bir basit bulanık sistemin yapısı	42
Şekil 22. Sinir ağı biçimindeki bulanık sistem örneği. Aynı dilsel değerler birkaç kuralda görünmektedir.	43
Şekil 23. (a) Sistem giriş-çıkış alanındaki veri kümeleri bulanık kuralların varlığına işaret eder.	44
Şekil 24. (b) Ürün uzayı kümeleme yöntemi.....	45
Şekil 25. Doğrusal olarak ayrılabilir veri kümeleri düz bir çizgi ile ayrılabilir.	45

Şekil 26.	A Sugeno bulanık girişim sistemi “If-Then” ve bulanık mantık mekanizması. b- b ANFIS mimarisi. [121].	49
Şekil 27.	İki giriş ve iki üyelik fonksiyonuna sahip örnek bir T2-ANFIS kontrolörünün yapısı	51
Şekil 28.	Genetik algoritmanın akış diyağramı	63
Şekil 29.	Bird flocking	65
Şekil 30.	Fish schoolling	65
Şekil 31.	Animal herding	65
Şekil 32.	PSO algoritmasının değişik topolojileri	66
Şekil 33.	PSO pseudo cod.	69
Şekil 34.	PSO akış şeması	71
Şekil 35.	Müzik doğaçlama ve mühendislik optimizasyonu arasındaki benzerlik	73
Şekil 36.	Uyum arama algoritmasının akış şeması	75
Şekil 37.	HS pseudo cod.	76
Şekil 38.	NSGA II algoritmasının akış şeması	82
Şekil 39.	MOPSO algoritmasının akış şeması	87
Şekil 40.	Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul (çalışma alanı)	89
Şekil 41.	Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul (çalışma alanı)	90
Şekil 42.	Üç girişi ve bir çıkışı olan ANFIS yapısı	91
Şekil 43.	Tasarlanmış üyelik fonksiyonları	92
Şekil 44.	Test alanı	95
Şekil 45.	VISSIM’de tasarlanmış platform.	96
Şekil 46.	Sabah için optimum devre süresi	97
Şekil 47.	Öğlen için optimum devre süresi	97
Şekil 48.	Akşam için optimum devre süresi	97
Şekil 49.	Sabah modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması	98
Şekil 50.	Sabah modelleme öncesi ve sonrası gecikmenin karşılaştırılması	99
Şekil 51.	Sabah modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması	99
Şekil 52.	Sabah modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması	100
Şekil 53.	Öğlen modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması	100

Şekil 54.	Öğlen modelleme öncesi ve sonrası gecikmenin karşılaştırılması.....	101
Şekil 55.	Öğlen modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması	101
Şekil 56.	Öğlen modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması	102
Şekil 57.	Akşam modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması	102
Şekil 58.	Akşam modelleme öncesi ve sonrası gecikmenin karşılaştırılması	103
Şekil 59.	Akşam modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması	103
Şekil 60.	Akşam modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması	104
Şekil 61.	Üç girişi ve bir çıkışı olan ANFIS yapısı	105
Şekil 62.	Tasarlanmış üyelik fonksiyonları	105
Şekil 63.	ANFIS eğitiminde kullanılan öncül ve sonuç parametrelerin gösterilmesi	106
Şekil 64.	GA'nın en iyi maliyet sonucu (Test).....	108
Şekil 65.	GA'nın en iyi maliyet sonucu (Eğitim)	108
Şekil 66.	PSO'nun en iyi maliyet sonucu (Test)	109
Şekil 67.	PSO'nun en iyi maliyet sonucu (Eğitim)	109
Şekil 68.	HS'in en iyi maliyet sonucu (Test)	110
Şekil 69.	HS'in en iyi maliyet sonucu (Eğitim).....	110
Şekil 70.	ANFIS ile birlikte meta-sezgisel algoritmaların performansının karşılaştırılması	113
Şekil 71.	NSGAIİ'nin F1ve F2 fonksiyonları ile en iyi maliyet sonucu (Test).....	114
Şekil 72.	NSGAIİ'nin F1ve F2 fonksiyonları ile en iyi maliyet sonucu (Eğitim).....	115
Şekil 73.	NSGAIİ'nin F1ve F3 fonksiyonları ile en iyi maliyet sonucu (Test).....	115
Şekil 74.	NSGAIİ'nin F1ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim).....	116
Şekil 75.	NSGAIİ'nin F2ve F3 fonksiyonları ile en iyi maliyet sonucu (Test).....	116
Şekil 76.	NSGAIİ'nin F2ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim).....	117
Şekil 77.	MOPSO'nun F1 ve F2 fonksiyonları ile dağılım şeması	117
Şekil 78.	MOPSO'nun F1 ve F2 fonksiyonları ile en iyi maliyet sonucu (Test)	118
Şekil 79.	MOPSO'nun F1 ve F2 fonksiyonları ile en iyi maliyet sonucu (Eğitim)	118
Şekil 80.	MOPSO'nun F1 ve F3 fonksiyonları ile dağılım şeması	119
Şekil 81.	MOPSO'nun F1 ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)	119

Şekil 82.	MOPSO'nun F1 ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)	120
Şekil 83.	MOPSO'nun F2 ve F3 fonksiyonları ile dağılım şeması	120
Şekil 84.	MOPSO'nun F2 ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)	121
Şekil 85.	MOPSO'nun F2 ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)	121



TABLÖLER DİZİNİ

	<u>Sayfa No</u>
Tablo 1.	GA, PSO ve HS'nin genel ve spesifik parametreleri..... 111
Tablo 2.	1000 fonksiyon 10 iterasyon sonunda GA deney sonuçları..... 111
Tablo 3.	1000 fonksiyon 10 tekr�r sonunda PSO deney sonuçları 112
Tablo 4.	1000 fonksiyon 10 tekr�r sonunda HS deney sonuçları..... 112
Tablo 5.	Aynı kriterler de en iyi maliyet, ortalama ve varyans sonuçları 112
Tablo 6.	Tekr�r sonunda F1 ve F2 fonksiyonlarının deney sonuçları..... 122
Tablo 7.	10 tekr�r sonunda F1 ve F3 fonksiyonlarının deney sonuçları 122
Tablo 8.	10 tekr�r sonunda F2 ve F3 fonksiyonlarının deney sonuçları 122
Tablo 9.	MOPSO algoritmasının F1, F2 ve F3 maliyet fonksiyonları ile 10 tekr�r sonu performans deęerlendirmesi..... 123
Tablo 10.	NSGAII algoritmasının F1, F2 ve F3 maliyet fonksiyonları ile 10 tekr�r sonu performans deęerlendirmesi..... 123
Tablo 11.	MOPSO ve NSGAII algoritmalarına ait parametreler 123

SEMBOLLER DİZİNİ

CHG	: Sera Gazı
SCATS	: Sydney Koordineli Uyarmalı Trafik Sistemi
SCOOT	: Ayrık Çevrim ve Ofset Optimizasyon Tekniği
OPAC	: Uyarmalı Kontrol İçin Optimizasyon politikaları
ANFİS	: Uyarlanabilir Ağ Tabanlı Bulanık Çıkarım sistemi
GA	: Genetik Algoritması
PSO	: Parçacık Sürü Optimizasyon Algoritması
HS	: Uyum Arama Algoritması
MLP	: Çok Katmanlı Algılayıcı
MFD	: Microscopic Fundamental Diagram
IETMS	: İnteraktif Elektronik Teknik El Kitabı
VISSIM	: Görsel Simülatör
λ	: Lipanov Parametresi
TÜİK	: Türkiye İstatistik Kürümü
g	: Yeşil Zaman Döngü Uzunluğu
c	: Tüm Döngü Uzunluğu
PCE	: Binek Araç Eşdeğeri
PCU	: Binek Araç Ünitesi
AI	: Yapay Zekâ
EA	: Evrimsel Algoritma
MOP	: Çok Amaçlı Optimizasyon
ABD	: Amerikan Birleşik Devleti
HCM	: Otoyol Kapasite Kılavuzu
SSG	: Sinyal Durumu Üretici
l_{∞}	: Mesafe Metrikleri
N	: Toplam Nüfus Boyutu
X_A	: Karakteristik Fonksiyon
U, X	: Evrensel Küme
C_i	: Bulanık Küme
A	: Net (gevrek) Küme
S, E	: Herhangi Bulanık küme

μ_S	: Üyelik Fonksiyonu
Sup (S)	: Destek Küme
Cor (S)	: Çekirdek Küme
P_i	: Üyelik Fonksiyonlarının Parametreleri
$\tilde{A}, \tilde{B}$	: Bulanık Kümeler
$\mu_{\tilde{A} \cup \tilde{B}}$	: Birleşim Üyelik Fonksiyonu
$\mu_{\tilde{A} \cap \tilde{B}}$	: Kesişim Üyelik Fonksiyonu
$\bar{\tilde{A}}$	: Bulanık Kümenin Tümlenyeni
$\mu_{\bar{\tilde{A}}}$	: Tamamlayan Üyelik fonksiyonu
$\emptyset$	: Boş Küme
APP	: Yaklaşan Trafik
QUE	: Kuyrukta Bekleyen Araçlar
EXT	: Kavşaktan Ayrılan Araçlar
$\mu_L(A)$	: A'nin L'deki Üyelik Derecesi
$P_L(B)$	: B'nin L'ye Ait Olma Olasılığı
T	: Seyahat Süresi
N	: Kuraldan Oluşan Kural Tabanı
m_i	: Kural Önem Ağırlıkları
V_i	: Küme Hacmi
w	: Ateş Gücü
T'	: Korelasyon-Minimum Çıkarım (Kırılmış Bulanık Küme)
$\mu_{T'}$	: Korelasyon-Minimum Çıkarım Kümesinin Üyelik Fonksiyonu
$T'_i, i = 1, 2, \dots, N$	: Kırılmış Bulanık Kümeler
w_i	: Bulanık kümelerin Ağırlıkları
m_i	: i. Kuralın Ağırlığı
$\mu_{T'_i yeni}$	: Normalize Edilmiş Üyelik Fonksiyonu
COA	: Alan Merkez Durulaştırma Yöntemi
MOM	: Maksi Mizanın Ortalaması Durulaştırma Yöntemi
y^*	: T' Bulanık Kümenin Merkezi
LCOA	: Maksi Mizanın Ortalaması Yerel Alan Merkez Durulaştırma Yöntemi
LMOM	: Maksi mizanın Ortalaması Yerel Ortalama Durulaştırma Yöntemi
M_{w_i}	: Birleşmiş Bulanık Kümelerin Ağırlığı
w_i	: i'nin Kuralın Ateş Gücüdür
V_i	: Ölçeklenmemiş ve Kesilmemiş T_i Sonuç Kümenin Hacmidir

NN	: Sinir Ağı
a_{ij} ve c_{jk}	: Ağ Ağırlık Parametreleridir
MLP	: Çok Katmanlı Algılayıcı
RBF	: Radyan Taban Fonksiyonu
NN	: Nöron Şebekesi
ANN	: Yapay Sinir Ağı
x_i	: Sistemin Girdi Değişkenleri
δ_j	: Sabit Girdi
z	: Gizli Katman Çıktısı
$k = \theta_k$	: Sabit Girdi
v_k	: Çıktı Katmanı Aktivasyon Fonksiyonu
E	: Hata Fonksiyonu
v	: Ağ Çıkışı
d	: İstenen Değer
ω	: Ağırlık Parametresi
η	: Bir Öğrenme Parametresidir
α	: Momentum Parametresi
SD	: Dağılım Miktarının
FIS	: Bulanık Çıkarım Sistemi
ANFIS	: Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi
MSE	: Ortalama Kare Hata Maliyet Fonksiyonu
RMSE	: Kök Ortalama Kare Hatası
SOC	: Şarj Durumu
ISMS	: Akıllı Sistemler ve Modelleme ve Simülasyon
C	: Sinyal Döngüsü Zamanı
q_i	: Trafik Akış Oranı
L	: Kayıp Zaman
g_e	: Etkin Yeşil Süresi
y	: Faz Boyunca Yanacak Yeşil Süresi
x_i	: Döngü Sonunda Her Bir Yönde Biriken Kuyruk Miktarı
x_i	: Bir Döngü Sonunda İlgili Yönde Biriken Kuyruk Miktarı,
s_i	: İlgili Yönün Doygunluk Debisi
q_i	: Trafik Akış Oranı
x_0	: Şimdiki Akım,

x_1	: Bir Sonraki akım
x_2	: üçüncü Akım
x_3	: Dördüncü akım
S	: Az
m	: Orta
L	: Yoğun (çok)
$\sim L$	: Yoğun değil (Az veya Orta'dır.)
Δ	: Kural Tabanlı Çalışan Kavşak ile Sıradan Kavşaktan Ayrılan Araç Sayısı Arasındaki Fark
R	: Kural Tabanlı Kavşaktan c Döngü Zamanı Sonunda Ayrılan Toplam Araç Sayısı
δ	: Sıradan Kavşaktan c Döngü Zamanı Sonunda Ayrılan Toplam Araç Miktarı
b	: Denge Değeri $\cong 0.5$
p	: Performans Değeri
q	: Trafik Akış Oranı
x	: Her yöndeki kuyruk uzunluğu
r	: Kural bloğu
f	: Durulaştırma Fonksiyonu ya da Karar Fonksiyonu
D	: Karar, Etkin Yeşil Süresindeki Değişim Oranı
P_{l-1}	: Bir Önceki Bloğun c Döngü Zamanındaki Verim Değeri
D_{l-1}	: Bir Önceki Bloğun c Döngüsü Zamanındaki Asıl Kararı
D_α	: Asıl Karar, Uygulanacak Gerçek Değişim Oranı
İTÜ	: İstanbul Teknik Üniversitesi
X_{best}	: Parçacığın En İyi Konumu
X_{worst}	: Parçacığın En kötü Konumu
$pid(t)$	: d Boyutunda Kişisel En İyi Tecrübesi
$gd(t)$	: d Boyutunda Genel En İyi Tecrübesi
D	: Boyut Sayısı
i	: Döngü İndisi
$\vec{u}_1$	: Eğitim Verileri İşlenirken Döngü Sonundaki Kavşağın Durumu
$\vec{u}$	: Deney Verileri İşlenirken Döngü Sonundaki Kavşağın Durumu
d_i	: Test Sırasında Kavşağın Döngü Sonundaki Durumunun Eğitim Esnasında Elde Edilen Döngü Sonundaki Kavşağın Durumuna Benzerliği, Mesafe Vektörü
p_i	: Eğitim Esnasında Bulunan Döngü Sonundaki Performans Değeri

p_a	: Test esnasında kullanılacak performans değeri
NSGA	: Hâkim Olmayan Genetik Algoritma
NSGA II	: Hâkim Olmayan 2 Genetik Algoritma
f_m^{min}	: (m) Hedef Fonksiyonunun En Düşük Değeridir
f_m^{max}	: (m) Hedef Fonksiyonunun Maksimum Değeridir
MOPSO	: Çok Amaçlı Parçacık Sürü Optimizasyon
MOO	: Çok Amaçlı Optimizasyon
PAES	: Pareto Arşivlenmiş Evrim Stratejisi
POP	: Popülasyon
REP	: Depolanmış Vektörleri Temsil Eden Parçacıkların Konumlarını
W	: Atalet Ağırlığı

1. GENEL BİLGİLER

1.1. Giriş

Kentsel trafik, trafik sıkışıklığının getirdiği sorunlarla karşı karşıya kalma ve gelişmekte olan ülkelerdeki en kritik sıkıntılardan biri haline gelmektedir. Zamanında ve etkin hizmet sunmak için kentsel trafiğin düzenlenmesi bir ihtiyaç haline gelmiştir. Kentsel trafiğin büyük miktarda sera gazı (CHG) üretmesi, kaza sayısındaki artış, trafik kuyruklarındaki zaman ve enerji israfı, trafik sıkışıklığının olumsuz etkilerinden en önemlileridir [1]. Kavşakları genişletmek ve kapasitesini artırmak yeterli bir çözüm değildir. Temel strateji, trafik ışıklarını kontrol ederken, trafik akışını hızlandırmaktır. Kavşak noktalarının kontrolü için bir çok çalışma yapılmış ve halen de yapılmaya devam edilmektedir. Kavşakları kontrol etmenin yaygın bir yolu da trafik sinyal ışıklarının kullanılması ve her trafik fazının zamanını ayarlamaktır. [2]. Birinci nesil elektrikli trafik ışıkları 5 Ağustos 1914'de Cleveland'da kullanılmaya başlanmıştır. Amerika uyruklu bilim adamı Garret Morgan vasıtasıyla geliştirilen ilk trafik lambası, 1914'deki çalışmalarından sonra yapılan elektrikli trafik ışıklarının tescili 1923 yılında yine Morgan'a verilmiştir. Türkiye'de 1929 yılından itibaren trafik lambası kullanılmaktadır. İlk trafik lambası İstanbul'da kullanılmıştır. Normal ve standart çalışan bir trafik lambası, kırmızı, sarı ve yeşil renk içerir. Trafik lambalarında kullanılan renklerin taşıdığı anlam şöyledir:

- Kırmızı ışık: trafiğin durması gerektiğini,
- Sarı ışık: ışığın yeşilden kırmızıya değişmek üzere olduğunu,
- Yeşil ışık: ilgili doğrultuda trafiğin akabileceğini gösterir.

Önerilen çözümler için üç kuşak düşünülmektedir. Genellikle sabit zaman yöntemi olarak adlandırılan ilk nesil, önceden ayarlanmış sinyal dizileri ve manuel bakım gerektirir. Trafik Ağları Çalışma Aracı sabit zaman planlarını hesaplamak için bir araç örneğidir. İkinci nesil trafik tespitine dayalı sinyal zamanlamasını ayarlamaya odaklanmaktadır [3]. Bölünmüş çevrim ofset optimizasyon tekniği ve Sidney koordine edilmiş uyarlamalı trafik Sistemi [4], [5]. Üçüncü nesil dinamik karar verme ve dağıtılmış kontrol sistemleri ile karakterize edilmiştir. Üçüncü nesil tamamen uyarlamalı ve sinyal zamanlamasının optimizasyonu kademeli olarak yapılır [6]. Yolların belli yerlerinde bulunan dedektörlerden toplanan güncel bilgileri dikkate alarak ilgili sinyal süreleri ve dizileri

hesaplanır. Trafik akışına birkaç saniye içinde giren ve çıkan araçların tahmini bu karar vermede önemli faktörlerdir. OPAC ve RODOD bu neslin bazı örnekleridir [7] ve [8]. Bu konuda net olan bir husus, modern şehirlerde yoğun ve her daim artan trafik miktarının klasik yöntemler ve manuel olarak kontrolünün mümkün olmadığını görmekteyiz [9]. Araştırma ve endüstride bilişimsel akıllı yöntemlerin çeşitli kullanımı, bu alandaki verimlilik ve önemi hakkında kanıt sağlamaktadır [10]. Hesaplamalı akıllı yöntemler kendi kendini organize eder ve kısıtlamaların ve koşulların dinamik değişimlerine tepki verir. Bu yöntemlerin, deneyimden öğrenecek yetenekleri nedeniyle gerçek dünyadaki problemleri ele alma potansiyeli vardır [11]. [12]. Denetimin performansını arttırmak için bu yöntemleri uygulanmasına birçok girişim yapılmıştır [13]. Son zamanlarda kentsel trafik üzerine yapılan önemli araştırmalar, bulanık kontrol, sinir ağı, yapay zekâ ve genetik GA, PSO ve HS (meta sezgisel algoritmalar) gibi tekniklere odaklanmaktadır. Zamanlanmış Petri Ağları, SPSA, karınca algoritması, bilgiye dayalı çok etmenli sistem ve mobil ajanıdır [14], [15], [16], [17]. Trabia [18] ve [19]'da doğru ve sola dönüş hareketleri ile izole bir kavşak için çok fazlı bir bulanık mantık denetleyicisi tasarlamıştır. Bingham [20], sinir ağlarından kesişim bulanık kontrol parametrelerini elde etmiş ve bulanık kontrol sonucunu geliştirmiştir. Chen Xiangjun [21] Head'in 1995'te önerdiği gibi, zaman faktörü trafiğe zaman- konum davranışı verdiği için ona daha çok tesadüfi davranış vermektedir, bu da trafik araştırmaların şehirlerde karayolundan çok daha zor kılmaktadır. Sonra, Stathopoulos ve Carlaftis gösterdiği gibi, şehrinin ana yollarındaki trafik akışı, geleneksel periyodik (Aylık, haftalık, günlük, saatlik) düzen anlaşılması ve alınmasını bozan rastgele bir durumu gösteren belirli bir zaman-mekansal davranışa sahiptir. Ayrıca, onlar bu tesadüfi davranışın çok düzensiz olduğunu ve hatta 2 saatlik bir periyotta deneyle özetlenemediğini göstermişlerdir. [22]. Kısa vadeli modelleme ve trafik akışının tahmini için, zaman serilerinin istatistiksel analizleri, ARIMA modelleri türünde hızlı bir şekilde geliştirilmiştir. Bununla birlikte, trafik akışındaki zaman farklılıklarının kısa vadeli tahmini üzerinde yıkıcı bir özellik olduğu göz önünd bulundurulduğunda , geleneksel zaman serisi yöntemleri genellikle kentsel alanda hızlı dalgalanmaları yönetmek için yetersizdir. Makaleler; modellemenin karmaşıklığının, modellerin artan karmaşık davranışlarına bağlı olduğunu göstermektedir. Bu sorunun üstesinden gelmek için, bazı araştırmacılar çok değişkenli modellemeyi ortaya çıkarmıştır. Bu durumda modeller, çakışan yerler gibi istenen konumlardan gelen verileri kullanmaktadır ve ardışık verileri yönetmek için daha fazla özelliği olan , kısa vadeli tahmini kullanılmıştır. Parametrik olmayan regresyon, lokal

lineer regresyon ve Kalman filtreleme modelleri gibi diğer tekniklerin kullanılması kısa süreli trafik modellemesinde iyileşmeler göstermiştir, fakat trafik ışıkları ile arteriyel yollarla test edilmemişlerdir. Akımın mekansal bağımlılığı, lokasyondaki difüzyonunun, akışı tahmin etmek için modelin verimliliğini iyileştirme üzerinde bir etkiye sahip olmasına neden olmuştur. Araştırmalar, bu davranışların modellenmesinin karmaşıklığının, modellerin artan karmaşıklığına bağlı olduğunu göstermektedir [23] Bu problemle başa çıkmak için, bazı araştırmacılar çok değişkenli modellemeye yönelmişlerdir. Bu durumda, modeller, aksi yönde konumlar gibi istenen konumlardan veri kullanmaktadır. Sıralı verileri yönetmek için daha fazla özelliğe sahip olan bu modeller, kısa vadeli akışları öngörmek için kullanılmaktadırlar [24]. Ağ aşırı yüklendiğinde Daganzo teorisine göre [25], giriş hacmindeki kritik miktardan daha küçük bir artış bile trafik durumu üzerinde önemli etkilere sahip olabilmekte , ağ performansının bozulmasına etki göstermektedir. Bu dengesiz koşullarda, akışın giriş hızının kontrol edilmesi, ağın engellenmesini önleyebilmektedir. Yönlendirme trafiğinin dikkate alındığı kentsel trafik ağlarında sinyal kontrollü kavşaklar için belirlenen optimum masraf Chiou tarafından yürütülen bir matematik programı ile elde edilmiştir. [26]. He ve Hou, çalışmalarında gecikme süresi, durma ve ağ kapasitesi gibi sentetik bir objektif işlevle trafik ışığı optimizasyon problemlerini çözmek için Ant Colony yöntemini kullanmışlardır [27]. Radhakrishnan ve Matthew, mikroskobik bir analizle dinamik binek araç birimlerine dayanan bir doyumluk akış modeli geliştirmek için bir metodoloji önermişlerdir [28]. Cantarella ve çalışma arkadaşları, denge varsayımları altında sinyal zamanlamalarının optimizasyonunun, operatif bir çözüm elde edileceğini vaat etmeyebileceğini bulmuştur, çünkü bu, zaman içindeki evrimin çekiciliği olmayabilir [29]. Varia ve ekibi sıkışık ağlarda trafik akışını dinamik akışla birlikte kontrol etme optimizasyon problemini çözmek için genetik algoritma yöntemi kullanmıştır [30]. Di Gangi ve grubu tek kavşak optimizasyonu (yeşil zamanlamalar) ve ağ koordinasyonuna dayanan optimizasyon stratejisi önermiştir [31]. Marciano ve diğer çalışma arkadaşları sıkışık bir karayolu taşımacılığı ağına seyahat talebinin atanmasıyla sinyal ayarı optimizasyonu için bir model ve teknikler sistemi sunmuştur. Optimize edilmiş sinyal ayar parametreleri durumunda ağdaki toplam gecikmelerde ve tahliye sürelerinde önemli azalmalar bulmuşlardır [32]. Chang ve Lieberman [33] tarafından geliştirilen yöntemde, doğrusal olmayan bir programlama, her döngü için yeşil bir periyodu, her doyumluk yaklaşımındaki gerçek kuyruk her zaman ona yakın olacak şekilde ayarlar. Karışık tam sayı doğrusal programlama (MILP) algoritması

kullanılarak mümkün kılınan optimize edilmiş kuyruk ayrıca, makroskopik temel diyagramlar (MFD'ler) son çalışmalarda ağıdaki araç trafik sıkışıklığını azaltmak ve yeniden dağıtmak için trafik sinyal kontrol modelleri geliştirmek için kullanılmaktadır. Leclercq ve arkadaşları MFD için mevcut tahmin yöntemlerini karşılaştırmaya çalışmıştır. Önyargısız MFD'ye ulaşmanın tek yolunun, ağ üzerindeki araç yörüngeleri hakkında tam bilgiye sahip olmak olduğunu göstermişlerdir [34]. Afandizadeh Zargari ve ekibi [35], şehir içi ağlar için, yukarı akış ve aşağı akışta optimal koşulları göz önünde bulundurarak, yukarı akışa gelen bağlantılarının koşullarını sürekli izleyerek ölçüme dayalı bir çözüm önermiştir. Modellerinde sadece kuyruk dökülmesini önlemekle kalmamış, aynı zamanda kuyruk uzunluğu optimizasyonu nedeniyle segment kullanımı da en üst düzeye çıkarılmıştır. Başka bir deyişle, gelen hacmin ölçülmesi ve sınırlandırılması, aşağı akış ağının (korunan ağ olarak da bilinir) gridlock durumundan uzak tutulmasına yardımcı olmuştur ve bu, genel sistem performansını geliştirmiştir. Tahran şehrinde gerçek verilere dayanarak gerçek bir senaryo ile olgusal ağa uygulanan bir model önermişlerdir. Afandizadeh Zargari ve ekibi [36], giriş-çıkış ölçüm stratejisine (IETMS) dayanan gelişmiş bir yöntem sunmuşlardır, sonuçları IETMS modelini kullanarak ortalama hız ve gecikmenin sırasıyla %14 ve 19 iyileşme gösterdiğini göstermektedir. Jarkko Niittymäki [37] uzman bilgisine dayanan dilsel kontrol kurallarını türetmek için bulanık trafik sinyali kontrolüne sistematik bir yaklaşım sunmuştur. Testlerinin sonuçları, bulanık sinyal kontrol algoritmalarının geleneksel araçla çalıştırılan kontrole göre daha iyi etkinlik ölçümleri sunduğunu göstermiştir. Nadia Slimani ve arkadaşları [38] günlük trafik akışını tahmin etmek için bir sinir ağı yapısı tasarlamıştır. Daha sonra, tahmin edilen trafik akışı, Morroco'nun trafik kontrol kuruluşu tarafından toplanan gerçek veri seti ile karşılaştırılmıştır. Bilgisayar simülasyon sonuçları, iyi bir genelleme kapasitesi ve eğitim setinde 0.00927 ve test setinde 0.01321 toplam ortalama kare hatası olan çok katmanlı algılayıcı mimarisi ile en iyi tahminlerin elde edildiğini göstermektedir.

1.2. Çalışma Amacı ve İzlenen Yol

Trafik sıkışıklığı, giderek artan araç ve yaya nedeniyle özellikle metropoliten alanlarda ele alınması gereken sorunlardan biri haline gelmektedir. Büyük şehirlerdeki trafiğin yol açtığı çevresel sorunların yanı sıra gecikmeleri ve finansal kayıpları azaltmak için çok sayıda yöntem mevcuttur. Kavşaklarının sinyal kontrolü için farklı yöntemler

kullanılmaktadır. Sabit çevrim sinyal kontrol sistemleri ucuz ve bakımı kolay olsa da, trafik akışı ağır olduğunda ve zamana bağlı olduğunda düşük performans gösterirler. Trafik hacimleri gün içinde saatler itibariyle dalgalanıyorsa; trafikle çalışan kontrol yöntemleri veya arama tabloları kullanılır, ancak sistemlerin yapısı ve artan gecikmeler ve gerçek zamanlı durumlara uyum sağlayamamaları nedeniyle sorunlar bildirilmiştir. Trafik hacminin yüksek olduğu büyük kavşaklarda, sinyal kontrol sistemlerine ek olarak, genellikle durumu gözlemlemek ve araçları buna göre yönlendirmek için polisler görevlendirilir. Yukarıdaki senaryodaki dezavantaj, polislerin deneyimine ve yargısına bağımlılıktır. Bu nedenle, sabit ve harekete geçirilmiş bir trafik sinyal zaman kontrolörleri yerine otomatik bir karar verme sistemi bulmak için birçok çalışma yapılmıştır. Normalde trafik verileri eksiktir ve kesin olarak belirtilmez veya hiç belirtilmez. Zadeh, 1965 yılında karar verme belirsizliği kavramlarını tanıtmıştır [54]. O zamandan beri, bulanık mantık tabanlı trafik kontrol sistemleri bir dizi araştırmacı tarafından denenmektedir. Bulanık mantık, matematiksel bir modelin geliştirilmesinin çok zor veya imkansız olduğu durumlarda uzmanın düşünmesini modellemek için kullanılır. Trafik sinyali kontrolündeki bulanık mantığın temel amacı, kavşaklardaki sürücüler tarafından gecikme deneyimini en aza indirmektir. Bu alanda kamuya açık alanda çok sayıda dergi ve konferans yayını bulunmaktadır. Yayınlanan çalışmaların çoğu, bulanık çıkarım sistemlerine dayanan sinyal kontrolünün umut verici sonuçlar verme eğiliminde olduğunu göstermektedir. Ancak, üyelik işlevlerini ve kurallarını seçerken doğru kararlara ihtiyaçları vardır. Bu bağlamda, çok daha doğru karar alabilmek amacıyla bir VISSIM trafik simülatörü modeli ve meta-sezgisel algoritmalara sahip bir uyarlanabilir sinirsel bulanık çıkarım sistemi (ANFIS) kullanılmasına karar verilmiştir. Bu işlem, mevcut trafik hacim verileri ve kavşağa ait bağımlı parametrelerin VISSIM trafik simülasyon ortamına girilmesi ile başlamaktadır. İlk aşamada optimal yaklaşan üç bağlantıda (kavşak kollarında) kuyruk uzunluğunu hesapladık. İkinci adımda, bağlantıların girişinde trafik ışığı zaman çizelgesi programını maksimize ettik. Bu optimizasyon, bağlantılardan geçen trafik akışını en üst düzeye çıkarmaya, gecikmeyi en aza indirmeye kuyruk uzunluğunu optimal değerine yakın tutmaya ve döngüler aracılığıyla minimum salınımlarla optimum yeşil zaman elde etmeye dayalı olarak yapıldı. ANIFS yapısı gerçeğe yakın olmaya çalışmaktadır, bu nedenle girdi olarak üç tanımlı zirve zamanı ve VISSIM çıktısını (kuyruk uzunluğu ve gecikmesi için optimize edilmiş veri kümesi) ve ANFİS yapısı için çıktı olarak optimize edilmiş yeşil zamanı kullanır. Bu çalışmada eğitim süresinde parametreleri ayarlamak ve hataları

minimize edebilmek için Tek-amaçlı (GA, PSO, HS) ve Çok-amaçlı (MOPSO, NSGAI) algoritmalar kullanılmıştır.

1.3. Veri Toplama

Trafik kontrolünde ele alınabilecek bir diğer konu veri toplamadır. Klasik yöntemlere dayalı trafik kontrol yöntemleri, istatistiksel yöntemler ve deneysel eğriler kullanılarak deneysel olarak çalışılmış ve sınırlı sayıda trafik parametresini kontrol edilebilmiştir. Bu sistemler, karmaşık trafik koşullarına sahip düşük bir zekâ seviyesine sahipti ve veri toplama için uygun araçların bulunmaması nedeniyle trafik koşullarının ayrıntılarına erişememiştir.

1.3.1. Veri Toplama Yöntemleri

Trafik verilerinin toplanması için birçok yöntem mevcuttur, fiziksel aletler, hidrolik veya piezoelektrik dedektörler gibi mekanik sensörleri içerir. Ultrasonik dedektörler ve kızılötesi sensörler, sokak seviyesine dalgalar göndererek ve araçların yüzeyinden yansıyan dalgalar ile sokak seviyesi arasındaki zaman farkını karşılaştırarak araçları algılar. Radar, aracın yüzeyinden salınan dalgaların sıklığında yapılan değişiklikleri kullanarak hızlarını tespit eder.

Halihazırda araç sayısını saymak ve kullanımını hızlandırmak için kullanılan yöntem, sokağın zeminine monte edilen manyetik rampaların kullanılmasıdır. Manyetik halkalar, basınç sensörleri, ultrason dalgaları ve video matkapları gibi yöntemler birçok ülkede yaygındır. Video görüntülerini incelemek hatayı azaltır ve sonuçları doğru bir şekilde inceler. Parametrelerin ve sonuçların doğruluğunu artırmak için farklı sistemleri doğal olarak birleştiren diğer çıkarımlar için önemli belgeler olarak kullanılabilir. Otomatik video trafik kontrol sisteminin birçok avantajı vardır:

İlk olarak, kamera sayesinde daha fazla parametre (araç sınıfları, araç sayısı, kuyruk uzunluğu, gecikme ve trafik hacmi) elde etme ve kullanma şansına sahip olmaktadır.

İkincisi, ultra kompakt video kameralarının, düşük maliyetli ve hızlı kuruluş avantajıdır.

Üçüncüsü, kameranın geniş açılı görünümü birden fazla kavşağı kapsamasıdır.

Dördüncü ise, kamera kullanımının insanlar için kesinlikle daha güvenli olmasıdır.

Otomatik video trafik kontrol sistemleri hem doğrudan hem de dolaylı modlarda kullanılabilir. Örneğin, uyarı lambalarının değiştirilmesinin zamanlamasını ve dolaylı moddaki sistem ihlallerini belirlemek için, trafik kontrolü yoğun nüfuslu alanlarda ve günlük gece değişiklikleri ile birlikte kullanılabilir. Video teknikleri kullanılarak yapılan öz tanı yaygın olarak kullanılmamıştır. Birincisi, bu sistemler, örneğin; yüksek tıkanıklık, uzun gölgeler ve ışık ve güneşli kaymaları ile, verimlilikte sınırlamalara sahiptir.

İkincisi, üretim maliyeti çok yüksektir.

Üçüncü olarak, görüntülerin işlemciye hızlı bir şekilde aktarılması için bağlantı hatlarının varlığı gereklidir.

Dördüncü olarak, gerçek zamanlı çalışma için gelişmiş donanım gereklidir [39].

Kameranın normal sensörler yerine kullanılmasının tüm dezavantajları ve avantajları ile bu çalışma, video kamera kullanılarak trafik kontrol sisteminin parametrelerini çıkarmaya çalışmıştır. Ve akıllı bir kontrol yaklaşımı kullanarak, gecikmeyi ve araç durma oranları azaltarak en iyi yeşil zamanını tahmin edebilmek açısından trafik sorunu optimize edilmiştir.

Belirli bir durumda trafik kontrol sisteminin tasarlanması için, sistemin tipine göre değişen araçların hareketiyle ilgili parametreler elde edilir. Bu parametreler, trafik hacimleri, araç tipleri, araç sayısı, araç hızı, sürüş ihlalleri, sürüş kazaları ve benzerleri olabilir. Parametrelerdeki farklılıklar nedeniyle farklı yöntemler kullanılır. Bilgi toplamak ve fiziksel enstrümanların da kullanabileceği bilgilerin istenilen parametrelerini çıkarmak için gözlemci insan kaynaklarının rutin kullanımı sağlanır. Halihazırda araç sayısında ve hızlarında en çok kullanılan yöntem, sokağın zeminine monte edilen manyetik rampaların kullanılmasıdır [40], [41] Video kameraların kullanımı son yıllarda araştırmacıları kendine çekmiştir. Kameraların kullanımı, daha önce sözü edilen birçok avantajı vardır. Daha önce de söylediğimiz gibi, bu çalışmada bilgi toplamak için bir video kamera kullanmış bulunmaktayız.

1.4. Lipanov'un Üstü

Uygulamalarından biri, bir sistemdeki kaosu tespit edilmesi ve aynı zamanda bir davranışın kaotik doğasının bir ölçüsüdür [42]. Başlangıç koşullarına ve yapısal

dengesizliğe duyarlılık ile ilgili olarak kaotik davranış niteliklidir. Bu bir diferansiyel denklem, özyinelemeli bir haritalama ya da bir zaman serisi olarak verilir. Bu nedenle, kaotik davranışı rastgele gürültüden algılayabilen herhangi bir sistemde Kaos tespit etmek için analitik yöntemlere sahip olmak gereklidir. Öte yandan, bu yöntem sistemin Kaosunu ölçebilmelidir. Sistemin davranışının kaotik koşullarda başlangıç değerlerine olan duyarlılığının açıklaması, Lyapunov'un üsünü tanımlayarak mümkündür. Kaos koşullarında birbirine yakın iki yolun başlangıçta d kadar uzak olması durumunda, t andaki uzaklığı aşağıdaki gibi olacaktır:

$$d(t) = d \cdot e^{\lambda t} \quad (1)$$

λ Lipanov parametresi denir. Bir sistemin Kaosunda, elde edilen en büyük Lipunov üsü sıfırdan büyük olmalıdır ve eğer bu sayı birden büyükse, maksimum tahmin uzunluğu zaman serilerindeki verilerin örnekleme sıklığından daha kısadır. Bu nedenle, sadece sıfır ile bir arasındaki Liapunovlu sistemlerin kaotik bir tahmini vardır. Üsü birinden çok küçükse, uzun vadeli tahmin kesin olarak mümkündür [43].

Bu çalışmada, Liapunov üsü hesaplama yöntemi, zaman serilerinin mevcut olan bir sistem için ele alınmıştır.

1.4.1. Liapunov Görünümünü Kullanarak kaotik Zaman Serileri Davranışını Tanımlamak

Liapunov'un görüşlerinin kullanılması, zaman serilerinin trafik hacminin zaman serileri de dahil olmak üzere öngörülebilirliğini belirlemek için en yaygın yöntemdir. Pozitif üstel ıraksama belirtisi ve negatif üstel yakınsama belirtisidir. Üstelin mutlak değerinin büyüklüğü yakınsama veya ayrıklık oranın göstermektedir [44].

$x_0, x_1, x_2, \dots, x_k$ dizisinin varsayalım ve aralarındaki zaman aralığı $t_n - t_0 = n\tau$ dan elde edilir. Bu τ , ardışık iki ölçüm arasındaki aralıktır. Sistemin kaotik bir davranışı varsa, aralarındaki farkların aralığına göre bitişik yolların ayrılığı şu şekildedir:

$$d_0 = |x_j - x_i| \quad (2)$$

$$d_1 = |x_{j+1} - x_{i+1}|$$

$$d_2 = |x_{j+2} - x_{i+2}|$$

⋮

$$d_n = |x_{j+n} - x_{i+n}|$$

n arttıkça, d_n Katlanarak büyür (en azından ortalama olarak) bu nedenle,

$$d_n = d_0 e^{\lambda n}$$

$$\lambda = \frac{1}{n} \ln \frac{d_n}{d_0} \quad (3)$$

Yukarıdaki denklem, bir zaman serisi için Liapunov üsteli vermektedir. Eğer bir zaman serisinin liapunov üstel değeri, sıfırdan daha büyük bir mikar elde edilirse, verilen zaman serileri rastlantısal değildir ve kaotik bir modelde açıklanabilir, bu nedenle tahmin edilebilmektedir.

1.5. VİSSİM

Sinyalize kavşak, her yönedeki gerginliğe ayrılan belirli sinyal zamanlaması yardımıyla serbest trafik akışına yardımcı olmaktadır. Bu, ilgili iş miktarının çok daha az olduğu serbest ve kolay trafik anında yardımcı olmaktadır. Sinyalsiz bir kavşak durumunda, trafiği fazla etkilemeden trafiğin kolay akışını sağlamak için bir trafik polisi görevlendirilecektir. Sinyalize kavşak ile ilgili önemli terimler aşağıdaki gibidir:

- Döngü uzunluğu: yeşil, kırmızı ve sarı olan sinyalin üç fazı dahil olmak üzere bir tam devrenin süresini gösterir.
- Kuyruk uzunluğu: kaldırımda belirli bir mesafe işaretlenerek not edilir. Sinyal kırmızıya döndüğünde, üretilen kuyruğun uzunluğu not edilir.
- Yeşil zaman: Bu aşamada araçların durmuş konumdan hareket etmelerine izin verilen döngü uzunluğunun bir parçasıdır.
- Yeşil oran: yeşil zamanın tüm döngü uzunluğuna oranına denir (g / c).
- Kapasite: Herhangi bir alanın veya kavşağın kapasitesi, mevcut trafiğin, yolun mevcut durumu tarafından fazla zorluk çekmeden karşılanıp karşılanmadığı olarak tanımlanabilir. Seçilen yoğunluğun kapasitesinin çok daha az olduğu bazı koşullar ortaya çıkar ve bu nedenle tıkanıklık ve gecikme gibi çeşitli sorunlara yol açar.

- Servis Düzeyi: Servis seviyesi, kavşağın gerekliliği ne kadar iyi karşıladığı ve aracın geçerli koşuldan etkilenip etkilenmediği olarak tanımlanabilir. Bu parametre esas olarak trafiğin ve bunun için davranışının ele alınması durumunda belirli bir kavşakın nerede durduğunu belirlemeye yardımcı olur.
- PCU: Binek Araç Eşdeğeri (PCE) veya Binek Araç Ünitesi (PCU), bir karayolu üzerindeki trafik akış hızını değerlendirmek için Ulaştırma Mühendisliğinde kullanılan bir metriktir.
- Homojen Trafik Akışı: Homojen bir trafik akışı, bir şerit sistemini takip ettikleri ABD, İngiltere gibi belirli ülkelerde önemli ölçüde görülür ve homojen bir araç kompozisyon sınıfından oluşur.
- Heterojen Trafik Akışı: Hindistan'da olduğu gibi, yolda hız, boyut vb. Farklı özelliklere sahip ve yolda seyahat eden farklı araç sınıflarından oluşur. Bu durumda yolun belirlenmesi, trafiği bozarak zahmetli olacaktır. Heterojen trafik durumunda, yanıl hareket oldukça yüksektir, homojen trafik durumunda durum böyle değildir.
- Şerit gruplaması: Sinyalize edilmiş bir kavşak şeridinde gruplandırma önemli bir husus olacaktır, çünkü belirli şeritte hareket eden araç sayısı ve dokuma işlemi dikkate alınmalıdır. Çalışma için en iyi sonuçlarla şerit gruplandırması yapılacaktır. Bazı yerlerde, her segmentin kendisi farklı taşıt hareketleri gruplarından oluşur.

Karayolu Kapasite El Kitabı (HCM) [45] prosedürleri gibi geleneksel yaklaşımlar genellikle kuyrukların ve aşırı doygun koşulların sistem üzerindeki etkilerini yeterince yakalayamaz. Araç veya trafikle çalışan sinyal kontrol stratejileri, çeşitli ülkelerin tasarım kılavuzlarında iyi bir şekilde belgelenmiştir. Bununla birlikte, farklı stratejilerin etkinliği gerçek test alanlarında değerlendirilemez. Çok sayıda olasılık, mevcut denetleyicilerin sınırlamaları, yasal kısıtlamalar ve kullanıcılar tarafından kabulü nedeniyle stratejilerin çeşitliliği ve her bir stratejinin parametre ayarları saha denemelerinde test edilemez. Bu tür kısıtlamalarda simülasyonun değerli bir araç olduğu kanıtlanmıştır. Ekonomik ve çevresel nedenlerden ötürü, sinyal kontrolü belirli bir siyasi çerçeve içinde mümkün olduğunca iyi olmalıdır. Araçlardan araçla çalıştırılan sinyal kontrolünün kalitesini değerlendirmeleri istenir. Trafiğin stokastik özellikleri nedeniyle bu detaylı değerlendirme için analitik formül uygulanamaz. Gelişmiş kontrol stratejileri, örneğin otobüs / tramvay öncelikli programlarında gerekli olduğu için tek araçlara yanıt verir. Varış zamanlarındaki bu

dalgalanmalar, tamamen analitik yöntemlerle yeterince doğru bir şekilde modellenemez [46]. VISSIM trafik tahrikli kontrol sistemlerinin tasarımı için yenilikçi bir simülasyon aracıdır. Genel amaçlı bir bilgisayar tabanlı trafik simülasyon sistemi olarak VISSIM bağlantıları, kavşakları ve “küçük” ağları yüksek düzeyde detaylandırır. Bununla birlikte, bu çalışmada VISSIM'in sinyal kontrolü için bir simülasyon aracı olarak yeteneklerine odaklanmaktadır. VISSIM, işlevsel olarak sınıflandırılmış tüm karayolları ve toplu taşıma operasyonlarını analiz etmek için geliştirilmiş mikroskopik, zaman adımı ve davranış tabanlı bir simülasyon modelidir [47]. VISSIM, tipik bir koridorda bulunan entegre yol ağlarının yanı sıra genel amaçlı trafik, otobüsler, hafif raylı sistem, ağır demiryolu, kamyonlar, yayalar ve bisikletlilerden oluşan çeşitli modları modelleyebilir. Model, 1970'lerin başında Almanya'nın Karlsruhe Üniversitesi'nde geliştirildi. VISSIM'in ticari dağıtımı 1993 yılında başlamıştır ve bakımını yapmaya devam eden PTV Transworld AG tarafından devam etmektedir. [48], [49] ve [50].

1.5.1. VISSİM Yapısı

Model iki ana bileşenden oluşur: (1) Simülatör ve (2) Sinyal durumu üretici (SSG). Simülatör trafik üretir ve kullanıcının grafik olarak ağı oluşturduğu yerdir. Kullanıcı, havadan bir fotoğrafı veya çalışma alanının şematik çizimini simülatöre aktararak başlar. Ardından, kullanıcı ağı “çizmeye” ve nitelikleri uygulamaya başlar (Şerit genişlikleri, hız bölgeleri, öncelik kuralları vb.). Simülatörde bağlantılar kullanılmasına rağmen, VISSIM'in geleneksel bir düğüm yapısı yoktur. [51]. Düğüm eksikliği, kullanıcıya bir kavşak veya kavşaktaki trafik işlemlerini (örn. Verim koşulları) ve araç yollarını kontrol etme esnekliği sağlar. SSG, simülatörden ayrıdır. Sinyal kontrol mantığının bulunduğu yer burasıdır. Burada kullanıcı, sinyal kontrol mantığını tanımlama ve böylece bir sinyal kontrol cihazı üreticisinin ürün yazılımında bulunan her türlü kontrol mantığını taklit etme yeteneğine sahiptir [52]. SSG, kullanıcının aşağıdakiler dahil ancak bunlarla sınırlı olmamak üzere sinyal işlemlerinin etkilerini analiz etmesine izin verir: sabit zaman, harekete geçirilmiş, uyarlanabilir, geçiş sinyali önceliği ve rampa ölçümü. Simülatörde sabit zaman kontrolünün uygulanabileceğini unutmamak önemlidir. SSG, dedektör bilgilerini her adımda simülatörden okur. Dedektör bilgisine dayanarak, SSG sonraki zaman adımı sırasında sinyal ekranının durumuna karar verir [53].

1.6. Bulanık Mantık

Bulanık mantık kavramları ve terminolojisi, 1965 yılında Lotfi A. Zadeh tarafından ortaya çıkmıştır [54]. Zadeh makalesinde bulanık kümeler ve üyelik işlevlerini ileri sürmüş ve bulanık kümeler üzerinde kuramsal işlemler ile birkaç tanım geliştirmiştir. İki değerli veya iki değerlikli mantıkta, bir ifadenin doğruluk değeri 0 "yanlış" veya 1 "doğru" dir. Oysa çok değerli mantıkta, ifadelerin gerçek değerleri, örneğin 0; 1/2 ve 1 olabilir. Bulanık mantıkta, bir doğruluk değeri 0 ile 1 arasında herhangi bir gerçek sayı varsayabilir. Kısaca, bulanık mantık çok değerli mantıktır. Daha geniş anlamda, bulanık mantık, çok değerli mantığın bir uzantısıdır ve belirsiz kavramları modelleyen çeşitli yöntemleri ifade eder.

Bulanık kümeler, belirsiz doğal dil terimleri için belirli matematiksel bir önerme sağlar [55]. Bulanık küme, bulanık kümenin elemanları kısmen de içerebileceği şekilde klasik ya da net bir kümenin genelleştirilmesidir; Burada gevrek bir kümenin x elemanı ya kümeye aittir veya değildir.

Karakteristik fonksiyon:

$$X_A: U \rightarrow \{0,1\} \quad (4)$$

Bir A gevrek kümenin, belirli bir U evrensel kümenin elemanlarını 0 veya 1'e eşleştirir ve bu şekilde tanımlanır:

$$X_A(x) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases} \quad (5)$$

Böylece net bir kümenin karakteristik işlevi, $A \subset U$ net kümenin üyeleri ve üye olmayan elemanların arasında ayırım yapar [56].

S bulanık küme şu şekilde tanımlanır:

$$S = \{(x, \mu_S(x)) | x \in U\} \quad (6)$$

Her $x \in U$, bir üyelik derecesile S 'e atanır ki üyelik fonksiyonuyla ölçülen: [57].

$$\mu_S: U \rightarrow [0,1] \quad (7)$$

U her zaman gevrek bir kümedir ve burada $U \subset R$ ile sınırlıdır. Üyelik fonksiyonu $\mu_s(x)$, x tamamen S ait olmadığı zaman sıfırdır; tamamen S ait olduğunda birdir ve x kısmen S ait olduğunda $0 < \mu_s(x) < 1$ dir. Zadeh, daha genel bir düzenlemede, üyelik fonksiyonu aralığının uygun bir kısmen düzenlenmiş küme olabileceğini, ancak aralığın birim aralığına $[0;1]$ kısıtlanması için uygun ve yeterli olduğunu belirtmektedir.

Bir bulanık S kümesindeki belirsizlik derecesi μ_s , y ile tam olarak temsil edilmektedir. Kesin kelimesi aslında üyelik işlevi S 'teki özensizliği (kesin olmamayı) tanımlar. Üyelik fonksiyonu μ_s ne kadar çok karakteristik işlevi (2.2) andırıyorsa (değeri yüksekse), bulanıklık da o kadar düşük olur. Benzer şekilde, S , 0 veya 1'den farklı olduğu sürece, S kümesi daha bulanık kabul edilir.

Üyelik fonksiyonlarına ait çeşitli formlar, sonraki bölümlerde tanıtılmış ve her biri ile ilgili yeterli açıklamalar yapılmıştır.

1.6.1. Dilsel Değişkenler

Bulanık çıkarma, önermeler eğer X , S 'dir o zaman Y 'de, T 'dir. Burada X ve Y dilsel değişkenlerdir, S ve T dilsel değerlerdir. Bir dilsel değişken, dilsel değerleri varsayan bir değişkendir. Örneğin, "yaş" dilsel değerleri varsayan linguistik bir değişken olabilir "Genç", "orta yaş" ve "yaşlı". Bu dilsel değerlerin her biri kendi üyelik fonksiyonuna μ' ya sahip bulanık küme olarak gösterilebilir. Bir kişi kısmen bu kümelerin bir veya birkaçına aynı anda üye olabilir: 25 yaşındaki bir kişi hala bir dereceye kadar genç, ancak bir dereceye kadar da orta yaşlı olarak kabul edilir. Yaş'ın evrenin çakışmayan bölümlere, yani gevrek kümelerle bölünmesi hem zor hem de pratik olmayacaktır. Bulanık mantığın temeli, birçok gündelik kavramlar muğlak ve belirsiz olmasıdır. Örneğin, "gökyüzü mavi" gibi bir ifade, sadece bir dereceye kadar doğru olabilir. Gökyüzü gittikçe gri hale geldiğinde, bu cümledeki gerçeklik değeri azalır, ancak renk mavi olmazsa belirli bir anı bulmak zordur. Ayrıca, bazen kesin ifadelerin kullanılması gereksizdir. Bedzek [58], buna güzel bir örnek vermiştir: "Diyelim ki, kırmızı ışığa yaklaşırken, sürücü öğrencisine frenleri ne zaman uygulayacağını bildirmelisiniz. 'Yaya geçidinden 74 adım önce frene basmaya başlayın' mı dersiniz? Veya 'kısa süre sonra frenleri uygulayın' gibi bir talimat mı verirsiniz? Tabii ki ikincisini seçmek daha mantıklıdır; birinci talimatın uygulanması çok hassas ve tehlikelidir.

Bulanıklık üzerine birçok çalışma yapılmıştır. Örneğin Jurva [59], Klir ve Yuan [56], Kosko [60], Lee [61], Mattila [62]. Niskanen Orlovski [63], Turunen [64], Yager ve Filev [65], Ya da Zimmermann [56]. Çalışmalarına müracaat edebilirsiniz. Ne yazık ki, birçok yazar bulanık mantık ve ihtimaller konusunda fazla iyimser değildirler. Bulanık sistemlerin ardındaki felsefe "yeni" ve hesaplamadaki uygulaması geleneksel bilgi işlem sistemlerinden farklı olmasına rağmen, bulanık sistemlerin uygulandıklarında daha iyi sonuçlar verdiği açık değildir. Örneğin, bir kontrol sisteminin seçimi (bulanık, sinirsel, sinirsel bulanık ya da matematiksel optimizasyon vb.), sadece arkasındaki felsefeye değil'de, sistemin performansına dayandırılmalıdır.

1.6.2. Üyelik Fonsiyon Tanımları

S Bulanık küme (veya eşdeğer olarak, üyelik işlevi μ_s) eğer $x \in U$ da $\mu_s(x) = 1$ olursa, normaldir. Aksi halde, S (veya μ_s) normal dışıdır. Bu tez çalışmasında tüm bulanık kümelerin normal olduğu varsayılmıştır.

Destek küme:

$$Sup(S) = \{x | \mu_s(x) > 0, x \in U\} \quad (8)$$

Çekirdek küme:

$$cor(S) = \{x | \mu_s(x) = 1, x \in U\} \quad (9)$$

Bulanık küme S (veya μ_s), tek noktadan oluştuğunda tek modeldir.

1.6.2.1. Üyelik Fonsiyonlarının Türleri

Üyelik fonsiyonlarının en sık kullanılan türleri üçgen, yamuk, Gauss, genelleştirilmiş zil ve sigmoid üyelik fonsiyonudur. Bu tezde sadece Gauss üyelik fonsiyonu ele alınmıştır.

