



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY BAĞIŞIKLIK ALGORİTMALARI KULLANILARAK BULANIK SİSTEM TASARIMI

Ayşe Merve ACILAR

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalını

Haziran - 2013
KONYA
Her Hakkı Saklıdır.

ÖZET

DOKTORA TEZİ

YAPAY BAĞIŞIKLIK ALGORİTMALARI KULLANILARAK BULANIK SİSTEM TASARIMI

Ayşe Merve ACILAR

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ahmet ARSLAN

2013, 144 Sayfa

Jüri
Prof. Dr. Ahmet ARSLAN
Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. Bekir KARLIK
Doç. Dr. Erkan ÜLKER
Yrd. Doç. Dr. Taner ÜSTÜNTAŞ

İnsan düşünce sistemin iki temel özelliği vardır. Bunlardan birincisi belirsiz, kesin olmayan ve eksik bilgilerin bulunduğu ortamlarda sözel düşünme ve muhakeme yeteneği ile makul kararlar verebilmesi, ikincisi de herhangi bir ölçüm veya hesaplama yapmadan edindiği tecrübe ve gözlemleri kullanarak fiziksel ve zihinsel işlemleri yerine getirebilmesidir. İşte bu iki özelliği formülize etmeye odaklanan yöntem *Bulanık Mantık* olarak adlandırılır. Bulanık mantık kullanılarak modellenmiş sistemlere de *Bulanık Sistemler* ismi verilir.

Bulanık sistemler; esnek yapısı sayesinde kesin olmayan verileri, belirsizlik ve olasılık durumlarını, karmaşık ve doğrusal olmayan fonksiyonları, dilsel değişkenler kullanarak anlaşılması kolay bir biçimde modeller. Ancak, sistemin modellenmesinden önce, bulanık sistemlere ait çeşitli parametrelerin tespit edilmesi gerekir. Bu aşamada konu üzerinde uzmanlaşmış kişilerin fikir ve tecrübelerinden faydalanılması gerekir ama bir uzman görüşüne ulaşmak her zaman mümkün olmayabilir, mümkün olduğunda da uzmandan çok büyük ve dinamik yapıda olan bir veri kümesinden bu parametreleri doğru şekilde tespit etmesini beklemek zordur. Bu yüzden öğrenme yeteneği olan algoritmalar kullanılarak, gerçek dünya verilerinden bulanık sistem parametrelerinin tespit edilmesi araştırmacıların ilgisini çeken bir konu olmuştur.

Bu tez çalışmasında, evrimsel algoritmalardan yapay bağışıklık sistemi (YBS) algoritmalarının öğrenme yeteneğinden faydalanılarak sistem parametrelerinin gerçek dünya verilerinden tespit edildiği, verimli ve yorumlanabilirliği yüksek bulanık sistemler tasarlanması üzerinde çalışılmıştır. Bunun için bulanık mantığın çekirdeğinde yer alan *derecelendirme* ve *taneciklendirme* (D/T) kavramlarının üzerinde durulmuştur. Tezin ilk bölümünde, bulanık üyelik fonksiyonlarına (ÜF) ait en uygun taban uzunluklarının tespiti yapılmıştır. Üyelik fonksiyonunun taban uzunluğu değiştikçe, bireylerin o kümeye olan aitlik dereceleri de değiştiği için bu süreç derecelendirme işlemine karşılık gelmektedir. Diğer bir deyişle, bir bireyin her hangi bir bulanık kümeye olan aitliğinin en uygun derecesinin tespit edilmesi bir optimizasyon problemi olarak ele alınmış ve çözümü için YBS tabanlı bir algoritma olan CLONALG'nin kullanılması önerilmiştir. Önerilen yöntem, tek girişli/çıkışlı ve çok girişli/çıkışlı iki bulanık sistem örneği üzerine

tatbik edilmiştir. Elde edilen sonuçlar, popüler bir evrimsel algoritma olan genetik algoritma kullanılarak elde edilen sonuçlar ile istatistikî olarak kıyaslanmış ve önerilen yöntemin daha avantajlı olduğu gösterilmiştir.

Tezin ikinci aşamasında, ilk olarak her bir niteliğin kaç üyelik fonksiyonuna sahip olması gerektiğinin ve ÜF'larının başlangıç yerleşim yerlerinin bulunması için önerilen **S**iluet küme **D**oğrulama indeksi ve **k**-**o**rtalama tabanlı (SDko) yeni bir ön işlem algoritması tanıtılmıştır. Bu önişlem adımı sayesinde *taneciklendirme* işlemi için gerekli olan ön bilgiler uzman yardımı olmaksızın eğitim verisi kullanılarak elde edilmiştir. Daha sonra, bulanık tanecikleri derecelendirmesinin yanı sıra dinamik şekilde kural tabanını da optimize edebilen yeni bir algoritma önerilmiştir. GYAopt-aiNet ismi verilen bu algoritma (**G**eliştirilmiş ve memetic algoritmanın **Y**erel **A**rama stratejisiyle birleştirilmiş **opt-aiNet**), yapay bağışıklık tabanlı opt-aiNet algoritmasının tez çalışması esnasında tespit edilen eksikliklerinin giderilerek elde edilmiş olan versiyonuna, Memetik algoritmanın yerel arama mekanizmasının eklenmesi ile oluşturulmuştur. Böylece, opt-aiNet algoritmasının hiper mutasyon ve baskılama mekanizmalarında yapılan değişiklikler ile iyileştirilen genel arama yeteneği, Memetik algoritmanın yerel arama stratejisi ile tamamlanmış ve bu mekanizmalar sayesinde optimum sonuca daha doğru şekilde ulaşılması sağlanmıştır.

Önerilen yöntemler, özel bir bulanık mantık sistem türü olan bulanık sınıflandırıcı tasarlamak için kullanılmıştır. Sunulan yaklaşım, 20 farklı sınıflandırma veri kümesi üzerinde test edilmiş ve sonuçlar 22 farklı sınıflandırma yöntemi ile kıyaslanmıştır. Sonuçlar, üç kez tekrarlanan 10 çapraz doğrulama yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak elde edilmiştir. Elde edilen sonuçlar istatistikî olarak kıyaslanmış ve GYAopt-aiNet algoritmasının, bulanık sınıflandırıcı tasarım problemi için diğer algoritmalara göre daha makul, kabul edilebilir ve başarılı sonuçlar ürettiği gösterilmiştir.

Anahtar Kelimeler: Bulanık mantık, Bulanık taneciklerin tespiti, Bulanık taneciklerin derecelendirilmesi, Yorumlanabilir bulanık sınıflandırıcı tasarımı, Yapay Bağışıklık Algoritmaları, k-ortalama algoritması, Siluet küme doğrulama indeksi.

ABSTRACT

Ph.D. THESIS

THE FUZZY SYSTEM DESIGNING USING ARTIFICIAL IMMUNE SYSTEM ALGORITHMS

Ayşe Merve ACILAR

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER ENGINEERING

Advisor: Prof. Dr. Ahmet ARSLAN

2013, 144 Pages

Jury
Prof. Dr. Ahmet ARSLAN
Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. Bekir KARLIK
Assoc. Prof. Dr. Erkan ÜLKER
Assist. Prof. Dr. Taner ÜSTÜNTAŞ

The human thought system has two main remarkable capabilities. First of these capability is that to make reasonable decisions in an environment of imprecision, uncertainty, incompleteness of information and second, the capability to perform a wide variety of physical and mental tasks without any measurements and any computations. *Fuzzy logic* is a method that tries to formalize these human capabilities and the systems which are modelled using fuzzy logic are called *Fuzzy Systems*.

The fuzzy systems could model imprecise information, partiality of truth and partiality of possibility situations, complex and nonlinear functions using linguistic variables via their flexible structures. Before the fuzzy systems are designed, some parameters about systems must be determined. Experiences and opinions of an expert could be used in this stage because the fuzzy systems could not be learned its parameters by itself. However, it is not always possible to reach opinion of an expert or an expert couldn't detect the value of parameters correctly. Because of that, learning of the system parameters from the real world data using evolutionary algorithms becomes a popular research area.

In this thesis, we study on detecting the parameters of the fuzzy system from real world data to design efficient and interpretable systems using the learning capability of artificial immune system algorithm. For this, we used the *graduation* and *granulation* concepts that are placed in the core of fuzzy logic.

In the first part of the thesis, determining the most suitable base lengths of the fuzzy membership functions was made. While the base lengths of the fuzzy membership functions changes, the graduation degree of the individual also changes. In other words, graduation process was realised. This process could be thought as an optimization problem and CLONALG algorithm, one of the most popular artificial immune system algorithms, was used to solve it. The proposed method was applied to two different fuzzy system examples. One of them is single input single output fuzzy system and the other is multi input multi output fuzzy system. This optimization problem also solved using genetic algorithm. The obtained results from these algorithms were compared statistically and it was showed that the proposed method is more advantageous to solve this problem.

In the second part of the thesis, firstly a new preprocesses algorithm called SPP (**S**ilhouette cluster validity index aided **P**re-**P**rocess step via k-means clustering algorithm) was introduced to determine the number of membership functions and their initial boundaries. The prior knowledge needed for granulation process was acquired by the SPP algorithm without the help of an expert person.

Then, Mopt-aiNetLS (**M**odified version of **opt-aiNet** combined with **L**ocal **S**earch strategy of memetic algorithm) algorithm was proposed to graduate the fuzzy granules as well as to optimize the fuzzy rule base dynamically. The Mopt-aiNetLS was the combination of the memetic algorithm and modified version of the opt-aiNet algorithm, in which some changes were made in the suppression and hypermutation mechanisms of the original opt-aiNet algorithm. These two new mechanisms were called as Intelligent Suppression Mechanism and Adaptive Hypermutation Operator. Combining the modified version of opt-aiNet with the local search strategy of memetic algorithms improve the accuracy of the classification rate. An effective search process was realised by the Mopt-aiNetLS, because the global search capability of opt-aiNet was complemented by the local search strategy of the memetic algorithm.

The proposed algorithms were used to design a fuzzy classifierier that is a special case of fuzzy systems. In order to test the performance of this new approach, 20 different well-known classification dataset benchmark problems from the UCI dataset were used. The average 3x10 cross-fold validation results obtained from these datasets are presented and compared with the results of 22 different classification algorithms reported in the literature. The Wilcoxon Signed-Rank Test was also used for statistical comparisons. The obtained results demonstrate the effectiveness of the proposed approach.

Keywords: Fuzzy logic, Fuzzy granulation detection and graduation, Interpretable fuzzy system design, Artificial immune system algorithms, k-means algorithm, Silhouette cluster validity index.

TEŞEKKÜR

Doktora tez çalışmam boyunca değerli katkılarını, yönlendirici desteğini ve anlayışını hiçbir zaman esirgemeyen danışmanım Sayın Prof. Dr. Ahmet ARSLAN'a;

Tezin gelişmesine yönlendirici görüş ve önerileri ile yardımcı olan ve manevi desteklerini esirgemeyen tez izleme komitesi üyelerim Sayın Prof. Dr. Şirzat KAHRAMALI'ya ve Sayın Yrd. Doç. Dr. Taner ÜSTÜNTAŞ'a;

Tez çalışmasının son halini alması aşamasında kıymetli fikirleri ile katkıda bulunan jüri üyelerim Sayın Prof. Dr. Bekir KARLIK'a ve Sayın Doç. Dr. Erkan ÜLKER'e;

Tez çalışmam boyunca bursiyeri olduğum '2211 Yurtiçi Doktora Burs Programı' kapsamında maddi destekte bulunan TÜBİTAK BİDEP'e;

Tez çalışmamızdan çıkan yayınlara vermiş oldukları maddi desteklerden dolayı Selçuk Üniversitesi BAP Koordinatörlüğü'ne ve TÜBİTAK'a;

Bu süre boyunca göstermiş oldukları manevi desteklerinden, ilgilerinden ve yardımlarından dolayı Sayın Arş. Gör. Semiye DEMİRCAN'a, Sayın Yrd. Doç. Dr. Humar KAHRAMANLI'ya, Bilgisayar Mühendisliği Bölümü'nün tüm öğretim elemanlarına ve diğer mesai arkadaşlarıma;

Çalışmalarım boyunca ve hayatımın her anında anlayışları ve destekleri ile yanımda ve yardımcım olan sevgili eşim Zafer ACILAR'a ve kayınvalidem Havva ACILAR'a;

Manevi destekleri ile her zaman bana moral veren sevgili kardeşlerime;

Üzerimde sonsuz hak ve emekleri olan, bugünlere gelmemi sağlayan ve tezim süresince her türlü destekleri ile beni hiç yalnız bırakmayan annem Şerife ve babam Samim ŞAKİROĞLU başta olmak üzere tüm aileme;

İçtenlikle teşekkür eder, şükran ve minnetlerimi sunarım.

Ayşe Merve ACILAR
KONYA-2013

Canım Oğlum
Osman'a

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **vi**

TEŞEKKÜR ..... **viii**

İÇİNDEKİLER ..... **x**

KISALTMALAR ..... **xii**

1. GİRİŞ **1**

1.1. Tezin Amacı 2
1.2. Tezin Konusu ve Literatüre Katkısı 4
1.3. Kaynak Araştırması 7
1.3.1. Bulanık mantığın gelişimi 7
1.3.2. Üyelik fonksiyonların tespiti ile ilgili kaynak araştırması 9
1.3.3. Bulanık sınıflandırıcı sistemler hakkında kaynak araştırması 12
1.4. Tezin Organizasyonu 14

2. MATERYAL VE YÖNTEM **15**

2.1. Kullanılan Algoritmalar ve Yöntemler 15
2.1.1. Genetik algoritmalar 15
2.1.2. Memetik algoritmalar 17
2.1.3. Yapay bağışıklık algoritmaları 19
2.1.4. k-ortalama kümeleme algoritması 19
2.1.5. Siluet küme doğrulama indeksi 21
2.1.6. *k*-kez çapraz doğrulama işlemi 22
2.1.7. Wilcoxon işaretli-sıra testi 22
2.2. Kullanılan Veri Kümeleri 24
2.3. Kullanılan Cihazlar ve Programlar 25

3. YAPAY BAĞIŞIKLIK SİSTEMLERİ **27**

3.1. İnsan Bağışıklık Sistemi 27
3.1.1. İnsan bağışıklık sisteminin yapısı 27
3.1.2. Bağışıklık hücreleri 28
3.1.3. Bağışıklık sistemi vücudu nasıl korur? 30
3.1.4. Klonal seçme teorisi 32
3.1.5. Bağışıklık ağ teorisi 33
3.2. Yapay Bağışıklık Sistemleri 35
3.2.1. Şekil uzayı 36
3.2.2. Duyarlılık hesaplaması 37
3.2.3. Yapay bağışıklık algoritmaları 38
3.2.3.1. Klonal seçme modeli ve CLONALG algoritması 38
3.2.3.2. Bağışıklık ağ modeli ve opt-aiNet algoritması 40

4. BULANIK MANTIK 44

4.1. Bulanık Mantığın İlkesel Yönleri 45
4.2. Bulanık Kümeler ve Üyelik Fonksiyonları 49
4.2.1. Üyelik fonksiyonları 50
4.2.2. Üyelik fonksiyonunun kısımları 50
4.2.3. Bulanık kümelerin sınıflandırılması 52
4.2.4. Bulanık küme işlemleri 53
4.3. Bulanık Sistemler 56
4.3.1. Bulanık bir sistemin genel yapısı ve birimleri 56
4.3.2. Bulanık bir sistemin parametreleri 63
4.3.3. Bulanık bir sistemin yorumlanabilirliği 63
4.4. Bulanık Sınıflandırıcı Sistemi 66

5. ÜYELİK FONKSİYONLARININ OPTİMİZASYONU İÇİN ÖNERİLEN YAPAY BAĞIŞIKLIK SİSTEMİ TABANLI BİR YÖNTEM 70

5.1. Antikorun/Bireyin Kodlanması 71
5.2. Duyarlılık Fonksiyonunun Hesaplanması 74
5.3. Örnek Uygulamalar ve Deneysel Sonuçlar 76
5.3.1. Tek girişli - çıkışlı bir bulanık sistem örneği için ÜF optimizasyonu 76
5.3.2. Çok girişli - çıkışlı bir bulanık sistem örneği için ÜF optimizasyonu 82

6. BULANIK SINIFLANDIRICI TASARIMI İÇİN ÖNERİLEN SDKo ve GYAopt-aiNet ALGORİTMALARI 90

6.1. Önerilen Önişlem Algoritması: SDko 92
6.2. GYAopt-aiNet Algoritması 98
6.2.1. Antikorun/Bireyin Kodlanması 99
6.2.2. Başlangıç Popülasyonun Oluşturulması 101
6.2.3. Uygunluk ve Duyarlılık Fonksiyonları 102
6.2.4. Akıllı Baskılama Mekanizması 103
6.2.5. Adaptif Hipermutasyon Mekanizması 105
6.2.6. Memetik Algoritmanın Yerel Arama Mekanizması 107
6.3. Sunulan Yöntemin Deneysel Sonuçları 109
6.3.1. GYAopt-aiNet ile opt-aiNet'in Kıyaslanması 110
6.3.2. Sunulan Yöntemin Diğer Yöntemlerle Kıyaslanması 114

7.SONUÇLAR VE ÖNERİLER 120

7.1. Sonuçlar 120
7.2. Öneriler 124

KAYNAKLAR 125

EKLER 133

ÖZGEÇMİŞ 143

KISALTMALAR

ABM	: Akıllı Baskılama Mekanizması
AHM	: Adaptif Hipermutasyon Mekanizması
BM	: Bulanık Mantık
BMKS	: Bulanık Mantık Kontrol Sistemi
CLONALG	: Klonal Seçme Algoritması
ÇG-Ç	: Çok Girişli ve Çok Çıkışlı Bulanık Mantık Sistemi
D/T	: Derecelendirme / Taneciklendirme
GA	: Genetik Algoritma
GYAopt-aiNet	: **G**eliştirilmiş ve memetic algoritmanın **Y**erel **A**rama stratejisiyle birleştirilmiş **opt-aiNet** algoritması
MA	: Memetik Algoritma
SDKo	: **S**iluet küme **D**oğrulama indeksi ve **k**-**o**rtalama tabanlı önişlem algoritması
TG-Ç	: Tek Girişli ve Tek Çıkışlı Bulanık Mantık Sistemi
ÜF	: Üyelik Fonksiyonu
YA	: Yerel Arama
YBS	: Yapay Bağışıklık Sistemi

1. GİRİŞ

Her sabah gözümüzü açtığımız andan itibaren gün içerisinde karşılaştığımız durumları değerlendirmek veya ifade etmek için kullandığımız cümleleri bir düşünelim.

- Hava sıcaklığı bugün -1 derece. Kabanımı giyinmeliyim.
- Çay makinesinde 10 ml çay kalmış. Bir kupayı doldurmaz.
- Akşam gittiğimiz alışveriş merkezinde metrekareye 8 kişi düşüyordu.

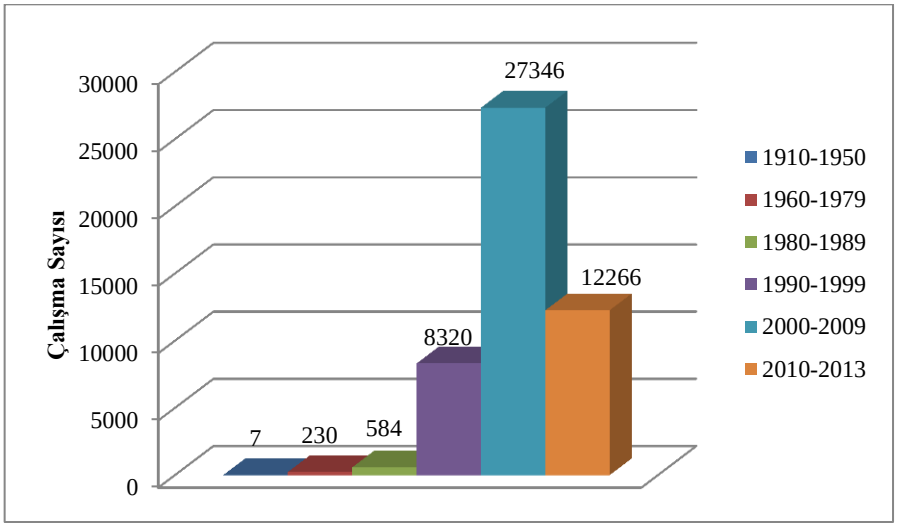
Sanırım kullanılan cümleler yukarıdakilere pek benzemezdi. Onların yerine;

- Hava bugün **çok** soğuk görünüyor. Kabanımı giyinmeliyim.
- Çay makinesinde **az** çay kalmış. Bir kupayı doldurmaz.
- Akşam gittiğimiz alışveriş merkezi **çok kalabalıktı.**

ifadelerini tercih ederdik. Çünkü insan düşünce sistemi sayısal değil sözel bilgileri işlemeye, onlar arasında ilişki kurmaya daha yatkındır. Kendisini dilsel değerler kullanarak ifade eder. Bunun aksine bilgisayarlar ise sayısal verileri kullanma ve işlemede daha iyidirler. Peki, bilgisayarların insan düşünce sistemlerinden faydalanılması sağlanamaz mı? Zirâ gerçek dünyada karşılaşılan karmaşık problemlerin çözümü için tasarlanması gereken sistemlerde kullanılmak üzere yeterince sayısal veriye ulaşılması genellikle zahmetli ve/veya maliyetli bir süreçtir. İşte elimizde yeterli bilginin bulunmadığı böylesi karmaşık durumlarda insanın değer yargılarından, düşünce ve muhakeme yeteneğinden faydalanılmasını öngören ilkelerin tümünü "Bulanık Mantık" başlığı altında toplayabiliriz.

Bulanık ilkeler hakkındaki ilk bilgiler, Azerbaycan asıllı Lütfi (Lotfi) Askerzade (Zadeh) tarafından bilimsel literatüre kazandırılmıştır (Zadeh, 1965). Zadeh, uzun yıllar boyunca çalıştığı kontrol alanında istediği kontrolü elde edebilmek için oldukça karmaşık ve doğrusal olmayan denklemler ile uğraşmış, çözüme ulaşmasının giderek zorlaşması sonucunda da yeni çözüm yolları aramaya başlamıştır. Bunun üzerine bulanık mantık kavramın temelini oluşturan *bulanık kümeleri* tanımladığı çalışma ortaya çıkmıştır. Ancak bu fikirler iki çıktılı Aristo mantığına alışık olan Batı dünyası tarafından oldukça tenkit almış ve dışlanmıştır. Bulanık mantığın dünya çapında önem kazanması ise Mamdani (1974) tarafından yapılan ve bir buhar makinesi kontrolünün ilk defa bulanık sistem kullanılarak gerçekleştirildiği uygulama çalışması sayesinde olmuştur. Bu çalışmadan, bulanık sistemlerle çalışmanın hem kolay hem de bir o kadar etkili olduğu anlaşılmıştır. Daha sonraki yıllarda özellikle Japonya, Singapur, Kore ve

Malezya gibi doğu ülkelerinde bulanık kavramların kullanılması sıkça gözlemlenmiştir. 1980'lerde ise bulanık sistemlerin kullanılmasında patlama yaşanmış, elektrik süpürgesi ve çamaşır makinesi gibi ev aletlerinden metro ve şirket işletimine kadar pek çok alanda başarılı uygulamalar gerçekleştirilmiştir. Günümüzde ise bilim ve mühendisliğin hemen hemen her dalında bulanık mantık kullanımın yaygınlaştığı görülmektedir. Başta bu konuya şüpheyle yaklaşan batıda ise, bu gelişmeler karşısında bulanık mantığın kullanılabilirliği kabul görmüş ve önemi giderek artmıştır (Şen, 2004).



Şekil 1.1. Bulanık Mantık konusunda yapılan çalışmaların yıllara göre dağılımı

Yukarıdaki grafik, bulanık mantık konusunda yapılan çalışmaların yıllara göre dağılımını göstermektedir. Grafik, ProQuest (erişim tarihi, 08.05.2012) veri tabanında "Fuzzy Logic" anahtar kelimesi arattırılıp elde edilen sonuçlar kullanılarak oluşturulmuştur. Grafikten araştırmacıların bulanık mantık konusuna karşı artan ilgisi net bir şekilde görülmektedir. 2010-2013 yılları arasında yapılan çalışmaların sayısı şimdiden, bir önceki on yılda yapılan çalışmaların neredeyse yarısına ulaşmış durumdadır. Diğer bir deyişle bulanık mantık, oldukça cazip ve güncel bir araştırma konusudur.

1.1. Tezin Amacı

Fiziksel veya kavramsal olarak birbiriyle ilişkili birimlerin oluşturduğu bütüne *sistem;* sistemin gelecekteki davranışını tahmin etmek için kullanılan araca *model* ismi verilir. Bileşenleri ve aralarındaki ilişkileri modellerken bulanık küme teorisi temelli bir

matematiksel disiplin olan bulanık mantık ilkelerinden faydalanan sistemler de *Bulanık Sistemler* olarak adlandırılır.

Sistemler, endüstri, sağlık, bilim, mühendislik vb. gibi çok çeşitli alanlarda durumların analiz edilebilmesi, doğru ve kaliteli kararlar alınabilmesi için insanlara yardım ederler. Bir önceki bölümde de bahsedildiği gibi son 40 yıla kadar sistemlerin modellenmesinde ikili Aristo mantığına dayalı matematik yapılarından (cebirsel eşitlikler, diferansiyel denklemler, doğrusal olmayan denklemler, sonlu durum makineleri vb.) faydalanılırdı. Ancak bazı durumlarda çözüme ulaşmak için oldukça karmaşık denklemler ile uğraşılması gerekiyordu veya matematiksel model kurmak için yeterli veriye ulaşmak çok maliyetli olabiliyor ve/veya uzun zaman alabiliyordu. Bu sebeplerden dolayı insan tecrübe ve algılarının sözel veriler olarak sisteme dâhil edilmesine imkân veren bulanık sistemler alternatif bir modelleme şekli olarak ortaya çıktı ve sistem tasarımında kullanımı giderek arttı.

Sözel verilerin derecelendirilerek modellenmesine imkân veren ve böylece insan düşence sistemini taklit edebilen bulanık sistemlerin en büyük dezavantajı öğrenme yeteneğinin olmamasıdır. Bulanık sistemler, ne problem çözme sırasında ne de sonrasında, temel bileşenleri olan üyelik fonksiyonlarını ve kurallarını öğrenemez. Bundan dolayı başlangıçta kuralların uzman deneyimlerine dayanarak tanımlanması gerekir. Ancak bir uzman görüşüne ulaşmak her zaman mümkün olmayabilir, mümkün olduğunda da uzmandan çok büyük ve dinamik yapıda olan bir veri kümesinden bulanık kümeleri doğru şekilde tespit etmesini beklemek zordur. Üyelik fonksiyonlarının belirlenmesinde ise belli bir yöntem yoktur. En sık kullanılan yöntem deneme yanılma yöntemidir ki bu da uzun zaman alabilmekte ve birçok test yapmayı gerektirmektedir.

Bu tez çalışmasının esas amacı, bulanık sistem tasarımı esnasında uzmana olan bağımlılığı en aza indirerek, evrimsel algoritmalar yardımı ile öğrenme yeteneği kazandırılmış verimli ve yorumlanabilirliği yüksek bulanık sistemler tasarlamayı mümkün kılmaktır. Evrimsel algoritma olarak, içerisinde çeşitlilik, hafıza ve baskılama (suppression) mekanizmalarını barındıran yeni bir yapay bağışıklık algoritmansının kullanılması önerilmiştir. Sunulan yeni yönteme memetik algoritmaların yerel arama özelliği de eklenmiştir. Böylece uzman yerine gerçek dünya eğitim verilerinden faydalanılarak, bulanık mantığın çekirdeğini oluşturan *derecelendirme* ve *taneciklendirme (D/T)* işlemleri yerine getirilmiş ve bulanık kurallar tespit edilmiştir.

1.2. Tezin Konusu ve Literatüre Katkısı

Bulanık mantığın çekirdeğini oluşturan derecelendirme ve taneciklendirme kavramları, bulanık mantığı diğer mantık sistemlerinden ayıran başlıca özellikleridir. Bulanık mantıkta izin verilen her şey yani bulanıklık bir olayın olup olmadığını değil, hangi dereceye kadar olduğunu ölçmektedir. Taneciklendirme de, değişkenlerin benzerliklerine, farklılıklarına, yakınlık veya işlevselliklerine bakılarak parçacıklara bölünmesidir. İkisi birlikte dilsel ifade kavramını oluştururlar. Bu tez çalışmasında, bulanık mantığa ait bu iki kavramın ve bulanık bir sisteme ait kuralların gerçek dünya verilerinden öğrenilmesi üzerine çalışılmıştır.

Tez çalışması iki aşamadan oluşmaktadır. İlk aşamasında, üyelik fonksiyonlarına (ÜF) ait parametrelerin en uygun değerlerinin belirlenmesi, bir optimizasyon problemi olarak ele alınmış ve çözümü için bir yapay bağışıklık algoritması olan Klonal seçme algoritmasının (CLONALG) kullanılması önerilmiştir. Bulanık mantık sistemine ait üyelik fonksiyonlarının belirlenmesi için kullanılan yöntem, giriş değişkenleri olarak üyelik fonksiyonunun tipini; bulanık mantık sistemine ait giriş/çıkış değişkenlerini ve onların tanımlandığı aralıkları; sistemin bulanık kural tabanını ve referans noktalarını alır. Bu değişkenlerden faydalanan CLONALG algoritması, bulanık mantık sisteminin ÜF'ları için en uygun taban uzunluklarını öğrenir, diğer bir değişle taban uzunluklarını optimize ederek derecelendirme işlemi gerçekleştirilir. Çünkü üyelik fonksiyonunun taban uzunluğu değiştikçe, bireylerin o kümeye olan aitlik dereceleri de değişmektedir. Bu sayede bir bireyin, bulanık kümeye olan aitliğinin en uygun derecesi tespit edilir.

Ancak bahsedilen giriş değişkenlerinin tespiti için bir uzmanın yardımına ihtiyaç duyulması, derecelendirme işlemi için önerilen bu yöntemin dezavantajıdır. Bu çalışmanın amacı bulanık sistem tasarlanırken uzmana olan bağımlılığı en aza indirmek olduğu için, tezin ikinci aşmasında bu giriş değişkenlerinin de gerçek dünya verilerinden öğrenilebilmesi için sunulan yöntemler açıklanmıştır.

İlk olarak, her bir niteliğin sahip olacağı ÜF sayısını ve ÜF'larının başlangıç yerleşim yerlerini bulması için önerilen **S**iluet küme **D**oğrulama indeksi ve **k**-**o**rtalama tabanlı (SDko) yeni bir ön işlem algoritması tanıtılmıştır. Bu önişlem adımı sayesinde *taneciklendirme* (bulanık mantığın çekirdeğinde bulunan diğer kavram) işlemi için gerekli olan ön bilgiler uzman yardımı olmaksızın eğitim verisi kullanılarak elde edilmiştir. SDko önişlem algoritmasının sağladığı avantajlar aşağıda listelenmiştir:

- Derecelendirme ve taneciklendirme işlemleri için gerekli ön bilgilerin eğitim verilerinden elde edilmesini sağlar. Uzmana olan bağımlılığı azaltır.
- Her bir değişkenin kaç üyelik fonksiyonuna ayrılacağını hesaplar.
- Her bir üyelik fonksiyonunun başlangıç taban aralıkları ile tepe noktalarını belirleyerek arama uzayının daraltılmasını ve doğru sonuca daha çabuk ulaşılmasını sağlar.
- Her bir değişken için tam, ayrıklaştırılabilir, tamamlayıcılığı olan ve makul sayıda üyelik fonksiyonu elde edilmesini mümkün kılar, sistemin yorumlanabilirliğini artırır.

Daha sonra, SDko önişlem adımı ile tespit edilen bulanık tanecikleri derecelendirmesinin yanı sıra dinamik şekilde kural tabanını da optimize edebilen GYAopt-aiNet algoritması ayrıntılı şekilde açıklanmıştır. GYAopt-aiNet (**G**eliştirilmiş ve memetik algoritmanın **Y**erel **A**rama stratejisiyle birleştirilmiş **opt-aiNet**), yapay bağışıklık tabanlı *opt-aiNet* algoritmasının bu tez çalışması esnasında geliştirilmiş şeklidir. Ayrıntılı bir şekilde incelenen opt-aiNet algoritması üzerinde aşağıda listelenen değişiklikler yapılarak GYAopt-aiNet algoritması literatüre kazandırılmıştır.

1. *Akıllı Baskılama (Suppression) Mekanizması - ABM*: Orijinal op-aiNet algoritmasının baskılama mekanizmasında, eğitim verisinin giriş vektörleri (antikorlar) arasındaki duyarlılık (benzerlikleri) hesaplanırken Öklid uzaklığı kullanılmaktadır. Eğer iki giriş vektörü arasındaki mesafe, baskılama eşiğinden düşükse ilk vektör uygunluk değerine bakılmaksızın baskılanır. Baskılamadan kastedilen, bu giriş vektörünün popülasyondan çıkarılmasıdır. Çünkü bu giriş vektörüne veya antikoruna benzeyen, yani bu antikoru baskılayan bir antikor zaten sistemde mevcuttur. Ancak bunun sonucunda muhtemel bir çözümün silinme riski ortaya çıkabilir. Bunu önlemek için, bu tez çalışmasında Akıllı Baskılama Mekanizması (ABM) isimli bir yöntem sunulmuştur. Mekanizmanın akıllı diye adlandırmasının sebebi, popülasyondaki her bir antikorun, kendisi tarafından silinen antikorların bir listesini tutuyor olmasındandır. Bir antikor diğer bir antikor tarafından baskılandığı zaman, baskılayan antikor baskılanan antikorun silme listesini kontrol eder. Eğer listedeki tüm elemanları da baskılıyorsa, baskılanan antikor silinir. Aksi takdirde antikor korunur. Bu

demektir ki, antikoru baskılayan antikor, baskıladığı antikorun daha önceden baskılamış olduğu tüm elemanları temsil ediyorsa ancak o zaman silinir. Böylece potansiyel bir optimum çözümün silinmesi engellenir. Bu yöntem baskılama esnasında antikorların uygunluk değerlerine de bakar. Kimin uygunluk değeri daha kötüyse her zaman o silinir. Önerilen mekanizmanın orijinal olanından bir farkı da, iki bireyin benzerliğinin giriş uzayında değil, amaç uzayında hesaplanmasıdır. Böylece antikorların benzerlikleri şekil uzayındaki yerlerinin yakınlığı ile değil, yaptıkları işlevlerin benzerliği ile değerlendirilir. Diğer bir farkı ise iki birey arasındaki duyarlılık hesabında Öklid uzaklığı yerine özel veya (xor) operatörünün kullanıldığı Hamming uzaklığının kullanılıyor olmasıdır.

2. *Adaptif Hipermutasyon Mekanizması - AHM*: Orijinal opt-aiNet algoritmasında hipermutasyon mekanizmasında, mutasyon değişim miktarı olan α'nın değişim miktarını kontrol eden sabit bir β sayısı bulunmaktadır. AHM'de ise sabit olan bu β sayısı, iterasyon adımına duyarlı olacak şekilde adaptif hale getirilmiştir. Bunun için Tan ve ark. (2006) tarafından tanımlanan adaptif mutasyon operatöründen faydalanılmıştır. Böylece mutasyon değişim miktarı olan α'nın büyük bir değerden başlatılması ve iterasyon numarası arttıkça mutasyon değişim miktarının azaltılması sağlanmıştır. Yani algoritmanın başlangıcında, arama uzayı büyük adımlar ile taranmış, algoritmanın daha ilerleyen aşamaların da ise adım boyu azaltılarak ve daha yerel bir arama gerçekleştirilmiştir.

3. *Memetik Algoritmanın Yerel Arama Stratejisin eklenmesi:* Orjinal opt-aiNet algoritmasına yukarıda anlatılan iki değişikliğin uygulanması ile *Geliştirilmiş opt-aiNe*t algoritması elde edilmiştir. Son olarak bu algoritmaya Memetik algoritmanın yerel arama stratejisi eklenmiş ve GYAopt-aiNet algoritması oluşturulmuştur. Bu son adım sayesinde opt-aiNet algoritmasının genel arama yeteneği, Memetik algoritmanın yerel arama stratejisi ile tamamlanmıştır.

Önerilen SDKo önişlem ve GYAopt-aiNet algoritmaları, bu tez çalışmasında bulanık sınıflandırıcı sistemi veya bulanık kural tabanlı sınıflandırıcı sistemi tasarlamak için kullanılmıştır. Tezin devamında bulanık sınıflandırıcı ismiyle anılacak olan sistem,

verilerin sınıflandırılması için kullanılan ve çıkış değeri, verinin hangi sınıfa ait olduğunu gösterdiği için kesin ve ayrık olan özel bir bulanık sistem türüdür.

Sunulan yaklaşım, California, Irvine Üniversitesinin makine öğrenmesi veri kümesi havuzundan (Blake and Merz, 1998) alınan herkese açık 20 farklı sınıflandırma veri kümesi üzerinde test edilmiş ve sonuçlar 22 farklı sınıflandırma yöntemi ile kıyaslanmıştır. Kıyaslamak için kullanılan sınıflandırma yöntemlerinin sonuçları, KEEL (http://www.keel.es/) isminde; sınıflandırma, regresyon, kümeleme v.b. alanlarda çalışan araştırmacıların çalışmalarında faydalanabilmesi için altı farklı İspanyol araştırma grubun işbirliği ile geliştirilen açık kaynak kodlu ve ticari olmayan bir yazılım aracı kullanılarak elde edilmiştir. Sonuçlar, üç kez tekrarlanan 10 çapraz doğrulama yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak bulunmuş ve sunulan algoritmanın verimliliği gösterilmiştir.

1.3. Kaynak Araştırması

Kaynak araştırması bölümü 3 ana başlık altında incelenecektir. Öncelikle bulanık mantığın gelişimi ele alınacak, sonra üyelik fonksiyonlarının tespit edilmesi ile ilgili yapılan başlıca çalışmalar verilecek ve son olarak da bulanık sınıflandırıcı sistemler ile ilgili konu hakkında alt yapı oluşturacak kaynaklar özetlenecektir.

1.3.1. Bulanık mantığın gelişimi

1900'lü yıllarda Polonya'lı mantıkçı Jan Lukasiewicz (1878-1956), önermelerin sadece bir veya sıfır doğruluk değeri alabildiği ikili Aristo mantığından farklı olarak, önermelerin bir ve sıfır arasında da “belki” diye isimlendirilebilen bir değer daha alabildiği "üç değerli" mantık ilkelerini açıklamıştır. Bu gelişmeyi takiben, Max Black (1937) liste ya da nesnelerden oluşan kümelere çok değerli mantığı tatbik ederek literatürdeki ilk bulanık eğrileri çizmiştir. Lukasiewicz ve Black tarafından alt yapısı hazırlanan bulanık mantık konusu Lotfi A. Zadeh’in (1965) yayınladığı "Bulanık Kümeler" adlı çığır açıcı yazısı ile literatürdeki yerini almıştır. Zadeh bu makalede, bir kümenin tüm elemanlarına Lukasiewicz'in mantığını uygulayarak bulanık kümeler için eksiksiz bir cebir geliştirmiştir. Ancak bu çalışma, Mamdani ve Assilian (1974) tarafından bir buhar makinesinin kontrolü için bulanık mantıkla çalışan bir sistemin

uygulaması yapılana kadar özellikle Batı dünyasında pek çok eleştiriye maruz kalmış ve kullanım alanı bulamamıştır. Mamdani ve Assilian'ın (1974) çalışmasından sonra bulanık mantık kuramı araştırmacıların ve ticari uygulamacıların dikkatini çekmeye başlamıştır.

Bunu takiben Zadeh (1975) yayınladığı bir dizi makale ile *dilsel değişken (linguistic variable)* kavramını açıklamış ve bir değişkenin alabileceği değerlerin sadece sayısal değil dilsel (sözel) de olabileceğini ifade etmiştir. Dilsel ifadelerin kullanılması ile birlikte, durumlar ve şartlar çok daha rahat ifade edilebildiği ve modellenebildiği için dilsel değişkenler bulanık mantığın ticari uygulamalarının gelişmesinde çok önemli rol oynamıştır.

Bulanık mantığın ilk önemli ticari uygulaması çimento sanayinde olmuştur. 1980'lerde Danimarka'da bir firma geleneksel denetleyici yerine bir bulanık mantık denetleyicisi kullanmış ve başarılı sonuçlar elde etmiştir (Holmblad ve Østergaard, 1982). Avrupa'da ve özellikle Amerika'da bulanık mantığa karşı olan şüpheci yaklaşım devam ederken, doğu'da özellikle Japonya'da bulanık mantık ile ilgili olarak hem ticari hem de akademik çalışmalara olan ilgi artmıştır. 1985 senesinde Togai ve Watanabe ilk bulanık çip'i geliştirmişler ve bu çip 1988 yılında OMRON firması tarafından ticari uygulamalarda kullanılmaya başlanmıştır. Bulanık çiplerin ticari olarak üretilmeye başlanması ile birlikte beyaz eşya, tren, asansör, trafik kontrolü, otomotiv sanayi vb. gibi alanlarda bulanık mantık uygulamalarının sayısı hızla çoğalmıştır. 1987 yılında diğer önemli bir bulanık mantık kontrol uygulaması olan Japonya'daki Sendai Metro treni kontrolü, taşımacılık sektöründe ön plana çıkmıştır (Hitachi Ltd.,1987). Bu uygulama sayesinde metro trenlerinin istenilen konumda durması üç kat iyileşmiş ve enerji tüketimi %10 oranında azalmıştır. 1989 yılında Japonya Uluslararası Ticaret ve Endüstri Bakanlığı (MITI), Japonya'nın 48 büyük şirketini bir araya getirerek bulanık sistemler konusunda çalışmaları için bir topluluk kurmalarını sağlamış ve bulanık mantığa verdiği önemi göstermiştir. Bu topluluk Uluslararası Bulanık Mühendisliği Laboratuarı (Laboratory of International Fuzzy Systems-LIFE) ismini almış ve bütçesi 70M$ olarak belirlenmiştir. Bu arada Amerika'da bulanık mantığa karşı olan tepkiler devam etse de, Avrupa'da Japonya'dakine benzer çalışmalar yavaş yavaş ortaya çıkmaya başlamıştır. 1991 yılında Fransa'da Araştırma Bakanlığı tarafından üniversite ve sanayi işbirliği ile bulanık mantık konusunda çalışacak CRIN topluluğu kurulmuştur. Almaya'da da aynı amaçla ELITE isimli bir grup oluşturulmuştur. Bu grupların amacı

Avrupa endüstrisinde bulanık mantığın etkinliğini artırmaktır. Bu gelişmelerin ışığında Amerika'da bulanık mantığa daha fazla duyarsız kalmamış ve bu konuda çalışmalara başlamıştır. NASA'da gelişmiş teknolojide kanat üretimi araştırmalarında bulanık mantıktan faydalanmıştır (Dutta,1993). Günümüzde ise bilim ve mühendisliğin hemen hemen her dalında bulanık mantık kullanımın yaygınlaştığı görülmektedir. Şu an başta Japonya, Çin, Amerika ve Batı Avrupa ülkeleri olmak üzere Türkiye'ninde içinde bulunduğu otuzdan fazla ülkede bulanık mantık konusunda araştırmalar yapılmaya devam etmektedir.

1.3.2. Üyelik fonksiyonların tespiti ile ilgili kaynak araştırması

Bulanık mantığın temelini oluşturan bulanık kümelerin her biri üyelik fonksiyonları kullanılarak karakterize edildiği için, üyelik fonksiyonları bulanık mantık sistem tasarımında önemli bir rol oynar. Bundan dolayıdır ki, pek çok bulanık mantık kontrol problemi, üyelik fonksiyonlarını tanımlayan parametrelerin tespiti konusuyla yakından ilgilidir. Bu parametrelerin tespiti için kullanılabilecek iki potansiyel kaynak vardır: uzman görüşü ve gerçek dünya verileri.

Zadeh (1972), ÜF'larına ait parametrelerinin tespiti için, insanın doğal zekâsının ve kavrayışının sonucunda edindiği sezgilerini yani uzmanlığını kullanabileceğini belirtmiştir. Ancak uzman görüşüne ulaşmak her zaman kolay olmayabilir. Bu yüzden gerçek dünya verilerinden üyelik fonksiyonlarının tespiti araştırmacıların ilgisini çeken bir konu olmuştur. Literatürde gerçek dünya verilerinin, üyelik fonksiyonuna çevrimi sırasında kullanılabilecek pek çok farklı yöntem önerilmiştir. Bu yöntemler genellikle istatistiksel yaklaşımlardan ve makine öğrenmesi tekniklerinden faydalanır.

İstatistikte, verilerin olasılık dağılımı hakkında fikir edinmemize yardımcı olan ve genellikle Gauss fonksiyonları ile temsil edilen Histogram grafikleri, üyelik fonksiyonlarının parametrelerinin belirlenmesinde de doğrudan ya da dolaylı olarak kullanılmışlardır. Kullanımı kolay bir yöntem olmasına karşın sıklıkla farklı sınıflara ait histogramların örtüşmesi sonucu, üyelik fonksiyonlarının dilsel değerlerinin tespiti zorlaşır. Cheng ve ark. (1997), görüntü ayrıştırma (image segmantation) problemindeki görüntü eşiklerinin tespitinde kullanılmak üzere geliştirdikleri bulanık sistemde, üyelik fonksiyonlarının genişliğinin otomatik olarak elde edilmesi için histogram bilgisinden faydalanmışlardır. Öncelikle veriye ait histogram incelenerek üyelik fonksiyonlarının

tepe noktaları belirlenmiş, daha sonrada iki tepe arasında olabilecek en uygun genişlik hesaplanmıştır. Dou ve ark. (2005) MRI görüntülerinden beyin dokularının özelliklerini çıkarmak için sundukları yöntemde üyelik fonksiyonlarının oluşturulmasında histogram analizini kullanmışlardır. Ancak, ilerleyen zamanlarda parametre tespiti için makine öğrenmesi tekniklerinin ve öğrenme algoritmalarının kullanılması daha popüler bir yaklaşım olarak benimsenmiştir. Bu yöntemler arasında en sık kullanılanı, bir veri kümesini benzer bireyleri içerecek şekilde alt kümelere ayıran kümeleme algoritmalarıdır. Ancak bu algoritmaların dezavantajı, küme sayısının veya kümeleri birbirinden ayıran eşik değerlerinin daha önceden bilinmesi gerekliliğidir. Tung ve Quek (2002a), bulanık üyelik fonksiyonu tanımlamak için, bu iki parametreden herhangi birinin önceden belirlenmesine ihtiyaç duymayan DIC isimli yeni bir kümeleme algoritması geliştirmişler ve bir sonraki çalışmalarında da (2002b) bu algoritmayı bir sinirsel-bulanık sistemin üyelik fonksiyonlarının dilsel değerlerinin tespitinde kullanmışlardır. Ancak bu algoritmaya ait *eğim (slope)* ve *adım (step)* isimli parametrelerin önceden belirlenmesi gerekmektedir.

Kümeleme algoritmalarında en uygun küme sayısının bulunması için önerilen bir yol ise küme doğrulama indeksi olarak isimlendirilen ölçütlerden faydalanılmasıdır. Shen ve ark. (2005) üyelik fonksiyonlarının oluşturulması için en büyük beklenti (expectation maximation) yöntemini ve küme doğrulama indeksli bir k-ortalama algoritmasının kullanılmasını önermişlerdir. Derbel ve ark. (2008) yılında CLUSTER (Bandyopadhyay, 2004) algoritması ile Davies-Bouldin* (DB*) küme doğrulama indeksinin birleşimi olan CLUSTERDB* (Hachani, 2007) algoritmasını kullanarak, sistemdeki bulanık üyelik fonksiyonların sayısını ve parametre değerlerinin değerlerini bulan bir yöntem geliştirmişlerdir.

Diğer bir makine öğrenme tekniği olan yapay sinir ağları da (YSA), bulanık üyelik fonksiyonlarının elde edilmesinde sıklıkla kullanılmıştır. Lin ve Lee (1991) üyelik fonksiyonlarının tespiti için danışmansız bir öğrenme yaklaşımı olan *kendini düzenleyen kümeleme* (self-organising clustering) ile YSA'nın birleşiminden oluşan bir yöntem önermişlerdir. Öncelikle üyelik fonksiyonlarının merkezlerinin ve genişliklerinin ilk değerleri kümeleme yöntemi ile belirlenmiş daha sonra bu değerler geriye yayılımlı YSA ile optimize edilmiştir. Medasani ve ark. (1998) ise YSA'daki bir sinir hücresinin sigmoid aktivasyon fonksiyonunun çıkış değeri ile bulanık kümelerin üyelik değerlerinin oldukça benzer olduğuna dikkat çekmişlerdir. Bir sınıflandırma

problemi için YSA'nın nasıl kullanıldığını ise şu şekilde açıklamışlardır. YSA'nın giriş düğümlerinin sayısı, eğitim örneğinin niteliklerinin sayısına, çıkış düğümleri ise sınıf sayısına eşittir. Giriş değerine göre ilgili tek bir çıkış düğümünün değeri 1, diğerleri ise 0'dır. Sınıf üyelik derecelerinin elde edilmesi için çok katmanlı YSA uygun bir eğitim algoritması (örneğin geriye yayılım algoritması gibi) ile eğitilir. Eğitim bittikten sonra, elde edilen ağ, üyelik değerlerinin elde edileceği ağ olur. Yang ve Bose (2006), bulanık üyelik fonksiyonlarının oluşturulması için kendini düzenleyen nitelik haritalarından faydalanmışlardır. Daha sonra bu yöntemlerini örüntü tanıma alanına uygulamışlardır. Ang ve Quek (2012) ise çalışmalarında, insan sinir sisteminin bir parçası olan, denge için bilinçaltı düzeyinde kas kasılmalarını kontrol eden beyinciğin çalışmasından esinlenerek geliştirdikleri SPSEC (Supervised Pseudo Self-Evolving Cerebellar) isimli algoritmayı üyelik fonksiyonlarının öğrenilmesinde kullanmışlardır.

Literatürdeki çalışmaların bir kısmı ise üyelik fonksiyonlarının parametrelerinin uygun değerlerinin tespitini bir optimizasyon problemi olarak ele alır ve çözümü için meta sezgisel optimizasyon algoritmalarının kullanımını önerir. Bağış (2003), üyelik fonksiyonlarının tespiti için Tabu Arama Algoritmasını kullanan bir yöntem sunmuştur. Simon (2005), H_∞ durum tahmin teorisini, üyelik fonksiyonlarının parametre optimizasyonuna uygulamıştır. Çalışmasında, H_∞ filtresine durum kısıtlarını ekleyerek filtre üzerinde bazı değişiklikler yapmıştır.

Üyelik fonksiyonlarının parametrelerinin tespitinde veya optimizasyonunda Genetik algoritmaların kullanımı da oldukça yaygındır. Karr (1991), üyelik fonksiyonlarının tespiti için Genetik Algoritmayı (GA) ilk defa bir Bulanık Mantık Kontrol Sistemine (BMKS) uygulamıştır. Meredith ve ark. (1993), ise bir helikopter için BMKS'e ait üyelik fonksiyonlarının en iyi şekilde ayarlanmasında GA'dan faydalanmışlardır. Alcalá-Fdez ve ark. (2009) üyelik fonksiyonların temsili için *ikili dilsel gösterim modeli* (2-tubles linguistic representation model) isimli yeni bir gösterim biçimi önermişlerdir. Bu gösterim biçimiyle birlikte üyelik fonksiyonlarının yatay olarak kaydırma imkânı veren bir parametre kullanılmaya başlanmıştır. En uygun kayma miktarının bulunması içinde GA kullanılmıştır. Arslan ve Kaya (2001), tek giriş ve çıkışlı (TG-Ç) bir bulanık sistem için bulanık mantık üyelik fonksiyonlarının tespitinde kullanılmak üzere yeni bir yöntem sunmuş ve bu yöntemi GA kullanarak kodlamışlardır. Çok girişli ve çıkışlı (ÇG-Ç) bulanık sistemlerde üyelik fonksiyonların tespiti, TG-Ç'lı olanlara göre daha karmaşıktır. Kaya, Karcı ve Arslan (2001) bu

sorunun çözümü için de GA'yı kullanmışlardır. Şakiroğlu ve Arslan (2007), Arslan ve Kaya (2001)'de kullandığı yöntemi bir yapay bağışıklık sistemi algoritması olan CLONALG ile gerçekleştirmiştir. Çalışmada üyelik fonksiyonlarının şekli üçgen olarak seçilmesiyle birlikte, matematiksel modeli bilinen herhangi bir şeklin kullanılabileceği belirtilmiştir. Sunulan algoritma ile üyelik fonksiyonlarının taban uzunluklarının optimum değerinin bulunması hedeflenmiştir. Acılar ve Arslan (2011) tarafından bu yöntem çok girişli çok çıkışlı bir bulanık mantık sistemine tatbik edilmiş ve aynı yöntemin GA ile optimize edildiği durum ile sunulan yöntem kıyaslanmış ve CLONALG'nin kullanımın verimliliği gösterilmiştir.

1.3.3. Bulanık sınıflandırıcı sistemler hakkında kaynak araştırması

Geleneksel yöntemde, bir bulanık sistem tasarlanırken dilsel bilgilerin yani üyelik fonksiyonların ve bulanık kuralların elde edilmesi için bir uzmanın varlığı gereklidir. Ancak bu süreç genellikle zor ve zaman alıcıdır. Ayrıca sistemin başarısı kişinin uzmanlık derecesi ile yakından ilgilidir. Literatürde, bu süreci uzmana gerek kalmaksızın otomatikleştirmek üzere giriş-çıkış verilerini bulanık kurallara çeviren çalışmalara sıklıkla rastlanmaktadır (Uebele ve ark., 1995). Bunun sonucunda, hem sınıflandırma hem de fonksiyon yakınsama problemleri için uzmana gerek kalmaksızın öğrenme algoritmalarının yardımı ile veri tabanlarından otomatik olarak gerekli bilgilerin elde edildiği adaptif bulanık sistemler tasarlanabilmiştir. Bu sistemlerin çoğu bulanık mantık ile YSA, evrimsel algoritmalar, makine öğrenmesi tekniklerinin birlikte kullanılması ile elde edilmiştir. Bu bölümde, verilerin sınıflandırılması için kullanılan ve çıkış değeri, verinin hangi sınıfa ait olduğunu gösterdiği için kesin ve ayrık olan bulanık sınıflandırıcıların adaptif şekilde tasarlanması için kullanılan yöntemler ele alınacaktır.

Bulanık sınıflandırıcı tasarımı ile ilgili pek çok çalışmada, YSA'nın öğrenme kabiliyeti ve bulanık mantığın sadeliği birleştirilmiştir. Ancak YSA'daki her bir sinir hücresinin ve ağırlıklarının ne anlama geldiğinin yorumlanması zordur. Kullanıcı sistemin iç dinamiklerini çözemez bu da bulanık mantığın kolay anlaşılabilir olma özelliği ile çelişir. Bu yüzden evrimsel algoritmalar, makine öğrenmesi teknikleri gibi diğer algoritmaların kullanımı ön plana çıkmıştır. Son yıllarda ise yapay bağışıklık sistemleri de bulanık sınıflandırıcı sistemlerin tasarımı için kullanılmaya başlanmıştır.

Konu ile ilgili genel bir fikir sahibi olmak için bu çalışmaların bazılarının özetleri aşağıda verilmiştir.

Uebele ve ark. (1995) eğitilmiş bir sinir ağından çıkarılan hiper düzlemleri kullanarak veri kümesinin sınıflandıran bir yöntem önermişlerdir. Daha sonra, bu düzlemlerden faydalanarak her bir sınıf için bulanık kurallar tanımlamışlar böylece sinir ağını kullanmadan doğrudan verinin sınıflandırılmasını sağlamışlardır. Abe ve Thawonmas (1997) ise çalışmalarında kümeleme tekniğinden faydalanmışlardır. Öncelikle eğitim verilerini kümelemişler, daha sonra her bir küme için merkezini ve içerdiği eğitim verileri kullanarak kovaryans matrisini hesaplamışlar ve bu matristen bulanık kuralları çıkarmışlardır. Ishibuchi ve ark. (1997) genetik algoritma tabanlı bir optimizasyon yöntemi ile sınıflandırma doğruluğu yüksek, mümkün olduğunca az kurallı bir bulanık sınıflandırıcı tasarlamayı hedeflemişlerdir. Chang ve Lilly (2004) herhangi bir ön bilgiye veya verinin dağılımı hakkında bir varsayıma ihtiyaç duymadan, sadece veri kümesini kullanarak bulanık sınıflandırıcı sistemi elde eden evrimsel bir yaklaşım önermişlerdir. Mansoori ve ark. (2007) kesin eşik değerine dayanan parçalı doğrusal bir ağırlık fonksiyonu kullanılarak etiketlenmiş veriler ile otomatik olarak bulanık sınıflandırıcı tasarlayan bir yöntem sunmuşlardır. Eşik değerini, bulanık kuralın kapsama alt uzayındaki örüntülerin dağılımına göre hesaplamışlardır. Örüntülerin uyumluluk dereceleri, eğer örüntüler bu kural tarafından doğru sınıflandırılmışsa güncellenmiş aksi takdirde aynı bırakılmıştır. Sonuçta, bulanık sınıflandırma kuralları Mansoori ve ark. (2007) tarafından elde edilen ağırlık fonksiyonun genel formu kullanılarak elde edilmiştir. Lei and Ren-hou (2008) klonal seçme ve hipermutasyon prensiplerini kullanan bir bulanık sınıflandırıcı sistem önermişlerdir. Yaklaşımlarına göre, popülasyonun her bir bireyi, bir bulanık sınıflandırıcı kural kümesini temsil etmektedir. Bu kural kümesi üyelik fonksiyonlarının parametrelerinden ve bulanık kurallardan oluşmaktadır. Çalışmanın amacı, bulanık kural kümesinin her iki parçasını da aynı anda optimize etmektir. Wang ve ark. (2008) klonal seçme ve harmoni arama algoritmalarının birleşiminden elde ettikleri yeni yöntemi bulanık sınıflandırıcı tasarımına tatbik etmişlerdir. Li ve Wang (2009) bulanık sınıflandırıcı tasarımı için HCGA isimli melez bir genetik algoritma geliştirmişlerdir. Yöntemlerinde her bir birey bir bulanık kuralı temsil etmektedir. Daha iyi sonuçlar elde etmek için yerel arama yöntemini kullanmışlar ve son olarak en son popülasyondaki gereksiz kurallar elenerek bulanık sınıflandırıcı sistemini elde etmişlerdir. García-Pedrajas ve Fyfe (2008) bulanık

sınıflandırıcılardan oluşan bir topluluk (ensemble) elde etmek için yapay bağışıklık sisteminden faydalanmışlardır. Her bir sınıflandırıcı popülasyondaki bir birey olarak ele alınmıştır. Popülasyon yapay bağışıklık dinamikleri kullanılarak eğitildikten sonra, hafıza hücreleri ve en son popülasyondaki en iyi bireyler farklı sınıflandırıcı toplulukları inşa etmek için kullanılmıştır.