1.6.2.1.1. Üçgen Üyelik Fonksiyonu

$$\mu(x; p_1, p_2, p_3) = \begin{cases} \frac{x-p_1}{p_2-p_1}, & x \in [p_1, p_2] \\ -\frac{x-p_3}{p_3-p_2}, & x \in [p_2, p_3] \\ 0, & \text{diğer} \end{cases} \quad (10)$$

Veya [24],

$$\mu(x; p_1, p_2, p_3) = \max \left(\min \left(\frac{x-p_1}{p_2-p_1}, \frac{p_3-x}{p_3-p_2} \right), 0 \right) \quad (11)$$

Şekil1.a'da görülmektedir. (p_1, p_2, p_3) parametreleri X koordinatların üç köşeni belirler.

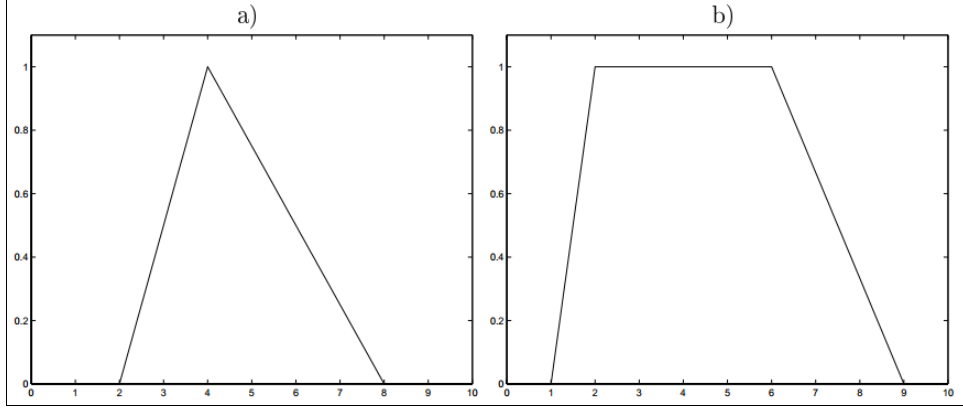
1.6.2.1.2. Yamuk Üyelik Fonksiyonu

$$\mu(x; p_1, p_2, p_3, p_4) = \begin{cases} \frac{x-p_1}{p_2-p_1}, & x \in [p_1, p_2] \\ 1, & x \in [p_2, p_3] \\ -\frac{x-p_4}{p_4-p_3}, & x \in [p_3, p_4] \\ 0, & \text{diğer} \end{cases} \quad (12)$$

Veya [20],

$$\mu(x; p_1, p_2, p_3, p_4) = \max \left(\min \left(\frac{x-p_1}{p_2-p_1}, 1, \frac{p_4-x}{p_4-p_3} \right), 0 \right) \quad (13)$$

Şekil 1. b'de görülmektedir. (p_1, p_2, p_3, p_4) parametreleri dört köşenin X koordinatlarını belirler. Yamuk üyelik işlevinin özel bir örneği, dikdörtgen üyelik işlevidir; $p_1 = p_2, p_3 = p_4$.



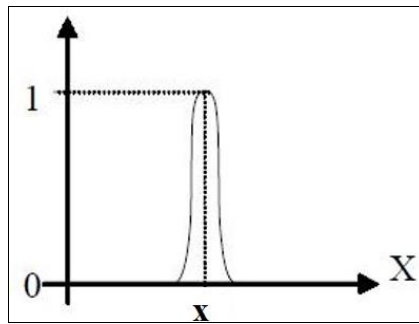
Şekil 1. (a,b) üçgen ve ymuk üyelik fonksiyonları ve parametreleri

1.6.2.1.3. Gauss Üyelik Fonksiyonu

Belirsiz, dilbilimsel terimleri temsil etmek için sıkça kullanılan bir diğer bulanık üyelik işlevi ise:

$$\mu_A^i(x) = \exp \left[-\frac{(x-c_i)^2}{2\sigma_i^2} \right] \quad (14)$$

c_i ve σ_i sırasıyla, bulanık küme A'nın merkezi ve genişliğidir. Gauss zili, orta değeri x ve $\sigma > 0$ değeri ile tanımlanır. σ ne kadar küçük olursa, zil o kadar dar olur.



Şekil 2. Gauss üyelik fonksiyonu

Gauss fonksiyonları, normal dağılımları tanımlamak için istatistiklerde, Gauss filtrelerini tanımlamak için sinyal işlemede, Gauss bulanıklaştırmaları için iki boyutlu Gauss'ların kullanıldığı görüntü işleminde ve ısı denklemlerini ve difüzyon denklemlerini çözmek ve matematikte Weierstrass dönüşümünü tanımlamak için matematikte kullanılır.

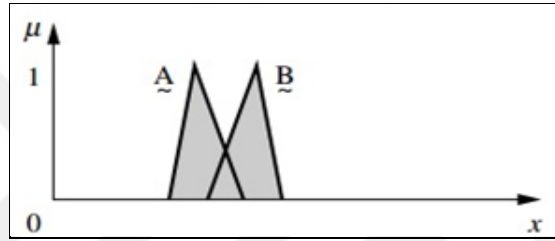
1.6.3. Bulanık Kümeler İşlemleri

Bulanık kümeler üzerinde en çok kullanılan işlemler tümleyen, kesişim ve birleşim işlemleridir. Bunların matematiksel yorumu aşağıda verilmiştir.

Burada “ $\vee$, *veya*” maksimum ve “ $\wedge$, *ve*” minimum anlamındadır.

➤ Birleşim:

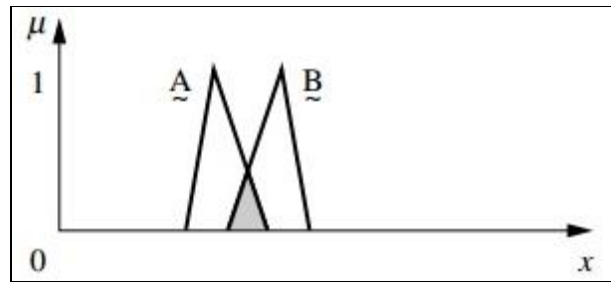
$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) \quad (15)$$



Şekil 3. Bulanık kümelerin birleşimi

➤ Kesişim:

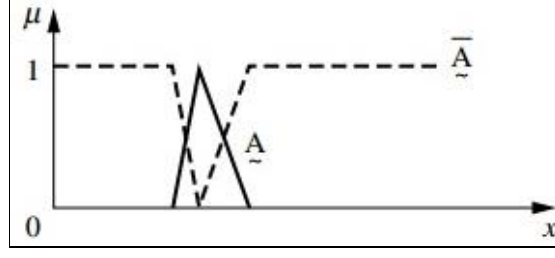
$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) \quad (16)$$



Şekil 4. Bulanık kümelerin kesişimi

➤ Tümleyen:

$$\mu_{\tilde{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x) \quad (17)$$



Şekil 5. Bulanık kümelerin tümleyeni

Not: Klasik kümelerin bütün özellikleri ayırık kuralı ve tümleyen ile birleşim dışında aynen bulanık kümeler içinde geçerlidir. Yani:

$$\begin{aligned} A \cup \bar{A} &= X \\ A \cap \bar{A} &= \emptyset \end{aligned} \quad (18)$$

➤ **Eiştlik:** İki bulanık kümenin eşit olması için:

$$\mu_{\bar{A}}(x) = \mu_{\bar{B}}(x), \forall x \in E \quad (19)$$

Olması gerekir.

1.7. Belirsizlik Türleri

1.7.1. Olasılığa Karşı Bulanıklık

Bulanıklık ve olasılık arasındaki fark konusunda birçok tartışma olmuştur. Öncelikle, farklı belirsizlik türleri arasında bir ayrım yapılmalıdır. Olasılık, bulanık ya da başka bir nitelikte olabilir [67]. Terminoloji literatürde değişiklik gösterir ve bazı metinlerde belirsiz sözcük sadece olasılık belirsizliğini ifade ederken; belirsiz kelimesi genellikle bulanıklığı ifade etmektedir. Niskanen [68] terminolojiye bir açıklama getirmiştir ve birçok yazar sözleri kendi tarzında açıklamayı tercih etmektedir. "X, S dir" bir öneri olduğunu düşünelim, eğer olasılıklı belirsizlik doğasında var ise, X de S'den başka değerler alabilir, ancak bizim S tanımımız açıktır. Eğer X'in S olduğuna dair kanıtımız varsa, eğer S ve S' X karşılık gelen değerler ise o zaman X'in aynı anda S' olamayacağını bilinmektedir. Öte

yandan bulanık belirsizlikte, S'yi açıkça tanımlayamayız ve X aynı zamanda S 'olabilir. Örneğin,"yarın yağmur yağacak" önerisi, kelimenin her iki anlamında belirsiz olabilir. Olasılık açısından, yağmurun yağıp yağmayacağını henüz bilmiyoruz, ancak değişikliğin "yağmur" ve "yağmursuz" olduğunu biliyoruz ve doğru cevap ertesi gün bilinecektir. Ertesi günün hava durumu bizim algımızla kısıtlı olduğundan durum sadece belirsizdir. Bulanık belirsizliğinde, yağmur kavramını net bir şekilde tanımlayamıyoruz- zaman zaman hafifden çiselerse, ikisi de hem yağmur yağdığını ve yağmadığını söyleyebiliriz. Belirsizlik ertesi gün hala mevcuttur, çünkü belirsizlik kavram ile ilişkilidir.

Zadeh [54], daha önce ‘‘bulanık küme kavramının doğada tamamen istatistiksel olmadığını’’ söylemiştir. O zamandan beri, bulanıklık ve ihtimalin farklılıkları ve benzerlikleri hakkında bir takım tartışma devam etmektedir. Bedzek [69], [70] bir örnekte bulanıklık ve olasılık arasındaki farkı özetlemektedir. L'in içilebilir dilsel bir değer olsun. Bir haftalığına iki şişe su ile (A ve B) çöle gittiğinizi varsayalım. A şişesi için L'deki üyelik derecesi $\mu_L(A) = 0.91$ iken, B şişesi için L'ye ait olma olasılığı $P_L(B) = 0.91$ 'dir. Hangi şişeyi seçersiniz? Üyelik fonksiyonu değeri 0.91, A içeriğinin içilebilir sıvılara (örneğin su) oldukça benzediği anlamına gelir. 0.91 olasılığı, deneylerde, B içeriğinin %91 'inde tamamen içilebilir olduğu anlamına gelir, ancak diğer %9 'da, B içeriği içilebilir olmayacaktır (örneğin, B, hidroklorür asit içerebilir). Böylece, güvenli tarafta olmak için A'yı seçmelisiniz, çünkü bataklık suyu içebilir, ancak hidro klorür asidi içermez. Bedzek örneği [69], [70] devamında A ve B'nin içeriğini inceledik ve A'nın bira ve B'nin hidroklorür asidi içerdiğini keşfettik. Şimdi, A üyelik fonksiyonu değeri hala 0.91 iken, bu gözlemden sonra B'nin olasılık değeri 0.91'den 0'a düşüyor. Bedzek örneğini bulanıklık ve olasılık arasındaki felsefi farkla özetlemiştir [70]. Üyelik fonksiyonu, bir nesnenin (sıvı) kesin olmayan bir şekilde tanımlanmış bir mülke (içilebilir) olan benzerliğini temsil eder. Oysaki olasılık yoğunluk fonksiyonuna, bağlı frekanslar hakkında bilgi içerir.

1.7.2. Belirsizlik

Belirsizliğin bir yönü, Klir ve Folger [57] ve Klir ve Yuan [56]'da sunulan ve Kikuchi ve Pursula [71] de tartışılan belirsizliktir. " X, S dir. " önermesinin belirsizliği, X hakkındaki bilgilerimizin eksik olduğunu, ancak S'nin kendisinin açıkça tanımlandığı anlamına geliyor. Aynı tanımlama Bölüm 2.4.1'de tartışılan olasılık belirsizliği için de geçerlidir ve [71]'e göre olasılıksal belirsizlik bir belirsizlik türüdür. Diğer belirsizlik

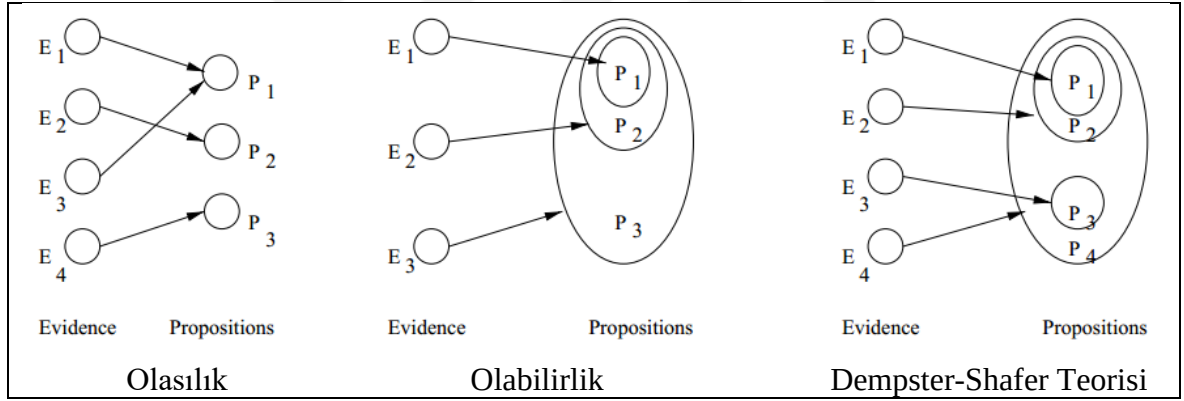
türleri olasılıksal belirsizlik ve olasılık ve olasılık belirsizliğinin birleşimidir. Belirsizlik için matematiksel bir çerçeve kanıt teorisidir. X hakkındaki bilgiler kanıttır. Belirsiz bir durumda, kanıt kısmen önermeyi desteklemektedir ve üç durum ortaya çıkabilmektedir [71]:

Birçok delil (örneğin, deney sonuçları) göz önüne alındığında, her bir kanıt sadece bir alt kümeyi göstermektedir. Bu tür belirsizlik bir olasılık ölçüsü ile ölçülür.

Delillerin gövdesi iç içe geçmiş kümelerle gösterilmektedir, bu nedenle bir delil önermenin tüm alt kümelerini desteklemektedir. Bu belirsizlik olasılık ölçüsü ile ölçülür. Olasılık dağılımı, bulanık kümenin üyelik fonksiyonu ile aynı sayısal değeri varsaymaktadır.

1 ve 2'nin birleşim belirsizliği, bir inanç önlemi ile ölçülür. Ölçüler Dempster-Schafer teorisinde bulunmuştur [57]. Bu teori olasılık ve olasılık teorilerini kapsamaktadır.

Bu durumlar Şekil 5'te karşılaştırılmıştır:

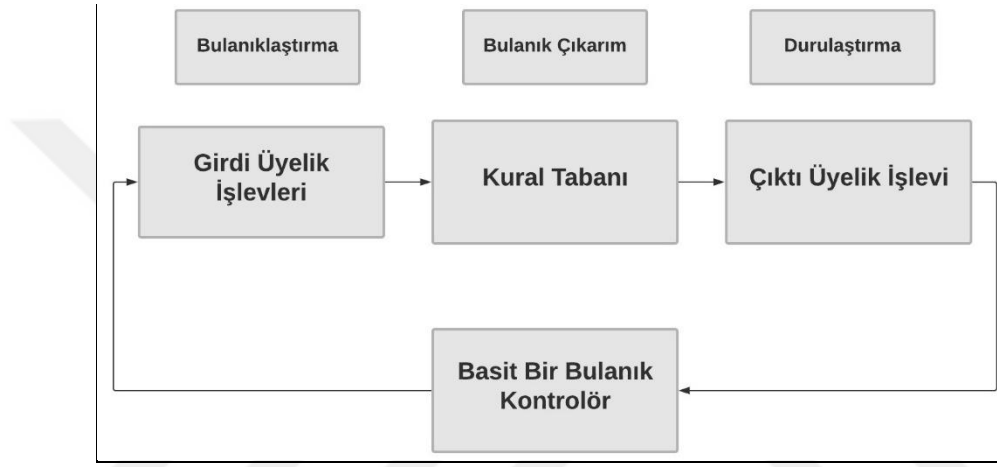


Şekil 6. Kanıt ve önermeler arasındaki belirsizlik durumundaki ilişkiler

Kikuchi ve Pursula [71], belirsizliklerin insan yargısı ve davranışıyla ilişkili olduğundan ulaşım sorunlarında belirsizlik önlemlerine örnek vermektedir. Gerçek dünyada, gözlemlenen veriler olasılıksal bir karaktere sahip olabilir, ancak insanlar genellikle onları olasılık dağılımı olarak işlemektedir. Örneğin, "Seyahat süresi T'den küçük" ibaresini düşünün; Kanıtlar, "seyahat süresi, $[T_1:T_2]$ aralığında" şeklinde olan seyahat süresi gözlemlerinden oluşur. Olasılık teorisinde, bir aralığı destekleyen bir kanıt parçası, içindeki aralıkları da destekler. Sürüş süresinin $[T_1:T_2]$ aralığında olması durumunda, seyahat süresinin daha geniş bir aralıkta olması da mümkündür $[T_1 - \delta_1, T_2 + \delta_2]$, $\delta_1 > 0$, $\delta_2 > 0$ şeklinde açıklanabilmektedir.

1.8. Bulanık Kontrol

Bulanık sistemlerin en pratik ve başarılı uygulaması bulanık kontroldür. Bulanık kontrol sistemlerinin geleneksel olanlara göre belirgin bir avantajı, bulanık kontrol sistemlerinin uzman bilgisini kullanma becerisidir. Bilgi, bir kural tabanı olarak ifade edilebilir; kurallar, "eğer X, S dir, o zaman Y, T'ye eşittir." şeklindeki önermelerdir; Burada S ve T bulanık kümelerdir.



Şekil 7. Basit bir bulanık kontrolör

Bulanık denetleyici Şekil 6'da görüldüğü gibi bulanıklaştırma, bulanık çıkarım ve bulanıklık giderme(durulaştırma) modüllerinden oluşur. Bulanıklaştırma modülü kontrolör altındaki sistemden kontrol girdi değişkenlerinin ölçümlerini alır. Onlar, üyelik fonksiyon değerlerine dönüştürülür. Duruma bağlı olarak, "Bulanıklaşma" sözcüğü, Madde 2.2.3'te açıklandığı gibi bulanık denetim sürecinde kullanılan üyelik işlevlerinin yapısını içerebilir. Devam eden bir kontrol sürecinde, üyelik fonksiyonları zaten oluşturulmuş ve kontrol süreçleri sırasında fonksiyonlar için en fazla yapılabilecek şey ince ayarlamalardır. Bulanık çıkarım motoru, bulanık kural tabanında saklanan kontrol kurallarını değerlendirir. Bulanık çıkarım, kural kesme kuvveti hesaplama, bulanık ima ve kural toplama işlemlerini içerir ve bunlar sonraki bölümlerde açıklanmaktadır. Bulanık çıkarımın sonucu, kontrol eylemi elde etmek için üyelik işlevleri önceden belirlenmiş bir veya birkaç çıktı bulanık kümesidir.

N kuraldan oluşan kural tabanı:

Eğer $X_1 = S_{11}$ ve ... ve ... veya $X_n = S_{n1}$ olursa ,

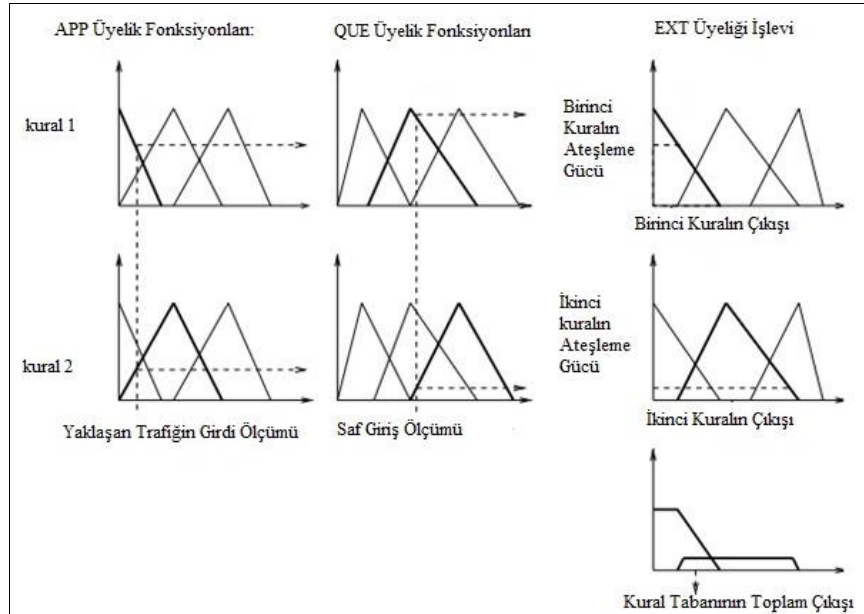
o zaman $Y_1 = T_{11}$ ve ... ve $Y_p = T_{p1}$

⋮

Eğer $X_1 = S_{1N}$ ve ... ve ... veya $X_n = S_{nN}$ olursa ,

o zaman $Y_1 = T_{1N}$ ve ... ve $Y_p = T_{pN}$ (20)

Her girdi değişkeni $X_j, j = 1, 2, \dots, n$ bir dilsel değer veya $S_{ji}, i = 1, 2, \dots, N$ bulanık küme olduğunu varsayalım ve her Y_k çıktı değişkeni bir dilsel değer $T_{ki}, k = 1, 2, \dots, p$ olsun. Bu bulanık kümeler için üyelik fonksiyonu sırasıyla $\mu_{S_{ji}}(x_j), \mu_{T_{ki}}(y_k)$ 'dir. X_j Değişkeninin aynı S_{ji} dilsel değerleri, i ve i' kuralları için $S_{ji} = S'_{ji}$ olmasını sağlayan birkaç kuralda görünebileceğini unutulmamalıdır. Aslında dilsel değerlerin sayısı N 'den küçük olabilir. Aynı uygulama, Y_k 'nin T_{ki} dilsel değerleri için de geçerlidir. Ayrıca, her girdi değişkeninin aynı sayıda dilsel değere sahip olmasına gerek olmadığı da unutulmamalıdır. Kural tabanındaki her kural, öncül olarak adlandırılan bir "eğer" bölümü ve sonuç olarak adlandırılan bir "o zaman" bölümü içerir. Her kurala bir kural ağırlık önemi m_i eklenebilir. Bu şekilde bazı kurallar diğerlerinden daha önemli görülebilir.



Şekil 8. İki kuralın bir kural tabanında bulanık çıkarım

Kural 1: "APP sıfır ve QUE birkaç ise, o zaman EXT sıfırdır ". Kural 2: " APP orta ve QUE uzunsa, EXT kısa ".

Hem kuralın öncüsünde hem de sonuç bölümünde, çeşitli bulanık kümeler, " ve " (kesişim) veya " veya " (birleşim) operatörleri kullanılarak birleştirilebilir. Bu çalışmada, öncelikde bir " ve " operatörü kullanılır ve sadece bir çıkış değişkeni Y vardır. Daha fazla çıktı değişkenine uzatma işlemi basittir. Şekil 8, iki girdi değişkeninin bulanık kontrol sisteminde yer alan basamakların bir örneğini vermektedir, bir çıktı değişkeni ve iki kural. Adımların her biri, aşağıdaki bölümlerde açıklandığı gibi birkaç farklı şekilde gerçekleştirilebilir. Göz önünde bulundurulanan sistem bulanık trafik sinyali denetleyicisidir. Giriş değişkenleri APP, yaklaşan yeşil yönden gelen trafik ve QUE, sıra kırmızı yöndür. Çıkış değişkeni yeşil sinyal uzantısı EXT'dir.

Bu örnekte, aşağıdaki iki kural kullanılır:

APP sıfır ve QUE birkaç ise, EXT sıfırdır,

APP orta ve QUE uzunsa, EXT kısadır.

Şekil 8'de her kuralda kullanılan dilsel değerler için üyelik fonksiyonu kalın bir çizgi ile çizilmiştir. Yaklaşan trafik ve kuyruk bir (x_1, x_2) giriş ölçüm çifti ile elde edilir. Noktalı yatay çizgiler üyelik fonksiyonu değerlerini göstermektedir: Kural 1' de $\mu_{sıfır}(x_1)$ ve $\mu_{biraz}(x_2)$, kural 2' den $\mu_{orta}(x_1)$ ve $\mu_{uzun}(x_2)$. Her kuralın kural frenleme güçleri, bu üyelik fonksiyonu değerlerine bağlıdır. Sistemin sayısal bir çıktı değerini üretmek için iki kuralın çıktıları bir araya getirilir ve bulanıklaştırılır. Bu adımların ayrıntıları ilerleyen bölümlerde tartışılmıştır. Bulanık kontrolün tamamen istatistiksel olmadığına dikkat edilmelidir. Belirli bir girdi, kontrol sistemi sabit tutulursa, daima aynı çıkışa neden olur. Bir istisna, öğrenme algoritması rassal arama motoru kullanan ayarlanabilir kontrolörlerdir. Bu algoritmalarda, denetleyicinin çıktısı rasgele bir miktarla sapmaktadır. Bu rassal keşif daha iyi kontrol performansı sağlayabilir. Bu öğrenme algoritmalarından biri, bu tezde kullanılan takviye öğrenmedir ve ayrıntılı olarak ele alınacaktır.

1.8.1. Bulanık Çıkarımda Adımlar

Atış (fren) Gücü Kuralı: fren gücü Kuralı w , kuralın öncülündeki bölümdeki koşulların ne kadar iyi olduğunu gösterir: bir x girdi, "fren (ateş)" kuralın bir dereceye $w \in [0,1]$ götürür. Yani bu izleyen bölümde çıkarılan sonuçların uygulanması derecesini belirler. Fren gücü kuralı, kuralda kullanılan bulanık kümeler için üyelik fonksiyonu

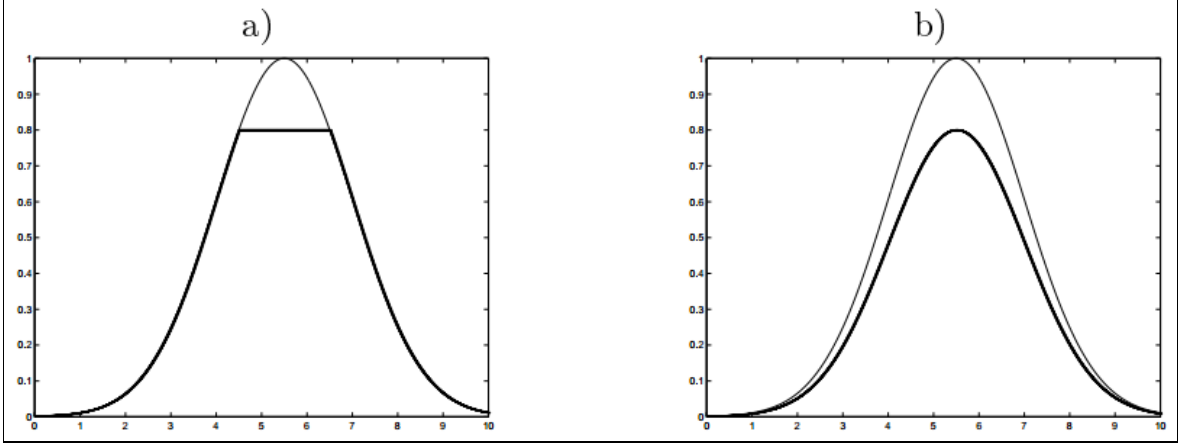
değerleri kullanılarak hesaplanmıştır ve formülü " ve " veya " veya " birleşimlerini kullanılıp kullanılmadığına bağlıdır. Modern Bulanık Edebiyatı " ve " (kesişim) için ürün birleştiricisini seçer çünkü düzgün ve bilgi israfından kaçınır. Dolayısıyla, fren gücü kuralı, önceki kuraldaki üyelik fonksiyonu değerlerinin çarpımı ile ilgilidir. Minimum birleştirici geleneksel bir seçenektir. Bir " veya " birleşimi, bir kontrolün öncülünde kural olarak sıklıkla kullanılmamaktadır, toplam varyantlarından biri maksimum operatör üzerinde tercih edilir ve kural öncülündeki üyelik fonksiyon değerlerinin bir toplamı kullanılır. Bir örnek olarak, Şekil 8'i düşünün. Kural 1 ve 2, kural öncülündeki koşulları birleştirmede "ve" birleştiriciyi kullanır. Kural 1'deki ateşleme gücü kuralı $w_1 = \min\{\mu_{sıfır}(x_1), \mu_{biraz}(x_2)\}$ dur ve kural 2' deki ateşleme kuvveti $w_2 = \min\{\mu_{orta}(x_1), \mu_{uzun}(x_2)\}$ dur. Ürün birleştirici kullanılıyorsa, kural ateşleme mukavemetleri sırasıyla üyelik fonksiyonu değerlerinin çarpımlarıdır.

Bulanıklık etkisi. Atış gücü kuralından sonra w hesaplanır, bulanık T kümesi kuralın sonucu bölümündeki, atış gücü kural ile belirtilen seviyede kesilir veya ölçeklendirilir. Karşılaştırma Şekil 9. a ve b, kırpma ve ölçeklendirme arasındaki farkı göstermektedir. Kırpma veya korelasyon-minimum çıkarım, elde edilen T' kümenin üyelik fonksiyonu $\mu_{T'}$:

$$\mu_{T'} = \min\{w, \mu_T\} \quad (21)$$

Kırpma bulanık kümenin orijinal şeklini korumamaktadır ve bu nedenle modern literatürde genellikle tercih edilmemektedir. Hesaplamalarda bazı sorunlar yaşanabilmektedir, sebebi ise şöyle açıklanabilir; minimum fonksiyonu tür evlenemez ve μ türevi daha sonra gerekli olacaktır. Ölçeklendirme veya korelasyon-ürün çıkarımı, üyelik fonksiyonu μ_T 'nin w atış gücü kuralı ile çarpılması anlamına gelmektedir.

$$\mu_{T'} = w \cdot \mu_T \quad (22)$$



Şekil 9. μ Üyelik fonksiyonu ve çeşitleri. a) Kırılmış çeşit b) ölçeklendirilmiş çeşidi ($w = 0: 8$).

Orijinal küme T ile aynı şekle sahip bir T' kümesi üretir. Ölçekleme, fonksiyonel olarak daha uygun bir iştir; çünkü ürün türevlenebilir bir fonksiyondur. Şekil 8'deki örneği tekrar düşünülürse bu durumda, korelasyon-minimum çıkarım kullanılır. Kural 1'de üyelik fonksiyonu EXT' den $\mu_{sıfır}$, w_1 seviyesinde kesilir ve kalan kümenin üyelik fonksiyonu ise $\mu_{sıfır'} = \min\{w_1, \mu_{sıfır}\}$. Karşılık olarak, Kural 2'deki geri kalan kümenin üyelik fonksiyonu $\mu_{kısıt'} = \min\{w_2, \mu_{kısıt}\}$.

Kural Toplama: Yukarıda açıklanan prosedürlerden biriyle elde edilen N ölçekli veya kırılmış bulanık kümeler $T'_i, i = 1, 2, \dots, N$ 'ın bir kümesi, artık kural tabanının toplam çıktısını temsil etmektedir. Her T'_i 'nin kendi üyelik fonksiyonu var $\mu_{T'_i}$. Bu bölüm, T'_i 'ların nasıl bir araya geldiklerini ve daha sonra durulaşma evresinde kullanılan bir $T' = \bigcup_{i=1}^N T'_i$ birleşimi oluşturduğunu açıklamaktadır. Bununla birlikte, bazı durulaştırma prosedürleri T'_i kümelerini kullanır ve algoritmaya bağlı olarak birliğe ihtiyaç duyulmamaktadır. Daha önce de tartışıldığı gibi, birlik toplulaştırması birçok farklı yolla gerçekleştirilebilir. Maksimum birleştirici (2.18), setlerin çakışmasını hesaba katmaz ve bu nedenle bilgileri yoksayar. Bazı bölgelerde üst üste gelen kümeler çakışırsa, bu bölge tabii ki kural set çıktısını en iyi açıklar ve çakışan değerlere daha fazla önem verilmelidir. Toplam birleştirici basitçe tüm üyelik işlevlerini ekler. Sonuçta ortaya çıkan kümenin üyelik fonksiyonu, Şekil 4'te görüldüğü gibi 1'den büyük değerleri varsayabilir. Bu pratik uygulamalarda bir sorun değildir, ancak genellikle üyelik fonksiyonu değerlerinin 0 ile 1 arasında sınırlandırıldığı varsayılmaktadır. Sınırlı toplam (2.20) veya olasılıksal toplam (2.21), 1'den küçük veya eşit olacak şekilde sınırlandırılır, ancak aynı alan üzerinde

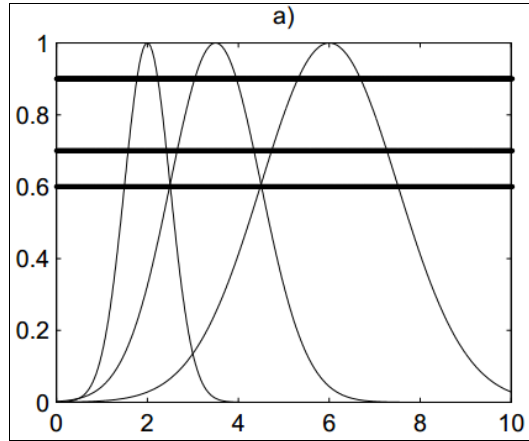
yoğunlaşan çıktı değerlerini ortaya koymazlar. Bu konuda, toplamı birleştirici (2.19) en fazla bilgiyi korur.

Şekil 2.'de çeşitli kural toplama yöntemleri sunulmuştur. Şekil 10. a'da, üç Gauss μ_1, μ_2 ve μ_3 üyelik fonksiyonu, bir çıkış dilsel değişken Y için, w_1, w_2 ve w_3 kuralı ateşleme gücü ile birlikte gösterilmiştir. Şekil 11. b'de, bu üyelik fonksiyonları ilk önce w_i seviyelerinde kesilir ve μ_i ve w_i 'in minimumun da sonuçlanır ve ardından birleşme maksimum olarak oluşturulur. Bu min-max ayrışımı, bulanık edebiyatta geleneksel tercihtir. Şekil 11. c, ürün-toplama ayrışmasını ki burada μ_i üyelik fonksiyonları ilk önce w_i lar ile çarpıldığını (Bkz. Formül (2.24)) ve kalan kümeler ile toplandığını (bkz. Formül (2.19)) göstermektedir. Şekil 2.8 d'de, sınırlanmış toplam ve olasılıksal toplam kullanılarak aynı işlem yapılır, Ölçekli üyelik fonksiyonları için formüller:

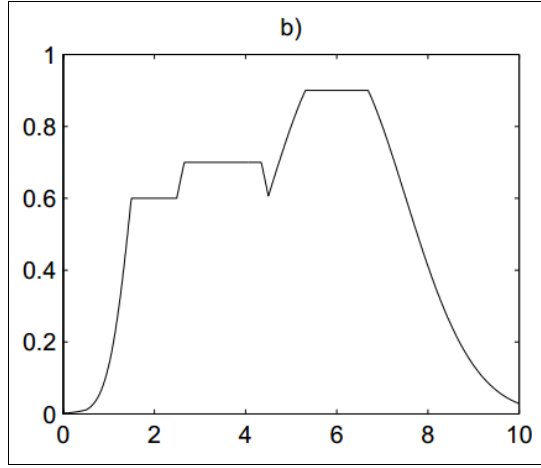
$$\mu_{T'} = \min\{1, \sum_{i=1}^N w_i \mu_{T_i}\} \text{ (sınırlanmış toplam)} \quad (23)$$

$$\mu_{T'} = 1 - \prod_{i=1}^N (1 - w_i \mu_{T_i}) \text{ (olasılıksal toplam)} \quad (24)$$

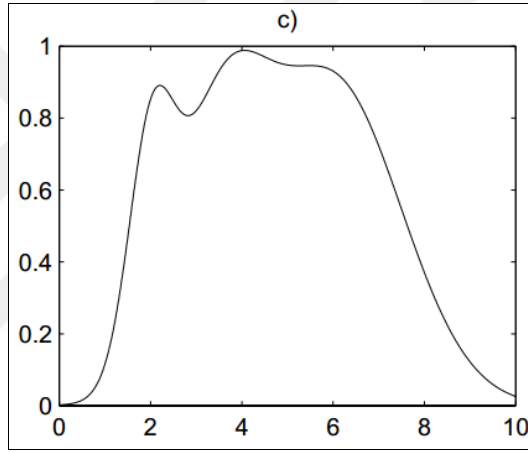
Not: Bu durumda Şekil 12 d'deki sınırlanmış toplam, Şekil 13 c'deki toplamla aynıdır, çünkü üyelik fonksiyonlarının toplamı 1'i aşmamaktadır.



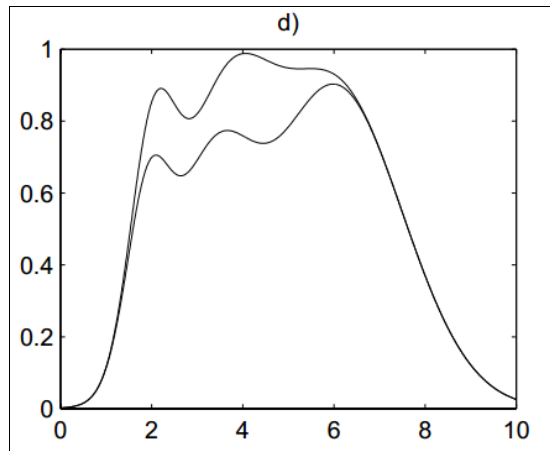
Şekil 10. (a) Üyelik fonksiyonları: $\mu_{T_1}, \mu_{T_2}, \mu_{T_3}$ ve kural ateşleme gücü $w_1 = 0.6, w_2 = 0.7, w_3 = 0.9$



Şekil 11. (b) Min-maks ayrışımı



Şekil 12. (c) işlenmiş (ürün)-toplam ayrışmış



Şekil 13. (d) Üst eğri: Sınırlı toplamı kullanarak ürün-toplam ayrışım. Düşük eğri: olasılıksal toplamı kullanarak ürün- toplam ayrışımın göstermektedir.

Eğer m_i ağırlık önemleri kullanılırsa, her $\mu_{T'_i}$ üyelik fonksiyonu önce m_i ile ölçeklendirilmeli ve sonuç normalize edilmelidir:

$$\mu_{T'_iyeni} = \frac{m_i \mu_{T'_i}}{\sum_{i=1}^N m_i}$$

Bu ölçeklendirme ayrıca durulaşma fazında, m_i ağırlıkları algoritmaya bağlı olarak çıkış kümeleri T_i veya T'_i 'nin birleştirilmesi sürecinde kullanılır. Şekil 8'deki örneğimize geri dönersek, ölçekler bulanık kümeler sıfır' ve kısa' EXT' den değerleri bir araya getirilmez, ancak ayrı olarak sunulur, böylece kümeler durulaşma fazında olduğu gibi kullanılabilir.

1.8.2. Durulaştırma

T' birliği, yukarıda açıklanan yollardan birini kullanılarak tanımlandıktan sonra, çıkış kontrol değişkeni Y 'ye sayısal bir değer elde etmek için durulaşmalıdır. T' birliğin üyelik fonksiyonunun durulaşması için birkaç yöntem vardır. Alan merkezi (COA) ve Maksimizanın ortalaması (MOM) ve bunların varyasyonları ve bunlar da devamda açıklanmaktadır. Aşağıda bazı durulaştırma yöntemleri sunulmaktadır:

Alan merkezi yöntemi. Alan merkezi durulaştırma yöntemi, T' bulanık kümenin merkezini hesaplar:

$$y^* = \frac{\int_U y \mu(y) dy}{\int_U \mu(y) dy}. \quad (25)$$

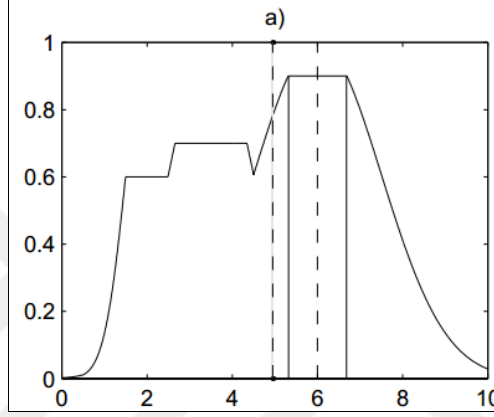
Kolaylık açısından burada $\mu_{T'}(y)$ yerine $\mu(y)$ kullanılmıştır. Bulanık kümelerin birleşimine uygulanan COA örneği Şekil 10. a'da görülmektedir. Ayırık kümelerde, μ sonlu evrensel bir $\{y_1, y_2, \dots, y_k\}$ kümede tanımlandığında, formül:

$$y^* = \frac{\sum_{i=1}^k y_i \mu(y_i)}{\sum_{i=1}^k \mu(y_i)}. \quad (26)$$

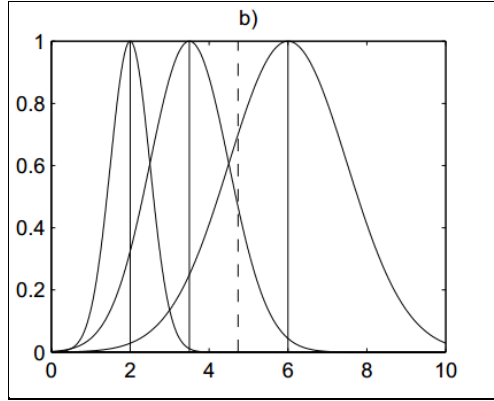
Not: Kesirler, $[0,1]$ aralığında olduğu için bir olasılık dağılımı oluşturur ve birlik için birleştirildi. Bu nedenle COA ile elde edilen durulaştırılmış değer, Y çıktı değişkeninin beklenen değeri olarak yorumlanabilir [56].

$$\frac{\mu(y)}{\int_U \mu(y) dy}$$

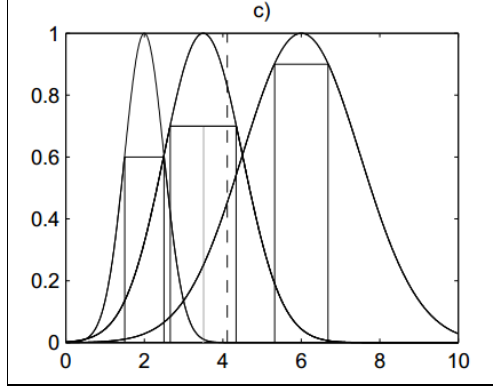
$$\frac{\mu(y_i)}{\sum_{i=1}^k \mu(y_i)}, \quad i=1, 2, \dots, k \quad (27)$$



Şekil 14. (a) μ_1, μ_2, μ_3 Üyelik fonksiyonlarının Durulaştırması (bk. Şekil 10)



Şekil 15. (b) Ağırlık olarak w_i ve V_i 'nin kullanıldığı LCOA yöntemi



Şekil 16. (c) Ağırlık olarak w_i 'nin kullanıldığı LMOM yöntemi

Şekil 10. a'da Kuralı ateşleme gücü $w_1 = 0.6, w_2 = 0.7, w_3 = 0.9$ dır.

a) COA ($y^* = 4.95$ Soldaki kesikli çizgi) ve MOM ($y^* = 6.00$, sağdaki kesikli çizgi) yöntemleri bir üyelik fonksiyonu birliğine uygulanmaktadır. Katı dikey çizgiler M'nin infimum ve supremumunu göstermektedir (Formül (2.35)).

b) Ağırlık olarak w_i ve V_i 'nin kullanıldığı LCOA yöntemi. Kesikli çizgi, $y^* = 4.73$ 'ü gösterir.

c) Ağırlık olarak w_i 'nin kullanıldığı LMOM yöntemi. Katı dikey çizgiler, M_{w_i} 'nin infima ve suprema'sını göstermektedir (Formül (2.39)). Kesikli çizgi $y^* = 4.11$ gösterir.

Yerel Alan Merkezi. Yerel Bölge Merkezi COA yönteminin "yerel" ve katma yaklaşımı, Kosko

[60] tarafından önerilmiş ve Jang ve Sun [72] ve Zimmerman [55]'da belirtilmiştir. Bu yaklaşımda, T0'ın merkezi, sonuçtaki takımların merkezlerinin dışbükey toplamı olarak ifade edilir. Toplam çıktı bulanık T' kümesi aslında gerekli değil-onun yerine, her kural ayrı ayrı durulaştırılır. Durulaştırılmış çıktı:

$$y^* = \frac{\sum_{i=1}^N m_i w_i V_i y_i}{\sum_{i=1}^N m_i w_i V_i} \quad (28)$$

Burada N kural tabanındaki kuralların sayısıdır, m_i , i'nin kuralın ağırlığıdır (eğer kural önem ağırlıkları kullanılırsa), w_i , i'nin kuralın ateş gücüdür, V_i (ölçeklenmemiş ve kesilmemiş) T_i sonuç kümenin hacmidir. Ve y_i , merkezin ortak formülün (2.28) kullanılarak hesaplanan T_i 'nin merkezidir. Bu nedenle, daha sonra bölüm kümesi ağırlık

merkezlerinin ağırlıklı bir ortalaması hesaplanır, Her ağırlık kuralın öneminin bir ürünüdür, kural ateş gücü ve sonraki bölüm hacim kümesi.

Bu, T' değerinin ürün-toplama ayrımı kullanarak elde edilen T' 'un merkezin almakla eşdeğerdir, ancak bu şekilde merkezin hesaplanması daha hızlıdır: Çıktı küme hacimleri ve küme centroid'ler önceden hesaplanabilir ve bir bakış (sonra-bakma) tablosunda saklanabilir.

Not: Çıktı bulanık kümesinin alanı, sadece eğer toplam birleştirici kullanılarak oluşturulmuşsa LCOA formülünün COA formülüne eş değer olduğunu unutmayın. Elbette maksimum, sınırlı toplamı veya olasılıksal toplamı kullanarak oluşturulan alanlar için merkez kuralı kullanılabilir, ancak daha sonra başka bir sorunla karşı karşıya kalınacaktır: kümelerin çakışması (üst üste gelmek) kısmen ya da tamamen gözardı edilecektir.

LCOA durulaştırmanın bir dezavantajı, sonraki kümelerin haciminde ağırlık ortalamaları kullanılmasıdır. Bu nedenle büyük kümelere küçük kümelerden daha fazla önem verilmektedir. Bulanık küme ne kadar genişse o kadar çok belirsizliğe sahip olur. İnce kümeler daha kesin bilgiler gösterir ve geniş kümelerden daha fazla ağırlık verilmelidir. Örneğin, kümelerin hacimlerinin tersi, onların ağırlıkları olarak kullanılabilir. LCOA'da, durum tam tersidir.

Bu nedenle amaç, büyük ve belirsiz kurallara vurgu yapmaktan kaçınmak ve diğer taraftan çakışan alanları hesaba katmaktır. Bir çözüm olarak, her çıktı kümenin herhangi bir ölçeklendirme veya kırpma olmaksızın durulaştırmaktır ve daha sonra y_i durulaşmış değerlerin ağırlıklı ortalamasını hesaplamaktır; w_i , kuralı çalıştırma kuvveti ağırlık olarak kullanır.

$$y^* = \frac{\sum_{i=1}^N m_i w_i y_i}{\sum_{i=1}^N m_i w_i} \quad (29)$$

Not: Burada, V_i küme haciminin hesaplanmasına gerek olmadığı için basit hesaplamalara neden olur. m_i Kural önem ağırlıkları isteğe bağlıdır. V_i Küme hacimlerin kullanılmaması nedeniyle centroid hesaplamasına (2.32) eşdeğer değildir.

Şekil 11 b'de, ağırlıklı ortalama ile birlikte üç üyelik fonksiyonu ve onların merkezleri gösterilmektedir. Bu örnekte m_i kural önemi kullanılmadı. Üyelik fonksiyonları Şekil 10 a'daki ile aynıdır, ancak artık w_i , $i = 1,2,3$ seviyelerinde kırılmamaktadır. Onun yerine, w_i , küme hacimi V_i ile birlikte ağırlık olarak kullanılır.

Maxima'nın ortalaması durulaştırma. Bu yöntemde maxima metodunun merkezi de denir, bu yöntemde, y^* nin en büyük ve en küçük değerin ortalamasıdır ki μ_y orada en büyük değerine ulaşır.

$$y^* = \frac{\inf M + \sup M}{2} \quad (30)$$

Sürekli μ lar için:

$$M = \{y \in U \mid \mu(y) = \max_y \{\mu(y)\}\} \quad (31)$$

Birleşmiş bulanık kümelere uygulanan MOM örneği Şekil 10. a'da sunulmuştur. Ayırık durumda, y^* , U söylem evreninin maksimum üyelik derecelerine sahip tüm değerlerinin ortalamasıdır [56].

$$y^* = \frac{1}{|M|} \sum_{y_i \in M} y_i \quad (32)$$

$|M|$, M 'deki elemanların sayısıdır.

MOM yöntemi, Şekil 15. b' deki gibi maksimal bir M platosuna sahip oldukları için, w_i veya 1 düzeyinde kırılmış çıkış bulanık kümeleri T' için uygundur. Bunun yerine, Bulanık Küme T' , Şekil 16. c'de tek modalıye, maksimum tekdir ve M sadece bir noktadan oluşur. Bu durumda, MOM yöntemine mod durulaştırma yöntemi denir, çünkü durulaştırılmış y^* değer $\mu(y)$ 'nin maksimumuna ulaştığı tek noktada olacaktır.

$$y^* = \operatorname{argmax} \mu(y) \quad (33)$$

Klir ve Yuan [56], eğer M bağlı değilse (yani, M birkaç aralıktan $[y_1, y_2]$ oluşur), y^* , uzunluğu sıfır aralıkları da dahil olmak üzere, M 'de bulunan tüm aralıkların ortalama değerlerinin aritmetik ortalaması olarak tanımlanabilir. Alternatif olarak, y^* , aralıkların ortalama değerlerinin ağırlıklı bir ortalaması olarak tanımlanabilir, burada ağırlıklar, aralıkların göreceli uzunlukları olarak yorumlanmıştır. Maksimum yöntemin ortalaması, M 'nin dışında kalan $\mu(y)$ 'leri gözardı etmenin bir dezavantajı vardır. Sadece maksimum üyelik derecesi olan çıktı değerleri dikkate almaktadır.

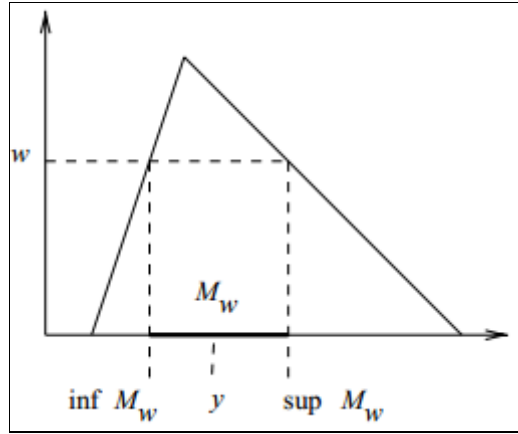
Maksimumun Yerel Ortalama (LMOM) Durulaştırma Yöntemi. Berenji ve Khedkar [73], yerel ortalama maksima (LMOM) durulaştırma yöntemini, her çıktı bulanık kümesi, çıktı kümelerin birleşimine MOM uygulamak yerine MOM metodunun adaptasyonu kullanılarak durulaştırılırı önermektedirler. W kural ateş gücü olsun. Eğer μ üyelik fonksiyonu çıktı bulanık kümesi için kesinlikle monoton olursa, durulaştırılmış çıktı y^* sadece $\mu^{-1}(w)$, μ 'nun tersidir. Üyelik fonksiyonları nadiren monoton olduğundan, matematiksel bir ters mevcut olmayabilir. Bu durumda y^* , Şekil 14'deki gibi w 'yi aşan $\mu(y)$, y kümesinin merkez noktası olarak tanımlanır:

$$y^* = \frac{\inf M_w + \sup M_w}{2}. \quad (34)$$

Ki,

$$M_w = \{y \in U | \mu(y) \geq w\} \quad (35)$$

Not: μ Simetrik olmak zorunda değildir. Eğer öyleyse, M_w 'nin merkezi, orijinal kümenin merkez (2.28) ile aynıdır.



Şekil 17. LMOM yöntemi

Bulanık sistemin çıktısı, tek tek kural çıktılarının ağırlıklı ortalamasıdır, ağırlıklar, (2.33) 'te olduğu gibi, kural atış kuvvetleri w_i ve kural önem dereceleri m_i ' larsan oluşturmaktadır:

$$y^* = \frac{\sum_{i=1}^N m_i w_i y_i^*}{\sum_{i=1}^N m_i w_i} \quad (36)$$

Yine, kural önem ağırlıkları m_i isteğe bağlıdır. LMOM durulaştırma şekil 16. c'de gösterilmektedir ki bu M_w 'nin inf ve sup'ını ve ağırlıklı ortalama y^* göstermektedir (Formül (2.39)). LMOM durulaştırma yöntemi, MOM'da olduğu gibi M_w dışındaki üyelik fonksiyonunun şeklini dikkate almaz; ancak her kurala LMOM uygulandığında, tüm kurallar dikkate alınır. MOM sadece w'si en çok olan kuralları göz önüne alır. LMOM' un diğer bir avantajı da budur. Daha önce de belirtildiği gibi çakışan çıktı bulanık kümeleri dikkate alınmamalıdır. LMOM'da, sonuçta ortaya çıkan her kural ayrı ayrı durulaştırılır ve birbirine yakın konsantrasyon da dikkate alınır. Ayrıca, durulaştırma değerlerin'de, ağırlık olarak V kuralın sonuç kümesinin hacimi yerine w kural atış gücü kullanılmaktadır.

1.9. Sinir Ağları

Yapay sinir ağı (ANN), yapısal bir ağda birbirine bağlanmış nöronlar veya hücreler olarak adlandırılan basit işlem öğelerinden oluşur. Her bağlantının bir ağ parametresi, bir ağırlık ile ifade edilen bir gücü vardır. Sinir ağları, doğrusal olmayan regresyon modelleri ve dinamik sistemlerin geniş bir sınıfıdır [74]. Bu anlamda sinir ağları, regresyon modellerine oldukça benzer yüzey uyumu algoritmalarıdır. Ağın yapısı karmaşık olduğu için elde ettiği bilgiyi yorumlamak genellikle zordur. Örneğin, farklı girdi değişkenlerinin önemleri ve katkıları, regresyon modellerinde olduğu gibi karşılaştırılamaz. Sinir ağları (NN) sistemleri üzerine yapılan araştırmalar daha sonra nevrobiyolojiden açıkça ayrılmış olsa da, "sinir ağı" adı, NN'lerin biyolojik sinir ağlarına benzer paralel veri işlemlerinden kaynaklanmaktadır (Neuro Network). Yapay sinir ağları, en iyi olasılıkla, bir kural tabanı veya sistem hakkında analitik bilgi olmadan bir sistemi modelleyebilirler [75]. Büyük bir veri kümesi kullanarak, sistemin davranışını öngörebilen bir ağ oluşturulabilir. Modellemenin başarısı büyük ölçüde, verilerin kalitesine -sistemin gerçek davranışını temsil edip etmediği- ve şebekenin yapısına bağlıdır [76]. Sinir ağındaki öğrenme, denetlenmiş veya denetlenmemiş olabilir. Denetlenen öğrenmede, ağ hem bir girdi değeri hem de istenen çıktı değeri olarak sunulmaktadır. Bir öğrenme algoritması, NN'nin çıktısını hedef çıktı ile karşılaştırır ve ağ ağırlığı parametrelerini çıktı hatasını azaltan bir yönde değiştirmektedir [77].