1.4. Tezin Organizasyonu

Yapılan bu tez çalışması 7 ana bölümden oluşmaktadır. İlk bölümde, tezin konusu tanıtılmış, amacı, literatüre olan katkısı ve konuyla ilgili kaynak araştırması verilmiştir. İkinci bölümde; tez çalışması sırasında kullanılan materyal ve yöntemler tanıtılmıştır. Üçüncü bölümde, tez uygulamasında kullanılan yapay bağışıklık sistemleri ve ilgili algoritmalar olan CLONALG ve opt-aiNet hakkında ayrıntılı bilgi verilmiştir. Dördüncü bölümde ise, bulanık kümeler, bulanık mantık, bulanık çıkarım sistemleri ve bulanık sınıflandırıcılar incelenmiştir. Beşinci ve altıncı bölümler, tezin uygulamasının anlatıldığı ve deneysel sonuçların sunulduğu kısımlardır. Beşinci bölümde, üyelik fonksiyonlarının uygun taban aralıklarının tespiti bir optimizasyon problemi olarak ele alınmış ve çözümü için bir yapay bağışıklık algoritması olan CLONALG'nin kullanılması önerilmiştir. Ayrıca tek/çok girişli tek çıkışlı ve tek/çok girişli çok çıkışlı bulanık sistemler için ayrı ayrı incelemeler gerçekleştirilmiştir. Elde edilen sonuçlar, üyelik fonksiyonlarının genetik algoritma ile optimize edildiği zaman elde edilen sonuçlar ile istatistiksel olarak kıyaslanmış ve algoritmanın verimliliği gösterilmiştir.

Altıncı bölümde ise, öncelikle tez çalışması kapsamında sunulan k-ortalama ve siluet küme doğrulama indeksi tabanlı ve SDKo isimli yeni bir önişlem algoritması tanıtılmıştır. Daha sonra bu tez çalışması esnasında geliştirilen GYAopt-aiNet algoritması ayrıntılı bir şekilde anlatılmıştır. Bu yöntemler bulanık sınıflandırıcı sistem tasarımı problemine uygulanmış ve elde edilen sonuçlar, başka sınıflandırma algoritmaları ile kıyaslanarak önerilen algoritmanın başarısı tespit edilmiştir.

Son bölüm olan yedinci bölümde, elde edilen sonuçların yorumları, yöntemlerin literatüre katkısı ve gelecekte yapılabilecek çalışmalar hakkında bilgi ve öneriler verilmiştir.

2. MATERYAL VE YÖNTEM

Bu bölümünde, öncelikle tez çalışması esnasında kullanılan algoritmalar ve yöntemler anlatılacak, uygulamalarda kullanılan veri kümeleri tanıtılacak ve son olarak tez çalışması esnasında kullanılan yazılımlar ve donanımlar hakkında bilgiler verilecektir.

2.1. Kullanılan Algoritmalar ve Yöntemler

2.1.1. Genetik algoritmalar

Genetik Algoritmaların (GA) temel prensipleri ve teorik esasları ilk kez Holland (1975) tarafından tanımlanmıştır. Ardından Holland'ın doktora öğrencisi olan Goldberg (1989), GA ile ilgili teorik bilgileri uygulamaya çevirmiş ve bu uygulama örneklerini "Arama, Optimizasyon ve Makine Öğrenmesinde Genetik Algoritmalar (Genetic Algorithms in Search, Optimization, and Machine Learning)" isimli kitabında toplamıştır. İçinde pek çok uygulama örneği bulunan bu çalışmadan sonra, GA'nın popülerliği artmış özellikle makine öğrenmesi ve optimizasyon alanlarında araştırmacılar tarafından sıklıkla kullanılmaya başlanmıştır.

GA, doğal seçilim ve güçlünün hayatta kalma prensiplerini savunan evrim sürecin modellenmesidir. GA, her biri muhtemel bir çözüm olan bireylerin oluşturduğu ve popülasyon olarak adlandırılan bir topluluk üzerinde eşleşme ve evrimsel operatörlerin tekrarlı olarak icra ettirilmesidir. GA'nın kullandığı ve bireyin gelişmişliğinin artırılmasını amaçlayan iki adet temel evrimsel operatör bulunmaktadır. Bunlar çaprazlama ve mutasyon operatörleridir. Her bir bireyin gelişmişliği ise, problemi karakterize eden bir uygunluk (fitness) fonksiyonu tarafından belirlenir.

Öncelikle popülasyondaki her bir bireyin uygunluğu belirlenir ve uygunluklarına göre sıralanır. Uygunluk derecesi yeterince iyi olan bireyler ebeveyn olarak seçilir ve eşleştirilirler. Eşlenen bireyler daha sonra sırasıyla çaprazlama ve mutasyon işlemine tabii tutularak yeni bir neslin oluşması sağlanır. Algoritmanın durdurma kriteri sağlanıncaya kadar bu süreç devam eder. Popülasyondaki uygunluğu en iyi olan birey çözüm olarak kabul edilir. Algoritma ile ilgili kullanılan terminoloji ve algoritmanın adımları aşağıda verilmiştir (Mitchell,1997).

Genetik algoritma ile ilgili kullanılan terminoloji:

- *Uygunluk (Fitness) :* Her bir bireyin gelişmişliğini karakterize eden fonksiyon
- *n:* Popülasyonda yer alan bireylerin sayısı
- *p:* Popülasyonda yer alan bir birey
- *m:* Mutasyon oranı

Algoritmanın Adımları:

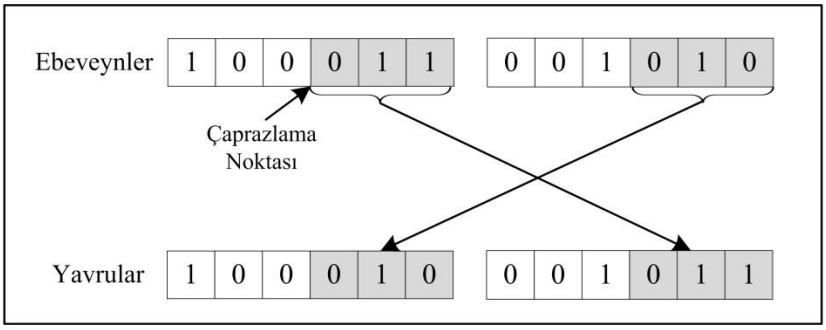
1. *n* tane birey içeren başlangıç *P* popülasyonu rastgele oluşturulur.
2. *P* popülasyonundaki her bir *p* bireyinin uygunlukları hesaplanır.
3. *Durdurma şartı sağlanasıya kadar tekrar edilir:*
    a. *Seçim:* Çaprazlama işlemine tabii tutulacak ebeveyn bireyler seçilir. Her bir p_i (i=1,..,n) bireyinin seçilme olasılığı denklem 2.1'e göre belirlenir.
    $$Olasılık(p_i) = \frac{Uygunluk(p_i)}{\sum_{j=1}^{n} Uygunluk(p_j)} \quad (2.1)$$
    b. *Çaprazlama:* Seçilen bireyler eşleştirilir ve çaprazlama işlemine tabii tutulur.
    c. *Mutasyon:* Çaprazlama sonucu oluşan yeni popülasyondaki bireyleri %*m* kadarı mutasyona uğratılır.
    d. *Değerlendirme:* Yeni oluşan popülasyondaki bireylerin uygunlukları hesaplanır ve *P* popülasyonuna güncellenir.
4. Durdurma şartı sağlanmışsa algoritmayı sonlandır ve uygunluğu en iyi olan bireyi çözüm olarak seç değilse adım (3.a)'a gidilir.

Durdurma şartı olarak en sık kullanılan yöntemler aşağıda listelenmiştir:

1. Daha önceden belirlenmiş bir iterasyon sayısına ulaşılması,
2. Daha önceden belirlenmiş uygunluk eşiğine ulaşılması,
3. Birbirini takip eden *k* adım boyunca ortalama hata değeri belirlenen değerin altında kalması halinde durdurma şartı sağlanmış olur.

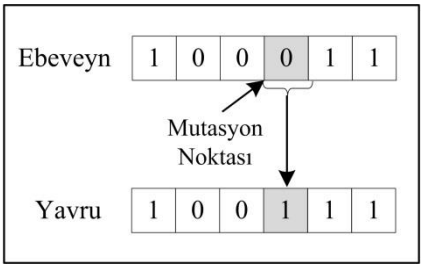
Bu tez çalışmasında, daha önceden belirlenmiş bir iterasyon sayısına ulaşılması durdurma şartı olarak kullanılmıştır.

Çaprazlama işlemi uygunluk değerine göre seçilmiş iki ebeveyn bireyden iyi özellikte yeni bireyler elde etmeyi amaçlar. Rastgele belirlenen çaprazlama noktasına göre karşılıklı bireylerde genlerin yer değiştirmesiyle yavru bireyler elde edilir. Bu tez çalışmasında Çaprazlama olarak da *tek noktaya göre çaprazlama* yöntemi kullanılmıştır. Şekil 2.1'de bu yöntemi açıklayan bir örnek gösterilmiştir.



Şekil 2.1. Tek noktalı çaprazlama işlemi

Mutasyon işlemi de çaprazlama gibi popülasyonda çeşitliliği sağlayan yöntemlerden biridir. Mutasyon işlemi belirlenen mutasyon oranına göre seçilen bireyin rastgele belirlenen bir geninin değer değiştirmesi şeklinde özetlenebilir. Bu sayede yeni bireylerin oluşması sağlanıp, popülasyonun yerel optimumlara takılması engellenir. Mutasyon noktası, çaprazlama noktası gibi rastgele belirlenir. Şekil 2.2'de bu tez çalışmasında kullanılan *tek nokta mutasyon işleminin* bir birey üzerinde uygulaması gösterilmektedir.



Şekil 2.2. Tek nokta mutasyon işlemi

2.1.2. Memetik algoritmalar

Memetic kelimesin kökü olan "meme" kelimesi, taklit anlamına gelen İngilizce "mimic" kelimesinden türemiştir. İlk defa Dawkins (1976) tarafından kullanılan ve

“kültürel taklidin veya iletimin en temel birimi” olarak tanımlanan “meme”, diğer bir ifadeyle toplum içindeki bireylerin kültürel değişimlerini modellemek için kullanılan en küçük yapı taşına verilen isimdir. Memetik Algoritma (MA) kavramını ise ilk defa Norman ve Moscato (1989) tarafından kullanılmıştır.

Sonraki yıllarda evrimsel hesaplama alanında oldukça popüler bir araştırma alanı haline gelen MA, popülasyon tabanlı global bir algoritma ile her bir birey tarafından gerçekleştirilen yerel bir arama algoritmasının bir birlikteliği olarak ele alınır. Memetik algoritmalarda popülasyonun her bir bireyinin uygunluğunu iyileştirmek için yerel bir arama uygulanır. Bunun devamında mevcut popülasyondaki yüksek kalitedeki bireyler seçilir, kullanılan evrimsel algoritmanın temel adımları bu seçilen bireylere tatbik edilir ve böylece bir sonraki iterasyon için yeni bir popülasyon elde edilmiş olunur (Garg, 2009). Bu yapıyı temel olan pek çok MA modeli oluşturmak mümkündür. Örneğin Norman ve Moscato (1989) geleneksel genetik algoritma ve benzetilmiş tavlama algoritmasını bir arada kullanan bir MA modeli kurmuşlardır.

Yerel arama algoritmaları, arama ve optimizasyon problemlerinin çözümünde en çok kullanılan yöntemlerdendir. Yerel araştırma mekanizması, araştırmayı yerel optimum seviyeye veya daha önceden belirlenmiş bir seviyeye ulaştıran bir yaklaşımdır. Bu algoritmalar, problemin eldeki bir çözümünü her yeni iterasyonla daha iyi bir çözüme götürmeye çalışır ve araştırma iyileşmenin durduğu ana kadar devam eder. Yerel arama algoritmalarının optimum çözüme ulaşması büyük oranda başlangıç çözümünün genel optimumun bulunduğu bölgeye uzaklığına ve iterasyon sayısına bağlıdır. Bütün bunlarla birlikte etkin, verimli ve kolay uygulanabilir olma özellikleriyle çok yaygın kullanılırlar (Akay, 2006).

Memetik algoritmaları diğer evrimsel algoritmalardan ayıran kısmı da içerisinde barındırdığı bu yerel arama mekanizmasıdır. Burada yerel arama adımı bireylerin kültürel değişikliklerini temsil eder. Buna karşın diğer evrimsel algoritmalar genellikle canlılardaki biyolojik değişiklikleri modellerler. Örneğin, GA’da bireylerin genleri üzerinde gerçekleşen değişiklikler ancak bir sonraki adımda gözlemlenebilirken, MA’daki bireylerin meme’lerindeki değişim yeni nesile geçmeden de gözlemlenebilmektedir. Evrimsel algoritma ile yerel arama algoritmasının bu etkin özelliklerin birleştirilmesi ile elde edilen MA modellerinin başarısı genellikle tatmin edicidir. MA’nın çalışma adımları genel olarak aşağıda verildiği gibidir.

1. Başlangıç *P* popülasyonu rastgele oluştur.
2. *P* popülasyonundaki her bir bireyinin uygunlukları hesapla.
3. *Durdurma şartı sağlanasıya kadar tekrar et:*
    a. Kullanılan evrimsel algoritmanın dinamiklerini çalıştır.
    b. Oluşan popülasyondaki bireylere yerel arama uygula.
    c. Oluşan yeni popülasyonun uygunluklarını hesapla ve *P* popülasyonunu güncelle.
4. Durdurma şartı sağlanmışsa algoritmayı sonlandır ve uygunluğu en iyi olan bireyi çözüm olarak seç değilse adım (3.a)'a git.

Algoritmanın 3.a ve 3.b adımları kurulan modele göre kendi aralarında yer değiştirebilmesi mümkündür. Daha önceden belirlenmiş bir iterasyon sayısı durdurma kriteri olarak kullanılabilir.

Bu tezde, MA'daki evrimsel algoritma olarak bir yapay bağışıklık algoritması olan opt-aiNet'in tez çalışması esnasında geliştirilen bir versiyonunu ve yerel arama algoritması olarak da Ang (2010) tarafından önerilen EMA-AIS isimli yöntem kullanılmıştır. Geliştirilen algoritma GYAopt-aiNet olarak isimlendirilmiş ve Bölüm 6'da ayrıntılı şekilde ele alınmıştır.

2.1.3. Yapay bağışıklık algoritmaları

Tez çalışmasında kullanılan yapay bağışıklık algoritmaları Bölüm 3'de ayrıntılı bir şekilde ele alınmıştır.

2.1.4. k-ortalama kümeleme algoritması

Kümeleme, veri tabanındaki nesnelerin ortak özelliklerine göre gruplanması işlemidir. K-ortalama (MacQueen, 1967), basitliği ve kolay uygulanabilir olması sebebiyle sıklıkla tercih edilen bir kümeleme algoritmasıdır. Değerlendirmeye alınan uzay ile sınıf merkezi olarak seçilen noktalar arasındaki mesafeye göre benzerlik grupları oluşturmayı amaçlar. Mesafe hesaplanırken çeşitli uzaklık ölçütlerinden faydalanılabilir. Bu tez çalışmasında denklem 2.2'de verilen Öklid uzaklığı mesafe hesabında kullanılmıştır.

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + ... + |x_{ip} - x_{jp}|^2} \quad (2.2)$$

K-Ortalama algoritması ile ilgili terminoloji ve algoritmanın adımları aşağıda verilmiştir (Han and Kamber,2001).

Algoritma ile ilgili kullanılan terminoloji:

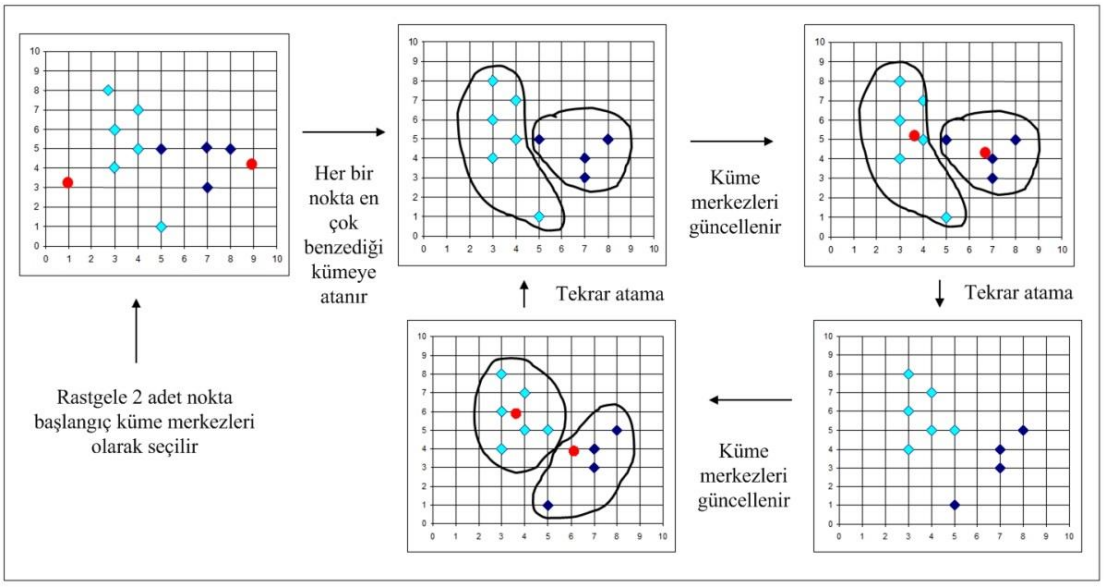
v: Veri kümesinin boyutu

k: Veri kümesinin kaç kümeye ayrılacağı

Algoritmanın adımları:

1. Başlangıç olarak rastgele k tane noktayı başlangıç küme merkezi olarak ata.
2. *Küme elemanlarında hiçbir değişiklik olamayana kadar tekrar et:*
   a. Geriye kalan her bir nokta ile küme merkezleri arasındaki uzaklığı denklem 2.2'yi kullanarak hesapla.
   b. Her bir noktanın hangi küme merkezine daha yakın olduğunu tespit et ve o kümenin elemanı olarak ata.
   c. Her bir küme için yeni merkez değerleri belirle. (Yeni merkez değerini hesaplama için, kümedeki noktaların ortalamasını hesapla)
3. Bitir.

Şekil 2.3'de k=2 için k-ortalama algoritmasının çalışmasına bir örnek verilmiştir. Şekilde algoritmanın tüm adımları sırası ile işaretlenmiştir.



Şekil 2.3. k=2 için k-ortalama algoritmasının çalışmasını gösteren bir örnek (Han and Kamber,2001)

2.1.5. Siluet küme doğrulama indeksi

Kümeleme analizlerinde, sonuç kümelemelerinin değerlendirilmesi kümeleme modeli geliştirme işleminin ayrılmaz bir parçasıdır. Çünkü bir veri kümesinde küme yapısı olmasa bile kümeleme algoritmaları bu veri seti içerisinde istenilen sayıda küme bulacaktır. Bundan dolayı kümeleme algoritmalarının sonuçlarının değerlendirilmesine yönelik çeşitli sayısal küme doğrulama (cluster validity) yöntemleri geliştirilmiştir. Bu sayede kümeleme çalışmalarında, küme kalitesi ve uygun küme sayısı belirlenerek kümeleme işlemleri başarıyla tamamlanabilir (Vatansever ve Büyüklü, 2009).

Bu tez çalışmasında küme doğrulama yöntemi olarak Rousseeuw (1987) tarafından önerilen Siluet (silhouette) küme doğrulama indeksi (SD) kullanılmıştır. SD indeksinin hesaplanması denklem 2.3'de verilmiştir.

$$SD(i) = \frac{(b(i)-a(i))}{\max\{a(i),b(i)\}} \tag{2.3}$$

$$b(i) = \min_{C \neq A} d(i, C), \text{ C=1,…,k} \tag{2.4}$$

Burada;

k : Sistemdeki küme sayısını

A : *i* noktasının ait olduğu kümeyi temsil eden numarayı;

a(i) : *i* noktasının, kendi kümesi içerisindeki diğer noktalara olan uzaklıklarının ortalamasını;

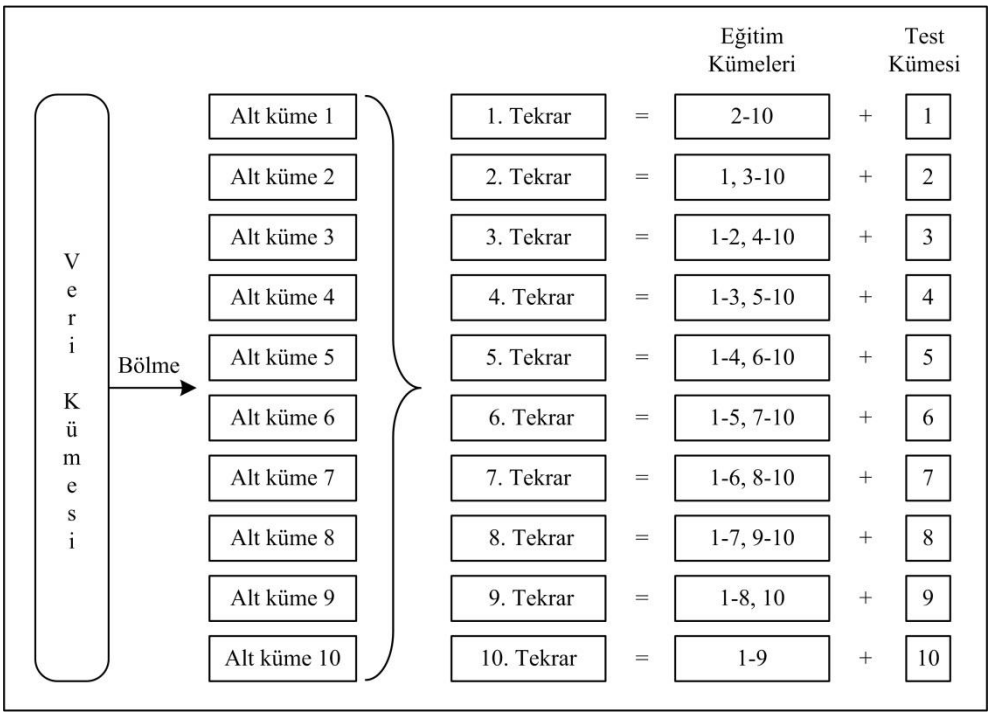
d(i,C) : *i* noktasının, *C* nolu kümede bulunan noktalara olan uzaklıklarının ortalamasını göstermektedir.

Denklemden de anlaşıldığı üzere, SD indeksi veri kümesindeki her bir noktanın diğer kümelerde bulunan noktalara göre kendi kümesinde bulunan noktalara ne kadar benzediğinin ölçüsüdür. Alabileceği değer -1 ile +1 arasında değişir. Eğer SD(i) değeri +1'e yakınsa, i noktası uygun kümeye ait, -1'e yakınsa yanlış kümeye ait yorumları yapılabilir.

K- ortalama kümeleme algoritması veri kümesinin kaç gruba ayrılacağını (*k*) giriş değeri olarak alınır. Sağlıklı bir kümeleme işlemi için en uygun *k* değerinin tespiti önemlidir. Bu aşamada SD indeksinden faydalanılır. Şöyle ki, düşünülen her bir *k* değeri için ayrı ayrı kümeleme yapılır. Her bir kümeleme için, veri kümesinde bulunan tüm verilerin SD değerlerinin ortalaması hesaplanır. Ortalama SD değeri en yüksek olan *k* uygun küme sayısı olarak tespit edilir. Bu düşünce, tez çalışmasında önerilen ve Bölüm 6'da açıklanan ön işlem algoritmasının temelini oluşturmaktadır.

2.1.6. *k*-kez çapraz doğrulama işlemi

Bu tez çalışmasında sunulan bulanık mantık sınıflandırıcısının performansını değerlendirmek için *k*-kez çapraz doğrulama işlemi kullanılmıştır. *k*-kez çapraz doğrulama işleminde, veri kümesi *k* adet alt kümeye bölünür. Bu alt kümelerden bir tanesi test kümesi olarak ayrılır. Geriye kalan *k-1* küme ise eğitim kümesi olarak kullanılır. Ayrılan test kümesi sistemin doğruluğunu test etmek için yani sistemi doğrulamak için kullanılır. Bu işlem *k* kez tekrarlanır. Böylece her bir verinin en az bir kez test kümesinde yer alması sağlanır. Sonuç olarak test kümeleri için *k* adet işlemin ortalaması alınarak sistemin başarısı hesaplanır. Şekil 2.4'de k=10 için çapraz doğrulama işleminin nasıl gerçekleştiği gösterilmiştir.



Şekil 2.4. 10-kez çapraz doğrulama işlemi

2.1.7. Wilcoxon işaretli-sıra testi

Wilcoxon işaretli-sıra testi (Wilcoxon signed-rank test), iki örneklem ortalamaları arasında anlamlı olan farklıları tespit etmeyi amaçlar. Parametrik bir istatistik test olan eşlenik t-testinin, parametrik olmayan alternatif testidir. Eğer iki algoritmanın çıkışlarının kıyaslanması için kullanılacaksa, test pratik olarak iki algoritmanın karşılıklı davranışlarını değerlendir. Örneğin N_{ds} veri kümesinin *i.* çıkışı

için iki algoritmanın ürettiği çıkış sonuçları arasındaki fark d_i olsun. Bu farklar mutlak değerlerine göre sıralanır ve her bir farka bir işaret değeri atanır. Mesela, ilk algoritmanın ikinci algoritmayı geçtiği durumlardaki sıra değerlerinin toplamı R^+ iken, ikinci algoritmanın ilk algoritmayı geçtiği durumlardaki sıra değerlerinin toplamı R^- olduğu kabul edilsin; d_i =0 olduğu durumlar varsa ve sayısı çift ise, ikiye bölünüp toplamlara eklenir, tekse bir durum ihmal edildir ve sayısının çift olması sağlanır. R^+ ve R^- değerlerinin hesaplanması sırası ile denklem 2.5 ve 2.6 da verilmiştir (Alcala ve ark., 2011)

$$R^+ = \sum_{d_i>0} sıra(d_i) + \frac{1}{2}\sum_{d_i=0} sıra(d_i) \quad (2.5)$$

$$R^- = \sum_{d_i<0} sıra(d_i) + \frac{1}{2}\sum_{d_i=0} sıra(d_i) \quad (2.6)$$

$$T = \min(R^+, R^-) \quad (2.7)$$

Eğer denklem 2.7'de verilen T değeri, Ek tablo-H 'da (Kartal,2006) verilen ve ilgili alana denk gelen wilcoxon işaretli-sıra testi kritik değerinden küçük ya da eşitse, sıfır hipotezi reddedilir, değilse kabul edilir.

Testin kullanımını bir örnek üzerinde açıklayalım. Mesela, SSMA (Garcia ve ark., 2008) ile bu tez çalışmasında önerilen GYAopt-aiNet algoritmalarının 20 adet veri kümesi üzerinde çalıştırıldıklarında elde edilen sınıflandırma doğruluk oranları arasında %95 anlam seviyesinde istatistiksel olarak anlamlı bir fark olup olmadığı Wilcoxon işaretli sıra testi kullanılarak araştırılsın. Çizelge 2.1'de testin hesaplama adımları verilmiştir. Öncelikle, sınıflandırma doğrulukları arasındaki farklar bulunmuş (sütun 4), daha sonra bu farkların mutlak değerleri hesaplanmış (sütun 5), farklar arasındaki işaretler tespit edilmiş (sütun 6) ve son olarak algoritmaların sıra değerleri bulunup, ilgili işaret ile çarpılıp son sütunda gösterilmiştir. Denklem 2.5 ve 2.6'da değerler yerine konulduğunda, R^+ =185, R^- =25 olarak bulunur. Denklem 2.7'ye göre T=25 olur. Ek tablo-H 'da (Kartal, 2006) 0.05 anlam seviyesi ve n=20 (veri kümesi sayısı) için okunan değer 52'dir. 25 sayısı 52'den küçük olduğu için H_0 hipotezi reddedilir ve algoritmalar arasında anlamsal bir fark vardır denilir. Hipoteze ait *p* değerleri ise Matlab 7.1 R14 programı kullanılarak hesaplanmıştır.

Çizelge 2.1. Wilcoxon işaretli-sıra testinin hesaplama adımları

Veri Tabanı No (N_{ds})	GYAopt-aiNet	SSMA	Fark (d_i)	Mutlak Fark	İşareti	İşaretli Sıra Değeri
1	0.747171	0.617187	0.129984	0.129984	+	20
2	0.798812	0.814914	-0.0161	0.016102	-	-3
3	0.72619	0.648673	0.077517	0.077517	+	17
4	0.75457	0.707921	0.046649	0.046649	+	7
5	0.993333	0.946667	0.046666	0.046666	+	8
6	0.981313	0.955267	0.026046	0.026046	+	5
7	0.798622	0.743229	0.055393	0.055393	+	11
8	0.715159	0.654514	0.060645	0.060645	+	13
9	0.930909	0.851515	0.079394	0.079394	+	18
10	0.764951	0.88149	-0.11654	0.116539	-	-19
11	0.63	0.56	0.07	0.07	+	15
12	0.939206	0.891693	0.047513	0.047513	+	9
13	0.818072	0.800898	0.017174	0.017174	+	4
14	0.845635	0.803213	0.042422	0.042422	+	6
15	0.929383	0.938013	-0.00863	0.00863	-	-1
16	0.825238	0.772063	0.053175	0.053175	+	10
17	0.79416	0.734141	0.060019	0.060019	+	12
18	0.74803	0.676075	0.071955	0.071955	+	16
19	0.958483	0.968286	-0.0098	0.009803	-	-2
20	0.970303	0.903939	0.066364	0.066364	+	14

Bu test, eşlenik t-testine göre biraz daha az kuvvetli bir test olmasına karşın, elde edilen sonuçların dağılımı ile ilgili bir ön şart veya bilgi gerektirmediği için kullanışlıdır. Örneğin eşlenik t-testini uygulamak için, verinin dağılımın normal olduğundan emin olmak gerekirken, Wilcoxon'da böyle bir zorunluluk yoktur.

2.2. Kullanılan Veri Kümeleri

Bölüm 6'da sunulan bulanık sınıflandırıcı sisteminin performansının testi için Irvine California Üniversitesinin (UCI) makine öğrenmesi veri deposunda (Blake and Merz, 1998) bulunan ve herkese açık 20 farklı veri kümesi kullanılmıştır. Bu veri kümelerinin genel özellikleri Çizelge 2.2'de verilmiştir.

Bu çizelgede, sırası ile her bir veri kümesinin kısa adını (ka), tam ismini, kaç örnek içerdiğini, her bir örneğin kaç niteliği bulunduğunu, her bir niteliğin gerçek sayı mı, tam sayı mı yoksa sözel mi olduğunu (**G**erçek / **T**am / **S**özel) ve son olarak veri kümesinin kaç sınıfa ayrıldığını göstermektedir. Ayrıca veri kümesinin eksik değerlere sahip olup olmadığı da belirtilmiştir. Şöyle ki eksik kayıta sahip örnekler veri

kümesinden çıkarılmış ve veri kümesinin esas örnek sayısı Mammograph, Wisconsin ve Cleveland veri kümelerinde olduğu gibi parantez içerisinde verilmiştir.

Çizelge 2.2. Kullanılan UCI Veri Kümeleri ve Özellikleri

Ka	İsmi	#Örnek	#Nit. (G/T/S)	#Sınıf
hab	Haberman	306	3 (0/3/0)	2
iri	Iris	150	4 (4/0/0)	3
bal	Balance Scale	625	4 (4/0/0)	3
new	New-thyroid	215	5 (4/1/0)	3
mam	Mammograph	830 (961)	5 (0/5/0)	2
bup	Liver disorder (bupa)	345	6 (1/5/0)	2
eco	Ecoli	336	7 (7/0/0)	8
app	Appendicitis	106	7 (7/0/0)	2
pim	Pima	768	8 (8/0/0)	2
gla	Glass	214	9 (9/0/0)	7
tic	Tic-tac-toe	958	9 (0/0/9)	2
wis	Wisconsin	683 (699)	9 (0/9/0)	2
mag	Magic	1902	10 (10/0/0)	2
pag	Page-blocks	548	10 (4/6/0)	5
cle	Cleveland	297 (303)	13 (13/0/0)	5
zoo	Zoo	101	16 (0/0/16)	7
veh	Vehicle	846	18 (0/18/0)	4
ion	Ionosphere	351	33 (32/1/0)	2
spe	Spectfheart	267	44 (0/44/0)	2
son	Sonar	208	60 (60/0/0)	2

2.3. Kullanılan Cihazlar ve Programlar

Bölüm 5'deki çalışmalar esnasında 1 Gb rastgele erişimli belleğe ve Intel Pentium M 1.8 Ghz işlemciye sahip bir bilgisayar kullanılmıştır. Bölüm 6'da anlatılan deneysel çalışmalar ise 4 Gb rastgele erişimli belleği ve Intel Core i7 işlemcisi bulunan bir bilgisayar üzerinde gerçekleştirilmiştir. Tez çalışması sürecinde yazılan programlar Matlab 7.1 R14 programlama dili kullanılarak kodlanmıştır.

Bölüm 6'da önerilen GYAopt-aiNet algoritması Bölüm 2.2'de listesi verilen 20 farklı sınıflandırma veri kümesi üzerinde test edilmiş ve sonuçlar 22 farklı sınıflandırma yöntemi ile kıyaslanmıştır. Kıyaslamak için kullanılan sınıflandırma yöntemlerinin sonuçları, KEEL (http://www.keel.es/, erişim tarihi: 23.02.2011) isminde ticari

olmayan ve sınıflandırma, regresyon, kümeleme v.b. alanlarda çalışan araştırmacıların çalışmalarında faydalanabilmesi için altı farklı İspanyol araştırma grubun işbirliği ile geliştirilen açık kaynak kodlu bir yazılım aracı kullanılarak elde edilmiştir. Kıyaslama için kullanılan algoritmalar türlerine gruplanmış ve listesi Çizelge 2.3'de verilmiştir.

Çizelge 2.3. Kıyaslama için kullanılan algoritmalar

Grubu	Kısa Adı	Uzun Adı	Referansı
Evrimsel Bulanık Kural Öğrenme Algoritmaları (Grup-1)	AdaBoost	Fuzzy-Rule-Learning, AdaBoost algorithm	Del Jesús ve ark., 2004
	GCCL	Fuzzy-Rule-Learning Algorithm	Ishibuchi ve ark., 1999
	SGERD	Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data	Mansoori ve ark., 2008
	SLAVE	Iterative Rule Learning of Fuzzy Rules with Feature Selection	Gonzalez ve Perez, 1999
	IGA	Intelligent Genetic Algorithm for Edition	Ho ve ark., 2002
	SSMA	Steady-State Memetic Algorithm for Instance Selection	Garcia ve ark., 2008
Evrimsel Kural Öğrenme Algoritmaları (Grup-2)	HIDER	Hierarchical Decision Rules	Aguilar-Ruiz ve ark., 2003
	CORE	Co-evolutionary Rule Extractor	Tan ve ark., 2006
	GASSIT	Genetic-Algorithm-based claSSIfier SysTem	Bacardi ve Garrell, 2003
	ILGA	Incremental learning with genetic algorithms	Guan ve Zhu, 2005
	CPSO	Constricted Particle Swarm Optimisation	Sousa ve ark., 2004
	MPLCS	Memetic Pittsburgh Learning Classifier System	Bacardit ve ark., 2009
Karar Ağaçları (Grup-3)	DT-GA	Hybrid Decision Tree - Genetic Algorithm	Cantu-Paz ve Kamath,2003
	TARGET:	Tree Analysis with Randomly Generated and Evolved Trees	Gray ve Fan, 2008
	C 4.5	C4.5 Decision Tree	Quinlan, 1993
	C 4.5 Rules	Classification model by covering rules (based on decision trees)	Quinlan, 1995
Yapay Sinir Ağları (YSA), İstatistiksel Yöntemler, SVM ve KNN (Grup-4)	LVQ	Learning Vector Quantisation for Classification Problems (YSA)	Bezdek ve Kuncheva, 2001
	MLP-BP	Multilayer perceptron for classification problems (YSA)	Rojas ve Feldman, 1996
	KERNEL	Kernel Classifier (İstatistiksel Sınıflandırıcı)	McLachlan, 2004
	LINEAR-LMS	Least Mean Square Linear Classifier (İstatistiksel Sınıflandırıcı)	Rustagi, 1994
	SVM	Classification model by means of C-Support Vector Machine	Cortes ve Vapnik, 1995
	KNN	K-Nearest Neighbours Classifier	McLachlan, 2004

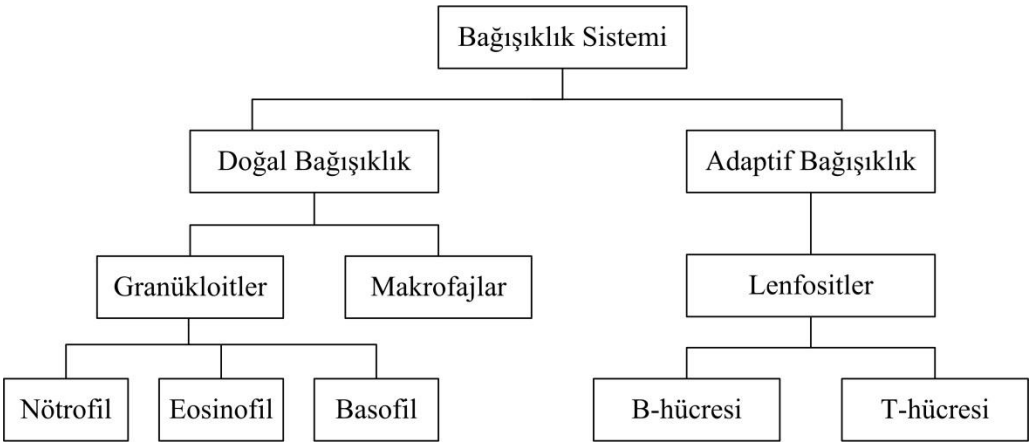
3. YAPAY BAĞIŞIKLIK SİSTEMLERİ

İnsan bağışıklık sistemi virüs, bakteri gibi neredeyse sınırsız sayıda olan yabancı saldırganlara karşı doğal, hızlı ve etkili cevap verme yeteneğine sahip karmaşık bir ağ yapısı olarak tanımlanabilir. Bağışıklık sistemi örüntü tanıma, öğrenme, hafıza edinme, farklılıkların genelleştirilmesi, gürültü toleransı, genelleme ve optimizasyon gibi pek çok işlemi yerine getirme kabiliyetine sahip hücrelerden meydana gelir. Bağışıklık sistemini daha iyi anlamak ve mühendislik problemlerinin çözümünde kullanabilmek için bu biyolojik prensiplere dayanan yeni hesaplama teknikleri geliştirilmektedir. Bu amaçla insan bağışıklık sisteminden esinlenerek yapay bağışıklık sistemleri tanımlanmıştır. Bu bölümünde ilk olarak insan bağışıklık sisteminin yapısı, temel bileşenleri ve özelliklerinden bahsedilmiştir. Daha sonra yapay bağışıklık sistemi tanıtılmış ve tezde çalışmasında kullanılan algoritmalar anlatılmıştır.

3.1. İnsan Bağışıklık Sistemi

3.1.1. İnsan bağışıklık sisteminin yapısı

İnsan bağışıklık sistemi, dışarıdan gelebilecek saldırılara karşı vücudu koruma yeteneğine sahip kuvvetli bir savunma mekanizmasıdır ve doğal ve adaptif olmak üzere ikiye ayrılır. Şekil 3.1'de de görüldüğü gibi doğal bağışıklık sisteminin elemanları granükloitler (granulocytes) ve makrofajlar (macrophages) iken adaptif bağışıklık sisteminin elemanları lenfositlerdir (lymphocytes) (de Castro, L. N. ve Timmis, 2002a).



Şekil 3.1. Bağışıklık Sisteminin Sınıflandırılması ve Temel elemanları (de Castro ve Timmis, 2002a)

İnsan vücudunun doğumla birlikte kazandığı bağışıklık doğal bağışıklık olarak adlandırılır. İnsan vücudunun sahip olduğu bu doğuştan gelen bağışıklık pek çok mikrobu ilk karşılaşmada tanır ve yok eder. Çalışma şekli iki moleküler yapının birbirini tanıyarak tamamlamasına dayanır. Patojen ismi verilen mikrop üzerinde bulunan yapı PAMP (pattern apattern associated molecular pattern) olarak adlandırılır. PAMP'ları tanıyan ve onun tamamlayıcısı olarak kodlanmış vücutta üretilen yapılara ise PRR (pattern recognition receptors) adı verilir. PAMP'lar sadece mikroplar tarafından üretilir. Asla vücutta üretilmez. O yüzden bir PRR, bir PAPM'ı tanıdığına dair bir sinyal üretirse bu vücuda mikrop girdiği anlamına gelir. Doğal bağışıklık sisteminin diğer önemli bir elemanı da APC hücreleridir. APC'ler doğal bağışıklık sistemi tarafından tanınamayan yabancı patojenleri T-hücrelerine sunmakla görevlidir. Böylece antijen olarak da isimlendirilen bu yapılara karşı adaptif bağışıklık cevabının başlatılması sağlanır.

Adaptif bağışıklık cevabı, tespit edilmiş bir antijeni yok etmekle görevli lenfosit hücrelerin üretilmesi işlemidir. B ve T olmak üzere iki çeşit olan bu hücreler, üzerlerinde bulunan reseptörler ile daha önceden hiç karşılaşmadıkları antijenleri tanıyabilirler. Diğer bir özellikleri ise tanıdıkları bu antijenler için bağışıklık hafızası geliştirme özelliğine sahip olmalarıdır. Böylece vücuda, daha önceden tanınmış bir antijen tekrar gelirse aktivasyon süresinin kısaltılması sağlanır. Bu hücreler ile ilgili daha detaylı açıklamalar bir sonraki bölümde yapılmıştır.

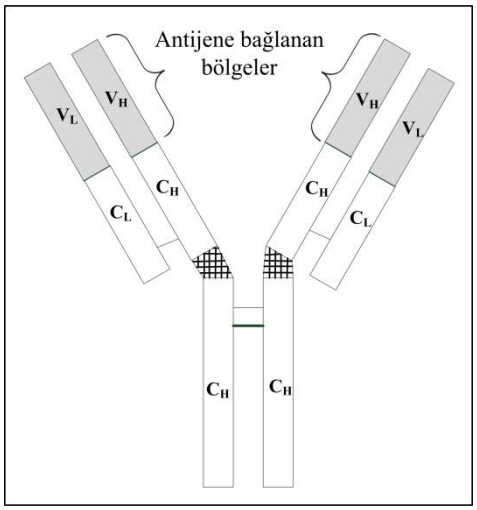
3.1.2. Bağışıklık hücreleri

Bağışıklık sistemini meydana getiren lenfoit organlar vücudun her yerine dağılmıştır. Bunlardan başlıcaları bademcik, lenf bezi, lenf damarları, kemik iliği, lenf düğümleri, timüs ve dalaktır. Bağışıklık hücreleri kemik iliğinde üretilir, lenf düğümlerinde timüste depolanır, lenf damarları sayesinde dokulara giderler. Üretilen hücrelerden bir kısmı vücudun genel savunmasından sorumlu iken diğerleri özel patojenlerle savaşmak için eğitilir. Etkili bir şekilde çalışmaları için hücreler arasında devamlı bir işbirliği gereklidir. Bu bölümde lenfositler, fagositler ve granülositlerden bahsedilecektir.

Lenfositler bağışıklık sisteminde önemli bir sorumluluğa sahip olan küçük lökositlerdir. Lenfositlerin iki ana tipi vardır. Bunlar B lenfositleri (veya B hücresi) ve T lenfositleridir (veya T hücresi). Lenfositlerin çoğu sadece özel bir aktivasyonun

çoğalması için gerekli ayrı antijenlerle etkileşime girdikten sonra fonksiyonel aktivite sergileyen küçük dinlendirici hücreler şeklinde düzenlenir. B ve T lenfositlerinin yüzeylerinde belirli antijenik yapılar için oldukça özelleşmiş reseptörler bulunur.

B hücrelerinin temel fonksiyonu bakteri, virüs ve tümör hücreleri gibi dış kaynaklı proteinlere tepki vermek için antikor üretmek ve salgılamaktır. Her B hücresi belirli bir antikor üretmek için programlanmıştır. Antikorlar bir başka belirli proteini tanıyan ve ona bağlanan belirli proteinlerdir. Antikor üretimi ve bağlanması genellikle maddeyi kuşatıp sindirme veya yok etme, diğer hücreleri öldürmek için sinyal verme şeklinde olur. Şekil 3.2'de bir antikorun yapısı gösterilmiştir (De Castro ve Von Zuben, 1999).



Şekil 3.2. Antikor'un Yapısı

Timüs içerisinde olgunlaşan T hücrelerinin görevi hücrelerin hareketlerini düzenlemek ve direkt olarak vücuda girmiş mikroplara saldırmaktır. T lenfositleri üç temel alt sınıfa ayrılabilirler. Bunlar T hepler hücreleri (Th), sitotoksik (cytotoxic-killer) T hücreleri ve bastırıcı (suppressor) T hücreleridir.

Aynı zamanda CD4 veya T4 hücreleri olarak da bilinen T hepler hücreleri veya basit olarak Th hücreleri diğer bağışıklık hücrelerinin yani B hücrelerinin, T hücrelerinin geri kalan türlerinin, makrofajların ve doğal killer (NK) hücrelerinin aktivasyonu için gereklidir. Killer T hücrelerinin veya cytotoxic T hücrelerinin mikrobik saldırganları, virüsleri veya kanserli hücreleri yok etme yeteneği vardır. Aktif hale geldikten sonra yabancı hücreleri kuşatma, onların yüzey zarına nüfuz edip yıkımlarına sebep olma ve çıkan zararlı kimyasal maddeleri sindirme özellikleri vardır. CD8 hücreleri olarak da adlandırılan Bastırıcı (supressor) T lenfositleri bağışıklık

tepkisinin devamını sağlamada önemli bir rol üstlenirler. Esas görevleri bağışıklık hücrelerinin kontrolsüz çoğalmasını engellemek yani onlara baskılayıcı bir etki uygulamaktır. Böylece bağışıklık, alerjik reaksiyonlar ve otoimmune hastalıklar kontrol altında tutulabilir (De Castro ve Von Zuben, 1999).

Fagositler ("hücre yiyenler") mikro organizmaları ve antijenik tanecikleri yeme ve sindirme yeteneği olan beyaz kan hücreleridir. Bazı fagositlerin aynı zamanda lenfositler için antijen sunma yeteneği vardır. Bu fagositler antijen sunan hücreler (APCs) olarak adlandırılmaktadır. Önemli fagositler monositler ve makrofajlardır. Monositler kan içinde dolaşırlar ve makrofaj ("büyük yiyiciler") haline geldikten sonra dokulara geçerler. Makrofajlar çeşitli fonksiyonları olan çok yönlü hücrelerdir. Onlar T lenfositleri için antijen sağlarlar, onları yerler ve sindirirler. Bağışıklık tepkisinin başlamasında da önemli bir role sahiptirler.

Nötropiller ve ösonopiller aynı zamanda makrofajlara benzer fonksiyonları olan granülositlerdir. Diğer bir granülosit olan basopiller ise kan dolaşımda yer alırlar, güçlü kimyasal maddelerle doldurulmuş tanecikler içerirler. Alerjik durumlara yanıt vermede önemlidir. İçlerinde yer alan kimyasal maddeler mikro organizmaları yok etmede görev alır (De Castro ve Von Zuben, 1999).

3.1.3. Bağışıklık sistemi vücudu nasıl korur?

Bağışıklık sistemi vücudu, dışardan gelen saldırılara karşı koruyan hücre ve moleküllerin karışık bir kümesinden oluşur. Vücudumuz bütün bağışıklık tepkilerinin nihai hedefi olan antijenin (Ag) tehdidine karşı adaptif bağışıklık sistemi tarafından korunmaktadır. Şekil 3.3'de bu savunma mekanizmasının basitleştirilmiş gösterimi bulunmaktadır. Şekilde, makrofajlar gibi antijen sunma yeteneğine sahip APC hücrelerinin yiyebilecekleri ve sindirebilecekleri antijenleri bulmak için vücuda dolaştığını ve bulduklarında ise onları *antijenik peptitlerine* ayırarak parçaladıklarını göstermektedir. Şekilde görülen bu adımların açıklamaları aşağıda verildiği gibidir.

(I) Antijenik peptitler MHC (major histocompatibility complex) moleküllerine bağlanır, APC hücre yüzeyi ile birleşerek MHC/Peptit birleşimini oluştururlar.

(II) T hücreleri, farklı MHC/Peptit birleşimlerini tanımak için her biri aktif olan reseptör moleküllerine sahiptirler.

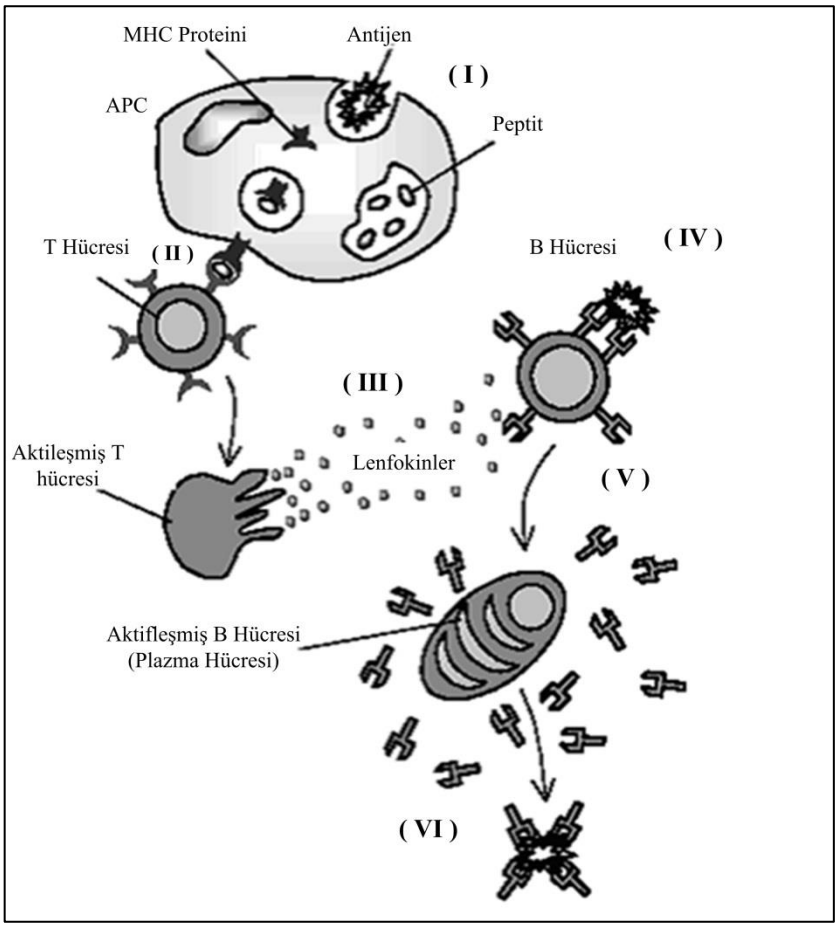
(III) T hücresi MHC/peptit birleşimini tanıdıktan sonra aktive olur ve parçalanarak lenfokinler ve ya kimyasal sinyaller salgılar. Bunlar diğer bağışıklık sistemi elemanlarını uyarır ve aktivasyona katılmasını sağlar.

(IV) T hücrelerinin tersine B lenfositleri MHC moleküllerinin yardımı olmaksızın serbest durumdaki antijenleri tanıma yeteneğine sahip reseptörlere sahiptir.

(V) B hücrelerinin reseptörleri tarafından uyarı sinyali alındığında B hücreleri *plazma hücrelerine* dönüşerek çok sayıda antikor molekülü salgılarlar.

(VI) Serbest kalan bu antikorlar patojenleri etkisiz hale getirmede kullanılır ve daha sonra bunlar enzimler veya temizleme hücreleri kullanılarak yok edilirler.

Aktive olan bu B ve T hücrelerinin bir kısmı değişerek *hafıza hücrelerini* oluştururlar. Bu hücreler organizma içinde dolaşmaya devam edecek ve gelecekte olabilecek aynı veya benzer antijene daha iyi bir bağışıklık cevabı üretmeyi garantileyeceklerdir (de Castro ve Timmis, 2002).



Şekil 3.3. Bağışıklık Sisteminin basit aktivasyon mekanizması (de Castro ve Timmis, 2002)

3.1.4. Klonal seçme teorisi

Klonal seçme teorisi, antijenik bir uyarıcıya karşı adaptif bağışıklık cevabının temel özelliklerini açıklamak için kullanılan bir teoridir. Klonal seçme prensibi hem T hücreleri hem de B hücrelerine uygulanır. Ancak B ve T hücrelerinin klonal çoğalması sırasındaki temel fark, B hücrelerinin çoğalma boyunca somatik mutasyondan etkilenmesi buna karşın T hücrelerinin etkilenmemesidir.

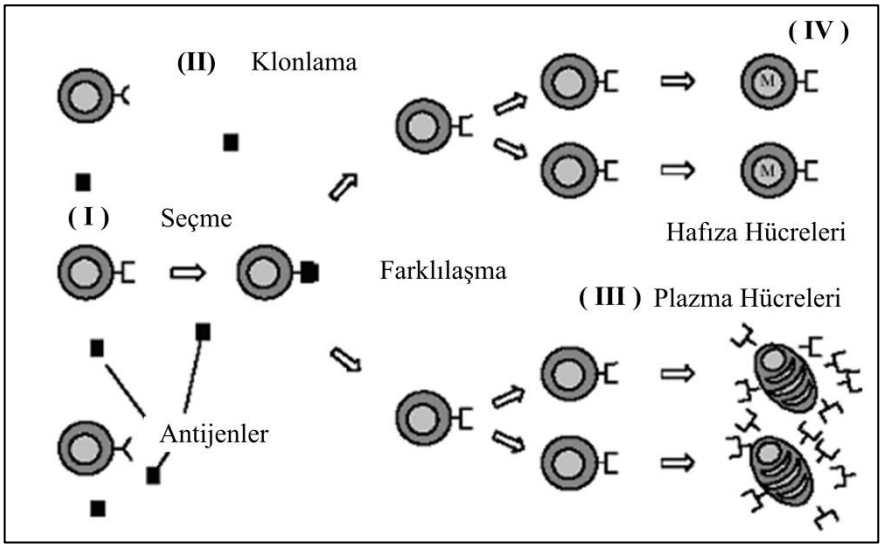
Bir birey antijene maruz kaldığı zaman, B lenfosit hücreleri antikor üreterek tepki verir. Her hücre antijenin özelliğine bağlı olarak antikorun sadece bir türünü salgılar. Şekil 3.4'de klonal seçme prensibinin basitçe gösterimi bulunmaktadır.

(I) B hücresi üzerindeki antijenik reseptörler bir antijen ile bağlanır ve T_H gibi yardımcı bir hücrenin aracılığı ile ikinci bir sinyal üretir. Bu sinyal B hücresinin o antijen için uyarılmasını sağlar.

(II) Bu uyarılma sonucu B hücresi bölünür ve olgunlaşarak antikor üreten plazma hücrelerine dönüşür.

(III) Plazma hücreleri hızlı antikor salgılarken, düşük bir oranla da olsa hızla bölünen B hücreleri de antikor salgılarlar. B hücreleri plazma hücrelerinden başka uzun ömürlü olan hafıza hücrelerine de dönüşebilirler.

(IV) Hafıza hücreleri kan aracılığıyla vücuda dolaşır ve antikor üretmezler. Ancak ikinci bir antijen uyarısı ortaya çıkarsa, hızla plazma hücrelerine dönüşürler ve yüksek duyarlılıklı antikor üretmeye başlarlar (de Castro, ve Timmis, 2002).



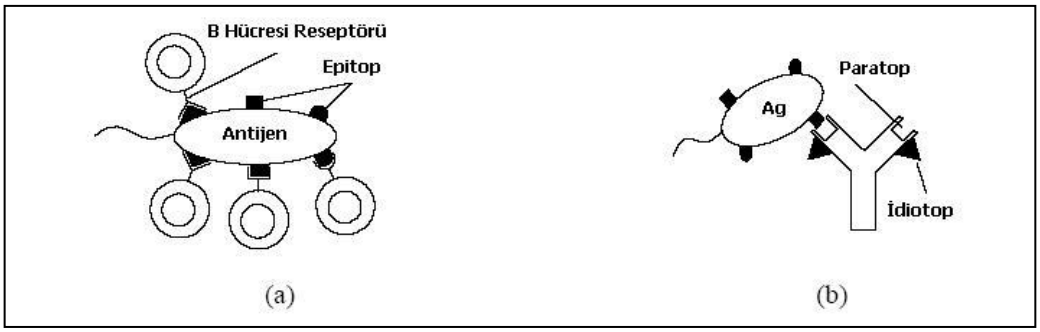
Şekil 3.4. Klonal Seçme prensibi

3.1.5. Bağışıklık ağ teorisi

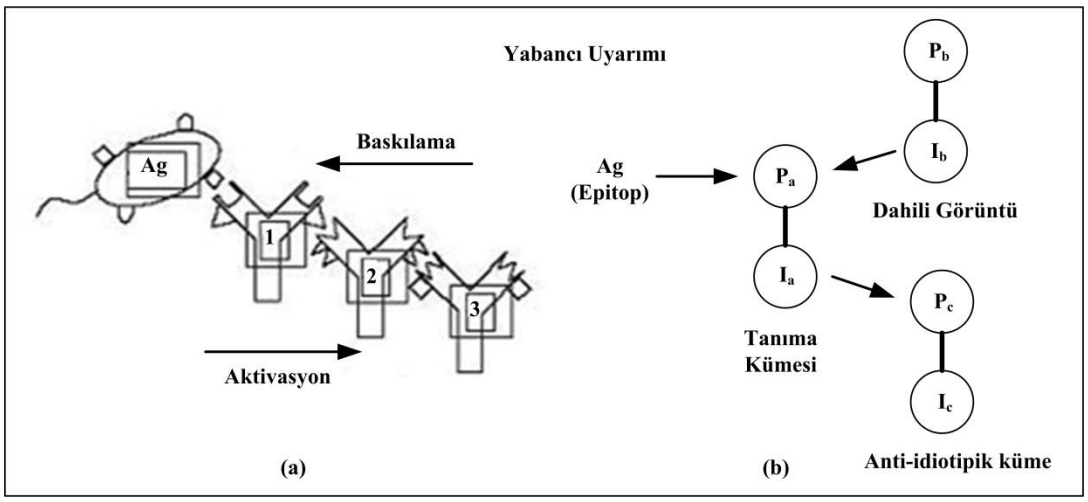
Jerne (1974), bağışıklık ağ teorisini ilk defa önerdiğinde, lenfosit aktivitelerini, doğal antikor üretimini, repertuar seçimini, tolerans ve kendi/kendi-değil ayırımını, bağışıklık sisteminin hafızasını ve evriminin bakış açısını hipotezlemeyi amaçlamıştır. Jerne'e göre bağışıklık sistemi antijenlerin yokluğunda bile birbirini tanıyabilen hücrelerin düzenlenmiş bir ağından meydana gelmiştir.

Bağışıklık ağını tarif etmek için Jerne'in kullanılmasını önerdiği gösterim Şekil 3.5'de verildiği gibidir. Antijen üzerinde bulunan ve antikorun reseptörleri sayesinde tanıyıp antijene bağlandıkları kısma *epitop* ismi verilir. Bir antikorun değişken bölgesinde yer alan ve tamamlayıcı epitopunu tanımakla görevli alan kısmı da *paratop* olarak adlandırılır. Değişken bölgede olduğu için bir antikorun paratobunun şekli değişebilir. *İdiotop* ise antikorun paratobunun çevresinde veya içerisinde yer alan antikoru tanımlayan kısma verilen addır. Bir antijen birden fazla epitopa sahip olabilirken antikorlar tek tip epitoba sahiptir ve burası *idiotop* olarak isimlendirilir (de Castro ve Von Zuben, 2001).

Bağışıklık sistemi biçimsel olarak, idiotopların kümelerini tanıyan paratopların ve paratop kümeleri tarafından tanınan idiotopların büyük ve karmaşık bir ağı olarak tanımlanabilir. Böylece her eleman tanıyabilmesinin yanı sıra tanına da bilir. Bu özellik ağın kurulmasında önderlik eder. Verilen antikor bir epitopu veya idiotopu tanıdıktan sonra, üretilen tanıma sinyaline pozitif veya negatif olarak cevap verebilir. Pozitif cevap hücre aktivasyonu, hücre çoğalması ve antikor salgılanması ile sonuçlanırken, negatif cevap baskılama oluşmasına sebebiyet verir. Şekil 3.6(a)'da bağışıklık modelinin pozitif ve negatif cevabı görülür (de Castro ve Von Zuben, 2001).



Şekil 3.5. Bağışıklık Ağ teorisinin basit kavramları. (a) Farklı B hücreleri tarafından tanınan çeşitli epitoplara sahip bir antijendir. (b) Antikorun V bölgesi ile paratop ve idiotopun gösterimidir.



Şekil 3.6. Bağışıklık Ağ teorisinin gösterimi. (a) Bir antijenin 1 nolu sınıfın bir antikorunu uyarması, bu antikorun 2 nolu sınıfın antikorunu uyarması vb. (b) İdiotipik ağ yapınsın ayrıntılı gösterimi

Şekil 3.6(b)'de bağışıklık ağ teorisinin davranışı gösterilmiştir. Bağışıklık sistemi bir antijen (Ag) tarafından uyarıldığında, bu antijenin epitopu farklı paratoplar tarafından farklı derecelerde tanınır. Bu paratopların oluşturduğu kümeye p_a adı verilir. Bu p_a paratopları, Ab ve reseptör moleküllerinin paratoplarıdır. Bu kümedeki her paratopun tanıdığı idiotoplar vardır. Böylelikle p_a paratop kümesinin tanıdığı bir idiotop kümesi yani i_a ortaya çıkar. P_ai_a sembolü, Ag'i tanıyan Ab molekülleri ile potansiyel olarak ona yanıt oluşturabilecek lenfositlerin oluşturduğu toplam kümeyi temsil eder. Bağışıklık ağında p_a kümesindeki her paratop, bir idiotop kümesini tanır. Böylece toplam p_a kümesi oldukça geniş bir idiotop kümesini tanımış olur. Buradaki i_b idiotop kümesi, Ag antijenini tanıyan p_a kümesi tarafından tanınan (eğer varsa) diğer epitopların imajını (internal image) temsil eder. p_b kümesi, i_b kümesini tanıyan diğer paratopların oluşturduğu kümedir. Bunun yanında, p_ai_a kümesindeki her idiotop, bir paratop kümesi tarafından tanınır. Bu da toplam i_a kümesinin daha büyük bir p_c paratop kümesi tarafından tanınması anlamına gelir. Yine benzer şekilde, bu p_c kümesindeki her bir paratop bir i_c idiotop kümesi tarafından tanınır ve toplam p_c kümesi daha geniş bir i_c kümesi ile birlikte ifade edilir (p_ci_c). Fakat bu sefer bu kümeye, tanıyıcı küme yerine anti-iditopik küme adı verilir çünkü bu küme asıl Ag'i değil, idiotop'ları tanımaktadır. Bu yöntemi bu şekilde devam ettirirsek, ağ içerisinde bir önce tanımlanan kümeleri tanıyan ve yine bu kümeler tarafından tanınan oldukça büyük kümelere ulaşılabilir (de Castro ve Von Zuben, 2001). Ağ yaklaşımı özellikle bilgisayar araçlarını geliştirmek için caziptir. Çünkü öğrenme ve bellek, hücre popülasyonunun boyutu ve çeşitliği gibi gelişen özelliklerin bir tanımını sağlamaktadır.

3.2. Yapay Bağışıklık Sistemleri

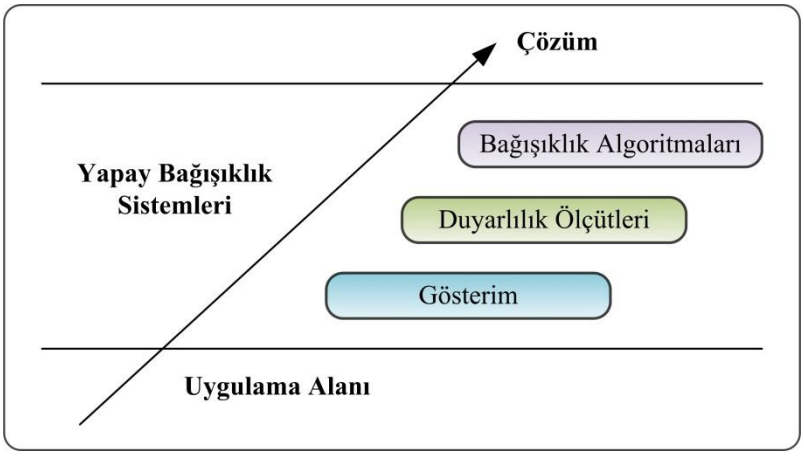
Son yıllarda biyolojiden esinlenerek geliştirilmiş sistemler hakkında yapılan çalışmalarda büyük bir miktarda artış gözlemlenmiştir. Bunların arasında yapay sinir ağları (YSA), evrimsel hesaplama, DNA hesaplama ve yapay bağışıklık sistemleri sayılabilir. Yapay bağışıklık sistemleri (YBS), doğal bağışıklıktan esinlenen adaptif sistemlerdir. Problemlerin çözümünde teorik bağışıklık fonksiyonları ve prensipleri gözlemlenerek oluşturulmuş YBS modellerini kullanırlar (de Castro ve Timmis, 2002a).

Bağışıklık sistemi uygulamalarının başlıca kullanım alanları örüntü tanıma, hata ve anormallik tespiti, veri analizleri (veri madenciliği, sınıflandırma vb.) etmen tabanlı sistemler, makine öğrenmesi, arama ve optimizasyon problemleri ve güvenlik ve bilgi sistemleridir.

Biyolojiden esinlenen diğer algoritmalarda olduğu gibi yapay bağışıklık sistemlerinde de tasarım aşamasında ilk olarak aşağıda verilen temel soruları cevaplamalıdır.

1. Sistemdeki elemanlar nasıl temsil edilecektir?
2. Bireylerin çevre ve birbirleri ile olan ilişkilerini değerlendirmek için hangi değerlendirme mekanizmaları kullanılacaktır?
3. Sistem dinamiğini yönetmek için adaptasyon prosedürlerinin nasıl belirlenecektir? Yani sistem davranışı zamanla neye göre değişecektir?

Yukarıdaki soruların cevaplanma süreçleri, aslında yapay bağışıklık sistemi oluşturulurken izlenmesi gereken katmanlı bir yapının aşamalarıdır.

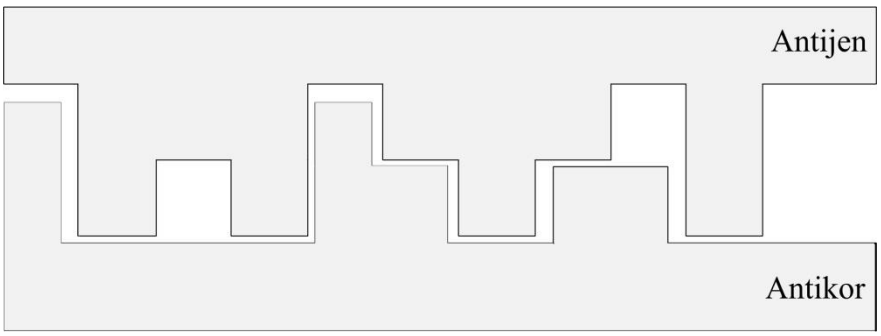


Şekil 3.7. YBS' nin katmanlı yapısı (de Castro ve Timmis, 2002a)

Şekil 3.7'de verilen bu yapıya göre tasarım, bağışıklık hücre ve moleküllerin gösteriminde kullanılacak soyut modelin belirlenmesi ile başlar. Perelson ve Oster'in (1979) sunduğu *Şekil Uzayı* en çok kabul gören ve en sık kullanılan yaklaşımdır. İkinci adım ise YBS elemanları arasındaki ilişkileri değerlendirmek için literatürde sunulan çeşitli ölçütlerden uygun olanını seçmektir. En son adımda ise sistemdeki dinamiği yönetmek için adaptasyon prosedürü olarak amaca uygun bir YBS algoritması seçilir.

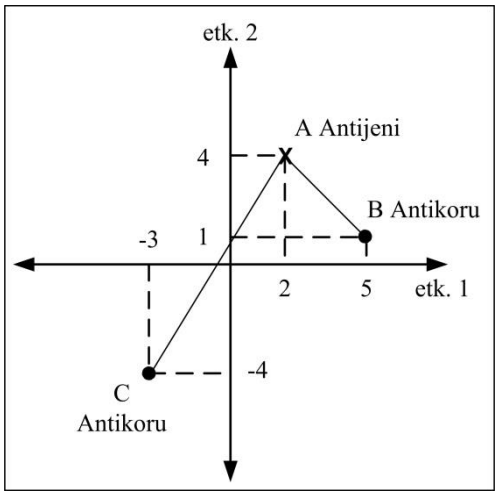
3.2.1. Şekil uzayı

Doğal bağışıklık sisteminde bir antijenin bir antikor tarafından tanınması için, yüzeylerinin birbirlerini tamamlayacak şekilde bağlanmaları gerekir. Bu yüzden moleküller üzerinde geniş *tamamlayıcı bölgelere* ihtiyaç duyulur. Şekil 3.8'de antijen ve antikorların üzerinde bulunan tamamlayıcı bölgeye bir örnek verilmiştir. Antijen veya antikorun tamamlayıcı bölgesinde yer alan her bir özelliğin oluşturduğu kümeye de antijen veya antikorun *geneleştirilmiş şekli* ismi verilir.



Şekil 3.8. Antijen ve antikorun tamamlayıcı bölgeleri

Şekil 3.8'den de görüldüğü üzere iki bağışıklık elemanı arasında bir etkileşimin gerçekleşmesi için *tamamlayıcılık* gereklidir. Perelson ve Oster'in (1979) bağışıklık elemanları arasındaki etkileşimi modellemek için sunduğu *Şekil Uzayı* yapısında bu *tamamlayıcılık* durumu *uzaklık* kavramına karşılık gelir. Örneğin şekil uzayında yer alan iki birey arasındaki mesafe ne kadar az ise, aralarındaki tamamlayıcılık o kadar fazladır. Şekil uzayındaki herbir eksen, antijen ve antikorlar arasındaki bir etkileşimi temsill eder. Mesela bir *A* antijeni ile bir *B* ve C antikorları arasında iki çeşit fizikokimyasal etkileşim gerçekleşsin. Her bir fizikokimyasal etkileşim bir eksen ile temsil edileceğinden dolayı Şekil 3.9'da gösterilen iki boyutlu şekil uzayı, *A* antijeni ile *B* ve *C* antikorlarını göstermek için kullanılabilir.



Şekil 3.9. Şekil Uzayı Gösterimi (de Castro ve Von Zuben, 1999)

Şekil 3.9'un yorumu şu şekilde yapılır: *A* antijeni ile *C* antikoru arasındaki uzaklık, *B* antikorundan fazla olduğu için, *A* antijeni ile *C* antikoru arasındaki etkileşimin şiddeti, *A* antijeni ile *B* antikoru arasındaki etkileşimin şiddetinden daha fazladır.

3.2.2. Duyarlılık hesaplaması

Matematiksel olarak, *S* şekil uzayındaki herhangi bir (*m*) molekülün genelleştirilmiş şekli *L* uzunluğunda bir nitelik dizisi olarak düşünülür. Yani $m=<m_1,m_2,\ldots m_L>$ dizisi, *L*-boyutlu bir şekil uzayındaki bir noktayı temsil eder. Bu dizi tam ve ya ondalık sayılardan, bit veya sembollerden oluşabilir. Bu niteliklerin türü YBS problemine göre seçilir. Bu tanımlamada önemli olan şey hangi ölçüt veya ölçütlerin, niteliklerin etkileşim kalitesini yani duyarlılığını belirlemede kullanılacağıdır.

Duyarlılık, bağışıklık birimlerinin birbirini tanıma kalitesini ifade eder. Çözülmesi gereken problemin türüne göre, duyarlılık hesabında kullanılacak yöntem seçilir. Genellikle antijen ile antikor arasındaki mesafeyi bulan uzaklık ölçütleri duyarlılık hesabında kullanılır. Yukarıda da bahsedildiği gibi iki eleman birbirinden ne kadar uzak ise birbirlerini o kadar tamamlayacağı düşünülür. Elemanların gösterimine göre kullanılacak uzaklık ölçütü de değişir. Sıklıkla kullanılan uzaklık ölçütleri Çizelge 3.1'de verilmiştir.

Sistemdeki bir antikor'un $Ab=<Ab_1, Ab_2,\ldots\ldots, Ab_L>$ ve bir antijen'nin $Ag=<Ag_1, Ag_2,\ldots\ldots, Ag_L>$ şeklinde gösterildiği kabul edilmiştir.

Çizelge 3.1. Duyarlılık ölçütü için kullanılabilecek Uzaklıklar

Uzaklık Ölçütü	**Formülü**
Öklid Uzaklığı	**D=** $\sqrt{\sum_{i=1}^{L}(Ab_i - Ag_i)^2}$
Manhattan Uzaklığı	**D=** $\sqrt{\sum_{i=1}^{L}\lvert Ab_i - Ag_i\rvert}$
Hamming Uzaklığı	D= $\sum_{i=1}^{L}\delta$, δ={1 eğer $Ab_i \neq Ag_i$, 0 diğer durumlarda.}

3.2.3. Yapay bağışıklık algoritmaları

Problemlerin çözümü için tasarlanan sistemlerin dinamiğini yönetmek için amaca uygun çeşitli YBS algoritması geliştirilmiştir. Bu YBS algoritmaları geliştirilirken baz alınan modelleri dört ana başlık altında toplayabiliriz.

- Kemik İlik (Bone Marrow) Modeli: Hücre ve moleküllerin repertuarlarının üretilmesini modeller.
- Timmus (Thymus) Modeli: Kendi/kendi-değil ayrımını yapabilme kabiliyetine sahip hücre ve moleküllerin repertuarlarının üretilmesini modeller.
- Klonal Seçme (Clonal Selection) Modeli: Bağışıklık sistemi bileşenlerinin harici çevre ve antijenler ile olan etkileşimin nasıl olacağını kontrol edeceğini modeller.
- Bağışıklık Ağ (Immune Network) Modelleri: Kendi yapılarını, dinamiklerini ve meta dinamiklerini içeren bağışıklık ağlarını modeller.

Bu bölümün geri kalanında öncelikle klonal seçme ve bağışıklık ağ modelleri daha ayrıntılı bir şekilde ele alınacaktır. Daha sonra bu modeller temel alınarak geliştirilmiş ve tez çalışmasında kullanılan bir Klonal Seçme Algoritması (KSA) olan CLONALG ve bir Yapay Bağışıklık Ağ Algoritması olan (YBAA) opt-aiNet algoritmaları tanıtılacaktır.

3.2.3.1. Klonal seçme modeli ve CLONALG algoritması

Klonal seçme, adaptif bağışıklık sisteminin patojenik mikroorganizmalar ile nasıl ilgilendiğini anlatan bir teoridir. Bu teoriye göre prolife evresinde belli bir duyarlılıkla antijeni tanıyan B hücreleri seçilir. Bu hücreler aseksüel oldukları için

çaprazlama olmaksızın mitoz bölünme ile çoğalırlar. Bu üreme evresi süresince B hücreleri hiper mutasyona tabi tutularak antijene karşı olan duyarlılıkları artırılır. Seçilen antijene olan duyarlılığın artırılması işlemi yani seçme ve mutasyon süreçlerine *bağışıklık cevabının duyarlılık olgunlaşması* adı verilir. B hücrelerindeki bu değişme işlemine ek olarak, antijenlere karşı yüksek duyarlılıkla aktive edilmiş B hücreleri, hafıza hücresi olmak üzere seçilirler. Uzun ömürlü olan bu hafıza hücreleri yüksek duyarlılık gösterdikleri antijenlerle veya benzerleriyle tekrar karşılaştıkları zaman daha çabuk cevap verilmesini sağlarlar.

Klonal seçme prensibinin iki temel özelliği bulunmaktadır:

1) Duyarlılık Orantılı Üreme: Antijene karşı yüksek duyarlılık gösteren seçilmiş B-hücrelerinden antijene olan duyarlılıkları ile orantı çoğaltılmasıdır.

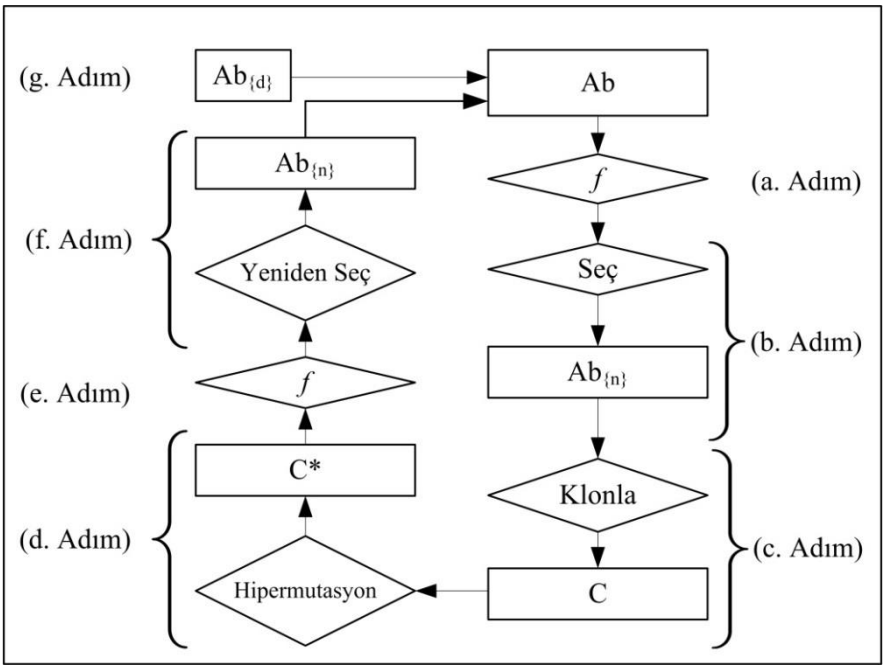
2) Mutasyon: Her bir bağışıklık hücresinin üreme esnasında maruz kaldığı mutasyon oranı, duyarlılığı ile ters orantılıdır. Yüksek duyarlılığa sahip hücre az mutasyona uğrarken duyarlılığı az olan hücre çok mutasyona uğrar. Bu olaya *hipermutasyon* adı verilir.

Bu prensiplere dayanarak örüntü tanıma ve makine öğrenmesi problemlerinde kullanılmak üzere de Castro, L. N. ve Von Zuben tarafından 2002 yılında CLONALG isimli bir klonal seçme algoritması önerilmiştir. Daha sonra bu algoritma optimizasyon problemlerine de uyarlanmıştır. Aşağıda optimizasyon problemlerinin çözümünde kullanılan CLONALG algoritmasının adımları verilmiştir.

1. Rastgele *j* tane *Ab* üretilir ve *P* başlangıç popülasyonu oluşturulur.
2. Daha önceden belirlenen iterasyon sayısı kadar aşağıdaki işlemler tekrar ettirilir:
    a. Her bir *Ab* antikorunun duyarlılığı hesaplanır. (Duyarlılık hesaplaması optimize edilecek amaç fonksiyonun değerlendirilmesine karşılık gelir.)
    b. En yüksek duyarlılığa sahip *n* tane antikor seçilir.
    c. Seçilen bu *n* tane antikor için N_c tane klon üretilir ve *C* klon popülasyonu elde edilir.
    d. *C* klon popülasyondaki antikorlar duyarlılıkları ile ters orantılı olacak şekilde hiper mutasyona tabi tutulurlar. Yani duyarlılığı yüksek olan birey az derecede mutasyona uğrarken, duyarlılığı az olan birey çok mutasyona uğrar.
    e. Mutasyona uğramış *C* klonlarının duyarlıkları hesaplanır.

f. *C* kümesi ve antikorlar arasından en yüksek duyarlılığa sahip *n* tane birey seçilir ve yeni *P* popülasyonunu oluşturulur.
g. Yeni *P* popülasyonundaki en düşük duyarlılığa sahip *d* tanesini yeni bireyler ile değiştirilerek çeşitlilik (diversity) sağlanır.

3. Daha önceden belirlenen iterasyon sayısına ulaşılmışsa algoritma sonlandırılır aksi takdirde adım 2'ye gidilir.



Şekil 3.10. CLONALG algoritmasının optimizasyon versiyonun akış diyagramı

CLONALG algoritmasının optimizasyon versiyonun akış diyagramı Şekil 3.10'da verilmiştir (de Castro, L. N. ve Von Zuben, 2002). Daha anlaşılır olması için algoritmanın her bir adımı akış diyagramın üzerinde belirtilmiştir.

3.2.3.2. Bağışıklık ağ modeli ve opt-aiNet algoritması

Bağışıklık sistemi için ilk önerilen yapıda, sistem yabancı bir mikroorganizma tarafından uyarılmadığı sürece pasif durumdadır. Ancak antijenik bir yapı tarafından uyarıldığında dinamik hale geçer. Ancak daha sonradan sunulan bağışıklık ağ teorisine göre B hücreleri, reseptörlerinde bulunan idiotop adlı yapı sayesinde diğer serbest durumdaki antikorlar veya reseptörler tarafından tanınabilirler. Bağışıklık sistemi elemanlarının tanıma ve tanınabilme yetenekleri sayesinde sistem sürekli dinamik bir yapıda kalır. İşte bağışıklık ağ modelleri, klonal seçme prensibinden farklı olarak

bağışıklık sisteminin bu dinamik yapısını modellerler. Klonal seçme prensibi ise sadece antijenik bir yapı tarafından uyarıldığında dinamik duruma geçtiği düşünülen bir sistemi modeller.

En popüler yapay bağışıklık ağ algoritmalarından birisi de Castro ve Von Zuben (2001) tarafından sunulan ve veri analizi, tanıma, sınıflandırma için kullanılabilinen aiNet algoritmasıdır. aiNet modeli antikor adı verilen ve birbirleri ile bağlantılı hücrelerden meydana gelir. Bu antikorların sisteme girebilecek antijenlerin dâhili bir görüntüsünü oluşturduğu varsayılır.

Bu tez çalışmasında ise aiNet algoritmasının optimizasyon problemlerine uyarlanmış şekli olan opt-aiNet algoritması temel alınarak yeni bir yapı geliştirilmiştir. Castro ve Timmis (2002b) tarafından sunulan orijinal opt-aiNet algoritması CLONALG algoritmasına bağışıklık ağ teorisinin eklenmesi ile elde edilmiştir. Algoritma ile ilgili terminoloji ve algoritmanın adımları aşağıda verilmiştir.

Algoritma ile ilgili terminoloji:

- Ağ Hücresi (Network Cell): Popülasyonun bireyleri.
- Uygunluk (Fitness) : Bir ağ hücresinin optimize edilecek amaç fonksiyonuna giriş olarak verildiğinde elde edilen çıkış değeri
- Duyarlılık (Affinity) : İki ağ hücresinin birbirine olan benzerliği
- Baskılama (Suppression) : İki ağ hücresinin birbirine olan benzerliği, daha önceden belirlenen bir eşik değerinden (baskılama eşiği) fazla ise, ağ hücrelerinden bir tanesinin ağdan çıkarılması işlemi

Algoritmanın Adımları:

1. Rastgele şekilde ağ hücrelerinden oluşan başlangıç popülasyonu oluştur.
2. Durdurma şartı sağlanasıya kadar tekrar et:
    a. Her bir ağ hücresinin uygunluğunu belirle ve uygunluk vektörünü normalize et.
    b. Her bir ağ hücresinden N_c tane klon üret.
    c. Her bir klonu, ebeveyn hücresinin uygunluğu ile ters orantılı olarak denklem 3.1 ‘deki formülü kullanarak mutasyona uğrat. Ebeveyn hücrenin aslını sakla.
    d. Mutasyona uğramış hücrelerin uygunluklarını hesapla.
    e. Her bir klonu kendi ebeveyni ile kıyasla. Uygunluğu daha iyi olanı seç ve bu şekilde güncellenen popülasyonun ortalama uygunluğunu hesapla.

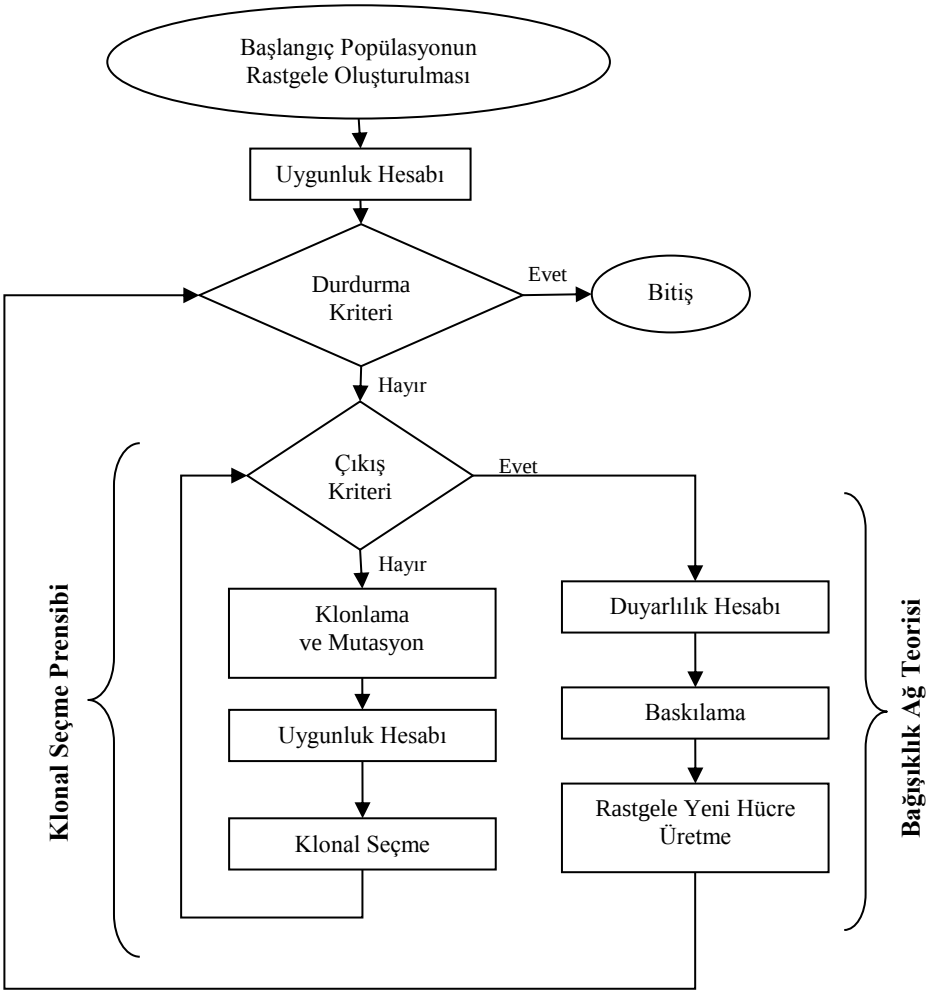
f. Eğer popülasyonun ortalama uygunluğu, bir önceki iterasyondaki popülasyonun ortalama uygunluğundan anlamlı bir şekilde farklı ise adım (a)'a git; değilse devam et.

g. Ağdaki her bir hücrenin birbirleri ile olan duyarlıklarını hesapla. Duyarlılığı baskılama eşiğinin altında olan hücrelerden birini baskıla. Baskılamadan sonra, belli sayıda ağ hücresini hafıza hücresi olarak sakla.

h. *%d* oranında yeni hücre oluştur.

3. Durdurma şartı sağlanmışsa algoritmayı sonlandır değilse adım (a)'a git.

Belli bir iterasyon sayısına ulaşıldığında veya birbirini takip eden *k* adım boyunca ortalama uygunluk değerleri arasındaki farkın daha önceden belirlenen bir değerin altında kalması halinde algoritmanın durdurma şartı sağlanmış olur (de Castro ve Timmis 2002a).



Şekil 3.11. opt-aiNet algoritmasının akış diyagramı

Algoritmanın akış diyagramı Şekil 3.11'de gösterildiği gibidir. Şekilde de görüldüğü üzere arama uzayının kullanılmasını (exploitation) iç döngüdeki klonal seçme mekanizması yönetir. Popülasyondaki hücreler arasında bu aşamada herhangi bir etkileşim yoktur. Hücreler arasındaki ağ etkileşimi dış döngüde bağışıklık ağ teorisine uygun olarak meydana gelir. Bu sayede benzer çözümler baskılanarak bir sonraki nesle aktarılması engellenmiş olunur. Bu süreç ağ hücrelerinin sayısının dinamik olarak kontrol edilmesini de sağlar. Rastgele yeni hücre ekleme adımı ile de popülasyonun çeşitliliği (diversity) garanti altına alınır.

Algoritmanın b. adımında oluşturulan klonlar, denklem 3.1'de verilen formül kullanılarak hipermutasyona uğratılırlar.

$$c' = c + \alpha * N(0,1) \quad (3.1)$$

Burada; c mutasyona uğrayacak klonu, c' klonun mutasyona uğradıktan sonraki halini α mutasyon miktarını ve N(0,1) ortalaması sıfır, varyansı bir olan Gaussian operatörünü temsil etmektedir. Mutasyon miktarı α ise denklem 3.2'ye göre hesaplanır.

$$\alpha = (1/\beta)\exp(-f^*) \quad (3.2)$$

Bu formülde ise β, α mutasyon miktarındaki değişim oranını kontrol eden parametredir. β arttıkça mutasyon oranı azalır. Algoritmanın yakınsamasını direkt etkileyen bir parametredir. f^* parametresi ise tüm ağ hücrelerinin [0-1] aralığında normalize edilmiş uygunluk değerlerini içeren vektörüdür.

4. BULANIK MANTIK

Mantık, Arapça nutk kelimesinden türemiştir ve Yunanca Logos kelimesine karşılık gelir. Mantıklı düşünme ise, gündelik yaşantıda doğru ve düzgün düşünmektir. Doğru ve düzgün düşünmek akıl yürütmek, yargıda bulunarak çıkarım yapmaktır. Diğer bir deyişle, mantık akıl yürüterek düşünce dünyasında bazı çıkarımlara ulaşan ve bu ulaşımın nasıl yapılacağını inceleyen bilim dalı olarak tanımlanabilir. Düşünceden mantığa geçerken önermeler sözel olarak yapılır, bundan dolayı mantık bir bakıma kelimelerle hesap yapma yöntemidir (Baykal ve Beyan, 2004a).

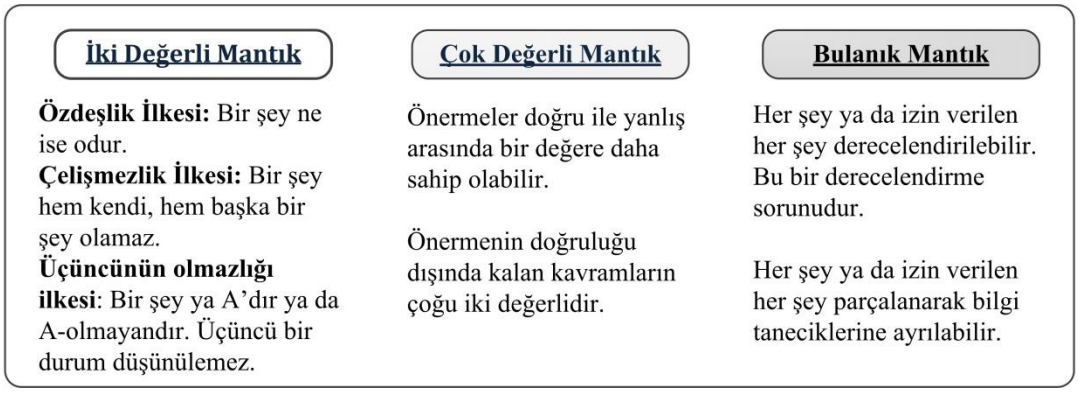
Geleneksel mantık sistemleri, sadece belirli koşullarda oluşan, kesin doğruluk değerleri doğru ya da yanlıştan birisine sahip önermelerle ilgilenirler. Belirsizlikle ilgilenmezler. Üçüncü bir durumun gerçekleşmesinin imkânsız olduğu varsayılır ve çoğu zaman bu tür durumlar paradoks olarak adlandırılır. Doğruluk önermeleri {Yanlış, Doğru} veya sayısal olarak {0,1} kümesinin elemanlarıyla ilişkilendirilen iki değerli kümelerdir (Baykal ve Beyan, 2004a).

Geleneksel olmayan mantıklar, çok değerli mantık olarak tanımlanır. Önermeler doğru ile yanlış arasında bir değere daha sahip olabilir. Ancak önermenin doğruluğu dışında kalan kavramların çoğu iki değerlidir (Zadeh,2006).

Bulanık mantık ise, ikili mantığın reddettiği belirsizliğin kendisiyle ilgilenir. Bulanıklığın anlamı, bir araştırmacının incelediği konunun kendisi tarafından tam kesinlikle bilinmemesi durumunda sahip olduğu eksik ve belirsiz bilgilerin tümüdür (Şen,2004). Bulanık mantığı, diğer mantık sistemlerinden ayıran en önemli özelliklerinden birisi, üçüncünün olmazlığı ve çelişmezlik ilkelerinin bulunmamasıdır. Bulanık mantıkta bir önerme hem doğru hem yanlış olamaz denilemez. Bir önerme [0,1] gerçel sayı aralığında sonsuz sayıda doğruluk derecesi alabilir. Şekil 4.1'de mantık sistemlerinin bu kıyaslaması gösterilmektedir.

Bu noktada bir konuya açıklık getirmek gerekmektedir. Olayları incelemede kullanılan veri ve bilgi bakımından bir bulanıklık söz konusu ise de bulanık mantığın kendisi bulanık değildir. Bulanık mantık yaklaşık akıl yürütme ve belirsizlikle ilgilenen işleyişi tamamen belirgin bir mantıktır. Bulanık mantık insana dair iki önemli özelliği formülize etmeye çalışır. Bu özelliklerden ilki belirsiz, kesin olmayan ve eksik bilgilerin bulunduğu ortamlarda sözel düşünme ve muhakeme yeteneği ile ilişkisel kararlar verebilmesidir. İkincisi ise herhangi bir ölçüm veya hesaplama yapmadan edindiği

tecrübe ve gözlemleri kullanarak fiziksel ve zihinsel işlemleri yerine getirebiliyor olmasıdır. Buna örnek olarak araba park etmeyi verebiliriz (Zadeh,2008).



Şekil 4.1. Mantık Sistemleri (Zadeh,2006)

Bulanık mantığın uygulamada pek çok yararı bulunmaktadır. Bulanık mantık kolayca anlaşılabilirdir ve arka plandaki teorileri basittir. Esnek yapısı sayesinde, kesin olmayan verileri, belirsizlik ve olasılık durumlarını, karmaşık ve doğrusal olmayan fonksiyonları uzman görüşüne dayanarak modelleyebilir. Ayrıca doğal olup, konuşma diliyle modelleme imkânı sağlamaktadır. Bulanık mantığın genel özellikleri maddeler halinde aşağıda verilmiştir:

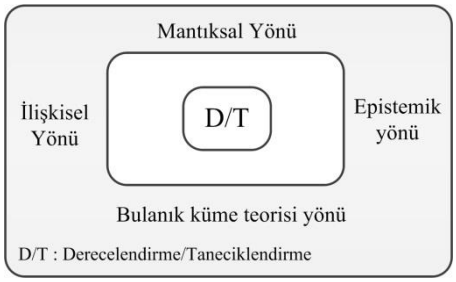
1. Bulanık mantıkta kesin nedenlere dayalı düşünme yerine yaklaşık değerlere dayanan düşünme kullanılır.
2. Bulanık mantıkta her şey [0,1] aralığında belirli bir derecede gösterilir.
3. Bulanık mantıkta bilgi büyük, küçük, çok az gibi sözel ifadeler şeklindedir.
4. Bulanık çıkarım işlemi sözel ifadeler arasında tanımlanan kurallar ile yapılır.
5. Her mantıksal sistem bulanık olarak ifade edilebilir.
6. Bulanık mantık matematiksel modeli çok zor elde edilen sistemler için uygundur (Baykal ve Beyan, 2004a).

Yine de daha basit çözüm yolları olduğunda bulanık mantık yerine bu yollar kullanılmalıdır. Ayrıca girdilerin çıktılar ürettiği bir süreç yoksa bulanık mantık kullanılamaz (Baykal ve Beyan, 2004a).

4.1. Bulanık Mantığın İlkesel Yönleri

Zadeh (1994), bulanık mantığın dar ve geniş olmak üzere iki tanımını yapmıştır. Dar anlamda, bulanık mantık çok değerli mantığın genelleştirilmiş şeklidir. Geniş

anlamda ise, mantıksal bir sistemden daha fazlasını ifade etmektedir ki günümüzde bu tanımın kullanılması daha baskındır. Buna göre bulanık mantığın dört ilkesel yönü vardır. Bunlar mantıksal yönü, bulanık küme teorisi yönü, epistemik yönü ve ilişkisel yönüdür. Bulanık mantığın ilkesel yönleri Şekil 4.2'de gösterilmiştir.

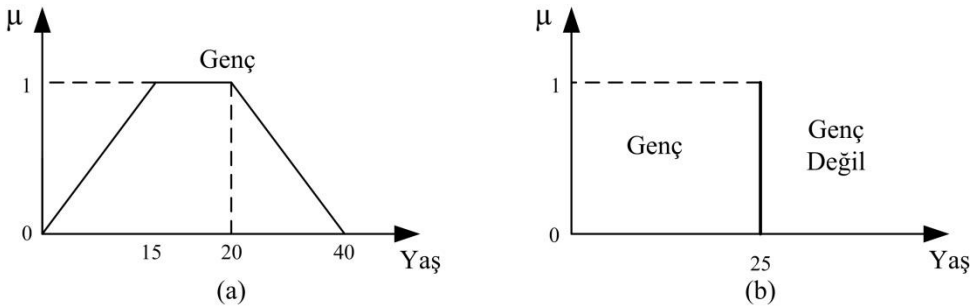


Şekil 4.2. Bulanık Mantığın İlkesel Yönleri (Zadeh, 2008)

Bulanık mantığın mantıksal yönü, dar anlamda bulanık mantığı ifade ettiğini ve çok değerli mantığın genişletilmiş hali olarak ele alınabileceğini daha önce belirtmiştik. Diğer bir deyişle mantıksal yön, doğruluğun dereceli olduğu mantıksal sistemlerle ilgilenir. Bulanık küme teorisi yönü, bulanık mantığın merkezini oluşturan bulanık kümelere odaklanır. Bulanık mantıkla ilgili matematiksel literatürün çoğu bulanık kümeler ile ilgilidir. Epistemik (bilgi bilimiyle ilgili) yönü bilginin sunumuyla, verinin analiziyle, doğal dilin yapısı ve temsiliyle ilgilidir. Doğal dil, insan algılarının ifade edilmesi için kullandığı bir sistem olarak görülür. İlişkisel yönü ise bulanık ilişkiler, taneciklendirme, sözel değişkenler ve bulanık kural kümelerine odaklanır. Bulanık mantıkla ilgili uygulamaların çoğu bulanık mantığın ilişkisel yönü ile alakalıdır (Zadeh, 2008).

Bulanık mantığın çekirdeğini oluşturan Derecelendirme ve Taneciklendirme (D/T) kavramları, bulanık mantığı diğer mantık sistemlerinden ayıran başlıca özellikleridir. Bulanık mantıkta her şey ya da izin verilen her şey derecelendirebilinir, yani bulanıklık bir olayın olup olmadığını değil, hangi dereceye kadar olduğunu ölçmektedir.

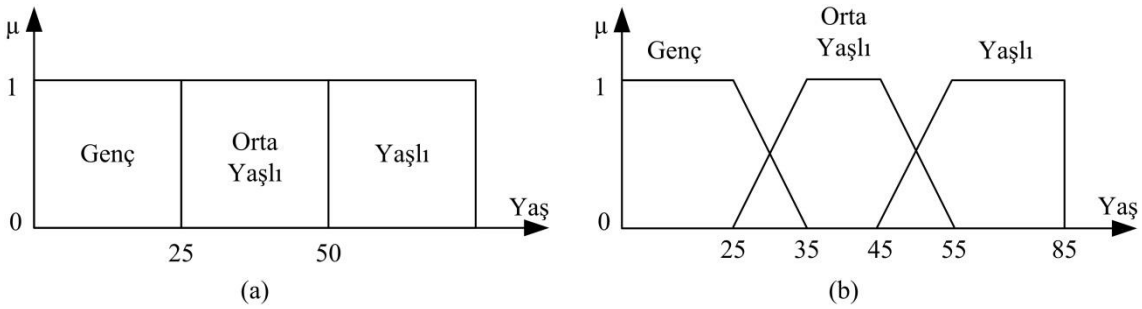
Şekil 4.3(a) 'da yaş dilsel değişkenin derecelendirilmiş olduğu durum gösterilmiştir. Şekilden, [0-1] aralığında bulunan ve yaş değişkeninin *Genç* bulanık kümesine olan aitliğini gösteren üyelik dereceleri elde edilebilir. Şekil 4.3(b)'de yaş değişkenin derecelendirilmediği durum gösterilmiştir. Bu şekle göre yaş değeri 25'den küçükse kişi genç, büyükse genç değil olarak nitelendirilir. Ara değerler yoktur.



Şekil 4.3. Yaş dilsel değişkeninin (a) derecelendirilmiş hali (b) derecelendirilmemiş hali

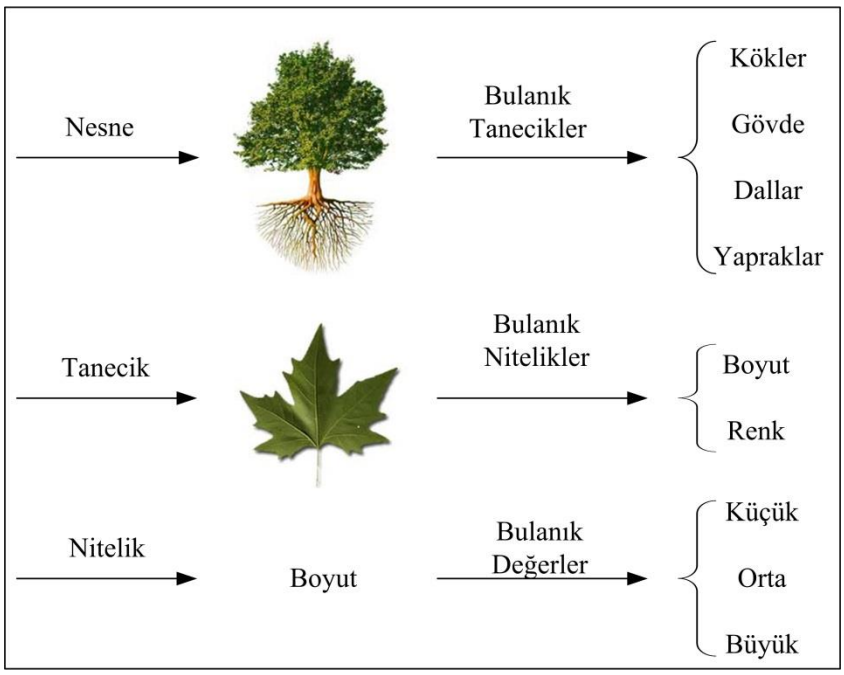
Bulanık mantığı olasılıktan ayıran kavram da derecelendirme işlemidir. Bulanıklığın bir olayın olma derecesini ölçtüğünü, olasılık gibi olayın olup olmayacağını ölçmediğini belirtmiştik. Olasılık olay sınıfının klasik tanımlandığını, tersinmezlik yasalarını ($A\cap\overline{A}=\varnothing$) ve üçüncünün olmazlığını ($A\cup\overline{A}=I$) kabul eder. Bulanıklık ise bu yasalar bozulduğunda ortaya çıkar. Bu konuyu bir örnek üzerinde açıklayalım. Kapının dışarısında A ve B isimli iki kişinin beklediğini farz edelim. Bu kişilerin yaşlarına dair elimizde iki adet bilgi olsun. İlk bilgimiz, A kişinin %80 olasılıkla genç olduğunu ikinci bilgimiz ise B kişisinin % 80 üyelik derecesiyle genç kümesine dâhil olduğudur. Burada A kişisinin yaşı 25 olabileceği gibi geriye kalan % 20 ihtimal gereği 95 de olabilir. Tam bir kestirim yapmak güçtür. Ancak B kişisinin yaşının 20 ile 30 arasında olabileceğini ve 95 olmasının mümkün olmadığını rahatlıkla söyleyebiliriz. Burada dikkat edilmesi gereken diğer bir nokta, A kişisi kapıdan girdiği zaman yaşı ile ilgili olan belirsizliğin ortadan kalkıp yerini kesinliğe bırakacağıdır. Ancak B kişisin genç bulanık kümesine dâhil olma derecesi kapıdan girdiğinde de değişmeyecektir.

Taneciklendirme (granulation), değişkenlerin benzerliklerine, farklılıklarına, yakınlık veya işlevselliklerine bakılarak parçacıklara bölünmesidir (Zadeh,1997). Kesin ve bulanık tanecikler olmak üzere ikiye ayrılır. Kesin taneciklerin sınırları belli ve nettir buna karşın bulanık taneciklerin sınırları tam olarak belli değildir. Taneciklerin her biri dilsel bir değer ile ifade edilir. Şekil 4.4'de kesin ve bulanık taneciklendirmeye bir örnek verilmiştir. Şekil 4.4(a)'da sürekli bir değişken olan yaş, kesin sınırlar kullanılarak parçalara bölünmüştür. Şekil 4.4(b)'de de tanecikler kişilerin doğdukları günden günümüze kadar yaşadıkları yıl sayılarının benzerliklerine göre oluşturulmuş, daha sonra bu tanecikler derecelendirilerek her biri bir dilsel değer ile etiketlenmiş ve bulanık tanecikler elde edilmiştir.



Şekil 4.4. Yaş değişkenin (a) kesin taneciklendirilmesi, (b) bulanık taneciklendirilmesi

Kesin taneciklendirme analog bilginin dijitale çevrimi, miktar ölçme (quantization) ve resim segmentasyonu gibi pek çok alanda kullanılmaktadır. Ancak kesin taneciklendirme, kısmi ve kesin olmayan bilgiler kullanarak ilişkisel kararlar verebilen insan düşünce sistemine uygun değildir. Örneğin bir ağacı kesin sınırlarına ayırmak güçtür. Bu aşamada da bulanık taneciklendirme devreye girer. Öncelikle ağaç değişkeni kök, gövde, dal ve yaprak dilsel değişkenleri kullanılarak bulanık taneciklerine ayrılır. Daha sonra her bir bulanık tanecik uygun bulanık nitelikler ile ilişkilendirilir. Örneğin, yaprak bulanık taneciği için kullanılan bulanık nitelikler boyutu ve rengidir. Son aşamada ise bulanık niteliklere bulanık değerler atanır. Yaprak bulanık taneciğinin boyut bulanık niteliğine küçük, orta ve büyük bulanık değerleri atanabilir. Bu işlemin adımları Şekil 4.5'de temsil edildiği gibidir (Ramirez, 2002).

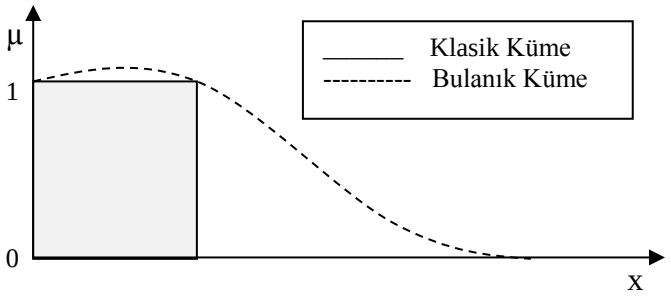


Şekil 4.5. Bulanık taneciklendirme adımları: Tanecikler, nitelikler ve değerler

Derecelendirme ve taneciklendirme birlikte dilsel ifade kavramını oluştururlar. Derecelendirilmiş taneciklendirme veya bulanık taneciklendirme, insanın karmaşıklık ve belirsizlikle ilgilenme biçiminden esinlenir ve hemen hemen tüm bulanık uygulamalarda kilit bir rol oynar. Bundan dolayı bulanık mantık insan düşünce sisteminin bir modeli olarak görülür (Zadeh, 2008).

4.2. Bulanık Kümeler ve Üyelik Fonksiyonları

Geleneksel küme teorisinde kullanılan küme kavramı bir nesnenin bir kümenin elemanı olması veya olmaması şeklinde iki seçenekli bir mantığa dayanmaktadır. Zadeh (1965) niteliklerin ikili üyelik fonksiyonları ile ifade edildiği klasik kümeler yerine, dereceli üyelik fonksiyonları ile ifade edildiği *bulanık kümeler* tanımlamasını önermiştir. Zadeh'e (1965) göre *Bulanık küme* değişik üyelik derecesinde öğeleri olan bir topluluktur. Yani bulanık küme kuramında, kümedeki her bir birey bir dereceye kadar kümenin üyesi olarak görülmektedir. Üyelik dereceleri arasındaki geçiş yumuşak ve süreklidir. Klasik kümelerdeki karakteristik fonksiyon $\mu_A : E \rightarrow \{0,1\}$, bulanık kümelerde yerini $\mu_A : E \rightarrow [0,1]$ olarak gösterilen üyelik fonksiyonuna bırakır (Baykal ve Beyan, 2004a). Şekil 4.6'da klasik küme ile bulanık kümenin kıyaslanması gösterilmiştir.



Şekil 4.6. Klasik küme ile bulanık kümenin kıyaslanması

Denklem 4.1'de X evrensel uzayında bulunan A bulanık kümesinin gösterimi verilmiştir.

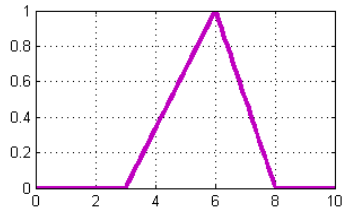
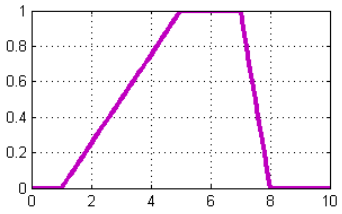
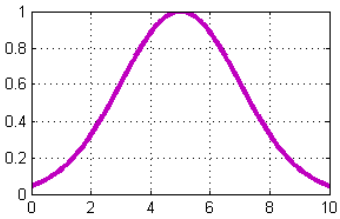
$$A = \sum_{x \in X} \mu_A(x)/x \qquad (4.1)$$

Buradaki 'Σ' sembolü matematikteki toplama operasyonunu temsil etmemektedir. A bulanık kümesine ait olan üyelik değerlerinin koleksiyonunu simgeleyen bir ekleme operatörüdür (Çelikyılmaz ve Türkşen, 2009).

4.2.1. Üyelik fonksiyonları

Bir bulanık kümedeki bulanıklığı, bulanık küme teorisinin matematiksel formülasyonunda kullanılması için grafik formda karakterize eden fonksiyonlara *Üyelik Fonksiyonu* ismi verilir (Ross,1995). Üyelik fonksiyonunun grafiğinde x ekseni üyeleri gösterirken, y ekseni üyelik derecelerini gösterir. Üçgen, yamuk, gauss, sigmoid, çan eğrisi, S ve Π gibi çok sayıda üyelik fonksiyonu tipi olmakla birlikte en fazla kullanılanlar üçgen, yamuk ve Gauss fonksiyonlarıdır. Çizelge 4.1'de sırasıyla bu fonksiyonların şekilleri ve formülleri verilmiştir.

Çizelge 4.1. Üçgen, yamuk ve gauss üyelik fonksiyonları ve formülleri

Üçgen Üyelik Fonksiyonu	Yamuk Üyelik Fonksiyonu	Gauss Üyelik Fonksiyonu
		
$f(x;a,b,c)$ $= \text{maks}\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$	$f(x;a,b,c,d)$ $= \text{maks}\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$	$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$

4.2.2. Üyelik fonksiyonunun kısımları

Üyelik fonksiyonlarının kısımlarına geçmeden önce, *α-kesim* (α-cut) kümesi ve *kuvvetli α-kesim* (strong α-cut) kümesi kavramlarının açıklanması faydalı olacaktır.

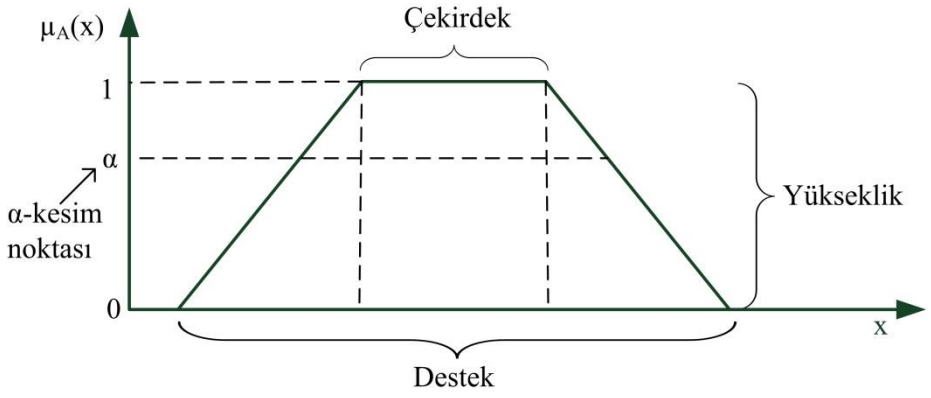
X evrensel kümesi üzerinde A bulanık kümesi verilsin ve $\alpha \in [0,1]$ aralığında herhangi bir sayı olsun. Buna göre A kümesinin α-kesim kümesi α_A ve kuvvetli α-kesim kümesi α_A^+ aşağıdaki şekilde tanımlanır.

$$\alpha_A = \{x | \mu(x) \geq \alpha\} \quad (4.2)$$

$$\alpha_A^+ = \{x | \mu(x) > \alpha\} \quad (4.3)$$

Denklem 4.2 ve 4.3'den de anlaşıldığı üzere A bulanık kümesinin α-kesim ve kuvvetli α-kesim kümeleri kesin kümelerdir. A'daki üyelik derecesi α'dan büyük ya da eşit olan X evrensel kümesinin tüm elemanları α-kesim kümesinin, sadece büyük olanlar da kuvvetli α-kesim kümesinin elemanlarıdır (Çelikyılmaz ve Türkşen, 2009).

Bölümün devamında Şekil 4.7'de gösterilen üyelik fonksiyonlarının temel kısımları olan *Çekirdek* (Core) ve *Destek* (Support) kavramları yukarıda açıklanan α-kesim kümelerinden de faydalanarak açıklanacaktır.



Şekil 4.7. Üyelik fonksiyonunun kısımları (Çelikyılmaz ve Türkşen, 2009)

Şekil 4.7'de X evrensel kümesine ait bireylerin A bulanık kümesine dahil olma üyelik derecelerini gösteren yamuk şeklinde bir üyelik fonksiyonunun kısımları verilmiştir. Şekilden de görüldüğü üzere 1 üyelik derecesine sahip olan bireyler hiçbir şüpheye yer vermeyecek şekilde A bulanık kümesine aittirler. İşte bu bireyler A kümesinin özünü yani *çekirdeğini (core)* oluştururlar. Diğer bir deyişle α=1 olduğu durumda A kümesinin α-kesim kümesi α_A, A kümesinin çekirdeğini temsil eder. Buna göre A bulanık kümesinin çekirdeğini veren formül denklem 4.4'deki gibi yazılır.

$$çekirdek(A) = \{x \in X | \mu_A(x) = 1\} \quad (4.4)$$

Bir üyelik fonksiyonunun tüm öğelerini içeren aralığa ise *destek (support)* denilir. Şekil 4.7'de *X* evrensel uzayı içerisinde yer alan bulanık A kümesinin *desteğinin (support)*, A kümesindeki üyelik dereceleri sıfırdan farklı olan X evrensel kümesinin tüm elemanlarını içeren bir küme olduğu görülmektedir. Burada da A kümesinin *desteği*, α=0 olduğu durumunda A kümesinin kuvvetli α-kesim kümesi ile aynı şeydir ve denklem 4.5'de gösterildiği şekilde formülize edilir.

$$destek(A) = \{x \in X | \mu_A(x) > 0\} \quad (4.5)$$

Üyelik fonksiyonunun yüksekliği ise, üyelik derecelerini gösterdiği bulanık kümedeki en büyük üyelik derecesine eşittir ve Şekil 4.7'deki A bulanık kümesi için denklem 4.6'daki gibi gösterilir.

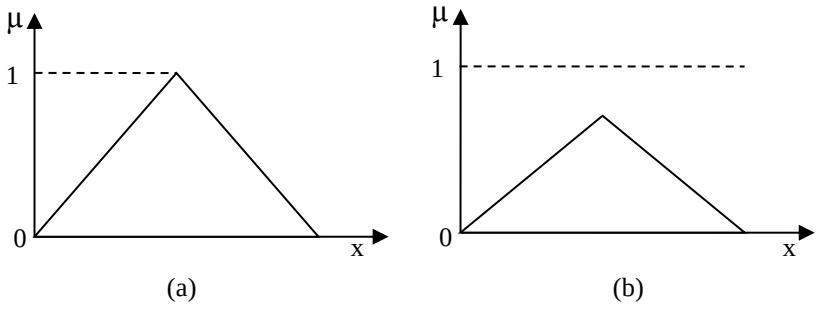
$$yükseklik(A) = sup_{x \epsilon X}(\mu_A(x)) \quad (4.6)$$

Bulanık kümelerin normalliğinin tespiti için yükseklik kavramından faydalanılır.

4.2.3. Bulanık kümelerin sınıflandırılması

Bulanık kümeler normaliklerine, dış bükeyliklerine (konvkes) ve simetrikliklerine göre sınıflandırabilinir. Bu kavramlar aşağıda açıklanmıştır.