Bu çalışmada eğitim sırasında öğrenme gücünü arttırabilmek için takviye öğretim kullanılmıştır. Güçlendirme öğrenim bölüm 5'te ele alınmaktadır. Denetlenmeyen öğrenmede, hiçbir hedef çıktı kullanılmaz. Bunun yerine, sinir ağı, girdileri paylaştıkları bazı ortak özelliklere göre sınıflandırır. Ağın çıkış sinyalleri, girdi kategorisine karşılık gelmelidir. Başka bir deyişle, ağ kendi kendini örgütlemelidir. Bu türden önemli bir NN mimarisi Kohonen'in kendi kendini organize eden haritasıdır. Bu tez, denetlenmeyen öğrenmeyi ele almaz ve okuyucu [78], [79], ya da [80] ya da bu bölümün sonunda listelenen literatüre atıf yapılır. Bilgisayar performansı arttıkça sinir ağları yaygınlaşmıştır. Geleneksel istatistiksel yöntemler artık modern bilgisayarların tüm özelliklerini kullanmamaktadır ve NN yöntemlerinin çoğu zaten bir dereceye kadar istatistiklerde kullanılmıştır ve eşdeğer istatistiksel yöntemler sıklıkla bulunabilir. Örneğin, en yaygın NN sistemi, çok tabakalı algılayıcı, genel bir doğrusal olmayan regresyondan başka bir şey değildir. Parametrik olmayan regresyon, diskriminant analizi ve ana bileşen analizi de NN literatüründe eşdeğerlerini bulmaktadır [74]. Ne yazık ki, bu iki sınıftaki terminoloji farklıdır ve yanlış anlaşılmalara neden olabilir. Sarle [81] sinir ağı biliminde kullanılan birkaç terimi listelemiş ve istatistiksel terminolojide eşdeğerlerini vermektedir.

Örneğin:

- NN terminolojisinde bağımsız değişken girdidir.
- Tahmini değer çıktıdır.
- Bağımlı değer hedef değer veya eğitim değeridir.
- Kalan değer hatadır.
- Tahmin, eğitim, öğrenme, uyarlama veya kendi kendini düzenleme.
- Tahmin kriteri, hata fonksiyonu veya maliyet fonksiyonudur.
- Gözlem model veya eğitim çiftidir.
- Parametre tahmini (sinaptik) ağırlıktır.
- Regresyon ve diskriminant analizi, denetlenen öğrenme veya hetero birlikteliktir.
- Veri azaltma denetlenmeyen öğrenme veya otomatik ilişkilendirmediir.
- Enterpolasyon ve ekstrapolasyon genellemedir.

Sinir ağlarının zekâsı ve yetenekleri hakkında aşırı derecede iyimser iddialar olmuştur. NN'lerin beyin'e benzer yapıları onları akıllı yapmaz ve eğer öğrenme ve genelleme zekâ belirtileri ise, o zaman birçok istatistiksel yöntem'de akıllıdır. Eğitim

aktivasyon fonksiyonunu, kullanılarak hesaplanır. Bir MLP'de, f aktivasyon fonksiyonu, örneğin:

$$\text{Kimlik: } f(x) = x \quad (37)$$

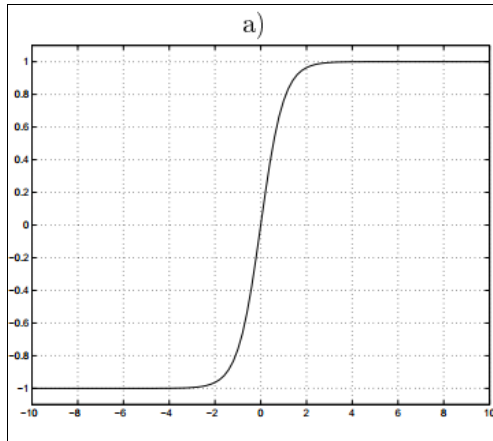
$$\text{Gauss: } f(x) = e^{-x^2/2} \quad (38)$$

$$\text{Hiperbolik: } f(x) = \tanh(x) \quad (39)$$

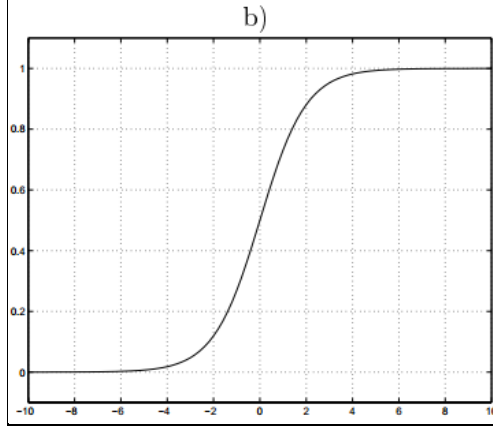
$$\text{Sigmoidal veya lojistik: } f(x) = (1 + e^{-x})^{-1} \quad (40)$$

$$\text{Eşik: } f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (41)$$

Genellikle aktivasyon fonksiyonu türevlenebilir olmalı ve her iki uçta da doymuş olmalıdır. Yukarıdaki fonksiyonların sadece hiperbolik ve sigmoidal (lojistik olarak da adlandırılır) fonksiyonları her iki uçta da türevlenebilirlik ve doymuşluk gereksinimlerini yerine getirir ve bu yüzden en yaygın aktivasyon işlevidir. Hiperbolik ve sigmoidal aktivasyon fonksiyonları Şekil 19 ve 20'de görülmektedir.



Şekil 19. (a) Hiperbolik aktivasyon fonksiyonu -1'den 1'e sıkıştır



Şekil 20. (b) Sigmoidal (veya lojistik) aktivasyon fonksiyonu 0'dan 1'e kadar sıkışır

Bir MLP ağında, aktivasyon fonksiyonu, doğrusal girdi birleşimi ile çalışır:

$$z_j = f(\delta_j + \sum_{i=1}^n a_{ij}x_i) \quad (42)$$

Bir RBF ağında, gizli tabaka etkinleştirme işlevi, giriş vektörü ile ağırlık vektörü arasındaki mesafeye göre çalışır:

$$z_j = f\left(\left[\sum_{i=1}^n \frac{(a_{ij}-x_i)^2}{2\delta_j}\right]^{1/2}\right) \quad (43)$$

Burada a_{ij} ağırlıkları genelde Gauss eğrilerinin merkezleri olarak kabul edilir ve her gizli düğümle ilişkili bir genişlik δ_j vardır. Bir RBF ağındaki aktivasyon fonksiyonu, $x \geq 0, x \in R$ üzerinde çalışan herhangi bir $f(x)$ fonksiyon olabilir. Ve sıfır'da maksimuma ulaşan ve sonsuz'da sıfıra yaklaşan, Gauss fonksiyonu gibi:

$$f(x) = e^{-x^2/2} \quad (44)$$

Hem MLP hem de RBF ağlarında, gizli tabakanın boyutu h değişebilir ve kaç hücrenin içermesi gerektiğini belirlemek için kesin kural yoktur. Hücre sayısına göre, ağın modellediği karmaşık sistemler olur, ancak hesaplama biraz artabilir. Çok karmaşık modeller aşırı parametризasyona neden olur: Veri kümesindeki rasgele bozulmalar olmamaları gerektiği halde modellenirler.

Birkaç gizli katman olabilir, buna rağmen genellikle bir tabaka yeterlidir. Her tabaka farklı bir aktivasyon fonksiyonuna sahip olabilir. Daha çok gizli tabaka eklenir ise, ağırlık karmaşık sistemleri modelleme yeteneğini artırır. Modelin yeterince karmaşık olup olmadığından emin olamazsınız ve şebeke kurma aşamasında deneme yanılma gerekmektedir.

Ayrıca i giriş tabaka hücreleri, b_{ik} ağırlıklı çıktı katmanı hücrelerine doğrudan bağlanabilir. Her bir gizli katmanın j hücresi (birkaçı varsa son gizli katmanda), her çıktı katmanın k hücresi, c_{jk} ağırlığıyla bağlıdır. Buna ek olarak, çıktı katmanı k hücresi, sabit bir girdi $-\theta_k$, alabilir; yine sıklıkla ağırlığı c_{h+1} olan $z_{h+1} = -1$ dahil edilir; gösterimde kolaylığı için $k = \theta_k$. Çıktı katmanının p boyutu dikkate alınan sistemdeki çıktı değişkenlerinin sayısı ile aynıdır. Çıktı katmanı aktivasyon fonksiyonu genellikle MLP'de ve RBF'de doğrusaldır:

$$v_k = \theta_k + \sum_{j=1}^h c_{jk} z_j \quad (45)$$

1.9.2. Sinir Ağlarında Yüzeysel Uyumu (Montajı)

Sinir ağları, yüzeysel uyum algoritmalarıdır. Yani gözlemlere en iyi uyan bir yüzey oluşturuyorlar. Yüzey küçük "tümseklere" sahiptir. Tümseklerin şekli, aktivasyon fonksiyonun seçimine bağlıdır. Hiperbolik ve sigmoidal aktivasyon fonksiyonları, Şekil 14'de görüldüğü gibi pürüzsüz "eşikler" üretir; Gauss aktivasyon fonksiyonu çan şeklindeki tepeler üretir ve eşik aktivasyon işlevi, dikdörtgen 'tuğla' kullanarak yüzeyler oluşturur.

Ağdaki bağlantı sayısı yüzeyin karmaşıklığını belirler. Her ağırlık parametresi, farklı ağırlık parametrelerinde aktivasyon fonksiyonunun şekli farklı olduğu için bir 'yumru' üretir. Öğrenme aşamasında, parametreler, yüzeyin gözlemlere daha iyi uyum sağlayacak şekilde değiştirilir. Parametrelerin değiştirilmesi, "yumru" şeklini ve yerini değiştirir. Orada daha fazla yumuru varsa, ağ daha karmaşık yüzey oluşturabilecektir.

Kabaca konuşmak gerekirse, "darbe", gözlem bölgelerine odaklanmaktadır. Şekil 19 ve 20'de görüldüğü gibi, aktivasyon fonksiyonu, kısıtlanmış bir alanda sabit olmayan çıkışlar verir ve bunun dışında, aktivasyon fonksiyonunun çıktısı neredeyse sabittir. Yüzeye oturan işlemde, gözlemlerin olmadığı veya az gözlemlenen alanlar, yüzey düz kalır

ve açıkça şebeke bu alanlarda sistemi modelleyemez. Bu nedenle, sinir ağları kullanılarak verilerin dışa vurumu mümkün değildir.

1.10. Neuro Bulanık Sistemler (Neuro Fuzzy Systems)

Bu bölümde, bulanık ve sinirsel yaklaşımların birleşimi sunulmaktadır. Bazı metinler, bulanık sinirsel ile sinirsel bulanık arasında fark olduğunu söylemiştir. Bir bulanık sinir ağı, bazı özelliklerin "bulanıklaşmış" bir sinir ağı denilmektedir. Bitişik katmanlarda, bir tabakanın her hücresi, Şekil 18'deki gibi diğer katmanın her hücresine bağlanmaktadır. Bir sinirsel bulanık sistem, sinir ağı olarak sunulan bulanık bir sistemdir ve tüm ağ bağlantıları mevcut değildir. Sinir bulanık sistemleri daha sonra tartışılacaktır. Bu tezde, bulanık sinirsel ve sinirsel bulanıklık için "nerö bulanık" kelimesi kullanılır, çünkü sistemin başlangıçta bir sinir ağı veya bulanık çıkarım motoru olup olmadığını değerlendirmek zor olabilir. Ayrıca, literatürde kullanılan terminoloji biraz belirsizdir. Bulanık sinirsel veya sinirsel bulanıklık gibi, bir sinirsel bulanık ağ, aşağıdaki hususların bir ya da birkaçında geleneksel bir sinir ağından farklıdır [56]:

- Girdiler bulanık sayılardır,
- Çıktılar bulanık sayılardır,
- Ağırlıklar bulanık sayılardır,

Aktivasyon fonksiyonları, ağırlıklı bir girdi toplamını kullanmaz; Bunun yerine girdileri birleştirmek için bulanık mantıksal işlemler kullanılır.

Burada bulanık bir sayı, klasik sayı sunan bulanık bir kümedir. Örneğin, "bulanık 5 ", üyelik fonksiyonu 5'de maksimuma ulaşan bulanık bir küme ile sunulabilir. Hesaplamanın bulanık sayılarda kapsamlı bir şekilde ele alınması için Kaufmann ve Gupta [87], Klir ve Yuan [56] veya Turunen [65] çalışmalarına bakmakta fayda vardır.

Sinirsel bulanık sistemlerde, bulanık mantığın parametreleri ağ bağlantı ağırlıkları olarak ifade edilir. Bu sistemler sinir ağları olduğu gibi yüzey uydurma algoritmalarıdır, ancak yapıları bulanık bir algoritma olarak yorumlanabilir. Tipik bir sinir ağında hücreler arasındaki bağlantı miktarı çok geniştir ve her katmanda her girdi hücresi her çıkış hücresine bağlıdır. Bu, bağlantılara çoğu zaman mantıksal bir yorum yapılamayacağı anlamına gelir.

Öte yandan, sinirsel bulanık bir ağda, bağlantı sayısı daha azdır ve bağlantıları izleyerek kural tabanını bulunabilir. Yalnızca bir kural oluşturan hücreler birbirine bağlıdır.

Brown ve Harris [88]: bir nero bulanık sistemin, geleneksel bir doğrusal olmayan modelden sadece modelleme ve öğrenme yetenekleri daha üstün olması durumunda kullanılmalıdır. Dilsel arayüz, uzman bilgisi dahil etmek veya sistemin davranışını doğrulamak için kullanışlıdır, ancak nöron bulanık sistem geleneksel bir modelden sadece sayısal giriş-çıkış davranışı daha uygun olması durumunda daha iyidir.

1.10.1. Neuro Fuzzy Sistemlerin Avantajları

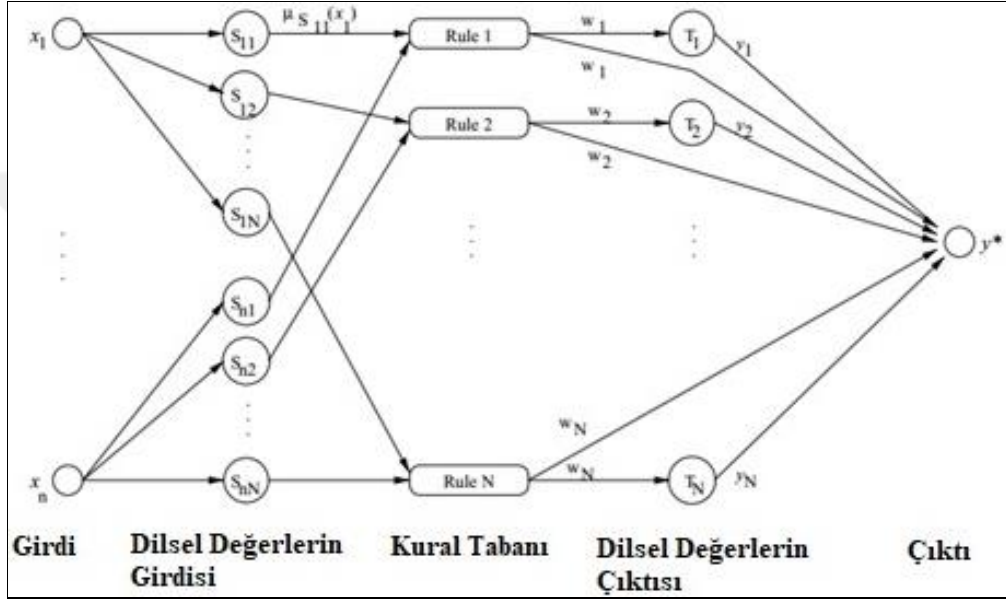
Bulanık mantık ve sinir sistemlerini birleştirmek, özellikle kontrol amaçlı geleneksel sistemlere kıyasla pek çok avantaj sağlayabilmektedir. Bulanık sistemler şeffaftır ve insana benzer mantığa kolayca adapte olmaktadır. Öte yandan, bulanık sistemler için çok az modelleme ve öğrenme teorisi vardır. Bir sinir ağı, veriden bilgi alabilir ve büyük paralel işlem yapabilir, ancak kara- kutu yaklaşımıyla bir NN sistemin yorumlamak kolay değildir. Bu sistemlerin bir birleşimi hem nitel hem de nicel bir yoruma sahip olabilir ve yalnızca bulanık veya sinirsel yaklaşımın dezavantajlarını önleyebilir. Örneğin, bulanık denetleyicilere, aynı anda çok sayıda kuralı eş zamanlı işlemek için bazen ihtiyaç duyulur. Sinir ağlarının paralel işleme yetenekleri ile bulanık kural işlemenin geliştirilmesi, yüksek hesaplama etkinliğini sağlar.

Bir sinir ağı, bulanık kontrolör için uygun bir yapı bulmak için kullanılabilir. Kuralların sayısı, kuralların formülasyonu ve üyelik işlevlerinin şekilleri, sistemin kontrol performansını etkiler; bunların hepsi bir sinirsel öğrenme şemasını kullanarak ayarlanabilir. Sinir ağında öğrenme, ağ ağırlık parametrelerini güncelleme anlamına gelir. Bir nerofuzzy sistemde, bu ağırlıklar bulanık sistemin parametreleridir.

Bulanık dilsel gösterim ağın modelleme yeteneklerine hiçbir şey katmamakla birlikte, tasarımcının sistemi başlatmasına veya doğrulamasına otomatik olarak oluşturulmuş bir sistem yapısının uygulanabilir olup olmadığını kontrol etmesine izin verdiği için önemli bir katkıdır [88]. Böylece, uzman bilgisi bulanık bir sisteme kolayca dahil edilebilir. Sinir ağında, uzman bilgisi genellikle yalnızca ağ parametrelerinin başlangıç değerlerini seçerken kullanılmaktadır.

1.10.2. Yapay Sinir Ağında Bulanık Bir Sistemin Yapısı

Bu tez çalışmasında kullanılan bulanık kontrol sisteminin yapısı ikinci bölümde ANFİS biçiminde sunulmaktadır. Bu, diğer yapılarda kullanılsa da, nevrofuzzy (daha doğrusu sinirsel bulanıklık) sistemi için ortak bir mimaridir. Bu mimari ilk defa Berenji ve Khedkar tarafından sunuldu [73].

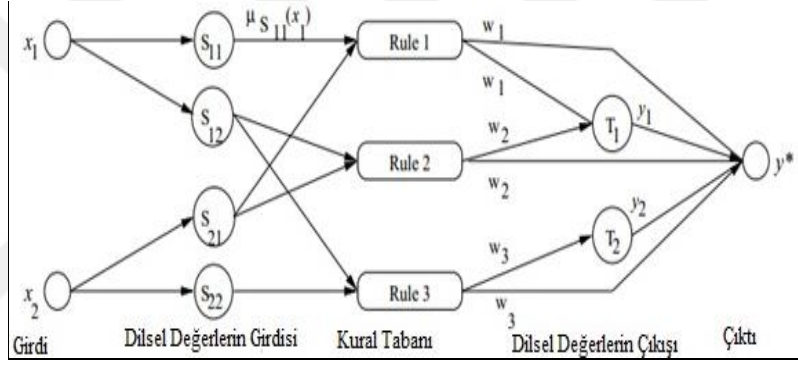


Şekil 21. Sinir ağı biçimindeki bir basit bulanık sistemin yapısı

Birinci katman girdi katmanı ve İkinci katmanın çıktısı üyelik fonksiyonu değeri $\mu_{s_{ji}}(x_j)$, üçüncü katmanın çıktısı kural ateşleme kuvveti w_i ve dördüncü katmanın çıktısı y_i kural i 'in çıktısıdır. Sinir ağının ağırlık parametreleri, 2 ve 4 katmanındaki üyelik fonksiyonlarının şekil parametreleridir. Ağırlık parametreleri şekil içinde gösterilmemiştir.

Bir neuro fuzzy ağın ilk katmanı, $x_j, j = 1, 2, \dots, n$ girdi değişkenlerinin x_j ölçümlerinden oluşur. İkinci katman, $\mu_{s_{ji}}$ üyelik fonksiyon değerlerini s_{ji} girdi dilsel değerleri için hesaplar; Örneğin, $\mu_{s_{1i}}(x_1)$ ve $\mu_{s_{2i}}(x_2)$. Birinci katmanlı her hücre sadece ikinci katmanlı dilsel değerler belirli girdi değişkenleri temsil eden hücrelere bağlıdır: x_1, s_{1i} 'ya ve x_2, s_{2i} 'ya ve bunun gibi bağlanmaktadır. Böylece, her ikinci katman hücresi, sadece birinci katmanın bir hücresinden girdi alır ve birinci ve ikinci kat arasındaki ağ ağırlık parametreleri, tipik bir sinir ağında olduğu gibi a_{ij} 'den zıade p_{ji} formundadır, p_{ji} burada j linguistik değişkeninin i inci üyelik fonksiyonu için bir parametre vektörüdür,

çünkü üyelik fonksiyonları genellikle sadece bir parametreden daha fazla parametre gerektirir. Örneğin, üçgen bir üyelik fonksiyonun da üçgenin köşelerinin x koordinatlarına karşılık gelen üç parametre gerektirir. Ayrıca, ağırlık vektörünün şimdi bir sinir ağındaki üyelik fonksiyonunun şeklini etkilediğini, aktivasyon fonksiyonunun f formu (bkz. Formül (3.1)) sabitlenmiş ve a_{ij} parametreleri, f'nin kanıt parametreleri olmuştur. Belirtilmesi gereken bir diğer husus, Bölüm 2.5.1'de belirtildiği gibi, dilsel değişken X_j 'nin aynı dilsel değerleri S_{ji} 'nin birkaç kuralda görünebileceğidir; böylece, kurallar i ve i' için $S_{ji} = S_{ji'}$ olur. Bunun bir örneği Şekil 4.2'de görülmektedir. Gösterim amaçlı kural i için; aynı dilsel değerlerin başka bir kuralda kullanılabilmesine rağmen, Şekil 22'deki dilsel değerler $S_{1i}, S_{2i}, \dots, S_{ni}$ 'yi ($i=1, 2, \dots, N$) kullanmaktadır.



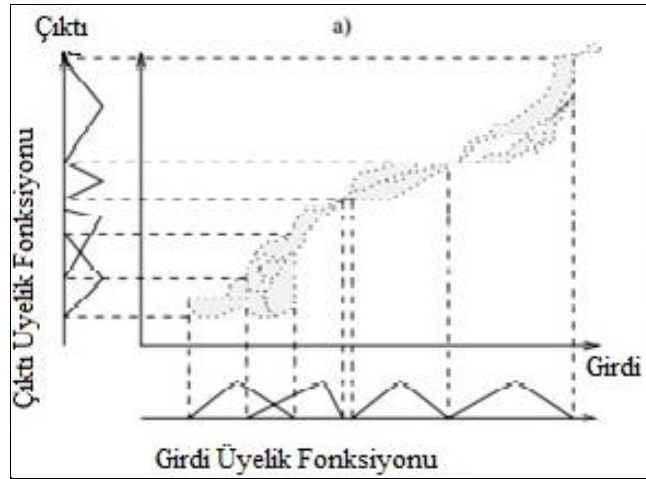
Şekil 22. Sinir ağı biçimindeki bulanık sistem örneği. Aynı dilsel değerler birkaç kuralda görünmektedir.

Üçüncü tabaka bulanık bir kural tabanına karşılık gelir. Üçüncü katmandaki i hücresi, kural i'nin öncülündeki tüm koşulları "eğer X_1, S_{1i} dir ve X_2, S_{2i} ve ... dir", karşılık gelen üyelik fonksiyonu değerlerinde "ve" ve "veya" operatörlerini kullanarak birleştirir (bkz. 2.3). i hücrenin çıktısı w_i , kural i'nin ateş gücüdür. Üçüncü katta ağ ağırlık parametreleri kullanılmaz. Dördüncü katman "o halde Y, T_i dir kural sonuçlarına karşılık gelir. Her hücre çıktı değişkeni için dilsel bir değeri temsil etmektedir. Birçok kural aynı dilsel değerleri kendi sonuçlarında kullandığından, 4. katmandaki hücreler birden fazla girdiye sahip olabilirler. Başka bir deyişle, aynı dilsel değer T_i , gibi birkaç kuralda görünebilir. Şekil 22'deki gösterimsel amaçlar için yine her i kuralın kendi sonuçta gelen dilsel değeri T_i vardır. Dördüncü katman, her kuralın durulaştırılmış çıktısını hesaplar. Bu katmandaki ağ parametreleri, çıktı bulanık kümeleri için üye fonksiyon P parametreleri dir.

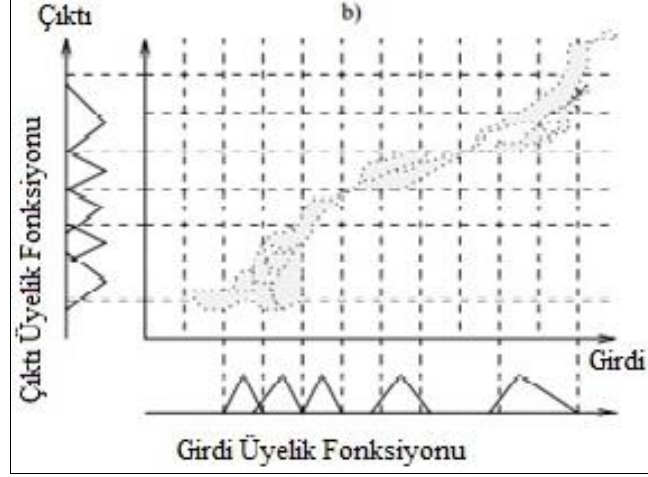
Beşinci katman, her kuralın çıktılarını birleştirir ve sistemin çıktı değerini verir. Kural çıktıları genellikle bazı ortalama yöntemlerin kullanılarak birleştirilir. Beşinci katmanda değiştirilebilir ağırlık kullanılmaz.

1.10.3. Neuro Fuzzy Sistemlerine Örnekler

Bu bölümde, nevrofuzzy sistemlerine bazı örnekler sunulmaktadır. Bu sistemlerde öğrenme, bilinen giriş-çıkış gözlemlerine ait bir eğitim verileri kümesinin daima gerekli olması bakımından denetlenmektedir. Lin ve Lee denetimsiz ve denetlenen öğrenmeyi birleştiren genel bir nevrofuzzy sistem önermektedir. Şekil 23. a'da görüldüğü gibi bulanık kuralların varlığını gösteren veri kümelerini bulmak için denetimsiz bir öğrenme algoritması kullanılır. Denetimli öğrenme üyelik fonksiyonu daha sonra ince ayarında kullanılır. Eğitim verileri önceden sınıflandırılmış olduğu için, denetlenen öğrenme sınıflandırmadan daha iyi performans gösterir. Kosko [89], vektör nicemleme ile bir ürün alanı kümeleme yöntemi sunmakta olup, burada giriş-çıkış alanı Şekil 24. b'deki gibi bir şebekeyle (tablo) bölünmüştür. Birkaç gözlem bir hücreye düşerse, bu hücreye bir kural yerleştirilir. Şebekede değiştirilebilir. Durum, Şekil 23. a'daki Lin ve Lee'nin yöntemine oldukça benzemektedir, ancak Şekil 23. a'da bir şebeke kullanılmamaktadır.

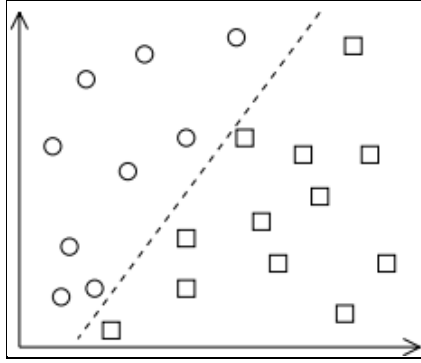


Şekil 23. (a) Sistem giriş-çıkış alanındaki veri kümeleri bulanık kuralların varlığına işaret eder.



Şekil 24. (b) Ürün uzayı kümeleme yöntemi.

Keller ve Hunt [90], bulanık üyelik fonksiyonlarını kullanan bulanık bir sinir ağının doğrusal olarak ayrılabilir veri kümesinin nasıl ayırabileceğini açıklamaktadır. Veri kümeleri, Şekil 25 gibi düz bir çizgi ile ayrılabilirse doğrusal olarak ayrılabilir veya daha genel olarak, n -boyutlu uzayda noktalar bir $(n-1)$ - boyutlu hiperplelane ile ayrılabilirse.



Şekil 25. Doğrusal olarak ayrılabilir veri kümeleri düz bir çizgi ile ayrılabilir.

Hayashi ve ark. [91] bulanık özellikler ile donatılmış klasik bir ileri beslemeli sinir ağı sunmaktadır. Klasik sinir ağını karakterize eden tüm gerçek sayılar (her bir hücrenin giriş ve çıkışları ve her katmandaki ağırlıklar), bulanık sayılar haline dönüşür. Her tabakanın aktivasyon fonksiyonları orijinal sinir ağında kullanılanlarla aynıdır, ancak Formül 3.1'deki girdilerin ağırlıklı toplamı bulanık aritmetik kullanılarak hesaplanmalıdır. Hata fonksiyonu ve geri yayılım öğrenme algoritması bulanıklaştırılmalıdır. Geri yayılım

algoritmasını bulanıklaştırmanın bir yolu, gerçek sayıları bulanık sayılarla değiştirmek ve bulanık aritmetiği uygulamaktır.

Horikawa ve ark. [92] özellikle modelleme amaçlı farklı nevrofuzzy ağ yapıları sunmaktadır. Ağ yapıları, bulanık çıkarım sisteminin sonucunun türüne göre kategorize edilmektedir. Sonuç, bulanık olmayan bir sabit olabilir: "Eğer X_1, S_1 dir, ... ve X_n, S_n ise o zaman Y, c dir". Yazarlar ağ yapısını, yani girdi değişkenlerinin birleşimi ve öncül ve sonuçlarda üye olma fonksiyonlarının sayısını belirlemek için bir yöntem önermektedir. Bu, bağlantı ağırlıkları üzerinde bir geri yayılım öğrenme algoritması kullanılarak yapılmaktadır. Öğrenimden sonra, sadece girdi ve çıktı değişkenleri arasında ki bağlantılar mevcuttur. Ağ yapısı, bulanık çıkarım sisteminin kurallarını belirler. Ağ yapısı seçildikten sonra, üyelik fonksiyonu parametreleri, geri yayılım algoritması kullanılarak değiştirilebilmektedir.

Jang [93] üyelik fonksiyonun ince ayarlaması için hem uzman bilgisi hem de giriş-çıkış veri çiftlerini kullanan bir ANFIS (Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi) mimarisin önermektedir. Bu sistemde, gradyan iniş tekniğini (bölüm 3.4) ve parametrelerin tanımlanmasını için en küçük kareler tahminin birleştiren bir hibrid öğrenme tasarımı kullanılır. Sonuç, sistemin davranışını modelleyen bir girdi-çıkış eşlemesidir. ANFIS, ağdaki tüm olası bağlantılar kullanıldığı için "sinirsel bulanık" dan çok "bulanık sinir" 'e sahiptir ve ağ yapısı bulanık kural tabanı olarak yorumlanamaz.

Klir ve Yuan [56], Kosko [60] ve Yager ve Filev [56] üyelik işlevlerinin inşası ve uyarlanması için bazı yöntemler de sunmaktadır. Bütün bu yöntemler, bilinen girdi-çıkış gözlemlerinin bir eğitim veri kümesin gerektirir. Giriş-çıkış çiftlerinin gözlemleri mevcut değilse durum daha hantaldır. Bu durumda, öğrenme başka kriterlere dayanmalıdır. Bu durum bölüm 5'te ele alınmaktadır.

1.11. Adaptif Neuro-Bulanık Çıkarım Sistemi (ANFIS)

Ağ tabanlı bulanık çıkarım değişimi (ANFIS), iki Soft-hesaplama yöntemi olan ANN ve bulanık mantığın birleşimidir [94]. ANFIS'in ağ yapısı, öncül ve sonuç parçaları olarak adlandırılan iki bölümden oluşur. ANFIS eğitimi bir optimizasyon algoritması kullanılarak bu parçalara ait parametrelerin belirlenmesi anlamına gelir. ANFIS eğitim sırasında mevcut giriş-çıkış veri çiftlerini kullanır. Daha sonra, bu parçaların birbiriyle bağlantılı olduğu IF-THEN bulanık kuralları elde edilir. Bulanık mantık, insan bilgisinin ve

kavrayışlarının niteliksel yönlerini kesin niceliksel analiz sürecine dönüştürme yeteneğine sahiptir. Bununla birlikte, dönüşüm ve insan düşüncesi sürecinde kural tabanı bulanık çıkarım sistemine (FIS) kılavuz olarak kullanılabilecek tanımlanmış bir yöntemle sahip değildir ve üyelik fonksiyonlarını (MF'ler) ayarlamak oldukça uzun zaman almaktadır [94]. ANFIS'te, bulanık mantığın karar verme mekanizması ile sanatsal sinir ağlarının öğrenme yeteneği ve ilişkisel yapısı birleştirilir. ANFIS, yapay sinir ağlarında olduğu gibi eğitim veri setini kullanarak örneklerle öğrenmeyi gerçekleştirir, böylelikle ilgili problemi çözmek için en ideal ANFIS yapısı elde edilir. Elde edilen yapı, daha önce hiç görmediği numuneler üzerindeki etkisini görmek için test sürecine tabi tutulur. Düşük hata değerleri ANFIS modelinin uygunluğunu gösterir. Yapay sinir ağının en önemli dezavantajlarından biri, elde edilen ağırlık değerlerinin açıklanamamasıdır. Bu dezavantaj, ANFIS'in yapısında bulunan bulanık çıkarım sistemi ile ortadan kaldırılmaktadır. Bu yapı nedeniyle birçok gerçek dünya probleminin çözümünde kullanılmaktadır [95]. Öğrenci modelleme sistemi [96], tıbbi sistem [97], ekonomik sistem [98], görüntü işleme ve özellik çıkarma [99], tahmin ve tahminler [100], imalat ve sistem modelleme [101], elektrik ve elektronik sistemleri [102], ANFIS'in kullanıldığı alanlardan bazılarıdır. Bunların dışında, farklı çalışmalar yapılmıştır. Cheng ve Lee [103] bulanık regresyon analizi için bir ANFIS modeli kullanmışlardır. Cai ve diğ. [104] akü şarj durumu (SOC) tahmini için karma öğrenme algoritması ile ANFIS'i eğitmiştir. Güney ve Sarıkaya [105] ince ve kalın dikdörtgen mikroşerit yama antenlerin giriş dirençlerini ANFIS ve HL kullanarak hesaplamıştır. Buna ek olarak Güney ve Sarıkaya [106], [107] mikroşerit antenler üzerinde yapılan farklı çalışmalarda aynı nöro-bulanık yöntemi kullandılar. Entchev ve Yang [108], konut mikro kuşak kurulumunda katı oksit yakıt hücresi performansını tahmin etmek için ANFIS ve hibrit algoritmayı kullandılar. Tabesh ve Dini [109] Tahran'da kısa vadeli su talebi tahmini için hibrit öğrenme algoritması ile ANFIS'den oluşan bir nöro-bulanık model önermişlerdir. Yıldız ve ark.[110], dalgalık-entropi özellik çıkarımı kullanarak uyanıklık düzeyi tahmini için ANFIS uygulamasını tanımlamıştır. Dalkilic ve Apaydin [111] bulanık kümelenmeye dayalı geçerlilik ölçütünü kullanmışlardır. Kavitha ve Duraiswamy [112] fundus görüntülerinde diyabetik retinopatinin otomatik taranması için bir yaklaşım sunmuşlardır. Rajasekaran ve Meena [113] manyetik tınlama (MR) beyin görüntü sınıflandırması ve tümör tespiti için nöro-bulanık bir yaklaşım sundular. Karamietal. [114] hava soğutmalı bir ısıda ısı transferinin modellenmesi için ANFIS'i eğitmek için hibrit bir öğrenme algoritması kullanmışlardır. Klasik bükümlü bant ekleri ile donatılmış eşanjör. Dalkilic ve

Apaydin [111], tatmin edici çözümler elde eden bulanık çıkarım sistemine dayanan bulanık regresyon modeli için bir yaklaşım sunmuşlardır. Taghavifar ve Mardani [115], tarım traktörü tahrik tekerleklerinin enerji verimliliği endekslerinin ANFIS modellemesi ile gerçekleştirmişlerdir. Bahadori ve diğ [116], ANFIS'li gazlı sistemlerde doğal gazın yanma etkinliğini tahmin etmişlerdir. Yukarıda verilen bilgiler ışığında, ANFIS'in popüler ve etkili bir mimari zekâ tekniği olduğu görülebilir. ANFIS ile başarılı sonuçlara ulaşmak, eğitildiği optimizasyon algoritmaları ile doğrudan ilişkilidir. ANFIS ilk geliştirildiğinde GD / BP ve LSE kullanan karma öğrenme algoritması kullanılmıştır. Daha sonraki yıllarda GD, LSE gibi türev tabanlı yöntemlerin ve GA, PSO, ABC gibi sezgisel algoritmaların ANFIS eğitiminde daha iyi performans elde etmek için kullanıldığı görülmüştür. Şimdiye kadar ANFIS üzerine çok sınırlı inceleme çalışmaları literatürde sunulmuştur [117], [118], [119], [120]. Sadece bunlardan biri ANFIS eğitim yaklaşımları ile ilgilidir [118]. ANFIS hızlı ve doğru bir öğrenimdir, uygulaması basit, bulanık kurallarla mükemmel bir açıklamadır ve problem çözme yeteneklerini artırmak için hem sayısal hem de dilsel değerleri ilişkilendirmek kolaydır. Kural tabanını önceden tanımlamaya gerek yoktur çünkü ANFIS en iyi parametrelerini eğitim yoluyla bulur. Yapılan çalışmalara bakıldığında Jang tarafından sunulan ANFIS yapısı karmaşık sistemlerin modellemesinde problemlere yaklaşmak ve doğrusal olmayan fonksiyonları çöze bilmek için kullanılmıştır [11], [12].

İki girişli ve her giriş için iki üyelik fonksiyonuna sahip basit bir ANFIS yapısı örneği, Şekil 22'de sunulmuştur. Bu bağlamda, daire sabitlenmiş bir düğüm içindir ve kare adaptif bir düğümü gösterir. İncelenen FIS'in iki giriş ve bir çıkarım olduğu varsayılmıştır.

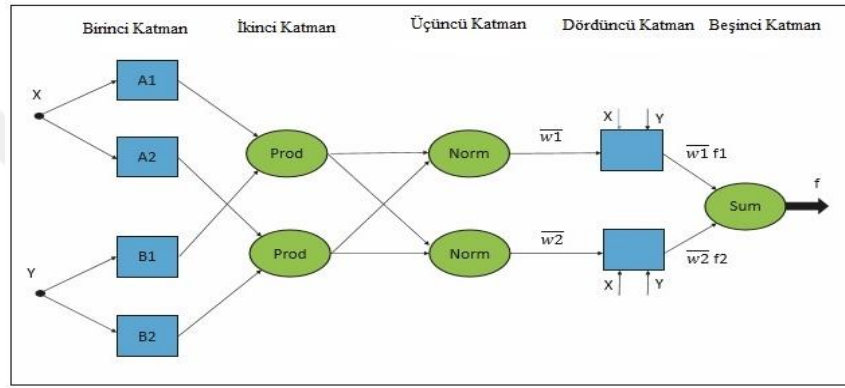
1.11.1. ANFIS Tip-1 Mimarisi

ANFIS mimarisi, Takagi-Sugeno bulanık çıkarım sisteminin modeline benzer bir işlevi olan, öğrenme algoritması üzerinde denetimli öğrenmeyi kullanan uyarlamalı bir ağıdır. Şekil 9. a, b, Takagi-Sugeno modeli ve ANFIS mimarisi için bulanık mantık yürütme mekanizmasını göstermektedir. Basitlik için, x ve y iki giriş ve bir çıkış f olduğunu varsayalım. Takagi-Sugeno modeli için “If-Then” yönteminde iki kural kullanılmıştır. İncelenen FIS'in iki giriş ve bir çıkarım olduğu varsayılmıştır.

First rule: If (x is A₁) and (y is B₁) then ($f_1(x) = P_1x + q_1y + Pr_1$)

Second rule: If (x is A₂) and (y is B₂) then ($f_2(x) = P_2x + q_2y + Pr_2$) (46)

Burada A₁, A₂ ve B₁, B₂ her x ve y girişin 'nin (işlevlerin bir kısmı) üyelik fonksiyonu, p₁, q₁, r₁ ve p₂, q₂, r₂ ise, Takagi–Sugeno bulanık çıkarım modeli 'nin (son kısmın) bölümünün lineer parametreleridir. Şekil 26'da sunulmuş olan ANFIS mimarisinin beş katmanı vardır. Birinci ve dördüncü katmanlar adaptif bir düğümü içerirken, diğer katmanlar sabit bir düğümü içerir. Her katmanın kısa bir açıklaması şu şekildedir:



Şekil 26. A Sugeno bulanık girişim sistemi “If-Then” ve bulanık mantık mekanizması. b- b ANFIS mimarisi. [121].

Katman 1: Bu tabakadaki her bir düğüm bir işlev parametresine adapte olur. Her düğümden gelen çıktı, üyelik işlevlerinin girdisi tarafından verilen bir üyelik değeri derecesidir. Altaki birinci katmanın açıklaması sunulmaktadır. ($O_{L,i}$ L katmanı düğümünün i'inci çıktısı)

$$O_{1,i} = \mu A_i(x), \quad i=1,2$$

$$O_{1,i} = \mu B_{i-2}(y), \quad i=3,4 \quad (47)$$

μA_i ve μB_{i-2} , sırasıyla A_i ve B_i kümeleri için üyelik fonksiyonlarının derecesidir ve {a_i, b_i, c_i} üyelik fonksiyonunun şeklini değiştirebilen üyelik fonksiyonunun parametreleridir. Bu katmandaki parametreler tipik olarak öncül parametreler olarak adlandırılır.

Katman 2: Bu katmanda Gauss işlevi, düğüm işlevidir. $\mu_{A_i}(x)$ ve $\mu_{B_i}(x)$ Genellikle aşağıdaki denklemde verilen bir Gauss fonksiyonu olarak seçilir. Çıkış düğümü, düğüme gelen girdinin çarpılmasının sonucudur ve bir sonraki düğüme iletilir. Bu katmandaki her düğüm, her kural için ateşleme gücünü temsil eder. İkinci katmanda, AND gibi genel performansa sahip T-norm operatörü, çıktıyı elde etmek için uygulanır:

$$\mu_{x_i}(T) = \exp \left[-\left(\frac{x - c_i}{2a_i} \right)^2 \right] \quad (48)$$

$$O_{2,i} = w_1 = \mu_{A_i}(x) * \mu_{B_i}(x), \quad i=1,2 \quad (49)$$

a_i ve c_i ise sırasıyla üyelik fonksiyonunun standart sapması ve merkez parametreleridir. Bu katmandaki her bir düğümün çıktısı $[0,1]$ aralığındadır ve üyeliğe göre girdinin üyelik derecesini gösterir. Bu parametreler geri yayılma sırasında modifiye olacaktırlar.

Katman 3: Bu katmandaki her düğüm sabit veya nonadaptif ve daire düğümü N olarak etiketlenir. Her düğüm, ateş eden kurallar ile tüm kuralların ateşleme güçlerinin toplamı arasındaki oranın hesaplanmasıdır. Bu sonuç normalize ateşleme gücü olarak bilinir.

$$O_{3,i} = \bar{w}_1 = \frac{w_1}{w_1 + w_2}, \quad i=1,2 \quad (50)$$

Katman 4: Bu katmandaki her düğüm, bir düğüm işlevi olarak tanımlanmış bir düğüm işlevi olan bir çıktının uyumlu bir düğümüdür:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (P_i x + q_i y + r_i), \quad i=1,2 \quad (51)$$

Burada $\bar{w}_i$, önceki katmandan (üçüncü katman) normalleştirilmiş ateşleme gücüdür ve $(P_i x + q_i y + r_i)$ düğümündeki bir parametredir. Bu katmandaki parametreler, sonuçta ortaya çıkan parametreler olarak adlandırılır.

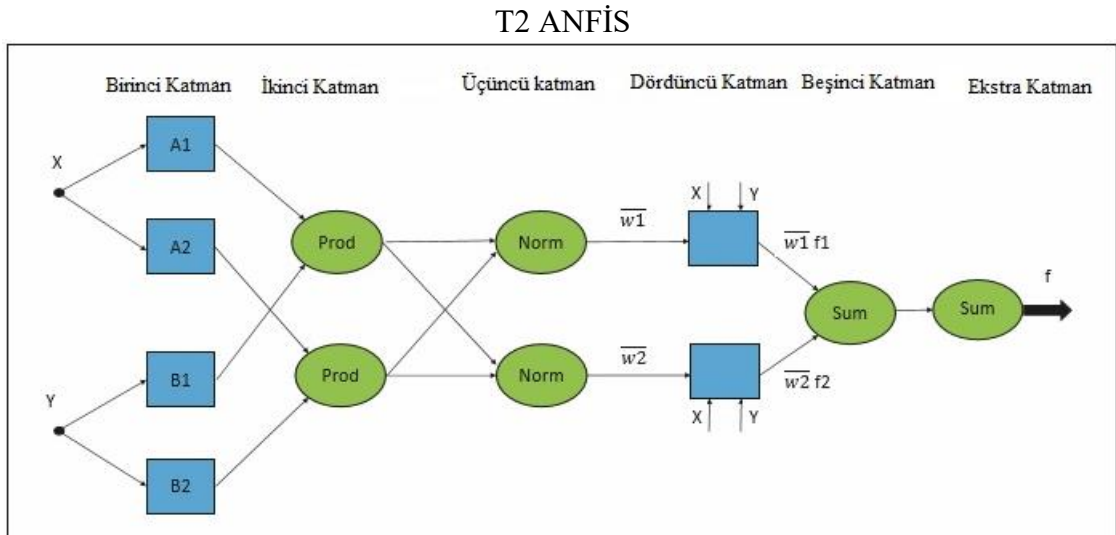
Katman 5: Bu katmandaki tek düğüm ve toplam (sum) olarak etiketlenir, genel çıkışı önceki düğümden gelen tüm sinyallerin toplamı olarak hesaplayan sabit veya nonadaptive bir düğümüdür.

$$O_{5,i} = \sum_{i=1}^2 \bar{w}_i f_i = \frac{\sum_{i=1}^2 w_i f_i}{w_1 + w_2} \quad (52)$$

Bu katmanda, bir daire düğümü, $\sum$ olarak etiketlenir.

1.11.2. ANFİS Tip-2 Mimarisi

1993 yılında Tip-1ANFİS tanıttıktan sonra, John ve Czarnecki [122] 1998'de Tip-2 ANFİSİ tasarladılar. Kavşak için kullanılacak operatörün seçilmesi, kuralların formatı ve faylanma yönteminin seçilmesi de dahil olmak üzere, tip-2ANFIS'de halen geçerli olan tip-1 ANFIS'in tasarımına ilişkin bazı endişeler vardır. John ve Czarnecki tarafından tasarlanan tip-2 ANFIS, Tip-1ANFIS'e kıyasla ilave bir katmana sahiptir. Bu ilave katman defuzzi için geçerlidir. R. JohnandC. Ve Czarneckinin araştırmasına dayanarak. Önerilen tip-2 ANFIS yapısı için Karnik ve Mendel [123] tarafından açıklanan tip azaltma işlemi benimsenmiştir. Bu işlem, Tip-2 setini bir tip-1 setine indirgemek ve daha sonra net bir değer elde etmek için normal sette bu setin buzunu çözmek için uygulanır. Elde edilen net değer, hedef çıktı ile karşılaştırılacak ve parametre değişikliği için ağda geri yayılma için hata hesaplanacaktır. Uygulamada, her katmanın çıktısı bir tür-2 kümesidir ve son kat ise tür azaltma işlemini yapar. İki girişli – ANFIS-2 tipinin bir örneği Şekil 27:



Şekil 27. İki giriş ve iki üyelik fonksiyonuna sahip örnek bir T2-ANFIS kontrolörünün yapısı

1.12. Takviye Öğrenim

Çoğu nero bulanık ve diğer kontrol sistemleri öğrenme aşamasında eğitim verilerin gerektirir. Eğer sistem, bilinen birkaç giriş-çıkış çifti ile sunulduysa, istenen bir çıktı vermeyi öğrenir. Bu durumda, kontrol işlemi izlemeye dayanmaktadır: Amaç, giriş-çıkış alanındaki istenen yörüngeyi takip etmektir. Giriş-çıkış verileri mevcut değilse ve sadece bazı performans endeksleri eldeyi ise, kontrol sistemi izlemeden başka yöntemlere dayandırılmalıdır. Sutton ve ark. Kontrol problemlerini iki sınıfa ayırmaktadırlar:

- Sorunları takip etme
- Optimal kontrol problemleri

İkincisinde amaç, sistem davranışının işlevselliğini en aza indirmek veya en üst düzeye çıkarmaktır ve fonksiyonsel bir referans yörüngesi ile tanımlanmamıştır.

İzleme problemlerinin bir referans yörüngesinin, önceden bilgisini gerektirdiğini unutmayın, ancak pek çok problemde, sapma yörüngesinin belirlenmesi, genel sorunun en önemli kısmı olmasa bile, önemli bir bölümdür [124]. Sutton ve ark. Hem izleme hem de optimal kontrol problemleri için dolaylı ve direkt uyarlanabilir kontrol yöntemlerini ayırt etmişler. Dolaylı bir yöntem, sistemin her adımında net bir modelini tahmin eder ve modelin kontrol kuralını belirler. Doğrudan yöntemler, sistem modellemeden kontrol eylemini belirler. Takviye öğrenmede denetleyici, önceki kontrol işleminin iyi olup olmadığı hakkında bir sinyal alır. Bu sinyal, eylemin hemen ardından veya birkaç adım sonra kullanılabilir; bu da öğrenmeyi daha zorlayıcı yapar. Güçlendirme öğrenme yöntemleri, giriş-çıkış verisinin mevcut olmadığı optimum kontrol problemleri için uygundur. Bunlar doğrudan yöntemlerdir: sistemin bir modelini tanımlamazlar.

Takviye öğrenimi, bir eylemin iyi sonuçları varsa, o eylemi üretme eğilimi güçlendiği, yani takviye edildiği fikrine dayanmaktadır. Sonuçlar birkaç zamana kadar ortaya çıkmazsa, sistem sonuçların iyiliğini "önceden" tahmin etmelidir. Takviye öğrenme teorisi, hayvan öğrenimi üzerine yapılan çalışmalardan kaynaklanır, ancak uyarlamalı optimal kontrol teorisine güçlü bağlar da vardır. Lin ve Lee [89]. Takviye öğrenme tarihini iki aşamaya bölmüşlerdir. 1950'lerin ilk aşamasında, insan ve hayvan öğrenmesinin hesaplama modelleri geliştirilmiştir. Öğrenme, bir tür rassal süreç olarak görülmekteyken. 1980'lerin ikinci aşamasında, bir girdi örüntüsü bir takviye sinyaline göre çıkmıştır ve modelleriyle ilişkilendirilen ilişkisel takviye öğrenme konseptini getirmiştir.

İki tür takviye öğrenme sistemi vardır [124]: aktör-eleştirel öğrenme ve Q-öğrenme sistemi. Bir aktöreleştirmen sistemi iki alt sistemi içerir: Biri her durumdaki en iyi kontrol eylemini seçmek için (aktör) ve her durumun uzun vadeli faydasını (eleştirmen) tahmin etmek için. Bir neuro bulanık sistem takviye öğrenme kullanıyorsa, genellikle oyuncu-eleştirel öğrenme seçilir. Bulanık kontrol sistemi, kontrol eylemini seçen aktördür ve bulanık sistem tarafından seçilen kontrol eyleminin başarısını değerlendiren eleştirmen, başka bir sistem (genellikle bir sinir ağı); bu eleştiri öğrenmede kullanmıştır. Bir Q-öğrenme sistemi mevcut x durumun 'deki tüm durum eylem çiftleri için hizmetleri(araçları) tahmin eder. Hizmetlerin (araç) değerlendirilmesi, sistemin gelecekte en iyi şekilde performans gösterdiği varsayımına dayanır: " x durumun 'da a eylemini seçmesi ve sonra orada en iyi performansın faydası nedir? " Sistem gelecekteki performansı hakkında bazı varsayımlar yapmalıdır ve optimalite varsayımı en doğal olanıdır [124], [125].

1.12.1. Denetlenmeyen Öğrenme

İlk görüşte, takviye öğrenme, sinir ağlarında kullanılan denetlenmiş veya hemen hemen denetlenmeyen öğrenme algoritmalar ile (bkz. Bölüm 3) aynı davranıştadırlar. Takviye öğrenimi, bir tür denetimli öğrenme olarak sınıflandırılabilir. Bununla birlikte, takviye öğrenmesi ile denetlenen ve denetlenmeyen öğrenme arasında önemli farklılıklar vardır. Denetlenen öğrenmede, bir "öğretmen", eğitim sırasındaki her girdi düzeninde istenen çıktıyı sisteme sağlayabilir. Sistem çıktısını verdikten sonra, çıktı ve istenen tepki karşılaştırılır ve fark, sistem parametrelerini güncellerken kullanılır. Bireysel elemanların sistem çıktısına katkısı bilinir ve elemanların çıktıya ne kadar katkıda bulunduğu göre her elemanın parametreleri güncellenir. Aksine, takviye öğrenme, kontrol eyleminin bazı sonuçlarının bilindiği problemlerde kullanılır; Ancak herhangi bir öğretmen, bireysel unsurların istenen tepkilerini sağlayamaz. Denetimsiz öğrenme sistemlerinin doğru çıktıyı söylemek için "öğretmen" e ihtiyaçları yoktur. Onlar, girdi alanındaki belirli bir alt kümeyle ait girdilere belli bir çıktı vererek bilgiyi kümelemektedir. Güçlendirme öğrenme algoritmaları, sistemin sadece bilgi kümelemesi değil giriş alanındaki farklı alt gruplara ait girişlerde doğru eylemleri bulması gereken problemlerde kullanılır. Takviye öğrenmesi ile denetlenen ve denetlenmeyen öğrenme arasındaki ek bir fark, en iyi çıktı için stokastik aramadır. Çevre, istenen çıktıları sağlayamadığı için, bilgi öğrenme sistemi, hangi cevapların performansın iyileştirilmesine yol açtığını kendi başına keşfetmelidir.