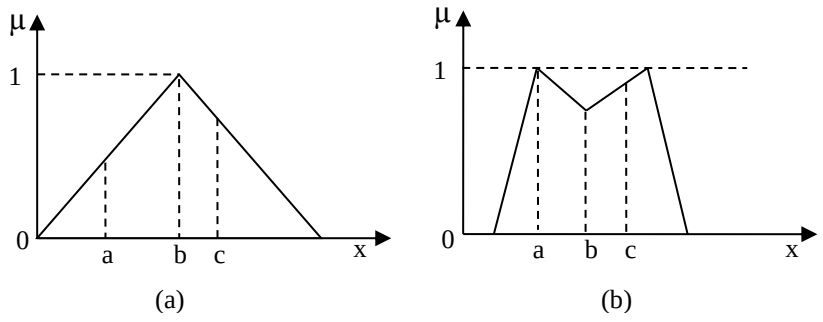
Normalliklerine göre Bulanık Kümeler: Eğer X evrensel uzayında bulunan en az bir eleman, üyelik fonksiyonunda 1 değerini alıyorsa, bu küme normal bulanık küme olarak isimlendirilir. Diğer bir deyişle, yüksekliği 1 olan bulanık kümeler normaldir. Bu şartı sağlamayan kümelere de *normal olmayan bulanık küme* denir. Şekil 4.8.'de normal ve normal olmayan bulanık kümelere birer örnek verilmiştir.



Şekil 4.8. (a) Normal (b) Normal olmayan bulanık kümeler

Bükeyliklerine göre Bulanık Kümeler: Eğer bir bulanık kümenin üyelik fonksiyonunun değeri kümenin dayanağı üzerinde, monoton olarak artıyorsa ya da monoton olarak azalıyorsa ve artan değerler için fonksiyonun değeri 1 olasıya kadar önce sürekli artıyor, sonra da sürekli azalıyorsa (üçgen fonksiyonlarda olduğu gibi) bu bulanık küme dış bükeydir diye nitelendirilir.

Bu şartı sağlamayan kümelere de *dış bükey olmayan bulanık küme* denir. Şekil 4.9'da dış bükey ve dış bükey olmayan bulanık kümelere birer örnek verilmiştir.



Şekil 4.9. (a) Dış bükey (b) Dış bükey olmayan bulanık kümeler

Dış bükeyliğin matematiksel tanımı şu şekildedir: Şekil 4.9'da a<b<c olacak şekilde verilen aynı kümeye ait a,b ve c noktaları için denklem (4.7) verilen bağıntı geçerliyse, bulanık küme dış bükeydir denilir.

$$\mu(b) \geq \min[\mu(a), \mu(c)] \tag{4.7}$$

Simetrikliklerine göre Bulanık Kümeler: Eğer bulanık küme üyelik fonksiyonunun belirli bir noktasına göre simetrik ise, bulanık küme simetriktir aksi takdirde simetrik değildir diye sınıflandırılır. Örneğin *x=k* noktasına göre simetrik olan bir A bulanık kümesi $\forall x \in E$ için, denklem (4.8)'de gösterildiği gibi ifade edilir.

$$\mu_A(x + k) = \mu_A(x - k) \tag{4.8}$$

Bulanık mantık sistem tasarımında kullanılacak olan bulanık kümeler, yani bulanıklaştırma adımında kullanılacak üyelik fonksiyonları simetrik olabilirde olmayabilir de ama her zaman normal ve dış bükey olmalıdır. Dış bükey olması şartı ile normal olmayan bir bulanık küme normal bulanık kümeye dönüştürülebilir. Bunun için kümedeki tüm üyelik derecelerinin, en büyük üyelik derecesine bölünmesi gerekir. Ancak bir sonraki bölümde bahsedilecek olan küme işlemlerinin kullanılması sonucunda oluşan kümeler normal olmayabilir (Şen,2004).

4.2.4. Bulanık küme işlemleri

Klasik küme teorisinde kümeler üzerinde tanımlı olan kesişim, birleşim, kapsama ve tümleyen alma temel işlemlerin hepsi bulanık kümeler içinde standart bulanık küme operasyonları olarak Zadeh (1965) tarafından tanımlanmıştır. Bölümün devamında bu işlemler Zadeh'in tanımlamasına göre sırasıyla incelenecektir. Zadeh (1965) x elemanın A bulanık kümesine ait üyelik derecelerinin gösteriminde $f_A(x)$ notasyonunu kullanmıştır. Ancak aşağıda verilen tanımlamalarda, tezin geri kalanı ile uyumlu olması için $\mu_A(x)$ notasyonun kullanılması tercih edilmiştir.

Tanım 4.1. Bulanık kümelerin eşitliği: X evrensel kümesindeki A ve B bulanık kümelerinin birbirine eşit olabilmesi için denklem 4.9'daki eşitliğin sağlanması gereklidir.

$$A=B \Leftrightarrow \mu_A(x) = \mu_B(x), \quad \forall x \in X \tag{4.9}$$

Tanım 4.2. Bulanık kümenin tümleyeni: X evrensel kümesindeki A bulanık kümesinin tümleyeni $\bar{A}$ olarak gösterilir denklem 4.10'daki gibi hesaplanır.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{4.10}$$

Tanım 4.3. Bulanık Kümeler üzerindeki Kapsama işlemi: X evrensel kümesindeki A ve B bulanık kümeleri için, A bulanık kümesi, B kümesi tarafından kapsanır veya A bulanık kümesi B bulanık kümesinin alt kümesidir demek için denklem 4.11'deki şartın sağlanması gerekmektedir.

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x), \forall x \in X \tag{4.11}$$

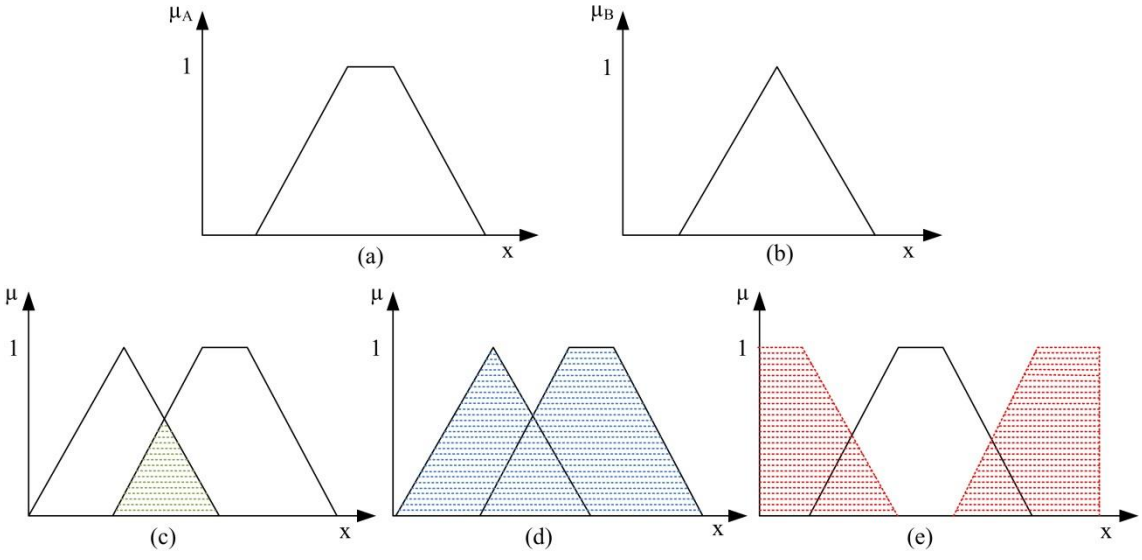
Tanım 4.4. Bulanık Kümeler üzerindeki Birleşim işlemi: X evrensel kümesindeki A, B ve C bulanık kümeleri tanımlanmış olsun. C=A∪B olmak üzere C bulanık kümesinin elemanları denklem 4.12'deki gibi tanımlanır.

$$\mu_c(x) = \text{maks}(\mu_A(x), \mu_B(x)), \forall x \in X \tag{4.12}$$

Tanım 4.5. Bulanık Kümeler üzerindeki Kesişim işlemi: X evrensel kümesindeki A, B ve C bulanık kümeleri tanımlanmış olsun. C=A∩B olmak üzere C bulanık kümesinin elemanları denklem 4.13'deki gibi tanımlanır.

$$\mu_c(x) = \min(\mu_A(x), \mu_B(x)), \forall x \in X \tag{4.13}$$

Bulanık küme işlemlerinde klasik mantık ve klasik kümelerde temel olan çelişmezlik ($\bar{A} \cap A = \emptyset$) ve üçüncünün olmazlığı ($\bar{A} \cup A = E$) ilkelerinin geçerli olmamasının sebebi A'nın tümleyenin sınırlarının bulanık olmasıdır. Tümleme, birleşim ve kesişim işlemleri için tümleme, maksimum ve minimum işlemleri basit ve esas olduğundan, standart bulanık küme işlemleri bunlar üzerinden geliştirilmiştir (Baykal ve Beyan, 2004a). Bulanık kümeler üzerindeki kesişim, birleşim ve tümleyen alma işlemlerine Şekil 4.10'da birer örnek verilmiştir.



Şekil 4.10. Standart bulanık küme işlemlerinin gösterimleri:
(a) A bulanık kümesi (b) B bulanık kümesi (c) A∩B (d) A∪B (e) $\bar{A}$

Standart bulanık küme işlemlerinin yanı sıra, bulanık kümelerde çok sık kullanılan iki çeşit işlem daha bulunmaktadır. Bunlar *üçgen norm (t-norm)* ve *üçgen conorm (t-conorm)* işlemleridir. Zadeh'e göre t-norm klasik mantıktaki *ve* işleminin, t-conorm da klasik mantıktaki *veya* işleminin karşılığıdır. t-norm ve t-conorm işlemleri bulanık kümelerde modelleme amacı ile kullanılırlar. Konuşma diliyle ifade edilen ve daha çok algıya dayanan bilgileri bulanık kümeler şeklinde modellemek ve modellenen bilgileri mantıksal çıkarımlarda kullanabilmek için sözel değişkenlere ve bulanık işlem kurallarına ihtiyaç vardır. Bu amaçla t-norm ve t-conorm ile işlem yapıldığında, çoğu zaman işleme sokulanlar üyelik fonksiyonlarıdır (Baykal ve Beyan, 2004a). Tanım 4.6. ve 4.7.'de sırasıyla t-norm ve t-conorm işlemlerinin tanımları ve özellikleri verilecektir.

Tanım 4.6. Üçgen norm (t-norm) İşlemi: $T:[0,1] \times [0,1\} \rightarrow \forall x,y,z,x',y' \in [0,1]$ olmak üzere aşağıda verilen 4 şartı sağlayan T işlemine *t-norm* denir.

1. $T(x,0) = 0,\ T(x,1) = x$ (Sınır koşulu)
2. $T(x,y) = T(y,x)$ (Değişme özelliği)
3. $(x \leq x',\ y \leq y') \rightarrow T(x,y) \leq T(x',y')$ (Monotonluk özelliği)
4. $T(T(x,y),z) = T(x,T(y,z))$ (Birleşme özelliği)

Tanım 4.7. Üçgen conorm (t-conorm) İşlemi: $\perp:[0,1] \times [0,1\} \rightarrow \forall x, y, z, x', y' \in [0,1]$ olmak üzere aşağıda verilen 4 şartı sağlayan $\perp$ işlemine *t-conorm* denir.

1. $\perp(x,0) = 0,\ \perp(x,1) = x$ (Sınır koşulu)
2. $\perp(x,y) = \perp(y,x)$ (Değişme özelliği)
3. $(x \leq x',\ y \leq y') \rightarrow \perp(x,y) \leq \perp(x',y')$ (Monotonluk özelliği)
4. $\perp(\perp(x,y),z) = \perp(x, \perp(y,z))$ (Birleşme özelliği)

Tanımlardaki ilk koşullar klasik kümelere genelleştirmeyi sağlar. İkinci koşullar t-norm ve t-conorm işlemlerinin sonuçlarının birleştirilecek kümelerin sıralamasıyla değişmediğini, üçüncü koşullar A veya B'nin üyelik fonksiyonu değerindeki azalmanın, işlem sonucu oluşturulacak kümede üyelik fonksiyonu değerinde artışa neden olmayacağını, dördüncü koşullar kümelerle değişik grup ve sıralamalarla işlem yapılabileceğini anlatır (Baykal ve Beyan, 2004a).

Literatürde tanımlanmış pek çok t-norm ve t-conorm işlemi bulunmaktadır. Bunlardan en sık kullanılanları Çizelge 4.2'de verilmiştir.

Çizelge 4.2. Sık kullanılan bazı t-norm ve t-conorm işlemleri (Çelikyılmaz ve Türkşen, 2009)

İsmi	***t-norm işlemleri***	***t-conorm işlemleri***
Zadeh	min(x,y)	maks(x,y)
Olasılıkçı	xy	x+y-xy
Lukasiewicz	maks(x+y-1,0)	min(x+y,1)
Schwizer ve Sklar	$\text{maks}(0,(x^p+y^p-1)^{1/p}),p\neq0$	$1-\text{maks}((1-x)^p+(1-y)^p-1),0)^{1/p}$
Weber ($\lambda>-1$)	$\text{maks}((x+y-1+\lambda xy)/(1+\lambda),0))$	$\min(x+y+(\lambda/1-\lambda)xy,1)$

4.3. Bulanık Sistemler

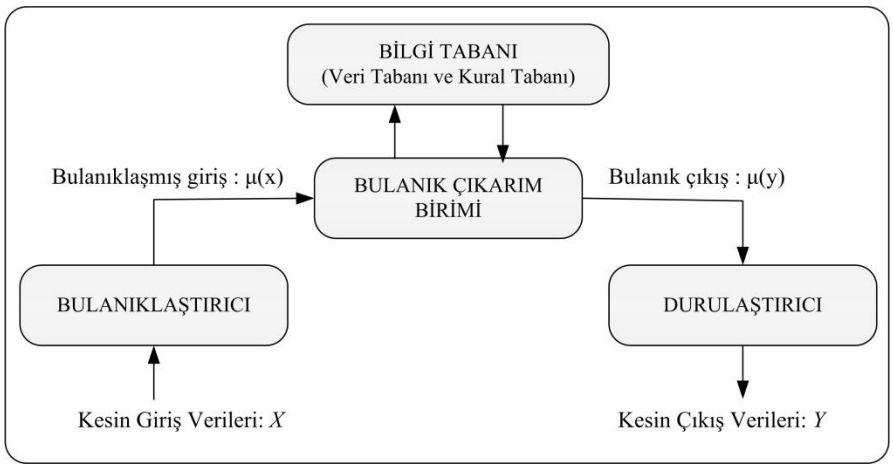
Fiziksel veya kavramsal olarak birbiriyle ilişkili birimlerin oluşturduğu bütün *sistem*, sistemin gelecekteki davranışını tahmin etmek için kullanılan araçlar *model* olarak isimlendirilir. Bileşenleri ve aralarındaki ilişkileri modellerken bulanık küme teorisi temelli bir matematiksel disiplin olan bulanık mantık ilkelerinden faydalanan sistemlere de *Bulanık Sistemler* adı verilir.

Bulanık bir sistem tasarlamak; dijital bir platformda ve esnek yöntemlerle, bulanık mantık çıkarım ve karar verme süreci sağlayacak bir sistem geliştirmeye karşılık gelmektedir. Bulanık sistemler "Eğer – O halde " şeklinde kurallarla tanımlanıyorsa *Kural Tabanlı Bulanık Sistemler* olarak adlandırılır (Baykal ve Beyan, 2004b). Bu tez çalışmasında, bulanık sistem denildiğinde *Kural Tabanlı Bulanık Sistemler* kastedilmektedir.

4.3.1. Bulanık bir sistemin genel yapısı ve birimleri

Bulanık bir sistem bulanıklaştırıcı, bilgi tabanı, bulanık çıkarım birimi ve durulaştırıcı olmak üzere temelde 4 birimin birleşiminden meydana gelmektedir. Bu birimlerin birbirleri ile olan ilişkileri Şekil 4.11'de gösterildiği gibidir. Bu birimleri sırası ile açıklayalım:

1) Bulanıklaştırıcı Birimi: Bulanık bir sistemin ilk adımıdır. Sisteme alınan kesin giriş verileri, Çizelge 4.1'de verilen üyelik fonksiyonlarından birisini veya başka bir üyelik fonksiyonu kullanılarak dilsel değerlere dönüştürülürler yani bulanıklaştırılırlar. Bu işlem sonucunda her bir giriş verisine karşılık gelen bir üyelik derecesi elde edilir.



Şekil 4.11. Bulanık bir sistemin genel yapısı

2) Bilgi Tabanı: Üyelik fonksiyonları ve bulanık kurallar ile ilgili parametrelerin depolandığı alandır ve bulanık çıkarım birimi ile sürekli iletişim halindedir. Bilgi tabanının iki temel bileşeni vardır. Bunlar veri tabanı ve kural tabanıdır. Veri tabanında, üyelik fonksiyonlarının sayısı ve değerleri ile ilgili veriler tutulur. Kural tabanında ise, bulanık sistemin çalışmasını belirleyen "Eğer - O halde" şeklinde tanımlanmış bulanık kuralların öncülleri, sonuçları ve ağırlıkları (kesinlik faktörleri) tutulur. Bulanık bir kuralın genel yapısı denklem 4.14'de gösterildiği gibidir.

$\boldsymbol{K_q}$**:** Eğer x_{p1} A_{q1} ve … ve x_{pj} A_{qj} ve... ve x_{pn} A_{qn}, O Halde y B_q, Ka_q ile (4.14)

Burada;

K_q: Bulanık kuralının etiketini

A_{qj} : Örüntü vektörünün j. niteliği için q. bulanık kuralın öncül (antecedenet) kısmını temsil eden bulanık kümeyi (j=1, ... , n)

B_q : K_q kuralının sonuç (consequent) kısmını temsil eden bulanık kümeyi

Ka_q : K_q kuralının ağırlığını ([0-1] aralığında gerçek bir sayıdır.)

göstermektedir.

Kural tabanında tutulan her bir kural aslında koşullu bir önermedir ve sistemin girişleri ve çıkışları arasındaki bağıntıları ifade eder.

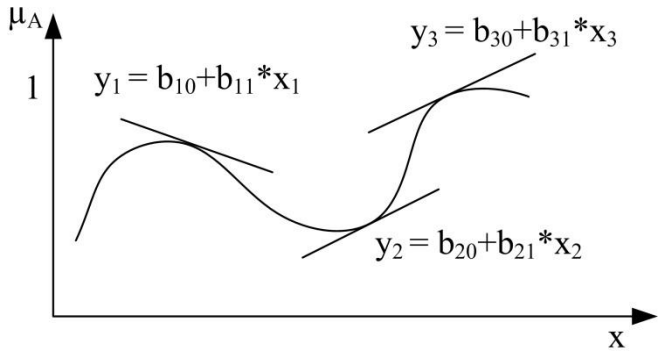
3) Bulanık Çıkarım Birimi: Bulanık mantık sisteminin çekirdek kısmını oluşturan bu birim, insanın karar verme ve çıkarım yapma yeteneğini taklit ederek bulanık çıkarımların yapıldığı birimdir. Yapılan işlem aslında üyelik fonksiyonlarından yararlanarak bulanık kuralların değerlendirilmesi (implication) ve ardından elde edilen

sonuçların bileşkesinin (aggregation) alınmasıdır. Bulanık çıkarım yöntemi olarak sıklıkla kullanılan 2 yöntem bulunmaktadır. Bunlar Mamdani (Mamdani ve Assilian, 1975) ve Takagi-Sugeno-Kang (TSK) (Takagi ve Sugeno, 1985) yöntemleridir. Bulanık sistemleri, kullandıkları çıkarım yöntemine göre isimlendirmek mümkündür. Örneğin Mamdani tipi bulanık sistem gibi.

Mamdani ve TSK tipi bulanık sistemler arasındaki temel fark sistemlerin çıkışlarıdır. Mamdani yönteminde sistemin çıkışı bulanık kümelerden oluşurken, TSK da fonksiyonlardan oluşur. Şöyle ki, Mamdani bulanık çıkarımını kullanan bir bulanık sistemin kurallarının genel yapısı denklem 4.14'deki gibiyken, TSK bulanık çıkarımını kullanan bir bulanık sistemin kurallarının genel yapısı denklem 4.15'de verildiği şekildedir.

$$\boldsymbol{K_q}\textbf{:}\ \text{Eğer } x_{p1}\ A_{q1} \text{ ve... ve } x_{pn}\ A_{qn}\text{, O Halde } y = b_{q0}+b_{q1}*x_{p1}+\ldots\ b_{qn}*x_{pn} \qquad (4.15)$$

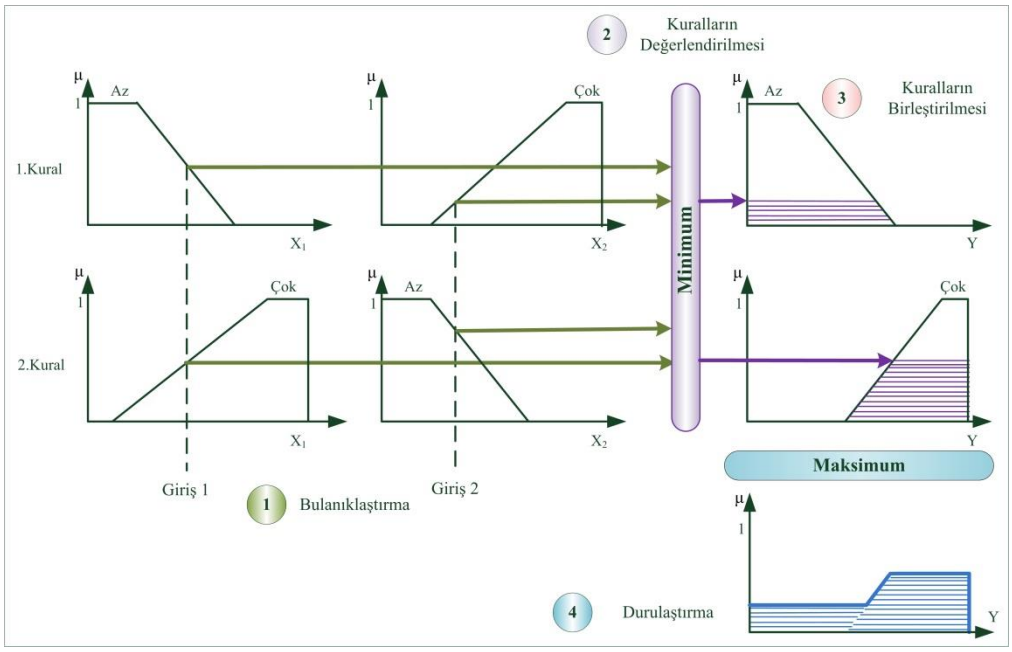
Burada b_{qi}, i=1,…,n olmak üzere gerçek bir sayıdır. Diğer parametrelerin açıklaması denklem 4.14'de verildiği gibidir. Kuralın çıkış kısmında verilen lineer denklem, geometrik olarak çıkış üyelik fonksiyonuna çizilen teğetin denklemi olarak yorumlanabilir. Şekil 4.12'de bu durumun bir örneği 3 kurallı bir bulanık sistem için gösterilmiştir. 3 tane kural olduğu için 3 tane de kural çıkışı bulunmaktadır.



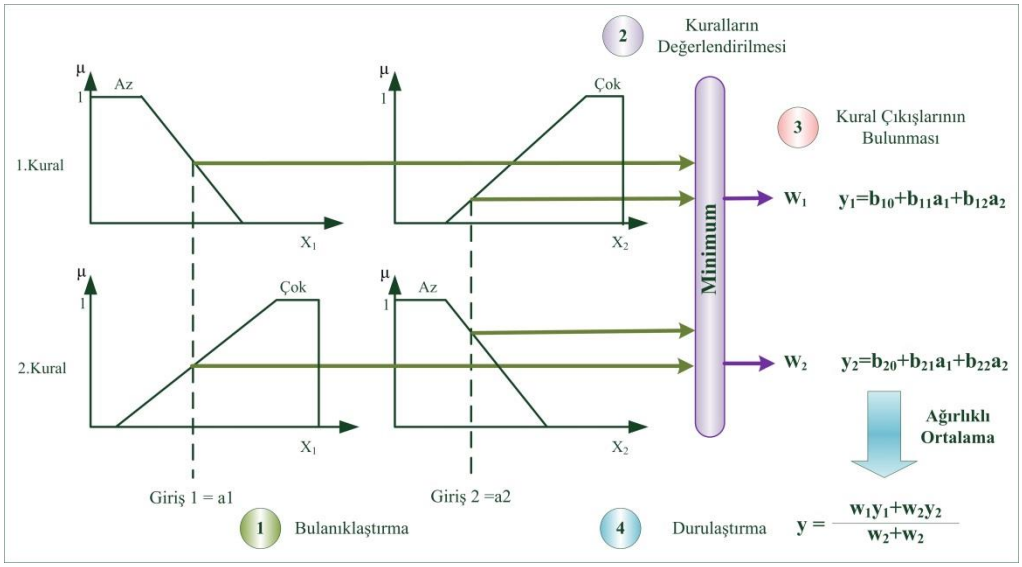
Şekil 4.12. TSK tipi bir bulanık sistemin kural çıkışlarının grafik gösterimi

Mamdani ve TSK yöntemleri kullanılarak nasıl çıkarım yapıldığı sırası ile Şekil 4.13 ve Şekil 4.14'de gösterilmiştir. Şekillerden de görüldüğü üzere, çıkarım sistemlerinin ilk iki adımı olan bulanıklaştırma ve kuralların değerlendirilmesi adımları aynıdır. Kural değerlendirilmesinde en sık kullanılan operatörlerden birisi minimum operatörü olduğu için şekilde bu operatör kullanılmıştır. Bunun yerine başka bir içerme

operatörü de (örneğin çarpma gibi) kullanılabilir. Üçüncü adıma gelindiğinde, iki sistemin kural çıkışlarını ifade etme biçimleri farklı olduğundan kuralları birleştirirken kullandıkları yöntemlerde farklılaşır. Mamdani kümelerin birleşimini alıp (maksimum operatörü kullanarak), buradan durulaştırma adımına geçer. TSK ise fonksiyon şeklinde ifade edilen kuralların çıkışlarını bulur. Daha sonra bu çıkışları ve tetiklenen kuralların uygunluk değerlerini kullanarak durulaştırma işlemini gerçekleştirir.



Şekil 4.13. Mamdani bulanık çıkarım sistemi



Şekil 4.14. TSK bulanık çıkarım sistemi

Mamdani yöntemi ve TSK yönteminin birbirlerine göre avantaj ve dezavantajları bulunmaktadır. Mamdani yöntemi, insan algılamasına daha çok hitap etmekte ve tasarımı ve eğitimi daha kolay yapılabilmektedir. Mamdani ile tasarlanan sistemlerin yorumlanabilirliği daha fazla olduğu için tercih edilmektedir. Buna karşın, çıkışları matematiksel ifadelerden oluşan TSK bulanık sistemleri insan anlayışına çok hitap etmemektedir. Çıkışlarını yorumlamak zordur ve uygun fonksiyon parametrelerin bulunması zaman alıcı olabilir. Ancak matematiksel analiz gerektiren problemlerde, doğrusal kontrol tasarımlarında kullanımı verimlidir. Sonuç olarak çözülecek problemin yapısına göre kullanılacak çıkarım modelinin seçilmesi faydalı olacaktır.

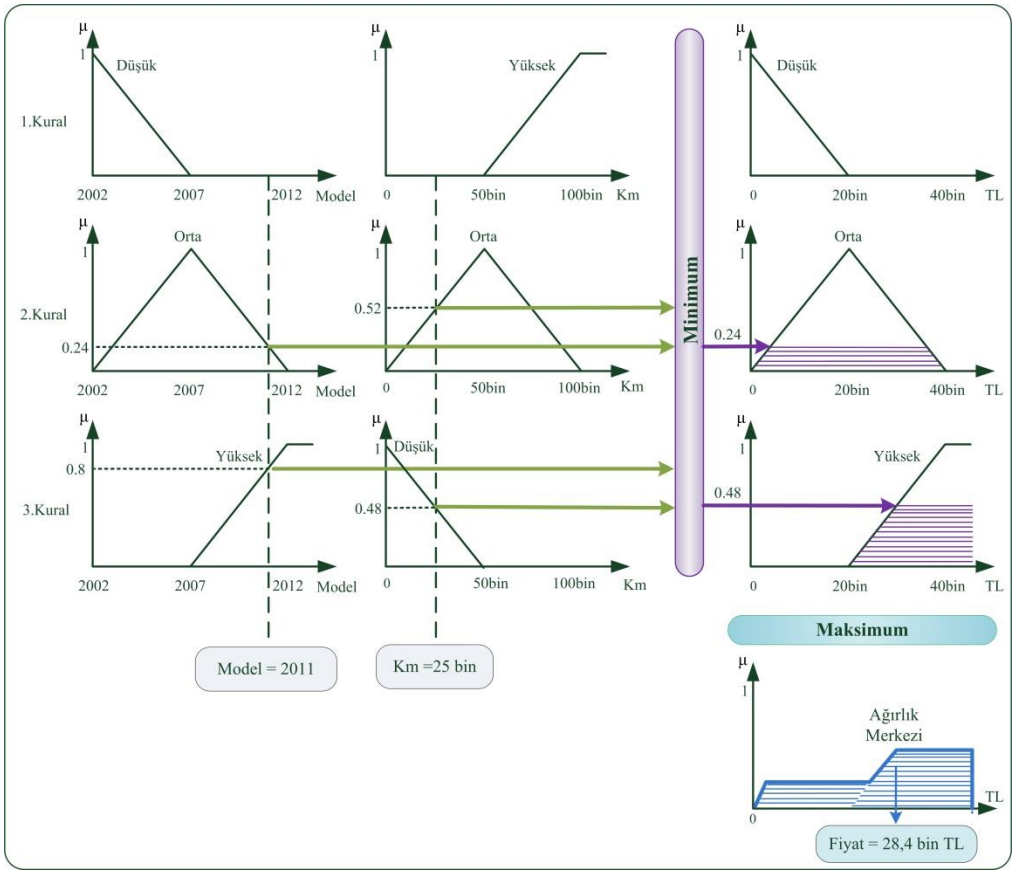
Bu tez çalışmasında tasarlanan bulanık sistemlerin yorumlanabilir olması istenmektedir ve bu yüzden çıkarım mekanizması olarak Mamdani bulanık çıkarım mekanizmasının kullanılması tercih edilmiştir. Bölümün devamında Mamdani bulanık çıkarım mekanizmasının nasıl çalıştığı bir örnek üzerinde ayrıntılı bir şekilde anlatılmıştır.

Farz edelim ikinci el bir aracın fiyatını tahmin etmek için kullanılan ve kuralları aşağıda verilen basit bir bulanık mantık sistemimiz olsun.

Kural 1: Eğer model düşük ise ve kilometresi yüksek ise O halde araç fiyatı düşüktür.
Kural 2: Eğer model orta ise ve kilometresi orta ise O halde araç fiyatı ortadır.
Kural 3: Eğer model yüksek ise ve kilometresi düşük ise O halde araç fiyatı yüksektir.

2011 model ve 25 bin kilometrede olan bir aracın fiyatının Mamdani çıkarım yöntemi kullanılarak nasıl tahmin edildiği Şekil 4.15'de gösterilmiştir. Şekilden görüldüğü üzere verilen değerler için 2. ve 3. kurallar tetiklenmiş, 1. kural tetiklenmemiştir. Öncelikle girişler bulanıklaştırılmıştır. Model olarak verilen 2011 sayısı 2. kuralın *orta* isimli bulanık kümesine $\mu_{orta}^{Model,2}$= 0.24 üyelik derecesi ile dahilken, 3. kuralın *yüksek* isimli bulanık kümesine $\mu_{yüksek}^{Model,3}$= 0.8 üyelik derecesi ile dahildir. Kilometre değeri olarak verilen 25 bin, 2. kuralın *orta* isimli bulanık kümesine $\mu_{orta}^{Km,2}$= 0.52 üyelik derecesi ile dahilken, 3. kuralın *düşük* isimli bulanık kümesine $\mu_{düşük}^{Km,3}$= 0.48 üyelik derecesi ile dahildir (Bu örnek için daha anlaşılır olması amacıyla bulanık üyelik derecesi gösteriminin $\mu_{Bulanık\ kümenin\ adı}^{Niteliğin\ adı,Kural\ No}$ şeklinde olduğu kabul edilmiştir). Mamdani çıkarım sisteminin ikinci adımı olan kuralların birleştirilmesinde

minimum operatörü kullanılmıştır. 2. kuralın uygunluk değeri $\mu_2 = min(\mu_{orta}^{Model,2}, \mu_{orta}^{Km,2})$ = 0.24 olarak hesaplanırken, 3. kuralın uygunluk değeri $\mu_3 = min(\mu_{yüksek}^{Model,3}, \mu_{düşük}^{Km,3})$ = 0.48 olarak hesaplanır. Daha sonra kuralların çıkış bulanık kümeleri maksimum operatörü kullanılarak birleştirilir. Son adım durulaştırma işlemidir. Birleştirilmiş olan bulanık kümelerin ağırlık merkezi denklem 4.16 kullanılarak hesaplanır ve kesin çıkış değeri olarak 28.4 bin TL rakamı elde edilir.

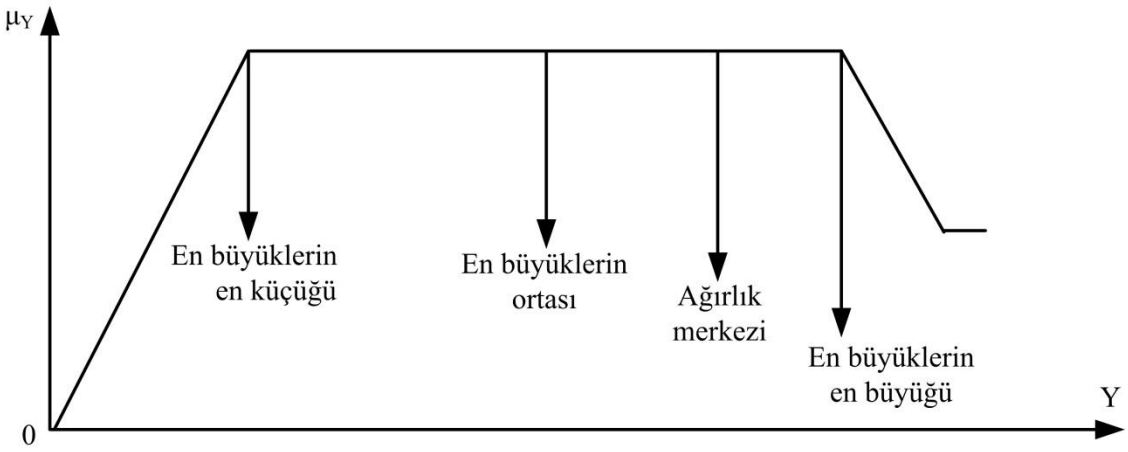


Şekil 4.15. Mamdani çıkarım mekanizmasına bir örnek

4) Durulaştırma: Bulanık çıkarım mekanizmasından gelen ve bulanık olan verilerin kesin sonuçlar haline dönüştürülmesi için yapılan işlemlere durulaştırma işlemleri denir. Durulama birimi çıkarım biriminden gelen bulanık bir bilgiden bulanık olmayan ve uygulamada kullanılacak gerçek değerlerin elde edilmesini sağlar (Elmas, 2003).

Tanımlanmış pek çok durulaştırma yöntemi mevcuttur ve bunların bir kısmı Şekil 4.16'da gösterilmiştir. En sık kullanılan durulaştırma yöntemlerinden birisi *Ağırlık Merkezi* yöntemidir. Denklem 4.16 da ağırlık merkezinin hesaplanmasında kullanılan formül verilmiştir.

$$Ağırlık Merkezi = \frac{\int \mu(y).ydy}{\int \mu(y).dy} \quad (4.16)$$



Şekil 4.16. Durulama yöntemleri

Bunların dışında durulama yöntemi olarak,

1) Tüm bulanık çıktıların birleşiminde en büyük ya da en küçük üyelik derecelerini veren *en büyüklerin en büyüğü* ve ya *en büyüklerin en küçüğü yöntemleri;*
2) En büyük üyelik derecesine sahip bulanık çıktıların ortalamasını veren *en büyüklerin ortalaması yöntemi;*
3) Bulanık çıktılarda en büyük üyelik derecesine sahip bulanık küme değerini veren *yükseklik yöntemi;*
4) Sadece simetrik üyelik fonksiyonlarında kullanılan ve bulanık çıktının ağırlık ortalamasını veren *ağırlıklı ortalama yöntemi;*
5) Bulanık çıktıların birleşimi yerine cebirsel toplamlarını kullanan *toplamların merkezi yöntemi;*
6) Bulanık çıktıların en büyük alanının ağırlık merkezini kullanan *en büyük alanın merkezi yöntemi* kullanılabilinir (Ross,1995).

Durulanmış değeri üyelik fonksiyonu belirler. Durulaştırma birimi daima sayısal bir değer hesaplar. Bu durulaştırmanın tanımı gereğidir. Bazen, iki bulanık küme aynı durulaştırılmış değeri verebilir. Ayrıca, durulaştırılmış değerin daima orijinal bulanık kümenin desteği üzerinde olduğu kabul edilir. İyi bir durulaştırma stratejisi seçmek için sistematik bir işlem yoktur ve bundan dolayı uygulamanın özelliklerini dikkate alan bir yöntem seçilmesi gerekir (Baykal ve Beyan, 2004b).

4.3.2. Bulanık bir sistemin parametreleri

Bulanık bir sistemde, her bölümün yerine getirmekle yükümlü olduğu görevler için çeşitli parametreler bulunmaktadır. Bu parametreler mantıksal, yapısal, bağıntısal ve işlemsel olarak sınıflandırılabilir. Bu parametrelerin seçimi ortaya konulan sistemin yapısını belirler. Çizelge 4.3'de hangi parametrenin hangi sınıfa dâhil olduğu ve bulanık sistemin genel yapısında hangi bölüme ait olduğu gösterilmiştir (Baykal ve Beyan, 2004b).

Çizelge 4.3. Bulanık sistemin parametreleri

Sınıf	Parametreler	Bölüm
Mantıksal	Çıkarım mekanizması Bulanık işlemciler	Bulanık Çıkarım Birimi
	Üyelik fonksiyonunun tipi	Bulanıklaştırıcı, Durulaştırıcı
	Durulama yöntemi	Durulaştırıcı
Yapısal	İlişkili değişkenler Üyelik fonksiyonlarının sayısı	Bilgi Tabanı
	Kuralların sayısı	Kural Tabanı
Bağlantısal	Kuralların öncülleri Kuralların sonuçları Kuralların ağırlıkları	
İşlemsel	Üyelik fonksiyon değerleri	Veritabanı

Mantıksal parametreler, bulanık modelin çalışma yapısını belirleyecek olan parametrelerdir. Bu parametreler ile çıkarım mekanizmasının türü, çıkarım esnasında hangi bulanık işlemcilerin kullanılacağı, üyelik fonksiyonlarının tipi ve durulama yöntemi belirlenir. Yapısal parametreler sistemde yer alacak ilişkili değişkenlerin, üyelik fonksiyonlarının sayısını ve kuralların sayılarını ifade eden parametrelerdir. Bağlantısal parametreler ile kuralların yapısı tanımlanır. Son olarak işlemsel parametre üyelik fonksiyon değerlerini tutar.

Bu tez çalışmasında, bağlantısal ve işlemsel parametreler ile yapısal bir parametre olan üyelik fonksiyonları sayısına ait en uygun değerlerin, verilen eğitim kümesi kullanılarak evrimsel algoritmalar yardımıyla tespit edilmesi amaçlanmıştır.

4.3.3. Bulanık bir sistemin yorumlanabilirliği

Bulanık bir sistemin yorumlanabilirliği sistemin anlaşılabilirliğinin göstergesidir. Daha çok son yıllarda araştırmacıların ilgisini çeken bir konudur. Yorumlanabilirlik nesnel bir kavram olduğundan ve pek çok farklı kıstas yorumlanabilirlik ölçütü olarak

kullanılabileceğinden, yorumlanabilirliği ölçmek için genel olarak kabul edilmiş bir standart henüz oluşmamıştır. Ancak pek çok araştırmacının fikir birliği sağladığı ölçütler mevcuttur. Bunlar:

- Kural sayısı mümkün olduğunca küçük olması,
- Kuralın öncül-soncul kısımlarında mümkün olduğunca az değişken olması,
- Üyelik fonksiyonları sezgisel olarak anlamlı olması şeklindedir.

Literatürde sıklıkla kullanılan bu ölçütler aslında bulanık bir sistemde bulunan ve Bölüm 4.3.2'de açıklanan parametreler ile yakından ilgilidir. Gacto ve ark. (2011) bulanık bir sistemin yorumlanabilirliğini ifade etmek için bulanık sistem parametrelerine dayanan ve Çizelge 4.4'de gösterilen sınıflandırmayı yapmışlardır.

Çizelge 4.4. Yorumlanabilirlik ölçütlerinin sınıflandırılması

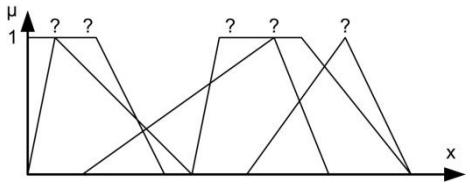
	Kural Tabanı Seviyesi	**Bulanık Kısım Seviyesi**
Karmaşıklık tabanlı yorumlanabilirlik	1. Kural Sayısı 2. Kuraldaki niteliklerin sayısı	1. Üyelik fonksiyonlarının sayısı 2. Niteliklerin Sayısı
Anlamsal tabanlı yorumlanabilirlik	1. Kural tabanının tutarlılığı 2. Aynı anda tetiklenen kural sayısı 3. Kural ağırlıkları	1. Tamlığı veya kapsamı 2. Normalliği 3. Ayrıklaştırılabilirliği 4. Tamamlayıcılığı

Bu kıstasların kısa özetleri aşağıda verilmiştir (Gacto ve ark., 2011) :

- *Kural Sayısı:* "Sistem davranışına uygun olan en basit model en iyi modeldir" şeklindeki prensibe göre, bulanık modelin performansını kabul edilen bir seviyede koruyan kural tabanları arasında en az kurala sahip olan kural kümesi yorumlanabilir bir bulanık sistem için en uygunudur.
- *Kuraldaki niteliklerin sayısı:* Miller'e (1956) göre insanın aynı anda kavrayabileceği kavram sayısı 7±2'dir. Buradan yola çıkarak bir bulanık kuralın mümkün oldukça anlaşılabilirliğin yüksek olması için, öncül kısmındaki niteliklerin sayısının mümkün olduğunca az olması (sistem performansını etkilememek şartı ile) ve 7±2'yi geçmemesi tavsiye edilir.
- *Niteliklerin veya değişkenlerin sayısı:* Yüksek boyutlu problemlerde boyutun azaltılması bilgi tabanının okunabilirliğinin artırabilir.
- *Üyelik fonksiyonlarının sayısı:* Kuralda yer alan nitelik sayısının tespitinde kullanılan mantık ile benzerlik gösterir. Bir niteliği bulanıklaştırmak için

kullanılan üyelik fonksiyonlarının sayısı 7±2 sınırını aşmamalıdır ki insanın yorumlayabileceği kavram sayısı (Miller,1956) geçilmemiş olsun.

- *Kural tabanının tutarlılığı:* Kural tabanındaki çelişkili kuralların olmaması gerekir. Yani benzer öncül kısma sahip kuralların, benzer sonuç kısımları olmalıdır.
- *Aynı anda tetiklenen kural sayısı:* Bir giriş için aynı anda tetiklenen kuralların sayısının minimize edilmesi istenir. Böylece kural tabanını oluşturan dilsel kuralların bireysel anlamları korunarak, kural tabanında aynı işi yapan birden çok kuralın bulunması engellenir.
- *Tamlık veya Kapsama*: Bir değişkenin evrensel uzayının (Universe of Discourse - UoD) tamamının üyelik fonksiyonları tarafından kapsanması gerekir. Yani evrensel uzaydaki her bir nokta en azından bir bulanık kümeye ait olmalı ve dilsel bir terimle etiketlenmiş olmalıdır.
- *Normalliği:* Bir değişkenin evrensel uzayında bulunan en az bir eleman, üyelik fonksiyonunda 1 değerini alıyorsa, bu küme normal bulanık küme olarak isimlendirilir (Şekil 4.8). Bulanık sistemlerde yer alan bulanık kümelerin normal olması gerekir.
- *Ayrıklaştırılabilirliği:* Bir üyelik fonksiyonu anlamlı bir manası olan bir dilsel terimi temsil etmeli ve ilgili değişkenin diğer üyelik fonksiyonlarından net bir şekilde ayrıklaştırılabilir olmamalıdır. Sistemin bir değişkeni için anlamsal olarak yorumlanabilirliği düşük bir bulanık bölümle örneği Şekil 4.17'de verilmiştir.
- *Tamamlayıcılığı:* Evrensel uzayın her bir elemanı için, üyelik fonksiyonlarının toplamı 1'e yakın bir sayı olmalıdır.



Şekil 4.17. Yorumlanabilirliği düşük bir bulanık bölümlemeye örnek

Bu tez çalışmasında önerilen ve Bölüm 6.1'de açıklanan ön işlem algoritması ile üyelik fonksiyonlarının tam, ayrıklaştırılabilir, tamamlayıcı olması ve üyelik fonksiyonunun sayısının makul seviyede seçilmesi sağlanmıştır. Niteliklerin ve

değişkenlerin sayısı ise Bölüm 6.2.3'de anlatılan başlangıç popülasyonu üretilirken izlenen yöntem ile kontrol altında tutulmuştur. Bölüm 4.4'de anlatılan ve Ishibuchi ve ark. (1997) tarafından önerilen algoritmanın kullanımı ile de, çelişkili kuralların veri tabanına dahil olması baştan engellenmiştir. Böylece kural tabanının tutarlılığı temin edilmiş ve tasarlanan sistemin yorumlanabilirliği mümkün oldukça artırılmıştır.

4.4. Bulanık Sınıflandırıcı Sistemi

Sınıflandırma, nitelik uzayını çeşitli alanlara bölme ve nesneleri bu alanlar ile tanımlanan sınıflar ile ilişkilendirme problemidir. Karar ağaçları, istatistiksel metotlar, Eğer - O halde kuralları, yapay sinir ağları, evrimsel algoritmalar, bulanık mantık gibi yöntemler kullanılarak sınıflandırma modelleri oluşturmak mümkündür. Bulanık mantık, kesin olmayan veya eksik olan verilerle çalışma imkanı sunması ve dilsel değerlerden oluştuğu için kural tabanının kullanıcılar tarafından daha anlaşılır bulunması sayesinde sınıflandırma sistemi tasarımında kullanılan güçlü bir yöntem haline gelmiştir (Mansoori ve ark., 2008).

Bulanık bir sınıflandırma sistemi Eğer – O halde kuralları ile temsil edilir. Bulanık sınıflandırıcı olarak da isimlendirilen kural tabanlı bir bulanık sınıflandırma sistemi, çıkışı sabit olan özel bir bulanık sistem türüdür. *n*-boyutlu sürekli bir örüntü uzayında *c*-sınıflı bir örüntü sınıflandırma problemi için *c* sınıflarına ait *v* tane etiketlenmiş eğitim örüntümüzden *p.* eğitim verisi $X_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ şeklinde verilirse bu örüntü kümesi için tanımlanacak bulanık bir kuralın gösterimi denklem 4.17 gibi olacaktır.

***K_q*:** Eğer x_{p1} A_{q1} ve … ve x_{pi} A_{qi} ve... ve x_{pn} A_{qn}, O Halde Sınıf S_q, Kf_q ile (4.17)

Burada;

K_q: Bulanık kuralının etiketini

A_{qj} : Örüntü vektörünün *j*. niteliği için *q*. bulanık kuralın öncül (antecedenet) gösteren bulanık kümeyi

S_q : K_q kuralının sonuç sınıfını (s sınıflarından bir tanesi)

Kf_q : K_q kuralının ağırlığını (Kesinlik faktörü, [0-1] aralığında gerçek bir sayıdır.) temsil etmektedir.

Öncül kısmı verilen bir bulanık kuralın sonuç kısmını oluşturan sınıfının ve kesinlik faktörünün hesaplanması için Ishibuchi ve ark. (1997) geliştirmiş oldukları yöntem bulanık sınıflandırıcı tasarımı çalışmalarında sıklıkla kullanılmaktadır. Bu yöntemin adımları aşağıda verildiği gibidir.

a. A_{qi} bulanık kümesine ait üyelik fonksiyonunun μ_{Aqi} ile gösterildiğini farz edelim. Eğitim kümesine ait X_p örüntüsünün K_q bulanık kuralına uygunluk derecesi μ_q denklem 4.18'de verilen şekilde hesaplanır.

$$\mu_q(X_p) = \mu_{A_{q1}}(X_{p1}) * \mu_{A_{q2}}(X_{p2}) * \ \dots * \mu_{A_{qn}}(X_{pn}) \quad (4.18)$$

b. Her bir sınıf için eğitim örüntülerinin K_q kuralı ile olan uygunluk derecelerinin toplamı denklem 4.19'a göre hesaplanır.

$$\beta_s(K_q) = \sum_{X_p \epsilon Sınıf_s} \mu_q(X_p) \quad (4.19)$$

c. En büyük değere sahip olan $\beta_s(K_q)$ sınıfı $\hat{s}$ olarak belirlenir. Bu işlemin matematiksel ifadesi denklem 4.20'de verilmiştir.

$$\beta_{\hat{s}}(K_q) = maks\{\beta_1(K_q), \dots, \beta_c(K_q)\} \quad (4.20)$$

Eğer tek bir sınıf maksimum değere sahipse, kuralın çıkışı olan S_q, $\hat{s}$ sınıfı olarak belirlenir. Ancak birden fazla sınıf maksimum değere sahipse ve ya K_q bulanık kuralı hiç bir eğitim örüntüsü ile uyumlu değilse (yani s=1,2,…,c için $\beta_s(K_q) = 0$ ise) kuralın çıkışı S_q tespit edilemez. Bu durumda S_q = 0 olarak atanır.

d. Son adımda, sınıfı belirlenen bulanık kuralın kesinlik faktörü denklem 4.21 ve denklem 4.22 kullanılarak hesaplanır. Ancak bulanık kuralın sınıfı 0 olarak belirlenmiş ise kuralın kesinlik faktörü de 0 olarak atanır.

$$KF_q = \{\beta_{\hat{s}}(K_q) - \bar{\beta}\} / \sum_{s=1}^{c} \beta_s(K_q) \quad (4.21)$$

$$\overline{\beta} = \sum_{s=1, s\neq\hat{s}}^{c} \beta_s(K_q)/(c-1) \quad (4.22)$$

Örneğin henüz sınıfı ve Kf'si hesaplanmamış bir bulanık sınıflandırıcı kuralı olan K_u kuralı denklem 4.23'de verildiği gibi olsun. İlk önce kuralın uygunluk değeri denklem 4.18'e göre hesaplanır.

K_u= **Eğer** x_1 çok ise **ve** x_2 az ise **ve** x_3 orta ise **O halde** Sınıfı=?, Kf=? (4.23)

$$\mu_u(x_p) = \mu_{çok}^{x1,u} * \mu_{azçok}^{x2,u} * \mu_{orta}^{x3,u}$$

Örneğin;

$$\mu_u(x_p) = 0.8 * 0.5 * 0.5 = 0.2$$

X_p eğitim verisi için **K_u kuralının** uygunluk değerinin hesaplanmasına bir örnek

K_u bulanık kuralın sınıfının ve kesinlik faktörünün hesaplanması *v*=8 tane eğitim verisi içeren ve Çizelge 4.5'de verilen örnek eğitim kümesi kullanılarak aşağıda açıklanmıştır. Eğitim kümesinde bulunan niteliklerin sayısı *n*=4 ve sınıf sayısı *c*=3'dür. Ayrıca Çizelge 4.5'in 3. sütununda, her bir x_p eğitim verisi için K_u kuralının uygunluk değeri de gösterilmiştir.

Çizelge 4.5. Örnek bir eğitim kümesi

Eğitim verisi (X_p)	**Sınıfı**	**$\mu_u(X_p)$**
1. [6.7 3.0 5.2 2.3]	1	0.8
2. [5.2 3.1 6.3 1.4]	1	0.5
3. [4.6 4.2 3.5 2.8]	2	0.3
4. [...]	3	0.2
5. [...]	3	0.4
6. [...]	2	0.1
7. [...]	1	0.6
8. [7.5 5.1 3.5 0.8]	3	0.9

Denklem 4.19 kullanılarak hesaplanan her bir sınıf için eğitim verilerin K_u kuralı ile olan uygunluk derecelerinin toplamları aşağıda verildiği gibidir.

1. Sınıfın toplam uygunluğu = 1.9

2.Sınıfın toplam uygunluğu = 0.4

3.Sınıfın toplam uygunluğu = 1.5

K_u kuralı ile olan uygunluk derecelerinin toplamı en büyük olan sınıf, denklem 4.20'ye göre K_u kuralının çıkışı olarak atanır. Bu örnek için K_u kuralının çıkışı 1'dir.

K_u kuralının kesinlik faktörü denklem 4.21 ve 4.22 kullanılarak aşağıdaki gibi hesaplanır.

$$\overline{\beta} = \sum_{s=1, s\neq\hat{s}}^{c} \frac{\beta_s(K_q)}{c-1} \qquad KF_q = (\beta_{\hat{s}}(K_q) - \bar{\beta}) / \sum_{s=1}^{c} \beta_s(K_q)$$

$$\overline{\beta} = \frac{0.4+1.5}{2} \qquad KF_q = (1.9 - 0.95)/(\frac{1.9+0.4+1.5}{3})$$

$$\overline{\beta} = 0.95 \qquad KF_q = 0.748$$

Tek Kazanan Stratejisi:

Tek kazanan stratejisi (single winner strategy), bulanık sınıflandırıcı sistemine verilen bir örüntünün, bulanık sınıflandırıcı kuralları kullanılarak hangi sınıfa ait olduğunun tespitinde kullanılan bir bulanık çıkarım yöntemidir.

B tane kurala sahip bir *BS* bulanık sınıflandırıcımız olduğunu farz edelim. *BS* sınıflandırıcısına sunulan bir $X_{giriş}$ giriş örüntüsünün sınıfının tek kazanan stratejisine

göre tespiti denklem 4.24 kullanılarak yapılır. Denklem 4.24, $X_{giriş}$ giriş örüntüsü ile en yüksek uygunluk derecesine sahip olan K_w kuralını tespit eder. Daha sonra K_w kuralının çıkışı, $X_{giriş}$ örüntüsünün sınıfı olarak atanır.

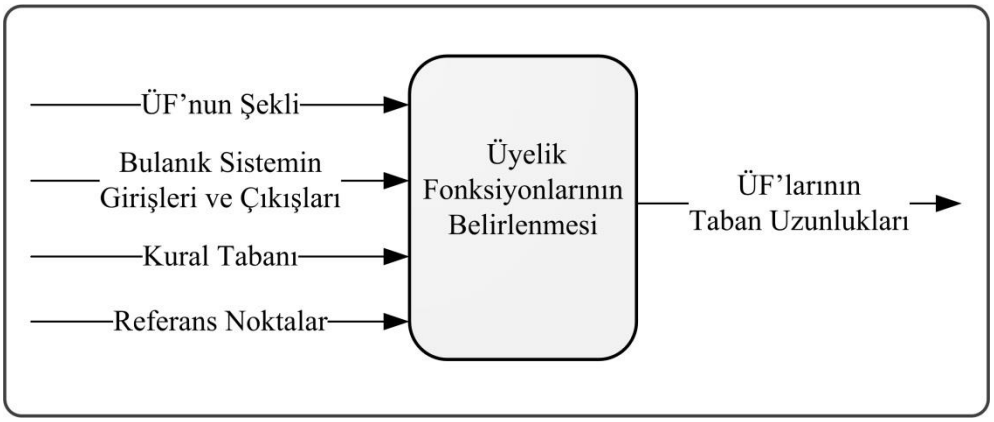
$$w = \arg \underset{1 \le q < B}{maks}(\mu_q(X_{giriş})) \tag{4.24}$$

Eğer bir $X_{giriş}$ örüntüsü için, farklı sınıflara sahip kurallar aynı uygunluk derecesine sahip ise, kesinlik faktörü (*KF*) büyük olan kural tek kazanan olarak seçilir ve $X_{giriş}$ örüntüsünün sınıfı bu kuralın çıkışı olarak atanır. Ancak *KF*'de de bir eşitlik söz konusu ise, $X_{giriş}$ örüntüsü sınıflandırılamaz.

5. ÜYELİK FONKSİYONLARININ OPTİMİZASYONU İÇİN ÖNERİLEN YAPAY BAĞIŞIKLIK SİSTEMİ TABANLI BİR YÖNTEM

Bulanık mantık sistem tasarımında üyelik fonksiyonlarına (ÜF) ait parametrelerin en uygun değerlerinin belirlenmesi bulanık mantığın çekirdeğinde bulunan derecelendirme kavramına karşılık gelir ve oldukça önemli bir adımdır. Çünkü dilsel değerlerin sistem tasarımında kullanılmasını mümkün kılan bulanık kümelerin her biri, üyelik fonksiyonları kullanılarak karakterize edilir. Bundan dolayıdır ki, pek çok bulanık mantık kontrol problemi, üyelik fonksiyonlarını tanımlayan parametrelerin tespiti konusuyla yakından ilgilidir. Bu parametrelerin tespiti için kullanılabilecek iki potansiyel kaynak vardır: uzman görüşü ve gerçek dünya verileri. Ancak bir uzman görüşüne ulaşmak her zaman mümkün olmayabilir, mümkün olduğunda da uzmandan çok büyük ve dinamik yapıda olan bir veri kümesinden bulanık kümeleri doğru şekilde tespit etmesini beklemek zordur. Bu yüzden gerçek dünya verilerinden üyelik fonksiyonlarının tespiti araştırmacıların ilgisini çeken bir konu olmuştur.

Kural tablosu ve üyelik fonksiyonlarının şekli daha önceden verilen tek girişli ve tek çıkışlı bir bulanık mantık sistemine ait üyelik fonksiyonlarının en uygun yerleşiminin belirlenmesi için Arslan ve Kaya (2001) bir yöntem sunmuşlar ve yöntemlerini GA kullanarak kodlamışlardır. Bu tez çalışmasının ilk adımında bu yöntemin daha da geliştirilmesi hedeflenmiştir. Bunun için yöntem bir optimizasyon problemi olarak ele alınmış ve çözümü için bir yapay bağışıklık algoritması olan CLONALG'nin kullanılması önerilmiştir. Önerilen yöntemin genel yapısı Şekil 5.1'de gösterilmiştir.



Şekil 5.1. Önerilen yöntemin genel yapısı

Şekil 5.1'den görüldüğü üzere, bulanık mantık sistemine ait üyelik fonksiyonlarının belirlenmesi için kullanılan yöntem üyelik fonksiyonunun şeklini, bulanık mantık sistemine ait giriş çıkış değişkenlerini ve onların tanımlandığı aralıkları, bulanık kural tabanını ve referans noktaları giriş değişkenleri olarak alır. Bu değişkenlerden faydalanan bir algoritma, bulanık mantık sisteminin ÜF'ları için en uygun taban uzunluklarını öğrenir yani taban uzunluklarını optimize eder. Diğer bir deyişle derecelendirme işlemini gerçekleştirir. Çünkü üyelik fonksiyonunun taban uzunluğu değiştikçe, bireylerin o kümeye olan aitlik dereceleri de değişmektedir. Yani bir bireyin bir bulanık kümeye olan aitliğinin en uygun derecesi tespit edilmektedir. Bu işlem için Arslan ve Kaya (2001) genetik algoritmayı kullanırken bu tez çalışmasında CLONALG algoritması kullanılmış ve bu algoritmaların birbirleri ile olan kıyaslamaları yapılmıştır.