[3]. Sistem çıktısını tipik bir sinir ağında olduğu gibi ağ ağırlık parametrelerinin ve girdi ölçümlerinin bir fonksiyonu olarak hesaplar, ancak bu çıktı daha sonra rasgele bir miktarda saptırılır veya sarsılır. Bu şekilde, sistemin hangi girdilerde başarılı olduğunu görmek için farklı çıktı türlerini denemek mümkündür. Özetlemek gerekirse, takviye öğrenme sistemi insanların ve hayvanların öğrenme şeklidir yani, deneme ve yanılma yöntemini ifade eder.

1.13. Üstsezgisel Algoritmalar (Tek-Amaçlı)

Günümüzde optimizasyon isteklerinin sayısı git gide artmaktadır. Her işlem optimize edilebilir. Hiç şüphe yoktur ki, bu dönemde birçok şirket optimizasyon problemlerini çözmede yer almaktadır ve başarılı projelere imza atmaktadırlar. Gerçek hayattaki birçok karmaşık problem, optimizasyon problemleri olarak formüle edilebilir. Optimizasyon, maliyet, enerji, zaman ve riskin en aza indirgenmesinde veya kalite, hız, doğruluk ve verimin en üst düzeye çıkarılmasında gerçekleşir. Örneğin, maliyet ve hizmet kalitesini optimize etmek için bir ağ önermek için birçok olası yöntem vardır, zamanı optimize etmek için bir yapı tasarlamak için birçok olası yöntem mevcuttur. Mühendislik, fen bilimleri, endüstri, ekonomi ve işletme alanlarındaki birçok gerçek hayat optimizasyon zorluğu karmaşık ve anlaşılması zor. Mantıklı bir zaman diliminde doğru bir şekilde çözümlenemezler. Yaklaşık algoritmalar uygulamak, bu tür sorunları çözmek için güçlü bir seçimdir. Yaklaşık algoritmalar, belirli sezgisel ve meta-nitelik olarak iki kategoriye ayrılır [126]. Özel sezgisel tarama problemlere dayanır ve ayrıca belirli bir soruna teklif edilir ve uygundur. Metaheuristik algoritmalar bilgisayar bilimleri mühendisliği ve matematikte algoritmalar ve hesaplamalı karmaşıklık teorisi ile bağlantılı bir optimizasyon alanıdır. Meta-teknik, çeşitli araştırma kapılarında ve gerçek hayatta iyi bilinmektedir. Bu durumu gösteren işaretlerden biri, çok sayıda toplantı, konferans ve çalıştayın meta-teknolojinin tasarımı ve kullanımı ile para kazanmasıdır. Son 15 yıl içinde, sanatsal zekâ, yumuşak bilgi işlemler, hesaplama zekâsı, matematiksel programlama ve işlem araştırması gibi farklı alanlarda pek çok metaheuristik algoritmanın gelişi görülmüştür ve bilim adamları tarafından başarılarına kanaat getirilmiştir.

1.14. Genetik

Farklı bünyeye sahip olan optimizasyon problemlerini halletmek için çeşitli algoritmalar üretilmiş ve geliştirilmiştir. Bu problemlerin bünyesi karmaşıklıkça, algoritmaların verimli çözümler bulma olasılıkları'da oldukça azalmaktadır. İlerleyen ihtimal tabanlı algoritmalar ile ortaya çıkmış çözümlerde global maksimum veya minimum noktalarını elde edeceğimize dair bir garanti veremeyiz. Fakat bulduğumuz değer ve gerçek değer arasındaki hatayı en küçük miktarlara çekilmesini sağlamamız mümkündür. Günümüzde birçok ihtimal tabanlı programlar vasıtasıyla karmaşık bünyeye sahip olan optimizasyon problemleri yaklaşık olarak hesaplanmaktadır. Basit bünyeye sahip olan problemleri çözmek için klasik algoritmalar yeterli olsa bile karışık bünyeye sahip olan problemler yapay zekâ özelliklerini kullanarak çözümlenmektedirler. Doğada mevcut olan bazı modelleri örnek alarak gelişen algoritmaların en popülerleri genetik algoritmalarlardır. Genetik algoritmalar bahsettiğimiz tekniklerden biridir. Bir seri doğal gelişme geçirmekte olan canlıların değişim sürecine evrim denilmektedir. Evrim hususundaki açıklamalar ve düşünceler 16. Ve 17. Yüzyıllarda biyolojinin modernleşmesi ile birlikte yaygınlaşmıştır ve doğrudan kanıtlanmamış çeşitli teorilere'de açıklama getirmiştir. Bu vasıtayla geçmiş ile gelecek arasındaki değişimlerin yorumlanmasını kolaylaştırarak canlı çeşitlerinin birbiriyle olduğu ilişkileri ortaya çıkarmaya çalışılmıştır. Belirli biyolojik işlemlerin gelişmesi ile birlikte canlılar ortaya çıkmaktadırlar. Bir kuşaktan kuşağa aynen iletilen karakterlere kalıtsal karakter tanımı verilir. Saç rengi, göz rengi, kan gurubu, kıvrırc veya duz saç oluşuna benzer özellikler kalıtsal karakterlerdir. DNA moleküllerinin (Deoxyribonucleic Acid) yer aldığı Kromozonun içinde spesifik özellik ile ilgili bilgisi olan bölüme gen denir. Bu genlerin tümüne veya semboller ile gösterilmelerine genotip denir. Canlıların veya başka deyişle genotiplerin dışardan görünüş şekillerine fenotip denir. DNA ve RNA'lar (Ribonucleic Acid) genlerin yapısını oluşturmaktadırlar ancak yapı taşları moleküllerin dizilmiş olan değerlerinden farklılardır. Yukarda ifade ettiğimiz nedenlere göre canlılarda özel bir enzim türü veya spesifik bir protein sentez yapısının belirli özellikleri sağlamasını sağlayan bölüm gendir [127]. Belirli sınırlar ve boyutlar çerçevesinde oğul evlat olarak tanımlanan yavrulara aktarılır. Bu sebepten dolayı belirli oranlarda yavruların babalarına benzemesi sağlanır. Biolojinin en önemli dallarında biri olan genetik ya da başka deyişle kalıtım bilimi, babalar oğul evlatların birbiyle benzerlik ve farklılıklarına sebebep olan kaynakları ve bu özelliklerin evlatlara nasıl aktarıldığını incelemektedir [128]. Bu

özelliklerin bir sonraki kuşaklara aktarılmasını incelemeye katılım veya soya çekim bilimi denir. Atalar ve evlatların birbiriyle olduğu benzerlik ile birlikte varyasyon adı verilen farklılıklar da vardır. Çevre ve katılım olmak üzere Kuşaklar arasındaki benzerlik ve farklılıkların oluşumuna neden olan iki önemli faktör olarak bilinir [129].

Canlı soyların birbiriyle benzerlik ve farklılıklarını öne süren ve ayrıca DNA yapısında şifrelenmiş genetik bilgilerin hepsine katılım denir. Bahs ettiğimiz bilgiler genler vasıtasıyla taşınır. Canlıların yaşadıkları yaşam boyunca etkileyen fiziksel etkenleri, etkileyen maddeleri ve yaşadıkları doğal ortam çevre olarak bilinir. Başka bir ifade ile, çevre genlerin piyasaya çıkma derecelerini kararlaştıran karmaşık dış etkindir. Canlıların yaşadıkları çevrenin etkisine maruz kalarak dış görüntülerinde uğradıkları değişimlere modifikasyon denilmektedir. Bahsettiğimiz modifikasyon işlemi hücrelerin genetik bilgilerine dokunmayıp sadece ve sadece sonradan elde edilen özellikleri kapsamaktadır. Buyüzden katkısız değildirler. G.J Mandel katılım alanını kuran bilim adamı olarak tanılan (1822-1884) yıllar arasında araştırdığı malzeme ile ilgili elde ettiği sayısal sonuçlara dayanarak genetik alanında açıklamalar getirmeyi düşünmüştür. Belirli bir canlının değişik binyedeki bireylerinin birbiriyle eşleşmesine ve kenetlenmesine malzeme denir. Başka bir ifade ile hibritleşme olarak da tanımlanmaktadır. Ortaya çıkan döleye hibrit veya melez adı da verilmektedir [130]. Mandel bezeliyeler üzerinde yaptığı deneylerde benzer özelliklere sahip olan çok sayıda evlat çaprazlama yaparak çok sayıda oğul evlat elde etmiştir. Yaptığı bu çalışmada bezeliyeleri seçmesini gerçekleştirirken hızlı bir şekilde büyümesi, kendi kendine çoğalma ve lekesiz ve tam evlatlar verebilme gibi faktörleri göz önünde bulundurulmuştur. I.Rechenberg'in 1960'lı yılların başında "Evrin Stratejileri" olarak tanılan çalışmasında ilk defa evrimsel hesaplamayı tanımlamıştır. Başka bilim adamları ve biyoloji araştırmacıları Rechenberg'in görüşlerinden faydalanarak bu konuyu geliştirmişlerdir. Evrim sürecinin bilgisayar desteği ile çözüm yöntemleri öğretebileceği hususunda ilk defa John Holland açıklamalar yapmıştır. 1970'li yıllarında John Holland'ın düşüncesinin peşinde giden öğrencileri ve bilim arkadaşları sayesinde genetik algoritmalar geliştirilmiştir [131]. 1990' yılların başında ise John Koza genetik algoritma destekli ve değişik problemleri çözebilen programlar geliştirerek genetik algoritma gelişiminde devrim yaratmışlar ve bu yönteme genetik programlama adını vermişlerdir [132].

1.15. Genetic Algoritmalar (GA)

Genetik algoritmalar, bilim adamı Darwin'in doğal seçim ve evrim mekanizması prensiplerine dayanan bir arama ve optimizasyon yöntemidir. Yukarıda bahsedildiği gibi bu çözüm ilk defa, John Holland , arkadaşları ve öğrencilerinin birlikte yaptığı araştırmalar sonucu (1970'li yıllarda) ortaya çıkmıştır. Sunulan algoritma önbilgi ve varsayımları kullanmadan sadece uygunluk fonksiyonuna dayanarak çalışabilmektedir [133].

Karmaşık çok boyutlu araştırma uzayında en iyinin hayatta kalması temeline göre bütünsel en iyi çözümü bulur. Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya çıkmıştır. Holland 1975 yılında yaptığı çalışmaları ve sonuçlarını “Adaptation in Natural and Artificial Systems” adlı kitabında bir araya getirip yayınlamıştır. Öncelikle Holland evrim kurallarını genetik algoritmalar içinde optimizasyon problemleri için kullanmıştır.

Genetik algoritmalar problemlere sadece bir çözüm oluşturmak yerine değişik ve farklı çözümlerden ortaya çıkan bir çözüm kümesi oluşturur. Böylece, arama uzayında aynı anda birçok nokta incelendirilmekte ve neticede bütünsel çözüme varma olasılığı yükselmektedir. Çözüm kümesindeki çözümlerin hepsi birbirinden bağımsızdır. Her biri çok boyutlu uzayda birer vektördür.

Genetik algoritmalar sorunların çözümü için evrimsel süreci yazılımlarda taklit ederler. Diğer optimizasyon metodlarında olduğu gibi çözüm için tek bir sistem inşa etmek yerine, böyle yapılardan oluşan bir küme geliştirirler. Problem için uygun birçok çözümü gösteren bu küme genetik algoritma terminolojisinde nüfus (populasyon) isimini alır. Populasyonlar vektör, kromozom veya birey adı verilen sayı dizilerinden meydana gelir. Birey içindeki her bir elemana gen denilmektedir. Populasyon daki bireyler evrimsel süreç içinde genetik algoritma işleyiciler tarafından belirlenirler [134]. Problemin bireyler içindeki izlenim sorundan soruna değişiklik gösterir. Genetik algoritmaların problemin çözümlerindeki başarısına karar vermedeki en önemli etken, sorunun çözümlerini temsil eden bireylerin izlemidir. Nüfus içindeki her bireyin soruna göre çözüm olup olmayacağına karar veren bir uygunluk işlev vardır. Uygunluk fonksiyonundan en yüksek değere sahip olan bireylere, nüfustaki diğer bireyler ile çiftleşmeleri için şans verilir. Bu genler çaprazlama işleminden sonra yeni genler yani çocuk üretirler. Çocukları onları üreten ebeveynlerin (anne, baba) özelliklerini taşır. Bulunur [135]. Yeni genler üretilirken uygunluk değeri kötü olan genler daha az seçileceğinden bu genler bir süre sonra

populasyon dışında bırakılırlar. Yeni nüfus, bir önceki populasyonun uygunluğu yüksek bireylerin bir araya gelip çoğalmalarıyla oluşur. Aynı zamanda bu nüfus önceki populasyonun uygunluğu iyi bireylerinin sahip olduğu özelliklerin çoğu kısmını içerir. Böylece, birçok nesil aracılığıyla iyi özellikler cemiyet içersinde dağılırlar ve genetik işlemler vasıtasıyla da diğer iyi özelliklerle kaynaşırlar. Uygunluk değeri yüksek olan ne kadar çok gen bir araya toplanıp, yeni genler oluştursa arama alanı içinde o kadar iyi bir çalışma uzayı bulunur [136]. Probleme ait en iyi çözümün elde edilmesi için; Bireylerin izlemi uygun bir şekilde yapılmalı, güçlü bir şekilde uygunluk fonksiyonu oluşturulmalı, uygun genetik işlemciler seçilmeli.

Böylece çözüm kümesi problem için bir noktada birleşecektir [137]. Genetik algoritmalar, diğer optimizasyon yöntemleri kullanılırken büyük güçlükler ile karşılaşılan, oldukça büyük arama alanına sahip sorunların çözümünde başarı sergilemektedir. Bir problemin bütünsel en iyi çözümünü bula bilmek için bir garanti yoktur. Sadece problemler uygun bir süre içinde, kabul edilebilir, iyi çözümler bulurlar. Genetik algoritmaların temel amacı, hiçbir çözüm yöntemin bulunmayan problemlere çıkış yolu aramaktır [138] Kendilerine özgü çözüm yolları olan özel problemlerin çözümü için kesin sonucun hızı ve mutlaklığı açısından genetik algoritmalar kullanılmazlar. Genetik algoritmalar sadece; Geniş ve karmaşık arama uzayına sahip olan problemler, Mevcut bilgiyle kısıtlı arama uzayında çözümün zor olduğu, problemin belirli bir matematiksel modeli olmadığı, Geleneksel optimizasyon yöntemlerinden istenen sonucun alınmadığı uzaylarda etkin ve kullanışlıdır. Genetik algoritmalar parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, planlama, yapay zekâ uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel optimizasyon problemleri ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer optimizasyon yöntemlerinin yanında başarılı sonuçlar vermektedir [139].

1.15.1. GA'nın Diğer Yöntemlerden Farkı

- Genetik algoritma problemlerin çözümünü parametrelerin değerleriyle yerine onların kodlarıyla arar. Parametreler kodlanabildiği halde çözüm bulunabilir. Bu nedenle genetik algoritmalar ne yaptığı ile değil, nasıl yaptığını bilir.

- Genetik algoritmalar tek bir noktadan yerine, noktalar kümesinden aramaya başlar. Böylece, genellikle bölgesel en iyi çözümde tıkanıp kalmazlar.
- Genetik algoritmalar türev değil belki uygunluk fonksiyonunun değerinden yararlanır. Böylece, yardımcı bir verinin kullanılmasını gerektirmez.
- Genetik algoritmalar deterministik kuralları değil olasılıksal kurallardan yararlanır.

1.15.2. Basit Genetik Algoritma

Birçok alanda uygulama olasılığı ve uygulamaları olan genetik algoritmaların çalışma basamakları şöyle izah edilebilir [140]:

- Arama uzayındaki bütün olası çözümler dizi olarak kodlanır.
- Çoğunlukla rastsal bir çözüm kümesi seçilir ve başlangıç nesil olarak kabul edilir.
- Her bir diziye bir uygunluk değeri hesaplanır, hesaplanan uygunluk değerleri dizilerin çözüm niteliğini gösterir.
- Bir takım dizi belli bir olasılık değerine göre rastgele seçilip yavrulama işlemi gerçekleştirilir.
- Yeni kromozomların uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine başlatılır.
- Bundan önce belirlenen nüfus sayısı boyunca yukarıdaki işlemler devam ettirilir.

İterasyon, belirlenen kuşak sayısına vardığında işlem sona erdirilir. Amaç fonksiyonuna göre en mantıklı dizi seçilir [136].

1.15.3. Çözümlerin (Kromozomun) Kodlanması

Bir problemin genetik algoritma ile çözümü için ilk adım, bütün çözümlerin aynı büyüklükte olan genler dizisi şeklinde gösterilmesidir. Dizilerden her biri, problemin muhtemel çözümler uzayındaki rastsal bir noktayı temsil eder [141]. Parametrelerin kodlanması, probleme özgü verilerin genetik algoritmanın kullanacağı şekile çevrilmesine imkân verir [136]. Problemlere göre farklı kodlama yöntemleri sunulmaktadır:

- İkili Kodlama
- Permutasyon Kodlama
- Değer Kodlama
- Ağaç Kodlama

1.15.4. İlk Populasyonun Oluşturulması

Mümkün çözümlerin kodlandığı bir çözüm topluluğu oluşturulur. Çözüm grubu populasyon, bunların kodları da kromozom olarak tanımlanır. İkili kodlama yöntemin kullanıldığı kromozomların gösteriminde, ilk neslin oluşturulması için tesadüfi sayı üreticileri kullanılabilir. Bu sayı üreticisi çağrılır ve değer 0,5'den küçükse konum 0'a değilse 1 değerine ayarlanır [139].

Gen sayısının ve kromozom uzunluğunun kısa olduğu problemlerde şans ile de konum değerleri belirlenebilmektedir. Genetik algoritmalarda ikili kodlama tekniği dışında, bazı sorunlarda probleme bağlı olarak farklı kodlama metodları da kullanılmaktadır [133].

1.15.5. Uygunluk Değerinin Hesaplanması

Bir populasyon oluşturulduktan sonraki ilk adım, bireylerin uygunluk değerini hesaplama adımıdır. Örneğin, bir maksimizasyon problemde i . bireyin uygunluk değeri $f(i)$, genelde o noktadaki amaç fonksiyonunun değeridir [142]. Her problem için bir uygunluk fonksiyonu vardır. Belli bir kromozom için uygunluk fonksiyonu, o kromozomun gösterdiği çözümün kullanımıyla veya kabiliyeti ile orantılı olan sayısal bir uygunluk değeri verir. Bu veri, her nesilde daha mantıklı çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar iyise, yaşama ve kaynaşma şansı o kadar fazladır ve bir sonraki nesilde temsil edilme oranı da o kadar yüksektir [143].

1.15.6. Çoğalma İşleminin Uygulanması

Çoğalma operatöründe diziler, amaç fonksiyonuna göre kopyalanır ve iyi intikal eden özellikleri gelecek kuşağa daha iyi taşıyacak genler seçilir [144]. Çoğalma operatörü yapay

bir seçimdir. Dizileri uygunluk değerlerine göre kopyalama, daha iyi uygunluk değerine sahip dizilerin, bir sonraki nesilde bir veya daha fazla çocuğa daha yüksek bir ihtimalle katkıda bulunması anlamına gelmektedir. Üreme, bireyleri seçme işleminden, seçilmiş genleri bir çiftleşme havuzuna kopyalama işleminden ve havuzda bireyleri eşler halinde gruplara ayırma işleminden oluşur [145].

Uygunluk değerinin hesaplanmasından sonra eldeki nesilden yeni bir populasyon oluşturulmalıdır. Seçim işlemi, bir sonraki populasyon için yavru üremek amacıyla hangi anne ve babanın yer alması gerektiğine karar vermektedir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere yayılma şansı tanımaktır. Bir dizinin çoğalma şansı, uygunluk fonksiyonuyla hesaplanan dizinin uygunluk değerine bağlıdır [146].

1.15.7. Çaprazlama İşleminin Uygulanması

Eldeki gen havuzunun olasılıklığını araştırmak üzere, bir önceki nesilden daha iyi kalite içeren yeni kromozomlar üretmek nedeni ile çaprazlama operatörü kullanılmaktadır. Çaprazlama genelde, verilen bir çaprazlama oranına eşit bir olasılıkla seçilen aile türlerine uygulanmaktadır [147].

Genetik algoritmanın verimini etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal kuşakdaki çaprazlamaya karşılık gelmektedir. Üretim işlemi sonucunda elde edilen yeni nesilden tesadüfen iki kromozom seçilmekte ve karşılıklı çaprazlama işlemine almaktır [148].

1.15.8. Mutasyon İşleminin Uygulanması

Çaprazlama eldeki gen potansiyellerini araştırırken kullanılır. Ancak populasyon gereken tüm kodlanmış bilgiyi kapsamaz ise, çaprazlama yeterli bir çözüm üretemez. Bu yüzden, var olan kromozomlardan yeni kromozomlar yaratma yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon yerine getirir [149]. İkili kodlama sisteminin yararlandığı problemlerde mutasyon, az bir olasılık değeri altında bir bit değerini (0 veya 1) diğer bit değerine dönüştürür. İkili kodlama sisteminin kullanılmadığı sorunlarda ise daha farklı dönüşüm tekniği kullanılmaktadır. Hangi metod kullanılırsa kullanılsın, mutasyonun yaygın amacı, genetik çeşitliliği sağlamak veya korumaktır [150].

1.15.9. Yeni Kuşağın Oluşması ve Döngünün Durdurulması

Yeni nesil çoğalma, çaprazlama ve mutasyon operasyonunun dan sonra tanımlanmakta ve bir sonraki jenerasyon ebeveynleri olmaktadır. Süreç yeni nesil ile çoğalma için tanımlanan uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe varana kadar ya da başka bir durdurma kriteri karşılanana kadar devam eder [151].

Zorunlu hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve tekrarlama bu sayıya eriştiğinde döngü durdurulabilmektedir. Durdurma ölçütü iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir [152], [153].

1.15.10. Genetik Algoritmalarda Parametre Seçimi

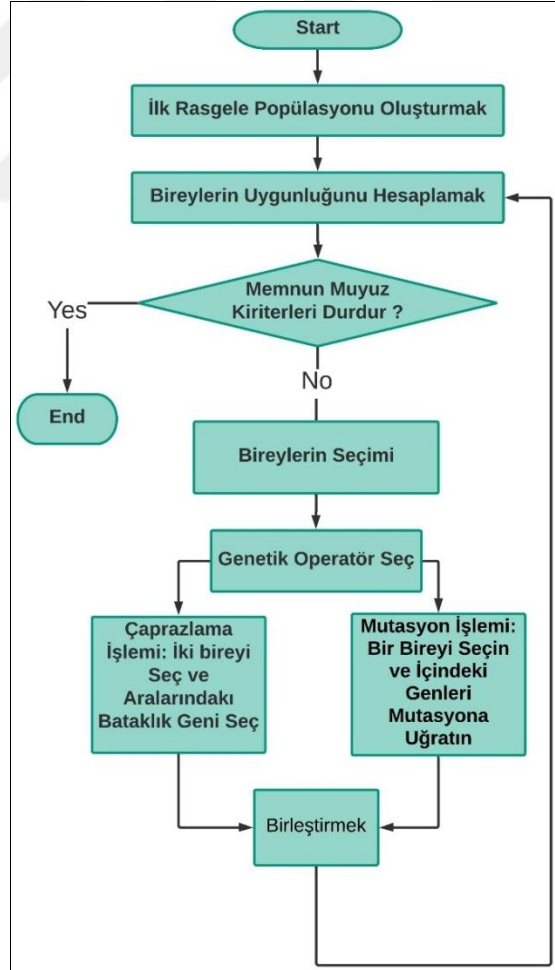
Parametreler, genetik algoritma verimi üzerinde önemli izlenime sahiptir. Optimal kontrol parametreleri sağlamak için birçok çalışma yapılmıştır ancak tüm problemler için genel olarak kullanılabilir parametreler elde edilememiştir [154].

Bu parametreler, kontrol parametreleri olarak tanımlanmaktadır. Kontrol parametreleri popülasyon boyutu, çaprazlama imkânı, mutasyon olasılığı, nesil aralığı, seçim taktiği ve fonksiyon ölçeklemesi olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır [155]:

- **Kuşak Büyüklüğü:** Genetik algoritma kullanıcısının verdiği en önemli kararlardan birisidir. Bu değer çok az olduğunda, genetik algoritma bölgesel bir optimuma takılabilmektedir. Nüfusun çok büyük olması çözüme ulaşma zamanını arttırmaktadır. Bu hususda Goldberg 1985’de, sadece kromozom uzunluğuna bağlı bir kuşak büyüklüğü hesaplama tekniği geliştirmiştir. Ayrıca Schaffer ve arkadaşları 1989’da çok sayıda test fonksiyonları araştırmalar neticesinde, 20-30 arası bir kuşak büyüklüğünün iyi sonuçlar verdiğini göstermişlerdi.
- **Çaprazlama Olasılığı:** Çaprazlamanın amacı, eldeki iyi kromozomların niteliklerini birleştirerek daha iyi kromozomlar üretmektir.
- **Mutasyon Olasılığı:** Mutasyonun hedefi kuşaktaki genetik çeşitliliği kayırmaktır. Eğer dönüşüm olasılığı artarsa, genetik arama tesadüfi bir aramaya

dönüşür. Fakat bu aynı anda kayıp kormozumu tekrar bulmada yardımcı olmaktadır.

- Kuşak Aralığı: Her popülasyondaki yeni kromozom oranına kuşak aralığı denmektedir. Genetik operatörler kaç tane kromozomun seçildiğini gösterir.
- Seçim Stratejisi: Eski nesili yenilemenin çeşitli teknikleri mevcuttur. Kuşaksal stratejide, eldeki popülasyonda kromozomlar tamamen çocuklar ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilendiğinden dolayı bir sonraki nesile aktarılamaz ve bu nedenle bu taktik en uygun (elitist) stratejisiyle beraber kullanılmaktadır.
- Fonksiyon Değerlendirmesi: Doğrusal ölçekleme, üstsel ölçekleme gibi metodlar mevcuttur. Probleme göre en mantıklı ölçekleme yönteminin seçilmesi genetik algoritmanın verimli işlemesi açısından önem taşımaktadır.



Şekil 28. Genetik algoritmanın akış diyağramı

1.16. Parçacık Sürü Optimizasyon Algoritmaları (PSO)

Parçacık sürüsü optimizasyonu (PSO), Kennedy ve Eberhart (1997) tarafından oluşturulan ve daha sonra Pedersen tarafından önerilen modifikasyonların bütünleşmiş edildiği, popülasyon bazlı bir rassal arama algoritmasıdır [156]. PSO, bazı sosyal hayvanların davranışlarının ve böcek iletişimi, balık okulu ve kuş sürüsü gibi dinamik hareketlerin simülasyonuna dayanan, biyolojik ve sosyolojik olarak ilham veren bir sürü istihbarat yöntemi olan bir arama ve optimizasyon yöntemi olarak geliştirilmiştir. PSO algoritmasında, her parçacık optimizasyon için aday bir çözümü temsil eder ve popülasyon bireyleri, belirli bir hızda hareket eden ağırlıksız ve hacimli parçacıklar olarak kabul edilir. Her bir parçacık diğer parçacıklara göre hızını dinamik olarak ayarlayabilir ve aynı zamanda grupların görevdeşlik içinde ilerleyebilmeleri için deneyime atıfta bulunabilir. Parçacık hızı, parçacık tarafından lokalize edilmiş en iyi konum ve çevresindeki en iyi konum olan arama alanındaki iki ana pozisyona doğru çekilmesine dayanarak değerlendirilir. Orijinal PSO algoritması, her bir parçanın indeksine dayanarak oluşturulan birçok çevre komşuluk topolojisi ile uygulanabilir. Genel durum için altı ana kategori belirtilmiştir: [157]. PSO algoritmasındaki hızlı ve kolay uygulama kabiliyeti onu Genetik algoritmalar gibi diğer evrimsel algoritmalara göre daha avantaj katmaktadır [158]. Optimizasyon problemlerin çözmesinde, problemlerin parametre belirlemesi ve kontrollerin tasarımı gibi alanlarda PSO algoritması kullanılmaktadır [159]. PSO davranışının kaygısı, her bir parçacığın hafızasına ve seneler boyunca elde edilen sürünün bilgisine dayanarak bazı gruplamalar yapmak için sosyal güçleri kullanmaktır. Sürü zekâsı bu davranışı yönlendiren görüngü olarak tanımlanmaktadır.

Sürü zekâsı, ilk olarak Hackwood ve Wang (1988) tarafından hücrel robotik sistemlerde kullanıldığı için multidisipliner bir karakter olarak işaretlendi [160]. Bireylerin birbirleriyle ve çevreleriyle ilgili ortak davranışlarına odaklanan bir disiplindir. Kuş sürüsü, balık sürüsü, kara hayvanlarının sürüsü, Şekil 29, 30 ve 31'de gösterildiği gibi sürü zekâsına örnektir. Sürü zekâsı dört prensipten oluşur. Yakınlık ilkesi olarak adlandırılan ilk prensip, alan ve zaman hesaplamaları kümesinin temel işler olması gerektiğine inanmaktadır. İkincisi, sürünün gıda maddeleri veya yerin güvenliği gibi kalite faktörlerine cevap verebildiğini kabul eden kalite ilkesidir. Üçüncü ilke, sürünün kaynakları birçok mod boyunca dağıtmasını sağlayan çeşitli ilkesidir. Son olarak, sürünün davranışının,

ortamın her dalgalanması üzerine bir moddan diğere bir deęişiklik yapmaması gerektiğini beyan etmesiyle istikrar ilkesi.



Şekil 29. Bird flocking



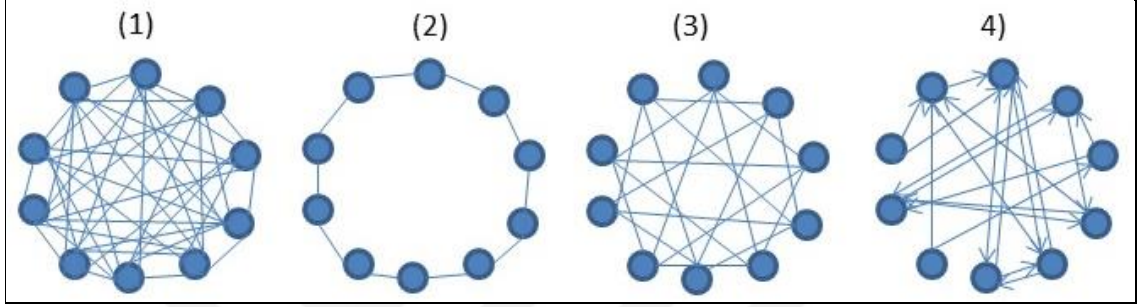
Şekil 30. Fish shoolling



Şekil 31. Animal herding

1.16.1. PSO Algoritmasının Farklı Topolojileri

PSO algoritması farklı topolojilere sahiptir. Topolojilerin tanımı ve arasındaki farklar bu bölümde tanımlanacaktır. Bu çalışmamızda global optimum kullanılmaktadır. Parçacık Sürü optimizasyon algoritmasının farklı topolojileri şekil32'de görüldüğü gibi global optimum (1), yerel optimum (2), Van Neuman (3), rasgele (4) olarak sunulmaktadır.



Şekil 32. PSO algoritmasının değişik topolojileri

1.16.1.1. Kare Topolojisi (Van Neuman)

Van Neuman veya Kare topolojisi olarak bilinen topolojide Neuman, materyallerin moleküler bünyesinden ilham alarak topolojiyi sunmuştur. Bu topolojinin yapısında iki boyutlu şebeke içerisinde her bir parçacık dört komşu parçacık (aşağı, yukarı, sol, sağ) ile ilişkisi vardır. Her bir zerre veya parçacık kendisine ait olan iyi tecrübeyi veya diğer dört komşu zerrelerinin iyi tecrübelerini kullanarak ilerler. Bu topoloji tipinde parçacıkların arasındaki ilişkiyi global optimum topolojisine göre sınırlıdır ancak yerel optimum topolojisine göre parçacıkların birbiriyle olduğu ilişki daha fazladır. Yakınsama açısından incelediğimizde kare topolojisi global optimum topolojisine göre daha yavaş ve yerel optimum topolojisine göre daha hızlıdır.

1.16.1.2. Halka Topolojisi (Yerel Optimum)

Halka topolojisi veya yerel optimum topolojisi olarak bilinen topoloji yapısında, her bir parçacığın iki komşu parçacık ile ilişkisi vardır. Yerel optimum topolojisinde her parçacık kendine ait olan en iyi tecrübe ve komşusundan aldığı en iyi tecrübeyi (bilgi)

taşıyarak bir sonraki hareketine devam eder. Bu nedenler ile bu topolojide yakınsama daha uzun sürebilir.

1.16.1.3. Yıldız Topolojisi (Global Optimum)

Yıldız topolojisi veya global optimum topolojisi olarak bilinen bu topoloji yapısında, her bir parçacık komşuluğundaki olan diğer parçacıklar ile ilişkisi olabilir. Dolaysıyla bu topolojide her parçacık toplumdaki en iyi parçacığın tecrübesini taşımaktadır ve bu bilgiyi ve çevresindeki en iyi bilgiden yararlanarak bir sonraki hareketini etmektedir.

1.16.2. Temel PSO Parametrelerinin Denklemleri (Hız ve Hareket)

PSO algoritmasının temel parametreleri birçok bölüme ayrılmıştır, ancak sadece en önemlileri aşağıdaki gibi açıklanmıştır:

- Bilişsel ve sosyal parametreler, C_1 and C_2

Ayrıca hızlanma sabiti olarak da adlandırılırlar, güven derecesini belirten " güven " ayarlarını temsil ederler ve parçacıkların yörüngesini belirleme konusunda önemli yer almaktadırlar. En iyi çözümü bulma sürecinde C_1 (bilişsel parametre), parçacıkların kendi içine yerleştirdiği güven yüzdesini belirtirken, sürünün tamamının güven yüzdesi C_2 (sosyal parametre) ile ifade edilir. Genellikle C_1 ve C_2 2 alınır, ancak

$0 < C_1 + C_2 < 4$ göre farklı değerler de düşünülebilir [161].

- Her boyut için bir yineleme sırasında izin verilen maksimum hızı içeren vektör V_{max}

$$V_{max} = k * x_{max} \quad (0,1 < k < 1) \text{ burda } x: \text{Arama uzunluğu olarak tanımlanır.} \quad (53)$$

Parçacığın hızı, aşağıdaki formülle ifade edilen Shi ve Eberhart'ın (1998) sürtünme sabiti modeline dayanarak hesaplanır.

$$V_{id}(t+1) = \omega V_{id}(t) + c_1 r_{1d}(t) (p_{id}(t) - x_{id}(t)) + c_2 r_{2d}(t) (g_d(t) - x_d(t)), \quad (54)$$

Bu formülde:

- $V_{id}(t)$: t zamanında d boyutundaki hız.
- $P_{id}(t)$: t zamanında d boyutundaki pozisyon.
- ω : sürtünme sabiti.
- G_d : t zamanında d boyutundaki global pozisyon.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} * \text{iter} \quad (55)$$

- İter : Tekrarlama veya iterasyon sayısı.
- $\text{İter}_{\max}$: Maksimum iterasyon sayısı.
- $\omega_{\max}, \omega_{\min}$: ω sabitinin sırayla maksimum ve minimum miktarlarıdır.
- c_1, c_2 : sırasıyla bilişsel ve sosyal katsayılar.
- r_1d, r_2d : algoritmanın random sayıları.
- $p_{id}(t), g_d(t)$: sırasıyla d boyutunda kişisel ve genel en iyi tecrübeleri.

Maksimum hız değeri genellikle, arama boyutunun aralığının uzunluğunun yarısı ile alt sınır ve üst sınırın ortalama değeri olan eşdeğeri olarak tanımlanır. Ardışık tekrarlamalar sırasında her parçacığın yörünge değişikliklerini düzenlemesi çok faydalıdır ve ayrıca problem arama alanındaki daha geniş açılımların önlenmesi için bir ortalamaı temsil eder. Aramayı ayarlamak ve parçacık aramada olası patlamaları önlemek için $v_{\max}$ değerine küçük değerler atamak daha iyidir. Hız sıkıştırmak uygulaması aşağıdaki gibidir:

$$v_{id}(t+1) = \begin{cases} -v_{\max} & \text{if } v_{id}(t+1) < -v_{\max} \\ v_{\max} & \text{if } v_{id}(t+1) > v_{\max} \\ v_{id}(t+1) & \text{otherwise} \end{cases} \quad (56)$$

Sürtünme sabiti, genellikle hız güncelleme denklemindeki önceki adımlar sırasındaki hıza bağlı ölçek faktörü olarak tanımlanır ve genellikle 1'den az olacak şekilde ayarlanır ancak yineleme sırasında da güncellenebilir. Başka bir deyişle, önceki hızların mevcut yenileme hızı üzerindeki etkilerini kontrol etmek gerekir. Sürtünme sabiti kontrolü stratejileri alanı çok geniştir ve hangi stratejinin en iyi performansa sahip olduğuna dair net bir açıklama yapılmamıştır. Sürtünme sabiti üzerinde yapılan önceki çalışmalarda, çoklu

kontrol stratejilerinin her biri nasıl ve ne zaman, teorik yakınsama kriterine en iyi şekilde uyacağını belirlememiştir. Sürtünme sabiti azaltan PSO, tanınmış araştırmacılar tarafından kullanılır. Tüm stratejiler, büyük bir atalet sürtünme sabiti, global araştırmanın daha kolay olduğunu, küçük bir sürtünme sabiti kullanılırsa yerel araştırmanın kolaylaştırdığını ortaya koydu. Uygun bir sürtünme sabiti parametresi seçimi, daha az yinleme ile optimizasyonu gerçekleştirir. Genelde $\omega_{\max}=0,9$ ve $\omega_{\min}=0,3$ arasında değerler almaktadırlar. Optimizasyon, uygun bir başlangıç popülasyonundan başlar, konumlar ve hızlar küme başlatma olarak adlandırılan bir işlemle tasarım alanı boyunca rastgele bir şekilde dağıtılır. Bir sonraki adım, tasarım alanı pozisyonlarını kullanarak, objektif fonksiyon değerlerini değerlendirmekten ibarettir. Daha sonra her bir yinlemede optimum parçacık pozisyonu ve global optimum parçacık pozisyonu güncellenmelidir. Optimum parçacık pozisyonu güncelledikten sonra, diğer tüm parçacıkların pozisyonları da güncellenir. Son adım, her parçacık için hız vektörü güncellemesi ile ilgilidir. Sonuç olarak, optimizasyon yakınsama kontrolünden sonra sonlandırılır. Daha açık ve net bir şekilde PSO algoritmasının uygulanması alta sunulacaktır. PSO akış şeması Şekil 33’de verilmiştir.

```

1 begin
2   t = 0;
3   initialize particles P(t);
4   evaluate particles P(t);
5   while (termination conditions are
      unsatisfied)
6     begin
7       t = t + 1;
8       update weights
9       select pBest for each particle
10      select gBest from P(t-1);
11      calculate particle velocity
        P(t);
12      update particle position P(t)
13      evaluate particles P(t);
14    end
15end

```

Şekil 33. PSO pseudo cod.

1.16.3. Parçacık Sürü Optimizasyon (PSO) Algoritmanın Uygulama Adımları

1. Bilim adamlarının tanımladığı cost fonksiyonu ya da bildiğimiz maliyet fonksiyonunun tanıtımı ve ayrıca nüfus sayısı, maliyet fonksiyonu değişken

sayısı veya boyut sayısı (D) ve son olarak c_1 ve c_2 gibi PSO parametrelerinin belirlenmesi.

2. Rasgele bir şekilde parçacıkların pozisyon ve hızları için D arama boyutu uzayında bir değer üretilmesi.
3. Popülasyonlarda yer alan her bir parçacık için cost fonksiyonunun hesaplanması:

$$\begin{aligned} &\text{for } i = 1: \text{nüfus sayısı} \\ &F_i(\vec{x}_i) = \text{cost}(\vec{x}_i) \end{aligned} \quad (57)$$

end

4. Parçacıkların her birinin en iyi tecrübesinin hesaplanması:

$$\begin{aligned} &\text{if } F(\vec{x}_i) > F_i(\vec{y}_i) \\ &\vec{p}_i = \vec{x}_i \\ &F(\vec{y}_i) = \text{cost } F(\vec{x}_i) \\ &\text{end} \end{aligned} \quad (58)$$

5. Parçacıklara ait en iyi global tecrübenin hesaplanması:

$$\begin{aligned} &\text{for } i = 1: \text{nüfus sayısı} \\ &\text{if } F(\vec{x}_i) > F(\vec{g}) \\ &\vec{g} = \vec{x}_i \\ &F(\vec{g}) = F(\vec{x}_i) \\ &\text{end} \end{aligned} \quad (59)$$

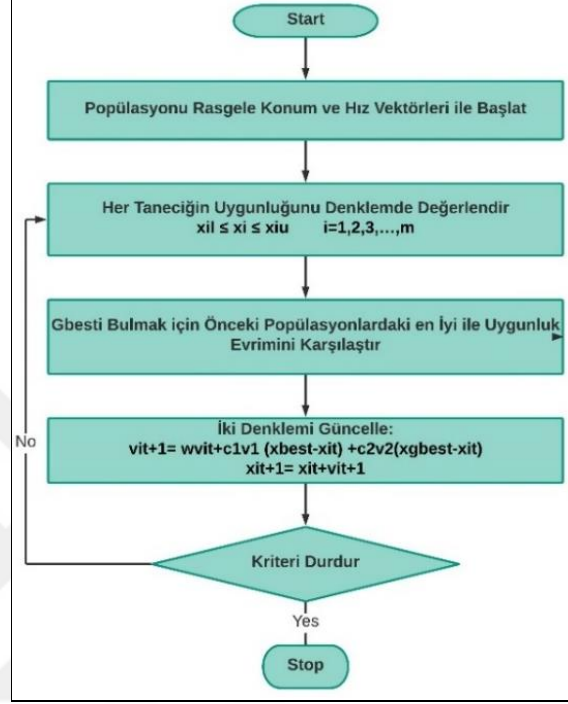
6. Parçacıkların hızının hesaplanması:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_{1d}(t) (p_{id}(t) - x_{id}(t)) + c_2 r_{2d}(t) (\vec{g}(t) - x_i(t)) \quad (60)$$

7. Hareket denklemlerinden faydalanarak parçacıkların pozisyonlarının güncellenmesi:

$$x(t) = x(t+1) + \vec{v}(t) \quad (61)$$

8. Algoritmanın 3 basamağından tekrarlanmasını sağlamak.
9. Algoritmanın 4 basamağından yakınsamaya varana kadar devam etmesini sağlamak.



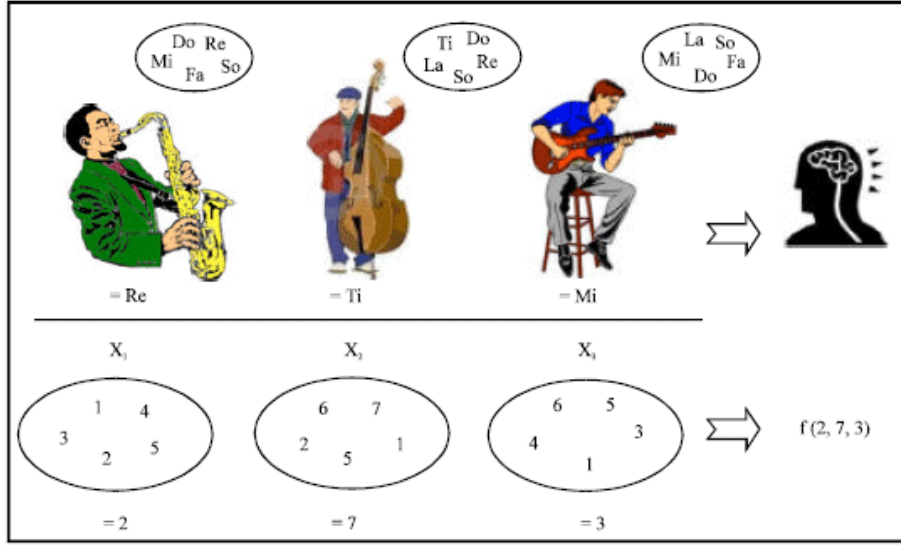
Şekil 34. PSO akış şeması

1.17. Harmoni Arama Algoritmaları (Harmony Search HS)

Doğada uyum, farklı frekanslarda yayılan birkaç ses dalgası ile insan kulağı arasındaki özel bir ilişki ortaya çıkarken, insan kulağı bu müzik tonlarını refleks olarak kabul eder, müzik armonisi, bir grup dinleyiciyi memnun eden, seslerin birleştirici bir yaratımı olarak kabul edilir. İlişki ilk olarak Pythagoras (582 BC-497 BC) tarafından geliştirildi. Yunan filozofu ve matematikçisi Gergin dizi konusundaki ünlü araştırmasından önemli sonuçlar elde etti ve daha sonra bu olgulara odaklanan birçok araştırmacı klasik uyum teorisini kurdu. Tanınmış bir Fransız besteci ve müzikolog olan Jean-Philippe Rameau klasik armoni arama teorisini (1683-1764) kurdu ve daha sonra müzikolog Tirro (1977) amerikan caz tarihi hakkında bir belge hazırladı [162]Harmoni arama algoritması (HS) , Lee ve Geem [163] tarafından, mükemmel bir armoni durumu arayışı olan müzik doğaçlama sürecine dayanan yeni bir meta-sezgisel algoritma olarak tanıtıldı. HS, en iyi

çözümlerle temsil edilen bazı hoş harmonileri ortaya çıkarmak için nüfus üyesi olarak kabul edilen müzik aletleri ile toplu performans sırasında müzisyenlerin davranışlarının doğal fenomenlerinden türetilen en yeni yöntemlerden biridir. Bir müzisyenin notaların daha iyi bir kombinasyonunu aradığı Caz doğaçlaması üzerinde durulur. Caz doğaçlamasının temel amacı, estetik bir standarda dayanan bir müzik aleti için her bir perdenin belirlediği mükemmel durum için, müzikal olarak hoş bir armoni bulmaktır. Son zamanlarda HS, evrimsel hesaplama alanında popüler bir algoritma haline geldi ve uygulama kolaylığı, ayarlama dikkate alınan birkaç parametre ve son olarak kavramın basitliği ile ilgili diğer birçok algoritmaya üstünlüğü nedeniyle çok çeşitli optimizasyon problemlerine başarıyla uygulandı. Bazı avantajları da Geem [164] tarafından özetlenmiştir, HS'yi uygun mühendislik uygulamalarından biri haline getirmektedir. HS algoritması, su dağıtımı ve oyunlar gibi büyük ölçekli problemleri çözmek için kullanılmıştır ve diğer optimizasyon tekniklerine kıyasla daha iyi performans gösterdiği kanatına varılmıştır. Son yıllarda, HS fonksiyon optimizasyonu, boru ağı optimizasyonu, mekanik yapı tasarımı Geem [165], stokastik denge ağı tasarımı ve veri sınıflandırma sistemlerinin optimizasyonu [166] gibi ilgili alanlarda kullanılmıştır.

Armoni, yıllarca hoş bir etki yaratmak için benzersiz bir amaca sahip bazı akorlarla ilgili sesli notaların eşzamanlı olarak üretilmesiyle karakterize edilir. Do, Re, Mi, Fa, Sol, La ve Si, belirli bir melodi üretmek için herhangi bir müzik aleti ile birleşimlerinin yapıldığı notlardır. Müzisyenler her zaman daha iyi bir armoni bulmak için uğraşırlar ve sadece çalınan notaları değiştirmenin çok sayıda bahanesini başararak başarılı olurlar. Belirli bir projedeki tüm müzisyenlerin doğaçlamaları, tek bir armoni vektörü olarak kabul edilir. Deneme performansı sırasında perdeler iyi bir armoni olarak değerlendirilirse, bir sonraki denemelerinde daha iyi bir armoni üretme olasılığı ile her oyuncunun hafızasında saklanır. Bu ilke inşaat mühendisliği optimizasyonunda da benzer şekilde uygulanır. Bir değer aralığı içinde, her karar değişkeni bir seçim yapar ve seçilen tüm değerler, bu vektör iyi bir çözüm olarak değerlendirildiği sürece her değişkenin belleğinde bir çözüm vektörü olarak saklanır. Bir sonraki deneyim sırasında daha iyi bir çözüm elde etme olasılığı artar. Müzik doğaçlama ve mühendislik optimizasyonu arasındaki bu benzerlik Şekil 35'te açıklanmaktadır [167].



Şekil 35. Müzik doğaçlama ve mühendislik optimizasyonu arasındaki benzerlik

Yukarıdaki şekil'de benzetmeyi bir kez daha göstermektedir ve belirtilen detaylar Caz müziğini çalan bu üçlünün esasına dayanarak Lee ve Geem [168]. Tarafından yapılan açıklamalarla anlaşılabilir. Caz müzisyeni tarafından notlarını seçmesi için kullanılan yöntem üç farklı aşamaya ayrılmıştır: önce hafızadan bir perde seçin, ikincisi mevcut bir perdeye yakın bir perde oluşturun veya nihayet olası sesler aralığından rastgele bir perde çalın. Bu üç süreç uyum arama algoritmasının temelini oluşturur. Burada ele alınan bir üçlü üzerinde oluşturulan örnek, her müzisyene aşağıdaki gibi belirli bir miktar perdesi verir: gitarist {Mi, Fa, Sol}, saksofoncu {La, Si, Do} ve kontrbas için {Do, Re, Me}. Kendilerine atanan müzikal armonilerden eğer gitar {Mi Sharp} çıkarırsa saksofon {Sol} üretir ve kontrbas serbest bırakılır {Do}, {Do, Mi Sharp, Sol} yeni uyum vektörüdür. Bu yeni uyum vektörü, uyum hafızasındaki (HM) başka bir uyumdan daha iyi olduğunda, yenisi HM'ye bütünleşmiş edilirken, mevcut en kötüsü HM'den hariç tutulur. İşlem en iyi uyum bulunana kadar tekrarlanır.

1.17.1. Uyum Arama Parametreleri

Uyum arama performansı birçok parametreye sıkı sıkıya bağlıdır: uyum hafıza değerlendirme oranı (HMCR), uyum boyutu (HMS), adım ayarlama oranı (PAR) ve bant genişliği parametresi (BW). Dikkate alınması gereken bir diğer parametre, redaksiyonda daha önce kullanılan uyum hafızasıdır (HM). HM, saklanan ve yeni akorlar (tasarım

değişkenleri) oluşturmak için kullanılacak ses birleşimleri kümesidir (tasarım değişkenleri). HMS, bellekte bulunan akorların (tasarım değişkenleri) sayısını temsil eden bir parametredir. Optimize etme problemine bağlı olarak, literatürdeki çoğu örnek, HMS'nin 10 ila 50 arasında değiştiğini gösterir. HMCR, HM'den bir tasarım değişkeni çizilme olasılığını gösteren $[0,1]$ aralığında seçilen bir parametredir. Örneğin, 0.60'lık bir HMCR değeri, HM'den bir değer seçme olasılığının %60 ve değişkenin HM'den rasgele seçilme olasılığının %40 olduğu anlamına gelir. HM'den bir değer (adım) almaya karar verildikten sonra, alınan değer adım ayarının ayarlanıp ayarlanmayacağını görmek için başka bir test gereklidir. PAR, mevcut tasarımın çevresinde daha iyi tasarım arayışında olan saha ayarlamasından sorumludur. Ayarlama işlemi, yeni bir tasarım değişkeni seçilecekse $HMCR * PAR$ formüllerine ve tasarım değeri tutulacaksa $HMCR * (1-PAR)$ formüllerine göre değerlendirilen olasılığa bağlıdır. PAR ayrıca 0 ile 1 arasında bir değerdir, BW en uygun çözümlere yakınsama oranını önemli ölçüde etkileyen önemli bir faktördür.

1.17.1.1. Uyum Arama Adımları

Uyum araştırması, Şekil.36'da akış şeması ile temsil edilen beş ana adımdan oluşur ve bilgisayar algoritmaları vasıtasıyla uygulanır. Şekil 37'deki sözde kodun açıklanması' da takiben sunulmuştur:

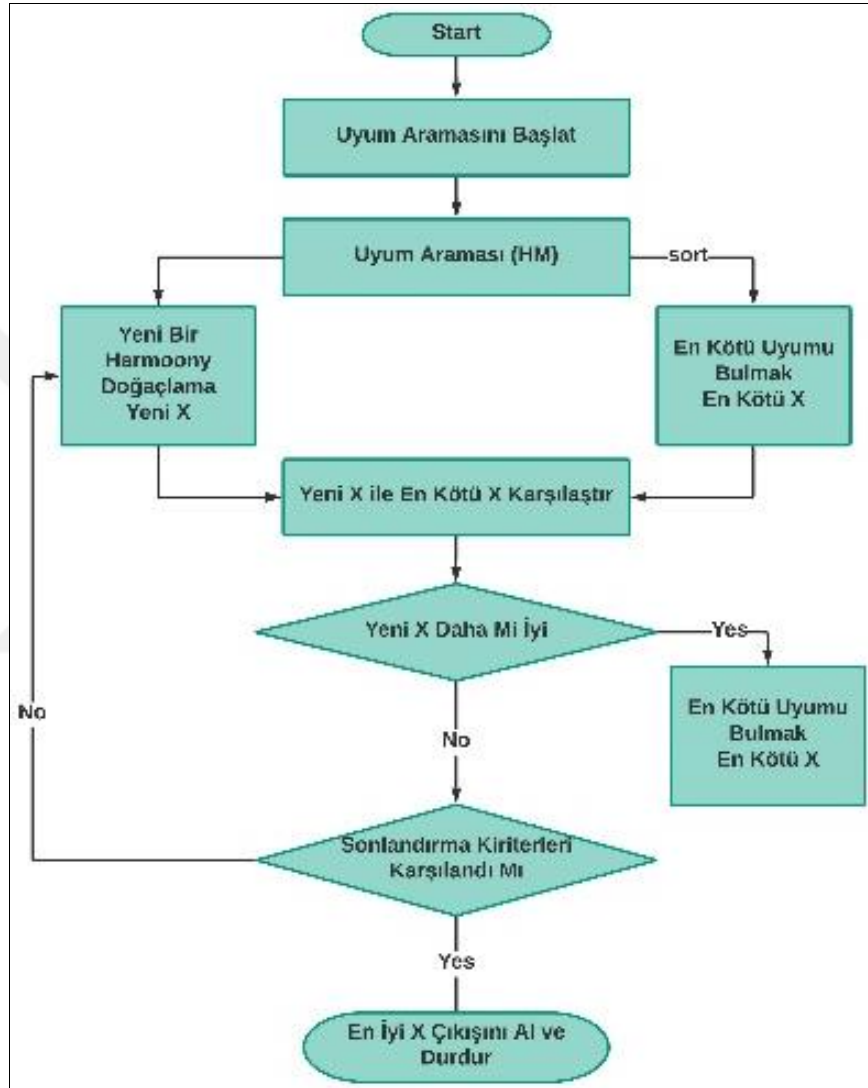
1. Optimizasyon problemini formüle edin ve algoritma parametre değerlerini (HMCR, PAR, HMS, BW) başlatın.
2. Uyum hafızasını başlat.
3. Yeni bir uyum geliştirin ve performansını değerlendirin.
4. Uyum hafızasını gerektiği kadar güncelleyin.
5. Sonlandırma kriterlerini kontrol edin.