Bu bölümde öncelikle bireyin kodlanması ve duyarlılığın/uygunluğun hesaplanması anlatılmış daha sonra üyelik fonksiyonların tek girişli/çıkışlı (Şakiroğlu ve Arslan, 2007) ve çok girişli/çıkışlı bulanık sistemler için (Acılar ve Arslan, 2011) nasıl optimize edildiği birer örnek üzerinde adım adım açıklanmış ve son olarak deneysel sonuçlara yer verilmiştir.

5.1. Antikorun/Bireyin Kodlanması

Kullanılan algoritmaya göre bireyin aldığı isim değişmektedir. Bu çalışmada yapay bağışıklık algoritması olan CLONALG'nin kullanılması önerildiği için, çalışmanın devamında birey yerine antikor (Ab) kavramı kullanılacaktır.

Sistemdeki antikorlar üyelik fonksiyonlarının taban uzunluklarından ve gerekirse kesişim noktalarından oluşmaktadır. İkili şekilde kodlanmışlardır. Antikorun uzunluğunu,

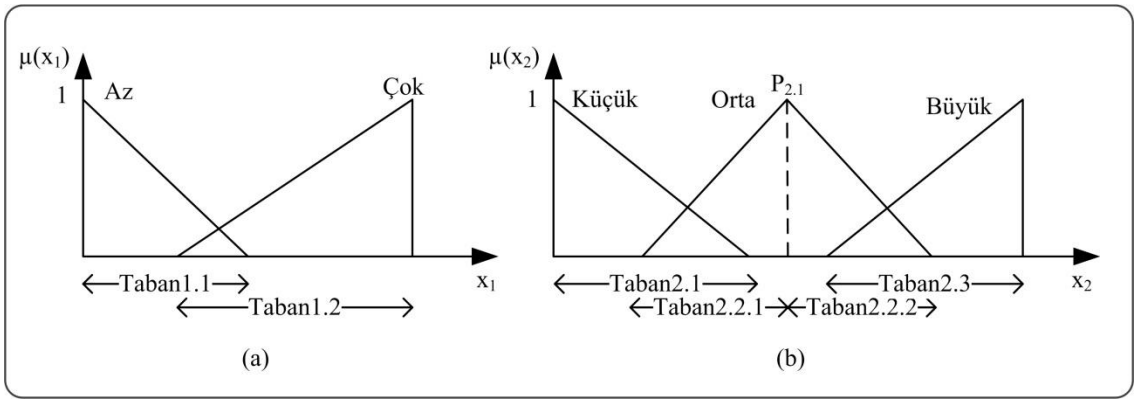
1) Üyelik fonksiyonunun şekline;

2) Bulanık mantık sisteminde bulunan giriş ve çıkış değişkenlerinin sayısına;

3) Her bir değişken için kullanılan üyelik fonksiyonu sayısına;

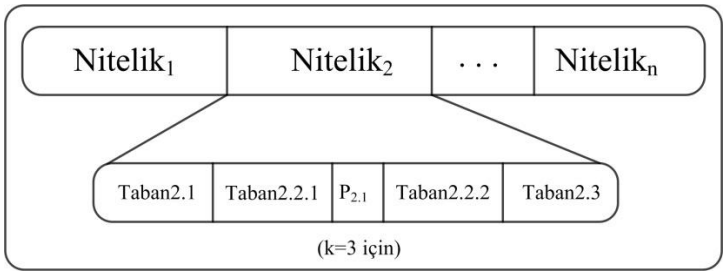
4) Sistemdeki değişkenlerin en büyüğünün üst sınırının ikili olarak kodlanması için gerekli olan bit sayısına göre değişmektedir. Antikorun kodlanma şeklini yakından ilgilendiren bu faktörleri Şekil 5.2'den faydalanarak inceleyelim:

Seçilen üyelik fonksiyonunun şekline göre kodlamada yer alacak taban sayısı değişmektedir. Bu çalışmada üçgen üyelik fonksiyonu kullanılmıştır. Bir değişkene ait ilk ve son üyelik fonksiyonları dik üçgen ile diğerleri ise çeşitkenar üçgen ile temsil edilmiştir. Bir değişkeni göstermek için en az iki üyelik fonksiyonu kullanılmalıdır ki bu durumda iki üçgende dik üçgen olur. Dik bir üçgeni temsil etmek için tek bir taban uzunluğu yeterlidir. Dik üçgenin tepe noktası ilk veya son üyelik fonksiyonu olmasına göre, ilgili değişkenin ya alt ya da üst sınırıdır. Örneği Şekil 5.2(a)'da gösterildiği gibidir. Şekil 5.2(b)'de de sistem değişkenin ikiden fazla üyelik fonksiyonuna sahip olduğu zamanki durum gösterilmiştir. Örnekte x_2 değişkeni *küçük, orta* ve *büyük* dilsel etiketlerine sahip 3 adet üyelik fonksiyonu bulunmaktadır. Birinci ve üçüncü üyelik fonksiyonları Şekil 5.2(a)'da olduğu gibi dik üçgendir ve tek taban uzunluğu ile temsil edilmişlerdir (*Taban2.1* ve *Taban2.3*). Ancak *orta* isimli ikinci üyelik fonksiyonu çeşitkenar bir üçgendir ve kodlamadaki temsili için 3 elemana ihtiyaç duyulmaktadır. Bunlar şekilde $P_{2.1}$ olarak gösterilen üçgenin tepe noktası ve tepe noktasına olan uzaklıkları gösteren *Taban2.2.1* ve *Taban2.2.2* olarak gösterilen taban uzunluklarıdır.



Şekil 5.2. Örnek üyelik fonksiyonu gösterimleri

Şekil 5.3'de *n* tane niteliğe sahip bir bulanık sistem için, üyelik fonksiyonlarının taban değerlerini tutan bir antikorun yapısı gösterilmiştir. Nitelik 2'nin k=3 dilsel değeri olduğu farz edildiğinde nitelik 2'nin içeriğinin nasıl olacağı da gösterilmiştir. Taban değişkenlerinin karşılıkları Şekil 5.2(b)'de görülmektedir.



Şekil 5.3. Antikorun yapısı

Antikorun uzunluğunu etkileyen faktörlerden birisinin de sistemde yer alan değişkenlerin üst sınırı en yüksek olanının ikili olarak kodlanması için gerekli olan bit sayısıdır. Bu sayı her bir tabanın kaç bit ile ifade edileceğini belirler. Çünkü antikor kodlamasında her bir taban uzunluğu ikili formatta bulunur. Bu değer *m* bit ile ifade ediliyorsa, diğer tüm değişkenlerin gösterimi içinde *m* bit kullanılır. *m* sayısının hesaplanması denklem 5.1'de verilmiştir.

$$m = YukarıYuvarla\{maks(\text{maks}_{1\le i\le n}(\log_2 X^i_{maks}), \text{maks}_{1\le j\le u}(\log_2 Y^j_{maks}))\} \quad (5.1)$$

Burada;

n = bulanık sistemde bulunan nitelik (giriş) sayısını,

u = bulanık sistemde bulunan çıkış sayısını,

X_{maks} = bulanık sistemdeki i. giriş değişkeninin üst sınırını,

Y_{maks} = bulanık sistemdeki j. çıkış değişkeninin üst sınırını gösterir.

Bu bilgilerin ışığında bir antikorun kaç bit uzunluğunda olduğu denklem 5.2 ve 5.3 kullanılarak hesaplanabilir.

$$NitelikUz_i = [(k_i - 2) * 3 + 2] * m \quad (5.2)$$

$$AntikorUz = \sum_{i=1}^{n} NitelikUz_i \quad (5.3)$$

Burada;

n = Bulanık sistemde bulunan nitelik sayısını,

k_i = i. Nitelikte bulunan dilsel değerlerin (üyelik fonksiyonların veya bulanık kümelerin) sayısını (i=1,2, … , n),

m = sistemdeki değişkenlerin üst sınırı en yüksek olanının ikili olarak kodlanması için gerekli bit sayısını gösterir.

Denklem 5.1'de her bir niteliğin uzunluğunun nasıl hesaplanacağı verilmiştir. k_i'den 2 çıkarılmasının sebebi, sistemdeki değişkenleri temsil eden üyelik fonksiyonlarının birincisinin ve sonuncusun dik üçgen olmasından dolayıdır. Daha önce de ifade edildiği gibi dik üçgenleri temsil etmek için tek bir taban uzunluğu yeterlidir.

Geriye kalan tabanları kodlamak için ise 3 değer lazımdır (2 taban uzunluğu ve bir kesişim noktası). Bu yüzden (k_i-2) değeri 3 ile çarpılır. Daha sonra dik üçgenlerin taban sayıları için 2 sayısı formüle eklenir. Son olarak bu ifade *m* ile çarpılarak *i.* niteliğin kaç bit uzunluğunda olduğu bulunur. Denklem 5.3'de de tüm niteliklerin uzunlukları toplanarak antikorun kaç bit uzunlukta olacağı hesaplanır.

5.2. Duyarlılık Fonksiyonunun Hesaplanması

Her bir antikorun duyarlılığı hesaplanmadan önce, antikorlarda yer alan taban değerlerinin ondalık sayı sistemine çevrilmesi gerekir. Çevirme işlemi yapılırken dikkat edilmesi gereken önemli bir nokta vardır. Şöyle ki, sistemdeki her bir değişkenin kodlanması için gerekli olan bit sayısı *m,* denklem 5.1 kullanılarak hesaplanırken sistemde bulunan değişkenlerden üst sınırı en yüksek olanının referans alındığı görülmektedir. Bunun anlamı her bir değişken için üst sınırın sistemdeki en yüksek üst sınıra eşit olacağıdır. Ancak her değişkenin kendi alt ve üst sınırları mevcuttur. Bundan dolayı her bir değişken denklem 5.4 kullanılarak kendi sınırları arasına yansıtılarak ondalık sayıya çevrilir.

$$D_i = D_{min}^i + \frac{desDeg_i}{(2^m - 1)} * (D_{maks}^i - D_{min}^i) \quad (5.4)$$

Burada;

D_i = i. değişkenin kendi sınırları arasındaki ondalık değerini,

D_{min}^i= i. değişkenin alt sınırını,

D_{maks}^i= i. değişkenin üst sınırını,

$desDeg_i$ = i. değişkeni ondalık değerini,

m = sistemdeki değişkenlerin üst sınırı en yüksek olanının ikili olarak kodlanması için gerekli bit sayısını gösterir.

Duyarlılık fonksiyonu olarak Arslan ve Kaya'nın (2001) tek girişli/çıkışlı bir bulanık sistem için kullandıkları ve denklem 5.5'de verilen fonksiyon kullanılmıştır.

$$Duyarlılık = MaksHata - ToplamHata \quad (5.5)$$

Burada *MaksHata* sistemin verebileceği maksimum hata miktarıdır ve denklem 5.6 kullanılarak hesaplanır. *ToplamHata* sistemin o anki taban değerlerine göre verdiği hatadır ve denklem 5.7 kullanılarak hesaplanır.

$$MaksHata = \sum_{p=1}^{v}(y_p - y_{maks})^2 \quad (5.6)$$

$$ToplamHata = \sum_{p=1}^{v}(y_{CLONALG_p} - y_p)^2 \quad (5.7)$$

Burada;

v = eğitim kümesinin içerdiği örüntü sayısını,

y_{maks} = sistem çıkışın alabileceği maksimum değeri,

y_p = p. eğitim verisine karşılık gelen gerçek çıkış değerini,

$y_{CLONALGp}$ = p. eğitim verisi için CLONALG algoritması ile hesaplanan çıkış değerini gösterir.

Amaç *ToplamHata'*nın mümkün olduğunca 0'a yaklaştırılmasıdır. *Duyarlılık, MaksHata*'dan *ToplamHata* çıkarılarak elde edildiği için problem bir maksimizasyon problemine dönüşmüş olur.

Bulanık sistemin çok girişli/çıkışlı olduğu durumda duyarlılığın hesaplanması için Kaya ve ark.'nın (2001) kullandığı, denklem 5.8'de verilen ve denklem 5.5'e oldukça benzeyen bir formülden faydalanılmıştır. Tek farkı sırasıyla denklem 5.9 ve 5.10 da verilen *MaksHata* ile *ToplamHata'nın* çoklu sisteme uyarlanmış hallerinin kullanılmasıdır.

$$Duyarlılık_{çoklu} = MaksHata_{çoklu} - ToplamHata_{çoklu} \quad (5.8)$$

$$MaksHata_{çoklu} = \sum_{p=1}^{v}\sum_{r=1}^{u} maks[\left(y_{r_{CLONALGp}} - y_{r_{min}}\right), (y_{r_{maks}} - y_{r_{CLONALGp}})]^2 \quad (5.9)$$

$$ToplamHata_{çoklu} = \sum_{p=1}^{v}\sum_{r=1}^{u}\left(y_{r_{CLONALGp}} - y_{r_p}\right)^2 \quad (5.10)$$

Burada;

v = eğitim kümesinin içerdiği örüntü sayısını,

u = bulanık sistemde bulunan çıkış sayısını,

y_{r_p}= p. eğitim verisine karşılık gelen r. çıkışın değerini,

$y_{r_{min}}$= r. çıkış değerinin alt sınırını,

$y_{r_{maks}}$= r. çıkış değerinin üst sınırını,

$y_{r_{CLONALGp}}$ = p. eğitim verisi için CLONALG algoritmasından gelen taban uzunluklarına göre hesaplanan çıkış değerini gösterir. $y_{r_{CLONALGp}}$ nasıl hesaplandığı Bölüm 5.3'de verilen örnekler üzerinde ayrıntılı şekilde anlatılmıştır.

5.3. Örnek Uygulamalar ve Deneysel Sonuçlar

Bu bölümde, tek girişli/çıkışlı ve çok girişli/çıkışlı bulanık sistemlerde üyelik fonksiyonlarının CLONALG kullanılarak nasıl optimize edildiği birer örnek üzerinde ayrıntılı şekilde açıklanacak ve elde edilen deneysel sonuçlar yorumlanacaktır.

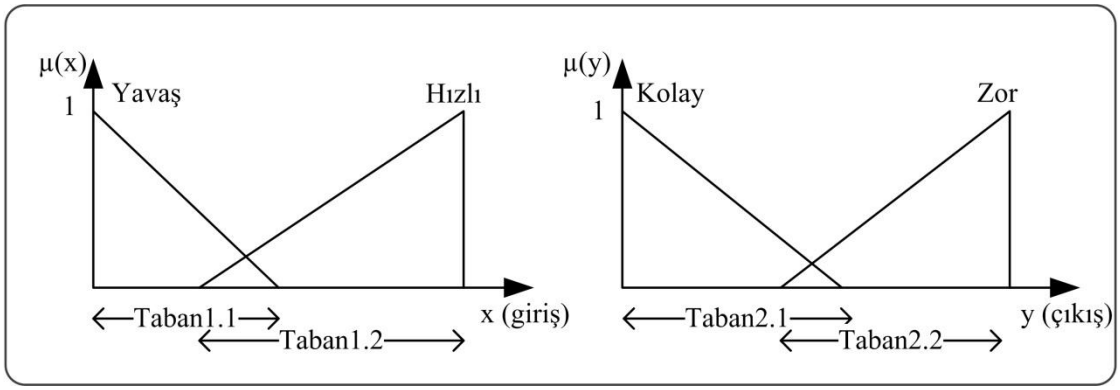
5.3.1. Tek girişli - çıkışlı bir bulanık sistem örneği için ÜF optimizasyonu

Her biri iki dilsel değişkene sahip olan tek girişli/çıkışlı bir bulanık sistem düşünelim. Üyelik fonksiyonlarını *x (giriş)* ve *y (çıkış)* olarak isimlendirelim. *x* değişkeni *yavaş* ve *hızlı* dilsel değerlerini kullanırken; *y* değişkeni *kolay* ve *zor* dilsel değerlerini kullansın. Her bir dilsel değeri temsil etmek için kullanılan üyelik fonksiyonu şekli ise dik üçgen olsun. Örnek bulanık sisteme ait kuralların ve referans giriş/çıkış değerlerin ise aşağıdaki gibi verildiği kabul edilsin.

Kural 1: Eğer *x* yavaş ise *y* kolaydır.
Kural 2: Eğer *x* hızlı ise *y* zordur.

Giriş Değerleri: $x_i = \{1, 3, 5, 7\}$ için;
Çıkış Değerleri: $y_i = \{1, 9, 25, 49\}$, i=1, 2, 3, 4

Giriş *x* değişkenin 0-7, çıkış *y* değişkeninde 0-49 arasında değer alabildiği farz edilirse, sisteme ait üyelik fonksiyonların görüntüsü Şekil 5.4'de gösterilene benzer olacaktır.



Şekil 5.4. Tek giriş/çıkışlı örnek bir bulanık sistem için üyelik fonksiyonlarının gösterimi

Önerilen yöntemde, CLONALG algoritmasından beklenen Şekil 5.4'deki dik üçgenlere ait en uygun taban uzunluklarını, verilen kural ve giriş çıkış değerlerini kullanarak bulmasıdır. Öncelikle her bir taban değerinin kodlanmasında kaç bit kullanılacağı denklem 5.1 kullanılarak hesaplanması gerekmektedir.

$$m = YukarıYuvarla\{maks(\text{maks}_{1\le i\le n}(\log_2 X^i_{maks}), \text{maks}_{1\le j\le u}(\log_2 Y^j_{maks}))\}$$

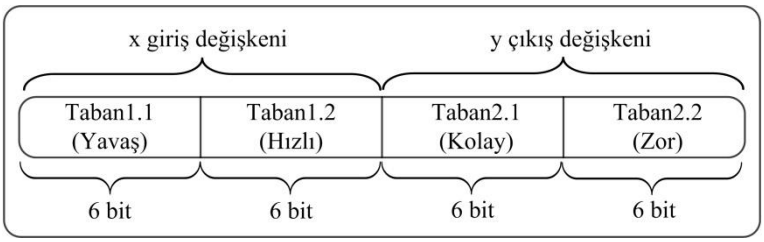
$$m = YukarıYuvarla\{maks(\log_2 7, \log_2 49)\}$$

$$m = YukarıYuvarla\{maks(2.8074,5.6147)\}$$

$$m = YukarıYuvarla\{5.6147\}$$

$$m = 6 \text{ bit.}$$

Böylece her bir taban uzunluğunun ikili olarak kodlanması için gerekli olan bit sayısı 6 olarak bulunmuş olur. Verilen örnek için antikorun yapısı Şekil 5.5'de gösterildiği gibi olacaktır.



Şekil 5.5. Verilen örnek için antikorun yapısı

Şekil 5.5'de verilen antikor yapısı kullanılarak 10 tane antikordan oluşan başlangıç popülasyonu rastgele oluşturulur. Çizelge 5.1'in 1. sütununda oluşturulan bu popülasyonun bireyleri verilmiştir. 2. ve 5. sütunlar arasında taban değerlerinin ondalık sistemdeki karşılıkları görülmektedir. 6. ve 9. sütunlar arasında da tabanların ondalık değerlerinin, ilgili değişkenin kendi alt ve üst sınırları arasına yansıtılmış değeri bulunmaktadır. Bu değerler denklem 5.4 kullanılarak hesaplanmıştır.

Çizelge 5.1. Başlangıç Popülasyonu

1	2	3	4	5	6	7	8	9	10
Antikorlar	Taban 1.1	Taban 1.2	Taban 2.1	Taban 2.2	Taban 1.1Y	Taban 1.2Y	Taban 2.1Y	Taban 2.2Y	Duyar-lılık
0101001100101**11101**0001000 (Ab_1)	20	50	58	8	2.22	5.56	45.11	6.22	2372.3
0101111000011**01001**101000 (Ab_2)	23	33	41	40	2.56	3.67	31.89	31.11	4217.8
1111101100011**00100**000100 (Ab_3)	62	49	36	4	6.89	5.44	28	3.11	4438.5
1010011010010**11011**111010 (Ab_4)	41	41	27	58	4.56	4.56	21	45.11	4449.2
1000101110101**10110**011001 (Ab_5)	34	58	54	25	3.78	6.44	42	19.44	3461.8
0101111101000**11011**111011 (Ab_6)	23	52	27	59	2.56	5.78	21	45.89	4294.3
1111110110010**10001**011110 (Ab_7)	63	25	17	30	7	2.78	13.23	23.33	2946.1
0111100100100**10101**001010 (Ab_8)	30	18	21	10	3.33	2	16.33	7.78	4169.1
1000111110100**11011**011110 (Ab_9)	35	58	27	30	3.89	6.44	21	23.33	4127.9
0101100110111**10111**010001(Ab_{10})	22	27	55	17	2.44	3	42.78	13.22	3896.2

Şekil 3.9'da akış diyagramı verilen CLONALG algoritmasının a.adımı gereği, her bir antikorun duyarlılığının hesaplanması gerekmektedir. Örneğimiz tek giriş tek çıkışlı bir bulanık sistem olduğu için Ab'lerin duyarlılıklarının hesabında denklem 5.5, 5.6 ve 5.7'yi kullanılacaktır. Öncelikle denklem 5.6'daki *MaksHata* değeri hesaplanır.

$MaksHata = \sum_{p=1}^{v}(y_p - y_{maks})^2$ (verilen örnek için v=4'dir.)

$MaksHata = (y_1 - y_{maks})^2 + (y_2 - y_{maks})^2 + (y_3 - y_{maks})^2 + (y_4 - y_{maks})^2$

$MaksHata = (1 - 49)^2 + (9 - 49)^2 + (25 - 49)^2 + (49 - 49)^2$

$MaksHata = 4480$ olarak bulunur.

Duyarlılık hesabı için sıra, denklem 5.7'de yer alan $y_{CLONALGp}$ değişkenin yani p. eğitim verisi için CLONALG algoritmasından gelen taban uzunluklarına göre hesaplanacak çıkış değerinin bulunmasına gelmiştir. Bu adım aslında bulanık sistemin durulama aşamasına karşılık gelmektedir. Durulma yöntemi olarak istenilen bir yöntem seçilebilir. Bu bölümde durulama işlemi için denklem 5.11'de verilen formül kullanılacaktır.

$$y_p^* = \sum_{i=1}^{ks} \mu(y_i) * y_i / \sum_{i=1}^{ks} \mu(y_i) \quad (5.11)$$

Burada;

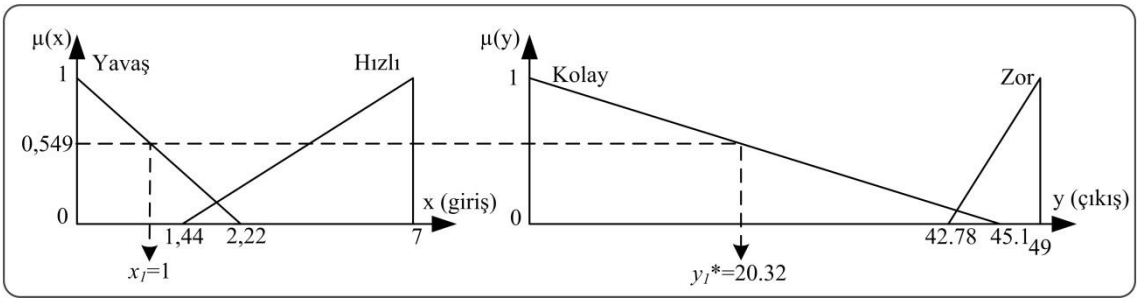
y_p^* = *p.* eğitim verisi için durulanmış çıkış değerini (*p*=1,…,*v*),

ks = *p.* eğitim verisinin tetiklediği kural sayısını,

y_i = *p.* eğitim verisinin tetiklediği *i.* kurala karşılık gelen çıkış değerini (*i*=1,…,*ks*)

$\mu(y_i)$ = y_i'nin üyelik derecesini göstermektedir.

x_1= 1 iken (yani, p=1) popülasyonun ilk bireyi olan Ab_1 antikoru için $y_{CLONALGp}$ değerinin hesaplanması Şekil 5.6'da gösterilmiştir. Şekilden de görüldüğü üzere x_1'in *yavaş* bulanık kümesine olan aitliği $\mu(x_1)$=0.549 olarak bulunur. Hızlı bulanık kümesine ise 0 üyelik derecesi ile aittir. Bu durumda sadece *Kural 1* tetiklenir. Şimdi yapılacak olan *y* değişkenin *kolay* bulanık kümesine 0.549 üyelik derecesi ($\mu(y_1) = 0.549$) ile ait olan y_1 değerinin bulunmasıdır. y_1 = 20.32 değeri Şekil 5.6 kullanılarak bulunur. Daha sonra ilgili değerler denklem 5.11'de yerine konur ve y_1* değeri hesaplanır. x_1= 1 için tetiklenen kural sayısı 1 tane olduğu için y_1* değeri y_1 değerine eşit çıkacaktır. Denklem 5.11 kullanılarak hesaplanan y_p^* değeri, aslında denklem 5.7'deki $y_{CLONALGp}$ değerine karşılık gelir. Yani y_1* değeri olarak bulunan 20.32 değeri $y_{CLONALG1}$ olarak alınacaktır. Aynı hesaplamalar geriye kalan 3 referans nokta içinde yapılır.



Şekil 5.6. Başlangıç popülasyonundaki ilk birey için çözüm

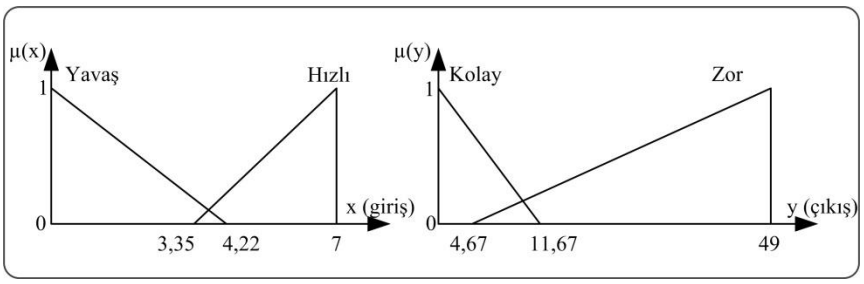
Şimdi sıra y çıkış noktaları için yapılan hataların toplamının bulunmasına gelmiştir. Örneğin p=1 için, gerçekte y_1=1 olması gerekirken, $y_{CLONALG1}$=20.32 olarak bulunmuştu. Buradaki hata $(1\text{-}20.32)^2$ = 373.2624 olarak hesaplanır. Diğer noktalar içinde hatalar hesaplanır ve tüm hatalar toplanır (denklem 5.6). Sonra *ToplamHata MaksHata*'dan çıkartılarak (denklem 5.5) duyarlılık değeri Ab_1 için 2372,3 olarak hesaplanmış olunur. Diğer antikorların duyarlılıkları da aynı şekilde hesaplanmış ve Çizelge 5.1'in 10. sütununda verilmiştir.

Şekil 3.9'da akış diyagramı verilen CLONALG algoritmasına göre en yüksek duyarlılığa sahip ilk *n* birey seçilir ve her bir birey için N_c tane klon üretilerek *C* klon popülasyonu oluşturulur. Daha sonra *C* popülasyonun bireyleri duyarlılıkları ile ters orantılı olarak mutasyona uğratılır. Bu işlemler algoritmanın durdurma şartı sağlanasıya kadar devam eder. Çizelge 5.2'de algoritma sonlandıktan sonraki antikor popülasyonun en son durumu verilmiştir.

Çizelge 5.2. Sonuç Popülasyonu

1	2	3	4	5	6	7	8	9	10
Antikorlar	Taban 1.1	Taban 1.2	Taban 2.1	Taban 2.2	Taban 1.1Y	Taban 1.2Y	Taban 2.1Y	Taban 2.2Y	Duyar-lılık
10010010001000111111001(Ab_1)	36	34	15	57	4	3.78	11.67	44.33	4476.0
10000110001000110111001(Ab_2)	33	34	13	57	3.67	3.78	10.11	44.33	4476.1
10001110001000110111001(Ab_3)	35	34	13	57	3.89	3.78	10.11	44.33	4475.7
10010010001100111011101(Ab_4)	36	35	14	61	4	3.89	10.89	47.44	4476.2
10001110001000111011011(Ab_5)	35	34	14	59	3.89	3.78	10.89	45.89	4476.3
10011010001100111011101(Ab_6)	38	35	14	61	4.22	3.89	10.89	47.44	4475.8
10011110001100111011101(Ab_7)	39	35	14	61	4.33	3.89	10.89	47.44	4475.4
10001010001100111111101(Ab_8)	34	35	15	61	3.77	3.89	11.67	47.44	4475.4
10011010000100111111001(Ab_9)	38	33	15	57	4.22	3.65	11.67	44.33	4476.4
101000100011001111111101(Ab_{10})	40	35	15	61	4.44	3.89	11.67	47.44	4476.0

Çizelge 5.2'deki en yüksek duyarlılığa sahip Ab_9 antikoru problemin optimal çözümü olarak kabul edilir. Ab_9 antikorunun içerdiği taban değerlerine göre bulanık sistemin üyelik fonksiyonların şekli, Şekil 5.7'de gösterilmiştir.



Şekil 5.7. Optimal çözüme ait üyelik fonksiyonlarının şekli

Bölümün devamında, verilen örnek için üyelik fonksiyonlarının CLONALG ile optimize edildiği durum (Şakiroğlu ve Arslan, 2007) ile GA ile optimize edildiği durumun (Arslan ve Kaya, 2001) kıyaslaması ve elde edilen sonuçların yorumları verilecektir.

Deney şu şekilde tasarlanmıştır. 20 farklı grup oluşturulmuştur. Her bir grup 10 farklı popülasyon ve her bir popülasyon 10 adet birey içermektedir. Her bir popülasyondaki en yüksek duyarlılığa sahip antikor çözüm olarak kabul edilmiştir. Algoritmalar gruplardaki her bir popülasyon için 20 kez çalıştırılmış ve sonuçlar grup grup kıyaslanmıştır. Sonuçların kıyaslanmasında, Wilcoxon işaret-sıra istatistiki testi kullanılmıştır. Çünkü iki farklı algoritma aynı popülasyonlar üzerinde çalıştırılmıştır. Popülasyondaki bireylerin dağılımı hakkında yeterince bilgi sahibi olunmadığı için parametrik olmayan bu test seçilmiştir. Eğer bireylerin dağılımının normal olunduğundan emin olunsaydı, parametrik bir test olan eşlenik t-testi kullanılırdı. Wilcoxon işaret-sıra testi kullanılarak % 95 önem seviyesinde doğruluğu sınanacak olan H_0 hipotezi verilen örnek için aşağıda verildiği gibi kurulabilir.

H_0: Verilen *örnek için CLONALG algoritması ve Genetik algoritma arasında %95 önem seviyesinde istatistiksel olarak anlamlı bir fark yoktur.*

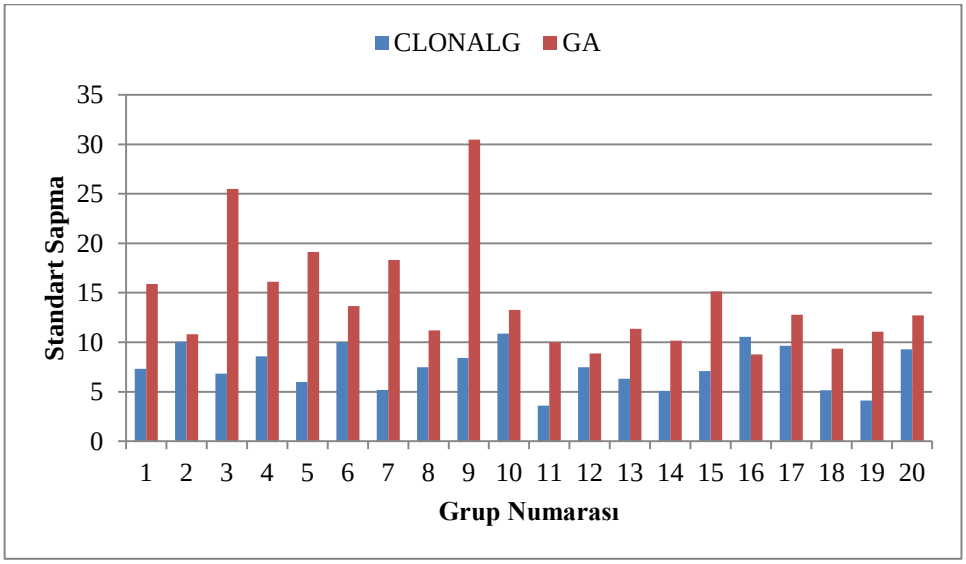
Wilcoxon işaret-sıra testin *p* değerine göre hipotezin kabul edilip edilmeyeceğine karar verilir. Örneğin, eğer *p* değeri 0.05'den büyükse H_0 hipotezi %95 önem seviyesi için kabul edilir aksi takdirde reddedilir. Test 20 grup üzerinde ayrı ayrı çalıştırılmış ve elde edilen p değerleri Çizelge 5.3'de gösterilmiştir. Çizelgeden de görüldüğü üzere H_0 hipotezi sadece 1 grup için kabul edilmiş, diğer tüm gruplar için reddedilmiştir.

Çizelge 5.3. Gruplara göre Wilcoxon işaret-sıra istatistikî testinden elde edilen *p* değerleri

Gr	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	***10***	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	20
p	,009	,001	,02	,02	,001	,02	,003	,005	,02	,08	,001	,001	,04	,003	,01	,02	,001	,02	,001	,04

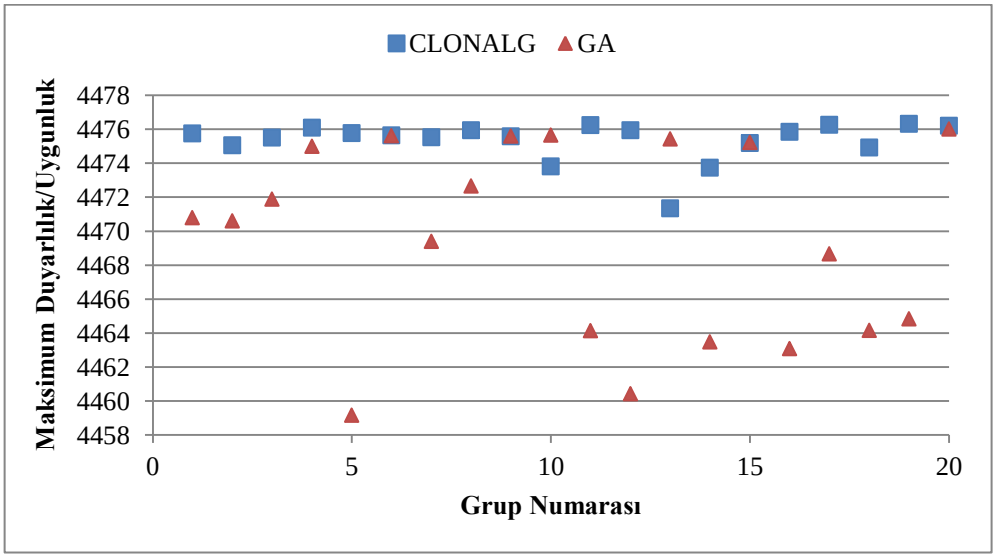
Buradan yola çıkarak, bu örneğin çözümü için iki algoritma arasında %95 önem seviyesinde istatistikî olarak anlamlı bir fark olduğu sonucuna varılabilir. Ancak hangi algoritmanın problemin çözümü için daha iyi olduğu hala bilinmemektedir. Bunun anlaşılması için bazı belirleyici istatistik değerlerin incelenmesi gerekmektedir.

İlk olarak Şekil 5.8'de verilen grupların standart sapma değerlerini inceleyelim.



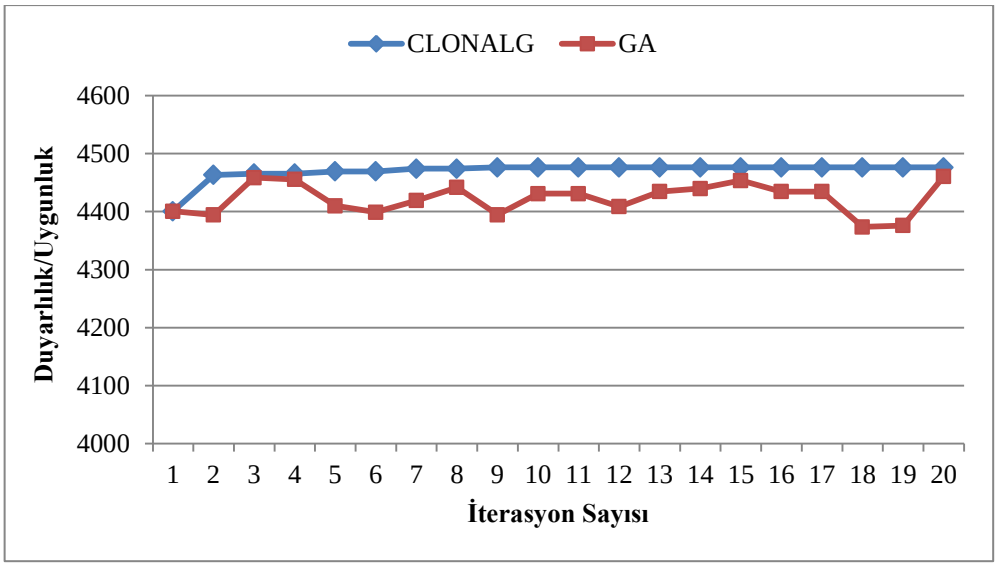
Şekil 5.8. Grupların standart sapma değerleri

Grafikten de görüldüğü üzere, tüm gruplar için CLONALG ile eğitilen grupların standart sapmaları, GA ile eğitilen grupların standart sapmalarından düşüktür. Buradan CLONALG ile daha istikrarlı sonuçlar elde edildiği çıkarımı yapılabilir.



Şekil 5.9. Gruplardaki en büyük uygunluk/duyarlılık değerleri

Şekil 5.9'da her bir grubun en büyük uygunluk/duyarlılık değerleri gösterilmiştir. Grafikten de açıkça görülmektedir ki optimum sonucu bulmakta CLONALG, GA'dan daha başarılıdır. Çünkü GA, sadece 6 grupta en büyük uygunluk derecesine yaklaşmıştır ve bunlardan sadece 2 tanesi CLONALG'den daha iyidir, diğer 4'ü CLONALG ile benzer sonuç vermiştir. Geriye kalan 14 grup için ise CLONALG GA'dan daha iyi sonuç üretmiştir.



Şekil 5.10. Gruplardaki en iyi bireyin iterasyon sayısına göre uygunluk/duyarlılık değerleri

Şekil 5.10'da algoritmanın yakınsaması hakkında fikir sahibi olmak için, CLONALG ve GA gruplarındaki en iyi birer bireyin iterasyon sayısına göre uygunluk/duyarlılık değerleri gösterilmiştir. Grafikten de görüldüğü üzere, CLONALG optimum sonuca GA'dan daha hızlı ulaşmış ve daha istikrarlı bir grafik oluşturmuştur.

Sonuçta, yapılan deneysel çalışmalara dayanarak verilen örnek için üyelik fonksiyonlarının en uygun yerleşiminin bulunmasında CLONALG'nin kullanımı daha avantajlı olduğu söylenebilir.

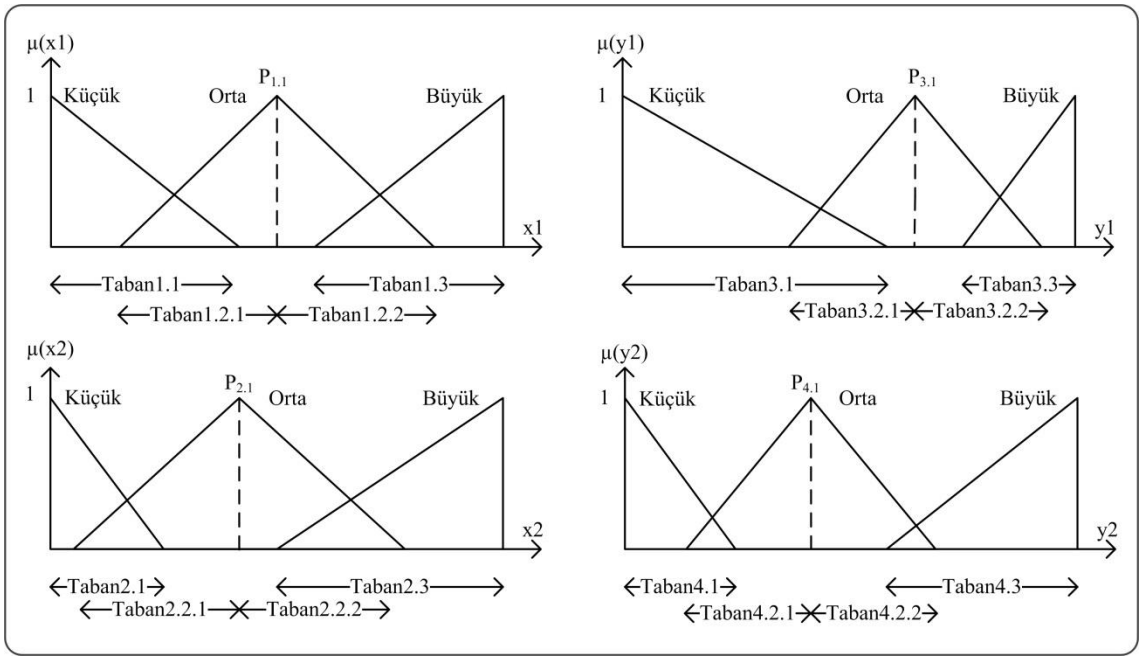
5.3.2. Çok girişli - çıkışlı bir bulanık sistem örneği için ÜF optimizasyonu

Bu bölümde, çok girişli çok çıkışlı bir bulanık mantık sistem örneği üzerinde önerilen yöntemin ile üyelik fonksiyonu optimizasyonunun nasıl yapılacağı anlatılacaktır.

Her biri üç dilsel değişkene sahip olan iki girişli/çıkışlı bir bulanık sistem düşünelim. *x1(giriş1), x2(giriş2), y1(çıkış1) ve y2 (çıkış2)* olarak isimlendirilen değişkenler *küçük, orta* ve *büyük* dilsel değerlerini kullansın. Her bir dilsel değeri temsil etmek için kullanılan üyelik fonksiyonu şekli ise üçgen olsun. Örnek bulanık sisteme ait kuralların aşağıdaki gibi verildiği kabul edilsin.

Kural 1: Eğer *x1* küçük ve *x2* büyük ise *y1* orta ve *y2* ortadır.
Kural 2: Eğer *x1* orta ve *x2* küçük ise *y1* küçük ve *y2* küçüktür.
Kural 3: Eğer *x1* orta ve *x2* orta ise *y1* orta ve *y2* ortadır.
Kural 4: Eğer *x1* orta ve *x2* büyük ise *y1* büyük ve *y2* ortadır.
Kural 5: Eğer *x1* büyük ise *y1* büyük ve *y2* büyüktür.

Sırasıyla değerlerinin [0-7], [0-80], [0-50] ve [0-125] aralığında değiştiği kabul edilen *x1(giriş1), x2(giriş2), y1(çıkış1) ve y2 (çıkış2)* değişkenlerinin ait olduğu bulanık sistemin üyelik fonksiyonların görüntüsü Şekil 5.11'de gösterilene benzer olacaktır.



Şekil 5.11. İki giriş/çıkışlı örnek bir bulanık sistem için üyelik fonksiyonlarının gösterimi

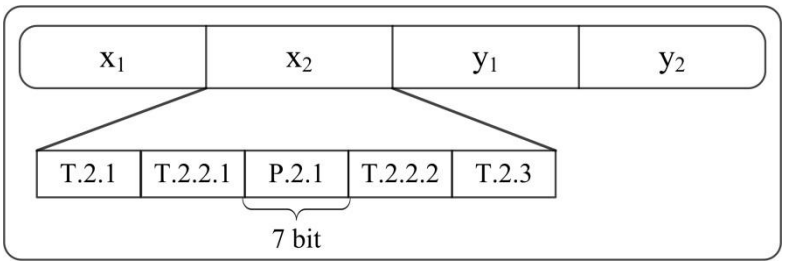
Sistemin eğitilebilmesi için önceden bazı referans giriş/çıkış değerlerinin ölçülmesi gerekmektedir. Bu ölçülen değerlerin de aşağıdaki gibi olduğu varsayılsın.

Ref. Değer 1: Eğer *x1*=1 ve *x2*=75 ise *y1*=76 ve *y2*=19.
Ref. Değer 2: Eğer *x1*=3 ve *x2*=20 ise *y1*=29 ve *y2*=16.
Ref. Değer 3: Eğer *x1*=4 ve *x2*=50 ise *y1*=66 ve *y2*=26.
Ref. Değer 4: Eğer *x1*=5 ve *x2*= 70 ise *y1*=95 ve *y2*=34.
Ref. Değer 5: Eğer *x1*=7 ve *x2*= 60 ise *y1*=109 ve *y2*=40.

Önerilen yöntemde, CLONALG algoritmasından beklenen Şekil 5.12'deki üçgenlere ait en uygun taban uzunluklarının ve kesişim noktalarının en uygun değerlerini verilen kurallara ve referans giriş/çıkış değerlerine dayanarak bulmasıdır. Öncelikle bir önceki örnekte olduğu gibi her bir taban değerinin kodlanmasında kaç bit kullanılacağı denklem 5.1 kullanılarak hesaplanması gerekmektedir.

$$m = YukarıYuvarla\{maks(\text{maks}_{1\le i\le n}(\log_2 X^i_{maks}), \text{maks}_{1\le j\le u}(\log_2 Y^j_{maks}))\}$$
$$m = YukarıYuvarla\{maks(maks(\log_2 7, \log_2 80), maks(\log_2 50, \log_2 125))\}$$
$$m = YukarıYuvarla\{maks(maks(2.8074, 6.3219), maks(5.6439, 6.9658))\}$$
$$m = YukarıYuvarla\{6.9658\}$$
$$m = 7 \text{ bit.}$$

Böylece, her bir taban uzunluğunun veya kesişim noktasının ikili olarak kodlanması için gerekli olan bit sayısı 7 olarak bulunmuş olur. Bir önceki örnekten farklı olarak, bu örnekte her bir değişken için 3 dilsel değer kullanıldığından dolayı çeşitkenar üçgenlerin kesişim noktalarının da bulunması gerekmektedir. Bundan dolayı antikorda taban uzunluklarının yanı sıra kesişim noktaları da yer almaktadır. Burada kullanılan için antikorun yapısı Şekil 5.12'de gösterildiği gibidir. Bu antikorda yer alan x_2 değişkenine ait olan kısmın açılımı da şekilde gösterilmiştir. Diğer değişkenlerinki de benzer yapıdadır ve her biri 7 bit olarak ikili şekilde kodlanmıştır.



Şekil 5.12. Antikorun yapısı

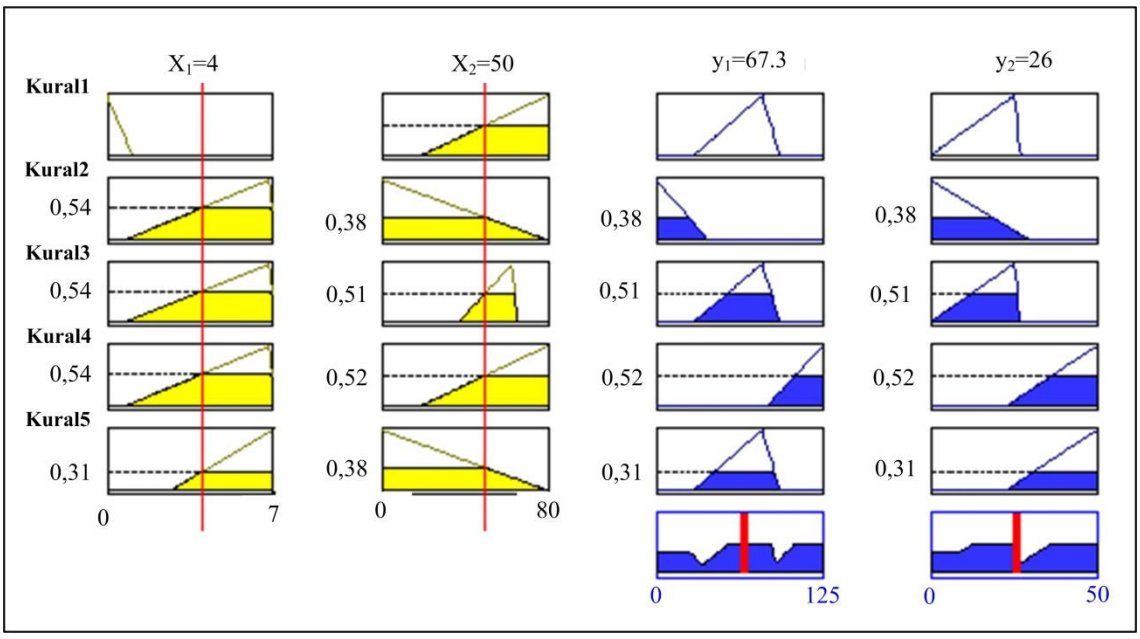
Şekil 3.9'da akış diyagramı verilen CLONALG algoritmasının a.adımı gereği, her bir antikorun duyarlılığının hesaplanması gerekmektedir. Örneğimiz çok giriş çok çıkışlı bir bulanık sistem olduğu için Ab'lerin duyarlılıklarının hesabında denklem 5.8, 5.9 ve 5.10'nu kullanılacaktır. Öncelikle denklem 5.9'daki $MaksHata_{çoklu}$ değeri Çizelge 5.4.'deki gibi hesaplanır.

Çizelge 5.4. $MaksHata_{çoklu}$'nun hesaplama adımları

p	$\sum_{r=1}^{u} maks[\left(y_{r_{CLONALGp}} - y_{r_{min}}\right), (y_{r_{maks}} - y_{r_{CLONALGp}})]^2$	Hata
1	maks[(19-0);(50-19)]^2+ maks[(76-0);(125-76)]^2 =31^2+76^2	6737
2	maks[(16-0);(50-16)]^2+maks[(29-0);(125-29)]^2 =34^2+96^2	10372
3	maks[(26-0);(50-26)]^2+maks[(66-0);(125-66)]^2 =26^2+66^2	5032
4	maks[(34-0);(50-36)]^2+ maks[(95-0);(125-95)]^2 =34^2+95^2	10181
5	maks[(40-0);(50-40)]^2+maks[(109-0);(125-109)]^2 =40^2+109^2	13481
$MaksHata_{çoklu} = \sum_{p=1}^{v}\sum_{r=1}^{u} maks[\left(y_{r_{CLONALGp}} - y_{r_{min}}\right), (y_{r_{maks}} - y_{r_{CLONALGp}})]^2$		45803

Sıra denklem 5.10'da verilen $ToplamHata_{çoklu}$ değerini hesaplamaya gelmiştir. Bunun için öncelikle her bir referans noktası p için $y_{r_{CLONALGp}}$değerinin hesaplanması gerekmektedir. Bu işlemin aslında, bir bulanık sistemdeki durulama adımına karşılık geldiği daha önce belirtilmişti. Yapılması gereken, sistemin referans giriş noktalarını durulaştırma adımına vermek ve elde edilen $y_{r_{CLONALGp}}$ durulanmış değerini, gerçek referans çıkış noktasından çıkararak hatayı hesaplamaktır. CLONALG tarafından bulunan taban değerleri kullanılarak çıkış değerlerinin nasıl hesaplanacağını bir örnek üzerinde açıklayalım. Bu örnek için denklem 4.16'da verilen ağırlık merkezi durulama yöntemi kullanılsın.

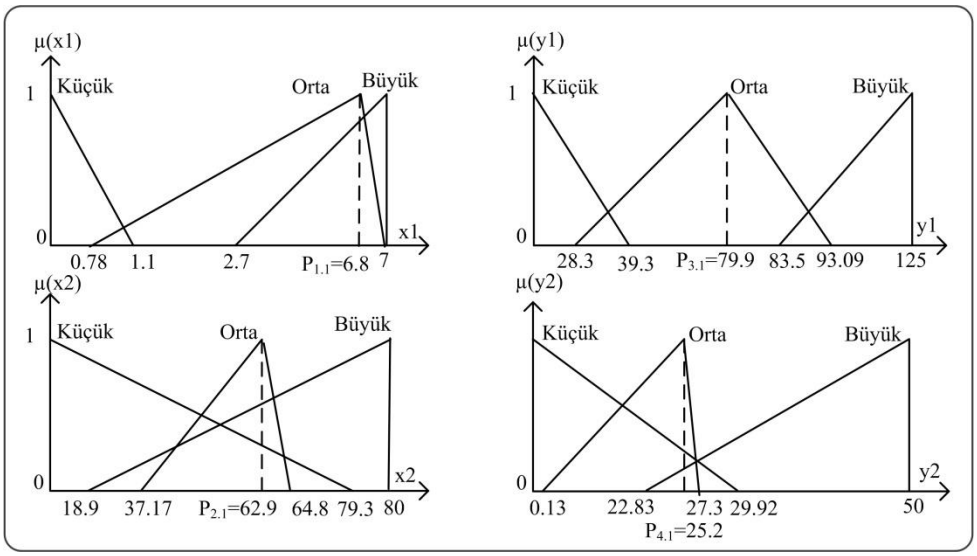
Örneğin x_1=4 ve x_2=50 olarak alınsın. Öncelikle bu örneklerin her ikisi için de ayrı ayrı hangi üyelik fonksiyonlarını kestikleri bulunur. Daha sonra bu üyelik fonksiyonlarından ikisinin de aynı kural içinde yer alıp almadığı araştırılır. Eğer birlikte bulundukları kurallar var ise, girişlerin üyelik fonksiyonlarını kestikleri değer hesaplanır. Kuralın öncülleri *VE* bağlacı ile birbirlerine bağlanmışsa, çıkışlar üyelik derecelerinden düşük olanlar kullanılarak bulunur. Eğer verilen girişler için, birden fazla üyelik fonksiyonu kesişiyorsa, çıkışı bulmak için ağırlık merkezi durulama yöntemi kullanılır. Şekil 5.13'de bu örnek gösterilmiştir.



Şekil 5.13. x_1=4 ve x_2=50 iken verilen çok girişli/çıkışlı bulanık sistem örneği için çıkışların hesaplanması

Şekilden de görüldüğü üzere, x_1 orta ve büyük dilsel etikete sahip üyelik fonksiyonlarını keserken, x_2 küçük, orta ve büyük üyelik fonksiyonlarını kesmiştir. Öncelikle her bir üyelik fonksiyonunun hangi derecede kesildiğinin tespit edilmesi gerekir. Bu değerlerin hepsi Şekil 5.13'de verilmiştir. Daha sonra hangi kuralların tetiklendiği bulunur. Eğer giriş değerlerinin kestiği her iki üyelik fonksiyonu da aynı kural içinde yer alıyorsa, bu kural tetiklenmiştir. Şekil 5.13'deki örneğimizde sırasıyla kural 2, 3, 4 ve 5 tetiklenmektedir. Tetiklenen kurallar bulunduktan sonra bu kuralların uygunluk dereceleri hesaplanır. $\mu_{orta}^{x1,2} = 0.54$ ve $\mu_{\text{küçük}}^{x2,2}$ = 0.38 ve iki kuralın arasındaki bağlaç *VE* olduğu için 2. kuralın uygunluk derecesi $\mu_2 = min(\mu_{orta}^{x1,2}, \mu_{\text{küçük}}^{x2,2})$ formülünden 0.38 olarak hesaplanır. Diğer kuralların uygunluk derecelerinde benzer şekilde teker teker hesaplanır ve μ_3 = 0.51, μ_4=0.52 ve μ_5 = 0.31 olarak bulunur. Bu değerler sırasıyla çıkış üyelik fonksiyonlarında karşılık gelen bulanık kümelerde işaretlenir. Daha sonra ağırlık merkezi durulama yöntemi kullanılarak çıkışlar $y_{1_{CLONALG_3}}$=67,3 ve $y_{2_{CLONALG_3}}$=26 olarak hesaplanır. Ancak verilen referans noktalara göre y_{1_3}=66 ve $y_{2_3} = 26$ olması gerekiyordu. Bu durumda $\sum_{r=1}^{2}\left(y_{r_{CLONALGp}} - y_{r_p}\right)^2$ formülü kullanılarak 3. referans nokta için hata 1.69 olarak hesaplanır. Diğer referans noktalar içinde aynı işlem tekrar edilir. Son olarak tüm hatalar toplanır ve $MaksHata_{çoklu}$ değerinden çıkarılarak bireyin duyarlılığı hesaplanır.

500 iterasyon sonunda algoritma tamamlanır ve en yüksek duyarlılığa sahip antikor çözüm olarak kabul edilir. Bu örnek için duyarlığı 45797 olan Ab en uygun çözüm olarak alınmış ve ilgili antikorda kodlanmış taban değerlerine göre Şekil 5.14'de gösterilen bulanık sistem oluşturulmuştur.



Şekil 5.14. Çok girişli/çıkışlı bulanık sistem örneği için en uygun çözüm

Bölümün devamında, verilen örnek için üyelik fonksiyonlarının CLONALG ile optimize edildiği durum ile GA ile optimize edildiği durumun kıyaslaması ve elde edilen sonuçların yorumları verilecektir.

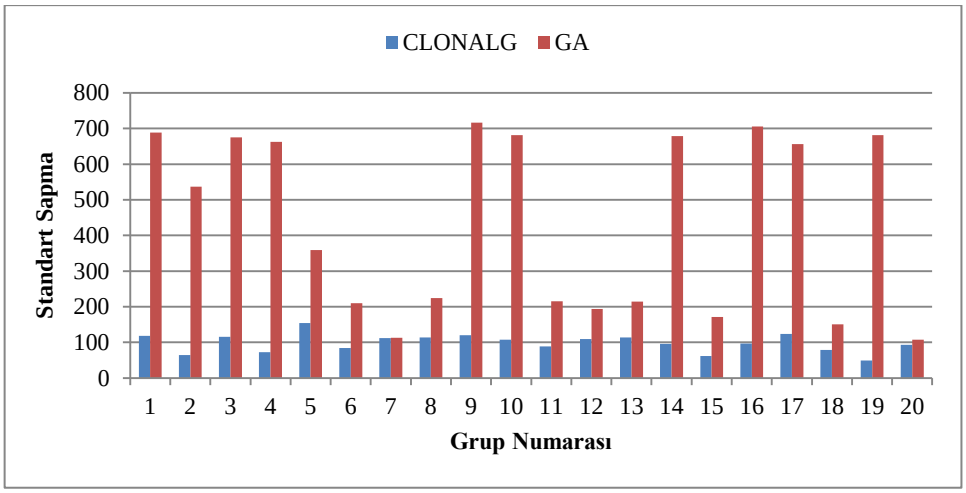
Deney şu şekilde tasarlanmıştır. 20 farklı grup oluşturulmuştur. Her bir grup 10 farklı popülasyon ve her bir popülasyon 10 adet birey içermektedir. Her bir popülasyondaki en yüksek duyarlılığa sahip antikor çözüm olarak kabul edilmiştir. Algoritmalar gruplardaki her bir popülasyon için 500 kez çalıştırılmış ve sonuçlar grup grup kıyaslanmıştır. Sonuçların kıyaslanması, Wilcoxon işaret-sıra istatistiki testi kullanılarak % 95 önem seviyesinde yapılmıştır. Doğruluğu sınanacak olan H_0 hipotezi verilen örnek için aşağıdaki gibi kurulabilir.

H_0: Verilen *örnek için CLONALG algoritması ve Genetik algoritma arasında %95 önem seviyesinde istatistiksel olarak anlamlı bir fark yoktur.*

Çizelge 5.5. Gruplara göre Wilcoxon işaret-sıra istatistikî testinden elde edilen *p* değerleri

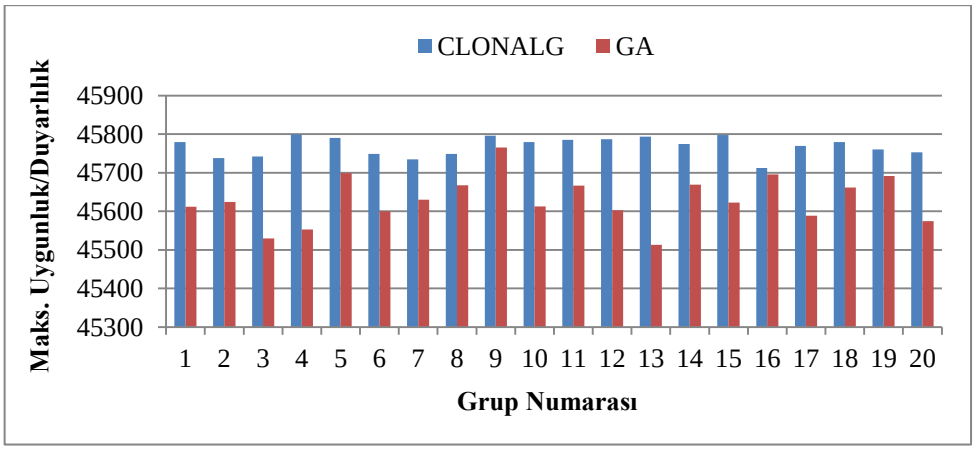
Gr	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	20
p	.013	.003	.001	.001	.003	.001	.013	.005	.019	.009	.037	.003	.001	.001	.003	.013	.009	.013	.001	.001

Wilcoxon işaret-sıra testinin *p* değerine göre hipotezin kabul edilip edilmeyeceğine karar verilir. Test 20 grup üzerinde ayrı ayrı çalıştırılmış ve elde edilen p değerleri Çizelge 5.3'de gösterilmiştir. Tüm *p* değerleri 0.05'den küçük olduğu için H_0 hipotezi tüm gruplar için reddedilmiştir. Yani bu örneğin çözümü için iki algoritma arasında %95 önem seviyesinde istatistikî olarak anlamlı bir fark vardır. Ancak hangi algoritmanın problemin çözümü için daha iyi olduğu hala bilinmemektedir. Bunun anlaşılması için öncelikle grupların standart sapma değerleri incelenmiştir.



Şekil 5.15. Grupların standart sapma değerleri

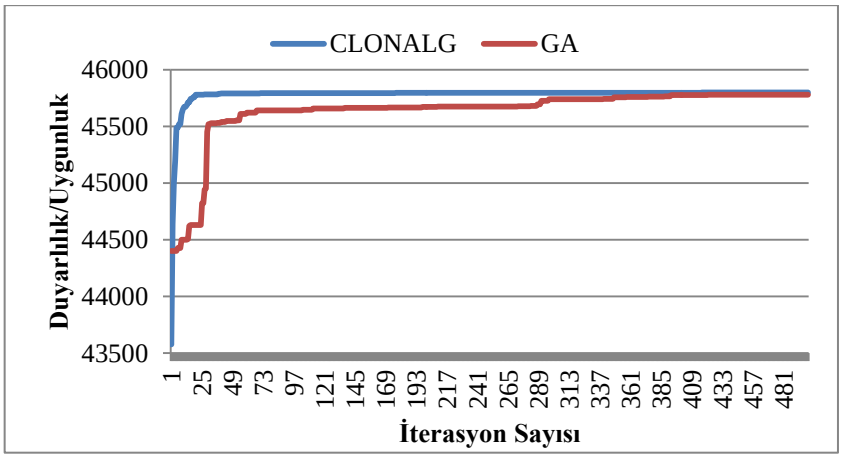
Şekil 5.15'de grupların standart sapma değerleri görülmektedir. Bu örnek içinde, bir önceki örnekte olduğu gibi tüm gruplar için CLONALG ile eğitilen grupların standart sapmaları, GA ile eğitilen grupların standart sapmalarından düşüktür. Yani popülasyondaki bireylerin dağılımı daha istikrarlıdır.



Şekil 5.16. Gruplardaki en büyük uygunluk/duyarlılık değerleri

Şekil 5.16'da her bir gruptaki en yüksek uygunluk veya duyarlılık değerlerini yani grupların çözümlerini gösteren grafik bulunmaktadır. Tüm gruplarda CLONALG daha başarılı sonuçlar elde etmiş ve en uygun sonuca yakınsamada daha başarılı olmuştur.

Şekil 5.17'de her bir grubun en iyi bireyinin iterasyon sayısına göre uygunluk/duyarlılık değerlerinin değişimleri gösterilmiştir. Grafik incelenerek algoritmaların sonuca yakınsamaları hakkında fikir edinmek mümkündür. Grafikten de görüldüğü üzere, CLONALG optimum sonuca GA'dan daha hızlı ulaşmış ve daha istikrarlı bir grafik oluşturmuştur.



Şekil5.17. Gruplardaki en iyi bireyin iterasyon sayısına göre uygunluk/duyarlılık değerleri

Sonuç olarak, çok girişli çok çıkışlı bir bulanık sistem örneği için CLONALG'nin üyelik fonksiyonlarının en uygun yerleşimini bulma işleminde daha başarılı bir performans gösterdiği deneysel çalışmalara dayanarak söylenebilir.

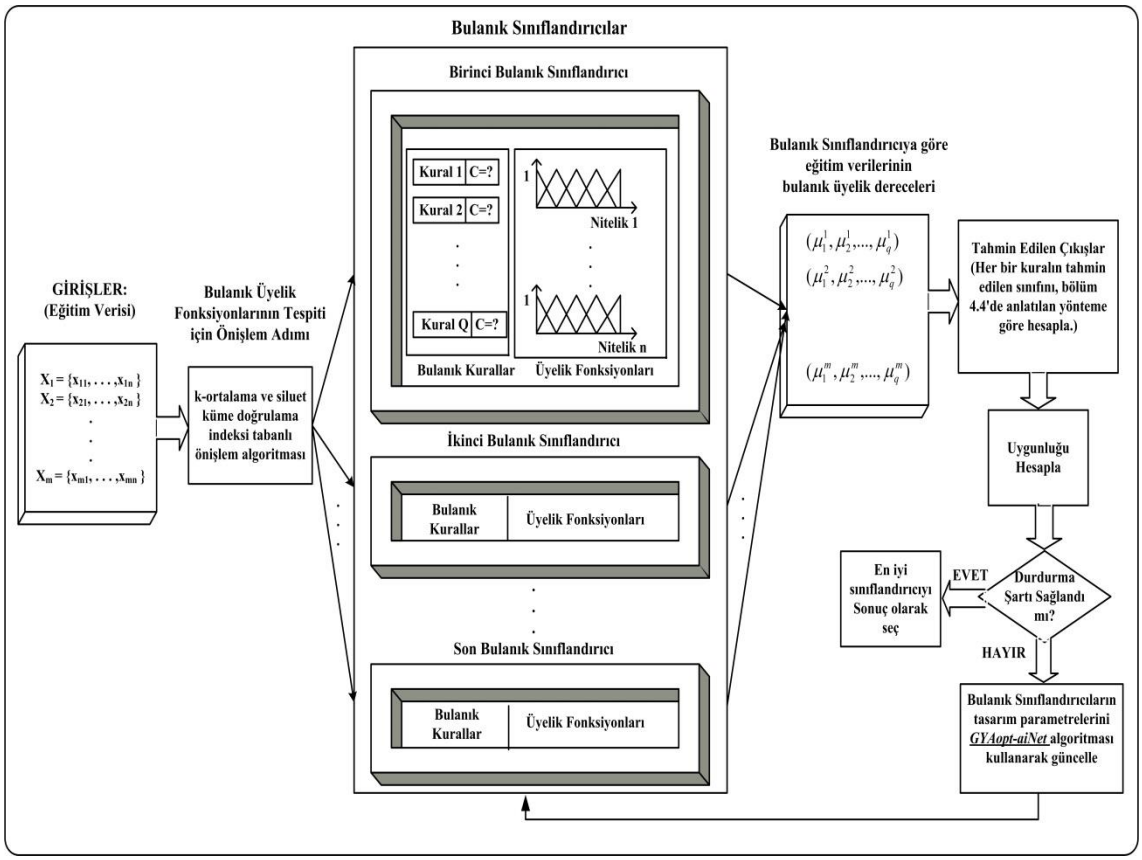
6. BULANIK SINIFLANDIRICI TASARIMI İÇİN ÖNERİLEN SDKo ve GYAopt-aiNet ALGORİTMALARI

Bir önceki bölümde, üyelik fonksiyonlarına ait parametrelerin en uygun değerlerinin belirlenmesi bir optimizasyon problemi olarak ele alınmış ve çözüm olarak bir yapay bağışıklık algoritması olan CLONALG'nin kullanılması önerilmiştir. Ancak bulanık mantığın çekirdeğindeki *derecelendirme* kavramına karşılık gelen bu probleminin çözümü için sisteme ait kural tabanının ve her bir niteliğe ait üyelik fonksiyonları sayısının önceden bilinmesi gerekmektedir. Bu bilgilerin temini esnasında bir uzmana ihtiyaç duyulması sunulan yöntemin dezavantajıdır. Bu çalışmanın amacı bulanık sistem tasarlanırken uzmana olan bağımlılığı en aza indirmek olduğundan, tezin bu bölümünde öncelikle her bir niteliğin kaç üyelik fonksiyonuna sahip olması gerektiğini ve ÜF'larının başlangıç yerleşim yerlerini bulan **S**iluet küme **D**oğrulama indeksi ve **k**-**o**rtalama tabanlı (SDko) yeni bir ön işlem algoritması önerilmiştir. Bu önişlem adımı sayesinde *taneciklendirme* (bulanık mantığın çekirdeğinde bulunan diğer kavram) işlemi için gerekli olan ön bilgiler uzman yardımı olmaksızın eğitim verisi kullanılarak elde edilir. Bölümün devamında, SDko önişlem adımı ile tespit edilen bulanık tanecikleri derecelendirmesinin yanı sıra dinamik şekilde kural tabanını da optimize edebilen GYAopt-aiNet algoritması tanıtılmıştır. GYAopt-aiNet (**G**eliştirilmiş ve memetic algoritmanın **Y**erel **A**rama stratejisiyle birleştirilmiş **opt-aiNet**), içerisinde çeşitlilik (diversity), hafıza ve baskılama (suppression) gibi mekanizmaları barındıran YBS tabanlı opt-aiNet algoritmasının bu tez çalışması esnasında geliştirilmiş olan versiyonuna, Memetik algoritmanın yerel arama mekanizmasının eklenmesi ile elde edilmiştir. Böylece hiper mutasyon ve baskılama mekanizmaları değiştirilerek elde edilen geliştirilmiş opt-aiNet algoritmasının genel arama yeteneği, Memetik algoritmanın yerel arama stratejisi ile tamamlanmış ve bu mekanizmalar sayesinde optimum sonuca daha doğru ve hızlı şekilde ulaşılması amaçlanmıştır.

Sonuç olarak, SDko önişlem ve GYAopt-aiNet algoritmalarının birlikte kullanımı ile bulanık taneciklerin derecelendirilmesi ve bulanık kural tabanının öğrenilmesi sağlanmıştır. Bu yapı sayesinde, listesi Çizelge 4.3'de verilen bulanık bir sisteme ait parametrelerden; yapısal bir parametre olan üyelik fonksiyonlarının sayısı, bağıntısal ve işlemsel parametrelerin değerleri bir uzmana ihtiyaç duyulmaksızın eğitim kümesinden öğrenilebilmektedir. Bir önceki bölümde sunulan yöntemde sadece

işlemsel parametre öğrenilmekte, diğer parametrelerin tümünün bir uzman tarafından belirlemesi gerekmekteydi.

Bu tez çalışmasında, önerilen yöntem özel bir bulanık mantık sistem türü olan bulanık sınıflandırıcı tasarımına tatbik edilmiş ve verimliliği araştırılmıştır. Bulanık sınıflandırıcı tasarımı için sunulan genel yapı Şekil 6.1'de gösterilmiştir.



Şekil 6.1. Bulanık sınıflandırıcı tasarımı için sunulan genel yapı

Şekil 6.1'de görüldüğü üzere, öncelikle eğitim verilerine önerilen SDko önişlem algoritması uygulanır. Elde edilen sonuçlar kullanılarak başlangıç bulanık sınıflandırıcılar oluşturulur. Her bir sınıflandırıcı popülasyondaki bir bireyi temsil eder. Bu çalışmada, sınıflandırıcıların çıkışları Ishibuchi ve ark. (1997) tarafından önerilen ve Bölüm 4.4'de anlatılan yönteme göre hesaplanıp her bir bireyin uygunluğu sınıflandırma doğruluğu kullanılarak bulunur. Eğer uygunluk değeri istenilen başarıya ulaştıysa veya belli bir iterasyon sayısı tamamlanmışsa durdurma şartı sağlanmış olur ve işlem sonlandırılır. Aksi takdirde GYAopt-aiNet algoritması kullanılarak, sistemin bulanık kural tabanı ve üyelik fonksiyonlarını belirleyen parametreleri güncellenir.

Bölümün devamında önerilen önişlem ve GYAopt-aiNet algoritmaları ayrıntılı şekilde açıklanacaktır.

6.1. Önerilen Önişlem Algoritması: SDko

Bu bölümde, değişkenlere ait üyelik fonksiyonlarının daha iyi optimize edilebilmesi için her bir değişkenin sahip olacağı üyelik fonksiyonu sayısını ve her bir üyelik fonksiyonunun başlangıç taban aralıkları ile tepe noktalarını bulması için önerilen SDko isimli önişlem algoritması açıklanacaktır. Bu önişlem algoritması siluet küme doğrulama indeksini ve k-ortalama algoritmasını kullanır ve aşağıda listelenen avantajları programa kazandırmış olur.

- Derecelendirme ve taneciklendirme işlemleri için gerekli ön bilgilerin eğitim verilerinden elde edilmesini sağlar. Uzmana olan bağımlılığı azaltır.
- Her bir değişkenin kaç üyelik fonksiyonuna ayrılacağını hesaplar.
- Her bir üyelik fonksiyonunun başlangıç taban aralıkları ile tepe noktalarını belirleyerek arama uzayının daraltılmasını ve doğru sonuca daha çabuk ulaşılmasını sağlar.
- Her bir değişken için tam, ayrıklaştırılabilir, tamamlayıcılığı olan ve makul sayıda üyelik fonksiyonu elde edilmesini mümkün kılar, sistemin yorumlanabilirliğini artırır.

SDko önişlem algoritması iki kısımdan oluşmaktadır. İlk kısmında sistemde yer alan her bir değişkenin sahip olacağı üyelik fonksiyonu sayısı diğer bir deyişle bulanık taneciklerin sayısı tespit edilirken, ikinci kısmında üyelik fonksiyonlarının başlangıç taban aralıklarıyla tepe noktaları bulunmaktadır.

Algoritmanın ilk kısmına göre, eğitim kümesindeki her bir nitelik, değeri 2 ile 7 arasında değişen farklı k değerlerine göre k-ortalama algoritması kullanılarak kümelenir ve her bir k değeri için SD ortalama değeri ve noktaların küme merkezine olan uzaklıklarının toplamı hesaplanır. Bu kümeleme işlemi her bir k değeri için 5 kez tekrar ettirilir. Her bir tekrar için küme merkezleri rastgele seçilir. Noktaların küme merkezine olan uzaklıklarının toplamı en küçük olan tekrar grupları tespit edilir ve bu grubu ait olan SD değeri ile k değerleri saklanır. Saklanan bu SD değerleri arasında en yüksek

değere sahip olan tekrar grubun k küme sayısı, niteliğe ait en uygun üyelik fonksiyonu sayısı olarak kabul edilir. Çizelge 6.1'de önerilen önişlem algoritmasının ilk kısmının sözde kodu verilmiştir.

Çizelge 6.1. SDko-I algoritmasının sözde kodu

Algoritma 1: SDko – I

Giriş: Eğitim Kümesi
Çıkış: Her bir niteliğe ait en uygun bulanık tanecik (ÜF'nu) sayısı - *I*

```
begin
  for eğitim kümesindeki her bir nitelik için do
    k=2;
    while k<7,
      for tekrarSayısı=1:5 do
        k değeri için k-ortalama algoritması ile niteliğe ait verileri kümele;
        k değeri için ortalama SD değerini hesapla;
        k değeri için noktaların küme merkezine olan uzaklıklarınının toplamını hesapla;
      end for;
      Küme merkezine olan uzaklıklarının toplamı en küçük olan tekrarSayısı'nı bul;
      Bu tekrar grubuna ait SD değerini bir dizi değişkenini olan SiluetDizi(k-1)'de sakla;
      k = k +1
    end while;
    SiluetDizi değişkenin en büyük elemanını bul ve elemanın index değerini I değişkeninde sakla;
    I=I+1;
    I değerini ilgli nitelik için en uygun tanecik sayısı olarak kaydet;
  end for;
end begin
```

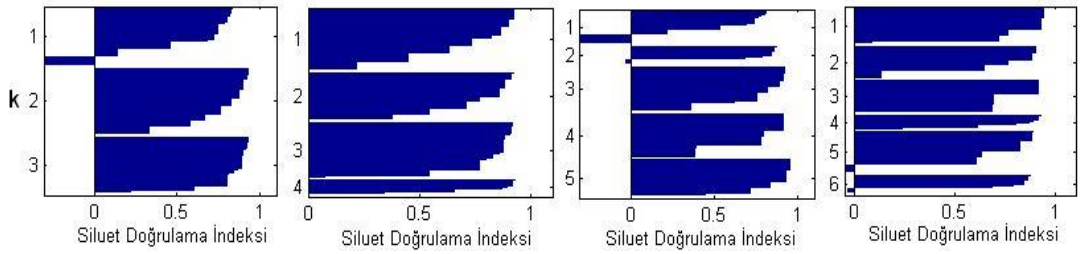
Algoritmanın çalışma mantığını açıklamak için örnek olarak iris veri kümesinin *sepal uzunluk* niteliğini kullanalım. Çizelge 6.2'de bu niteliğin çeşitli k değerlerine göre kümelendiğinde elde edilen SD değerlerinin ortalamaları ve noktaların küme merkezlerine olan uzaklıklarının toplamları verilmiştir. Bu işlem 5 kez tekrarlanmıştır.

Çizelge 6.2. *k* değerlerine göre tekrar gruplarının *SD* değerlerinin ortalamaları ve kümelerdeki noktaların merkezlerine uzaklıklarının toplamları

***k* değeri**		**Tekrar 1**	**Tekrar 2**	**Tekrar 3**	**Tekrar 4**	**Tekrar 5**
3	Toplam Uz.	15.6616	15.6616	15.6362	15.6616	*15.2699*
	SD ort.	0.678	0.678	0.7003	0.678	0.6867
4	Toplam Uz.	8.535	10.6283	7.7595	*7.665*	10.6283
	SD ort.	0.6917	0.6332	0.7401	**0.7414**	0.6332
5	Toplam Uz.	*5.3062*	5.3062	6.9487	5.7084	6.7218
	SD ort.	0.7381	0.7381	0.6359	0.6761	0.6805
6	Toplam Uz.	4.3961	3.493	4.4662	*3.401*	4.6513
	SD ort.	0.7047	0.7278	0.7401	0.7237	0.6531

Çizelge 6.2'de görüldüğü üzere öncelikle her bir *k* değerinin tekrar grupları arasında, noktaların küme merkezlerine olan uzunlukları toplamı en düşük olan grup tespit edilir. Bunlar sırası ile k=3 için 5. tekrar grubu, k=4 için 4. tekrar grubu, k=5 için 1. tekrar grubu ve k=6 için 4. tekrar grubudur Daha sonra tespit edilen bu gruplamaların SD ortalama değerleri incelenir. En yüksek SD ortalama değerinin k=4 için 4. tekrar grubunda olduğu görülmektedir. Bu durumda sepal uzunluk niteliği için en uygun üyelik fonksiyonu sayısı k=4 olarak alınır.

Şekil 6.2'de 4. tekrar grubunun k=3,4,5 ve 6 değerlerine göre bireylere ait siluet doğrulama indeks değerlerini gösteren siluet grafikleri verilmiştir. Küme sayısı *k*=4 olduğu durumda en iyi siluet değerlerinin elde edildiği şekilden de görülmektedir.



Şekil 6.2. k=3,4,5 ve 6 değerlerine göre sepal uzunluk niteliği kümelendiğinde elde edilen siluet grafikleri

Önerilen önişlem algoritmasının ikinci kısmında üyelik fonksiyonlarının başlangıç taban aralıkları ve tepe noktaları bulunur. Buradaki amaç, çeşitli kısıtlar kullanarak bir niteliğe ait üyelik fonksiyonları arasında boşlukların oluşmasını engellemek ve tam, ayrıklaştırılabilir, tamamlayıcılığı olan üyelik fonksiyonları tanımlamaktır. Böylece yorumlanabilirliği yüksek bulanık sistemler oluşturulabilmektedir.

Üyelik fonksiyonlarına ait başlangıç taban aralıkları ve tepe noktaları belirlenirken, önişlem algoritmasının ilk adımında elde edilen kümelere ait belirleyici bilgiler kullanılır. Örneğin her bir kümenin merkezi, o üyelik fonksiyonunun tepe noktası olarak atanabilir. Ancak her bir birey o kümeye eşit şekilde dâhil değildir ve hepsinin kendisine ait bir SD değeri vardır. Bundan dolayı her durumda küme merkezini tepe noktası olarak almak uygun olmayabilir. Bunun bir örneği Şekil 6.2'de görülmektedir. Çözüm olarak Çizelge 6.3'de sözde kodu verilen ve üyelik fonksiyonlarının tepe noktaları ile başlangıç sınır değerlerinin tespitinde kullanılabilecek SDko-II önişlem adımı önerilmiştir.

Çizelge 6.3. SDko-II önişlem algoritmasının sözde kodu

Algoritma 2: SDko – II

Giriş: Küme merkezleri ve kümeye ait olan noktaların SD değerleri
Çıkış: Üyelik fonksiyonlarının tepe noktaları ve başlangıç taban aralıkları

Begin
for eğitim kümesindeki her bir nitelik için **do,**
for niteliğin her bir üyelik fonksiyonu için **do**,
$Ort_{Yüksek_{SD}}$ ve $Ort_{Düşük_{SD}}$değerlerini hesapla;
if niteliğin $Ort_{Yüksek_{SD}}$ ve $Ort_{Düşük_{SD}}$değerleri 0.7'den büyükse,
Kümenin merkezini, üçgen üyelik fonksiyonunun tepe değeri olarak ata;
else
Tepe noktasının kayma yönünü $Ort_{Yüksek_{SD}}$ ve $Ort_{Düşük_{SD}}$değerlerine göre tespit et;
Kayma miktarını hesapla;
Tepe noktasının yerini, kayma yönü ve miktarına göre güncelle;
end if;
end for;
Niteliğin üyelik fonksiyonlarının başlangıç taban aralıklarını tespit et;
end for;
end begin

SDko-II önişlem adımında geçen tanımlamalar ve işlemler aşağıda sırası ile açıklanmıştır:

Tanım 6.1. Düşük$_{SD}$ ve $Ort_{Düşük_{SD}}$: Küme merkezi olarak atanan noktadan küçük olan noktaların SD değerlerinin oluşturduğu kümedir. *Düşük$_{SD}$* kümesinin elemanlarını tespit etmek için kullanılan eşitlik denklem 6.1'de ve elemanların ortalaması bulan eşitlik denklem 6.2'de verilmiştir.

$$Düşük_{SD}(a) = \{x \in ÜF_i | SD(x) < SD(m_i)\}, i = 1, \dots, k \quad (6.1)$$

$$Ort_{Düşük_{SD}} = \frac{\sum_{a=1}^{nd} Düşük_{SD}(a)}{nd} \quad (6.2)$$

Tanım 6.2. Yüksek$_{SD}$ ve $Ort_{Yüksek_{SD}}$: Küme merkezi olarak atanan noktadan büyük olan noktaların SD değerlerinin oluşturduğu kümedir. Bu kümenin elemanlarının tespiti için denklem 6.3'de verilen eşitlik kullanılır. Küme elemanlarının ortalamasını veren eşitlik ise denklem 6.4'de verilmiştir.

$$Yüksek_{SD}(b) = \{x \in ÜF_i | SD(x) > SD(m_i)\}, i = 1, \dots, k \quad (6.3)$$

$$Ort_{Yüksek_{SD}} = \frac{\sum_{b=1}^{ny} Yüksek_{SD}(b)}{ny} \quad (6.4)$$

Burada;

k: Sınıflandırılacak niteliğin bölüneceği küme sayısını yani ÜF sayısını,

$ÜF_i$: Niteliğe ait *i.* üyelik fonksiyonunu,

SD(x): *x* elamanın $ÜF_i$ kümesine olan aitlik derecesini gösteren siluet küme doğrulama indeks değerini,

m_i : *i.* kümenin küme merkezini,

nd: $Düşük_{SD}$ kümesinin eleman sayısını,

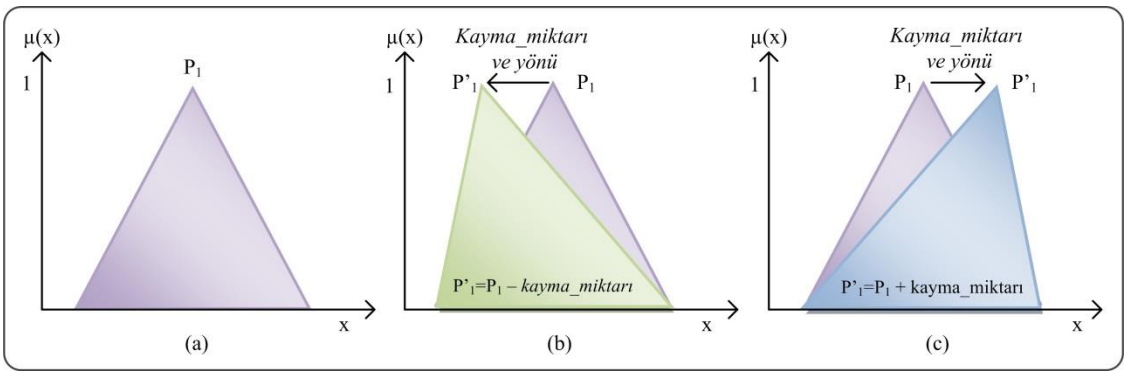
ny: $Yüksek_{SD}$ kümesinin eleman sayısını göstermektedir.

Tanım 6.3. Kayma miktarı ve yönü: Üyelik fonksiyonunun tepe noktasının ne yöne, ne kadar kayacağının ölçüsüdür. Kayma yönü, $Ortalama_{Düşük_{SD}}$ ve $Ortalama_{Yüksek_{SD}}$ değerlerine göre Çizelge 6.3'deki gibi bulunur. Kayma miktarlarının nasıl hesaplanacağı ise tablonun son sütununda gösterilmiştir.

Çizelge 6.4. Kayma yönü ve miktarını tespit etmek için kullanılan kriterler

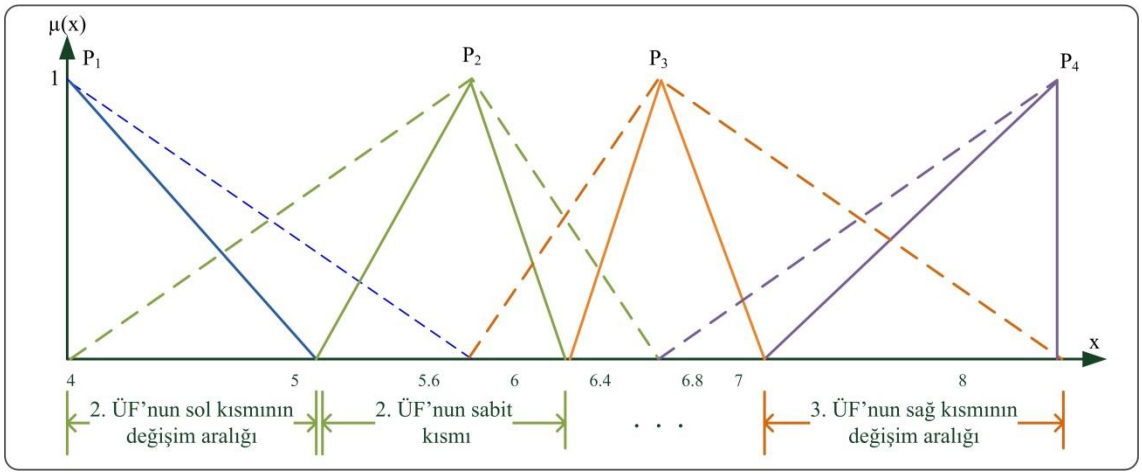
$Ort_{Düşük_{SD}}$	$Ort_{Yüksek_{SD}}$	**Kayma Yönü**	**Kayma Miktarı**
<0.7	<0.7	$Ort_{Düşük_{SD}} < Ort_{Yüksek_{SD}}$ ise sağa	(1- $Ort_{Düşük_{SD}}$)**rand*
		$Ort_{Düşük_{SD}} \geq Ort_{Yüksek_{SD}}$ ise sola	(1- $Ort_{Yüksek_{SD}}$)**rand*
<0.7	>0.7	Sağa	(1- $Ort_{Düşük_{SD}}$)**rand*
>0.7	<0.7	Sola	(1- $Ort_{Yüksek_{SD}}$)**rand*
>0.7	>0.7	Merkez Nokta	Kayma yok

Çizelgeden de görüldüğü üzere, $Ort_{Yüksek_{SD}}$ veya $Ort_{Düşük_{SD}}$ değerlerinden en az bir tanesi 0.7'den küçükse üyelik fonksiyonunun tepe noktasında kayma olacaktır. Kayma miktarı ise $Ort_{Yüksek_{SD}}$ veya $Ort_{Düşük_{SD}}$değerleri ile ters orantılı olarak belirlenir. Ortalama değeri ne kadar az ise, kayma miktarı o kadar fazla olmaktadır. Kayma miktarı hesaplanırken kullanılan *rand* ifadesi, 0 ile 1 arasında rastgele bir sayıyı temsil etmektedir. Şekil 6.3'de kayma miktarı ve yönünün nasıl olacağına örnekler verilmiştir. Şekilde x niteliğine ait bir üyelik fonksiyonuna ait 3 durum gösterilmiştir. Şekil 6.3(a)'da $Ort_{Yüksek_{SD}}$ ve $Ort_{Düşük_{SD}}$ değerlerinin ikisinin de 0.7'den yüksek olduğu durum verilmiştir. Tepe noktası kümenin merkezi olarak atanmış, herhangi bir kayma olmamıştır. Şekil 6.3(b) ve (c)'de de tepe noktaların kaydığı durumlar gösterilmiştir.



Şekil 6.3. *x* niteliğine ait bir ÜF'nun tepe noktasının alabileceği durumlar

Üyelik fonksiyonlarının tepe noktaları belirlendikten sonra, taban uzunluklarının da belirlenmesi gerekmektedir. Bunu için SDko algoritması, komşu üyelik fonksiyonlarının tepe noktalarından ve üyelik fonksiyonu temsil eden kümenin maksimum ve minimum elemanlarından faydalanır. Burada amaç, bir niteliğe ait her bir üyelik fonksiyonu için sabit taban kısımları tanımlayıp, üyelik fonksiyonları arasında boşluklar oluşmasını engellemektir. Böylece bir değişkenin evrensel uzayının (Universe of Discourse - UoD) tamamının üyelik fonksiyonları tarafından kapsanması, diğer bir deyişle evrensel uzaydaki her bir nokta en azından bir tane dilsel terimle etiketlenmesi sağlanır.



Şekil 6.4. k=4 için bir *x* niteliğine ait ÜF'larının başlangıç taban ve değişim aralıkları

Şekil 6.4'de 4 adet üyelik fonksiyonuna sahip bir *x* niteliğine ait ÜF'larının başlangıç taban ve değişim aralıkları gösterilmiştir. Şekilden de görüldüğü üzere her ÜF'nun değişmeyen sabit bir kısmı bulunmaktadır. Bu kısımlar sayesinde ÜF'ları arasında boşluklar oluşması önlenmektedir. ÜF'ların sağ ve sol ayakları için de değişim

aralıkları tanımlanmıştır. ÜF'nun sol ayağının değişim aralığının başlangıç noktası, kendisinden önceki ÜF'nun tepe noktası iken, bitiş noktası ise kendisinden önceki ÜF'nun sabit kısmının alabileceği en büyük değerdir. ÜF'nun sağ ayağının değişim noktaları belirlenirken de benzer bir süreç kullanılır. ÜF'nun sağ ayağının değişim aralığının başlangıç noktası, kendisinden sonraki üyelik fonksiyonunun sabit kısmının alabileceği en düşük değere karşılık gelir. Değişim aralığının bitiş noktası da, kendisinden sonraki ÜF'nun tepe noktasıdır.

Sabit kısımlar belirlenirken, ÜF'na karşılık gelen ve k-ortalama algoritması ile belirlenen kümeye ait tanımlayıcı istatistiklerden faydalanılır. Şöyle ki, bir ÜF'nun sabit kısmının alt sınırı, o kümeye dâhil olan en küçük değerli nokta olarak kabul edilir. Aynı şekilde, sabit kısmın üst sınır da o kümeye dâhil olan en büyük değerli nokta olarak alınır. Ancak ilk ve son ÜF'ları dik üçgen kullanılarak temsil edildikleri için, onların sabit kısımlarının belirlenmesinde istisnai bir durum oluşur. İlk ÜF'nun sol ayağı, son ÜF'nun da sağ ayağı yoktur. Bundan dolayı ilk ÜF'nun tepe noktası ve sabit kısmın alt sınırı, ilgili niteliğin alabileceği minimum değere, son ÜF'nun tepe noktası ve sabit kısmının üst sınırı da ilgili niteliğin alabileceği maksimum değere karşılık gelir.

6.2. GYAopt-aiNet Algoritması

Bu bölümde, SDko önişlem adımı ile tespit edilen bulanık tanecikleri derecelendirmesinin yanı sıra dinamik şekilde kural tabanını da optimize edebilen GYAopt-aiNet algoritması tanıtılacaktır. Daha önceden de belirtildiği gibi GYAopt-aiNet (**G**eliştirilmiş ve memetic algoritmanın **Y**erel **A**rama stratejisiyle birleştirilmiş **opt-aiNet**) algoritması hiper mutasyon ve baskılama mekanizmaları değiştirilerek elde edilen geliştirilmiş opt-aiNet algoritmasına Memetik algoritmanın yerel arama mekanizması eklenmesi ile elde edilmiştir. Önerilen algoritmanın adımları aşağıda verildiği gibidir.

1. Başlangıç Ab popülasyonunu Çizelge 6.5 'de verilen algoritmaya göre oluştur (Bölüm 6.2.2).
2. Boş olarak hafıza popülasyonu *arşiv_pop*'u oluştur.
3. Daha önceden belirlenen iterasyon sayısı kadar aşağıdaki işlemleri tekrar et:

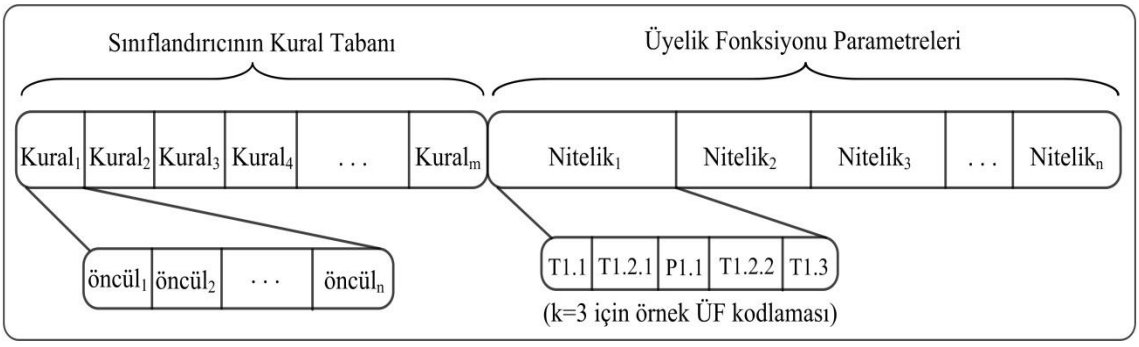
a. (i) ve (iv) arasındaki adımları, popülasyonun ortalama uygunluğu, bir önceki iterasyondaki popülasyonun ortalama uygunluğundan anlamlı bir şekilde farklı olduğu müddetçe tekrarla:
    i. Popülasyondaki Ab'lerin uygunluğunu hesapla (Bölüm 6.2.3).
    ii. Her bir Ab için N_c tane klon üret ve *pop_c*'yi oluştur. *pop_c*'deki her bir klona *Apaptif Hipermutasyon operatörünü* (Bölüm 6.2.4) uygula ve mutasyona uğramış klonların uygunluklarını hesapla.
    iii. Her bir klonları kendi ebeveyn'i ile kıyasla ve uygunluğu iyi olanı bir sonraki popülasyon için seç.
    iv. Popülasyonun ortalama uygunluğunu hesapla.
b. Popülasyondaki bireylerin birbirleri ile olan duyarlılıklarını hesapla ve ağ hücrelerine *Akıllı Baskılama Mekanizmasını* (Bölüm 6.2.5) uygula.
c. En yüksek uygunluğa sahip 3 Ab'yi *arşiv_pop*'a koymak için seç ve *arşiv_pop'a, yerel arama* (Bölüm 6.2.6) prosedürü tatbik et. Oluşan yeni *arşiv_pop* popülasyonu, orijinal popülasyona kopyala.
d. Popülasyonun farklılığını korumak için *%d* oranında yeni hücre oluştur.

4. Durdurma şartı sağlanmışsa algoritmayı sonlandır değilse adım (a)'a git.

Bölümün devamında GYAopt-aiNet algoritmasının her bir adımı ayrıntılı şekilde açıklanacaktır.

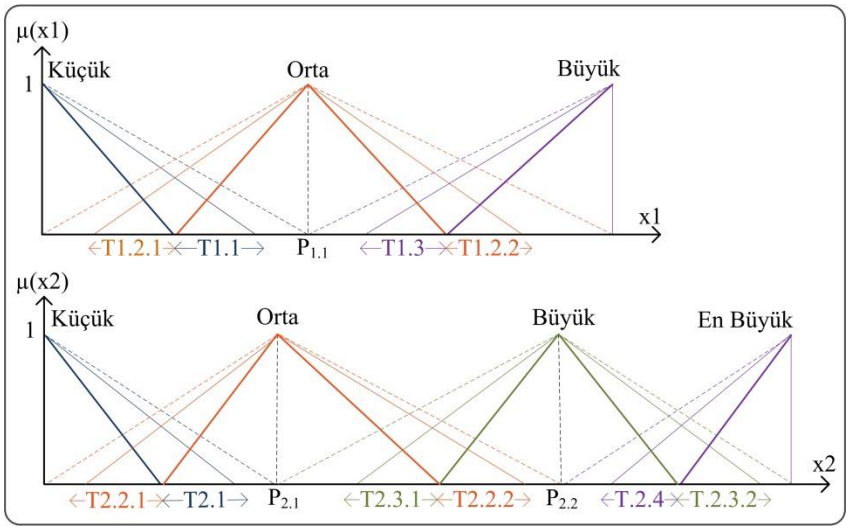
6.2.1. Antikorun/Bireyin Kodlanması

Her bir antikor iki kısımdan oluşmaktadır. İlk kısımda sistemin bulanık kural tabanına ait öncüller bulunurken, diğer kısmında üyelik fonksiyonunun parametreleri yer almaktadır.



Şekil 6.5. Örnek bir antikor yapısı

Şekil 6.5'de *m* tane kurala ve *n* tane niteliğe sahip bir bulanık sistem için, bulanık sınıflandırıcının kural tabanı ve üyelik fonksiyonlarının parametre değerlerini tutan bir antikorun yapısı gösterilmiştir. Şekilden görüldüğü üzere, Ab'nin bulanık kural tabanı kısmı, sisteme ait bütün bir bulanık kural kümesini temsil etmektedir. Bu örnek için *m* tane kural içerdiği varsayılmıştır. Örnek *n* tane nitelik içerdiği için, her bir kural *n* tane öncül kısımdan oluşmaktadır. Her bir $öncül_i$ (i=1,2,..,n) değişkeni 0 ile k_i arasında tam sayı bir değer tutar. k_i, ilgili değişkenin sahip olduğu üyelik fonksiyonu sayısıdır. Eğer $öncül_i$ =0 ise, $öncül_i$ 'nin içerisinde yer alan, 0'dan farklı değerler ilgili niteliğin hangi üyelik fonksiyonunun veya dilsel değerinin kuralda yer alacağını gösterir. Öncül kısmı antikorda tutulan kuralın sonuç kısmı ve kesinlik faktörü, Bölüm 4.4'de anlatılan yönteme göre hesaplanır. Antikorun üyelik fonksiyonu parametrelerini tutan kısmında üçgen üyelik fonksiyonlarının taban ve tepe değerleri bulunmaktadır. Her bir üyelik fonksiyonu, niteliğe ait dilsel bir değeri temsil etmektedir. Her bir *i* niteliği k_i tane dilsel değere sahiptir ve k_i sayısı SDko önişlem algoritması kullanılarak bulunmuştur.



Şekil 6.6. İki girişli bir bulanık sınıflandırıcı sistemi için üyelik fonksiyonu örneği

Şekil 6.6'da iki girişli bir bulanık sınıflandırıcı sistemi için üyelik fonksiyonu örneği verilmiştir. İlk değişken x_1, 3 üyelik fonksiyonuna sahipken, x_2 4 üyelik fonksiyonuna sahiptir ve her birinin dilsel bir etiketi vardır. Şekilden de görüldüğü üzere üyelik fonksiyonlarının kodlaması Bölüm 5.1'de anlatılana benzer şekildedir. Tek farkı taban ve tepe değerlerinin gerçek sayı formatında antikorda tutuluyor olmasıdır. Bu da antikorun uzunluğunun hesabında değişiklik olması anlamına gelir. Buna göre bir antikorun uzunluğu hesaplanırken denklem 6.5 ve 6.6 kullanılır.

$$NitelikUz_i = [(k_i - 2) * 3 + 2] \quad (6.5)$$

$$AntikorUz = n * m + \sum_{i=1}^{n} NitelikUz_i \quad (6.6)$$

Burada;

n = Bulanık sistemde bulunan nitelik sayısını,

k_i = i. Nitelikte bulunan dilsel değerlerin (üyelik fonksiyonların veya bulanık kümelerin) sayısını (i=1,2, … , n),

m = sistemdeki bulanık kuralların sayısını gösterir.

Formül 6.6'da *m* kural sayısın *n* ile çarpılmasının sebebi, her bir kuralın *n* tane öncüle sahip olmasıdır. Toplam *NitelikUz'a* bu çarpım sonucu da eklenince antikorun uzunluğu bulunur.

6.2.2. Başlangıç Popülasyonun Oluşturulması

Başlangıç popülasyonu, daha önceden belirlenen sayıda Ab kullanılarak rastgele şekilde oluşturulur. Denklem 6.5 ve 6.6 incelendiğinde popülasyondaki her bir antikorun uzunluğunun, niteliklerin içerdikleri üyelik fonksiyonu sayısına göre farklılık göstereceği görülmektedir. Özellikle çok nitelikli ve büyük veri kümelerinde, Ab'nin uzunluğu çözümlerin başarısını ve yakınsama hızını etkilemektedir.

Çizelge 6.5. Başlangıç popülasyonu oluşturulurken kullanılan nitelikler

İsmi	#Örnek	#Nit.	Seçilen Nitelik No.
Mammograph	830	5	1,3,4
Pima	768	8	1,2,6,8
Glass	214	9	1,2,3,6,7
Tic-tac-toe	958	9	1,3,5,7,9
Wisconsin	683	9	1,2,4,6,8
Cleveland	297	13	1,3,4,11,12
Zoo	101	16	1,2,4,5,6,9,13
Vehicle	846	18	1,3,5,6,8,9,17,18
Ionosphere	351	33	4,6,11,16
Spectfheart	267	44	15,21,28,39
Sonar	208	60	12,16,28,40,44,52

Bu problem, başlangıç popülasyonları oluşturulurken nitelik sayısı 8'den veya örnek sayısı 800'den fazla olan veri kümeleri için Çizelge 6.5'de listesi verilen niteliklerin kullanılması ile çözülmüştür. Bu nitelikler Derrac ve ark.'nın (2010) çalışmasında yer alan analizlerden faydalanılarak tespit edilmiştir. Diğer durumlarda,

başlangıç popülasyonunu oluşturulurken tüm nitelikler kullanılmıştır. Böylece niteliklerin ve değişkenlerin sayısı kontrol altına alınmış ve sistemin yorumlanabilirliğine katkı sağlanmıştır.

6.2.3. Uygunluk ve Duyarlılık Fonksiyonları

Uygunluk fonksiyonu, her bir bireyin kalitesini değerlendirir. Bu çalışmada, doğru sınıflandırılmış eğitim örüntülerinin sayısı, uygunluk fonksiyonu olarak kullanılmaktadır. Uygunluk fonksiyonun hesaplanması denklem 6.7'de verilmiştir.

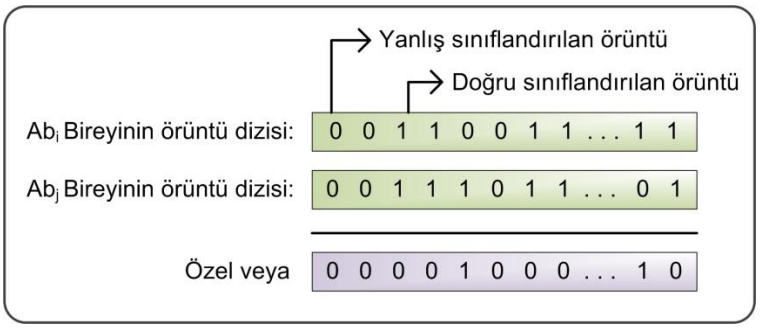
$$Uygunluk(Ab_j) = DoğruSınıflandırılanÖrüntüSayısı(Kural_Tabanı_j) \quad (6.7)$$

Burada;

Ab_j = popülasyondaki *j*. bireyi,

$Kural_Tabanı_j = Ab_j$ bireyinin kural tabanı temsil eder.

Diğer bir deyişle, Şekil 6.7'deki Ab_j bireyinin örüntü dizisindeki 1'lerin toplamı Ab_j bireyinin uygunluk değerini verir.



Şekil 6.7. Duyarlılık fonksiyonun hesaplanması

Duyarlılık, yapay bağışıklık sisteminin elemanları arasındaki ilişkinin ölçüsüdür. Genellikle, sistemdeki bireylerin birbirleri ile olan benzerliklerini veya farklılıklarını ölçmek için çeşitli uzaklık ölçüleri duyarlılık fonksiyonu olarak kullanılır. Bu uzaklık ölçütlerinden en sık kullanılanları Öklid ve Manhattan uzaklıklarıdır. Bu ölçütler, bireylerin şekil uzayında bulundukları konumlar arasındaki mesafeyi ölçerler. Ancak bu çalışmada ilgilenilen, bireylerin yerlerinin benzerlikleri değil bireylerin işlevselliklerinin benzerlikleridir. Bundan dolayı, *n*-boyutlu Ab şekil uzayı Hamming uzayına dönüştürülmüş ve uzaklık ölçütü olarak Hamming uzaklığı kullanılmıştır. Bu dönüşümün nasıl olduğu Şekil 6.7'de gösterilmiştir.

Her bir Ab bireyi için Şekil 6.7'de gösterilen şekilde bir örüntü dizisi tanımlanır. Dizinin boyutu, sınıflandırılacak eğitim veri kümesin boyutuna eşittir. Eğer bir örüntü, bulanık sistem tarafından doğru sınıflandırılmış ise, ona karşılık gelen dizideki bitin değeri 1, değilse 0 olarak ayarlanır. İki dizi aynı örüntüleri aynı şekilde sınıflandırıyorsa, bu diziler işlevsel olarak birbirine benziyor demektir. Şekil uzayında konum olarak nerede yer aldıkları önemsizleşmiştir.

Şekil 6.7'deki örüntü dizileri Hamming uzaklığı kullanılarak kıyaslanır ve sonuç dizisinde yer alan 0'ların yüzdesi bu çalışmada duyarlılık değeri olarak kullanılır. Hamming uzaklığı, aynı uzunluktaki iki dizide yer alan farklı elemanların sayısıdır ve farklı elemanların sayısını bulurken özel-veya (xor) lojik operatöründen faydalanır. Bilindiği üzere, eğer iki bit aynıysa özel-veya operatörü sonuç olarak 0, değilse 1 değerini döndürür. Bu işlem sonucunda iki bireyin işlevsel olarak birbirine ne kadar benzediği tespit edilir.

6.2.4. Akıllı Baskılama Mekanizması

Opt-aiNet algoritmasında, baskılama mekanizmasındaki duyarlılık Öklid uzaklığı kullanılarak hesaplanır. Her giriş, eğitim kümesindeki bir örüntü satırına karşılık gelir. Eğer hesaplanan mesafe, baskılama eşiğininden düşükse, ilk giriş vektörü baskılanır, diğeri sistemde kalır. Bu esnada girişlerin uygunluk değerleri dikkate alınmaz. Bu da, daha iyi bir bireyin belki de optimal çözümün silinmesine yol açabilir.

Bu problemin üstesinden gelebilmek için bu çalışmada, duyarlılık hesabını bir önceki bölümde anlatılan yöntem ile yapan, yeni ve *akıllı* bir baskılama mekanizmasının kullanılması önerilmiştir. Önerilen baskılama mekanizmasının akıllı olarak adlandırılmasının sebebi, her bir antikorun baskıladığı diğer antikorların listesini tutuyor olmasıdır. Bir antikor baskılanacağı zaman, onu baskılayan antikor, baskılanacak antikorun listesini kontrol eder. Eğer baskılayan antikor, listedeki tüm antikorları temsil edebiliyorsa baskılanacak antikor bağışıklık ağından çıkarılır. Çünkü onun baskıladığı yani temsil ettiği tüm antikorları temsil edebilecek yeni ve daha iyi bir antikor bağışılık ağına dâhil olmuş demektir. Ağdan çıkarılan antikor, onu baskılayan antikorun silme listesine eklenir. Aksi takdirde, antikor korunur. Böylece muhtemel bir çözümün silinmesi engellenir. Baskılama esnasında ayrıca antikorların uygunlukları da dikkate alınır. İki antikordan her zaman uygunluk değeri küçük olan sistemden çıkarılır

yani baskılanır, uygunluğu yüksek olan ise her zaman korunur. Çizelge 6.6'da önerilen akıllı baskılama mekanizmasına ait sözde kod verilmiştir.

Çizelge 6.6. Akıllı baskılama mekanizması algoritmasına ait sözde kod

Algoritma 4: Akıllı Baskılama Mekanizması

Girişler: Ab popülasyonu – *popAb*,
Eğitim Kümesi – *egtKüme*,
Baskılama Eşiği – *baskılama_Esik*
Çıkış: Baskılanmış *popAb*

begin
popBoy = |*popAb*|;
Antikorlar arası benzerlik matrisini oluştur (Bölüm 6.2.4'deki duyarlılık fonksiyonu kullanarak);
Duyarlılıkları *baskılama_Esik'den* büyük olan antikor çiftlerini tespit et;
for antikor çiftlerinin sayısı kadar **do,**
Ab çiftinden uygunluğu yüksek olan antikorunu (Ab_{maks}) seç;
if (Ab_{maks} , diğer antikorun (Ab_{min}) listesindeki tüm antikorları temsil ediyorsa)
Ab_{min}'i baskıla;
Ab_{maks}'ın listesine, baskıladığı Ab_{min}'i ekle;
end if;
end for;
end begin.

Akıllı baskılama mekanizmasının çalışma mantığını bir örnek üzerinde açıklayalım. 3 antikorlu bir popülasyonun olduğunu ve bu popülasyon üzerinde akıllı baskılama mekanizmasının çalıştırılacağını varsayalım. Bu antikorların uygunluk değerleri arasındaki ilişki de *Uyg*(Ab_3)>*Uyg* (Ab_1)>*Uyg* (Ab_2) şeklinde olsun. Şimdi aşağıdaki adımları sırası ile gerçekleştirelim:

1. *Duyarlılık(Ab_1,Ab_2)* değerine bak;
2. Eğer *Duyarlılık(Ab_1,Ab_2)>baskılama_eşiği* ise,
   - Ab_2'yi sil; (uyg(Ab_2)daha düşük olduğu için)
   - Ab_1'in listesinine Ab_2'yi ekle;
3. *Duyarlılık(Ab_1,Ab_3)* değerine bak;
4. Eğer *Duyarlılık(Ab_1,Ab_3)>baskılama_eşiği* ise,
   - *Duyarlılık(Ab_2,Ab_3)* değerine bak;
   - Eğer *Duyarlılık(Ab_2,Ab_3)>baskılama_eşiği* ise,
     - Ab_1 silinir;
     - Ab_3'ün listesinine Ab_1'i ekle;
   - Değilse, Ab_1 korunur.

Örnekten de görüldüğü üzere, Ab_1 sadece Ab_3'ün Ab_2'yi temsil ettiği durumda silinebilir. Böylece Ab_1 silinse bile, Ab_1 tarafından daha önce baskılanan Ab_2, Ab_3 tarafından temsil edilmeye devam edecektir.

6.2.5. Adaptif Hipermutasyon Mekanizması

Evrimsel algoritmalarda, popülasyonda çeşitliliği sağlamak ve bireylerin yerel optimum noktalarının çevresinde birikmesini engellemek için çeşitli operatörlerden faydalanılır. Bu amaç için genetik algoritmalarda, çaprazlama ve mutasyon operatörü kullanılırken; yapay bağışıklık algoritmalarında sadece mutasyon operatörü kullanılır ve bu operatör hiper mutasyon olarak adlandırılır. Bundan dolayı etkin bir mutasyon işlemi YBS algoritmaları için oldukça önemlidir. Bu çalışmada, orijinal opt-aiNet algoritmasında kullanılan hipermutasyon mekanizması yerine, adaptif bir hiper mutasyon mekanizmasının (AHM) kullanılması önerilmiştir. Böylece algoritmanın arama verimliliği arttırılmıştır.

Orijinal opt-aiNet algoritmasındaki hipermutasyon mekanizmasında, klonlar denklem 3.1'de verilen $c' = c + \alpha * N(0,1)$ formülü kullanılarak mutasyona uğratılır. Bu formüldedeki α değişim oranı denklem 3.2'de verilen $\alpha = (1/\beta)exp(-f^*)$ förmülü kullanılarak hesaplanır. Denklemden görüldüğü üzere α'daki değişim oranını β sabiti kontrol eder. β sabit sayısı ne kadar büyük seçilirse, mutasyondaki değişim oranı o kadar az olur. Diğer bir deyişle β, α ile ters orantılıdır ve algoritmanın yakınsamasını direkt etkileyen bir parametredir. β sabiti, AHM'de iterasyon adımına duyarlı olacak şekilde adaptif hale getirilmiştir. Bunun için Tan ve ark. (2006) tarafından genetik algoritma için tanımlanan ve denklem 6.8'de verilen adaptif mutasyon operatöründen faydalanılmıştır. Böylece mutasyon değişim miktarı olan α'nın büyük bir değerden başlatılması ve iterasyon numarası arttıkça mutasyon değişim miktarının azaltılması sağlanmıştır. Yani algoritmanın başlangıcında, arama uzayı büyük adımlar ile taranmış, algoritmanın ilerleyen aşamaların da ise adım boyu azaltılarak ve daha yerel bir arama gerçekleştirilmiştir.

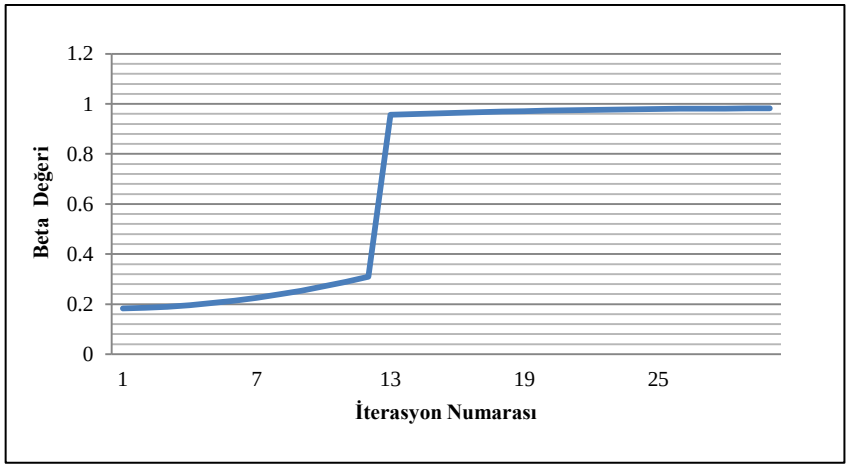
$$\beta(i) = \begin{Bmatrix} a\left[1 - \left(\frac{i}{itrNum}\right)^2\right] + b, & 0 \le i < in \\ a\left[0.1\left(\frac{i-itrNum}{itrNum}\right)^2\right] + b, & in \le i \le itrNum \end{Bmatrix} \tag{6.8}$$

Burada;

i = anlık iterasyon numarasını,

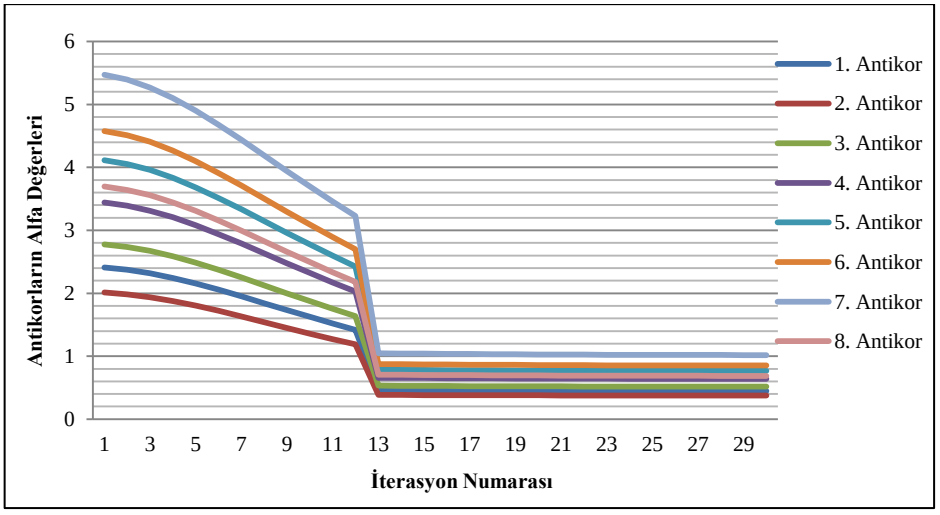
$\beta(i)$ = i. iterasyonda kullanılacak β değerini,

itrNum = algoritmanın iterasyon sayısını gösterir. *a,b* ve *in* sabitlerinin değerleri başlangıçta kullanıcı tarafından belirlenir. Bu çalışmada da, a=0.8, *in*=*itrNum*/2 ve *b*=1/*AntikorUz* olarak kullanılmıştır.