Tüm bu adımlar aşağıdaki satırlarda ayrıntılı olarak incelenmiştir.

Birinci adım: Optimizasyon problemi formülasyonu ve HS parametrelerinin belirlenmesi.

Yukarıda belirtilen HS algoritması parametreleri bu adımda seçilir ve problemin türüne bağlı olarak seçimi yapılır. Amaç, alt sınır ve üst sınır ile sınırlı değişkenlerle $f(x)$ 'nin minimize edilmesidir. Uyum arama parametreleri, özellikle çözelti vektörlerinin geliştirilmesinde çok önemli olan HMCR ve PAR için dikkatle belirtilmelidir. Örneğin,

birine yaklaşan nispeten yüksek bir HMCR değeri, HM'de yeni bir değer üretme olasılığını azaltır. Tersine, sıfıra yakın küçük bir HMCR değeri HM'de depolanan birleşimlerinin istismarını azaltır. Bu nedenle, iyi bir parametre seti, en uygun çözümü aramak için algoritmanın kapasitelerini artıracaktır.



Şekil 36. Uyum arama algoritmasının akış şeması

The HS Algorithm

```

begin
Define fitness function  $f(\mathbf{a})$ ,  $\mathbf{a} = (a_1, a_2, \dots, a_N)^T$ 
Define (HMCR), (PAR), (HMS), (EOR), (NI), (bw)
 $HM \leftarrow GenerateInitialPopulation()$ 
min = minimum visible value.
max = maximum visible value.
while ( $iter \leq NI$ ) do
  while ( $i \leq N$ ) do
    if ( $rand \in (0, 1) \leq HMCR$ ) then
       $randLoc = 1 + floor(rand \in (0, 1) \cdot N)$ 
       $a'_i = a_i^{randLoc}$ 
      if ( $rand \in (0, 1) \leq PAR$ ) then
        adjust the value of  $a'_i$  by:
         $a'_i = a'_i \pm rand \in (0, 1) \cdot bw$ 
      end if
    else
      choose a random variable:
       $a'_i = min + rand \in (0, 1) \cdot (max - min)$ 
    end if
  end while
  if ( $f(\mathbf{a}') \leq f(\mathbf{a}^{worst})$ ) then
     $\mathbf{a}' \in HM$ ,  $\mathbf{a}^{worst} \notin HM$ 
  end if
end while
best=find the current best solution
end

```

Şekil 37. HS pseudo cod.

İkinci adım: Uyum hafızasını başlat

Parametreler ayarlandıktan sonra, uyum hafızası matrisi oluşturulur ve rasgele oluşturulan veya toplam sayı uyum hafızası boyutuna eşit bir ayrık değerler popülasyonundan seçilen bazı özel çözeltilerle doldurulur. HM matrisinin her satırı tasarım değişkenlerinden etkilenen değerleri içerir. Uyum hafıza matrisi 61 nolu formülde gösterilmektedir [168].

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & \dots & \dots & \dots & x_N^1 & f(\vec{x}^1) \\ x_1^2 & x_2^2 & x_3^2 & \dots & \dots & \dots & \dots & x_N^2 & f(\vec{x}^2) \\ x_1^3 & x_2^3 & x_3^3 & \dots & \dots & \dots & \dots & x_N^3 & f(\vec{x}^3) \\ \vdots & \vdots & \vdots & & & & & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & x_3^{HMS} & \dots & \dots & \dots & \dots & x_N^{HMS} & f(\vec{x}^{HMS}) \end{bmatrix} \quad (62)$$

Uyum Hafıza Matrisi

HM matrisinde depolanan tasarım değişkenleri, çözeltilere göre azalan sırada depolanan ilgili objektif fonksiyonların sonucunu üretmek için analiz edilir.

Burada $X^j = (X_1^j, \dots, X_N^j)$ bir çözelti vektörüdür, $f(X^j)$ X^j 'deki objektif fonksiyonun değeridir ve X_i^j , $j = 1, \dots, HMS$ için j th çözelti vektöründeki i değeridir. Referanslar [169], [170] ve [171] ve [172] arasındaki fark, ilişkili objektif fonksiyonun değerlerinin HM'ye dahil edilip edilmediğidir. Pratikte, HM matrisi (Figure.) objektif fonksiyonun sıklığına göre sıralanmış bir matristir, örneğin Ingram ve Zhang [172] en iyi çözüm vektörünü HM matrisinin ilk sırasında ve en kötüsü son satır, yani $X_1 = X_{best}$ ve $X_{HMS} = X_{worst}$.

Üçüncü Adım: yeni bir uyum doğaçlama

Üçüncü adımda, Caz müzik çalarlar kavramında belirtilen üç kurala dayanarak yeni bir uyum vektörü oluşturulacaktır: hafıza değerlendirmesi, ses perdesi ayar oranı ve seçim rastgele yapılır. Örneğin, yeni vektörün ilk değeri mevcut HM'den ve diğer değişkenler için seçilebilir. Bu işlem, değerin rastgele oluşturulup oluşturulmayacağını belirlemekten sorumlu olan HMCR adında bir yönetim faktörüne sahiptir. Bileşenin değeri HM'den geliyorsa, PAR değerine bağlı olarak eğim ayarlanabilir. Aralık ayarı, sürekli değişkenler ve ayrık değişkenler için farklıdır.

Dördüncü adım: uyum hafızasını güncelle

Geliştirilen uyum, nesnel işlev değerine ilişkin eski uyumdan daha iyi bir performansa ve öngörülen kısıtlama ihlallerine sahipse, bu en yeni uyum, en kötü olarak kabul edilen eski uyum HM'den çıkarılırken HM'de gerçekleşir. Aksi takdirde doğaçlama uyum basitçe ihmal edilir.

Beşinci adım: sonlandırma kriterleri

Son adımda, uyum aramasının 3. ve 4. adımları sonlandırma kriterleri sağlanana kadar tekrarlanır. HS algoritmasının ilk formülasyonlarında sadece bir sonlandırma kriteri kullanılmış ve maksimum yinleme sayısına dayanmaktadır. İlerlemeyle birlikte,

hesaplama sürecini durdurmak için Cheng ve ark. (2007) tarafından başlatılan yeni bir faktör bütünleştirilmiştir. Bu ikinci kriter, yalnızca çözümde bir gelişme yoksa ve uygulanan yineleme numarasına ulaşmadan önce algoritmayı durdurur.

1.18. Çok Amaçlı Optimizasyon (MOP)

MOP birtakım hedeflere ve genellikle kısıtlamalara sahiptir. Kısıtlamaların herhangi bir uygulanabilir çözüm tarafından (en uygun çözüm dahil) karşılanması gerekir. MOP şu şekilde formüle edilir:

$$\begin{aligned} &\text{Minimiz / Maximiz } f_n(x), n=1,2,\dots,N; \\ &\text{Subjet to } g_j(x) \geq, j=1,2,\dots,J; \\ &h_k(x)=0, k=1,2,\dots,K; \\ &x_i^{(l)} \leq x_i \leq x_i^{(u)}, i=1,2,\dots,m. \end{aligned} \tag{63}$$

Çözüm x , m karar değişkenlerinin bir vektörüdür $x=(x_1, x_2, \dots, x_m)^T$. İlk kısıtlamalar kümesi minimizasyon problemi için eşitsizlik kısıtlaması iken, maksimizasyon problemi için bu sınırlama şuna eşittir i.e $\leq$. Bir sonraki sınırlamalar kümesi, eşitlik kısıtlamaları ve ardından değişken sınırlar olarak adlandırılan son sınırlamalar kümesidir ve her bir karar değişkeni x_i 'nin daha düşük bir $x_i^{(l)}$ bağlı bir üst içinde bir $x_i^{(u)}$ değer almasını kısıtlar. Genel olarak, MOP'ları çözmek için klasik ve Yapay Zekâ (AI) teknikleri kullanılır. MOP'ları çözmek için en popüler iki AI tekniği Evrimsel Yaklaşımlar (EA) ve PSO'dur.

1.18.1. Klasik Yöntemler

Tercih bilgilerinin artan kullanım sırasına göre klasik yöntemler şunlardır: Ağırlıklı toplam yöntemi; ε -kısıtlama yöntemi; Ağırlıklı metrik yöntem; Bensons yöntemi; Değer fonksiyon yöntemi ve Hedef programlama yöntemi. Ağırlıklı toplamda yöntem, her bir hedefin kullanıcı tarafından sağlanan bir ağırlıkla önceden çarpılmasıyla hedefler tek bir hedefte skaler hale getirilir. ε -kısıtlama yöntemi, dışbükey olmayan objektif alanlara sahip problemlerin çözümünde ağırlıklı toplam yaklaşımının karşılaştığı zorlukları, sadece

hedeflerden birini koruyarak ve hedeflerin geri kalanını kullanıcı tarafından belirlenmiş değerler içinde kısıtlayarak MOP'u yeniden düzenleyerek hafifletir. Ağırlıklı metrik yöntemde, ağırlıklı olarak hedeflerin toplamı kullanmak yerine, l_p ve l_∞ mesafe metrikleri gibi ağırlıklı metrikler kullanılır, bu nedenle ağırlıklı metrikler, birden çok hedefi tek bir hedefte birleştirme aracıdır. Bensons yöntemi, referans çözeltinin uygulanabilir pareto olmayan optimal çözelti olarak alınması dışında, ağırlıklı metrik yaklaşıma benzer. Değer fonksiyonu yönteminde kullanıcı, tüm M hedefleriyle ilgili bir $U: R^M \rightarrow R$ matematiksel değer fonksiyonu sağlar. Değer işlevi, mümkün olan tüm arama alanı boyunca geçerli olmalıdır. Hedef programlama, bir veya daha fazla objektif fonksiyon için önceden belirlenmiş bir hedefe ulaşan çözümler bulmaya yardımcı olur. Tüm objektif işlevlerde önceden belirlenmiş hedeflere ulaşan bir çözüm yoksa görev, hedeflerden sapmaları en aza indiren çözümleri bulmaktır. Ancak, istenen hedefe sahip bir çözüm varsa, görev o çözümü belirlemektir.

1.18.2. Evrimsel Yöntemler

Evrimsel yöntemlere dayanan yaklaşımlar temel olarak üç tipe ayrılır [173] Toplama fonksiyonları; Nüfus temelli yaklaşımlar, Pareto tabanlı yaklaşımlar. Toplama işlevleri, herhangi bir aritmetik işlemle tüm hedefleri tek bir amaçta birleştirme kavramını taşır. Doğrusal toplama işlevleri nedeniyle bu yöntemler çok etkileyici değildir. Nüfus temelli yaklaşımlar aramayı çeşitlendirmek için EA'nın nüfusunu kullanır [174], popülasyon temelli yaklaşımların klasik örneği olarak kabul edilen Vektör Değerlendirilmiş Genetik Algoritmayı (VEGA) sunmuşlardır. Her nesilde, oransal seçim ile alt popülasyonlar üretilir. Toplam nüfus boyutu N ve n toplam hedef sayısı ise, alt nüfusun büyüklüğü N / n olacaktır. Nüfus temelli yaklaşımların kullanımı kolaydır, ancak bunların ana sınırlaması pareto optimallerine dayanmayan seçim şemasıdır. Pareto tabanlı yaklaşımlar ilk olarak Goldberg tarafından önerilmiştir [175]. Daha sonra çeşitliliği korumak ve yakınsamadan kaçınmak için niching ve ite paylaşımı önerilmiştir [176] Pareto tabanlı yaklaşımlar, iki kuşakta bölünmüş en popüler yaklaşımlardır. Etkinlik paylaşımı ile birinci nesil, pareto sıralaması ile birlikte niş, ikinci nesil elitizm nosyonu.

1.19. Domine Olmayan Genetic Algorithm (NSGA II)

Genetik algoritma, biyomedikal popülasyon modellemesinin bir sonucu olan en iyi bilinen problem çözme algoritmalarından biridir. Başka bir deyişle, bu algoritma, algoritmayı tahmin etmek veya uyarlamak için bir formül veya optimal bir cevap bulmak için Darwin'in doğal seçim idolojisini kullanır. Domine olmayan sıralama genetik algoritması (NSGA) genetik algoritmanın çok amaçlı modellerinden biridir. Aslında, problemdeki çok amaçlı varlığın varlığı, optimal bir çözümden ziyade bir dizi optimal çözümün (yaygın olarak pareto-optimal çözüm olarak bilinir) ortaya çıkmasına yol açar. Bir problem çözme için önerilen klasik optimizasyon yöntemi, birden fazla çözüm elde etmek için böyle bir yöntem seçildiğinde, en uygun zamanda pareto-optimal çözüme odaklanarak çok amaçlı optimizasyon problemini tek-objektif optimizasyon problemine dönüştürmektir. Her simülasyonun ardından farklı çözümler bulmayı umarak tekrar tekrar kullanılmalıdır. Geçtiğimiz yıllarda yapılan çalışmalarda (NSGA) gibi farklı çok amaçlı evrimsel algoritmalar önerilmiştir [151], [177], [178], [179]. N. Srinivas ve K. Deb tarafından önerilen NSGA, evrimsel algoritmaların ilk örneklerinden biri olarak düşünülebilir.

Baskın olmayan sınıflamanın yüksek hesaplama karmaşıklığı, elitizm eksikliği ve paylaşım parametrelerini belirtmek için gerekli olan temel (NSGA'S) eleştirilerdir. Bu çalışmada (NSGA) (NSGAII) adı verilen geliştirilmiş çeşiti kullanıyoruz. Genetik algoritmanın çok amaçlı modlarından biri olan (NSGAII) metodolojisinde aşağıdaki gibidir:

- İlk nüfusu yaratma.
- Uygunluk ölçütlerinin hesaplanması.
- Nüfusu Sıralama.
- Seçim: başlangıç popülasyonu, sürü koşullarına göre düzenlenir edilmez, sürü mesafesi hesaplanır ve ilk popülasyondan seçim başlar. Seçim iki parametreye dayanır:

İlk parametre popülasyon sıralamasıdır, popülasyonun daha düşük aşamalardan seçildiği açıklanmalıdır. İkinci parametre sürü mesafesidir. P ve q'nun bir rütbenin iki parametresi olduğunu varsayarsak. Daha büyük bir sürü mesafesine sahip bir üye seçilir. Seçim önceliğinin önce sıralama ile ve daha sonra sürüden uzaklığa bağlı olduğuna dikkat edilmelidir.

- Yeni çocuklar oluşturmak için genetik değişim ve Mutasyonun yapılması.
- Birincil popülasyon ile çaprazlama ve mutasyondan elde edilen popülasyonun kombinasyonu.
- Anne-baba nüfusunu, önceki seviyelerden birleştirilmiş nüfusun en iyi üyeleriyle değiştirmek. Bu adım böyle yapılmaktadır:

İlk adımda, daha düşük üyeler, önceki parametrelerle değiştirilir ve daha sonra sürülerden uzaklığa göre sıralanır. Sonraki adım. Tüm adımlar istenen optimal koşullara ulaşana kadar tekrarlanır. Şekil 38 Domine olmayan genetik algoritma sürecini gösterir.

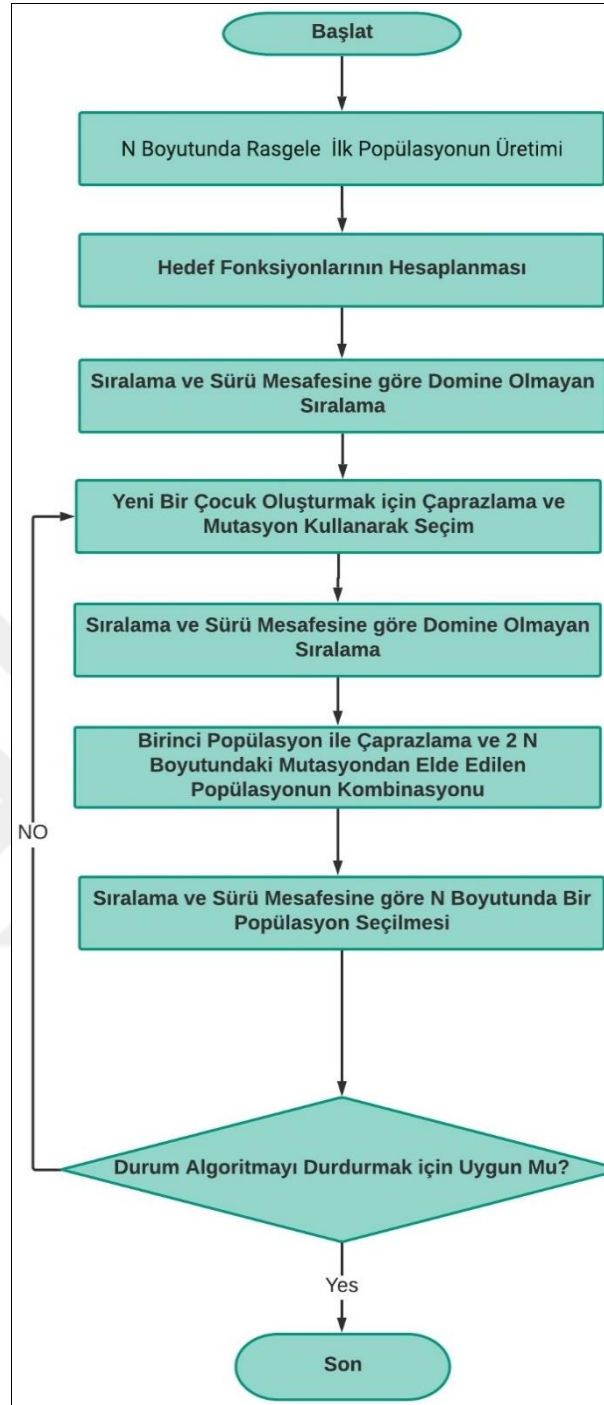
1.19.1. Algoritmasının Avantajları (NSGA II)

(NSGAII), (NSGA) algoritmalarının problemlerine ve karmaşıklıklarına sahip değildir, ayrıca aşağıdaki avantajlara sahiptir:

1. Çözüm sıralamadaki diğer yöntemlere kıyasla daha hızlıdır ve önceki algoritmaların hesaplama karmaşıklığı ortadan kaldırılmıştır. M'nin hedef fonksiyonların sayısı ve N'nin popülasyon büyüklüğü olduğu varsayıldığında, önceki algoritmaların hesaplama karmaşıklığı $O(MN^3)$ olmuştur [180]. Bu yöntemde bu $O(MN^2)$ 'dir [181].
2. NSGAII, diğer algoritmalar yerine daha düzgün ön cevap bulmak ve cevapların etrafındaki noktaların yoğunluk tahminini süreklilik mesafesi kullanır. Sürünün uzaklığının, bir cephedeki dağılım açısından en iyi çözümü seçmek için kullanılan bir faktör olduğu ve aşağıdaki gibi tanımlandığı unutulmamalıdır:
 - Bir cephenin başındaki ve sonundaki nokta için sonsuz olduğu varsayılır.
 - Diğer noktalar için 2'den K-1'e aşağıdaki ilişki olarak tanımlanır:

$$CD[i] = \frac{(f_m^{i+1} - f_m^{i-1})}{(f_m^{i+1} - f_m^{i-1})} \quad (64)$$

Burada CD [i], öndeki (F) bireyin (i) sürü mesafesidir, f_m^i , öndeki (F) bireydeki (i) m'deki hedef fonksiyonun değeridir ve f_m^{min} , f_m^{max} sırasıyla ön (m) hedef fonksiyonunun en düşük ve maksimum değeridir. Daha iyi bir cevap, daha büyük bir mesafeye sahip olan cevaptır [182].



Şekil 38. NSGA II algoritmasının akış şeması

1.20. Parçacık Sürüsü Optimizasyonu ve Evrimsel Algoritmalar

PSO, ebeveyn temsilinde, bireylerin seçilmesinde ve parametre ayarlamasında [183] 'de gösterildiği gibi farklılıklar açısından EA'lardan farklıdır:

- PSO'da ebeveyn bilgileri, Evrimsel Optimizasyon (EO) 'da paylaşılrken her parçacık içinde bulunur.
- PSO, işleyişinden EO'nun yaptığı açık bir seçim işlevi içermez.
- PSO, EO'da çok yönlü iken bireyleri manipüle etmek için oldukça yönlü bir mutasyon işlemi kullanır.
- PSO'nun hız adım boyutunu yerel bölge arama alanına uygun bir değere uyarlaması için bir mekanizma yoktur, oysa EO her bireyin bileşeni için mutasyon şiddetini içerir.

Farklı yöntemler kullanan farklı çözümler, farklı hedefler arasında uygun senaryolar üretebilir. Bir amaca göre en uygun çözüm, diğer amaçlar için bir uzlaşma gerektirir. Bu, kullanıcının yalnızca bir hedefe göre en uygun çözümü seçmesini vurgular [184]. MOO'nun temel amacı, en uygun çözümlere yakın ve en uygun çözümlerin gerçek yayılımını temsil edecek kadar çeşitli olan bir dizi çözüm bulmaktır. MOPSO algoritmaları, yukarıda belirtilen her iki koşulu da doğrudan doğrular. MOPSO'nun sadeliği, düşük hesaplama maliyeti ve artan popülerliği, basit ve karmaşık huylu gerçek yaşam problemlerini çözme yeterliliğini artırır.

1.21. Çok Amaçlı Parçacık Sürüsü Optimizasyonu (MOPSO)

Sezgisel yöntem tabanlı çok amaçlı optimizasyon tekniklerinin kullanımı ve geliştirilmesi son birkaç yılda önemli ölçüde artmıştır [185]. Son araştırma eğilimlerinden biri, daha verimli algoritmalar üretmek olmuştur. Verimlilik için bu amaç normalde çeşitliliği korumak için akıllı teknikler (PAES [186] tarafından kullanılan uyarlanabilir ızgara gibi) ve küçük nüfus boyutlarının [187] kullanımını gerektirir. Parçacık sürü optimizasyonu (PSO), bir kuş sürüsünün koreografisinden esinlenilen nispeten yeni bir buluşsal yöntemdir. Çeşitli optimizasyon görevlerindeki mevcut başarısına rağmen, PSO, şimdiye kadar çok amaçlı optimizasyon üzerinde çok fazla çalışma yapılmayan buluşsal yöntemlerden biri olmaya devam etmektedir.

1.21.1. Çok Amaçlı Parçacık Sürüsü Optimizasyonunun Gelişimi

Bu bölümde, PSO algoritmasının çok amaçlı optimizasyon problemleriyle başa çıkmasını sağlayan “çok amaçlı partikül sürüsü optimizasyonu” (MOPSO) adı verilen popüler algoritmalarından bahsetmekteyiz. Yaklaşımın uygulanması nispeten basittir, nüfusa dayalıdır, çeşitliliği korumak için harici bir bellek (“depo” olarak adlandırılır) ve coğrafi olarak dayalı bir yaklaşım kullanır. MOPSO, özel literatürde rapor edilen ve oldukça rekabetçi iki EM0 algoritması ile karşılaştırılan bazı standart test fonksiyonları kullanılarak doğrulanmıştır: Pareto Arşivlenmiş Evrim Stratejisi [si (PAES) ve hâkim Olmayan Sıralama Genetik Algoritması II [188] (NSGA II).

Parçacık sürüsü optimizasyonunun evrimsel algoritmalarla analogisi, bir Pareto sıralama şeması [sı'nın çok amaçlı optimizasyon problemlerini ele alma yaklaşımını genişletmenin kolay yolu olabileceği fikrini ortaya koymaktadır. Bir parçacık (Le., Bir birey) tarafından bulunan en iyi çözümlerin tarihsel kaydı, geçmişte üretilen belirsiz ilaçları saklamak için kullanılabilir (bu, evrimsel çok amaçlı optimizasyonda kullanılan elitizm kavramına benzer olacaktır). Küresel çekim mekanizmalarının kullanımı, daha önce bulunmamış, bulunmamış vektörlerin tarihsel bir arşivi ile birleştirildiğinde, küresel olarak belirlenmemiş çözümlere yakınsamayı motive edecektir.

Bu nedenle, MOPSO, her parçacığın her uçuş döngüsünden sonra uçuş deneyimlerini yatıracağı küresel bir depoya sahip olma fikrine dayanır. Ayrıca, depodaki güncellemeler, her bireyin objektif işlev değerleri açısından tanımlanan coğrafi tabanlı bir sistem dikkate alınarak gerçekleştirilir. Bu teknik, Pareto Arşiv Evrim Stratejisi (PAES) ile kullanılan harici dosyadan esinlenmiştir. Daha önce bahsedilen depo, parçacıklar tarafından aramaya rehberlik edecek bir lideri tanımlamak için kullanılır. Her parçacığın farklı bir kılavuz seçebileceği bir mekanizma uyguladık. Mekanizmamız, araştırılan arama alanını bölerek üretilen hiper küplerin üretimine dayanmaktadır. MOPSO'nun akış şeması Figure. 'da sunulmuştur.

MOPSO algoritması şu şekildedir: (65)

1. Popülasyon POP'u başlat:

(a) $i = 0$ dan MAX 'a / MAX = parçacık sayısı.

(b) POP[i] başlat.

2. Her parçacığın hızını başlatın:
 - (a) i için $= 0$ dan Max 'a
 - (b) $Vel[i] = 0$
3. POP'daki parçacıkların her birini değerlendirin.
4. Depolanmamış vektörleri temsil eden parçacıkların konumlarını depo REP'sinde saklayın.
5. Şimdiye kadar keşfedilen arama alanının hiper küplerini oluşturun ve bu hiper küpleri her bir parçacığın koordinatlarının objektif işlevlerinin değerlerine göre tanımlandığı bir koordinat sistemi olarak kullanarak parçacıkları bulun.
6. Her parçacığın belleğini başlatın (bu bellek, arama alanında gezinmek için bir kılavuz görevi görür. Bu bellek ayrıca depoda saklanır):
 - (a) i için $= 0$ dan Max 'a (66)
 - (b) $PBESTS[i] = POP[i]$
7. Maksimum döngü sayısına DO ulaşılmadığında:

Aşağıdaki ifadeyi kullanarak her parçacığın hızını hesaplayın:

$$(a) \quad VEL[i] = W \times VEL[i] + R_1 \times (PBESTS[i] - POP[i]) + R_2 \times (REP[h] - POP[i]) \quad (67)$$

Burada W (atalet ağırlığı) 0.4 değerini alır; R_1 ve R_2 , $[0..1]$ aralığında rasgele sayılardır; $PBESTS[i]$, i parçacığının 2'inci sahip olduğu en iyi konumdur; $REP[h]$, depodan alınan bir değerdir; h indeksi şu şekilde seçilir: birden fazla parçacık içeren hiperküplere, herhangi bir $z > 1$ bölümünün sayısına bölünmesi sonucuna eşit bir uygunluk atanır. Bu, daha fazla parçacık içeren hiperküplerin uygunluğunu azaltmayı amaçlamaktadır ve bir fitness paylaşımı biçimi olarak görülebilir [189]. Ardından, karşılık gelen parçacığı alacağımız hiperküpu seçmek için bu durum değerlerini kullanarak rulet tekerleği seçimini uyguluyoruz. Hiper küp seçildikten sonra, bu hiperküp içinde rastgele bir parçacık seçeriz. $POP[i]$, i parçacığının geçerli değeridir.

- (b) Önceki adımdan üretilen hızı ekleyerek parçacıkların yeni konumlarını hesaplayın:

$$POP[i] = POP[i] + VEL[i] \quad (68)$$

- (c) Sınırlarının ötesine geçmeleri durumunda arama alanı içindeki parçacıkları koruyun (geçerli arama alanında bulunmayan çözümler üretmekten kaçının).
- (d) POP'daki parçacıkların her birini değerlendirin.
- (e) Hiper küpler içindeki parçacıkların coğrafi gösterimi ile birlikte REP içeriğini güncelleyin. Bu güncelleştirme, o anda belirtilmemiş olan tüm konumları depoya eklemekten ibarettir. Depodan domine edilen yerler süreçte elimine edilir. Deponun boyutu sınırlı olduğu için, dolduğu zaman, saklama için ikincil bir kriter uyguluyoruz: daha az nüfuslu objektif alan yerlerinde bulunan bu parçacıklara yüksek nüfuslu bölgelerde yatanlara öncelik verilir.
- (f) Parçacığın mevcut konumu, belleğindeki konumdan daha iyi olduğunda, parçacığın konumu aşağıdakiler kullanılarak güncellenir:

$$PBESTS[i] = POP[i]$$

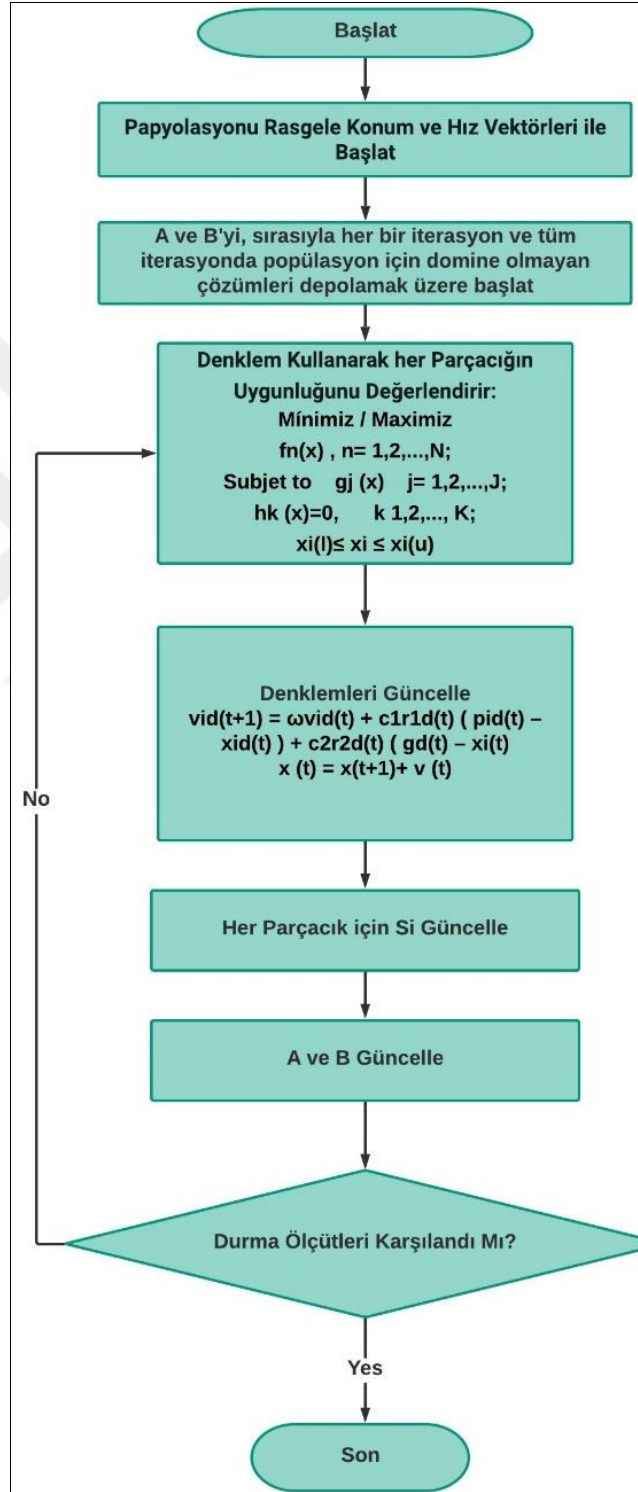
Bellekten hangi pozisyonun korunması gerektiğine karar verme kriteri, basitçe Pareto hâkimiyetini uygulamaktır (yani, mevcut konuma bellekteki konumun baskın olması durumunda, bellekteki konum korunur; aksi takdirde, mevcut konum bellekteki konumun yerini alır Bunlardan hiçbirisi diğerine baskın değilse, birini rastgele seçeriz).
- (g) Döngü sayacını artırma

8. Sonlandır

1.21.2. MOPSO'nun En Çok Uygulandığı Alanlar

- Uzay mühendisliği [190].
- Biyolojik bilimler [191].
- Kimya mühendisliği [192].
- İnşaat mühendisliği [193].
- Veri madenciliği [194].
- Elektrik mühendisliği [195].
- Elektromanyetik mühendisliği [196].
- Çevre bilimleri [197].

- Nöral ağlar [198].
- Robotik [199].
- Yazılım mühendisliği [200].



Şekil 39. MOPSO algoritmasının akış şeması

2. YAPILAN ÇALIŞMALAR

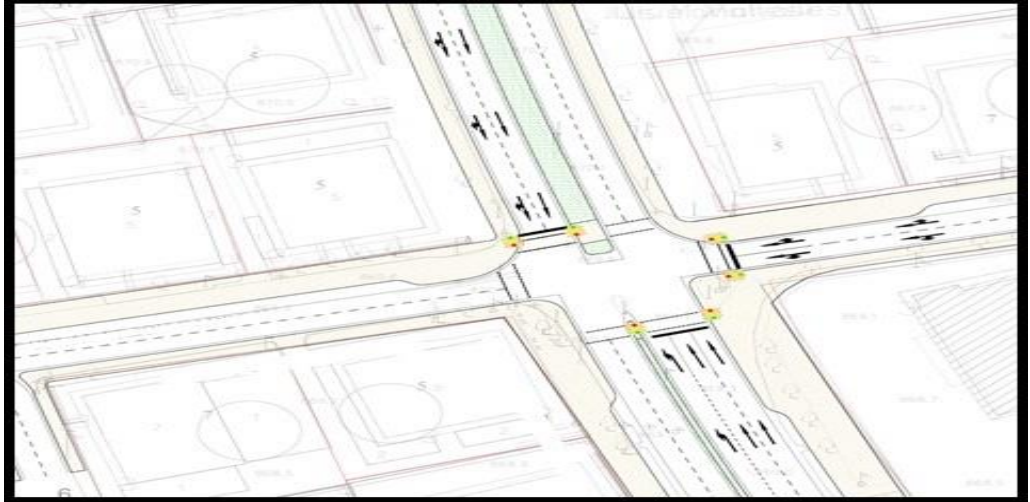
2.1. Çalışma Adımları ve Veri Toplama Yöntemi

Bu tez çalışmasındaki veriler, video kameralar vasıtasıyla Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul üzerinde dört kollu ve üç fazlı kavşaktan toplanmış ve Excel ortamında kullanabilecek veri formatına dönüştürülmüştür. Kavşağa ait hacim sayımı tamamlandıktan sonra zirve saat standardı elde edilmiştir. Sabah, öğlen ve akşam için belirli saatlerde yüksek araç hacmi (aşırı doymun durumlar) zirve saatin ne zaman gerçekleştiğini göstermektedir. Bu çalışmada, ANFİS mimarisine uygun, kaliteli ve hatasız girdiler elde edebilmek için VISSİM simülasyon modeli kullanılmıştır. VISSİM trafik simülasyon ortamında kavşağın modellenmesi için trafik hacim verileri ve diğer bağımlı parametreler girilmiş, programın tasarım özelliği kullanılarak optimum yeşil süre, kuyruk uzunluğu ve gecikme değerleri elde edilmiştir. Tasarım sonrası elde edilen değerlerde iyileşmeler olduğu, kavşaktaki doymunluk oranının kritik değerlerin altında kaldığı gözlenmiştir. Tasarladığımız ANFİS-Type1 mimarisi üç girdi ve bir çıktıdan oluşmaktadır. ANFİS, sistemin kurallarını uyarlamalı olarak öğrenen ve değiştiren birinci sıra Sugeno-FIS'e dayanmaktadır. ANFİS eğitimi, bir optimizasyon algoritması kullanarak öncül ve sonuç parametrelerinin belirlenmesi anlamına gelmektedir. ANFİS ile başarılı sonuçlara ulaşmak doğrudan eğitildiği optimizasyon algoritmalarına bağlıdır. Bu çalışmada eğitim süresinde parametreleri ayarlamak ve hataları minimize edebilmek için Tek-amaçlı (GA, PSO, HS) ve Çok-amaçlı (MOPSO, NSGAIİ) algoritmalar kullanılmıştır. Tek-amaçlı algoritmalar ile yapılan simülasyon sonuçları incelendiğinde, ANFİS-GA'nın ANFİS-PSO ve ANFİS-HS'den daha iyi tahminler gösterdiği gözlenmiştir. Çok-amaçlı algoritmalar ile yapılan simülasyon sonuçları incelendiğinde, ANFİS-MOPSO'nun ANFİS- NSGAIİ 'dan daha iyi tahminler gösterdiği gözlenmiştir.

2.2. Test Ortamı

İstanbul şehri, Karadeniz ve Marmara Denizi arasındaki kara parçaları üzerinde yer almaktadır (Geymen, 2013). Türkiye'nin en büyük şehridir ve 20. yüzyılın ikinci yarısında

hızlı sanayileşme ve kentleşme yaşamıştır. 1950'lerden bu yana İstanbul, kırsal alanlardan kentsel alanlara büyük ölçekli göç akışının ana hedefi olmuştur. Bu süre zarfında İstanbul, hızlı sanayileşme ve plansız kentleşme yaşamış ve çok sayıda ciddi sorunla karşı karşıya kalmıştır. Bunlar arasında artan yoğunluk, tıkanıklık, trafik sıkışıklığı, kirlilik ve kentsel arazi kıtlığı en önemlileridir (Uzun, 2010). Türkiye İstatistik Kurumunun (TÜİK) en son 2019 yılında açıkladığı gelir ve yaşam koşulları anketine göre İstanbul, Türkiye'de gelir eşitsizliğinin en yüksek olduğu bölgedir. 2019 yılında Türkiye nüfusu ise 83.154.997 kişi, İstanbul ise 15.519.267 kişi olarak açıklanmıştır. İstanbulun nüfus yoğunluğu 2921/km² dir kilometrekareye 2921 insan düşmektedir). İstanbul'da 23 milyon 170 bin 769 adet motorlu kara taşıtı bulunmaktadır. Son nüfus rakamları dikkate alındığında Türkiye'deki 21 ilin nüfusu toplamının, İstanbul'daki motorlu kara taşıtları sayısına denk geldiği anlaşılr. Şehir uluslararası, ticari, siyasi, akademik, sosyal, sanatsal ve kültürel anlamda her yıl çok büyük miktarda turist ağırlamaktadır. İstanbul 2019 yılında toplam 16 milyon yabancı ziyaretçiye ev sahipliği yapmıştır. İstanbul'da bulunan dünyanın en büyük havalimanı, İstanbul Avrupa yakasında 76,5 milyon metrekaare civarında bir alanda yer almaktadır. 200 milyonluk yolcu kapasitesi ve 350'den fazla uçuş noktası ile birlikte dünya listelerinde çok büyük avantaja sahiptir. Sabiha Gökçen Havalimanı İstanbul'daki ikinci havalimanıdır. 2001 yılında, Türkiye'nin ilk kadın savaş pilotu olan Sabiha Gökçen'in ismini alarak açılmıştır. İstanbul'un Anadolu yakasında, Pendik-Kurtköy sınırları içerisinde yer almaktadır. Yıllık 45 milyon yolcu kapasitesine sahiptir. Dolayısıyla bahsedilen bütün faaliyetler, şehrin ne denli bir trafik yüküne maruz kaldığını görme noktasında yeterlidir.



Şekil 40. Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul (çalışma alanı)

2.3. Mikroskobik Simülasyon (VISSİM)

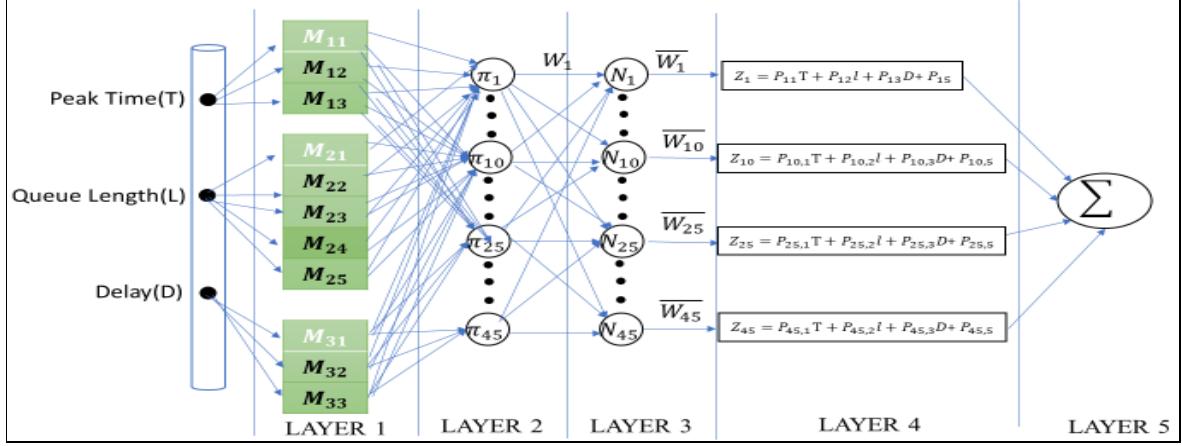
Bu tez çalışmasındaki veriler, video kameralar vasıtasıyla Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul üzerinde dört kollu ve üç fazlı kavşaktan toplanmış ve Excel ortamında kullanabilecek veri formatına dönüştürülmüştür, şekil 40'da gösterildiği gibi çalışma alanı olarak uygun görülmüştür. Her şeritteki araçlar, yeşil zamana ve fazın yönüne bağlı olarak kavşaktan geçme yetkisine sahiptir. Simülasyonumuzda dört bölge ele alınmıştır. Video kameralardan aldığımız trafik verileri kullanılarak zirve saatler belirlenmiştir. Her yön için kuyruk uzunluğu, gecikme ve yeşil sinyal süresi hesaplanmış ve kavşağın mevcut durumu gözlenmiştir. Gün boyunca üç ayrı zirve saati, sabah (08.00-09.00), öğlen (12.00-13.00) ve akşam (16.00-17.00) olarak ayrılmıştır.



Şekil 41. Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul (çalışma alanı)

2.4. ANFİS Tasarımı

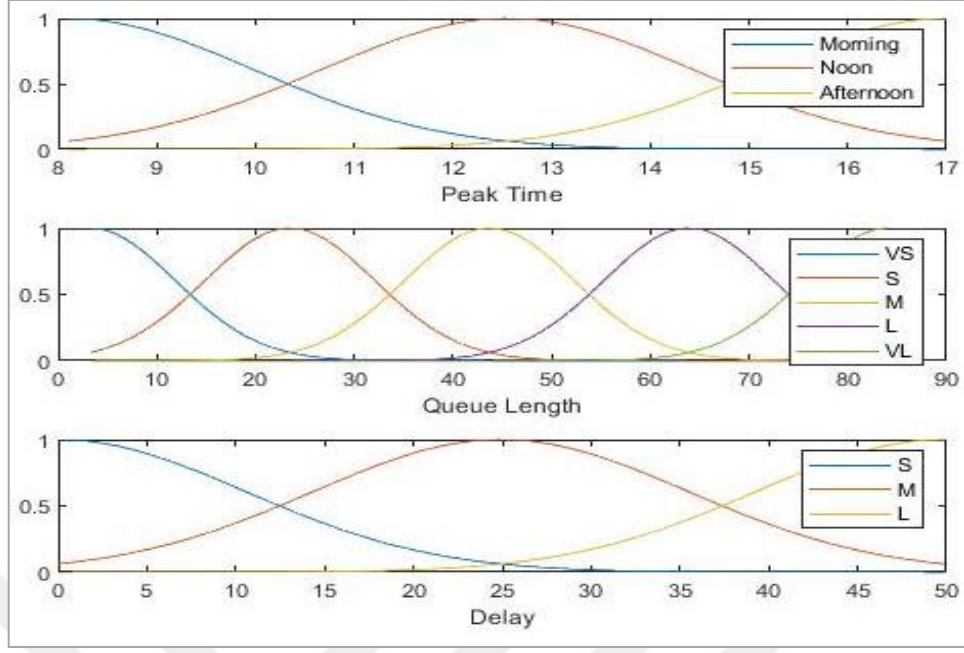
ANFİS mimarisinde hem sinir ağı NN hem de bulanık mantık FL kullanılır. ANFİS'in ağ yapısı, öncül ve sonuç kısımları olarak adlandırılan iki bölümden oluşur. ANFİS, eğitim sırasında mevcut giriş-çıkış veri çiftlerini kullanır. Daha sonra bu parçaların birbirine bağlı olduğu IF-THEN bulanık kuralları elde edilir. Bu çalışmada tasarladığımız ANFİS yapısı Şekilde 42'de görüldüğü gibi beş katmandan oluşmaktadır.



Şekil 42. Üç girişi ve bir çıkışı olan ANFIS yapısı

2.5. Üyelik İşlevlerinin Tasarımı

Şekil 42’de 3 girişe sahip ANFIS yapısı görülmektedir; ilk girdi (zirve zamanı) için sabah, öğlen ve akşam olarak üç üyelik fonksiyonu ele alınmıştır, ikinci giriş (kuyruk uzunluğu) için beş üyelik fonksiyonu VS (çok küçük), S (küçük), M (orta), L (büyük) ve VL (çok büyük) olarak, son girdi (gecikme) için üç üyelik fonksiyonu S (küçük), M (orta) ve L (büyük) olarak değerlendirilmiştir. Üyelik işlevlerinin her biri ile ilişkili iki üyelik işlevi (alt sınır, üst sınır) mevcuttur. İki aralıkta 11 üyelik fonksiyonu göz önüne alındığında, birinci katmanda toplam 22 karar değişkenine sahip olmaktadır (veya kromozom, vektör olarak adlandırılır). Her bir giriş üyelik fonksiyonunun aralığı, Şekilde 43’de gösterilmiştir. Üyelik işlevlerini birbiriyle çarparak ($3 * 5 * 3 = 45$) doğrusal denklemi simgeleyen toplam 45 kuralımız vardır. Her doğrusal denklem 4 parametreden oluşmaktadır. Dördüncü katmanda toplam 180 bireysel (kromozom, vektör, karar değişkeni) karar verilmiştir ($4 * 45$). Bu yapıda metasezgisel algoritmalar tarafından ayarlanan toplam 202 karar değişken parametresine sahip olunmuştur.



Şekil 43. Tasarlanmış üyelik fonksiyonları

2.6. Tek-Amaçlı Algoritmalar İçin Kullanılan Hata Fonksionuları-Maliyet Fonksiyonları

Mutlak ve bağlı hata, her deneysel bilimcinin aşına olması gereken iki hata türüdür. Aralarındaki Farklılık önemlidir. Mutlak hata (Δ_x), bir ölçümdeki fiziksel hatanın miktarıdır. Bağlı hata, bir ölçümün ölçülen şeyin boyutuna göre ne kadar iyi olduğuna dair bir gösterge verir.

$$\text{Relative Error: } \frac{\Delta x}{x} \quad (69)$$

Burda Δ_x mutlak hata ve x ise herhangi bir değişkendir.

Ağın ağırlıkları, hata fonksiyonu en aza indirecek şekilde değiştirilir. SD (σ) bir değerler kümesinin değişim veya dağılım miktarının bir ölçüsüdür. Düşük bir standart sapma, değerlerin kümenin ortalamasına (beklenen değer olarak da bilinir) yakın olma eğiliminde olduğunu gösterirken, yüksek bir standart sapma, değerlerin daha geniş bir alana yayıldığını gösterir:

$$SD = \sqrt{\frac{\sum (x - \bar{x})^2}{n-1}} \quad (70)$$

Yukardaki fonksiyonda yer alan $\sum$ ekleme anlamına gelir, x bireysel gözlemdir, $\bar{x}$ ortalama miktarıdır ve n gözlem sayısıdır.

MSE (ortalama kare hata maliyet fonksiyonu) ve RMSE (kök ortalama kare hatası)

Olarak yaygın bir şekilde minimizasyon problemlerinin çözümünde kullanılmaktadırlar:

$$MSE = \frac{1}{N} \sum_i^n (f_i - y_i)^2 \quad (71)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i^n (f_i - y_i)^2} \quad (72)$$

Yukardaki hata fonksiyonunda yer alan N veri noktası sayısıdır, f_i model tarafından dödürülen değerdir ve y_i i 'inci veri noktası için gerçek değerdir.

2.7. Çok-Amaçlı Algoritmalar İçin Kullanılan Hata Fonksionuları

Mutlak ve bağlı hata, her deneysel bilimcinin aşına olması gereken iki hata türüdür. aralarındaki farklılık önemlidir. Mutlak hata (Δ_x), bir ölçümdeki fiziksel hatanın miktarı ya da dönemidir. Bağlı hata, bir ölçümün ölçülen şeyin boyutuna göre ne kadar iyi olduğuna dair bir gösterge vermektedir.

$$\text{Relative Error: } \frac{\Delta x}{x} \quad (73)$$

Burda Δ_x mutlak hata ve x ise herhangi bir değişkendir.

Ağın ağırlıkları, hata fonksiyonu en aza indirecek şekilde değiştirilir. SD (σ) bir değerler kümesinin değişim veya dağılım miktarının bir ölçüsüdür. Düşük bir standart sapma, değerlerin kümenin ortalamasına (beklenen değer olarak da bilinir) yakın olma eğiliminde olduğunu gösterirken, yüksek bir standart sapma, değerlerin daha geniş bir alana yayıldığını gösterir:

$$SD = \sqrt{\frac{\sum (x - \bar{x})^2}{n-1}} \quad (74)$$

Yukardaki fonksiyonda yer alan $\sum$ ekleme anlamına gelir, x bireysel gözlemdir, $\bar{x}$ ortalama miktarıdır ve n gözlem sayısıdır.

MSE (ortalama kare hata maliyet fonksiyonu) ve RMSE (kök ortalama kare hatası) olarak yaygın biçimde minimizasyon problemlerinin çözümünde kullanılmaktadırlar:

- $F1 \text{ MSE} = \frac{1}{N} \sum_i^n (f_i - y_i)^2 \quad (75)$

- $F2. \text{RMSE} = \sqrt{\frac{1}{N} \sum_i^n (f_i - y_i)^2} \quad (76)$

Yukardaki hata fonksiyonunda yer alan N veri noktası sayısıdır, f_i model tarafından dödürülen değerdir ve y_i i 'nci veri noktası için gerçek değerdir.

F3. Toplam kare hatası, ölçülen verilerin gerçek ortalamadan sapma ölçüsüdür. Hataların toplamı ortalama olarak sıfırdır, çünkü hatalar eşit derecede pozitif veya negatif olabilir. Bu, doğru olmayan hiçbir hata olmadığı anlamına gelmektedir.

- $F3 \text{ SSE} = \sum (Y_i - Y)^2 \quad (77)$

Burada hata fonksiyonunda yer alan SSE, toplam kare hatası ve $\sum$ tüm veri noktalarının toplamı, Y_i ise her bir veri noktası ve Y ise ortalama değerini temsil etmektedirler.

Bu tez boyunca hata fonksiyonlarımızı $F1$, $F2$ ve $F3$ ile göstermekteyiz.

3. BULGULAR

3.1. VISSİM ve Verilerin Girişi

İstanbul, ulaşım sistemlerinin yoğun trafik saatlerinde daha karmaşık hale geldiği ve trafik kontrol merkezleri için önemli zorluklara neden olduğu Türkiye'nin önemli bir metropol şehridir. Bu nedenle geleneksel düşük performanslı trafik kontrol yöntemlerini kullanmak yerine yoğun saatlerde kentsel ağ tıkanıklığını çözmek için akıllı ulaşım kontrol sistemlerinin kullanmasında fayda vardır. Trafik ışığı planı için iyi planlanmış ayarlar, ağ kapasitesini artırmak ve doymamış durumlarda performansını artırmak için çok önemli bir çözümdür. VISSİM trafik simülasyon ortamında Şekil.44'de gösterildiği gibi dört kollu ve üç fazlı bir kavşağa uygulanmıştır. Her şeritteki araçlar, yeşil zamana ve fazın yönüne bağlı olarak kavşaktan geçme yetkisine sahiptir. Simülasyonumuz dört bölgede ele alınmıştır. Modelleme ve trafik hacim verileri ve diğer bağımlı parametreleri girme ile başlamaktadır. Öngörücünün dinamiği için, geçmiş veri kümesinden öğrenmek ve geleceği tahmin etmek için ANFIS'in avantajı kullanılmıştır.

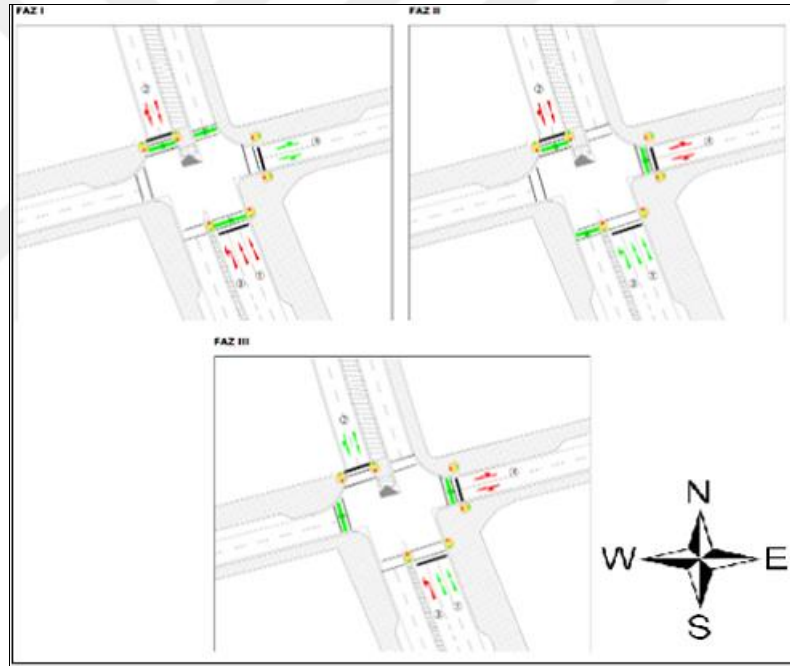


Şekil 44. Test alanı

3.2. Kavşağın VISSİM Ortamında Tasarlanması

Hacim sayımı tamamlandıktan sonra zirve saat faktörünü elde ettik. Zirve saat sabah 08.00-09.00), öğlen (12.00-13.00) ve akşam (16.00-17.00) olmak üzere günün üç farklı

saatlerinde dikkate alınmıştır. İlk adımda, yaklaşan üç bağlantıda optimum kuyruk uzunluğuna dair kesin bir değerlendirme yapılmıştır. İkinci adımda, trafik ışık ayarları bağlantıların girişinde optimize edilmiştir. Bu optimizasyon, bağlantılardan geçen trafik akışını en üst düzeye çıkarmaya ve gecikmeyi en aza indirmeye ve kuyruk uzunluğunu optimum değerine yakın tutmaya ve döngüler aracılığıyla minimum salınımlarla optimum yeşil zaman elde etmeye dayanarak yapılmıştır. Çalışma alanımızdan toplanan veri kümesini (video kameralar vasıtasıyla) kullanarak her yön için kuyruk uzunluğu, gecikme ve yeşil sinyal süresi hesaplanmış ve kavşağın mevcut durumu gözlenmiştir. VISSİM’de tasarladığımız modelin performansını değerlendirmek için ağıın mevcut durumu ile karşılaştırılmış, zirve saatlerinde kuyruk uzunluğu ve gecikme açısından iyileşmeler olduğu gözlemlenmiştir.