Şekil 6.8. İterasyon numarasına göre β değerindeki değişim

Şekil 6.8'de, 8 antikora sahip bir popülasyon için, algoritma 30 (*itrNum*=30) kez çalıştırıldığında; iterasyon numarası i'ye göre β değerindeki değişim gösterilmiştir. Algoritmanın ilk yarısında, 0.2 ile 0.3 aralığında değer alan β değeri, ikinci yarısında 0.9 ile 1 arasında değer almıştır. Anlık iterasyon numarasına göre, popülasyondaki her bir antikorun α değerlerindeki değişim Şekil 6.9'da verilmiştir.



Şekil 6.9. İterasyon numarasına göre popülasyondaki antikorların α değerlerindeki değişim

Şekil 6.9'dan da görüldüğü üzere, antikorların α değerleri iterasyon numarası arttıkça azalmaktadır. Bireylerdeki değişim başlarda fazla iken, *in* sabitinde tutulan eşik değerini geçtikten sonra oldukça az olduğu şekilden net bir şekilde gözlemlenmektedir. Diğer bir deyişle, algoritmanın ilk yarısında (*in*=*itrNum*/2 olarak belirlendiği için) arama uzayı büyük adımlarla taranmış, son yarısında ise daha küçük adımlarla taranarak daha yerel bir arama yapılması sağlanmıştır. Bunun sebebi, algoritma sonlara yaklaştıkça bireylerin çözüm noktasına yaklaşma olasılıklarının artıyor olmasıdır. Mutasyon değişim oranını fazla tutarak çözümden uzaklaşılması istenmeyen bir durumdur.

6.2.6. Memetik Algoritmanın Yerel Arama Mekanizması

Evrimsel algoritmalar, içerisinde barındırdıkları kalıtsal genel arama yetenekleri sayesinde, yerel optimumlara takılmaktan kurtulur ve çözüm uzayının farklı bölgelerini tarama imkânı bulurlar. Ancak, yerel arama mekanizmasını da kullanan hibrit evrimsel algoritmalar veya başka bir deyişle memetik algoritmalar çözüm uzayının daha verimli taranabilmesini sağlarlar.

Memetik algoritmaları diğer evrimsel algoritmalardan ayıran kısmı içerisinde barındırdığı bu yerel arama mekanizmasıdır. MA'daki yerel arama adımı bireylerin kültürel değişikliklerini temsil eder ve evrimsel algoritmalardan farklı olarak bireylerdeki değişim bir sonraki nesle geçmeden gözlemlenebilir. Yerel araştırma mekanizmasının öncelikli amacı eldeki bir çözümünü daha iyi bir çözüme götürmektir.

Bu tez çalışmasında, akıllı baskılama mekanizması ve adaptif mutasyon operatörü eklenerek geliştirilen orijinal opt-aiNet algoritmasına, algoritmanın arama verimliliğini daha da artırmak için bir yerel arama metodu eklenmiştir. Yerel arama metodu olarak Ang (2010) tarafından önerilen orijinal adı EMA-AIS (Evolutionary Memetic Algorithm based Artificial Immune System) olan ve bu çalışmada EMA-YBS olarak isimlendirilen yöntem kullanılmıştır. Sonuç olarak, GYAopt-aiNet algoritması elde edilmiştir.

Şekil 6.10'da GYAopt-aiNet algoritmasında kullanılan EMA-YBS yerel arama yönteminin çalışma mantığını gösteren akış diyagramı verilmiştir. EMA-YBS yöntemi, bağışıklık sisteminin klonal seçme prensibinden esinlenmiştir. Ancak hiper mutasyon operatörü yerine, mikro-mutasyon olarak isimlendirdiği bir mutasyon operatörü

kullanmıştır. Bu operatör de, adaptif mutasyon mekanizmasındaki gibi iterasyon sayısına duyarlıdır. Orijinal eşitlikte, mutasyon miktarındaki değişim verilmiştir. Bu çalışmada ise, denklem 6.8'de olduğu gibi mikro-mutasyon formülü, hiper mutasyon mekanizmasındaki β değişim sabitine uyarlanmıştır. Eşitliğin yeni hali, denklem 6.9'da verilmiştir.

$$\beta(i) = a * \left(\frac{1}{i}\right) + b, \{i \in Z_+ : [1, itrNumYA]\} \quad (6.9)$$

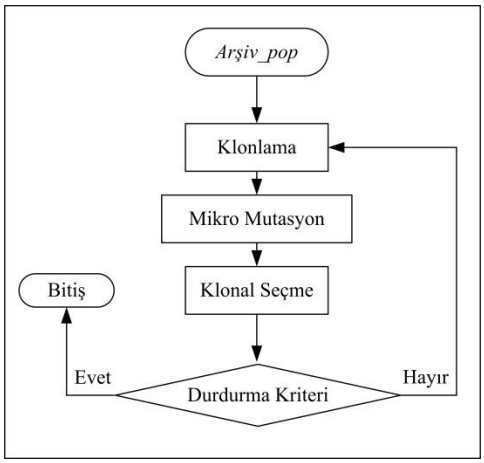
Burada;

i = anlık iterasyon numarasını,

$\beta(i)$ = i. iterasyonda kullanılacak β değerini,

itrNumYA = yerel arama algoritmanın kaç iterasyon çalışacağını gösterir. a ve b sabitlerinin değeri başlangıçta kullanıcı tarafından belirlenir. Bu çalışmada da, a=0.8 ve b=1/*AntikorUz* olarak belirlenmiştir.

Şekil 6.10'da verilen EMA-YBS algoritması, tüm popülasyona değil sadece *arşiv_pop*'a uygulanır. Çünkü yerel arama algoritmalarının optimum çözüme ulaşması büyük oranda başlangıç çözümünün genel optimumun bulunduğu bölgeye uzaklığına ve iterasyon sayısına bağlıdır. *Arşiv_pop*, popülasyonun en iyi 3 bireyinden oluşmaktadır ve mevcut popülasyondaki optimum çözüme en yakın bireyler bunlardır.



Şekil 6.10. EMA-YBS yerel arama algoritması (Ang,2010)

EMA-YBS yerel arama algoritmasının her bir iterasyonunda, *arşiv-pop*'daki bireylerin uygunlukları hesaplanır. Daha sonra bireyler klonlanır ve her bir klona denklem 6.9'da verilen mikro-mutasyon operatörü uygulanır. Yeni bireylerin de uygunlukları hesaplanır ve en iyi olanlar saklanır. Bu işlem durdurma kriteri sağlanana

kadar devam eder. Durdurma kriteri olarak, önceden belirlenen iterasyon sayısı kullanılır.

6.3. Sunulan Yöntemin Deneysel Sonuçları

Bu bölümde önerilen yöntemler, özel bir bulanık mantık sistem türü olan bulanık sınıflandırıcılar problemine tatbik edilmiş ve yöntemlerin verimlilikleri araştırılmıştır. Deneysel çalışmalar iki aşamada gerçekleştirilmiştir. İlk olarak, orijinal opt-aiNet algoritması ile GYAopt-aiNet algoritması kıyaslanmış ve yapılan değişikliklerin performansı nasıl etkilediği incelenmiştir. Bu bölümde ayrıca SDko önişlem algoritmasının kullanılmasının bulanık sınıflandırıcı sistem üzerindeki etkisine de bakılmıştır.

Çizelge 6.7. Orjinal opt-aiNet ve GYAopt-aiNet algoritmalarının kullandıkları parametreler ve değerleri

Algoritmanın Parametreleri	**Parametrelerin Değerleri**
Başlangıç popülasyonun boyutu:	20 (Her iki algoritma için de aynı)
Her bir hücre için üretilecek klon sayısı:	8 (Her iki algoritma için de aynı)
Algoritmanın çalıştırılacağı iterasyon sayısı:	30 (Algoritmalar dâhili bir *while* döngüsü içerdiğinden, iterasyon sayıları 600 ile 1000 arasında değişebilmektedir.)
Baskılama eşiği:	0.90 (Her iki algoritma için de aynı ancak GYAopt-aiNet için akıllı baskılama mekanizması kullanılmıştır.)
İki ortalama popülasyon uygunluğu arasındaki hata eşiği:	0.0001 (Her iki algoritma için de aynı)
Mevcut popülasyonun yüzde kaçı kadar yeni birey oluşturulacağı:	%20 (Her iki algoritma için de aynı)
Mutasyon miktarı:	GYAopt-aiNet için adaptif mutasyon parametresi, opt-aiNet için 0.2 değeri kullanılmıştır.

İkinci aşamada, listesi Çizelge 2.3'de verilen diğer sınıflandırıcı algoritmalar ile bulanık sınıflandırıcı için bu tez çalışmasında önerilen yöntem kıyaslanmıştır. Kıyaslamak için kullanılan sınıflandırma yöntemlerinin sonuçları, KEEL (http://www.keel.es/) isminde ticari olmayan açık kaynak kodlu bir yazılım aracı kullanılarak elde edilmiştir. Sonuçlar, üç kez tekrarlanan 10 çapraz doğrulama

yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak bulunmuştur. Yani her bir sonuç için algoritmalar 30'ar kez çalıştırılmıştır. Tüm kıyaslamalar, listesi Çizelge 2.2'de verilen veri kümeleri kullanılarak gerçekleştirilmiştir. Bu deneysel çalışmalar esnasında, GYAopt-aiNet ve orjinal opt-aiNet algoritmalarının kullandıkları parametreler ve değerleri Çizelge 6.7'de verilmiştir.

6.3.1. GYAopt-aiNet ile opt-aiNet'in Kıyaslanması

Bu bölümde, GYAopt-aiNet algoritması ile orijinal opt-aiNet algoritmasının kıyaslaması yapılmıştır. Her iki yöntemin de bulanık sınıflandırıcı tasarımındaki verimliliği incelenmiştir. Çizelge 6.8'de, algoritmalardan elde edilen sonuçlar verilmiştir.

Çizelge 6.8. GYAopt-aiNet ile orijinal opt-aiNet'in kıyaslanması

	opt-aiNet	**opt-aiNet (SDko ile)**	**GYAopt-aiNet**
Bupa	0.556611	0.606583	**0.747171**
Ecoli	0.411794	0.528075	**0.798812**
Glass	0.235137	0.304401	**0.72619**
Haberman	0.729099	0.736828	**0.75457**
Iris	0.715556	0.840000	**0.993333**
Newthroid	0.732540	0.843074	**0.981313**
Pima	0.487822	0.585538	**0.798622**
Vehicle	0.24866	0.436261	**0.715159**
Appendicitis	0.640303	0.811212	**0.930909**
Balance	0.4149	0.419099	**0.764951**
Cleveland	0.004444	0.152414	**0.63**
Ionosphere	0.600238	0.663810	**0.939206**
Magic	0.465263	0.554329	**0.818072**
Mammographic	0.599603	0.640079	**0.845635**
Page-blocks	0.52211	0.854074	**0.929383**
Sonar	0.504206	0.602381	**0.825238**
Spectfheart	0.770228	0.793875	**0.79416**
Tic-tac-toe	0.048081	0.412891	**0.74803**
Wisconsine	0.35888	0.571341	**0.958483**
Zoo	0.406667	0.363030	**0.970303**
Ortalama	0.472607	0.585965	**0.833477**

Çizelgenin ilk sütununda, opt-aiNet algoritmasının herhangi bir önişlem uygulamaksızın veri kümelerine tatbik edildiğinde elde edilen bulanık sınıflandırıcılarının, sınıflandırma doğrulukları verilmiştir. GYAopt-aiNet algoritmasından önce önişlem adımı olarak SDko yöntemi kullanıldığından, kıyaslama da eşitliği sağlamak için opt-aiNet algoritmasından önce de önişlem adımının kullanıldığı durumda elde edilen sınıflandırma doğrulukları çizelgenin ikinci sütunun da

verilmiştir. Son sütunda ise, bulanık sınıflandırıcı tasarımı için bu tez çalışmasında önerilen, SDko önişlem adımı ile GYAopt-aiNet algoritmalarını kullanan ve Şekil 6.1'de genel yapısı verilen yöntemden elde edilen sonuçlar verilmiştir. Son satır da, her bir algoritmanın ortalama sınıflandırma doğrulukları verilmiştir. Çizelgeden de görüldüğü üzere, en yüksek ortalama değere SDko önişlem adımı ile birlikte kullanılan GYAopt-aiNet algoritması sahiptir. Ortalamanın haricinde, tüm veri kümeleri içinde en yüksek sonucu yine önerilen yöntem vermiştir. Tezin devamında GYAopt-aiNet denildiğinde, bulanık sınıflandırıcı için sunulan yöntem kast edildiği için kendisinden önce SDko önişlem algoritmasının kullanıldığı her seferinde tekrar edilmeyecektir. Sonuç çizelgelerinde bu sebepten dolayı GYAopt-aiNet ismi tek başına kullanılmıştır.

Çizelge 6.8 incelendiğinde, opt-aiNet algoritmasının tek başına kullanıldığı durumda elde edilen sonuçların hiç tatminkar olmadığı görülmektedir. Bunun sebebi, üyelik fonksiyonları oluşturulurken, bulanık kümelerin yerleşiminde hiçbir kontrol yapılmamasıdır. Çünkü üyelik fonksiyonlarını temsil eden bulanık kümeler arasında sıklıkla boşluklar oluşabilmekte, bu da bulanık sistemin yorumlanabilirliğinin ve sınıflandırma başarısının düşmesine sebep olmaktadır. Çözüm olarak, Bölüm 6.1 'de anlatılan SDko önişlem adımı önerilmiştir. Sonuçlar incelendiğinde, önişlem yönteminin kullanılmasının sonuçlar üzerinde olumlu bir etki yaptığı gözlemlenmiştir. Ancak halen istenilen sonuçlar elde edilememiştir. Bu sebepten dolayı opt-aiNet algoritması daha ayrıntılı olarak incelenmiş ve Bölüm 6.2 bahsedilen eksiklikler tespit edilerek bu eksikliklerin giderilmesi için baskımla ve hiper mutasyon mekanizmalarına dair çözümler önerilmiştir. Önerilen çözümlerin sonuçlar üzerindeki olumlu etkileri Çizelge 6.8'den net bir şekilde görülmektedir.

Bu sonuçlar arasında gerçekten anlamlı bir fark olup olmadığını araştırmak için Wilcoxon işaret-sıra istatistikî testi kullanılmıştır. Bunun için oluşturulan H_0 hipotezleri şu şekilde oluşturulmuştur:

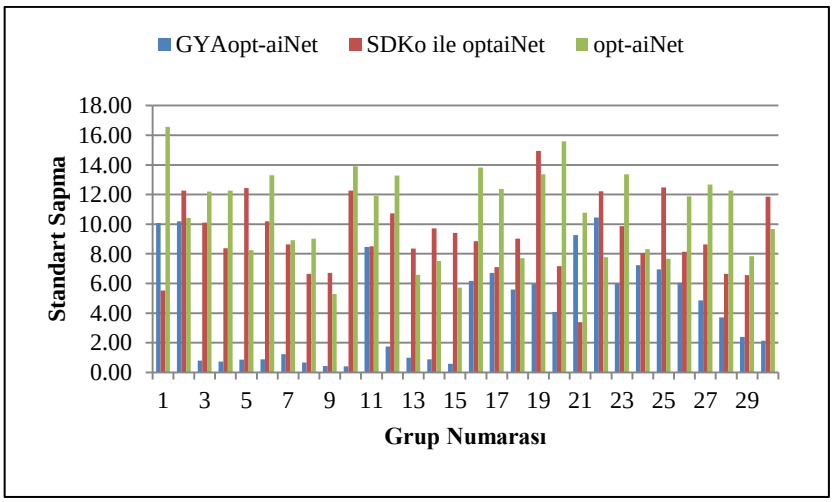
1. "$H_{0:}$ GYAopt-aiNet ile orijinal opt-aiNet algoritması arasında %95 önem seviyesinde istatistiksel olarak anlamsal bir fark yoktur."
2. "$H_{0:}$ GYAopt-aiNet ile SDko ön işlem algoritmalı opt-aiNet algoritması arasında %95 önem seviyesinde istatistiksel olarak anlamsal bir fark yoktur."

Bu iki testin doğruluğu %95 önem seviyesinde Wilcoxon işaret-sıra istatistikî testi kullanılarak test edilmiştir. Her iki hipotez için de p değeri 0.0 olarak

hesaplanmıştır. Bunun anlamı, H_0 hipotezlerinin reddedildiğidir. Yani GYAopt-aiNet algoritması ile diğer iki algoritma arasında %95 önem seviyesinde istatistiksel olarak anlamsal bir farklılık vardır. Özellikle 2. H_0 hipotezi bizim için daha anlamlıdır. Çünkü her iki algoritmada, veri kümelerinin SDko önişlem algoritması ile belirlenen başlangıç üyelik fonksiyonları üzerinde çalıştırılmıştır.

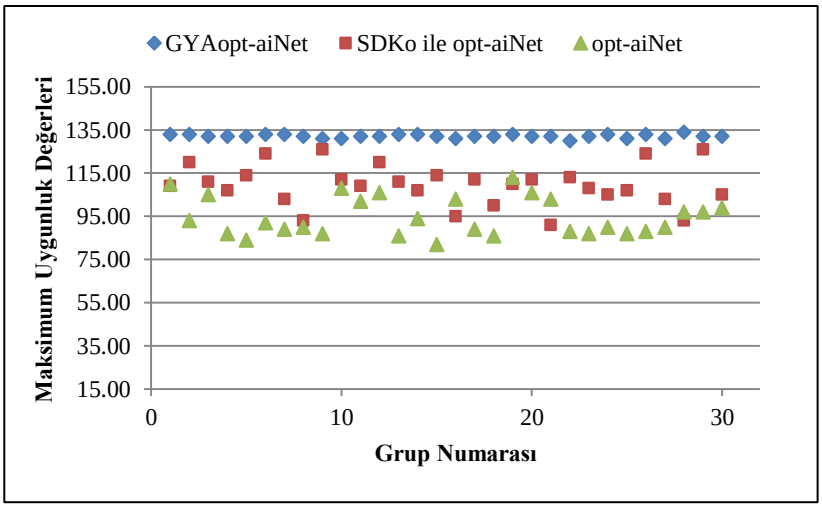
İstatistiki test sonuçlarıyla iki algoritmanın birbirlerinden farklı olduğu gösterilmiştir, ancak hangi algoritmanın daha iyi olduğu henüz belli değildir. Bundan dolayı, algoritmaların iris veri kümesi için bazı tanımlayıcı istatistikî bilgileri incelenecektir. Şekil 6.11 ve Şekil 6.12'de sırası ile her bir grubun standart sapması ve maksimum uygunluk değeri verilmiştir.

Tüm işlemler 3x10-kez çapraz doğruma işlemi sonucunda elde edilmiştir. Dolayısı ile her bir algoritma 30 farklı grup üzerinde test edilmiştir. Uygunluk olarak, grup içerisindeki doğru sınıflandırılan bireylerin sayısı kullanılmıştır. İris veri kümesi 150 tane veri satırı içermektedir. Veri seti, 10-kez çapraz doğrulama işlemine tabi tutulduğunda, her bir eğitim kümesi 135'er veri satırı içerirken her bir test kümesi de 15'er veri satırı içerecektir. Diğer bir deyişle bir eğitim veri kümesi için en yüksek uygunluk değeri 135 olacaktır.



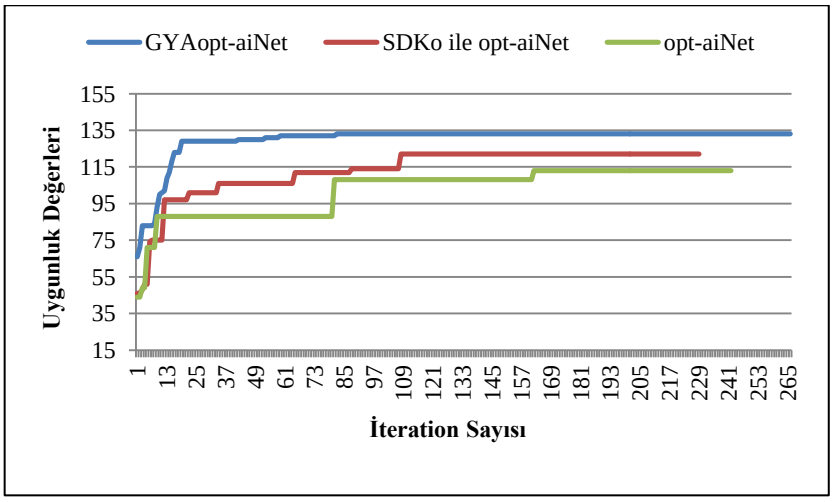
Şekil 6.11. Grupların standart sapma değerleri

Şekil 6.11'de grupların standart sapma değerleri verilmiştir. Şekilden de görüldüğü üzere, GYAopt-aiNet'in standart sapması tüm gruplarda opt-aiNet'den, 2 grup dışında SDKo'lu opt-aiNet'den daha küçüktür. Yani GYAopt-aiNet algoritmasının diğer iki algoritmadan daha tutarlı sonuçlar ürettiği söyleyenebilir.



Şekil 6.12. Gruplardaki en büyük uygunluk değerleri

Şekil 6.12'de her bir gruptaki en yüksek uygunluğa sahip birey grafik üzerinde işaretlenmiştir. Bir bireyin alabileceği maksimum uygunluk değerinin iris eğitim kümesi için 135 olduğu belirtilmişti. Grafik incelendiğinde, GYAopt-aiNet algoritmasının tüm gruplarda diğer algoritmalardan daha yüksek uygunluk değerine sahip olduğu görülmektedir. Bununla birlikte GYAopt-aiNet tüm gruplar da 130'ün üzerinde uygunluk değeri üretmiştir.



Şekil 6.13. Gruplardaki en iyi bireyin iterasyon sayısına göre uygunluk değerleri

Şekil 6.13'de, algoritmaların yakınsamalarını incelemek için her bir grubun en iyi bireyinin iterasyon sayısına göre uygunluk değerindeki değişimler gösterilmiştir. Grafikten de görüldüğü üzere, GYAopt-aiNet algoritması optimum sonuca diğer iki algoritmadan daha hızlı ulaşmış ve daha istikrarlı bir grafik oluşturmuştur.

6.3.2. Sunulan Yöntemin Diğer Yöntemlerle Kıyaslanması

Bu bölümde, listesi Çizelge 2.3'de verilen diğer sınıflandırıcı algoritmalar ile GYAopt-aiNet algoritması kıyaslanmıştır. Sonuçlar, daha önceki bölümde de olduğu gibi üç kez tekrarlanan 10 çapraz doğrulama yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak bulunmuştur. Kıyaslama sonuçları, Çizelge 2.3'de belirtilen gruplara göre tasnif edilmiş ve 4 ayrı çizelge de sunulmuştur.

İlk olarak, grup-1'de yer alan evrimsel bulanık kural öğrenme algoritmaları ile GYAopt-aiNet algoritmasının sonuçları Çizelge 6.9'da verilmiştir. Bu tez çalışmasında önerilen GYAopt-aiNet algoritması da grup-1'de yer almaktadır yani evrimsel bir bulanık kural öğrenme algoritmasıdır. Bu sebepten dolayı bu gruptaki kıyaslama sonuçları bizim için önemlidir.

Çizelge 6.9'da verilen sonuçlara göre, GYAopt-aiNet algoritması 20 veri kümesinin 15'inde en yüksek ortalama sınıflandırma doğruluğuna sahiptir. Algoritmaların tüm veri kümelerinde elde ettiği sınıflandırma doğruluklarının genel ortalamaları incelendiğinde, GYAopt-aiNet algoritması 0.833477 değeri ile ilk sırada yer aldığı görülmektedir.

Çizelge 6.9. Evrimsel Bulanık Kural Öğrenme Algoritmaları (Grup-1) ile GYAopt-aiNet'in kıyaslanması

	GCCL	**SGERD**	**SLAVE**	**AdaBoost**	**IGA**	**SSMA**	**GYAopt-aiNet**
Bupa	0.58559	0.57219	0.589898	0.673286	0.610889	0.617187	**0.747171**
Ecoli	0.646168	0.750178	**0.848336**	0.599198	0.693316	0.814914	0.798812
Glass	0.620323	0.617675	0.587296	0.440739	0.567132	0.648673	**0.72619**
Haberman	0.732043	0.733047	0.713369	0.718925	0.663513	0.707921	**0.75457**
Iris	0.948889	0.955556	0.957778	0.96	0.922222	0.946667	**0.993333**
Newthroid	0.85974	0.856205	0.909091	0.897835	0.871429	0.955267	**0.981313**
Pima	0.683629	0.728847	0.732378	0.745207	0.713086	0.743229	**0.798622**
Vehicle	0.577386	0.529552	0.632857	0.579281	0.653291	0.654514	**0.715159**
Appendicitis	0.821212	0.872424	0.838182	0.875455	0.790909	0.851515	**0.930909**
Balance	0.816368	0.734187	0.748191	0.686354	0.829186	**0.88149**	0.764951
Cleveland	0.535364	0.496015	0.545441	0.54908	0.50751	0.56	**0.63**
Ionosphere	0.876614	0.809074	0.927857	0.643889	0.822513	0.891693	**0.939206**
Magic	0.749352	0.724161	0.803369	0.804932	0.783537	0.800898	**0.818072**
Mammographic	0.790763	0.769478	0.790361	0.793574	0.774699	0.803213	**0.845635**
Page-blocks	0.905118	0.903939	0.938002	0.899663	0.917351	**0.938013**	0.929383
Sonar	0.651825	0.704048	0.676746	0.46619	0.676111	0.772063	**0.825238**
Spectfheart	0.79169	0.781814	0.79169	0.20584	0.723789	0.734141	**0.79416**
Tic-tac-toe	0.653443	0.694178	0.653443	**0.803417**	0.673659	0.676075	0.74803
Wisconsine	0.96342	0.93712	0.960053	0.942903	0.954589	**0.968286**	0.958483
Zoo	0.405455	0.847576	0	0.405455	0.726061	0.903939	**0.970303**
Ortalama	0.73072	0.750863	0.732217	0.684561	0.74374	0.793485	**0.833477**

Çizelge 6.10'da GYAopt-aiNet algoritması, grup-2'de yer alan evrimsel kural öğrenme algoritmaları ile kıyaslanmıştır. GYAopt-aiNet, 20 veri kümesinin 14'ünde diğer algoritmalardan daha iyi bir sonuca ulaşmıştır ve genel ortalamada birinci sıradadır.

Çizelge 6.10. Evrimsel Kural Öğrenme Algoritmaları (Grup-2) ile GYAopt-aiNet'in kıyaslanması

	GAssist	**CORE**	**CPSO**	**MPLCS**	**Hider**	**ILGA**	**GYAopt-aiNet**
Bupa	0.613229	0.565544	0.602753	0.635095	0.630898	0.558094	**0.747171**
Ecoli	0.772906	0.652882	0.655823	**0.827481**	0.766964	0.618271	0.798812
Glass	0.644595	0.516147	0.475702	0.681153	0.633861	0.492229	**0.72619**
Haberman	0.737133	0.731649	0.674086	0.702115	0.743728	0.729857	**0.75457**
Iris	0.968889	0.942222	0.915556	0.957778	0.962222	0.928889	**0.993333**
Newthroid	0.93824	0.915584	0.838528	0.916378	0.916667	0.89329	**0.981313**
Pima	0.743504	0.717558	0.672224	0.740018	0.725884	0.731035	**0.798622**
Vehicle	0.666321	0.392806	0.484276	0.708058	0.641839	0.57767	**0.715159**
Appendicitis	0.852121	0.878182	0.781212	0.854848	0.855152	0.839697	**0.930909**
Balance	0.783862	0.648413	0.739128	**0.807331**	0.722137	0.513637	0.764951
Cleveland	0.556973	0.538429	0.52318	0.55364	0.527203	0.525326	**0.63**
Ionosphere	0.921111	0.547989	0.802487	0.905132	0.752302	0.809286	**0.939206**
Magic	0.820513	0.750785	0.722382	**0.82403**	0.769866	0.776885	0.818072
Mammographic	0.826908	0.794779	0.739357	0.834137	0.818072	0.767068	**0.845635**
Page-blocks	0.934321	0.922783	0.906251	**0.942862**	0.933176	0.920393	0.929383
Sonar	0.767698	0.53381	0.53381	0.756905	0.51	0.708492	**0.825238**
Spectfheart	0.75812	0.621842	0.667759	0.775451	0.671462	0.764245	**0.79416**
Tic-tac-toe	0.953414	0.695201	0.710899	**1**	0.699386	0.780128	0.74803
Wisconsine	0.962489	0.940921	0.953616	0.957069	**0.964898**	0.906259	0.958483
Zoo	0.898182	0.943333	0.877576	0.963333	0.953333	0.857576	**0.970303**
Ortalama	0.806026	0.712543	0.71383	0.817141	0.759952	0.734916	**0.833477**

Çizelge 6.11. Karar Ağacı Algoritmaları (Grup-3) ile GYAopt-aiNet'in kıyaslanması

	DT_GA	**C45**	**C45Rules**	**Target**	**GYAopt-aiNet**
Bupa	0.667499	0.693101	0.655118	0.617344	**0.747171**
Ecoli	0.770856	0.794652	0.773708	0.629026	**0.798812**
Glass	0.620532	0.674427	0.680693	0.450092	**0.72619**
Haberman	0.734839	0.731613	0.711935	0.72086	**0.75457**
Iris	0.96	0.96	0.966667	0.86	**0.993333**
Newthroid	0.92114	0.925758	0.903247	0.8386	**0.981313**
Pima	0.733124	0.739721	0.721903	0.704895	**0.798622**
Vehicle	0.715037	**0.74105**	0.667437	0.462176	0.715159
Appendicitis	0.844242	0.840909	0.832727	0.851818	**0.930909**
Balance	0.758175	0.782284	**0.805206**	0.728973	0.764951
Cleveland	0.518736	0.522529	0.558927	0.536705	**0.63**
Ionosphere	0.891587	0.891587	0.883016	0.80619	**0.939206**
Magic	0.817031	**0.819656**	0.80247	0.744285	0.818072
Mammographic	0.838554	0.838554	0.832129	0.827309	**0.845635**
Page-blocks	0.945286	**0.958081**	**0.958081**	0.927666	0.929383
Sonar	0.711587	0.755	0.744127	0.68119	**0.825238**
Spectfheart	0.744255	0.78661	0.790408	0.783048	**0.79416**
Tic-tac-toe	0.78045	**0.845548**	0.844857	0.699386	0.74803
Wisconsine	0.940992	0.942967	0.956614	0.943393	**0.958483**
Zoo	0.900606	0.94	0.94	0.66697	**0.970303**
Ortalama	0.790726	0.809202	0.801463	0.723996	**0.833477**

Çizelge 6.11'de, GYAopt-aiNet algoritması temel sınıflandırma algoritmalarından olan ve grup-3'de yer alan karar ağacı algoritmaları ile kıyaslanmıştır. 5 veri kümesi dışındaki tüm veri kümelerinde en iyi sonucu üretmiştir.

Çizelge 6.12. Yapay Sinir Ağları (YSA), İstatistiksel Yöntemler, SVM ve KNN (Grup-4) ile GYAopt-aiNet'in kıyaslanması

	LVQ	**MLP-BP**	**Kernel**	**LinearLMS**	**SVM**	**KNN**	**GYAopt-aiNet**
Bupa	0.529277	0.514692	0.689222	0.686734	0.701355	0.610805	**0.747171**
Ecoli	0.018627	0.440374	**0.815775**	0.714706	0.75918	0.806952	0.798812
Glass	0.171387	0.351244	0.355661	0.328981	0.625908	**0.736102**	0.72619
Haberman	0.656774	0.663047	0.722258	0.732043	0.72871	0.669677	**0.75457**
Iris	0.4	0.651111	0.913333	0.853333	0.966667	0.933333	**0.993333**
Newthroid	0.536147	0.831457	0.883983	0.865584	0.981602	0.972078	**0.981313**
Pima	0.651081	0.703118	0.703237	0.769677	0.770976	0.699474	**0.798622**
Vehicle	0.251793	0.364622	0.663011	0.763175	**0.79902**	0.700966	0.715159
Appendicitis	0.807273	0.730606	0.801818	0.878182	0.869091	0.819091	**0.930909**
Balance	0.492192	0.639657	0.876677	0.860676	**0.916667**	0.78236	0.764951
Cleveland	0.527548	0.498276	0.438276	0.585977	0.595862	0.532184	**0.63**
Ionosphere	0.654365	0.700767	0.641032	0.874921	0.866111	0.854762	**0.939206**
Magic	0.624581	0.731336	0.773899	0.795985	0.800179	0.795478	**0.818072**
Mammographic	0.514056	0.733735	0.706024	0.810843	0.824096	0.739759	**0.845635**
Page-blocks	0.886779	0.760988	0.928788	0.93798	**0.958081**	0.936162	0.929383
Sonar	0.536984	0.629524	0.53381	0.725238	0.77381	**0.864524**	0.825238
Spectfheart	**0.79416**	0.510304	0.734188	0.778205	0.782621	0.700285	**0.79416**
Tic-tac-toe	0.634247	0.606184	0.653443	0.710921	0.701502	0.73068	**0.74803**
Wisconsine	0.650064	0.961445	0.907758	0.960507	**0.964876**	0.961978	0.958483
Zoo	0.926667	0.490606	0.405455	0.94	0.96	0.92	**0.970303**
Ortalama	0.5632	0.625655	0.707382	0.778683	0.817316	0.788333	**0.833477**

Çizelge 6.12'de, GYAopt-aiNet algoritması, sınıflandırma için kullanılabilen muhtelif algoritmalar ile kıyaslanmıştır. Grup-4'de yer alan bu algoritmalar arasında yapay sinir ağı algoritmaları olan LVQ ve MLP-BP, Destek vektör makineleri (SVM) ve çeşitli istatistiksel yöntemler bulunmaktadır. Bu sonuçlar da incelendiğinde 13 veri kümesi için GYAopt-aiNet algoritmasının diğer algoritmalardan daha verimli sonuçlar ürettiği görülmektedir.

Çizelge 6.13. Grup-1 ve Grup-2 için Wilcoxon işaret-sıra istatistikî testinin bulduğu *p* değerleri

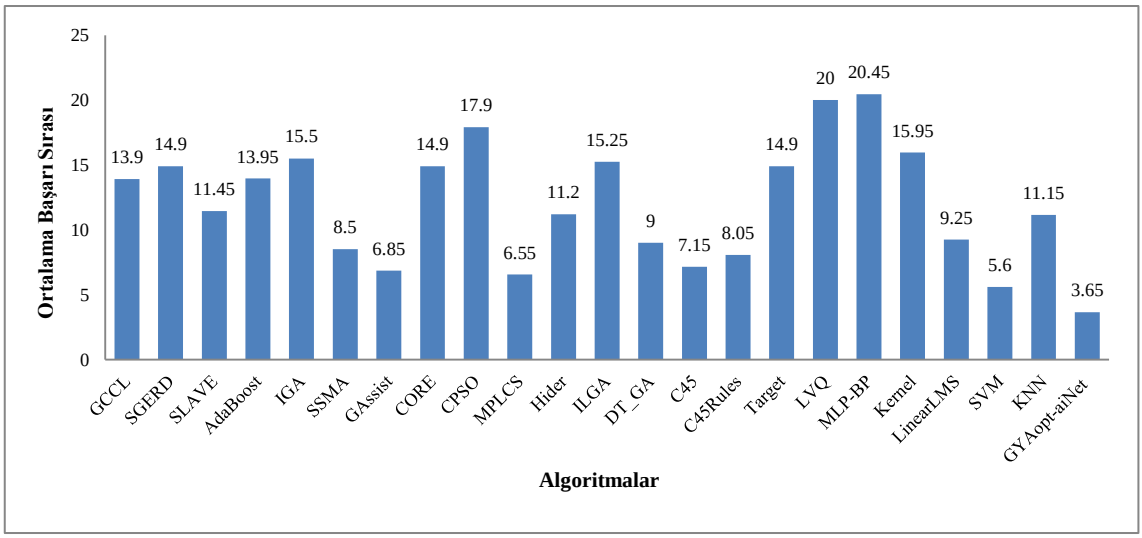
Grup-1	***p* değeri**	***Grup- 2***	***p* değeri**
GYAopt-aiNet ile GCCL	0.0000	GYAopt-aiNet ile GASSIT	0.0072
GYAopt-aiNet ile SGERD	0.0001	GYAopt-aiNet ile CORE	0.0001
GYAopt-aiNet ile SLAVE	0.0006	GYAopt-aiNet ile CPSO	0.0001
GYAopt-aiNet ile AdaBoost	0.0003	GYAopt-aiNet ile MPLCS	0.0304
GYAopt-aiNet ile IGA	0.0002	GYAopt-aiNet ile HIDER	0.0001
GYAopt-aiNet ile SSMA	0.0028	GYAopt-aiNet ile ILGA	0.0002

Çizelge 6.14. Grup-3 ve Grup-4 için Wilcoxon işaret-sıra istatistikî testinin bulduğu *p* değerleri

Grup-3	***p* değeri**	***Grup-4***	***p* değeri**
GYAopt-aiNet ile DT-GA	0.0007	GYAopt-aiNet ile LVQ	0.0001
GYAopt-aiNet ile C4.5	0.0206	GYAopt-aiNet ile MLP-BP	0.0001
GYAopt-aiNet ile C 4.5 Rules	0.0090	GYAopt-aiNet ile KERNEL	0.0008
GYAopt-aiNet ile TARGET	0.0001	GYAopt-aiNet ile LINEAR-LMS	0.0051
		GYAopt-aiNet ile SVM	0.0438
		GYAopt-aiNet ile KNN	0.0045

Genel olarak sonuçlara bakıldığında, GYAopt-aiNet algoritmasının her grupta en yüksek ortalama sınıflandırma doğruluğuna sahip olduğunu görülmektedir. Daha ayrıntılı bir inceleme için de önerilen algoritma, her bir gruptaki algoritmalar ile de tek tek istatistiki olarak karşılaştırılmıştır. Örnek bir H_0 hipotezi, GYAopt-aiNet ile GCCL algoritmaları arasında *"H_0:GYAopt-aiNet ile GCCL algoritmaları arasında %95 önem seviyesinde istatistikî olarak anlamlı bir fark yoktur. "* şeklinde kurulmuştur. Benzer H_0 hipotezleri GYAopt-aiNet ile diğer algoritmalar arasında da kurulmuştur ve her bir hipotez Wilcoxon işaret-sıra istatistikî testi kullanılarak test edilmiştir. Test sonucu elde edilen *p* değerleri, gruplara göre gruplanmış ve Çizelge 6.13 ve Çizelge 6.14'de verilmiştir. Bilindiği üzere *p* değeri %95 önem seviyesi için 0.05'den küçük ise, önerilen hipotez testi reddedilmekteydi. Çizelgelerdeki *p* değerlerinin hepsi 0.05'den küçüktür. Bu durumda tüm hipotezler reddedilmiş demektir. Yani GYAopt-aiNet algoritması ile diğer algoritmalar arasında anlamlı bir fark vardır. Peki, bu algoritmalardan hangi daha iyidir? Bu soruyu cevaplamada, algoritmaların ortalama başarı sırası (rank) değerlerinden ve kutu diyagramlarından faydalanılacaktır.

Her bir algoritmanın ortalama başarı sırası belirlenirken izlenecek yol şu şekildedir: *N* adet algoritmanın, *M* adet veri kümesi üzerinde test edildiğini varsayalım. Algoritmalar sırası ile ilk veri kümelerinde çalıştırılır. Veri kümesinde en iyi sonucu veren algoritmanın sıra değeri 1 olarak atanır. İkinci en iyi sonucu verenin sıra değeri 2 olur ve bu tüm algoritmalar için devam ettirilir. En kötü sonucu veren algoritmanın sıra değeri *N* olarak atanır. Bu işlem geriye kalan (*M-1)* için tekrar ettirilir. Böylece her bir algoritma, her veri kümesi için bir sıra değerine sahip olur. Bu sıra değerlerinin ortalaması da, algoritmanın başarı sıra değerini verir. Başarı sıra değerinin düşük olması istenilen bir durumdur. Başarı sıra değeri ne kadar küçük olursa, algoritma o kadar çok veri kümesinde en yüksek başarıyı elde etmiş demektir. Bu deneysel çalışmada kullanılan algoritmaların başarı sıraları Şekil 6.14'deki grafikte verildiği gibidir.

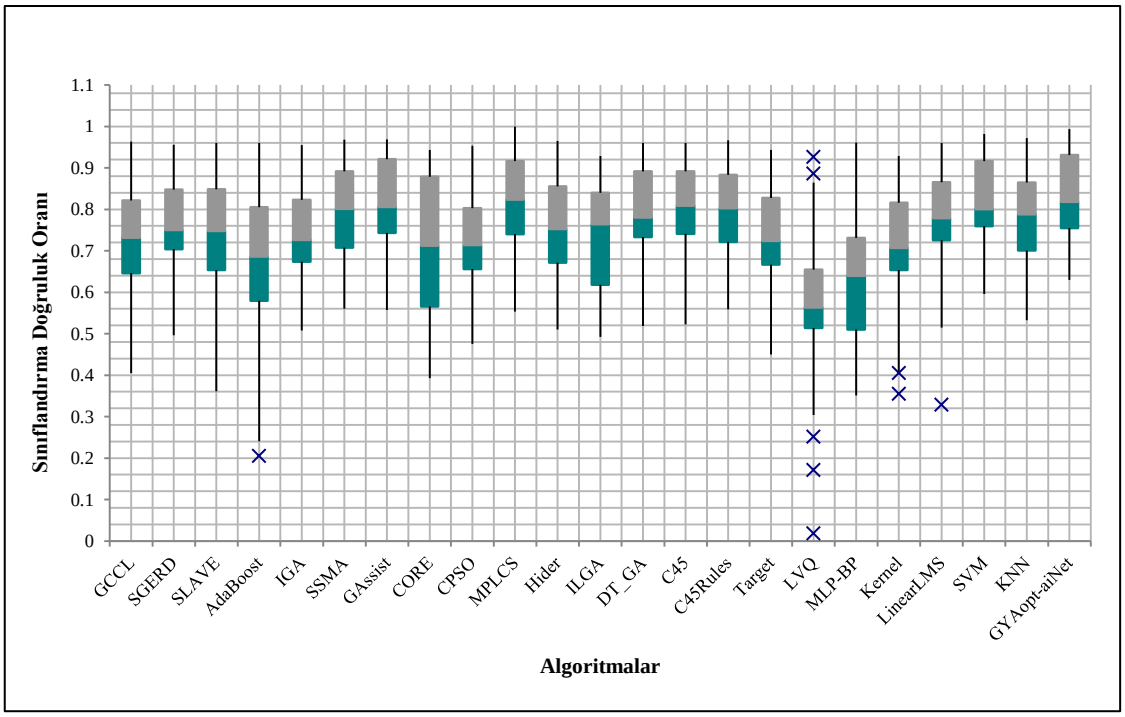


Şekil 6.14. Algoritmaların veri kümeleri üzerindeki ortalama başarı sıraları

Bu grafik, her bir algoritmanın tüm veri kümeleri üzerindeki başarısının bir özetini göstermesi açısından fikir vericidir. Grafik incelendiğinde, GYAopt-aiNet algoritmasının en düşük ortalama başarı sırasına sahip olduğu görülmektedir.

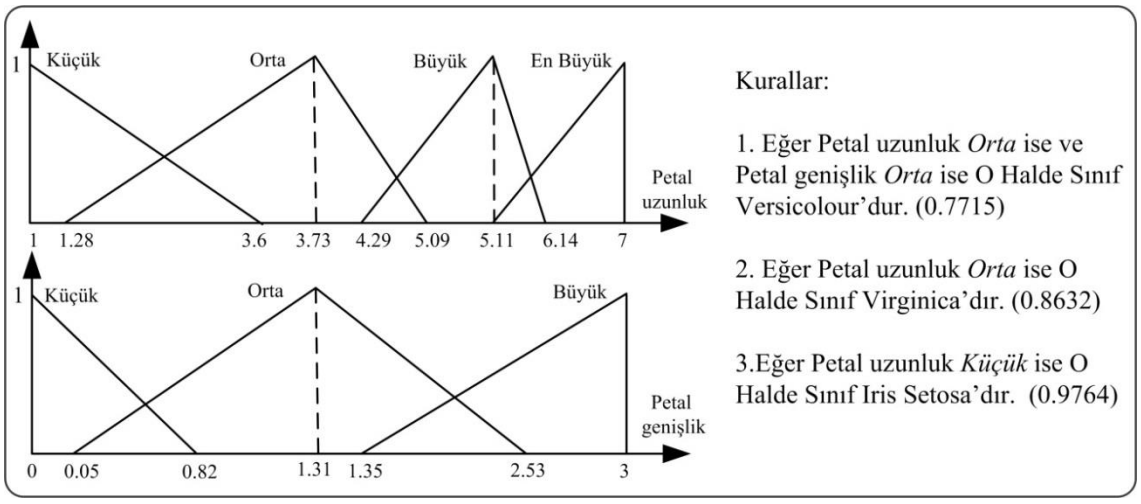
Algoritmaları kıyaslamak için kullanılan bir diğer yöntem ise kutu diyagramlarıdır. Bu yöntem ile algoritmaların sınıflandırma başarıları ile ilgili tanımlayıcı istatistiksel bilgiler görselleştirilerek, bu bilgilerin yorumlanması kolaylaştırılır. Grafik üzerinde, algoritmaların veri kümelerinden elde ettiği sınıflandırma başarılarının en küçük değeri, ortanca değeri ile en büyük değeri arasındaki dağılımının yanı sıra, uç değerli noktalarda gösterilir. Bu da bize algoritmanın istikrarı hakkında bilgi verir.

Şekil 6.15'de bu deneysel çalışmada kullanılan sınıflandırma algoritmalarının kutu diyagramları gösterilmektedir. Grafik incelendiğinde, GYAopt-aiNet algoritmasının maksimum ve minimum doğruluk değerleri arasındaki farkın diğer algoritmalarınkinden daha düşük olduğu görülmektedir. Buna ek olarak, GYAopt-aiNet'in ortanca değeri, MPLCS algoritması dışındaki tüm algoritmalardan daha yüksektir. Ancak ortanca değerin üstünde MPLCS algoritmasındakinden daha çok noktaya sahiptir. Ayrıca hiç uç noktası yoktur. İstikrarlı bir dağılıma sahiptir.



Şekil 6.15. Algoritmaların sınıflandırma doğruluk oranlarına göre kutu diyagramları

Sonuç olarak, GYAopt-aiNet algoritması, bulanık sınıflandırıcı tasarım problemine tatbik edildiğinde diğer algoritmalara göre makul, kabul edilebilir ve başarılı sonuçlar ürettiği söylenebilir.



Şekil 6.16 İris veri kümesi için bulanık sınıflandırıcı çözümü

Örnek olarak GYAopt-aiNet'in iris veri kümesi için olan bulanık sınıflandırıcı çözümü Şekil 6.16'da verilmiştir. Geriye kalan veri kümelerine ait çözümler Ek-1 ve Ek-2'de verilmiştir.

7.SONUÇLAR VE ÖNERİLER

7.1. Sonuçlar

Gerçek dünyada karşılaşılan günümüz problemlerinin çoğu karmaşık, doğrusal olmayan, içerisinde eksik verilerin ve belirsizliklerin bulunduğu durumlarla ilgilidir. Bu tarz problemlerin çözümü için tasarlanması gereken sistemlerde, bahsedilen eksik ve belirsizliklerin giderilmesinde kullanılacak olan gerekli verilere ulaşılması genellikle zahmetli ve/veya maliyetli bir süreçtir. Tüm verilere ulaşıldığında dahi, problemin yapısı gereği sayısal olarak modellenmesi güç olabilir. Böyle durumlarda, insan tecrübe ve algıların sözel veriler olarak sisteme dâhil edilmesine imkân veren bulanık sistemler alternatif bir modelleme şekli olarak ortaya çıkmış ve sistem tasarımında kullanımı giderek yaygınlaşmıştır.

Sözel verilerin derecelendirilerek modellenmesine imkân veren ve böylece insan düşence sistemini taklit edebilen bulanık sistemlerin en büyük dezavantajı öğrenme yeteneğinin bulunmamasıdır. Ne problem çözme sırasında ne de sonrasında sistem, temel bileşenleri olan üyelik fonksiyonlarını ve kurallarını öğrenemez. Bundan dolayı bulanık sisteme ait parametreler, sistem tasarlanmadan önce tespit edilmelidir. Bu parametreler mantıksal, yapısal, bağıntısal ve işlemsel olarak sınıflandırılabilir. *Mantıksal parametreler* ile çıkarım mekanizmasının türü, çıkarım esnasında hangi bulanık işlemcilerin kullanılacağı, üyelik fonksiyonlarının tipi ve durulama yöntemi, *yapısal parametreler* ile sistemde yer alacak ilişkili değişkenlerin, üyelik fonksiyonlarının ve bulanık kuralların sayıları; *bağlantısal parametreler* ile kuralların yapısı; *işlemsel parametreler* ile de üyelik fonksiyonlarına ait değerlerin ne olacağı belirlenir. Parametreler belirlenirken konu üzerinde uzmanlaşmış kişilerin fikir ve tecrübelerinden faydalanılması gerekir ama bir uzman görüşüne ulaşmak her zaman mümkün olmayabilir, mümkün olduğunda da uzmandan çok büyük ve dinamik yapıda olan bir veri kümesinden bu parametreleri doğru şekilde tespit etmesini beklemek zordur. Örneğin sistemin üyelik fonksiyonlarına ait değerlerin tutulduğu işlemsel parametrelerin belirlenmesi esnasında en sık kullanılan yöntem deneme yanılma yöntemidir ki bu da uzun zaman alabilmekte ve birçok test yapmayı gerektirmektedir. Bu yüzden öğrenme yeteneği olan algoritmalar kullanılarak, gerçek dünya verilerinden bulanık sistem parametrelerinin tespit edilmesi araştırmacıların ilgisini çeken bir konu olmuştur.

Bu tez çalışmasında, parametrelerin en uygun değerleri tespit edilirken, bulanık mantığın çekirdeğini oluşturan *derecelendirme* ve *taneciklendirme* kavramlarından faydalanılmıştır. Bu kavramlar bulanık mantığı diğer mantık sistemlerinden ayıran başlıca özellikleridir. Bulanıklık, bir olayın olup olmadığını değil, hangi dereceye kadar olduğunu ölçmektedir. Taneciklendirme de, değişkenlerin benzerliklerine, farklılıklarına, yakınlık veya işlevselliklerine bakılarak parçacıklara bölünmesidir. İkisi birlikte dilsel ifade kavramını oluştururlar. Bu tez çalışmasında bulanık mantığa ait bu iki kavram ve bulanık bir sisteme ait kurallar, gerçek dünya verilerinden öğrenilerek özellikle sınıflandırma problemleri için yorumlanabilirliği yüksek bulanık sistemler tasarlanmıştır.

Tez çalışması iki aşamadan oluşmuştur. İlk aşamasında, üyelik fonksiyonlarına ait en uygun taban uzunlukları optimize edilerek işlemsel parametrelerin en uygun değerleri tespit edilmiş böylece derecelendirme işlemi gerçekleştirmiştir. Bulanık bir sisteme ait olan diğer parametrelerin bir uzman tarafından önceden belirlendiği kabul edilmiştir.

Önerilen ilk yöntem, üyelik fonksiyonunun taban uzunluğu değiştikçe, bireylerin o kümeye olan aitlik derecesinin de değiştiği fikri üzerine bina edilmiş, bu sayede bir bireyin, bulanık kümeye olan aitliğinin en uygun derecesi tespit edilmiştir. Optimizasyon işlemi bir yapay bağışılık algoritması olan Klonal seçme algoritması (CLONALG) kullanılarak gerçekleştirilmiş ve sonuçlar genetik algoritma kullanılarak elde edilen sonuçlar ile kıyaslanmıştır. Algortmaların bu sorunun çözümündeki verimliliğinin araştırılması için gerçekleştirilen deneysel çalışmalarda tek girişli/tek çıkışlı (TG-Ç) ve çok girişli/çok çıkışlı (ÇG-Ç) olmak üzere iki farklı bulanık sistem kullanılmıştır. Deneyler 20 farklı grup için tekrarlanmıştır. Her bir grup 10 farklı popülasyon ve her bir popülasyon 10 adet birey içermektedir. Her bir popülasyondaki en yüksek duyarlılığa sahip antikor çözüm olarak kabul edilmiştir. Algoritmalar gruplardaki her bir popülasyon için ayrı ayrı çalıştırılmış ve sonuçlar grup grup kıyaslanmıştır. Kıyaslama için Wilcoxon işaret-sıra istatistiki testi kullanılmış ve hem TG-Ç hem de ÇG-Ç bulanık sistemlere ait işlemsel parametrelerin en uygun değerlerinin tespitinde CLONALG ile genetik algoritma kullanılması arasında %95 önem seviyesinde istatistiksel olarak anlamlı bir fark olduğu sonucuna varılmıştır. Hangi algoritmanın daha iyi olduğunun anlaşılması için belirleyici istatistiklere bakılmış

ve CLONALG algoritmasının her iki sistem içinde daha istikrarlı sonuçlar ürettiği ve kullanımın daha avantajlı olacağı çıkarımı yapılmıştır.

Bulanık bir sisteme ait mantıksal, yapısal ve bağlantısal parametrelerin belirlenmesinde bir uzmanın yardımına ihtiyaç duyulması, derecelendirme işlemi için önerilen bu yöntemin eksikliğidir. Bundan dolayı, tezin ikinci aşamasında mantıksal parametreler ile sistemde kullanılacak olan ilişkili değişkenlerin önceden belirlendiği, diğer parametrelerin de gerçek dünya verilerinden öğrenildiği yeni bir yaklaşım sunulmuştur.

Öncelikle, her bir niteliğin sahip olacağı ÜF sayısını ve ÜF'larının başlangıç yerleşim yerlerini bulan **S**iluet küme **D**oğrulama indeksi ve **k**-**o**rtalama tabanlı (SDko) yeni bir ön işlem algoritması önerilmiştir. Böylece bulanık taneciklerin başlangıç değerleri uzman yardımı olmaksızın belirlenmiş ve arama uzayı daraltılarak doğru sonuca daha çabuk ulaşılması sağlanmıştır.

Bu tez çalışması ile SDko önişlem adımı ile tespit edilen bulanık tanecikleri derecelendirmesinin yanı sıra dinamik şekilde kural tabanını da optimize edebilen GYAopt-aiNet algoritması da literatüre kazandırılmıştır. GYAopt-aiNet bir yapay bağışıklık algoritması olan opt-aiNet'in baskılama ve hiper mutasyon mekanizmalarında değişiklikler yapılmış şekline memetik algoritmalarda yer alan yerel arama stratejisinin eklenmesi ile elde edilmiştir. Bu sayede opt-aiNet algoritmasının genel arama yeteneği, memetik algoritmanın yerel arama stratejisi ile tamamlanmıştır.

İlk olarak yapılan değişikliklerin etkisini ölçmek için GYAopt-aiNet algoritması ile orijinal opt-aiNet algoritmasının kıyaslaması yapılmıştır. Kıyaslama için California, Irvine Üniversitesinin makine öğrenmesi veri kümesi havuzundan (Blake and Merz, 1998) alınan herkese açık 20 farklı sınıflandırma veri kümesi kullanılmıştır. Her iki yöntem ile de bu veri kümeleri için bulanık kural tabanlı sınıflandırıcı sistemleri tasarlanmıştır. Sonuçlar, üç kez tekrarlanan 10 çapraz doğrulama yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak bulunmuştur. Buna ek olarak, GYAopt-aiNet algoritmasından önce SDko önişlem algoritması kullanıldığından, kıyaslamada eşitliği sağlamak için opt-aiNet algoritmasından önce de önişlem algoritmasının kullanıldığı durumda incelenmiştir. Sonuç olarak, GYAopt-aiNet algoritmasının, hem orijinal opt-aiNet algoritmasından hem de SDko algoritmasından sonra çalıştırılan opt-aiNet algoritmasından daha iyi ve daha tutarlı sonuçlar ürettiği gözlemlenmiştir.

Deneysel çalışmanın devamında, listesi Çizelge 2.3'de verilen diğer sınıflandırıcı algoritmalar ile GYAopt-aiNet algoritması kıyaslanmıştır. Kıyaslamak için kullanılan sınıflandırma yöntemlerinin sonuçları, KEEL (http://www.keel.es/) isminde ticari olmayan açık kaynak kodlu bir yazılım aracı kullanılarak elde edilmiştir. Sonuçlar, daha önceki bölümde olduğu gibi üç kez tekrarlanan 10 çapraz doğrulama yönteminden gelen sınıflandırma doğruluklarının ortalaması alınarak bulunmuştur. Kıyaslamada kullanılan algoritmalar tasnif edilmiş ve türlerine göre, evrimsel bir bulanık kural öğrenme algoritmaları, evrimsel kural öğrenme algoritmaları, karar ağacı algoritmaları ve sınıflandırmada sıklıkla kullanılan diğer yöntemler (yapay sinir ağları, istatistiksel yöntemler, SVM ve KNN) olmak üzere 4 ana başlık altında toplanmıştır.

İlk olarak, grup-1'de yer alan evrimsel bulanık kural öğrenme algoritmaları ile GYAopt-aiNet algoritması kıyaslanmıştır. GYAopt-aiNet algoritması da evrimsel bir bulanık kural öğrenme algoritması olduğundan dolayı bu gruptaki kıyaslama sonuçları önemlidir. Sonuçlara göre, GYAopt-aiNet algoritması 20 veri kümesinin 15'inde en yüksek ortalama sınıflandırma doğruluğuna sahiptir. Algoritmaların tüm veri kümelerinde elde ettiği sınıflandırma doğruluklarının genel ortalamaları incelendiğinde, GYAopt-aiNet algoritması 0.833477 değeri ile ilk sırada yer aldığı görülmektedir. Diğer gruplar içinde GYAopt-aiNet'in tutarlı ve iyi sonuçlar ürettiği gözlemlenmiştir.

Bunlara ek olarak, daha ayrıntılı bir inceleme için GYAopt-aiNet algoritması, her bir gruptaki algoritmalar ile de tek tek istatistiki olarak karşılaştırılmış ve her bir H_0 hipotezi Wilcoxon işaret-sıra istatistikî testi kullanılarak test edilmiştir. Test sonuçlarına göre GYAopt-aiNet algoritması ile diğer algoritmalar arasında istatistiksel olarak %95 önem seviyesinde anlamlı bir fark olduğu tespit edilmiştir. Bu algoritmalardan hangisinin daha iyi olduğu sorusunu cevaplarken de algoritmaların ortalama başarı sırası (rank) değerlerinden ve kutu diyagramlarından faydalanılmıştır.

Başarı sırası değerleri, her bir algoritmanın tüm veri kümeleri üzerindeki başarısının bir özetini göstermesi açısından fikir vericidir. Başarı sırası sonuçları incelendiğinde, GYAopt-aiNet algoritmasının en düşük ortalama başarı sırasına sahip olduğu görülmüştür. Başarı sırasının düşük olması, eğitim kümelerinin çoğunda en iyi sınıflandırma başarısının elde edildiğinin bir göstergesidir.