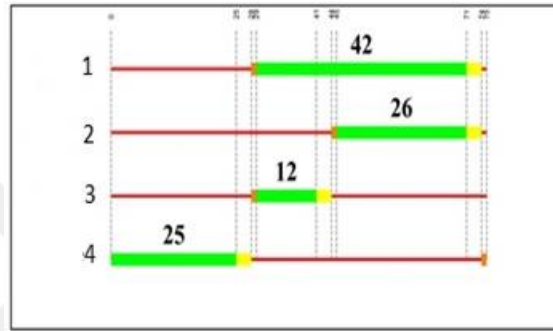


Şekil 45. VISSİM'de tasarlanmış platform.

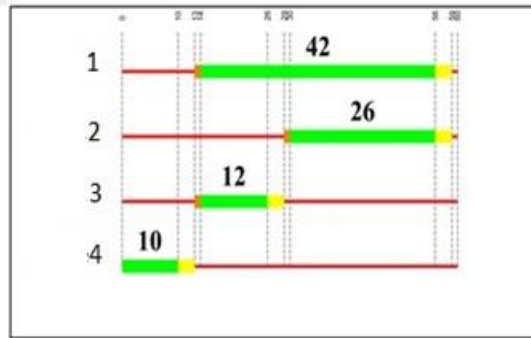
3.3. Tasarım Sonrası Optimum Yeşil Süreyi Kullanarak Faz kuruluşu

Simülasyon öncesi ve sonrası kavşağın performansı karşılaştırılmıştır. Simülasyon sonuçları, simülasyonda optimum yeşil sinyal süresini kullanarak kuyruk uzunluğu ve gecikme açısından iyileşme göstermektedir. Bu çalışmada simülasyon VISSİM’de modellenmiştir. Optimum döngü süresi sabah 75S, öğlen 60S ve akşam 75S olarak

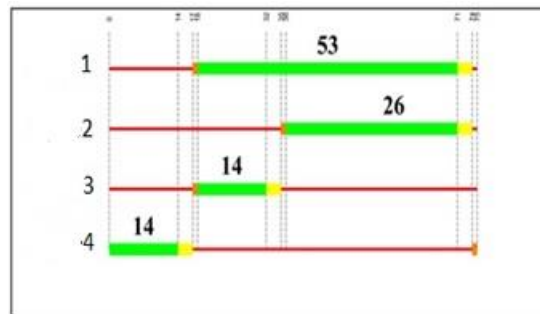
hesaplanmıştır. Her faz için tasarlanan zaman Şekil 46, 47 ve 48'de sunuldu. Ayrıca 1 saatlik simülasyonda simüle edilen modelin ayrı ayrı 3 zirve saatinde (08:00-09:00, 12:00-13:00 ve 16:00-17:00) kuyruk uzunluğu ve gecikme faktörlerini her yön için ve nihayetden toplam yönler için karşılaştırılmıştır, tasarlanan modelin daha iyi performans gösterdiği tespit edilmiştir. Kuyruk uzunluğu ve gecikme karşılaştırmaları Şekil 49-60'da sunulmuştur. Tasarım sonrası Seçilen kavşaktaki doygunluk oranı kritik değerinin altında kaldığı ve maksimum çıkış elde edildiği görülmüştür.



Şekil 46. Sabah için optimum devre süresi



Şekil 47. Öğlen için optimum devre süresi

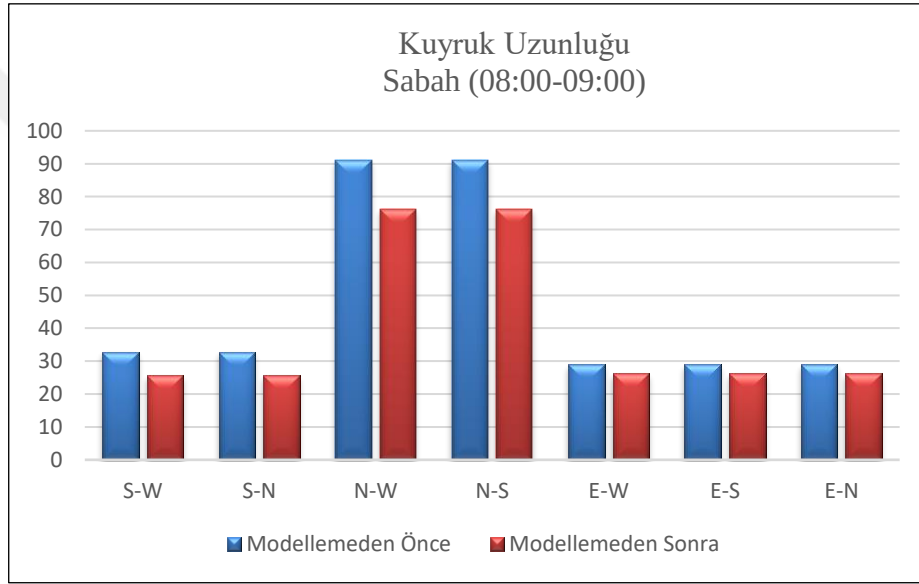


Şekil 48. Akşam için optimum devre süresi

3.4. Sabah 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme

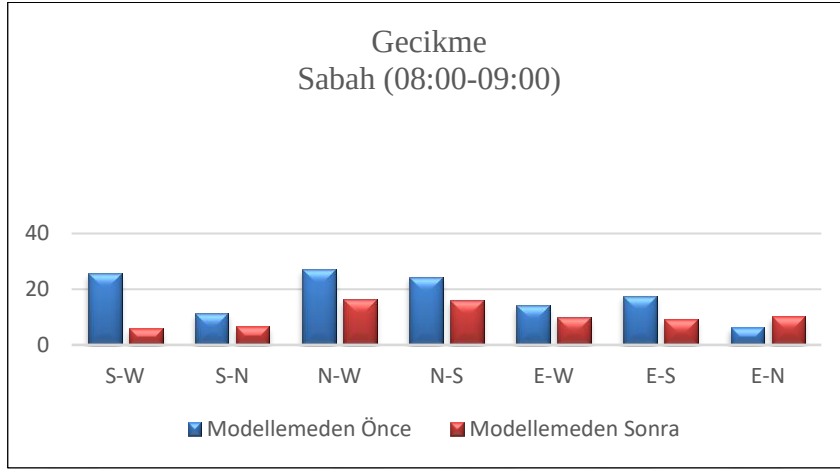
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (75 S) kullanarak modellemeden önce ve modellemeden sonra kavşağımızda yönlerin hepsinde (S-W, S-N, N-W, N-S, E-W, E-S ve E-N) kuyruk uzunluğu ve gecikme göz önüne aldığımızda ikisinde de iyileşmeler olduğu gözlemlenmiştir.

3.4.1. Sabah Her Yön İçin Kuyruk Uzunluğu Karşılaştırma sonuçları.



Şekil 49. Sabah modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması

3.4.2. Sabah Her Yön İçin Gecikme Sonuçları

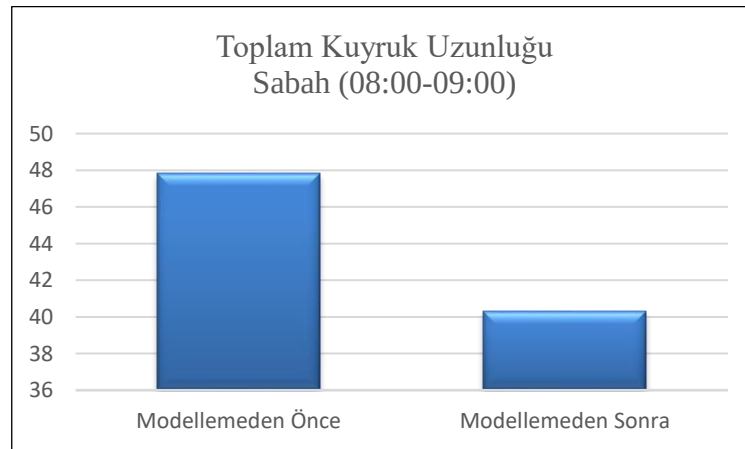


Şekil 50. Sabah modelleme öncesi ve sonrası gecikmenin karşılaştırılması

3.5. Sabah 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme

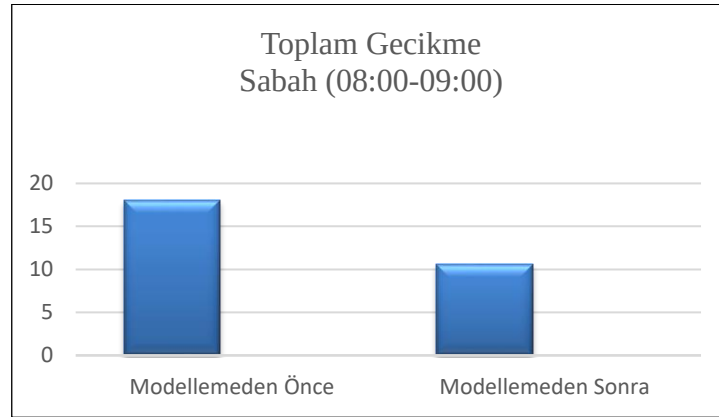
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (75 S) kullanarak modellemeden önce ve modellemeden sonra 1 saat içinde kavşağımızda Toplam kuyruk uzunluğu ve toplam gecikme göz önüne alındığında ikisinde de iyileşmeler olduğu gözlemlenmiştir.

3.5.1. Sabah Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları



Şekil 51. Sabah modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması

3.5.2. Sabah Toplam Gecikme Karşılaştırma Sonuçları

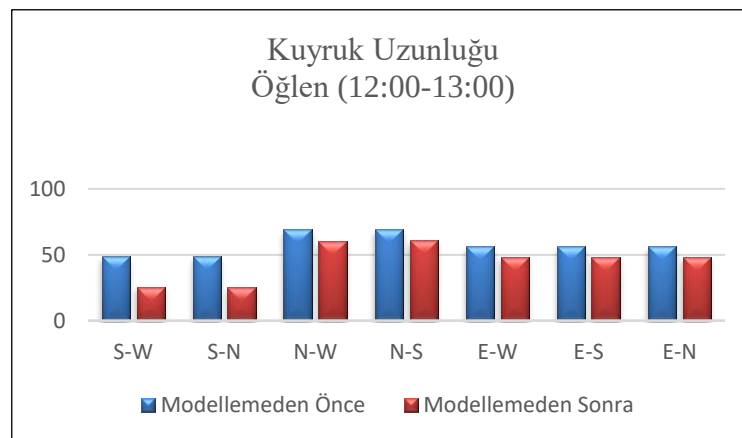


Şekil 52. Sabah modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması

3.6. Öğlen 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme

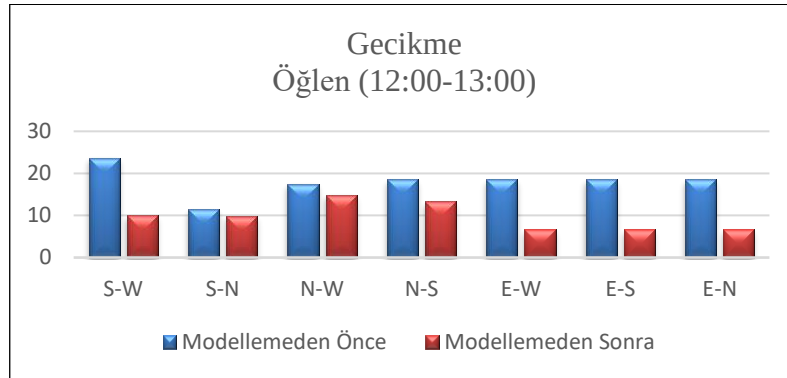
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (60 S) kullanarak modellemeden önce ve modellemeden sonra kavşağımızda yönlerin hepsinde (S-W, S-N, N-W, N-S, E-W, E-S ve E-N) kuyruk uzunluğu ve gecikme göz önüne alındığında ikisinde de iyileşmeler olduğunu gözlemlenmiştir.

3.6.1. Öğlen Her Yön İçin Kuyruk Uzunluğu Karşılaştırma Sonuçları



Şekil 53. Öğlen modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması

3.6.2. Öğlen Her Yön İçin Gecikme Karşılaştırma Sonuçları

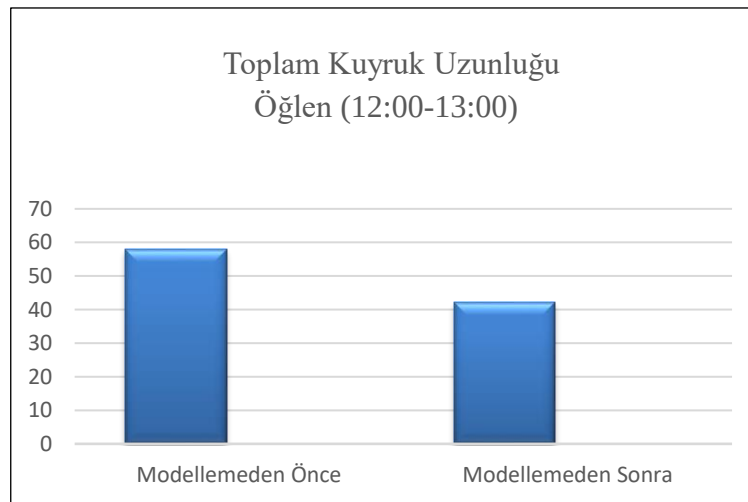


Şekil 54. Öğlen modelleme öncesi ve sonrası gecikmenin karşılaştırılması

3.7. Öğlen 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme

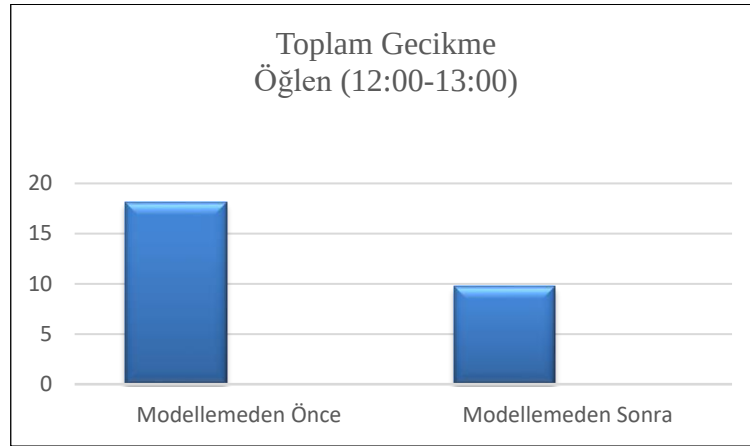
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (60 S) kullanarak modellemeden önce ve modellemeden sonra 1 saat içinde kavşağımızda Toplam kuyruk uzunluğu ve toplam gecikme göz önüne alındığında ikisinde de iyileşmeler olduğu gözlemlenmiştir.

3.7.1. Öğlen Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları



Şekil 55. Öğlen modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması

3.7.2. Öğlen Toplam Gecikme Karşılaştırma Sonuçları

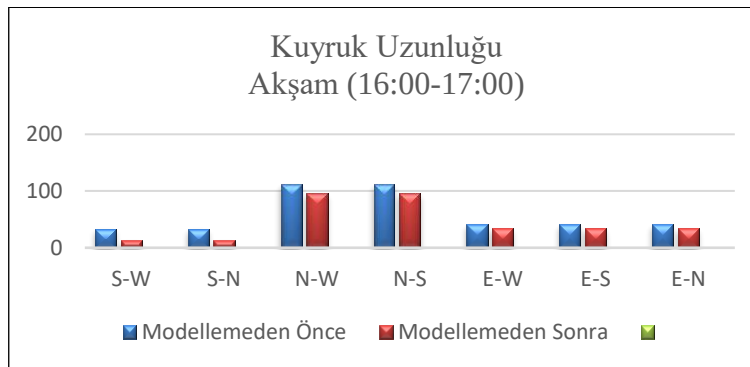


Şekil 56. Öğlen modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması

3.8. Akşam 1 Saat İçinde Kavşağın Her yönü İçin Kuyruk Uzunluğu ve Gecikme

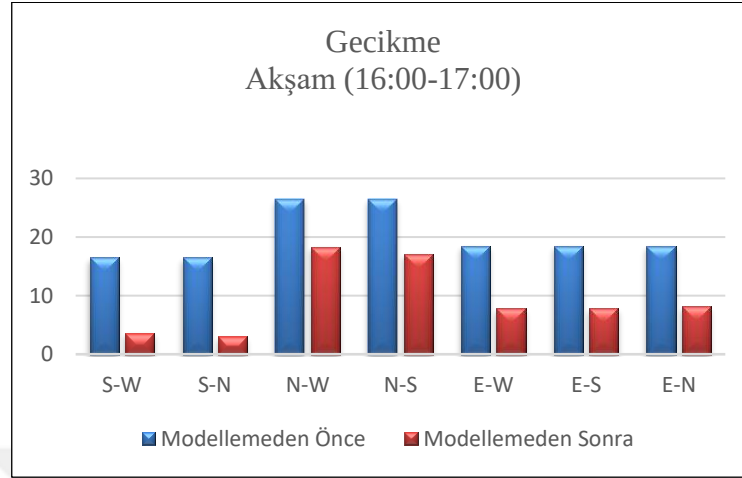
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (75 S) kullanarak modellemeden önce ve modellemeden sonra kavşağımızda yönlerin hepsinde (S-W, S-N, N-W, N-S, E-W, E-S ve E-N) kuyruk uzunluğu ve gecikme göz önüne alındığında ikisinde de iyileşmeler olduğu gözlemlenmiştir.

3.8.1. Akşam Her Yön İçin Kuyruk Uzunluğu Karşılaştırma Sonuçları



Şekil 57. Akşam modelleme öncesi ve sonrası kuyruk uzunluklarının karşılaştırılması

3.8.1. Akşam Her Yön İçin Gecikme Karşılaştırma Sonuçları

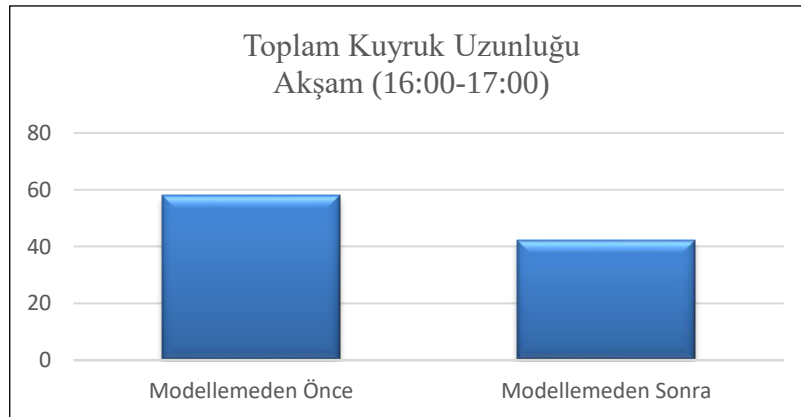


Şekil 58. Akşam modelleme öncesi ve sonrası gecikmenin karşılaştırılması

3.9. Akşam 1 Saat İçinde Kavşağın Toplam Kuyruk Uzunluğu ve Gecikme

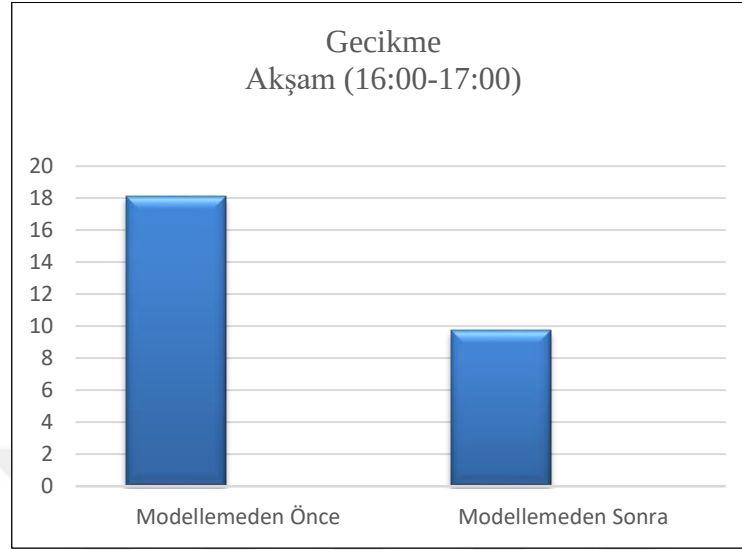
VİSSİM kullanarak elde ettiğimiz Optimum yeşil döngü süresini (60 S) kullanarak modellemeden önce ve modellemeden sonra 1 saat içinde kavşağımızda Toplam kuyruk uzunluğu ve toplam gecikme göz önüne alındığında ikisinde de iyileşmeler olduğu gözlemlenmiştir.

3.9.1. Akşam Toplam Kuyruk Uzunluğu Karşılaştırma Sonuçları



Şekil 59. Akşam modelleme öncesi ve sonrası toplam kuyruk uzunluklarının karşılaştırılması

3.9.2. Akşam Toplam Gecikme Karşılaştırma Sonuçları

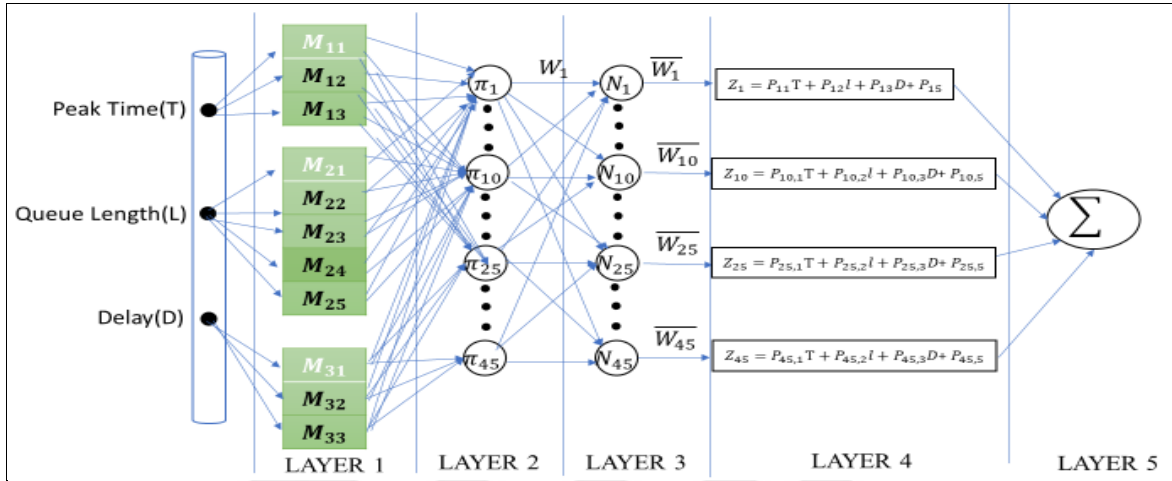


Şekil 60. Akşam modelleme öncesi ve sonrası toplam gecikmenin karşılaştırılması

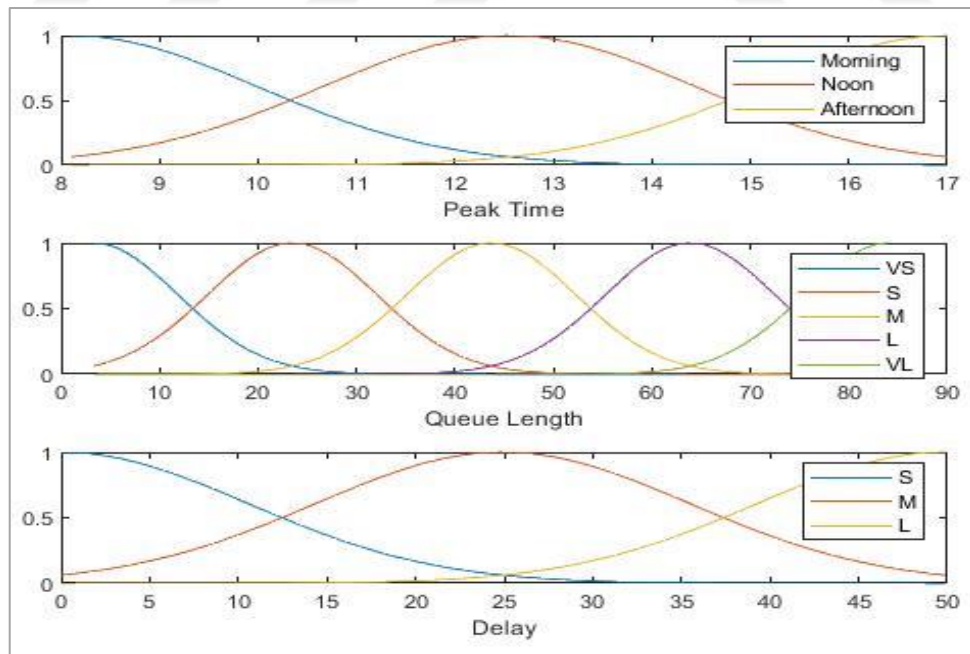
3.10. Tasarlanmış ANFIS Yapısı

ANFIS mimarisinde hem sinir ağı (NN) hemde bulanık mantık (FL), kullanılmaktadır. ANFIS'in ağ yapısı, öncül ve sonuç kısımları olarak adlandırılan iki bölümden oluşmaktadır. ANFIS eğitimi bir optimizasyon algoritması kullanılarak bu parçalara ait parametrelerin belirlenmesini ifade eder. ANFIS eğitim sırasında mevcut giriş-çıkış veri çiftlerini kullanmaktadır. Daha sonra bu parçaların birbirine bağlı olduğu IF-THEN bulanık kuralları elde edilmiştir. Bu çalışmada tasarlanan ANFIS yapısı Şekilde 61'de görüldüğü gibi beş katmandan oluşmaktadır. Bu şekilde, 3 girişe sahip ANFIS yapısı görülebilmektedir; ilk giriş (zirve zamanı) için Sabah, Öğlen ve akşam olarak üç üyelik fonksiyonu ele alınmıştır, ikinci giriş (kuyruk uzunluğu) için beş üyelik fonksiyonu VS (çok küçük), S (küçük), M (orta), L (büyük) , VL (çok büyük) ve son girdi (gecikme) için üç üyelik fonksiyonu S (küçük), M (orta) ve L (büyük) olarak değerlendirilmiştir. Üyelik işlevlerinin her biri ile ilişkili iki üyelik işlevi (alt sınır, üst sınır) mevcuttur. İki aralıkta 11 üyelik fonksiyonu göz önüne alındığında, birinci katmanda toplam 22 karar değişkenine sahip olmaktadır (veya kromozom, vektör olarak adlandırılır). Her bir giriş üyelik fonksiyonun aralığı, Şekilde 62'de gösterilmiştir. Üyelik işlevlerini birbiriyle çarparak (3 *

5 * 3=45) doğrusal denklemi simgeleyen toplam 45 kuralımız elde edilmiştir. Her doğrusal denklem 4 parametreden oluşmaktadır. Dördüncü katmanda toplam 180 bireysel karar mevcuttur (4 * 45). Bu yapıda metasezgisel algoritmalar tarafından ayarlanan toplam 202 karar değişken parametresine sahip olunmuştur.



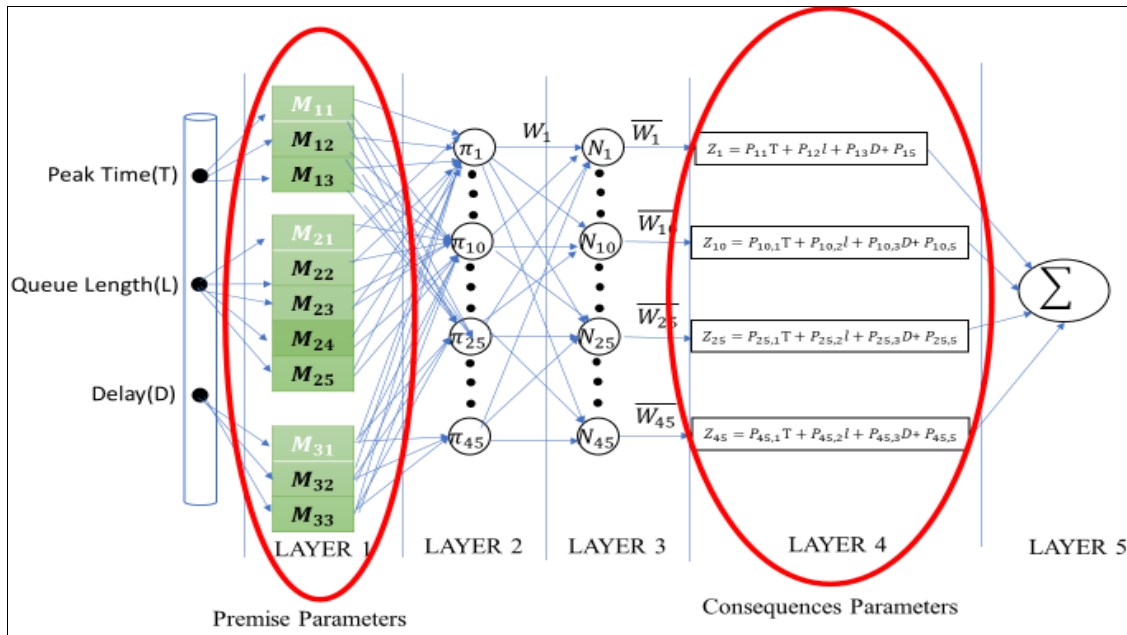
Şekil 61. Üç girişi ve bir çıkışı olan ANFIS yapısı



Şekil 62. Tasarlanmış üyelik fonksiyonları

3.10.1. Tek Amaçlı Optimizasyon Algoritmalarına Dayalı ANFİS Uygulanması

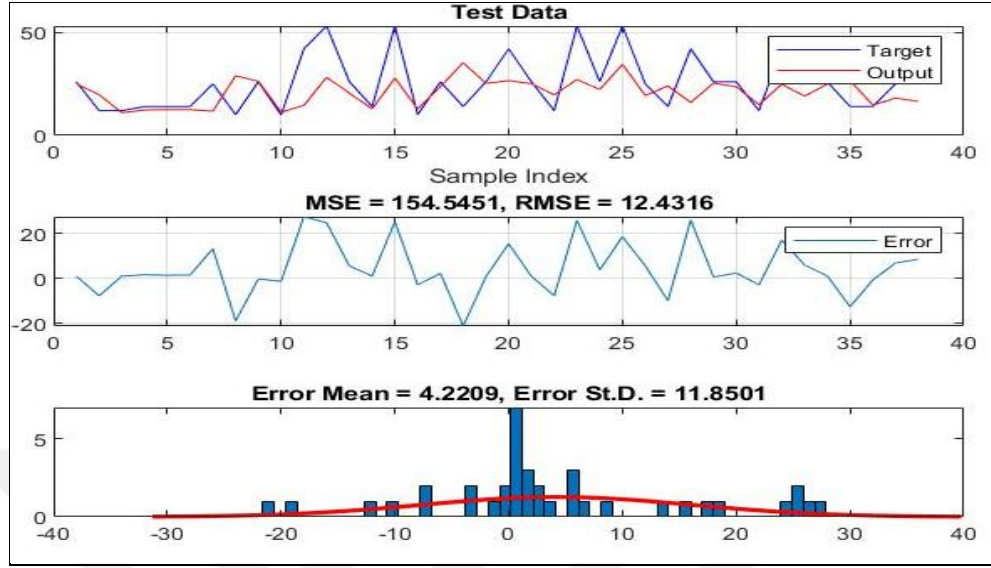
ANFİS'in eğitimi, bir optimizasyon algoritması kullanarak yapısındaki parametrelerin belirlenmesi anlamına gelmektedir. Eğitim aşamasında Şekil 63'de gösterildiği gibi öncül ve sonuç parametreleri kullanılmaktadır. ANFİS kullanarak etkili ve gerçeğe yakın sonuçlar elde edebilmemiz için başarılı bir eğitim yapmamız şarttır. ANFİS'in ilk gelişiminden bu yana, daha iyi performans elde edilebilmesi için farklı eğitim yaklaşımları önerilmiştir. Bu yaklaşımlar türev tabanlı, meta-sezgisel tabanlı ve melez olarak üçe ayrılır. ANFİS eğitiminde türev tabanlı optimizasyon algoritmaları kullanıldığında, yerel minimum risk söz konusudur. Bu nedenle, meta-sezgisel algoritmalar ANFİS eğitiminde yaygın olarak kullanılmaktadırlar. Literatürdeki çalışmalar incelendiğinde, ANFİS'in tüm bölümlerinin veya bir kısmının meta-sezgisel algoritmalar ile eğitildiği görülmektedir. ANFİS eğitimi için standart metaheuristik algoritma versiyonları ve varyantları kullanılır. ANFİS eğitiminde kullanılan meta-sezgisel algoritmaların sayısı son yıllarda hızla artmaktadır. Bu çalışmada kullanılan meta-sezgisel algoritmalar GA, PSO, HS'dir. Mutlak hata, tahmin değerlendirmesi ile fiili değerlendirme arasındaki farktır. Standart sapma Sd (σ), MSE ve RMSE hata fonksiyonları hesaplanarak hatalar hesaplanmıştır.



Şekil 63. ANFİS eğitiminde kullanılan öncül ve sonuç parametrelerin gösterilmesi

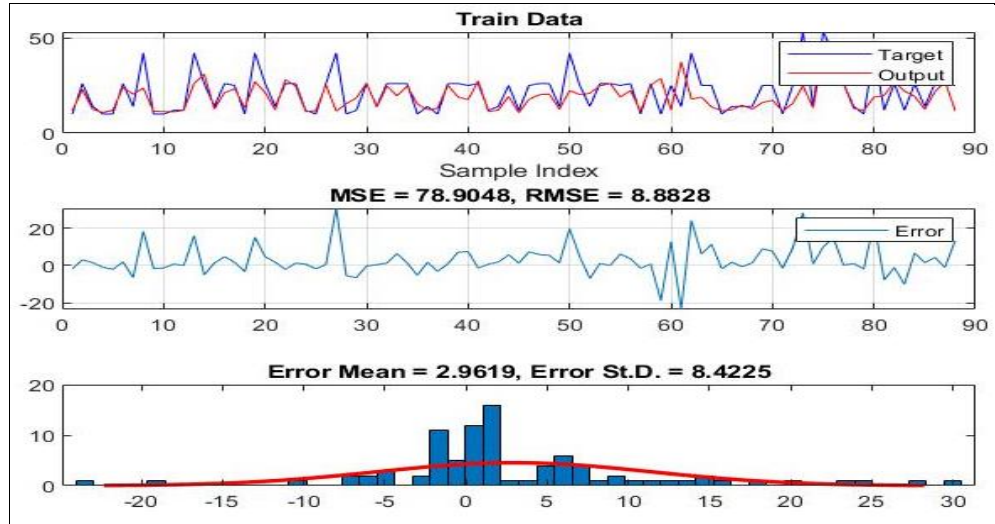
Simülasyon öncesi ve sonrası kesişimin performansı karşılaştırılmıştır. Optimal yeşil sinyal süresi kullanılarak, üç zirve saatlerinde toplam kuyruk uzunluğu ve toplam gecikme açısından iyileşme gözlenmiştir ve Şekil 49-560'da gösterilmiştir. Bu çalışmada tasarlanan trafik tahmin sistemi, MATLAB'da gerçek senaryolan altında simüle edilmiş üç fazlı bir kavşak kullanılarak test edilmiştir. Üç meta-sezgisel algoritma GA, PSO ve HS ile tasarlanan ANFIS yapısının tahmin verimliliği karşılaştırılmıştır. Greene ve çalışma arkadaşlarının açıklamalarına dayanarak [Tanımı: Greene, William H.; "Econometric Analysis" (fifth edition), Prentice-Hall, 1993, p. 926, fn. 11] Bir tahmincinin ortalama karesel hata değerini (MSE) ifade etmektedir. Bu fonksiyonda çekirdek yoğunluğunu tahmin edilmektedir, tahmincisinin, tahmin ettiği değerler ve gerçek değerleri ima eden değerler arasındaki farkı ölçmek için kullanılan yollardan biridir. MSE karesel hata kaybı ya da ikinci dereceden kaybı beklenen değere karşılık gelen bir risk fonksiyonu olarak kabul edilir. Bu çalışmada kullanılan her algoritmanın genel ve spesifik parametreleri Tablo 1'de ayrıca liste halinde verilmiştir. Her bir meta-sezgisel algoritmanın eğitim ve test sonuçları, altıda ayrı ayrı sunulmuştur. Standart sapma (SD), ortalama karesel hata (MSE) ve bağıl ortalama karesel hata (RMSE) her algoritma için ortalama Tablo.2, Tablo.3 ve Tablo.4'te verilmiştir. En iyi maliyet, varyans ve ortalama sonuçları Tablo 5'te sunulmuştur. GA algoritmasının diğer algoritmalarla aynı koşullarda en iyi maliyeti MSE için **(78,9041)** ve RMSE **(8.8828)** gösterdiği gözlemlenmiştir. Öte yandan, istikrar açısından HS diğerlerinden daha düşük varyansı **(0.1571)** gösterdi ve PSO aralarındaki en iyi ortalamayı **(10.138)** göstermiştir. Değerleri incelediğimizde, tüm algoritmalar bir hususda yetenekli oldukları ve iyi performansa sahip oldukları tespit edilmiştir. (En İyi Maliyet, Varyans, Ortalama). Genel olarak, maliyet ve uygun şekilde düşük varyansa sahip GA algoritması diğerlerinden daha iyi performans göstermiştir. Şekil 65'te olduğu gibi Modellenen kavşakta **GA-ANFIS** yapısının tahmin yeteneğinin **GA-PSO** ve **GA-HS'den** daha iyi sonuç verdiği gözlemlenmiştir.

3.10.1.1. GA'nın Test Sonu En İyi Maliyeti



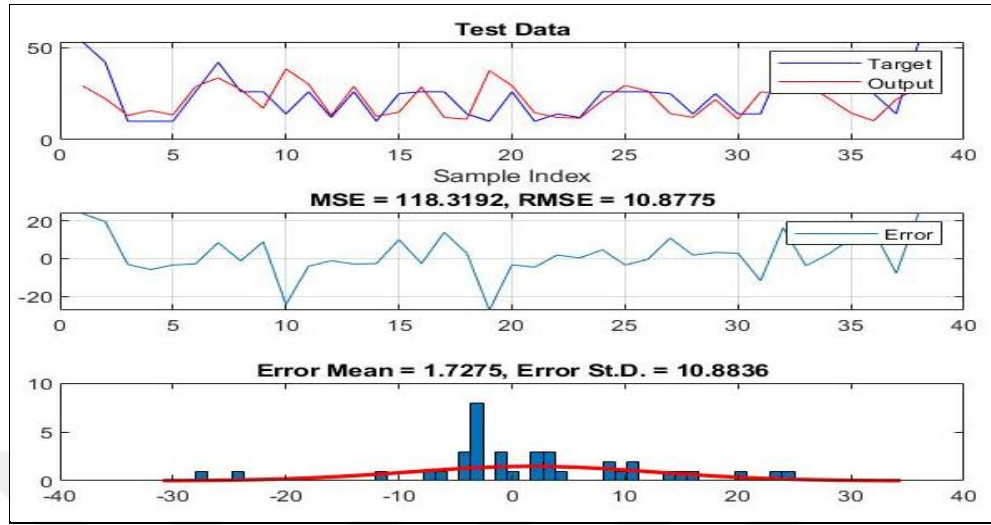
Şekil 64. GA'nın en iyi maliyet sonucu (Test)

3.10.1.2. GA'nın Eğitim Sonu En İyi Maliyeti



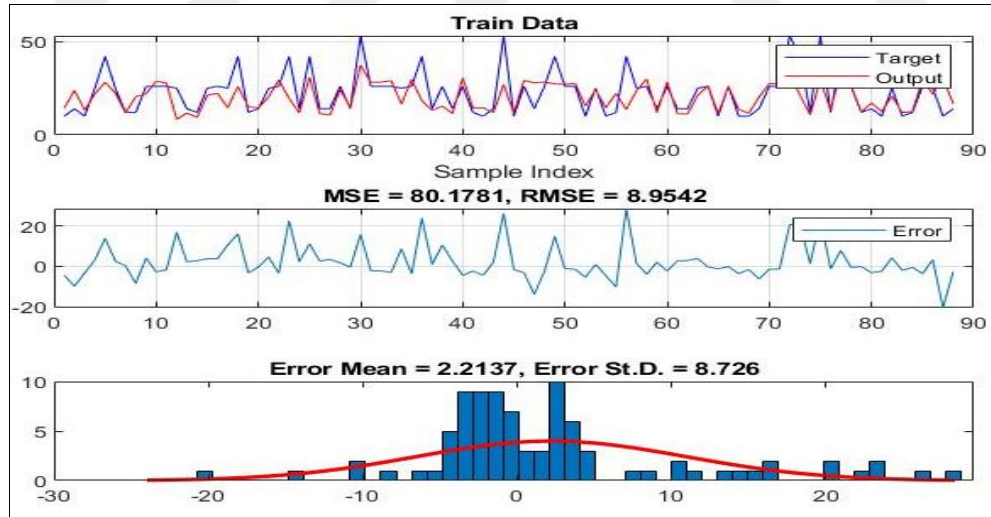
Şekil 65. GA'nın en iyi maliyet sonucu (Eğitim)

3.10.1.3. PSO'nun Test Sonu En İyi Maliyeti



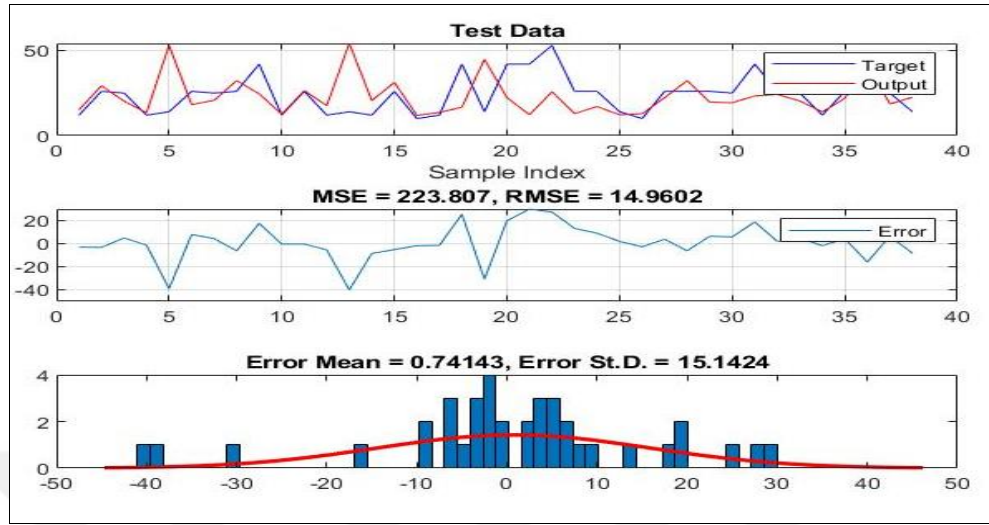
Şekil 66. PSO'nun en iyi maliyet sonucu (Test)

3.10.1.4. PSO'nun Eğitim Sonu En İyi Maliyeti



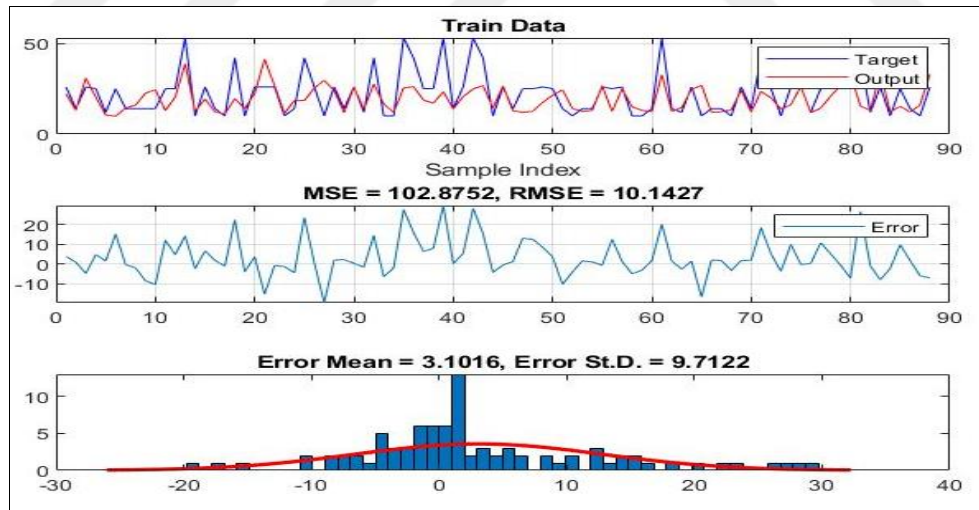
Şekil 67. PSO'nun en iyi maliyet sonucu (Eğitim)

3.10.1.5. HS'in Test Sonu En İyi Maliyeti



Şekil 68. HS'in en iyi maliyet sonucu (Test)

3.10.1.6. HS'in Eğitim Sonu En İyi Maliyeti



Şekil 69. HS'in en iyi maliyet sonucu (Eğitim)

Tablo 1. GA, PSO ve HS'nin genel ve spesifik parametreleri

Her Algoritma İçin Parametreler					
Her Algoritma İçin Genel Parametreler					
Sıra	Popülasyon boyutu	Maliyet Fonksiyon Sayısı	Karar Değişkenlerinin Sayısı		Üst Sınır/ Alt Sınır
1	25	1000	202		0.1/1
Her Algoritma için Özel Parametreler					
Adı		Gama	Mutasyon Yüzdesi	Mutasyon Oranı	Seçim Basıncı
GA	pc=0.4	0.7	pm=0.7	mu=0.15	Beta=5
Adı	Hareketsizlik Ağırlığı	Sönümleme Oranı	Kişisel Öğrenme Katsayısı		Global Öğrenme Katsayısı
PSO	w=1	W damp=0.99	c ₁ =1		c ₂ =2
Adı	Uyum Belleği Düşünme Oranı		Pitch Ayarlama Oranı		Perde Genişliği Damp Oranı
HS	HMCR=0.9		PAR=0.1		FW damp=0.995

Tablo 2. 1000 fonksiyon 10 itirasyon sonunda GA deney sonuçları

Iteration	Best Cost	RMSE (Tr)	ER.St.D (Tr)	ER.Mean (Tr)	ER.RMSE (Ts)	ER.St.D (Ts)	ER.Mean (Ts)
1	11.76049985	11.7605	11.7459	1.3828	16.8792	17.1005	0.41884
2	8.882838884	8.8828	8.4225	2.9619	12.4316	11.8501	44.2209
3	10.66358885	10.6636	10.0981	3.5914	10.0468	9.0969	4.5924
4	10.46555928	10.4656	10.1778	2.6678	12.9987	13.0968	1.3972
5	11.37986919	11.3799	10.8959	3.4852	12.175	12.0907	2.4277
6	10.09006546	10.0901	9.5103	3.5203	14.3815	14.0431	3.8483
7	9.568401517	9.5684	9.2578	2.6116	11.8227	10.8292	5.0587
8	10.54950012	10.5495	10.5855	0.71533	14.6317	14.8272	-0.16032
9	12.05980484	12.0598	11.7917	2.8239	9.8268	9.2893	3.5419
10	11.76049985	11.7605	11.7459	1.3826	16.8792	17.1005	0.41844

Tablo 3. 1000 fonksiyon 10 tekr r sonunda PSO deney sonu ları

Iteration	Best Cost	RMSE (Tr)	ER.St.D (Tr)	ER.Mean (Tr)	ER.RMSE (Ts)	ER.St.D (Ts)	ER.Mean (Ts)
1	10.96031598	10.9603	10.5717	3.1046	7.3678	7.2607	1.7183
2	8.954222742	8.9542	8.726	2.2137	10.8775	10.8836	1.7275
3	9.788586236	9.78.86	9.3292	3.126	12.1447	12.082	2.3149
4	13.85598915	13.856	13.8428	1.5949	14.2084	14.3982	-0.1562
5	9.009749371	9.0097	8.7582	2.3111	12.6672	12.7944	1.0339
6	9.255105177	9.2551	9.028	2.2537	12.5431	12.6575	1.1546
7	10.10974672	10.97	9.8905	2.3445	10.0512	9.1912	4.3326
8	9.659392225	9.6594	9.4998	2.3223	10.1669	9.5308	3.8627
9	9.358697992	9.3587	9.2467	1.7481	17.9272	18.2386	-0.86387
10	10.43044628	10.944	9.1465	2.1027	10.9815	9.4526	4.7193

Tablo 4. 1000 fonksiyon 10 tekr r sonunda HS deney sonu ları

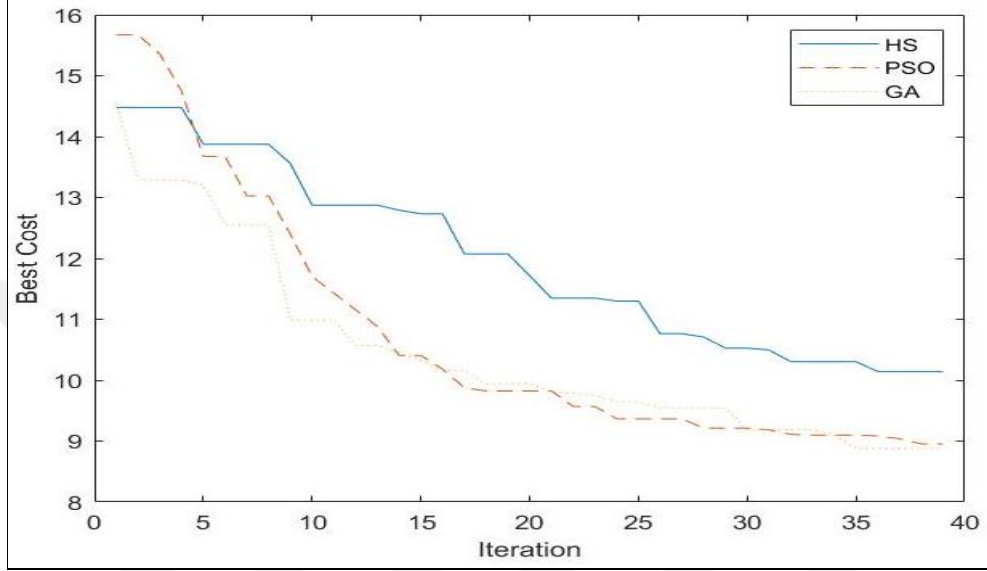
Iteration	Best Cost	RMSE (Tr)	ER.St.D (Tr)	ER.Mean (Tr)	ER.RMSE (Ts)	ER.St.D (Ts)	ER.Mean (Ts)
1	10.14274137	10.1427	9.7122	3.1016	14.9602	15.1424	0.74143
2	11.38296531	11.383	10.5295	4.4671	11.4969	11.3076	2.7716
3	10.14274137	10.1427	9.7122	3.1016	14.9602	15.1424	0.74143
4	10.89375463	10.8938	10.1777	4.0331	11.762	10.3535	4.2786
5	10.80272467	10.7422	10.0904	3.5440	11.2243	11.102	2.0272
6	10.73027223	10.7303	10.2134	3.4655	11.6683	11.7177	1.5673
7	10.75195862	10.752	10.1948	3.5851	11.3179	11.194	2.4672
8	10.96160988	10.9616	10.7472	2.4427	10.3179	10.2493	-2.0432
9	11.01776643	11.0178	10.4485	3.6688	11.2907	11.4217	0.6768
10	11.14744368	11.1474	10.455	4.0249	13.6499	13.3446	3.5954

Tablo 5. Aynı kriterler de en iyi maliyet, ortalama ve varyans sonu ları

Row/Name	Best Cost		
	PSO	HS	GA
1	10.96031598	10.14274137	11.76049985
2	8.954222742	11.38296531	8.882838884
3	9.788586236	10.14274137	10.66358885
4	13.85598915	10.89375463	10.46555928
5	9.009749371	10.80272467	11.37986919
6	9.255105177	10.73027223	10.09006546
7	10.10974672	10.75195862	9.568401517
8	9.659392225	10.96160988	10.54950012
9	9.358697992	11.01776643	12.05980484
10	10.43044628	11.14744368	11.76049985
Varyans	2.110913086	0.157109225	1.065655198
Ortalama	10.13822519	10.79739782	10.71806278

3.11. Tek Amaçlı Optimizasyon Algoritmalarının Karşılaştırması

Modellenen kavşakta **GA-ANFIS** yapısının tahmin yeteneğinin **GA-PSO** ve **GA-HS**'den daha iyi sonuç verdiğini görmekteyiz.



Şekil 70. ANFİS ile birlikte meta-sezgisel algoritmaların performansının karşılaştırılması

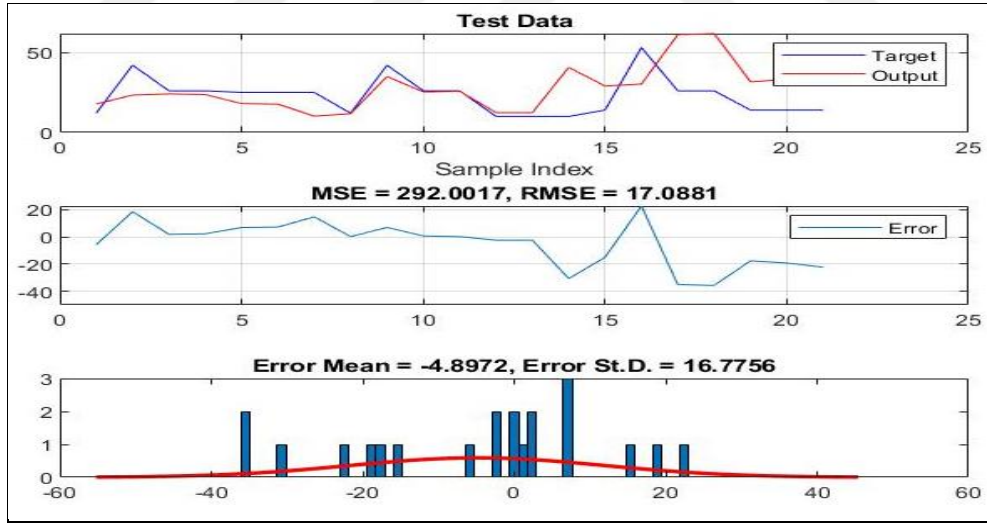
3.12. Çok Amaçlı Optimizasyon Algoritmalarına Dayalı ANFİS Uygulanması

Bu bölümde, çok-amaçlı algoritmaları kullanabilmek için önceki bölümde tasarladığımız ANFİS yapısı kullanılmıştır. Şekil.61'de ANFİS yapısının tüm özellikleri ve parametreleri teker teker ve açık bir şekilde sunulmuştur. NSGAIİ ve MOPSO'ya ait genel ve spesifik parametreler Tablo.10'da verilmiştir. Bu çalışmada çok-amaçlı algoritma bölümünde bahsettiğimiz gibi üç hata (maliyet) fonksiyonu kullanılmıştır ve bu fonksiyonların birbiri ile aynı kriterler altında verdiği sonuçlar. NSGAIİ ve MOPSO algoritmaları aynı koşullar altında F1 ve F2 maliyet fonksiyonlarını kullandıkları zaman, NSGAIİ için en iyi maliyet 57,55 ve MOPSO için 45,48 olduğu hem test ve hem eğitim aşamasında sırasıyla Şekil 71, şekil 72, Şekil.78 ve Şekil.79'de görülmektedir. Ayrıca elde ettiğimiz tüm sonuçlar Tablo.6'da sunulmuştur. NSGAIİ ve MOPSO algoritmaları aynı koşullar altında F1 ve F3 maliyet fonksiyonlarını kullandıkları zaman, NSGAIİ için en iyi maliyet 52,56 ve MOPSO için 43,40 olduğu hem test ve hem eğitim aşamasında sırasıyla

şekil.73, şekil.74, Şekil.81 ve Şekil 82'de görülmektedir. Ayrıca elde ettiğimiz tüm sonuçlar Tablo 7'de sunulmuştur. NSGAIİ ve MOPSO algoritmaları aynı koşullar altında **F2 ve F3** maliyet fonksiyonlarını kullandıkları zaman, NSGAIİ için en iyi maliyet **56,36** ve MOPSO için **49,30** olduğu hem test ve hem eğitim aşamasında sırasıyla şekil.75, şekil.76, Şekil.82 ve Şekil 83'de görülmektedir. Ayrıca elde ettiğimiz tüm sonuçlar Tablo 8'de sunulmuştur. Her bir algoritmaya ait en iyi ortalama ve varyans Tablo 9 ve Tablo 10'da açıklanmıştır. Sonuçları incelediğimiz zaman her bir algoritmanın (NSGAIİ, MOPSO) kendine ait özellikleri olduğu görülmektedir, Ancak MOPSO algoritmasının performansı ve sonuçları NSGAIİ'a göre daha iyi olduğu tespit edilmiştir.

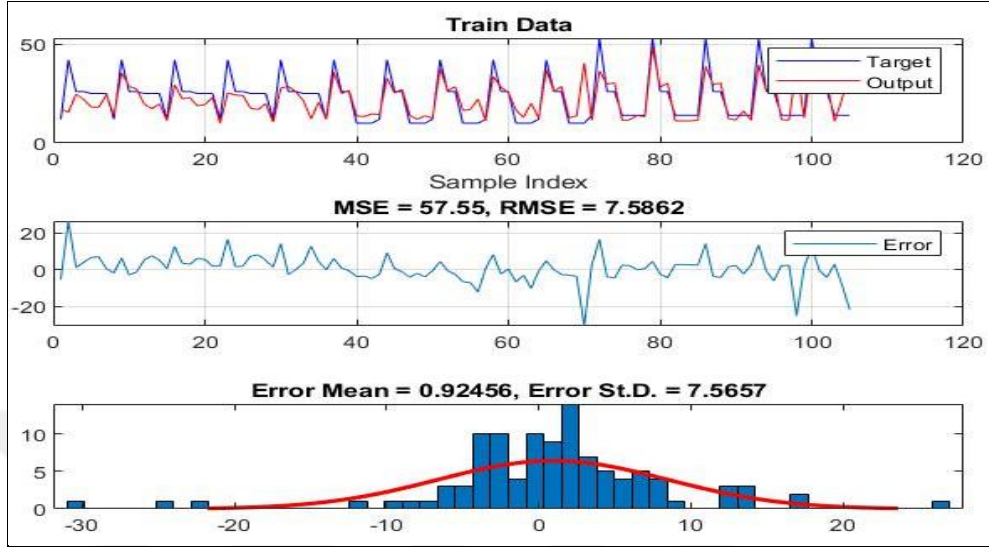
3.12.1. NSGAIİ Algoritması F1 ve F2 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.1.1. NSGAIİ'nın F1ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu



Şekil 71. NSGAIİ'nın F1ve F2 fonksiyonları ile en iyi maliyet sonucu (Test)

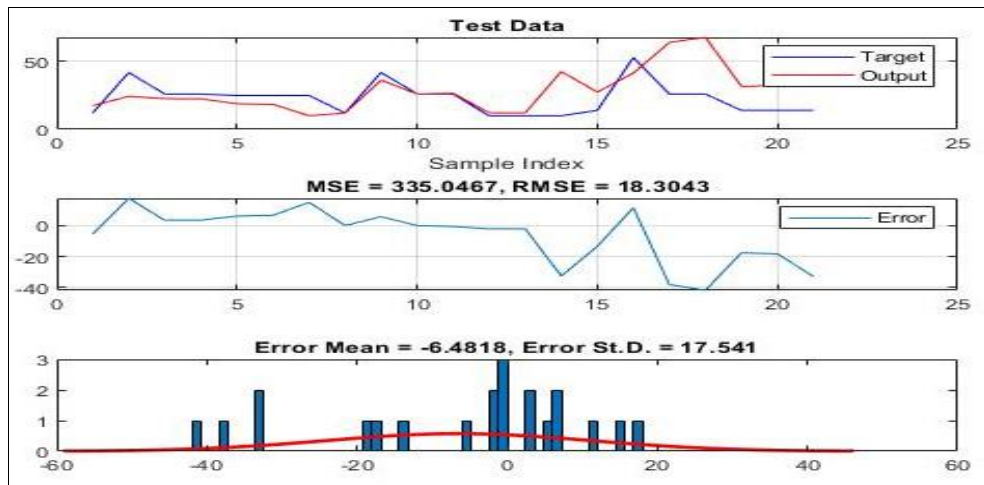
3.12.1.2. NSGAII'nın F1ve F2 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu



Şekil 72. NSGAII'nın F1ve F2 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

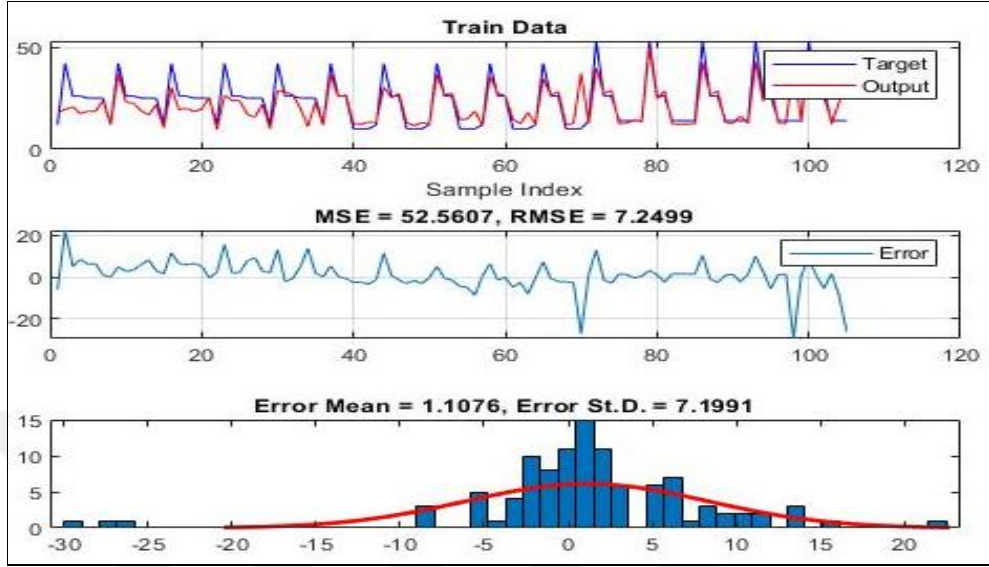
3.12.2. NSGAII Algoritması F1 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.2.1. NSGAII'nın F1ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu



Şekil 73. NSGAII'nın F1ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)

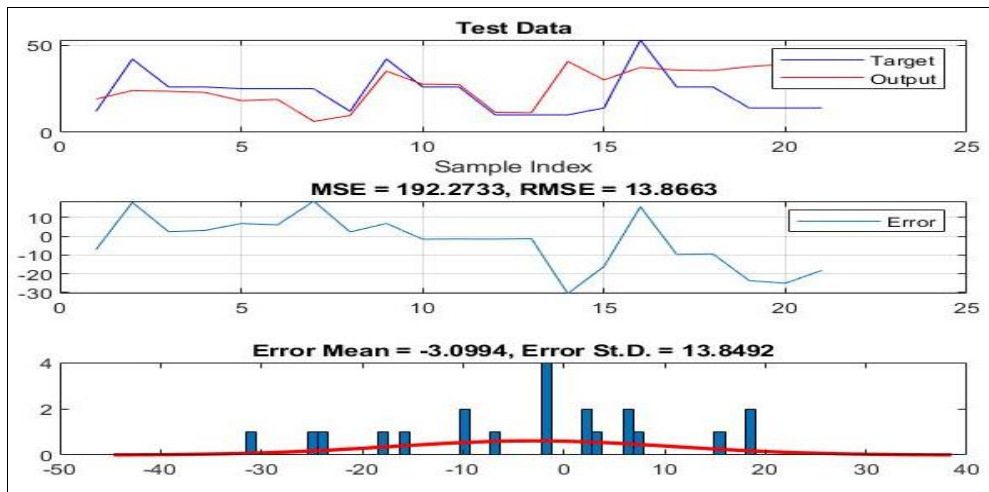
3.12.2.2. NSGAII'nın F1ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu



Şekil 74. NSGAII'nın F1ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

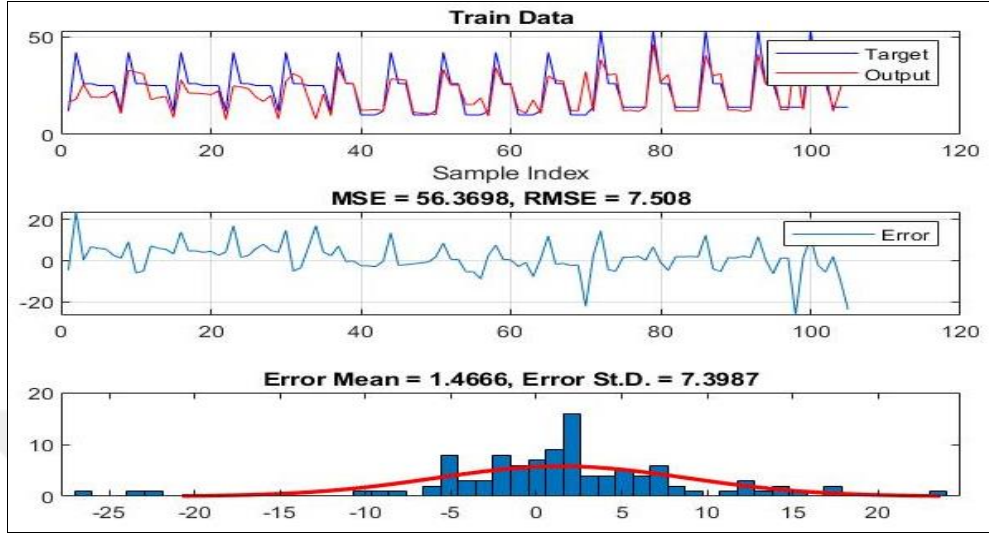
3.12.3. NSGAII Algoritması F2 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.3.1. NSGAII'nın F2ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu



Şekil 75. NSGAII'nın F2ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)

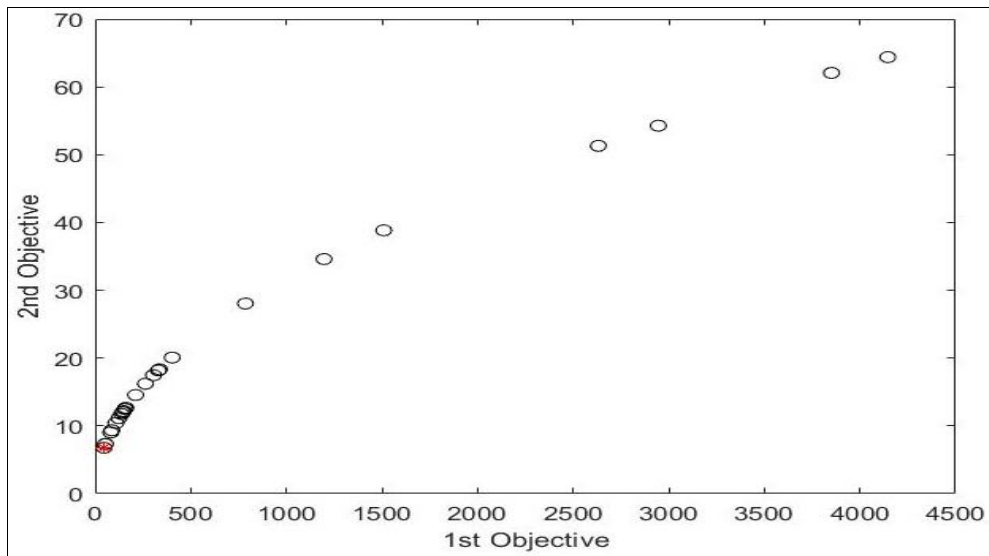
3.12.3.2. NSGAIİ'nin F2 ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu



Şekil 76. NSGAIİ'nin F2ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

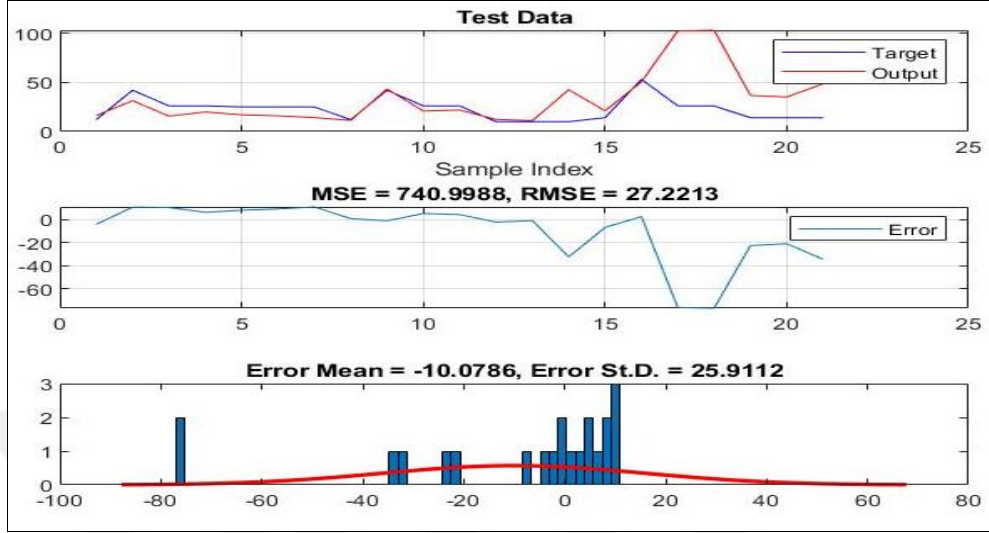
3.12.4. MOPSO Algoritmasının F1 ve F2 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.4.1. MOPSO Algoritmasının F1 ve F2 Maliyet Fonksiyonu ile Birlikte Dağılımı



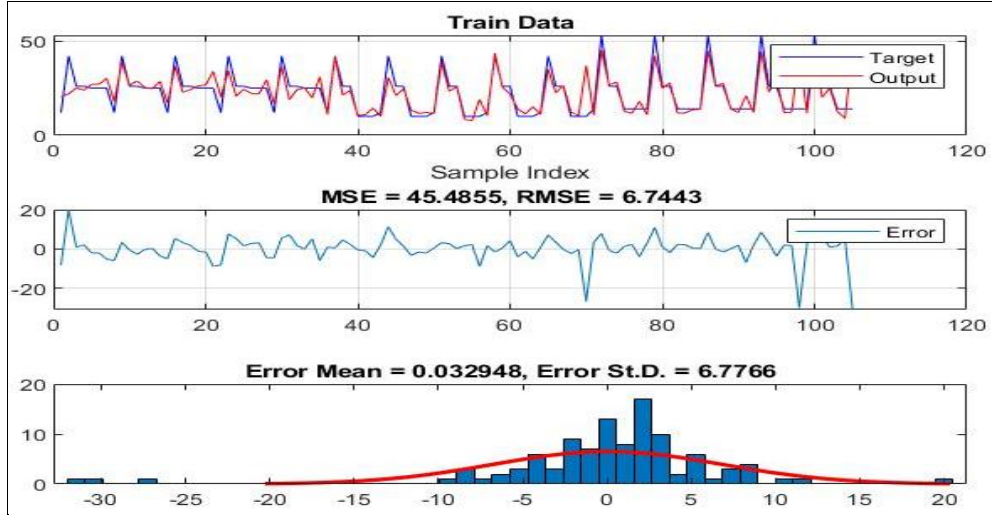
Şekil 77. MOPSO'nun F1 ve F2 fonksiyonları ile dağılım şeması

3.12.4.2. MOPSO Algoritmasının F1 ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu



Şekil 78. MOPSO'nun F1 ve F2 fonksiyonları ile en iyi maliyet sonucu (Test)

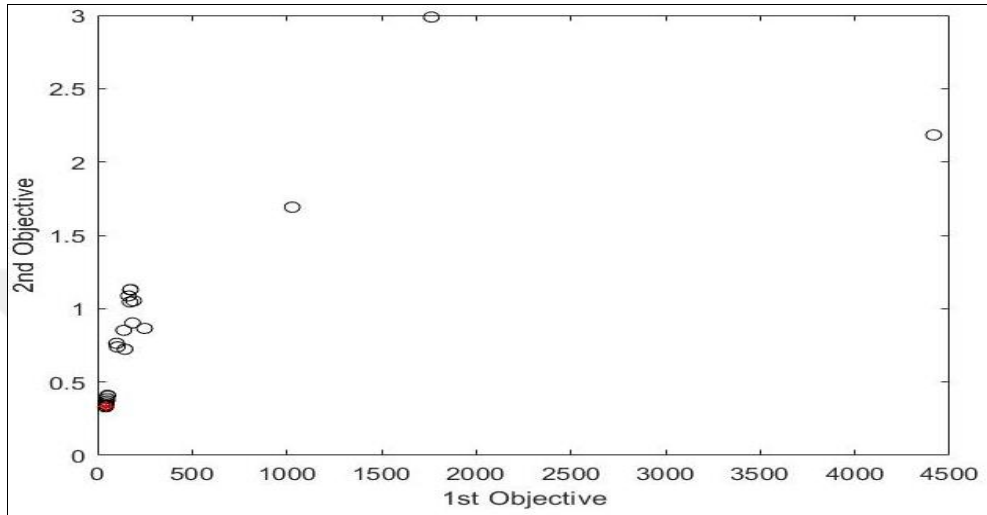
3.12.4.3. MOPSO Algoritmasının F1 ve F2 Fonksiyonları ile Eğitim Sonu En İyi Maliyet sonucu



Şekil 79. MOPSO'nun F1 ve F2 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

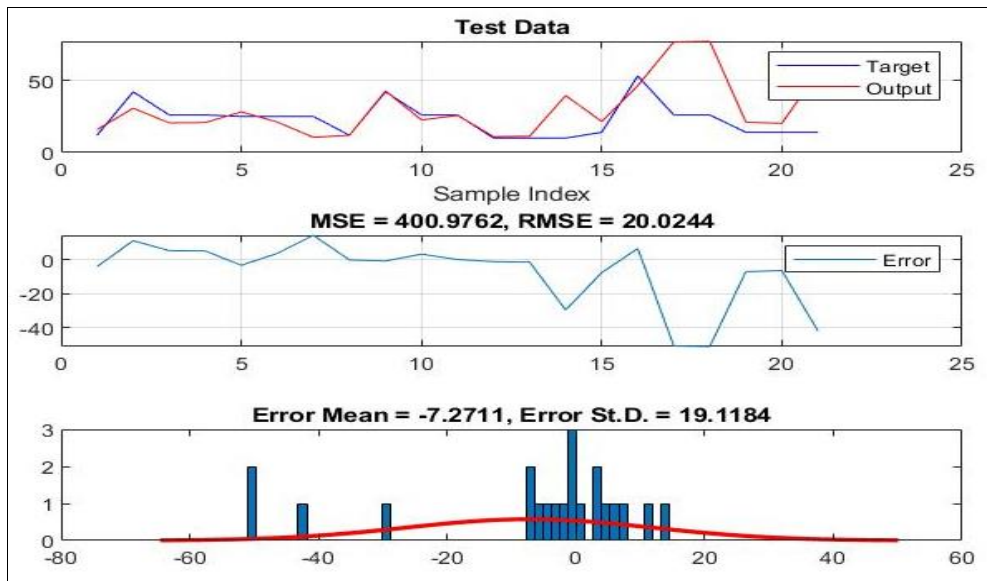
3.12.5. MOPSO Algoritmasının F1 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.5.1. MOPSO Algoritmasının F1 ve F3 Maliyet Fonksiyonu ile Birlikte Dağılımı



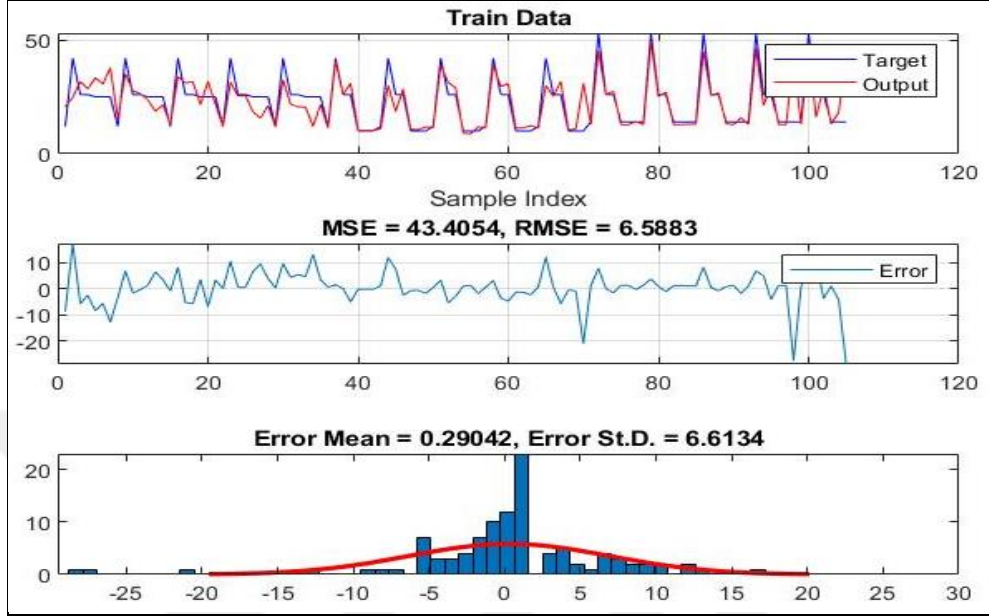
Şekil 80. MOPSO'nun F1 ve F3 fonksiyonları ile dağılım şeması

3.12.5.2. MOPSO Algoritmasının F1 ve F2 Fonksiyonları ile Test Sonu En İyi Maliyet Sonucu



Şekil 81. MOPSO'nun F1 ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)

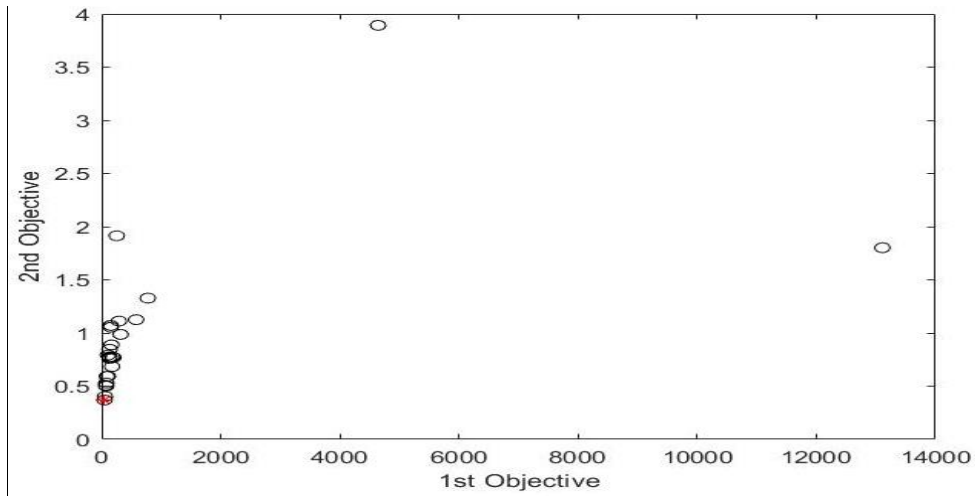
3.12.5.3. MOPSO Algoritmasının F1 ve F3 Fonksiyonları ile Eğitim Sonu En İyi Maliyet Sonucu



Şekil 82. MOPSO'nun F1 ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

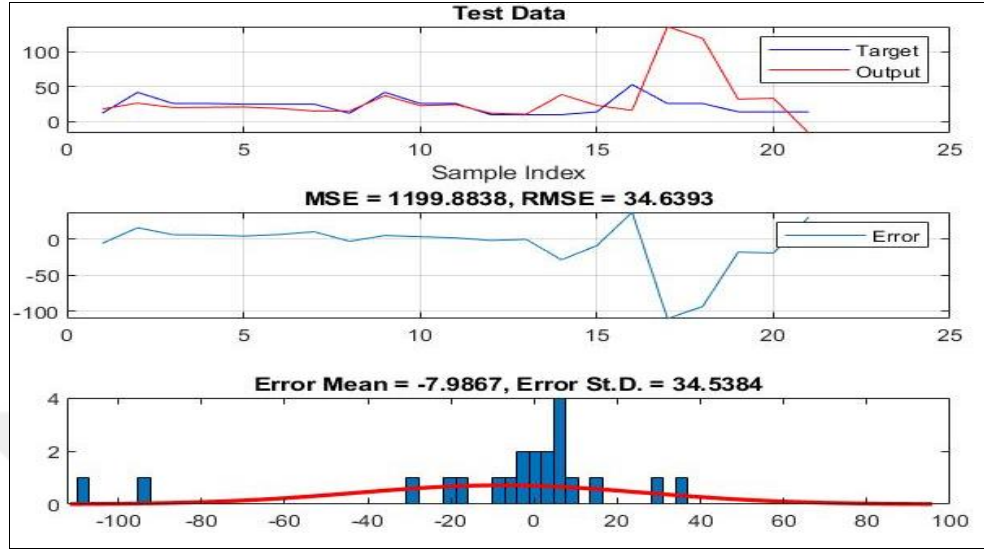
3.12.6. MOPSO Algoritmasının F2 ve F3 Maliyet Fonksiyonu ile Birlikte Performans Sonuçları

3.12.6.1. MOPSO Algoritmasının F2 ve F3 Maliyet Fonksiyonu ile Birlikte Dağılımı



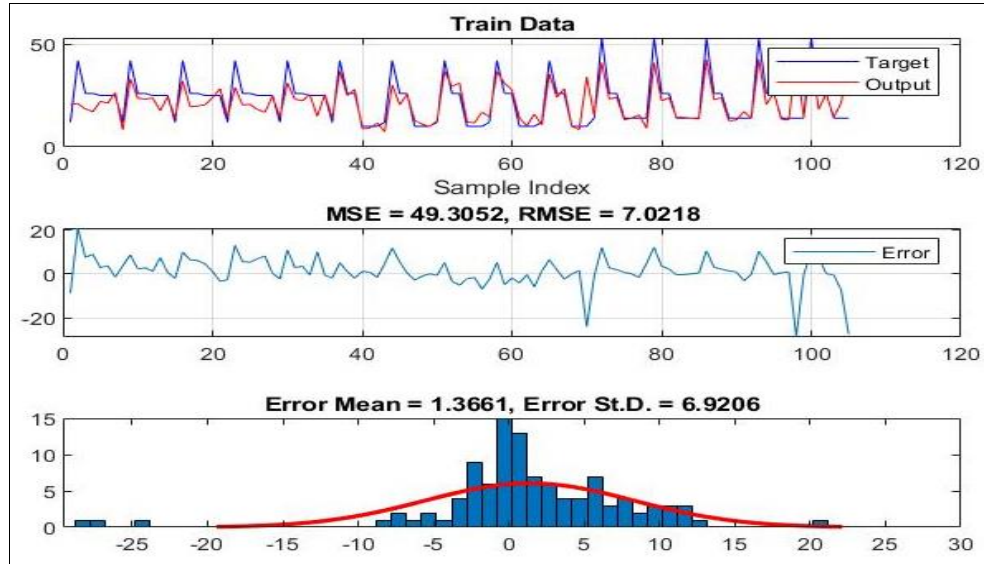
Şekil 83. MOPSO'nun F2 ve F3 fonksiyonları ile dağılım şeması

3.12.6.2. MOPSO Algoritmasının F2 ve F3 Fonksiyonları ile Test Sonu En İyi Maliyet sonucu



Şekil 84. MOPSO'nun F2 ve F3 fonksiyonları ile en iyi maliyet sonucu (Test)

3.12.6.3. MOPSO Algoritmasının F2 ve F3 Fonksiyonları ile Rğitim Sonu En İyi Maliyet Sonucu



Şekil 85. MOPSO'nun F2 ve F3 fonksiyonları ile en iyi maliyet sonucu (Eğitim)

Tablo 6. Tekerrür sonunda F1 ve F2 fonksiyonlarının deney sonuçları

Tekerrür	MOPSO		NSGAI	
	F1	F2	F1	F2
1	64.93303	8.058104	73.5309	8.575
2	56.0535	7.4869	57.55	7.5862
3	55.54185	7.452641	56.0957	7.4898
4	50.50261	7.106519	94.7183	9.7323
5	56.05349	7.486888	94.6961	9.7312
6	55.54185	7.452641	94.7183	9.7323
7	53.37808	7.30603	94.6961	9.7312
8	56.87676	7.541668	59.3576	7.7044
9	45.48548	6.744292	65.9408	8.1204
10	49.26505	7.018906	65.3519	8.0841

Tablo 7. 10 tekerrür sonunda F1 ve F3 fonksiyonlarının deney sonuçları

Tekerrür	MOPSO		NSGAI	
	F1	F3	F1	F3
1	54.17621	7.3604	74.3768	8.6242
2	72.78091	8.5312	62.7529	7.9217
3	44.48834	6.67	56.5465	7.5197
4	43.40544	6.5883	66.9076	8.1797
5	54.212	7.3629	77.7665	8.8185
6	51.28458	7.1613	76.2823	8.734
7	121.9387	11.0426	73.495	8.5729
8	60.65953	7.7874	90.302	9.5027
9	57.69877	7.596	52.5607	7.2492
10	49.59777	7.0426	81.3869	9.0215

Tablo 8. 10 tekerrür sonunda F2 ve F3 fonksiyonlarının deney sonuçları

Tekerrür	MOPSO		NSGAI	
	F2	F3	F2	F3
1	117.9106	10.8587	68.5903	8.2819
2	56.0455	7.4764	76.5492	8.7492
3	49.30515	7.0218	79.0134	8.8889
4	60.6595	7.7884	61.2886	7.8287
5	81.1355	9.0075	56.3698	7.508
6	65.7	8.1056	62.5277	7.9074
7	71.8668	8.4774	78.0871	8.8367
8	60.6596	7.7884	75.5067	8.6895
9	81.1355	9.0075	56.9578	7.547
10	60.6595	7.7884	57.0493	7.5531

Tablo 9. MOPSO algoritmasının F1, F2 ve F3 maliyet fonksiyonları ile 10 tekerrür sonu performans değerlendirmesi

Maliyet Fonksiyonu	MOPSO					
	F1		F2		F3	
	Ortalama	Varyans	Ortalama	Varyans	Ortalama	Varyans
F1-F2	54.36317	5.240989	7.365459	0.354656		
F2-F3			70.50777	19.57197	8.33201	1.090585
F1-F3	61.02423	23.00379			7.71427	1.296508

Tablo 10. NSGAIİ algoritmasının F1, F2 ve F3 maliyet fonksiyonları ile 10 tekerrür sonu performans değerlendirmesi

Maliyet Fonksiyonu		NSGAIİ					
		F1		F2		F3	
		Ortalama	Varyans	Ortalama	Varyans	Ortalama	Varyans
F1	F2	75.66557	17.10406	8.64869	0.98087		
F2	F3			67.19399	9.41886	8.17904	0.574528
F1	F3	71.23772	11.57046			8.41441	0.694707

Tablo 11. MOPSO ve NSGAIİ algoritmalarına ait parametreler

MOPSO	nRep=5; w=0.5; wdamp=0.99; c1=2; c2=2; nGrid=5; $\alpha=0.1$; $\beta=2$; $\sigma=2$; mu=2;	% Depo Büyüklüğü % Atalet Ağırlığı % Atalet Ağırlığının Baskı Oranı % Personal Öğrenme Katsayısı % evrensel Öğrenme Katsayısı % Her Boyuttadki Bölme Sayısı % Şişme Boyutu % Leader Seçim Basıncı % Silme Seçim Basıncı % Mutasyon Oranı
NSGAIİ	PC=0.8; NC=2*round(PC*Toplum Sayısı/2); PM=0.9; NM=round(pMutation*nPop); mu=0.2; $\sigma=0.1*(VarMax-VarMin)$;	% çaprazlama yüzdesi % Ebevelerin Sayısı %Mutasyon Yüzdesi % Mutasyon Boyutu Mutants % Mutasyon Oranı % Mutasyon Adım Büyüklüğü

4. SONUÇLAR

Kentsel trafiğin en önemli sorunlarından biri de tıkanıklıktır ve gelişmekte olan ülkelerin en kritik sorunlarından biri haline gelmeye devam etmektedir. Zamanında ve etkin hizmet sunmak için çözümün teşvik edilmesi gerekmektedir. Büyük miktarda sera gazı (CHG) salınımı, kaza sayısındaki artış, trafik kuyruklarında oluşan zaman ve enerjini kaybı gibi birçok yönden kent insanının çözülmesi gereken problemlerinden biri haline gelmiştir. Günümüzde trafik sinyal zamanlamasını optimize eden yöntemler, kavşaktaki trafik dinamiklerini yeterince yakalaymayan matematiksel modellere dayanmaktadır. Bu nedenle trafiğin yoğun olduğu saatlerde kavşaklardaki tıkanıklık sorununu aşabilmek için düşük performanslı matematiksel modeller yerine uyarlamalı trafik kontrol modellerinin kullanımı kaçınılmaz hale gelmiştir.

İstanbul, Türkiye'nin en önemli metropol şehridir. Şehir uluslararası, ticari, siyasi, akademik, sosyal, sanatsal ve kültürel anlamda her yıl çok büyük miktarda turist ağırlamaktadır. Bu yüzden gün içinde ulaşım sistemlerinin yoğun trafik saatlerinde çok karmaşık bir hale geldiği ve trafik kontrol organları için önemli zorluklara neden olduğu açıkça görülmektedir. Bu tez çalışmasındaki veriler, video kameralar vasıtasıyla Cumhuriyet Caddesi Yarımburgaz 34303 Küçükçekmece / İstanbul üzerinde dört kollu ve üç fazlı kavşaktan toplanmıştır, Excel ortamında kullanabilecek veri formatına dönüştürülmüştür ve mevcut durum olarak kayıt edilmiştir. Kavşağa ait hacim sayımı tamamlandıktan sonra zirve saat standardı elde edilmiştir. Sabah (08.00-09.00), öğlen (12.00-13.00) ve akşam (16.00-17.00) olarak üç zaman dilimine ayrılmıştır. Kuyruk uzunluğu ve gecikme hesaplanmış ve VISSİM ortamına aktarılmıştır. ANFİS mimarisine uygun, kaliteli ve hatasız girdiler elde edebilmek için VISSİM simülasyon modeli kullanılmıştır. İkinci aşama VISSİM trafik simülasyon ortamında kavşağa gelen üç bağlantı ve üç fazlı bir kavşak modellenmesiyle başlanmış, trafik hacmi verileri ve diğer bağımlı parametrelerin girilmesiyle devam edilmiştir. Her faz için optimum yeşil süresi elde edilmiştir. Optimum yeşil süresi kullanılarak kuyruk uzunluğu ve gecikme hesaplanmıştır. VISSİM modeli ile mevcut durumun Kuyruk uzunluğu ve gecikmesi karşılaştırılmış ve VISSİM modelimizde iyileşmeler olduğu görülmüştür. Geçmiş veri kümesinden öğrenmek ve geleceği tahmin edebilmek için ANFİS'in avantajları kullanılmıştır. Tasarladığımız ANFİS-Type1 mimarisi üç girdi (zirve saati, optimum

kuyruk uzunluğu ve optimum gecikme) ve bir çıktıdan (optimum yeşil zaman süresi) oluşmaktadır. ANFİS sistemin kurallarını uyarlamalı olarak öğrenen ve değiştiren birinci sıra Sugeno-FIS kullanılmıştır.

ANFİS yapısında İlk girdi, zirve saat için üç üyelik fonksiyonu sabah, öğlen ve akşam olarak belirlenmiştir. İkinci girdi, optimal kuyruk uzunluğu için çok küçük (VS), küçük (S), orta (M), büyük (L) ve çok büyük (VL) olarak ayarlanmış, Son girdi, gecikme için küçük (S), orta (M) ve büyük (L) olarak belirlenmiştir. Üyelik işlevlerinin her biri ile ilişkili iki üyelik işlevi dikkate alınmıştır (alt sınır, üst sınır). İki aralıklı 11 üyelik fonksiyonu göz önüne alınmıştır, birinci katmanda toplam 22 karar değişkenine oluştu (veya kromozom, vektör olarak adlandırılır). Üyelik işlevlerini birbiriyle çarparak $(3 * 5 * 3) = 45$ doğrusal denklemi simgeleyen toplam 45 kuralımız elde edilmiştir. Her doğrusal denklem 4 parametreden oluşmuştur. Dördüncü katmanda toplam 180 bireysel karar verilmiştir $(4 * 45) = 180$. Tasarladığımız ANFİS-Type1 mimarisi tarafından ayarlanan toplam 202 karar değişken parametresi elde edilmiştir.

ANFIS eğitimi, bir optimizasyon algoritması uygulayarak sistemindeki standartları saptamak demektir. ANFIS ile etkili sonuçlar elde etmek için güçlü bir eğitim kullanmak çok önemli bir rol oynamaktadır. Bu yüzden, çalışmamızda ANFIS eğitimi için tek-amaçlı popülasyona dayalı algoritmalar GA, PSO, HS ve çok-amaçlı popülasyona dayalı algoritmalar NSGAI ve MOPSO kullanılmıştır. Tasarladığımız şebekede Öncül (ikinci katman) ve sonuç (Dördüncü katman) olarak belirlenmiştir. Çalışmamızda tek-amaçlı algoritmalar için MSE, RMSE hata fonksiyonları tek tek uygulanmış, çok-amaçlı algoritmalar için ise MSE, RMSE ve SSE hata fonksiyonları birbiriyle kullanılmıştır ve hepsinin değerleri ve hata miktarları açıklanmıştır.

Bu çalışmada tasarladığımız ANFIS ile tek-amaçlı meta-sezgisel algoritmalar ve çok-amaçlı algoritmalara dayalı trafik tahmin modelimizi incelediğimizde. ANFİS ile tek-amaçlı metasezgisel algoritmanın performanslarına baktığımızda, ANFIS-GA'ın tahmin performansı açısından diğerlerinden daha iyi olduğunu tespit edilmiştir. ANFİS-GA modelimizin, Test setinde 154.54 ve eğitim setinde 78.90, ANFİS-PSO modelimizin, Test setinde 118.31 ve eğitim setinde 80.17, ANFİS-HS modelimiz ise Test setinde 223.80 ve eğitim setinde 102.87 olarak elde edilmiştir. Değerlere baktığımızda en düşük hata değerini ANFİS-GA ile elde ettik ve analiz sonuçlarını sunduk. Çok-amaçlı algoritmaların sonuçlarını üç farklı maliyet fonksiyonu ile incelediğimizde, ANFİS-NSGAI ve ANFİS-MOPSO algoritmaları aynı koşullar altında F1 ve F2 maliyet fonksiyonlarını kullandıkları

zaman, NSGAI için en iyi maliyet 57,55 ve MOPSO için 45,48 olduğunu hem test ve hem eğitim aşamasında elde edilmiştir ve sırasıyla şekil.71, şekil.72, Şekil.78 ve Şekil.79’da sunulmuştur. ANFİS-NSGAI ve ANFİS-MOPSO algoritmaları aynı koşullar altında F1 ve F3 maliyet fonksiyonlarını kullandıkları zaman, ANFİS-NSGAI için en iyi maliyet 52,56 ve ANFİS-MOPSO için 43,40 olduğunu hem test ve hem eğitim aşamasında elde edilmiştir ve sırasıyla şekil 71, şekil 72, Şekil 79 ve Şekil 80’de sunulmuştur. ANFİS-NSGAI ve ANFİS-MOPSO algoritmaları aynı koşullar altında F2 ve F3 maliyet fonksiyonlarını kullandıkları zaman, ANFİS-NSGAI için en iyi maliyet 56,36 ve MOPSO için 49,30 olduğunu hem test ve hem eğitim aşamasında elde edilmiştir ve sırasıyla şekil.75, şekil.76, Şekil.84 ve Şekil.86’de sunulmuştur. Derin bir karşılaştırma yaptığımızda ve sonuçları incelediğimizde her bir algoritmanın (ANFİS-NSGAI, ANFİS-MOPSO) kendine ait özellikleri olduğunu görmekteyiz, Ancak ANFİS-MOPSO algoritmasının performansı ve sonuçları ANFİS-NSGAI’ye nazaran daha iyi olduğunu tespit edilmiştir. Teknik olarak tek-amaçlı ve çok-amaçlı algoritmaların farklı yapılara ve davranışlara sahip oldukları için birbiriyle karşılaştıramayız ancak bizim tasarladığımız ANFİS yapısında çok-amaçlı algoritmaların daha iyi uyum gösterdiği görülmüştür.

Bu çalışmada önerdiğimiz başlık ile ilgili olarak, olan üstü çalışmalar yapıldığını gördük, ancak bugüne kadar çalışma alanımız için gerçek veriler kullanılarak yeşil sinyal süre tahmini için ANFIS ve metaheuristik algoritmalara sahip bir VISSIM simülatör modeli önerilmemiştir.

5. ÖNERİLER

Trafik sinyal kontrolü alanında bir çok araştırma yapılsa da, dünya genelinde günlük trafik sıkışıklığında hala önemli ölçüde zaman ve enerji kaybı yaşanmaktadır. Sebeplerden biri, teorik çalışmalar ile endüstriyel gelişmeler arasındaki boşluk ve halen uygulanmakta olan tekniklerin yetersizliğidir. Bununla birlikte, şehir trafiğindeki araç ve seyahat sayısındaki artışın rolü yadsınamaz bir gerçektir. Burada; gelecekteki çalışmalar için bazı öneriler sunulmaktadır:

1. Trafik sinyal zamanlamasının kontrolündeki çalışmaların çoğu tek bir kavşak veya birkaçı için yapıldı. Daha büyük bir trafik ağında trafik sinyalini kontrol etmek ve kavşaklar arasındaki etkin koordinasyonu dikkate almak, seyahatin gecikme süresini azalmaya yardımcı olacaktır.
2. Genellikle optimizasyon yöntemlerinin birçoğu çok fazla parametre ile karşılaştığında yakınsama gücünü çekmektedir. Gelecekteki çalışmalardan biri, optimizasyon yöntemlerinin kalitesini iyileştirmek ve çok sayıda parametreyi işlemek için yetenekler sağlamaktır.
3. Trafik sinyal zamanlaması ve en uygun kontrolörün bulunma aşamasında eğitim için yeterli veri setine ihtiyaç vardır. Doğru kontrolörlere sahip olmak, trafik koşullarını etkileyen tüm farklı faktörleri dikkate almakta yardımcı olurken, eğitim için tüm farklı vakaları sağlamak pratik değildir ve çok zaman alıcıdır. Yararlı faktörleri seçmek için bir çözüm bulmak ve hatta eğitim sırasında gereksiz faktörleri bulmak bile yararlı olacaktır.
4. ANFIS denetleyicisinin üyelik işlevlerinin sayısı sabitlenir ve manuel olarak ayarlanır. Üyelik işlevlerinin sayısını artırmak daha sorunsuz ve daha güçlü bir denetleyiciye imkan sağlayacaktır, ancak gereksiz üyelik işlevlerinin dikkate alınması olumsuz etkiler getirebilir. Bu nedenle, tam olarak doğru sayıları bulmak kontrolörün performansına yararlı olacaktır. Bu amaç için akıllı teknik desteği sağlamakta fayda vardır.
5. Bu çalışmada trafik sinyal zamanlama kontrolörü için sağlanan tüm kodlar çalıştığımız kavşağa özel yazılmıştır. Her türlü kavşağa kolayca uyarlanabilecek bir çerçeve tasarlamakta ve geliştirmekte faydalı olacaktır.

6. Önceki maddeye ilişkin olarak, bu çalışmada kullanılan tekniklerin ve ayrıca geliştirilmiş sistemlerin farklı uygulamalar için uygun olabileceği de eklenmelidir.
7. Gelecek için verilebilecek en önemli önerilerden biri de, tasarlanan SCATS modeli gibi merkezi kontrol sistemlerini, ortak yazılımlar ile birleştirmektir.
8. Gelecekte, büyüyen verilerle ilgili olarak büyük sorunlar ile karşılaştığımızda, LSTM, CNN ve RNN gibi derin öğrenme algoritmaları uygulayarak bu sorunun üstesinden gelebileceğiz.



6. KAYNAKLAR

1. Iqbal, M., Zahin, A., Islam, Z., and Kaiser, M., Neuro-fuzzy based adaptive traffic flow control system, International Conference on Communications, Devices and Intelligent Systems (CODIS), 2012, pages 349–352.
2. He, J., Hou, Z., Ant colony algorithm for traffic signal timing optimization , Advances in Engineering Software, 2012, 43 pp.14–18.
3. Robertson, D., A Traffic Network Study Tool, Road Research Laboratory, Ministry of Transport, Crowthorne, Berkshire, 1969.
4. Vincent, R., Mitchell, A., and Robertson, D., User Guide to TRANSYT Version 8, Traffic Engineering Department, Transport and Road Research Laboratory, Crowthorne, Berkshire, U.K., LR888, 1980.
5. Hunt, P., Robertson, D., and Bretherton, R., The SCOOT on-line traffic signal optimization technique (Glasgow), Traffic Engineering and Control, vol. 23, no. 4, 1982, pp. 190192,.
6. Luk, J., Two traffic-responsive area traffic control methods: SCATS and SCOOT, Traffic Engineering and Control, vol. 25, no. 1, 1984, pp. 1422,.
7. Cai, C., Adaptive Traffic Signal Control Using Approximate Dynamic Programming, University College London, 2010.
8. Gartner, N., Demand-responsive Decentralized Urban Traffic Control: Part 1, Single-intersection Policies, U.S. Department of Transportation, Office of University Research, 1982, DOT/RSPA/DPB-50/81/24,.
9. Wannige, C. T. and Sonnadara, D. U. J., Adaptive neuro-fuzzy traffic signal control for multiple junctions. In 2009 International Conference on Industrial and Information Systems (ICIIS), 2009, pages262–267. 00008.
10. Gartner, N. H., Kaltenbach, M. H., and Miyamoto, M. M., Demand-responsive Decentralized Urban Traffic Control: Part II, Network Extensions, U.S. Department of Transportation, 1983'a , DOT/RSPA/P-34/85/009,.
11. Jang, J., ANFIS: Adaptive-Network Based Fuzzy Inference System, IEEE Trans On Systems, Man and Cybernetics, No. 3, 1993, vol. 23, pp. 665-685,.
12. Buragohain, M., Mahanta, C., A novel approach for ANFIS modeling based on full factorial design, Applied Soft Computing, 2008, vol8, pp. 609–625,.

13. Mirchandani, P., and Head, L., A real-time traffic signal control system: architecture, algorithms, and analysis, Transportation Research Part C: Emerging Technologies, no. 6, 2001, vol. 9, pp. 415-432, Dec..
14. Spall, J. C., and Chin, D. C., Traffic Responsive Signal Timing for System-Wide Traffic Control, Transportation Research Part C: Emerging Technologies, vol. 5, no. 3-4, 1997, pp. 153-163,.
15. Chin, Y. K., Bolong, N., Kiring, A., Yang, S. S., and Teo, K. T. K., Q-learning based traffic optimization in management of signal timing plan, International Journal of Simulation: Systems, Science and Technology, no. 3, 2011, vol. 12, pp. 2935,.
16. Schouten, S., Reinforcement Learning of Traffic Light Controllers under Partial Observability, Faculty of Science University of Amsterdam, The Netherlands, 2007.
17. Cai, C., Wong, C. K., and Heydecker, B. G., Adaptive traffic signal control using approximate dynamic programming, Transportation Research Part C: Emerging Technologies, vol. 17, no. 5, 2009, pp. 456-474, Oct..
18. Trabia, M. B., Kaseko, M. S., and Ande, M., A twostage fuzzy logic controller for traffic signals, Transportation Research Part C, 1999, pp.353-367,.
19. Li, Y., and Fan, X., Design of signal controllers for urban intersections based on fuzzy logic and weightings, IEEE Conference, 2003, vol.1, pp.867- 871,.
20. Bingham, E., Reinforcement learning in neurofuzzy traffic signal control, European Journal of Operational Research, Vol.131, No.1, pp.232-241.
21. Xiangjun, C., Zhaoxia, Y., A Self-learning Traffic Signal Control Approach and Simulation, Journal of System Simulation, No.7, 2004, Vol.16, pp:1519-1524,.
22. Head, L. K., Event-based short-term traffic flow prediction model, Transportation Research Record, 1995, 1510, pp. 45–52.
23. Aliyari, T., and Khaki, B., and Moghadam, R., Reinforcement Learning of Traffic Light Controllers under Partial Observability, 2008.
24. Williams, B. M., Multivariate vehicular traffic flow prediction: An evaluation of ARIMAX modeling, Transportation Research Record, 2001, 1776, pp 194-200.
25. Daganzo, C. F., Urban Gridlock: Macroscopic Modeling and Mitigation Approaches, Transportation Research Part B: Methodological, 2007, 41: 49–62.
26. Chiou, S. W., Simultaneously Optimizing Link Tolls and Signal Settings in a Road Network.” Applied Mathematical Modelling, 2009, 33 (5): 2311–2323.

27. He, J., and Hou, Z., Ant Colony Algorithm for Traffic Signal Timing Optimization, Advances in Engineering Software, 2012, 43: 14–18.
28. Radhakrishnan, P., and Mathew, T. V., Passenger Car Units and Saturation Flow Models for Highly Heterogeneous Traffic at Urban Signalized Intersections, Transport metric, 2011, 7 (2): 141–162.
29. Cantarella, G. E., Velonà, P., and Vitetta, A., Signal Setting with Demand Assignment: Global Optimization with Dayto-Day Dynamic Stability Constraints.” Journal of Advanced Transportation, 2012, 46 (3): 254–268.
30. Varia, H., Gundaliya, P., and Dhingra, S., Application of Genetic Algorithms for Joint Optimization of Signal Setting Parameters and Dynamic Traffic Assignment for the Real Network Data.” Research in Transportation Economics, 2013, 38: 35–44.
31. Di Gangi, M., Cantarella, G. E., Di Pace, R., and Memoli, S., Network Traffic Control based on a Mesoscopic Dynamic Flow Model, Transportation Research Part C: Emerging Technologies, 2016, 66: 3–26.
32. Marcianò, F. A., Musolino, G., and Vitetta, A., Signal Setting Optimization on Urban Road Transport Networks: The Case of Emergency Evacuation, Safety Science, 2015, 72: 209–220.
33. Lieberman, E., Chang, J., and Prassas, E., Formulation of Real-time Control Policy for Oversaturated Arterials, Transportation Research Record: Journal of the Transportation Research Board, 2000, 77–88.
34. Leclercq, L., Chiabaut, N., and Trinquier, B., Macroscopic Fundamental Diagrams: A Cross-comparison of Estimation Methods, Transportation Research Part B: Methodological, 2014, 62: 1–12.
35. Afandizadeh Zargari, S., Dehghani, N., Mirzahosseini, H., , and Hamed, M., Improving SCATS Operation during Congestion Periods Using Internal/External Traffic Metering Strategy, PROMET-Traffic & Transportation, 2016, 28 (1): 41–47.
36. Afandizadeh Zargari, S., Dehghani, N., Mirzahosseini, H., and Hamed, M., Optimal traffic lights control using meta heuristic algorithms in high priority congested networks, The International Journal of Transportation Research , 2018, 1942-7867 (Print) 1942-7875 (Online) Journal homepage.
37. Jarkko, N., General fuzzy rule base for isolated traffic signal control-rule formulation, Transportation Planning and Technology, 2001, 24:3, 227-247, DOI: 10.1080/03081060108717669.
38. SLIMANI, N., SLIMANI, I., SBITI, N., AMGHAR, M., Traffic forecasting in Morocco using artificial neural networks, The 10th International Conference on Ambient Systems, Networks and Technologies (ANT) , 2019, Leuven, Belgium.

39. Beymer, D., McLauchlan, P., Coifman, B., Malik, J., A Real Time Computer Vision System for Measuring Traffic Parameters, Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1997.
40. Dellart, F., Pomerlain, D., Thorpe, C., Model-based car tracking with a Road Follower, International Conference on Robotics and Automation, May 1998.
41. Smith, S.M., Real-Time Motion Segmentation and shape Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, Number 8, August 1995, Volume 17,.
42. Yang, C., and Wu, C., A robust method on estimation of Lyapunov exponents from a noisy time series, Nonlinear Dynamics, No. 3, 2011, Vol. 64, pp. 279-292.
43. Pengjian, S., Xuwei, A. L., and Santi, K., Chaotic analysis of traffic time series, Chaos, Solutions and Fractals, 2005, Vol. 25, pp.121–128.
44. Yousefpoor, P., Esfahani, M.S., and Nojumi, H., Looking for systematic approach to select chaos tests, Applied Mathematics and Computation, 2008, 198, PP. 73–91.
45. Transportation Research Board, Highway Capacity Manual, Special Edition, 1997.
46. Silcock, J. P.; Sang, A., a phase-based optimisation program for individual signal controlled junctions, Traffic Engineering and Control, 1990, 31 (5), 291-298.
47. Hubschneider, H., Mikroskopisches Simulationssystem für Individualverkehr und Öffentlichen Personennahverkehr, Schriftenreihe des Instituts für Verkehrswesen, Heft 26, Universität (TH), Karlsruhe, 1983,.
48. Wiedemann, R., Simulation des Verkehrsflusses, Schriftenreihe des Instituts für Verkehrswesen, Heft 8, Universität (TU) Karlsruhe, Germany, 1974.
49. Hoyer, R., and Fellendorf, M., Parametrization of Microscopic Traffic Flow Models Through Image Processing, 8th IFAC Symposium on Transportation Systems, Chania, Greece, June 1997,.
50. Fellendorf, M., Public Transport Priority within SCATS – A Simulation Case Study in Dublin. Institute of Transportation Engineers, 67th Annual Meeting, Boston, August 1997,.
51. Prevedrouros, P., Jackson, D., and Wang, Y., Comparison of INTEGRATION, TSIS/CORSIM and WATSim in Replicating Volumes and Speeds on Three Small Networks, Transportation Research Board 77th Annual Meeting, January 1998,.
52. Elefteriadou, L., Leonard, J., Lieu, H., List, G., Beyond the Highway Capacity Manual: A Framework for Selecting Simulation Models in Traffic Operational Analyses. Transportation Research Board 78th Annual Meeting, January 1999,.

53. Panos, D., and Wang, Y., Simulation of a Large Freeway/Arterial Network with CORSIM, INTEGRATION, and WATSim. Transportation Research Board 78th Annual Meeting, January 1999,.
54. Zadeh, A., Fuzzy sets, Information and Control, 1965, Vol. 8, No. 3,.
55. Hans, J., Fuzzy Set Theory- and its Applications, Kluwer Academic Publishers, 1996,.
56. George J., Klir, Bo., Fuzzy Sets and Fuzzy Logic, Theory and Applications Prentice Hall, 1995,.
57. George J., Tina, A., Fuzzy Sets, Uncertainty, and Information, Prentice Hall, 1988,.
58. Bedzek, C., Fuzziness vs. Probability-Again, IEEE Transactions on Fuzzy Systems, No. 1, February 1994, Vol. 2,.
59. Erkki, J., Sumean logiikan mahdol lisuudet, Tekes, 1998,.
60. Kosko, B., Fuzzy Engineering. PrenticeHall, 1997,.
61. Chuen Chien, L., Fuzzy Logicin Control Systems: Fuzzy Logic Controller- Part1, Part2 IEEE Transactionson Systems, Man and Cybernetics, No.2, 1990, Vol. 20,.
62. Mattila, J., Sumean logiikan oppikirja. Johdatus sumeaan matematiikkaan. Art House, 1997,.
63. Niskanen, V., Sumeat järjestelmät. In: Eero Hyvönen, Ilkka Karanta Markku Syrjänen (eds.): Tekoälyn ensyklopedia. Gaudeamus, 1993,.
64. Orlovski, A., Calculus of Decomposable Properties, Fuzzy Sets, and Decisions. Allerton Press, 1994,.
65. Turunen, E., Sumean logiikan matematiikka. Otatieto, 1997,.
66. Yager, R., Filev, D., Essentials of Fuzzy Modelling and Control John Wiley & Sons, 1994,.
67. Niskanen, A., Inexactness, Vagueness, Impreciseness < URL:<http://www.helsinki.fi/~niskanen/vague.html>.
68. Niskanen, A., Uncertainty, Probability, 2002,.
69. Bedzek, C., Fuzzy Models-What Are They and Why?, IEEE Transactions on Fuzzy Systems, February 1993, Vol. 1,.

70. Bedzek, C., The Thirsty Traveler Visits Gamont: A Rejoinder to -Comments on Fuzzy Sets-What Are They and Why?, IEEE Transactions on Fuzzy Systems, Vol. 2, No. 1, February 1994.
71. Kikuchi, S., Pursula, M., Treatment of Uncertainty in Study of Transportation: Fuzzy Set Theory and Evidence Theory, Journal of Transportation Engineering, No. 1, February 1998, Vol. 124,.
72. Shig, J., Jang, R., and Sun, C., Neuro-Fuzzy Modelling and Control Proceedings of the IEEE, Vol. 83, No. 3, March 1995.
73. Berenji, H., Khedkar, P., Learning and Tuning Fuzzy Logic Controllers Through Reinforcements. IEEE Transactions on Neural Networks, No. 5, September 1992 Vol. 3.
74. Sarle, W., Neural Networks and Statistical Models, Proceedings of the 19 th Annual SAS Users Group International Conference, April 1994,.,s
75. Udofia, K., and Emagbetere, J., Development of multi-agent ANFIS-based model for urban traffic signal control, In 2013 International Conference on Connected Vehicles and Expo (ICCVE), 2013, pages 662–669.
76. Spall, J.C., and Chin, D. C., Traffic-responsive signal timing for system-wide traffic control, Transportation Research Part C: Emerging Technologies, 1997, 5(34):153– 163.
77. Chin, Y., Bolong, N., Kiring, A., Yang, S., and Teo, K., Q-learning based traffic optimization in management of signal timing plan, International Journal of Simulation: Systems, Science and Technology, 2011, 12(3):29–35.
78. Kohonen, T., Self- Organization and Associative Memory, Springer-Verlag 3rd edition, 1989,.
79. Kohonen, T., The Self- Organizing Map, Proceedings of the IEEE 78, 1990, pp. 1464-1480,.
80. Mäkisara, K., Simula, O., Kangas, J., Artificial Neural Networks Proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN91), 1991, 24-28, Vol. 2,.
81. Sarle, W.S., Neural Network and Statistical Jargon URL: <ftp://ftp.sas.com/pub/neural/jargon.txt>,.
82. Cichocki, A., Unbehauen, R., Neural Networks for Optimization and Signal Processing, 1993,.
83. Nielsen, R., Neuro computing, Addison-Wesley, 1990,.

84. Hertz, J., Krogh, A., Palmer, R. G., Introduction to the Theory of Neural Computation., Addison-Wesley, 1991,.
85. Holmström, L., Kohonen, T., Hyvönen, E., and Karanta, I., and Syrjänen, M., Tekoälyn ensyklopedia, 1993, Gaudeamus,.
86. Rumelhart, D.E., Hinton, Williams, R.J., Learning Internal Representations by Error Propagation, Explorations in the Microstructure of Cognition I. MIT Press, 1986,.
87. Kaufmann, A., Gupta, M.M., Fuzzy Mathematical Models in Engineering and Management Science, Elsevier Science Publishers, 1988,.
88. Brown, M., Harris, C. J., A Perspective and Critique of Adaptive Neuro Fuzzy Systems Used for Modelling and Control Applications, International Journal of Neural Systems, No. 2, June 1995, Vol. 6,.
89. Lin, C.T., Lee, C.S., Neural-Network- Based Fuzzy Logic Control and Decision System, IEEE Transactions on Computers, No.12, December 1991, Vol.40,.
90. Keller, J.M., Hunt, D.J., Incorporating Fuzzy Membership Functions into the Perceptron Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, No.6, November 1985, Vol. 7,.
91. Hayashi, Y., Buckley, J.J., Czogala, E., Fuzzy Neural Network with Fuzzy Signals and Weights, International Journal of Intelligent Systems, Vol. 8 No. 4, 1993,.
92. Horikawa, S., Furuhashi, T., Uchikawa, Y., On Fuzzy Modelling Using Fuzzy Neural Networks with the Back-Propagation Algorithm, IEEE Transactions on Neural Networks, No. 5, September 1992, Vol. 3,.
93. Shing, J., Jang, R., Adaptive-Network-Based Fuzzy Inference Systems, IEEE Transactions on Systems, Man and Cybernetics, May/June 1993, No.3, Vol.23,.
94. Jang, J., Adaptive network-based fuzzy inference systems, IEEE Trans Sys Man Cybern, 23: 665–685.
95. Kar, S., Das, S., Ghosh, P.K., Applications of neuro fuzzy systems: a brief review and future outline, Appl Soft Compute, 2014,15: 243–259.
96. Ali, M., Ghatol, A., A neuro-fuzzy inference system for student modeling in web-based intelligent tutoring systems, Proceedings of international conference on cognitive systems, 2004, pp 14–19.
97. Babazade Khameneh, N., Arabalibeik, H., Salehian, P., and Setayeshi, S., Abnormal red blood cells detection using adaptive neuro-fuzzy system, Studies in health technology and informatics, pp 30–34,.

98. Fang, H., Adaptive neurofuzzy inference system in the application of the financial crisis forecast., Int J Innov Manag Technol, 2012, 3: 250,.
99. Chakrapani, Y., Soundararajan, K., Adaptive neuro-fuzzy inference system based fractal image compression, Department of Electronic and Communications, JNTU College of Engineering, Hyderabad, 2009, p 2,.
100. Patil, S., Mandal, S., Hegde, A., Alavandar, S., Neuro-fuzzy based approach for wave transmission prediction of horizontally interlaced multilayer moored floating pipe breakwater, Ocean Eng , 2011,38: 186–196.
101. Topalov, A.V., Kayacan, E., Oniz, Y., Kaynak, O., Adaptive neuro-fuzzy control with sliding mode learning algorithm application to a tilock braking system, Asian Control Conference, 2009, ASCC2009. 7th, IEEE, pp 784–789.
102. Dastranj, M.R., Ebrahimi, E., Changizi, N., Sameni, E., Control DC motorspeed with adaptive neuro-fuzzy control (ANFIS), Aust J Basic Application, 2011, Sci 5:1499–1504.
103. Cheng, C.B., Lee, E.S., Applying fuzzy adaptive network to fuzzy regression analysis, Compute Math Application, 1999, 38(2):123–140.
104. Cai, C.H., Du, D., Liu, Z.Y, Battery state-of-charge (SOC)estimation using adaptive neuro-fuzzy inference system (ANFIS), IEEE International Conference on Fuzzy Systems, 2003,pp 1068–1073,.
105. Guney, K., Sarikaya, N., Adaptive neuro-fuzzy inference system for the input resistance computation of rectangular microstrip antennas with thin and thick substrates, Electromagn Waves Appl, 2004, 18:23–39. <https://doi.org/10.1163/156939304322749599>.
106. Guney, K., Sarikaya, N., Computation of Resonant Frequency for Equilateral Triangular Microstrip Antennas Using the Adaptive Neuro-Fuzzy Inference System, Int J RF Microwave Compute-Aided Eng, 2004b, 14: 134–143. <https://doi.org/10.1002/mmce.10125>.
107. Guney, K., Sarikaya, N., Input resistance calculation for circular microstrip antennas using adaptive neuro-fuzzy inference system, Int J Infrared Millimeter Waves, 2004c, 25: 703–716. <https://doi.org/10.1023/B:IJIM.0000020756.48454.31>.
108. Entchev, E., Yang, L., Application of adaptive neuro-fuzzy inference system techniques and artificial neural networks to predict solid oxide fuel cell performance in residential microgeneration installation, J Power Sources, 2007, 170:122–129. <https://doi.org/10.1016/j.jpowsour.2007.04.015>.
109. Tabesh, M., Dini, M., Fuzzy and neuro-fuzzy models for short-term water demand forecasting in Tehran, Iran J Sci Technol, Transaction B, 2009, Eng 33: 61–77.

110. Yildiz, A., Akin, M., Poyraz, M., Kirbas, G., Application of adaptive neuro-fuzzy inference system for vigilance level estimation by using wavelet-entropy feature extraction, Expert Syst Appl, 2009, 36: 7390–7399. <https://doi.org/10.1016/j.eswa.2008.09.003>,.
111. Dalkilic, T.E., Apaydin, A., A fuzzy adaptive network approach to parameter estimation in cases where independent variables come from an exponential distribution, J Comput Appl Math, 2009, 233(1):36–45.
112. Kavitha, S., Duraiswamy, K., Adaptive neuro-fuzzy inference system approach for the automatic screening of diabetic retinopathy in fundus images, J Comput, 2011, Sci7:1020–1026.<https://doi.org/10.3844/jcssp.2011.1020.1026>.
113. Rajasekaran, M.P., Meena, R.S., Application of adaptive neuro-fuzzy inference systems for MR image classification and tumour detection, Int J Biomed EngTechnol, 2012, 9:133–146.<https://doi.org/10.1504/IJBET.2012.047746>.
114. Karami, A., Rezaei, E., Rahimi, M., Khani, S., Modeling of heat transfer in an air cooler equipped with classic twisted tape inserts using adaptive neuro-fuzzy inference system, Chem Eng Commun, 2013, 200:532– 542. <https://doi.org/10.1080/00986445.2012.715105>.
115. Taghavifar, H., Mardani, A., On the modeling of energy efficiency indices of agricultural tractor driving wheels applying adaptive neuro-fuzzy inference system, J Terramech, 2014, 56: 37–47. <https://doi.org/10.1016/j.jterra.2014.08.002>.
116. Bahadori, A., Baghban, A., Bahadori, M., Lee, M., Ahmad, Z., Zare, M., Abdollahi, E., Computational intelligent strategies to predict energy conservation benefits in excess air controlled gas-fired systems, Appl Therm Eng, 2016, 102:432–446. <https://doi.org/10.1016/j.applthermaleng.2016.04.005>.
117. Kabini, K., Review of ANFIS and its application in control of machining processes, 2011, Sustain Res Innov Proc 3.
118. Salleh M.N.M., Hussain, K., A review of training methods of ANFIS for applications in business and economics, Int J u- e-Serv Sci Technol,9:165–172,.
119. Walia, N., Kumar, S., Singh H (2015a) A survey on applications of adaptive neuro fuzzy inference system, Int J Hybrid Inf Technol ,2016, 8:343–350.
120. Walia, N., Singh, H., Sharma, A., ANFIS: adaptive neuro-fuzzy inference system—a survey, Int J Comput Appl, 2015b, 123:32–38,.
121. Niittymaki, J., Kosonen, T., and Nevala, R., Fuzzy traffic signal control in major arterials, in IEEE Conference on Intelligent Transportation Systems, 2001, pp. 346–351,.

122. Johnand, R., Czarnecki, C., Atype2 adaptive fuzzy inferencing system, in Systems, Man, and Cybernetics, 1998 IEEE International Conference on. IEEE, 1998, vol. 2, pp. 2068–2073,.
123. Karnik, N.N., and Mendel, J.M., Introduction to type-2 fuzzy logic systems, in Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on. IEEE, 1998, vol. 2, pp. 915– 920,.
124. Richard, S., Andrew, G., J., Reinforcement Learning is Adaptive Optimal Control, IEEE Control Systems, No.2, April 1992, Vol.12,.
125. Richard, S., Andrew, G., Reinforcement Learning. Course Notes.
126. Modern Sezgisel Teknikler ve Uygulamaları, Papatya Yayınları, İstanbul, 14-15.
127. Karasulu, B., Balli, S., Image segmentation using fuzzy logic, neural networks and genetic algorithms: survey and trends, Mach Graph Vis Int J , 2010, 19: 367–409.
128. Jurado, F., Ortega, M., Carpio, J., Power quality enhancement in fuel cells using genetic algorithms and ANFIS architecture. In: IEEE international symposium on industrial electronics, 2006, pp 757–762. [https:// doi.org/10.1109/ISIE.2006.295729](https://doi.org/10.1109/ISIE.2006.295729).
129. Newell, A., Unified Theories of Cognition, MA: Harvard University Press, Cambridge, 1990.
130. Gökay, G., TAŞKIN, E., ve TAŞKIN, Ç., Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXI, sayı 1, 2002, s. 129-152.
131. Karaboga, D., Design of Fuzzy Logic Controllers using Tabu Search algorithm, Biennial Conf, of the North American Fuzzy Information Processing Society- NADIPS, 1996, Berkeley, California, USA, SAYFA 489-491.
132. Fleurent, C., ve Ferland, J., Genetic Hybrids for the Quadratic Assignmeny problem, Amer. Math, Soc., 1994.
133. Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989,.
134. İşlier, A.A., Üretim Hücrelerinin Bir Genetik Algoritma Kullanılarak Oluşturulması, Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, Cilt: 2 , 2001, Sayı: 1, s. 137-157.
135. Cárdenas, J.J., García, A., Romeral, J.L., Kampouropoulos, K., Evolutive ANFIS training for energy load profile forecast for an IEMS in an automated factory,

- IEEE international conference on emerging technologies and factory automation, ETFA, 2011, <https://doi.org/10.1109/ETFA.2011.6059079>.
136. Jang, J.S.R., *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, 1997, s. 173-196.
 137. Akachukwu, C.M, Aibinu, A.M., Nwohu, M.N., and Salau, H.B., A decade survey of engineering applications of genetic algorithm in power system optimization. 2014 5th international conference on intelligent systems, modelling and simulation (ISMS), IEEE, 2014, pp 38–42.
 138. Bhatt, N., Chauhan, N.R., Genetic algorithm applications on Job Shop Scheduling Problem: a review, 2015 international conference on soft computing techniques and implementations (ICSCTI), IEEE, 2015, pp 7–14.,
 139. Yeo, M.F., ve Agyel, E.O., Optimising Engineering Problems Using Genetic Algorithms, Engineering Computations, 1996, Volume: 15, Number: 2, s. 268-280.,
 140. Engin, O., Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu, İTÜ Fen Bilimleri Enstitüsü, Yayınlanmamış Doktora Tezi, 2001.,
 141. Yeniay, Ö., An Overview of Genetic Algorithms, *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, Cilt: 2, 2001, Sayı: 1, s. 37-49.,
 142. Foy, M.D., Benekohal, R.F., and Goldberg, D.E., Signal Timing Determination Using Genetic Algorithms, Transportation Research Record , 2001, 1365, pp.108- 115, 1992.
 143. Horn, J., Nafpliotis, N., Goldberg, D. E., A Niche Pareto Genetic Algorithm for Multiobjective Optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, 1994.
 144. Makinen, R.A.E., Toivanen, J., and Piaux, J., Multidisciplinary Shape Optimization in Aerodynamics and Electromagnetics Using Genetic Algorithms, International Journal for Numerical Methods in Fluids, 1999, v.30, pp.149-159,
 145. Fırlı, A., ve Engin, O., Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu, İTÜ Dergisi, 2002, s. 1-6.
 146. Eddy, J., Lewis, K., Effective Generation of Pareto Sets Using Genetic Programming. *Proceedings of DETC2001, ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2001.

147. Park, B., Hlesser, C.J. and Urbanik, L. Traffic Signal Optimization Program for Oversaturated Conditions, Genetic Algorithm Approach, Damportation Research Record 1683, 1999, pp.133-142, Nov.
148. Thrift, P. R., Fuzzy logic synthesis with genetic algorithms. Proceedings of the 4th International Conference on Genetic Algorithms (ICGA'91), 1991, San Diego, CA, 509–513.
149. Yuan, Y., and Zhuang, H., A Genetic Algorithm for Generating Fuzzy Classification Rules, Fuzzy Sets and Systems, 1996, 18 (1): 1–19,.
150. Deb, K., and Agrawal, S., Understanding interactions among genetic algorithm parameters, *Foundations of Genetic Algorithms V*, W. Banzhaf and C. Reeves, Eds. San Mateo, CA: Morgan Kauffman, 1998, pp. 265–286,.
151. Deb, K., *Multiobjective Optimization Using Evolutionary Algorithms*, Chichester, 2001, U.K.: Wiley,.
152. Harik, G., Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms, Illinois Genetic Algorithms Lab., Univ. Illinois at Urbana-Champaign, Urbana, IL, IlliGAL Rep. 97005, 1997,.
153. Fung R.Y.K., Tang, J., ve Wang, D., Extension of A Hybrid Genetic Algorithm for Nonlinear Programming Problems with Equality and Inequality Constraints, Computers & Operations Research, 2001, Volume: 29 Issue: 3, s. 261-274.
154. Altıparmak, F., Dengiz, B., ve Smith, A.E., An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks *Transactions on Operational Research*, 2000, Volume: 12, Number: 1-2, s. 57-75,.
155. Ergül, E., Çok amaçlı genetik algoritmalar: temelleri ve uygulamaları, Ankara: Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 2010,.
156. Shi, Y., Ve Eberhart, R.C., A modified particle swarm optimizer, Proceedings of the IEEE International Conference on Evolutionary Computation, 1998, Anchorage, 69-73
157. Medina, A.J.R., Pulido, G.T., and Torres, J.G.R., A Comparative study of Neighbourhood for Particle Swarm Optimizers, Proceedings of the International Joint Conference on Computational Intelligence, Funchal, Maderia, Portugal, 2009,.
158. Tunchan, C., Particle Swarm Optimization Approach to Portfolio Optimization, *Nonlinear Analysis Real World*, 2009, 10, 2396-2406.
159. sakhavati, A., PSO Algorithm an overview.

160. Hakwood, S., And Wang, J., The Engineering of Cellular robotic systems, Proceedings of Institute of Electrical and Electronics Engineers International Symposium, Arlington, USA, 1988,.
161. Perez, R.E., Behdian, K., Particle Swarm Approach for Structural Design Optimization, International Journal of Advanced Robotic Systems, First Edition, September 2007,.
162. Kirkpatrick, S., Gelatt, C., and Vecchi, M., Optimization by Simulated Annealing, Science, 1983, Vol.220,4598,671-680.
163. Lee, K.S., WooGie, J., A new structural optimization method based on the harmony search algorithm, Computers & Structures Volume 82, April 2004, Issues 9–10, Pages 781-798.
164. Geem, Z.W., Harmony Search Algorithm for Solving Sudoku. In: Apolloni B., Howlett, R.J., Jain, L., (eds) Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, 2007vol 4692, Springer, Berlin, Heidelberg.
165. Lee, K.S., Geem, Z.W., A new structural optimization method based on the harmony search algorithm, Computers & Structures Volume 82, Issues 9–10, April 2004, Pages 781-798.
166. Wang, X., Gao, X.Z., and Ovaska, S.J., Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems, International Journal of Bio-Inspired Computation, December 20081(1/2):80-88,.
167. Alatas, B., Chaotic harmony search algorithms, Applied Mathematics and Computation, 1 July 2010, Volume 216, Issue 9, Pages 2687-2699.
168. Lee, K.S., Geem, Z.W., A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering, volume 194, Issues 36–38, 23 September 2005, Pages 3902-3933.
169. Wang, X., Gao, X.Z., Zenger, K., An Introduction to Harmony Search Optimization Method, Springer, 2015,.
170. Ingram, G., Zhang, T., Overview of Applications and Developments in the Harmony Search Algorithm, Springer-Verlag, Berlin Heidelberg, 2009, pp.15–37. Ch.2.
171. Lee, K.S., Geem, Z.W., A new structural optimization method based on the harmonysearch algorithm, Comput, 2004, Struct.82781–798.
172. Ingram, G., Zhang, T., An introduction to the harmony search algorithm, Vol.2 of Advances in Process Systems Engineering, World Scientific, Singapore, 2010, pp.301–335.Ch.9.

173. Coello, C.C.A., Recent Trends in Evolutionary Multiobjective Optimization, Evolutionary Multiobjective Optimization, 2005, 7-32,.
174. Abido, M.A., Multiobjective particle swarm for environmental/economic dispatch problem, International Power Engineering Conference, 2007, 1385-1390,.
175. Goldberg, D.E., Genetic Algorithms in search, Optimization and Machine learning, Addison Wesley Longman, Inc. Boston, 1998,.
176. Deb, K., and Goldberg, D.E., An investigation of niche and species formation in genetic function optimization, Third international conference on Genetic algorithms, 1989,42-50,.
177. C.M., Fonseca and P.J., Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest, Ed. San Mateo, CA: Morgan Kauffman, 1993, pp. 416–423.
178. Horn, J., Nafploitis, N., and Goldberg, D.E., A niched Pareto genetic algorithm for multiobjective optimization, In Proceedings of the First IEEE Conference on Evolutionary Computation, Michalewicz, Z., Ed. Piscataway, NJ: IEEE Press, 1994, pp. 82–87.
179. Srinivas, N., and Deb, K., Multiobjective function optimization using nondominated sorting genetic algorithms, Fall 1995, Evol Comput, vol. 2, no. 3, pp. 221–248,,
180. Zitzler, E. and Thiele, L., Multiobjective optimization using evolutionary algorithms—A comparative case study, In Parallel Problem Solving From Nature, V., Eiben, A.E., Bäck, T., Schoenauer, M., and Schwefel, H.P., Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 292–301.
181. Rudolph, G., Evolutionary search under partially ordered sets, Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67/99, 1999,.
182. Senthilkumar, C., Ganesan, G., Karthikeyan, R., Optimization of ECM Process Parameters Using NSGA-II, Journal of Minerals and Materials Characterization and Engineering, 2012, 11, 931-937 Published Online October 2012.
183. Arivoli, R., and Chidambaram, I.A., Design of multi-objective particle swarm optimization (MOPSO) based controller for load-frequency control of power systems inter connected with elastic and stiff tie-lines, International Journal of Latest Trends in Computing, 2 no,2011, 541-554,.
184. K., Deb., Current trends in evolutionary multi-objective optimization, International Journal for Simulation and Multidisciplinary Design Optimization, 1 no. 12007, 1-8,.

185. Carlos, A., David, A., Veldhuizen, V., and Lamont, G.B., *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Boston, 2002. ISBN 0-3064-6762- 3,.
186. Joshua, D., Knowles and David, W., Corne. Approx- imating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionay Compu- tation, 8(2):149-172, 2000,.
187. Carlos, A., and Pulido, G., Multiobjective Optimization using a Micro- Genetic Algorithm, In Lee S., Erik, D., Annie, W.u., Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Euo- lutionay Computation Conference (GECCO'ZOOof)*, pages 274-282, San Francisco, California, 2001, Morgan Kaufmann Publishers,.
188. Kalyanmoy, Deb., Samir, A., Pratab, A., and Meyarivan, T., A Fast Elitist Non- Dominated Sorting Genetic Algorithm for Multi-Objective Op- timization: NSGA-11. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 2000, pages 849-858. Springer,
189. Kalyanmoy, Deb., and David, E., Goldberg. An In- vestigation of Niche and Species Formation in Genetic Function Optimization, In David, J., SchafTer, editor, Proceedings of the Third International Conference on Genetic Algorithms, June 1989, pages 42-50, San Mateo, California, George Mason Uni- versity, Morgan Kaufmann Publishers,.
190. Blasi, L., Barbato, S., and Mattei, M., A particle swarm approach for flight path optimization in a constrained environment, Aerosp. Sci. Technol, 26 no. 1 (2012) 128-137. doi: <http://dx.doi.org/10.1016/j.ast.2012.02.021>.
191. Janson, S., Merkle, D., and M., Middendorf, Molecular docking with multi- objective Particle Swarm Optimization, Applied Soft Computing, 8 no. 1, 2007 666-675,.
192. Rao, R.V., Pawar, P.J., and Shankar, R., Multi-objective optimization of electrochemical machining process parameters using a particle swarm optimization algorithm, Journal of Engineering Manufacture, 222 no. 8, 2008, 949-958.]
193. Baltar, A.M., and Fontane, D.G., A generalized multiobjective particle swarm optimization solver for spreadsheet models: application to water quality, *Hydrology Days*, (2006) 1–12.
194. Carvalho, A.B., and Pozo, A., Non-ordered data mining rules through multi- objective particle swarm optimization: dealing with numeric and discrete attributes, IEEE Conference on Hybrid Intelligent Systems, 2008, 495-500.

195. Wang, L., and Singh, C., Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm, Electric Power Systems Research, 77 no, 2006, 1654-1664,.
196. Goudos, S.K., and Sahalos, J.N., Microwave absorber optimal design using multi-objective particle swarm optimization, Microwave and Optical Technology Letters, 48 no, 2006, 1553-1558,.
197. Liu, Y., Wang, H., Liu Z., and Zhao, X., Land Use Zoning at the County Level Based on a Multi-Objective Particle Swarm Optimization Algorithm: A Case Study from Yicheng, China, Int. J. Environ. Res. Public Health, 9 no. 2012, 2801-2826.
198. Karpat, Y., and Ozel, T., Multi-objective optimization for turning processes using neural network modeling and dynamic-neighborhood particle swarm optimization, Int. j. Adv. Manuf. Technol, 35 no. 3-4 2007, 234-247.
199. Kennedy, J., and Eberhart, R.C., Particle swarm optimization, IEEE International Conference on Neural Network, 1995,1942-1948,.
200. Masehian, E., and Sedighizadeh, D., A multi-objective PSO-based algorithm for robot path planning, IEEE Conference on Industrial Technology, 2010, 465-470.
201. Carvalho, A.B., Pozo, A., Vergilio, S., and Lenz, A., Predicting fault proneness of classes trough a multiobjective particle swarm optimization algorithm, 20th IEEE International conference on tools with artificial intelligence, 2008, 387-394.

Ek 1'in devamı

SABAH

\$MOVEMENTEVALUATION:SIMRUN	TIMEINT	MOVEMENT\DIRECTIO N	MOVEMENT	QLEN	QLENMAX	VEHS(ALL)	PERS(ALL)	VEHDELAY(ALL)	PERSDELAY(ALL)	STOPDELAY(ALL)	STOPS(ALL)
1	0-600	S-W	7-2@41.4-9@67.3	3,27	32,69	24	24	11,35	11,35	8,59	0,25
1	0-600	S-N	7-2@41.4-37@76.2	3,27	32,69	211	211	1,95	1,95	0,81	0,03
1	0-600	N-W	7-5@32.4-9@67.3	21,1	72,4	37	37	19,1	19,1	11,15	0,62
1	0-600	N-S	7-5@32.4-36@40.8	21,1	72,4	165	165	21,04	21,04	12,35	0,67
1	0-600	E-W	7-8@505.2-9@67.3	5,35	26,41	26	26	14,11	14,11	8,49	0,62
1	0-600	E-S	7-8@505.2-36@40.8	5,35	26,41	23	23	14,93	14,93	11,06	0,43
1	0-600	E-N	7-8@505.2-37@76.2	5,35	26,41	2	2	6,24	6,24	2,5	0,5
1	0-600	Total	7	9,91	72,4	488	488	11,45	11,45	6,78	0,35
1	600-1200	S-W	7-2@41.4-9@67.3	13,41	49,84	42	42	37,27	37,27	30,23	0,71
1	600-1200	S-N	7-2@41.4-37@76.2	13,41	49,84	244	244	3,56	3,56	1,56	0,07
1	600-1200	N-W	7-5@32.4-9@67.3	33,02	106,9	53	53	31,5	31,5	19,7	0,92
1	600-1200	N-S	7-5@32.4-36@40.8	33,02	106,9	157	157	28,6	28,6	18,15	0,83
1	600-1200	E-W	7-8@505.2-9@67.3	4,99	26,84	20	20	11,75	11,75	7,67	0,45
1	600-1200	E-S	7-8@505.2-36@40.8	4,99	26,84	27	27	18,08	18,08	13,26	0,59
1	600-1200	E-N	7-8@505.2-37@76.2	4,99	26,84	1	1	0,4	0,4	0	0
1	600-1200	Total	7	17,14	106,9	544	544	17,13	17,13	11,13	0,46
1	1200-1800	S-W	7-2@41.4-9@67.3	9,16	31,72	27	27	32,62	32,62	27,24	0,67
1	1200-1800	S-N	7-2@41.4-37@76.2	9,16	31,72	228	228	4,02	4,02	1,86	0,08
1	1200-1800	N-W	7-5@32.4-9@67.3	24,93	86,61	60	60	24,87	24,87	15,47	0,75
1	1200-1800	N-S	7-5@32.4-36@40.8	24,93	86,61	137	137	23,16	23,16	14,89	0,73
1	1200-1800	E-W	7-8@505.2-9@67.3	7,16	32,48	33	33	17,22	17,22	11,44	0,58
1	1200-1800	E-S	7-8@505.2-36@40.8	7,16	32,48	22	22	14,46	14,46	9,56	0,55
1	1200-1800	E-N	7-8@505.2-37@76.2	7,16	32,48	0	0				
1	1200-1800	Total	7	13,75	86,61	507	507	14,49	14,49	9,3	0,42
1	1800-2400	S-W	7-2@41.4-9@67.3	6,69	26,22	28	28	25,56	25,56	20,51	0,57
1	1800-2400	S-N	7-2@41.4-37@76.2	6,69	26,22	269	269	3,64	3,64	1,38	0,06
1	1800-2400	N-W	7-5@32.4-9@67.3	26,19	78,21	48	48	27,31	27,31	17,16	0,88
1	1800-2400	N-S	7-5@32.4-36@40.8	26,19	78,21	150	150	23,27	23,27	14,99	0,73
1	1800-2400	E-W	7-8@505.2-9@67.3	5,09	31,56	34	34	17,77	17,77	12,68	0,59
1	1800-2400	E-S	7-8@505.2-36@40.8	5,09	31,56	16	16	19,15	19,15	14,31	0,69
1	1800-2400	E-N	7-8@505.2-37@76.2	5,09	31,56	0	0				
1	1800-2400	Total	7	12,65	78,21	545	545	13,59	13,59	8,58	0,4
1	2400-3000	S-W	7-2@41.4-9@67.3	8	37,13	27	27	30,17	30,17	25,22	0,67
1	2400-3000	S-N	7-2@41.4-37@76.2	8	37,13	226	226	3,97	3,97	1,86	0,09
1	2400-3000	N-W	7-5@32.4-9@67.3	30,86	117,73	46	46	31,94	31,94	20,86	0,85
1	2400-3000	N-S	7-5@32.4-36@40.8	30,86	117,73	159	159	26,37	26,37	17,21	0,81
1	2400-3000	E-W	7-8@505.2-9@67.3	5,48	30,55	20	20	10,99	10,99	7,49	0,35
1	2400-3000	E-S	7-8@505.2-36@40.8	5,48	30,55	18	18	22,75	22,75	17,38	0,72
1	2400-3000	E-N	7-8@505.2-37@76.2	5,48	30,55	0	0				
1	2400-3000	Total	7	14,78	117,73	496	496	16,14	16,14	10,6	0,46
1	3000-3600	S-W	7-2@41.4-9@67.3	4,99	19,11	23	23	17,83	17,83	14,07	0,39
1	3000-3600	S-N	7-2@41.4-37@76.2	4,99	19,11	237	237	3,38	3,38	1,66	0,07
1	3000-3600	N-W	7-5@32.4-9@67.3	25,72	85,16	43	43	27,37	27,37	17,52	0,72
1	3000-3600	N-S	7-5@32.4-36@40.8	25,72	85,16	150	150	23,86	23,86	15,84	0,7
1	3000-3600	E-W	7-8@505.2-9@67.3	4,2	26,23	25	25	13,98	13,98	8,79	0,44
1	3000-3600	E-S	7-8@505.2-36@40.8	4,2	26,23	15	15	16,01	16,01	11,38	0,67
1	3000-3600	E-N	7-8@505.2-37@76.2	4,2	26,23	1	1	33,55	33,55	28,7	1
1	3000-3600	Total	7	11,64	85,16	494	494	13,34	13,34	8,63	0,37

Ek 1'in devamı

ÖĞLEN

TIMEINT	MOVEMENT\ DIRECTION	MOVEMENT	QLEN	QLENMAX	VEHS(ALL)	PERS(ALL)	VEHDELAY(ALL)	PERSDELAY(ALL)	STOPDELAY(ALL)	STOPS(ALL)
0-600	S-W	7-2@41.4-9@67.3	9,52	43,94	33	33	33,66	33,66	25,27	1
0-600	S-N	7-2@41.4-37@76.2	9,53	43,95	153	153	5,72	5,72	2,58	0,22
0-600	N-W	7-5@32.4-9@67.3	13,06	61,67	39	39	14,88	14,88	8,21	0,54
0-600	N-S	7-5@32.4-36@40.8	13,06	61,67	177	177	13,38	13,38	6,55	0,54
0-600	E-W	7-8@505.2-9@67.3	16,28	44,59	39	39	30,9	30,9	22,06	0,87
0-600	E-S	7-8@505.2-36@40.8	16,28	44,59	44	44	29,77	29,77	21,01	0,86
0-600	E-N	7-8@505.2-37@76.2	16,28	44,59	4	4	24,23	24,23	16,85	1
0-600	Total	7	12,1	61,67	489	489	15,43	15,43	9,32	0,53
600-1200	S-W	7-2@41.4-9@67.3	15,84	60,48	48	48	35,05	35,05	26,36	1
600-1200	S-N	7-2@41.4-37@76.2	15,85	60,5	154	154	11,58	11,58	6,33	0,39
600-1200	N-W	7-5@32.4-9@67.3	15,34	62,76	58	58	16,79	16,79	8,07	0,64
600-1200	N-S	7-5@32.4-36@40.8	15,34	62,76	163	163	14,18	14,18	7,14	0,53
600-1200	E-W	7-8@505.2-9@67.3	12,4	52,15	46	46	23,73	23,73	15,97	0,7
600-1200	E-S	7-8@505.2-36@40.8	12,4	52,15	34	34	27,97	27,97	20,66	0,85
600-1200	E-N	7-8@505.2-37@76.2	12,4	52,15	2	2	37,52	37,52	29,95	1
600-1200	Total	7	14,86	62,76	505	505	17,56	17,56	10,63	0,58
1200-1800	S-W	7-2@41.4-9@67.3	10,6	56,01	36	36	34,67	34,67	26,99	1
1200-1800	S-N	7-2@41.4-37@76.2	10,61	56,03	169	169	7,21	7,21	3,59	0,27
1200-1800	N-W	7-5@32.4-9@67.3	11,94	58,66	57	57	12,95	12,95	6,34	0,46
1200-1800	N-S	7-5@32.4-36@40.8	11,94	58,66	161	161	11,92	11,92	5,62	0,5
1200-1800	E-W	7-8@505.2-9@67.3	20,35	71,05	53	53	33,74	33,74	23,85	1,02
1200-1800	E-S	7-8@505.2-36@40.8	20,35	71,05	38	38	35,73	35,73	27,02	0,95
1200-1800	E-N	7-8@505.2-37@76.2	20,35	71,05	2	2	21,4	21,4	15	1
1200-1800	Total	7	13,37	71,05	516	516	16,11	16,11	10,01	0,54
1800-2400	S-W	7-2@41.4-9@67.3	8,48	37,15	35	35	31,99	31,99	23,6	1,03
1800-2400	S-N	7-2@41.4-37@76.2	8,49	37,16	160	160	3,88	3,88	1,3	0,16
1800-2400	N-W	7-5@32.4-9@67.3	12,03	50,96	50	50	12,09	12,09	5,91	0,44
1800-2400	N-S	7-5@32.4-36@40.8	12,03	50,96	159	159	13,76	13,76	6,84	0,54
1800-2400	E-W	7-8@505.2-9@67.3	12,31	40,31	45	45	22,28	22,28	15,25	0,67
1800-2400	E-S	7-8@505.2-36@40.8	12,31	40,31	30	30	33,74	33,74	25,27	1,03
1800-2400	E-N	7-8@505.2-37@76.2	12,31	40,31	4	4	20,53	20,53	16,23	0,75
1800-2400	Total	7	10,33	50,96	483	483	13,73	13,73	8,13	0,48
2400-3000	S-W	7-2@41.4-9@67.3	12,07	65,59	33	33	32,44	32,44	24,29	1
2400-3000	S-N	7-2@41.4-37@76.2	12,08	65,61	187	187	8,89	8,89	4,79	0,35
2400-3000	N-W	7-5@32.4-9@67.3	14,51	76,51	51	51	15,09	15,09	5,94	0,57
2400-3000	N-S	7-5@32.4-36@40.8	14,51	76,51	172	172	12,95	12,95	6,18	0,52
2400-3000	E-W	7-8@505.2-9@67.3	15,22	68,61	51	51	27,06	27,06	17,75	0,78
2400-3000	E-S	7-8@505.2-36@40.8	15,22	68,61	36	36	32,39	32,39	23,51	1
2400-3000	E-N	7-8@505.2-37@76.2	15,22	68,61	3	3	9,12	9,12	4,27	0,67
2400-3000	Total	7	13,47	76,51	533	533	15,58	15,58	9,06	0,55
3000-3600	S-W	7-2@41.4-9@67.3	8,34	31,08	34	34	34,05	34,05	26,56	1
3000-3600	S-N	7-2@41.4-37@76.2	8,36	31,09	144	144	5,06	5,06	1,87	0,22
3000-3600	N-W	7-5@32.4-9@67.3	13,8	63,1	37	37	15,92	15,92	8,62	0,54
3000-3600	N-S	7-5@32.4-36@40.8	13,8	63,1	154	154	15,16	15,16	8,04	0,59
3000-3600	E-W	7-8@505.2-9@67.3	12,28	44,08	47	47	25,42	25,42	18,26	0,79
3000-3600	E-S	7-8@505.2-36@40.8	12,28	44,08	29	29	30,98	30,98	23,02	0,93
3000-3600	E-N	7-8@505.2-37@76.2	12,28	44,08	1	1	4,26	4,26	0	0
3000-3600	Total	7	10,7	63,1	446	446	15,49	15,49	9,54	0,54

Ek 1'in devamı

AKŞAM

TİMET	QLEN	QLENMAX	VEHŞ(ALL)	PERS(ALL)	VEHDELAY(ALL)	PERSDELAY(ALL)	STOPPELAY(ALL)	STOPPE(ALL)	EMISSIONSCO	EMISSIONSNOX	EMISSIONSVOC	FUELCONSUMPTION
0-600	8,09	38,07	49	49	25,14	25,14	20,03	0,51	48,463	9,439	11,232	0,693
0-600	8,11	38,1	134	134	1,82	1,82	0,32	0,04	60,223	11,717	13,957	0,862
0-600	23,41	96,02	41	41	13,34	13,34	8,92	0,59	39,146	7,616	9,073	0,56
0-600	23,41	96,02	238	238	17,08	17,08	9	0,56	217,222	42,263	50,343	3,108
0-600	9,93	39,29	23	23	28,93	28,93	22,34	0,74	64,5	27,832	5,415	0,388
0-600	9,93	39,29	21	21	25,62	25,62	18,51	0,9	24,722	4,81	5,73	0,354
0-600	9,93	39,29	2	2	40,66	40,66	35,75	1	2,928	0,57	0,679	0,042
0-600	12,38	96,02	508	508	14,91	14,91	8,87	0,44	420,372	81,789	97,435	6,014
600-1200	14,25	45,82	75	75	31,5	31,5	24,93	0,6	84,165	16,376	19,506	1,204
600-1200	14,27	45,85	139	139	6	6	3,7	0,11	75,31	14,653	17,454	1,077
600-1200	23,69	99,14	63	63	20,3	20,3	10,66	0,67	64,352	12,521	14,914	0,921
600-1200	23,69	99,14	210	210	17,61	17,61	8,93	0,6	197,303	38,388	45,727	2,823
600-1200	8,95	45,05	23	23	26,24	26,24	20,2	0,65	25,867	5,033	5,995	0,37
600-1200	8,95	45,05	24	24	26,45	26,45	19,94	0,83	27,776	5,404	6,437	0,397
600-1200	8,95	45,05	2	2	29,38	29,38	23,8	1	2,673	0,52	0,62	0,038
600-1200	15,29	99,14	536	536	17,67	17,67	11,05	0,49	477,14	92,834	110,582	6,826
1200-1800	9,52	24,42	57	57	29,83	29,83	24,27	0,63	63,495	12,354	14,716	0,908
1200-1800	9,54	24,45	141	141	1,71	1,71	0,29	0,04	63,445	12,344	14,704	0,908
1200-1800	23,28	81,29	45	45	18,77	18,77	9,42	0,6	43,892	8,54	10,172	0,628
1200-1800	23,28	81,29	225	225	17,56	17,56	9,31	0,59	209,473	40,756	48,547	2,997
1200-1800	10,57	39,46	30	30	32,37	32,37	24,95	0,83	38,823	7,554	8,998	0,555
1200-1800	10,57	39,46	24	24	26,77	26,77	20,72	0,75	26,824	5,219	6,217	0,384
1200-1800	10,57	39,46	1	1	49,84	49,84	45,4	1	1,618	0,315	0,375	0,023
1200-1800	13,23	81,29	523	523	16,06	16,06	10,01	0,47	447,578	87,082	103,731	6,403
1800-2400	9,54	33,1	57	57	26,08	26,08	21,56	0,53	57,435	11,175	13,311	0,823
1800-2400	9,56	33,13	153	153	2,29	2,29	0,99	0,05	70,282	13,674	16,289	1,085
1800-2400	22	109,02	56	56	20,44	20,44	10,95	0,64	56,822	11,055	13,169	0,813
1800-2400	22	109,02	228	228	14,85	14,85	7,36	0,5	194,853	37,911	45,159	2,788
1800-2400	10,45	32,67	40	40	26,02	26,02	19,64	0,75	46,595	9,066	10,799	0,667
1800-2400	10,45	32,67	22	22	27,06	27,06	21,29	0,77	25,253	4,913	5,853	0,361
1800-2400	10,45	32,67	0	0					0	0	0	0
1800-2400	12,89	109,02	556	556	14,39	14,39	8,86	0,42	451,3	87,807	104,593	6,456
2400-3000	10,13	33,25	64	64	24,64	24,64	19,7	0,52	62,96	12,25	14,591	0,901
2400-3000	10,15	33,28	148	148	1,89	1,89	0,21	0,07	69,039	13,432	16	0,888
2400-3000	17,48	84,07	36	36	19,01	19,01	9,83	0,61	35,417	6,891	8,208	0,307
2400-3000	17,48	84,07	204	204	14,93	14,93	7,41	0,53	177,608	34,556	41,162	2,541
2400-3000	8,35	38,51	30	30	26,97	26,97	19,63	0,8	36,416	7,085	8,44	0,521
2400-3000	8,35	38,51	19	19	18,6	18,6	14,15	0,53	17,226	3,352	3,992	0,246
2400-3000	8,35	38,51	0	0					0	0	0	0
2400-3000	11,53	84,07	501	501	13,47	13,47	8,02	0,42	398,528	77,539	92,363	5,701
3000-3600	8,12	25,86	48	48	29,23	29,23	23,93	0,6	52,31	10,178	12,123	0,748
3000-3600	8,14	25,89	127	127	1,84	1,84	0,43	0,04	57,117	11,113	13,277	0,817
3000-3600	16,17	84,09	31	31	15,01	15,01	8,01	0,42	25,821	5,024	5,984	0,369
3000-3600	16,17	84,09	216	216	13,94	13,94	7,54	0,5	180,656	35,149	41,869	2,584
3000-3600	7,25	37,54	24	24	24,76	24,76	16,84	0,79	27,861	5,421	6,457	0,399
3000-3600	7,25	37,54	18	18	29,5	29,5	23,96	0,78	21,071	4,1	4,883	0,301
3000-3600	7,25	37,54	4	4	8,76	8,76	4,82	0,75	3,69	0,718	0,855	0,053
3000-3600	9,21	84,09	468	468	13,4	13,4	8,41	0,41	366,9	71,774	85,496	5,278

Ek 1'in devamı

SABAH

TIME	MOVEMENT	QLEN	DELAY	GREEN TIME
08:00-08:10	S-W	3,27	11,35	12
08:00-08:10	S-N	3,27	1,95	42
08:00-08:10	N-W	21,1	19,1	26
08:00-08:10	N-S	21,1	21,04	26
08:00-08:10	E-W	5,35	14,11	25
08:00-08:10	E-S	5,35	14,93	25
08:00-08:10	E-N	5,35	6,24	25
08:10-08:20	S-W	13,41	37,27	12
08:10-08:20	S-N	13,41	3,56	42
08:10-08:20	N-W	33,02	31,5	26
08:10-08:20	N-S	33,02	28,6	26
08:10-08:20	E-W	4,99	11,75	25
08:10-08:20	E-S	4,99	18,08	25
08:10-08:20	E-N	4,99	0,4	25
08:20-08:30	S-W	9,16	32,62	12
08:20-08:30	S-N	9,16	4,02	42
08:20-08:30	N-W	24,93	24,87	26
08:20-08:30	N-S	24,93	23,16	26
08:20-08:30	E-W	7,16	17,22	25
08:20-08:30	E-S	7,16	14,46	25
08:20-08:30	E-N	7,16	0	25
08:30-08:40	S-W	6,69	25,56	12
08:30-08:40	S-N	6,69	3,64	42
08:30-08:40	N-W	26,19	27,31	26
08:30-08:40	N-S	26,19	23,27	26
08:30-08:40	E-W	5,09	17,77	25
08:30-08:40	E-S	5,09	19,15	25
08:30-08:40	E-N	5,09	0	25
08:40-08:50	S-W	8	30,17	12
08:40-08:50	S-N	8	3,97	42
08:40-08:50	N-W	30,86	31,94	26
08:40-08:50	N-S	30,86	26,37	26
08:40-08:50	E-W	5,48	10,99	25
08:40-08:50	E-S	5,48	22,75	25
08:40-08:50	E-N	5,48	0	25
08:50-09:00	S-W	4,99	17,83	12
08:50-09:00	S-N	4,99	3,38	42
08:50-09:00	N-W	25,72	27,37	26
08:50-09:00	N-S	25,72	23,86	26
08:50-09:00	E-W	4,2	13,98	25
08:50-09:00	E-S	4,2	16,01	25
08:50-09:00	E-N	4,2	33,55	25

Ek 1'in devamı

ÖĞLEN

TIME	MOVEMENT	QLEN	DELAY	GREEN TIME
12:00-12:10	S-W	9,52	33,66	12
12:00-12:10	S-N	9,53	5,72	42
12:00-12:10	N-W	13,06	14,88	26
12:00-12:10	N-S	13,06	13,38	26
12:00-12:10	E-W	16,28	30,9	10
12:00-12:10	E-S	16,28	29,77	10
12:00-12:10	E-N	16,28	24,23	10
12:10-12:20	S-W	15,84	35,05	12
12:10-12:20	S-N	15,85	11,58	42
12:10-12:20	N-W	15,34	16,79	26
12:10-12:20	N-S	15,34	14,18	26
12:10-12:20	E-W	12,4	23,73	10
12:10-12:20	E-S	12,4	27,97	10
12:10-12:20	E-N	12,4	37,52	10
12:20-12:30	S-W	10,6	34,67	12
12:20-12:30	S-N	10,61	7,21	42
12:20-12:30	N-W	11,94	12,95	26
12:20-12:30	N-S	11,94	11,92	26
12:20-12:30	E-W	20,35	33,74	10
12:20-12:30	E-S	20,35	35,73	10
12:20-12:30	E-N	20,35	21,4	10
12:30-12:40	S-W	8,48	31,99	12
12:30-12:40	S-N	8,49	3,88	42
12:30-12:40	N-W	12,03	12,09	26
12:30-12:40	N-S	12,03	13,76	26
12:30-12:40	E-W	12,31	22,28	10
12:30-12:40	E-S	12,31	33,74	10
12:30-12:40	E-N	12,31	20,53	10
12:40-12:50	S-W	12,07	32,44	12
12:40-12:50	S-N	12,08	8,89	42
12:40-12:50	N-W	14,51	15,09	26
12:40-12:50	N-S	14,51	12,95	26
12:40-12:50	E-W	15,22	27,06	10
12:40-12:50	E-S	15,22	32,39	10
12:40-12:50	E-N	15,22	9,12	10
12:50-13:00	S-W	8,34	34,05	12
12:50-13:00	S-N	8,36	5,06	42
12:50-13:00	N-W	13,8	15,92	26
12:50-13:00	N-S	13,8	15,16	26
12:50-13:00	E-W	12,28	25,42	10
12:50-13:00	E-S	12,28	30,98	10
12:50-13:00	E-N	12,28	4,26	10

Ek 1'in devamı

AKSAM

TIME	MOVEMENT	QLEN	DELAY	GREEN TIME
16:00-16:10	S-W	8,09	25,14	14
16:00-16:10	S-N	8,11	1,82	53
16:00-16:10	N-W	23,41	18,34	26
16:00-16:10	N-S	23,41	17,08	26
16:00-16:10	E-W	9,93	28,93	14
16:00-16:10	E-S	9,93	25,62	14
16:00-16:10	E-N	9,93	40,66	14
16:10-16:20	S-W	14,25	31,5	14
16:10-16:20	S-N	14,27	6	53
16:10-16:20	N-W	23,69	20,3	26
16:10-16:20	N-S	23,69	17,61	26
16:10-16:20	E-W	8,95	26,24	14
16:10-16:20	E-S	8,95	26,45	14
16:10-16:20	E-N	8,95	29,38	14
16:20-16:30	S-W	9,52	29,83	14
16:20-16:30	S-N	9,54	1,71	53
16:20-16:30	N-W	23,28	18,77	26
16:20-16:30	N-S	23,28	17,56	26
16:20-16:30	E-W	10,57	32,37	14
16:20-16:30	E-S	10,57	26,77	14
16:20-16:30	E-N	10,57	49,84	14
16:30-16:40	S-W	9,54	26,08	14
16:30-16:40	S-N	9,56	2,29	53
16:30-16:40	N-W	22	20,44	26
16:30-16:40	N-S	22	14,85	26
16:30-16:40	E-W	10,45	26,02	14
16:30-16:40	E-S	10,45	27,06	14
16:30-16:40	E-N	10,45	0	14
16:40-16:50	S-W	10,13	24,64	14
16:40-16:50	S-N	10,15	1,89	53
16:40-16:50	N-W	17,48	19,01	26
16:40-16:50	N-S	17,48	14,93	26
16:40-16:50	E-W	8,35	26,97	14
16:40-16:50	E-S	8,35	18,6	14
16:40-16:50	E-N	8,35	0	14
16:50-17:00	S-W	25,86	29,23	14
16:50-17:00	S-N	25,89	1,84	53
16:50-17:00	N-W	84,09	15,01	26
16:50-17:00	N-S	84,09	13,94	26
16:50-17:00	E-W	37,54	24,76	14
16:50-17:00	E-S	37,54	29,5	14
16:50-17:00	E-N	37,54	8,76	14

ÖZGEÇMİŞ

Amir SHAHKAR, 1989 yılında İranın Tebriz Şehrinde doğdu. 2001 yılında şehit Bakeri Okulundan İlköğretimini ve 2004 yılında Sefa Ortaokulundan mezun olarak İlk ve Ortaokul hayatını Tebriz şehrinde tamamladı. 2008 yılında Pajohesh Lisesinden mezun oldu. Aynı yıl girdiği üniversite sınavından Tebriz Azad Üniversitesi Müh. Fakültesi İnşaat Mühendisliği bölümünü kazanarak üniversite hayatına başladı. 2011 bahar dönemi İnşaat Mühendisi olarak mezun oldu. Ardından 2012 yılında Karadeniz Teknik Üniversitesi Fen Bilimler Enstitüsü İnşaat Mühendisliği Anabilim Dalında Yüksek Lisans yapmaya hak kazandı. 2015 yılının bahar dönminde hazırladığı “Trabzon Bölünmüş Sahil Yolu Üzerinde Coğrafi Bilgi Sistemleri ile Trafik Kazalarının Analizi” başlıklı tez ile Yüksek İnşaat Mühendisi ünvanıyla mezun oldu. 2016 yılında yılında Karadeniz Teknik Üniversitesi Fen Bilimler Enstitüsü İnşaat Mühendisliği Anabilim Dalında Doktora yapmaya hak kazandı.

Amir SHAHKAR Türkçe, Farsca, Arapca ve İngilizce bilmektedir. Trafik kaza analizleri, coğrafi bilgi sistemleri kullanarak kaza analizleri, Trafik ışıkları tasarımı, Trafik akış modelleme, kavşak analizi, kavşak tasarımı, akıllı ulaşım sistemleri, Matlab’de Modelleme ve adaptive trafik kontrolü gibi alanlarda çalışmaktadır.