Kutu diyagramları ile de algoritmaların sınıflandırma başarıları ile ilgili tanımlayıcı istatistiksel bilgiler görselleştirilerek, bu bilgilerin yorumlanması kolaylaştırılmıştır. Bu deneysel çalışmada kullanılan sınıflandırma algoritmalarının kutu

diyagramları incelendiğinde, GYAopt-aiNet algoritmasının maksimum ve minimum doğruluk değerleri arasındaki farkın diğer algoritmalarınkinden daha düşük olduğu görülmüştür. Buna ek olarak, GYAopt-aiNet'in ortanca değeri, MPLCS algoritması dışındaki tüm algoritmalardan daha yüksektir. Ancak, ortanca değerin üstünde MPLCS algoritmasındakinden daha çok noktaya sahip olup hiç uç noktası yoktur. Aynı zamanda istikrarlı bir dağılıma sahiptir.

Sonuç olarak, GYAopt-aiNet algoritması, bulanık sınıflandırıcı tasarım problemine tatbik edildiğinde diğer algoritmalara göre makul, kabul edilebilir ve başarılı sonuçlar üretmiş, verimli bir algoritmadır. Ayrıca SDko önişlem algoritması ve GYAop-aiNet'in içinde barındırdığı adımlar sayesinde, yorumlanabilirlikleri yüksek sistemlerin tasarlanabilmiştir. Şöyle ki, ön işlem algoritması ile üyelik fonksiyonlarının tam, ayrıklaştırılabilir, tamamlayıcı olması ve üyelik fonksiyonunun sayısının makul seviyede seçilmesi sağlanmıştır. Niteliklerin ve değişkenlerin sayısı başlangıç popülasyonu üretilirken izlenen yöntem ile kontrol altında tutulmuştur. Ishibuchi ve ark. (1997) tarafından önerilen algoritmanın kullanımı ile de, çelişkili kuralların veri tabanına dâhil olması baştan engellenmiştir. Böylece kural tabanının tutarlılığı temin edilmiş ve tasarlanan sistemin yorumlanabilirliği mümkün oldukça arttırılmıştır.

7.2. Öneriler

Bulanık sistemlerin başarısını etkileyen faktörlerin başında değişkenlere ait bulanık taneciklerin sayısının ve sınırlarının düzgün bir şekilde belirlenmesi gelmektedir. Bu çalışmada k-ortalama algoritması ve siluet küme doğrulama indeksinden faydalanılarak bulanık taneciklerin başlangıç değerleri elde edilmiş ve en uygun değerleri GYAopt-aiNet algoritması ile öğrenilmiştir. İlerleyen çalışmalarda bulanık taneciklerin en uygun değerlerini, eğitim verilerinden doğrudan tespit eden bir önişlem algoritması kullanılarak, GYAopt-aiNet'in arama uzayı azaltılıp çalışma süresi kısaltılabilir.

Önerilen GYAopt-aiNet algoritması, özel bir bulanık sistem türü olan bulanık sınıflandırıcı tasarımında kullanılmıştır. Aynı yöntem, uygun değişikliler yapılarak regresyon problemlerine de uygulanabilir. Bu durumda, çıkış olarak verinin ait olduğu sınıf yerine, çıkış noktasının/noktalarının tahmini yapılacağından özellikle durulaştırma adımında yapılacak iyileştirmeler sistemin başarısını etkileyecektir.

KAYNAKLAR

Abe, S., and Thawonmas, R., 1997, A fuzzy classifier with ellipsoidal regions, *IEEE Transactions on Fuzzy Systems*, 5(3), 358–368.

Acılar A.M. and Arslan A., 2011, Optimization of multiple input–output fuzzy membership functions using clonal selection algorithm, *Expert Systems with Applications*, 38, 1374-1381.

Akay R., 2006, Memetik algoritma kullanarak PID denetleyici tasarımı, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü,* Kayseri, 16.

Ang, J.H., Tan, K.C., and Mamun, A.A., 2010, An evolutionary memetic algorithm for rule extraction, *Expert Systems with Applications*, 37(2), 1302-1315.

Ang, K.K. and Quek, C., 2012, Supervised pseudo pelf-evolving cerebellar algorithm for generating fuzzy membership functions, *Expert Systems with Applications*, 39, 2279-2287.

Aguilar-Ruiz, J.S., Riquelme, J.C., and Toro M., 2003, Evolutionary learning of hierarchical decision rules, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 33(2), 324-331.

Alcala, R., Alcala-Fdez, J. and Herrera, F., 2007, A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection, *IEEE Transactions on Fuzzy Systems*, 15(4), 616–635.

Alcala, R., Nojima Y., Herrera F. and Ishibuchi H., 2011, Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions, *Soft Computing*, 15, 2303–2318.

Arslan, A., and Kaya, M., 2001, Determination of fuzzy logic membership functions using genetic algorithms, *Fuzzy Sets and Systems*, 118, 297–306.

Bacardit, J. and Garrell, J.M., 2003, Evolving multiple discretizations with adaptive intervals for a pittsburgh rule-based learning classifier system, *In Genetic and Evolutionary Computation Conference, GECCO 2003*, Lecture Notes on Computer Science 2724, Illinous, USA, 1818-1831.

Bacardit, J. and Krasnogor, N., 2009, Performance and efficiency of memetic pittsburgh learning classifier systems, *Evolutionary Computation Journal*, 17(3), 307-342.

Bağiş, A., 2003, Determining fuzzy membership functions with tabu search – an application to control. *Fuzzy Sets and Systems*, 139, 209-225.

Bandyopadhyay S., 2004, An automatic shape independent clustering techniques, *Pattern Recognition*, 33–45.

Baykal N. ve Beyan T., 2004a, Bulanık mantık ilke ve temelleri, *Bıçaklar*, 413 sayfa.

Baykal, N. ve Beyan T., 2004b, Bulanık mantık uzman sistemler ve denetleyiciler, *Bıçaklar*, 509 sayfa.

Bezdek, J.C. and Kuncheva. L.I.,2001, Nearest prototype classifier designs: An experimental study, *International Journal of Intelligent Systems*, 16(12), 1445-1473.

Blake, C., and Merz, C., 1998, UCI repository of machine learning databases, http://www.ics.uci.edu /mlearn/MLRepository.html

Black, M., 1937 Vagueness: An exercise in logical analysis, *Philosophy of Science*, 4, 427–455.

Cantu-Paz, E. and Kamath, C., 2003, Inducing oblique decision trees with evolutionary algorithms, *IEEE Transactions on Evolutionary Computation*, 7(1), 54-68.

Cerrada, M., Aguilar, J., Colina, E. and Titli, A., 2005, Dynamical membership functions: an approach for adaptive fuzzy modeling, *Fuzzy Sets and Systems*, 152, 513–533.

Chang, X., and Lilly., J.H., 2004,Evolutionary design of a fuzzy classifier from data, IEEE Transactions on Systems Man and Cybernetics 34 (4) (2004) 1894–1906.

Chen, C.H., Hong, T.P. and Tseng V., 2009, An improved approach to find membership functions and multiple minimum supports in fuzzy data mining, *Expert Systems with Applications*, 36, 10016-10024.

Cheng, H.D. and Lui, Y.M., 1997, Automatic bandwidth selection of fuzzy membership functions, *Information Sciences*, 103, 1-27.

Cortes, C. and Vapnik,V., 1995, Support vector networks, Machine Learning, 20, 273-297.

Çelikyılmaz A. and Türkşen B., 2009, Modeling uncertainty with fuzzy logic with recent theory and applications, Studies in Fuzziness and Soft Computing, *Springer-Verlag*, Berlin Heidelberg, 13-17.

Dawkins, R., 1976, The selfish gene, *Oxford University Press*, Oxford.

de Castro, L. N. and Timmis, J.I., 2002(a), Artificial immune systems: a new computational intelligence approach, *Springer -Verlag*, London, 4-108.

de Castro, L. N. and Timmis, J.I., 2002(b), An artificial immune network for multimodal function optimization, *Proc. of the 2002 IEEE Congress on Evol. Computation, CEC*, vol. 2, 699–704.

de Castro, L. N. and Von Zuben, F. J., 1999, Artificial immune systems: Part I – Basic theory and applications, Techinical Report – TR - DCA 01/99, 8-38.

de Castro, L. N. and Von Zuben, F. J., 2001, aiNet: An artificial immune network for data analysis, Data Mining: A Heuristic Approach, section: XII, H. A. Abbass, R. A. Sarker, and C. S. Newton (eds.), *Idea Group Publishing*, USA, 231-259.

de Castro, L. N. and Von Zuben, F. J., 2002, Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, 6(3), 239-251.

Derrac, J., García, S. and Herrera, F., 2010, IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule, *Pattern Recognition*, 43(6), 2082–2105.

Del Jesús, M. J., Hoffmann, F., Junco, L. and Sánchez, L., 2004, Induction of fuzzy-rule-based classifiers with evolutionary boosting algorithms, *IEEE Transactions on Fuzzy Systems*, 12(3), 296-308.

Derbel, I., Hachani, N. and Ounelli H., 2008, Membership functions generation based on density function, *International Conference on Computational Intelligence and Security-CIS'08,* Suzhou, China, 96-101.

Dou, W.B., Ren, Y., Chen, Y.P., Ruan, S., Bloyet, D. and Constans, J.M., 2005, Histogram-based generation method of membership function for extracting features of brain tissues on MRI images, *Lecture Notes In Artificial Intelligence*, 3613, 189-194.

Dutta S., 1993, Fuzzy logic applications: Technological and strategic issues, *IEEE Transactions on Engineering Management* , 40(3), 237-254.

Elmas, Ç., 2003, Bulanık mantık denetleyiciler (Kuram, Uygulama, Sinirsel Bulanık Mantık), *Seçkin Yayıncılık*, Ankara, 39-41.

Gacto, M.J., Alcalá R. and Herrera F., 2011, Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures, *Information Sciences*, 181,4340-4360.

García-Pedrajas, N. and Fyfe, C., 2008, Construction of classifier ensembles by means of artificial immune systems, *Journal of Heuristics*, 14, 285–310.

Garcia, S., Cano, J.R. and Herrera, F.,2008, A Memetic algorithm for evolutionary prototype selection: A scaling up approach, *Pattern Recognition*, 41 (8), 2693-2709.

Garg, P., 2009, A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm, *International Journal of Network Security & Its Applications*, 1(1), 34-42.

Goldberg, E., 1989, Genetic algorithms in search, optimization, and machine learning, *Addison-Wesley*.

Gonzalez, A. and Perez, R., 1999, SLAVE: A genetic learning system based on an iterative approach, *IEEE Transactions on Fuzzy Systems*, 7(2), 176-191.

Gray J. B. and Fan, G., 2008, Classification tree analysis using TARGET, *Computational Statistics & Data Analysis*, 52(3), 1362-1372

Guan S. U. and Zhu, F., 2005, An incremental approach to genetic algorithms-based classification, *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, 35(2), 227-239.

Hachani, N. and Ounelli, H., 2007, Improving cluster method quality by validity indices, *Association for the Advancement of Artificial Intelligence (www.aaai.org)*, 479-483.

Han, J. and Kamber, M., 2001, Data Mining: concepts and techniques, *Morgan Kaufman Publishers*, 550 pages.

Hitachi Ltd., 1987, Fuzzy control ensures a smooth ride, *Age of Tomorrow,Japan*,12-19.

Ho, S.Y., Liu C.C. and Liu, S., Design of an optimal nearest neighbour classifier using an intelligent genetic algorithm, Pattern Recognition Letters, 23, 1495-1503

Holmblad, P. and Østergaard, J.-J., 1982, Control of a cement kiln by fuzzy logic, Gupta M. and Sanchez E. (eds.), *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, 398-399.

Holland, J. H., 1975, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence, *University of Michigan Press*, Ann Arbor.

Ishibuchi, H., Murata, T. and Turksen, I.B., 1997, Single-objective and two objective genetic algorithms for selecting linguistic rules for pattern classification problems, *Fuzzy Sets and Systems*, 89, 135–149.

Ishibuchi, H., Nakashima, T. and Murata, T., 1999, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(5), 601-618.

Jerne, N.K., 1974, Towards a network theory of the immune system, *Annales d'immunologie*, 125(1-2), 373–389.

Karr, C.L., 1991, Design of an adaptive fuzzy controller using a Genetic algorithm, *Proc. of the 4th Intl. Conf. on Genetic Algorithms*, 450-457.

Kartal M.,2006, Bilimsel araştırmalarda hipotez testi, *Nobel Yayıncılık*, Ankara, 233.

Kaya, M., Karci, A. and Arslan, A., 2001, Determination of membership functions in multiple input-output fuzzy systems by genetic algorithms. *Proceeding of 3rd International Symposium on Intelligent Manufactiring System*, Sakarya,Turkiye.

KEEL yazılım aracı, http://www.keel.es/, [erişim tarihi: 23.02.2011].

Lei, Z. and Ren-hou, L., 2008, Designing of classifiers based on immune principles and fuzzy rules, *Information Sciences*, 178, 1836-1847.

Li, M. and Wang, Z., 2009, A hybrid coevolutionary algorithm for designing fuzzy classifiers, *Information Sciences*, 179, 1970-1983.

Lin, C.T. and Lee, C.S.G, 1991, Neural-Network-Based fuzzy logic control and decision system, *IEEE Transactions On Computers*, 40(12), 1320-1336.

Mamdani, E. H., 1974, Application of fuzzy algorithms for control of simple dynamic plant, *Proc. Inst. Elec. Eng.*, 121, 1585-1588.

Mamdani, E.H. and Assilian S., 1974, A case sudy on the application of Fuzzy set theory of automatic control, in Proc. IFAC stochastic control symp., Budapest, Hungary.

Mansoori, E.G., Zolghadri, M.J. and Katebi, S.D., 2007, A weighting function for improving fuzzy classification systems performance, *Fuzzy Sets and Systems*, 158 (5), 583–591.

Mansoori, E. G., Zolghadri, M. J. and Katebi, S. D., 2008, SGERD: A Steady-State Genetic algorithm for extracting fuzzy classification rules from data, *IEEE Transactions on Fuzzy Systems*, 16(4), 1061-1071.

MacQueen, J. B., 1967, Some methods for classification and analysis of multivariate observations, *1. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 281–297.

McLachlan, G. J.,2004, Discriminant analysis and statistical pattern recognition, Wiley Series in Probability and Mathematical Statistics.

Medasani, S., Kim, J., and Krishnapuram, R., 1998, An overview of membership function generation techniques for pattern recognition, *International Journal of Approximate Reasoning*, 19(3–4), 391–417.

Meredith, D.L., Karr, C.L. and Krishna Kumar, K., 1992, The use of genetic algorithms in the design of fuzzy logic controllers, *3rd Workshop on Neural Networks: Academic/Industrial/Defence, WNN'92*, pp. 1105-1109.

Miller, G.A., 1956, The magical number seven plus or minus two: some limits on our capacity for processing information, *The Psychological Review*, 63, 81–97.

Mitchell T.M.,1997, Machine learning, *The McGraw-Hill Companies Inc.*,New york, 250-252.

Norman, M.G., and Moscato, P., 1989, A competitive and cooperative approach to complex combinatorial search, *Technical Report, Caltech Concurrent*

Computation Program, Report 790, California, Institute of Technology, Pasedena, California, USA, 1989.

Perelson, A. S. and Oster, G. F., 1979, Theoretical studies of clonal selection: Minimal antibody repertuarie size and reliability of self-nonself discrimination, *J. Theor. Biol.*, 81,645-670.

ProQuest online veri tabanı, http://search.proquest.com?accountid=16935**,** [Ziyaret Tarihi:08.05.2012]

Quinlan, J.R.,1993, C4.5: Programs for machine learning, *Morgan Kauffman Publishers*, San Mateo-California.

Quinlan, J.R., 1995, MDL and categorical theories (Continued), In Machine Learning: Proceedings of the Twelfth International Conference, *Morgan Kaufmann*, Lake Tahoe, California, 464-470.

Ramirez, L.M., 2002, Support vector machines, radial basis functions neural networks, and fuzzy granulation in pattern recognition, Ms thesis, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, 16-17.

Rojas, R. and Feldman,J., 1996, Neural Networks: A systematic introduction, *Springer*.

Ross, T.J., 1995, Fuzzy logic with engineering applications, *McGraw-Hill Inc*, United States of America.

Rousseeuw, P.J., 1987, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, 20,53-65.

Rustagi, J. S., 1994, Optimization techniques in statistics, *Academic Press*, San Diego.

Shen, J.D., Gavade, A., Chang, S.I. and Lee, E.S., 2005, Generation of fuzzy membership function by expectation maximization and k-means algorithms, *Proceedings of the Fourth International Conference on Information and Management Sciences Book Series: Series of Information and Management Sciences*, : 4, Pages: 234-242.

Simon, D., 2005, H∞ Estimation for Fuzzy membership function optimization, *International Journal of Approximate Reasoning*, 40, 224–242.

Sousa, T., Silva, A. and Neves A., 2004, Particle swarm based data mining algorithms for classification tasks, *Parallel Computing*, 30, 767–783.

Şakiroglu, A.M. and Arslan, A., 2007, Optimization of fuzzy membership function using clonal selection, *Lecture Notes in Computer Science*, vol: 4431, Springer-Verlag, Berlin Heidelberg, 694-701.

Şen, Z., 2004, Mühendislikte bulanık (fuzzy) mantık ile modelleme prensipleri, *Su Vakfi*, İstanbul.

Takagi T. and Sugeno, M., 1985, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, 15 (1), 116-132.

Togai M. And Watanabe H., 1985, A VLSI implementation of a fuzzy inference engine, towards an expert system on a chip, *Information Science*, 38, 147-163.

Tan, K.C., Goh, C.K., Yang, J.Y. and Lee, T.H., 2006, Evolving better population distribution and exploration in evolutionary multi-objective optimization, *European Journal of Operational Research*, 171, 463–495.

Tan, K. C., Yu, Q., and Ang, J. H., 2006, A coevolutionary algorithm for rules discovery in data mining, *International Journal of Systems Science*, 37(12), 835-864.

Tung, W. L. and Quek, C., 2002a, DIC: a novel discrete incremental clustering technique for the derivation of fuzzy membership functions, *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, Lecture Notes in Computer Science, 2417, 485-491.

Tung, W. L. and Quek, C., 2002b, GenSoFNN: a generic self-organizing fuzzy neural network, *IEEE Transactions on Neural Networks*, 13(5), 1075–1086.

Uebele, V., Abe, S. and Lan, M.S., 1995, A Neural-Network-Based fuzzy classifier, *IEEE Transactions On Systems, Man and Cybernetics*, 25(2), 353-361.

Yang, C.C. and Bose, N.K., 2006, Generating fuzzy membership function with self-organizing feature map, *Pattern Recognition Letters*, 27, 356-365.

Vatansever M., Büyüklü A.H., 2009, Using visual data mining techniques in clustering analysis and an application, *Journal of Engineering and Natural Sciences Sigma 27*, 83-104.

Wang, X., Gao, X.Z. and Ovaska, S.J., 2008, A hybrid optimization method for fuzzy classification systems, *Eighth International Conference on Hybrid Intelligent Systems, HIS'08,* IEEE, Barcelona, Spain, 264-271.

Zadeh, L.A., 1965, Fuzzy sets, *Informatıon and Control*, 8, 338-353.

Zadeh, L.A., 1972, A rationale for fuzzy control, *J. Dyn. Syst. Meas. Control Trans. ASME*, 94, 3-4.

Zadeh, L.A., 1975, The concept of a linguistic variable and its applications to approximate reasoning, parts I,II, *Information Sciences*, vol.8, part III, vol.9.

Zadeh, L.A., 1994, Fuzzy Logic, Neural networks, and Soft computing, *Communications of the ACM*, 37(3), 77-84

Zadeh, L.A, 1997, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*,90, 111-127

Zadeh, L.A., 2006, Generalized theory of uncertainty (GTU)—principal concepts and ideas, *Computational Statistics & Data Analysis*, 51, 15-46.

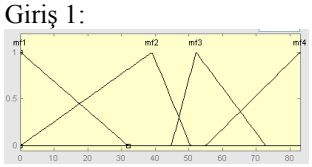
Zadeh L.A., 2008, Is there a need for fuzzy logic?, *Information Sciences,* 178, 2751-2779.

EKLER

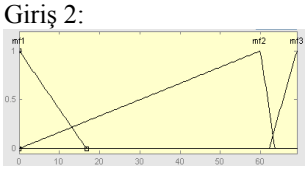
EK-1 Tez çalışmasında kullanılan sınıflandırma veri tabanları için Bölüm 6'da önerilen yöntem ile bulunan çözümler

Haberman Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

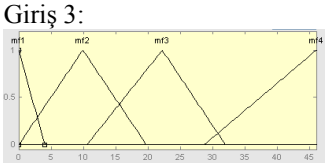
Giriş 1:



Giriş 2:



Giriş 3:

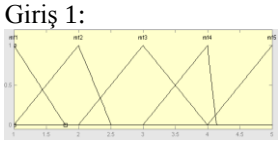


Bulanık Kurallar:

1. Eğer (Giriş2 mf3) ise Sınıf=1 (0.4313)
2. Eğer (Giriş1 mf4) ve (Giriş3 mf1) ise Sınıf=1 (0.5952)
3. Eğer (Giriş1 mf3) ve (Giriş2 mf2) ve (Giriş3 mf1) ise Sınıf=1 (0.7664)
4. Eğer (Giriş1 mf3) ve (Giriş2 mf2) ve (Giriş3 mf2) ise Sınıf=2 (0.0409)
5. Eğer (Giriş1 mf2) ve (Giriş2 mf2) ise Sınıf=1 (0.6601)

Balance Veri Tabanı için Bulanık Sınıflandırıcı Yapısı:

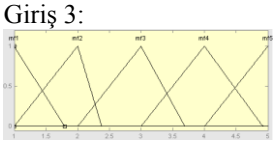
Giriş 1:



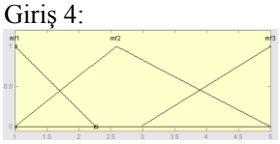
Giriş 2:



Giriş 3:



Giriş 4:

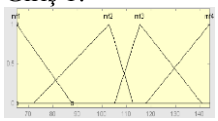


Bulanık Kurallar:

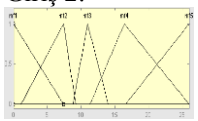
1. Eğer (Giriş1 mf1) ise Sınıf=3 (0.6919)
2. Eğer (Giriş4 mf3) ise Sınıf=3 (0.5185)
3. Eğer (Giriş4 mf1) ise Sınıf=1 (0.6233)
4. Eğer (Giriş2 mf5) ise Sınıf=1 (0.5530)
5. Eğer (Giriş2 mf1) ve (Giriş4 mf2) ise Sınıf=3 (0.7614)
6. Eğer (Giriş3 mf1) ise Sınıf=1 (0.6549)

New Thyroid Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

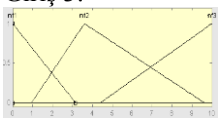
Giriş 1:



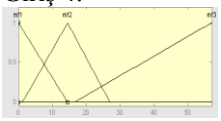
Giriş 2:



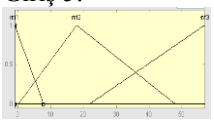
Giriş 3:



Giriş 4:



Giriş 5:

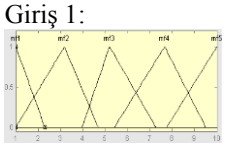


Bulanık Kurallar:

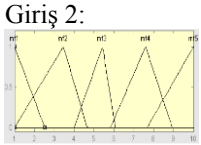
1. Eğer (Giriş2 mf1) ise O Halde (Sınıf = 3) (0.72833)
2. Eğer (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş5 mf1) ise O Halde (Sınıf = 1) (0.85428)
3. Eğer (Giriş2 mf4) ve (Giriş4 mf1) ise O Halde (Sınıf = 2) (0.66112)
4. Eğer (Giriş1 mf2) ve (Giriş3 mf2) ve (Giriş5 mf2) ise O Halde (Sınıf = 1) (0.94644)

Wisconsin Veri Tabanı için Bulanık Sınıflandırıcı Yapısı:

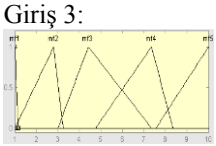
Giriş 1:



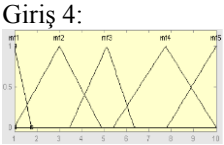
Giriş 2:



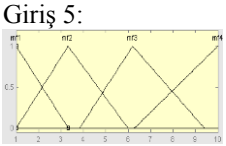
Giriş 3:



Giriş 4:



Giriş 5:

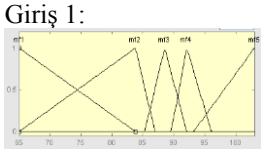


Bulanık Kurallar:

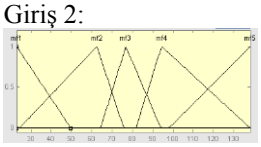
1.Eğer (Giriş1 mf4) ve (Giriş2 mf2) ise Sınıf=2 (0.4909)
2.Eğer (Giriş2 mf1) ve (Giriş5 mf1) ise Sınıf=1 (0.9872)
3.Eğer (Giriş1 mf3) ve (Giriş5 mf4) ise Sınıf=2 (0.9802)
4.Eğer (Giriş1 mf3) ve (Giriş4 mf2) ise Sınıf=2 (0.1430)
5. Eğer (Giriş1 mf5) ise Sınıf=2 (0.9880)
6. Eğer (Giriş4 mf5) ise Sınıf=2 (0.9486)
7. Eğer (Giriş4 mf2) ise Sınıf=1 (0.0341)
8. Eğer (Giriş3 mf3) ve (Giriş5 mf3) ise Sınıf=2 (0.8)

Liver Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

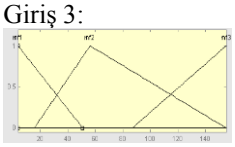
Giriş 1:



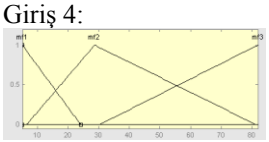
Giriş 2:



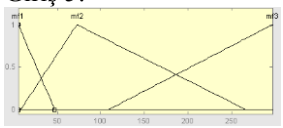
Giriş 3:



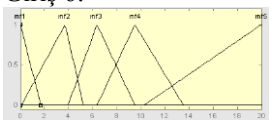
Giriş 4:



Giriş 5:



Giriş 6:

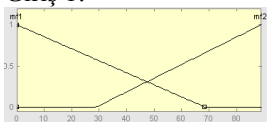


Bulanık Kurallar:

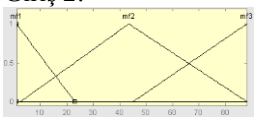
1. Eğer (Giriş4 mf2) ve (Giriş5 mf2) ise Sınıf=2 (0.2977)
2. Eğer (Giriş3 mf2) ve (Giriş5 mf1) ise Sınıf=1 (0.1645)
3. Eğer (Giriş2 mf3) ve (Giriş3 mf2) ve (Giriş4 mf1) ve (Giriş5 mf1) ise Sınıf=1 (0.6948)
4. Eğer (Giris1 mf4) ve (Giriş2 mf4) ve (Giriş3 mf1) ve (Giriş5 mf2) ve (Giriş6 mf1) ise Sınıf=1 (0.7860)
5. Eğer (Giris1 mf5) ve (Giriş2 mf4) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş5 mf2) ve (Giriş6 mf3) ise Sınıf=1 (0.5)

Ecoli Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

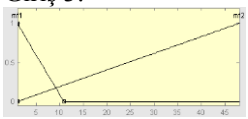
Giriş 1:



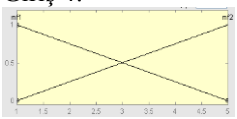
Giriş 2:



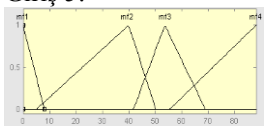
Giriş 3:



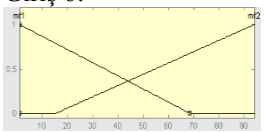
Giriş 4:



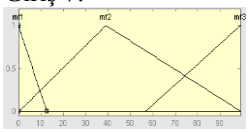
Giriş 5:



Giriş 6:



Giriş 7:

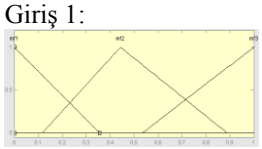


Bulanık Kurallar:

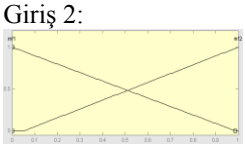
1. Eğer (Giriş1 mf1) ve (Giriş6 mf2) ise Sınıf=2 (0.3473)
2. Eğer (Giriş1 mf2) ve (Giriş2 mf2) ise Sınıf=5 (0.1479)
3. Eğer (Giriş3 mf1) ve (Giriş4 mf2) ve (Giriş6 mf1) ise Sınıf=7 (0.8046)
4. Eğer (Giriş1 mf2) ve (Giriş2 mf3) ve (Giriş4 mf2) ve (Giriş6 mf1) ise Sınıf=8 (0.4583)
5. Eğer (Giriş1 mf1) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş6 mf1) ise Sınıf=1 (0.8643)
6. Eğer (Giriş1 mf2) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş5 mf4) ve (Giriş6 mf1) ise Sınıf=6 (0.7189)

Appendisit Veri Tabanı için Bulanık Sınıflandırıcı Yapısı:

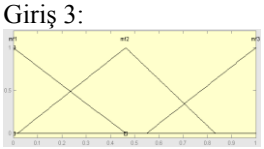
Giriş 1:



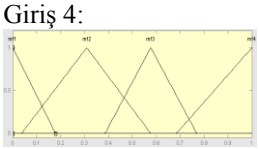
Giriş 2:



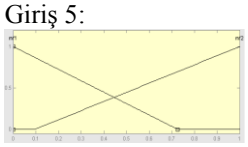
Giriş 3:



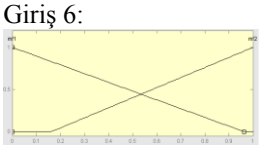
Giriş 4:



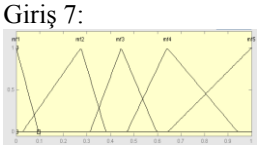
Giriş 5:



Giriş 6:



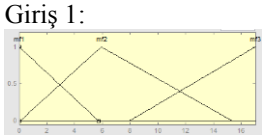
Giriş 7:

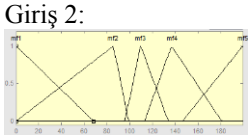


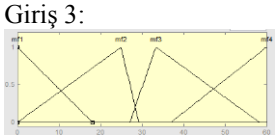
Bulanık Kurallar:

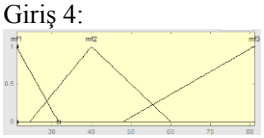
1. Eğer (Giriş1 mf2) ve (Giriş6 mf1) ise Sınıf=2 (0.6709)
2. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ve (Giriş3 mf1) ve (Giriş7 mf1) ise Sınıf=1 (0.4639)
3. Eğer (Giriş2 mf2) ve (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş5 mf1) ve (Giriş6 mf1) ve (Giriş7 mf2) ise Sınıf=1 (0.3)
4. Eğer (Giriş1 mf2) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş6 mf2) ve (Giriş7 mf4) ise Sınıf=2 (1)

Pima Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

Giriş 2: 

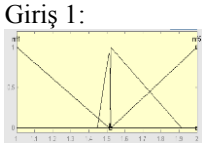
Giriş 3: 

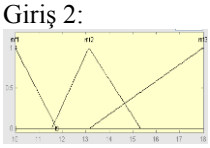
Giriş 4: 

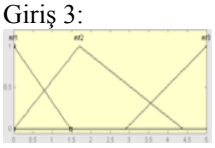
Bulanık Kurallar:

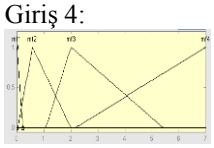
1. Eğer (Giriş1 mf1) ve (Giriş2 mf5) ve (Giriş3 mf2) ve (Giriş4 mf1) ise O Halde (Sınıf = 2) (0.80258)
2. Eğer (Giriş2 mf4) ve (Giriş4 mf1) ise O Halde (Sınıf = 2) (0.37379)
3. Eğer (Giriş3 mf2) ve (Giriş4 mf3) ise O Halde (Sınıf = 2) (0.62695)
4. Eğer (Giriş2 mf4) ve (Giriş3 mf3) ve (Giriş4 mf2) ise O Halde (Sınıf = 1) (0.20721)
5. Eğer (Giriş2 mf5) ise O Halde (Sınıf = 1) (0.60968)

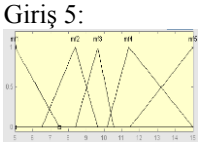
Glass Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

Giriş 2: 

Giriş 3: 

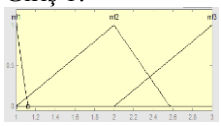
Giriş 4: 

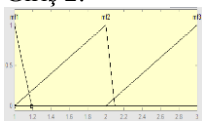
Giriş 5: 

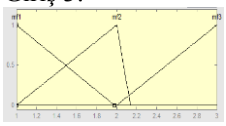
Bulanık Kurallar:

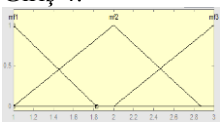
1. Eğer (Giriş1 mf3) ve (Giriş2 mf3) ve (Giriş3 mf2) ve (Giriş4 mf1) ise O Halde (Sınıf = 6) (0.3783)
2. Eğer (Giriş1 mf5) ise O Halde (Sınıf = 2) (0.75021)
3. Eğer (Giriş2 mf2) ve (Giriş3 mf3) ve (Giriş5 mf3) ise O Halde (Sınıf = 1) (0.60181)
4. Eğer (Giriş1 mf3) ve (Giriş5 mf4) ise O Halde (Sınıf = 5) (0.54585)
5. Eğer (Giriş3 mf3) ve (Giriş4 mf2) ise O Halde (Sınıf = 2) (0.35585)
6. Eğer (Giriş2 mf2) ve (Giriş3 mf2) ve (Giriş4 mf2) ise O Halde (Sınıf = 2) (0.33706)
7. Eğer (Giriş1 mf3) ve (Giriş2 mf2) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş5 mf2) ise O Halde (Sınıf = 1) (0.53311)
8. Eğer (Giriş2 mf2) ve (Giriş3 mf1) ve (Giriş5 mf2) ise O Halde (Sınıf = 4) (0.9036)

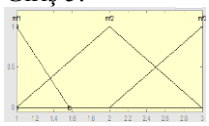
Tic-tac-toe Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

Giriş 2: 

Giriş 3: 

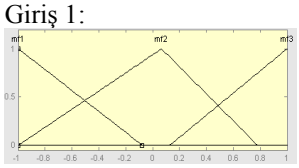
Giriş 4: 

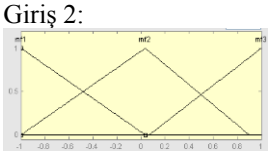
Giriş 5: 

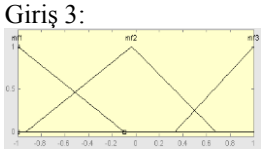
Bulanık Kurallar:

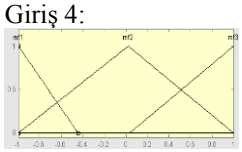
1. Eğer (Giriş3 mf1) ise O Halde (Sınıf = 1) (0.58231)
2. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ise O Halde (Sınıf = 1) (0.58108)
3. Eğer (Giriş2 mf1) ve (Giriş5 mf1) ise O Halde (Sınıf = 1) (0.58108)
4. Eğer (Giriş1 mf1) ve (Giriş4 mf1) ise O Halde (Sınıf = 1) (0.57971)
5. Eğer (Giriş4 mf1) ve (Giriş5 mf1) ise O Halde (Sınıf = 1) (0.59124)
6. Eğer (Giriş3 mf2) ise O Halde (Sınıf = 2) (0.13443)

Ionosphere Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

Giriş 2: 

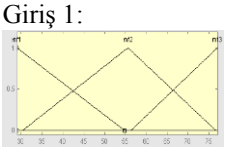
Giriş 3: 

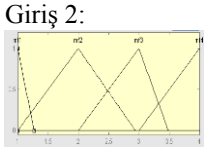
Giriş 4: 

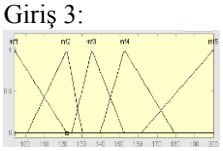
Bulanık Kurallar:

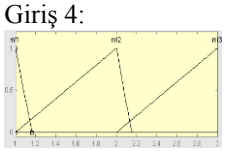
1. Eğer (Giriş1 mf3) ve (Giriş2 mf2) ve (Giriş4 mf2) ise Sınıf=1 (0.8398)
2. Eğer (Giriş1 mf2) ve (Giriş2 mf3) ve (Giriş4 mf3) ise Sınıf=1 (0.8856)
3. Eğer (Giriş2 mf1) ise Sınıf=2 (0.4390)
4. Eğer (Giriş1 mf2) ise Sınıf=2 (0.09710)
5. Eğer (Giriş1 mf3) ve (Giriş2 mf3) ise Sınıf=2 (0.1296)

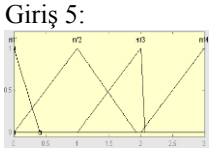
Cleveland Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

Giriş 2: 

Giriş 3: 

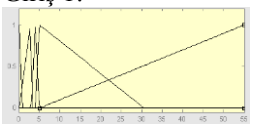
Giriş 4: 

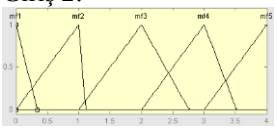
Giriş 5: 

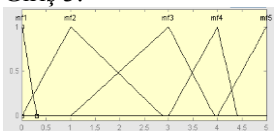
Bulanık Kurallar:

1. Eğer (Giriş1 mf1) ve (Giris4 mf2) ve (Giris5 mf4) ise Sınıf=3 (1)
2. Eğer (Giriş1 mf2) ve (Giriş3 mf3) ve (Giris5 mf2) ise Sınıf=1 (0.2259)
3. Eğer (Giriş1 mf3) ve (Giriş2 mf4) ve (Giriş5 mf4) ise Sınıf=4 (0.3384)
4. Eğer (Giriş1 mf2) ve (Giriş2 mf4) ve (Giriş3 mf1) ve (Giriş5 mf2) ise Sınıf=2 (0.4230)
5. Eğer (Giriş1 mf2) ve (Giriş2 mf4) ve (Giriş5 mf3) ise Sınıf=3 (0.3356)
6. Eğer (Giriş5 mf1) ise Sınıf=5 (0.7054)
7. Eğer (Giriş1 mf1) ve (Giris5 mf2) ise Sınıf=5 (0.4072)
8. Eğer (Giriş1 mf2) ve (Giriş3 mf3) ve (Giris4 mf2) ve (Giris5 mf3) ise Sınıf=2 (0.2259)
9. Eğer (Giriş1 mf2) ve (Giriş4 mf1) ise Sınıf=5 (0.6460)

Mammograph Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

Giriş 1: 

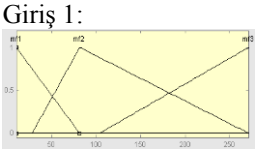
Giriş 2: 

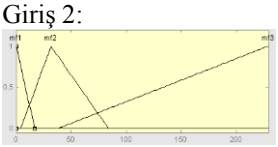
Giriş 3: 

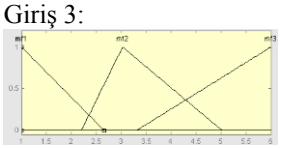
Bulanık Kurallar:

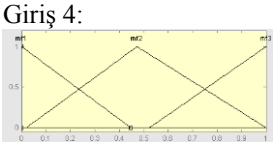
1. Eğer (Giriş1 mf4) ise O Halde (Sınıf = 1) (0.80604)
2. Eğer (Giriş2 mf5) ve (Giriş3 mf5) ise O Halde (Sınıf = 1) (0.67347)
3. Eğer (Giriş1 mf3) ve (Giriş2 mf5) ve (Giriş3 mf4) ise O Halde (Sınıf = 1) (0.14771)

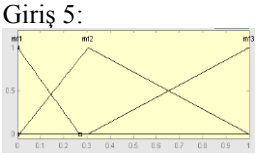
Magic Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

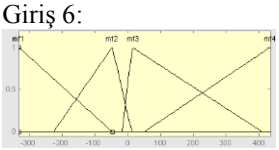
Giriş 1: 

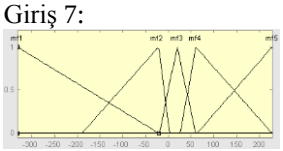
Giriş 2: 

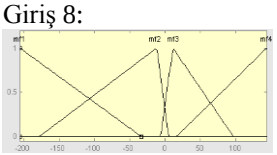
Giriş 3: 

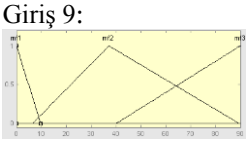
Giriş 4: 

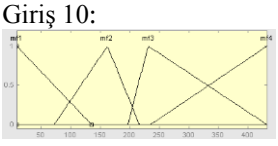
Giriş 5: 

Giriş 6: 

Giriş 7: 

Giriş 8: 

Giriş 9: 

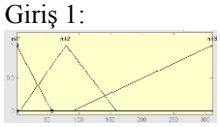
Giriş 10: 

Bulanık Kurallar:

1. Eğer (Giriş1 mf2) ve (Giriş3 mf2) ve (Giriş9 mf2) ise O Halde (Sınıf = 2) (0.47497)
2. Eğer (Giriş1 mf1) ve (Giriş2 mf2) ve (Giriş4 mf2) ve (Giriş5 mf2) ise O Halde (Sınıf = 1) (0.4783)
3. Eğer (Giriş1 mf2) ve (Giriş2 mf3) ve (Giriş3 mf2) ve (Giriş4 mf1) ve (Giriş5 mf1) ve (Giriş6 mf1) ve (Giriş8 mf2) ve (Giriş10 mf3) ise O Halde (Sınıf = 2) (0.93521)
4. Eğer (Giriş1 mf2) ve (Giriş2 mf2) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş5 mf2) ve (Giriş9 mf1) ise O Halde (Sınıf = 1) (0.82574)
5. Eğer (Giriş3 mf2) ve (Giriş9 mf3) ise O Halde (Sınıf = 2) (0.60473)
6. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ve (Giriş3 mf1) ve (Giriş4 mf2) ve (Giriş7 mf3) ve (Giriş8 mf2) ise O Halde (Sınıf = 1) (0.55192)

Page Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

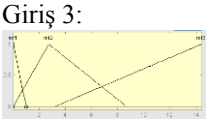
Giriş 1:



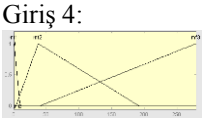
Giriş 2:



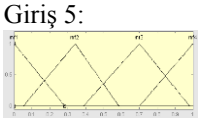
Giriş 3:



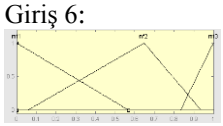
Giriş 4:



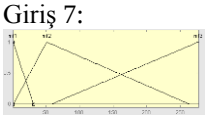
Giriş 5:



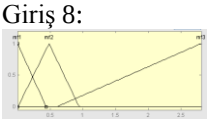
Giriş 6:



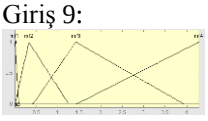
Giriş 7:



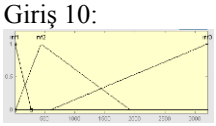
Giriş 8:



Giriş 9:



Giriş 10:

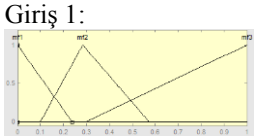


Bulanık Kurallar:

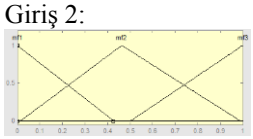
1. Eğer (Giriş1 mf3) ve (Giriş2 mf2) ve (Giriş4 mf1) ve (Giriş6 mf2) ve (Giriş7 mf1) ise O Halde (Sınıf = 3) (1)
2. Eğer (Giriş1 mf2) ve (Giriş4 mf1) ve (Giriş5 mf2) ve (Giriş7 mf2) ve (Giriş9 mf4) ve (Giriş10 mf2) ise O Halde (Sınıf = 5) (1)
3. Eğer (Giriş3 mf1) ve (Giriş7 mf2) ve (Giriş9 mf1) ise O Halde (Sınıf = 2) (0.2585)
4. Eğer (Giriş4 mf1) ve (Giriş8 mf1) ve (Giriş10 mf1) ise O Halde (Sınıf = 1) (0.92828)
5. Eğer (Giriş4 mf2) ve (Giriş10 mf2) ise O Halde (Sınıf = 1) (0.98529)

Sonar Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

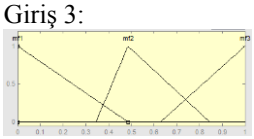
Giriş 1:



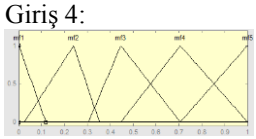
Giriş 2:



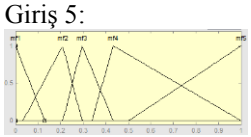
Giriş 3:



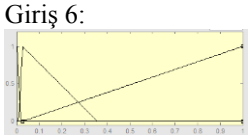
Giriş 4:



Giriş 5:



Giriş 6:

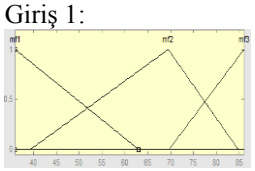


Bulanık Kurallar:

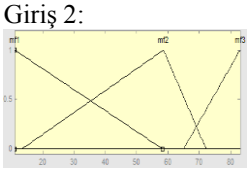
1. Eğer (Giriş2 mf2) ise O Halde (Sınıf = 2) (0.1909)
2. Eğer (Giriş3 mf2) ve (Giriş4 mf4) ve (Giriş5 mf2) ve (Giriş6 mf2) ise O Halde (Sınıf = 1) (0.76553)
3. Eğer (Giriş1 mf2) ve (Giriş3 mf3) ve (Giriş5 mf3) ve (Giriş6 mf2) ise O Halde (Sınıf = 2) (0.96721)
4. Eğer (Giriş1 mf2) ve (Giriş2 mf1) ve (Giriş3 mf2) ve (Giriş4 mf3) ve (Giriş5 mf1) ve (Giriş6 mf1) ise O Halde (Sınıf = 2) (1)
5. Eğer (Giriş1 mf1) ve (Giriş3 mf3) ise O Halde (Sınıf = 1) (0.6907)
6. Eğer (Giriş1 mf2) ve (Giriş4 mf4) ve (Giriş5 mf3) ise O Halde (Sınıf = 1) (0.36087)
7. Eğer (Giriş1 mf1) ve (Giriş2 mf3) ve (Giriş5 mf2) ve (Giriş6 mf2) ise O Halde (Sınıf = 1) (0.57674)
8. Eğer (Giriş2 mf3) ve (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş5 mf2) ve (Giriş6 mf2) ise O Halde (Sınıf = 2) (1)
9. Eğer (Giriş3 mf2) ve (Giriş6 mf1) ise O Halde (Sınıf = 1) (0.36911)
10. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ise O Halde (Sınıf = 1) (0.82759)
11. Eğer (Giriş1 mf2) ve (Giriş2 mf3) ve (Giriş3 mf1) ve (Giriş4 mf2) ve (Giriş5 mf1) ve (Giriş6 mf1) ise O Halde (Sınıf = 2) (0.84078)
12. Eğer (Giriş1 mf2) ve (Giriş3 mf2) ve (Giriş5 mf3) ve (Giriş6 mf2) ise O Halde (Sınıf = 1) (0.13296)

Spectfheart Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

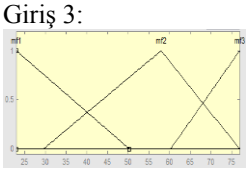
Giriş 1:



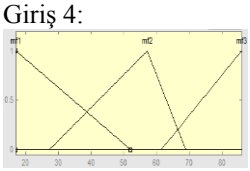
Giriş 2:



Giriş 3:



Giriş 4:

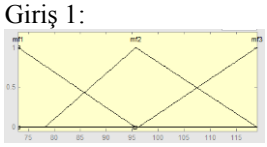


Bulanık Kurallar:

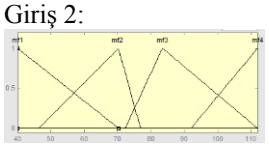
1. Eğer (Giriş1 mf2) ise O Halde (Sınıf = 1) (0.57338)
2. Eğer (Giriş3 mf2) ise O Halde (Sınıf = 1) (0.59893)

Vehicle Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

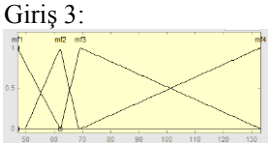
Giriş 1:



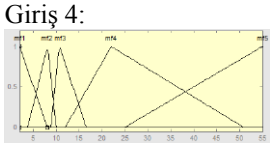
Giriş 2:



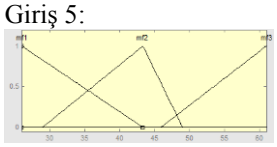
Giriş 3:



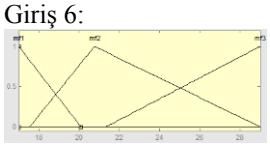
Giriş 4:



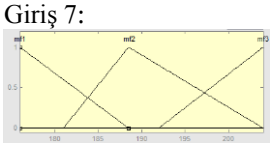
Giriş 5:



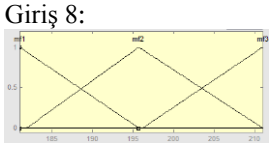
Giriş 6:



Giriş 7:



Giriş 8:

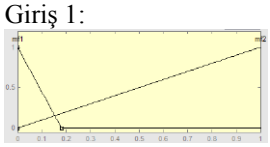


Bulanık Kurallar:

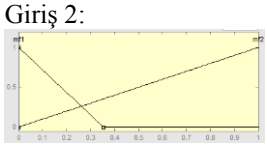
1. Eğer (Giriş1 mf2) ve (Giriş4 mf1) ve (Giriş5 mf1) ve (Giriş8 mf2) ise O Halde (Sınıf = 3) (0.77593)
2. Eğer (Giriş2 mf4) ve (Giriş3 mf2) ve (Giriş4 mf1) ve (Giriş6 mf2) ve (Giriş8 mf1) ise O Halde (Sınıf = 3) (1)
3. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ve (Giriş5 mf2) ve (Giriş7 mf1) ve (Giriş8 mf1) ise O Halde (Sınıf = 3) (0.72898)
4. Eğer (Giriş1 mf2) ve (Giriş2 mf3) ve (Giriş4 mf5) ve (Giriş7 mf2) ve (Giriş8 mf2) ise O Halde (Sınıf = 1) (1)
5. Eğer (Giriş2 mf3) ve (Giriş4 mf3) ve (Giriş6 mf1) ise O Halde (Sınıf = 1) (0.95219)
6. Eğer (Giriş2 mf4) ve (Giriş3 mf2) ve (Giriş4 mf3) ve (Giriş6 mf3) ve (Giriş7 mf3) ve (Giriş8 mf2) ise O Halde (Sınıf = 2) (0.36022)
7. Eğer (Giriş1 mf1) ve (Giriş4 mf1) ve (Giriş8 mf2) ise O Halde (Sınıf = 3) (0.13539)
8. Eğer (Giriş2 mf2) ve (Giriş3 mf4) ve (Giriş5 mf1) ve (Giriş6 mf1) ise O Halde (Sınıf = 3) (1)
9. Eğer (Giriş5 mf3) ve (Giriş6 mf1) ve (Giriş8 mf2) ise O Halde (Sınıf = 1) (0.43729)
10. Eğer (Giriş2 mf4) ve (Giriş3 mf2) ve (Giriş4 mf2) ve (Giriş5 mf1) ve (Giriş6 mf2) ve (Giriş8 mf3) ise O Halde (Sınıf = 2) (0.35522)
11. Eğer (Giriş1 mf3) ve (Giriş2 mf4) ve (Giriş5 mf1) ve (Giriş6 mf3) ve (Giriş7 mf2) ve (Giriş8 mf3) ise O Halde (Sınıf = 2) (0.55099)
12. Eğer (Giriş1 mf2) ve (Giriş5 mf1) ve (Giriş7 mf2) ve (Giriş8 mf2) ise O Halde (Sınıf = 4) (0.26291)
13. Eğer (Giriş1 mf3) ve (Giriş2 mf4) ve (Giriş3 mf2) ve (Giriş4 mf3) ve (Giriş5 mf1) ve (Giriş7 mf2) ve (Giriş8 mf2) ise O Halde (Sınıf = 2) (0.51579)
14. Eğer (Giriş1 mf2) ve (Giriş3 mf1) ve (Giriş6 mf2) ve (Giriş7 mf3) ve (Giriş8 mf3) ise O Halde (Sınıf = 2) (0.34891)
15. Eğer (Giriş2 mf3) ve (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş5 mf2) ve (Giriş6 mf2) ve (Giriş8 mf2) ise O Halde (Sınıf = 4) (0.26388)

Zoo Veri Tabanı için Bulanık Sınıflandırıcı Yapısı

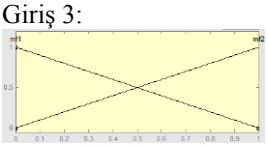
Giriş 1:



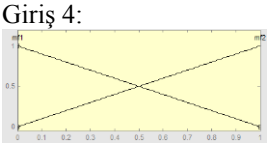
Giriş 2:



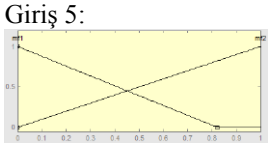
Giriş 3:



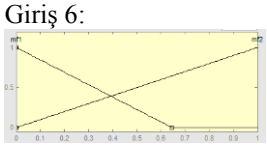
Giriş 4:



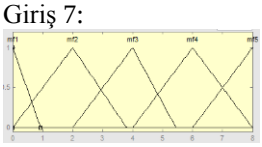
Giriş 5:



Giriş 6:



Giriş 7:



Bulanık Kurallar:

1. Eğer (Giriş2 mf1) ve (Giriş3 mf1) ve (Giriş5 mf1) ve (Giriş6 mf1) ve (Giriş7 mf4) ise O Halde (Sınıf = 6) (1)
2. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ve (Giriş6 mf1) ise O Halde (Sınıf = 7) (0.66667)
3. Eğer (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş7 mf1) ise O Halde (Sınıf = 4) (0.51961)
4. Eğer (Giriş3 mf1) ve (Giriş4 mf1) ve (Giriş5 mf2) ve (Giriş7 mf3) ise O Halde (Sınıf = 5) (0.64529)
5. Eğer (Giriş1 mf1) ve (Giriş2 mf1) ve (Giriş3 mf1) ve (Giriş5 mf1) ve (Giriş6 mf2) ise O Halde (Sınıf = 3) (1)
6. Eğer (Giriş3 mf2) ve (Giriş6 mf2) ise O Halde (Sınıf = 1) (1)
7. Eğer (Giriş1 mf1) ise O Halde (Sınıf = 2) (0.2451)

EK-2 Ek-1'de verilen üçgen üyelik fonksiyonlarına ait parametre değerleri

Haberman	Giriş1	mf1	0.00	0.00	32.04
		mf2	0.00	39.09	50.51
		mf3	44.84	52.11	72.85
		mf4	54.91	83.00	83.00
	Giriş2	mf1	0.00	0.00	16.73
		mf2	0.00	59.97	63.49
		mf3	62.10	69.00	69.00
	Giriş3	mf1	0.00	0.00	4.03
		mf2	0.00	9.91	19.73
		mf3	10.55	22.18	31.95
		mf4	28.69	46.00	46.00

Balance	Giriş1	mf1	1.00	1.00	1.80
		mf2	1.00	2.00	2.51
		mf3	2.00	3.00	4.00
		mf4	3.00	4.00	4.14
		mf5	4.00	5.00	5.00
	Giriş2	mf1	1.00	1.00	1.62
		mf2	1.00	2.00	2.97
		mf3	2.00	3.00	3.58
		mf4	3.00	4.00	5.00
		mf5	4.00	5.00	5.00
	Giriş3	mf1	1.00	1.00	1.80
		mf2	1.00	2.00	2.38
		mf3	2.00	3.00	3.69
		mf4	3.00	4.00	4.92
		mf5	4.00	5.00	5.00
	Giriş4	mf1	1.00	1.00	2.27
		mf2	1.00	2.58	5.00
		mf3	2.97	5.00	5.00

New Thyroid	Giriş1	mf1	65.00	65.00	87.79
		mf2	71.94	102.95	112.74
		mf3	105.10	115.53	140.88
		mf4	117.93	144.00	144.00
	Giriş2	mf1	0.00	0.00	7.41
		mf2	1.11	7.41	9.20
		mf3	8.79	10.97	14.02
		mf4	11.34	16.40	26.00
		mf5	16.58	26.00	26.00
	Giriş3	mf1	0.00	0.00	3.15
		mf2	0.94	3.60	9.62
		mf3	4.40	10.00	10.00
	Giriş4	mf1	0.00	0.00	14.45
		mf2	1.19	14.51	26.95
		mf3	16.60	57.00	57.00
	Giriş5	mf1	-1.00	-1.00	7.81
		mf2	0.12	17.88	47.99
		mf3	21.92	57.00	57.00

Wisconsin	Giriş1	mf1	1.00	1.00	2.30
		mf2	1.00	3.17	4.69
		mf3	3.97	5.19	7.27
		mf4	5.41	7.66	9.50
		mf5	7.75	10.00	10.00
	Giriş2	mf1	1.00	1.00	2.52
		mf2	1.00	3.45	4.65
		mf3	4.00	5.43	6.06
		mf4	5.83	7.59	8.93
		mf5	7.67	10.00	10.00
	Giriş3	mf1	1.00	1.00	1.16
		mf2	1.00	2.80	3.22
		mf3	3.00	4.41	7.35
		mf4	4.75	7.35	8.34
		mf5	7.55	10.00	10.00
	Giriş4	mf1	1.00	1.00	1.77
		mf2	1.00	3.00	4.88
		mf3	3.46	5.13	6.35
		mf4	5.40	7.74	10.00
		mf5	7.81	10.00	10.00
	Giriş5	mf1	1.00	1.00	3.31
		mf2	1.00	3.31	5.98
		mf3	3.84	6.19	9.39
		mf4	6.29	10.00	10.00

Liver (Bupa)	Giriş1	mf1	65.00	65.00	83.78
		mf2	65.00	83.79	86.96
		mf3	85.29	88.52	92.09
		mf4	89.56	92.09	96.07
		mf5	93.18	103.00	103.00
	Giriş2	mf1	23.00	23.00	49.57
		mf2	24.33	62.49	75.68
		mf3	64.35	76.72	94.03
		mf4	81.74	94.36	138.00
		mf5	97.90	138.00	138.00
	Giriş3	mf1	4.00	4.00	50.88
		mf2	15.85	56.20	154.75
		mf3	87.00	155.00	155.00
	Giriş4	mf1	5.00	5.00	24.05
		mf2	6.41	28.60	81.28
		mf3	30.35	82.00	82.00
	Giriş5	mf1	5.00	5.00	47.01
		mf2	7.65	72.27	266.51
		mf3	108.80	297.00	297.00
	Giriş6	mf1	0.00	0.00	1.70
		mf2	0.10	3.72	5.22
		mf3	3.95	6.35	9.50
		mf4	6.36	9.50	13.53
		mf5	10.26	20.00	20.00

pima	Giriş1	mf1	0.00	0.00	5.73
		mf2	0.07	5.90	15.33
		mf3	7.96	17.00	17.00
	Giriş2	mf1	0.00	0.00	68.11
		mf2	0.00	85.31	98.84
		mf3	95.33	109.31	133.68
		mf4	112.75	136.37	180.52
		mf5	145.96	199.00	199.00
	Giriş3	mf1	0.00	0.00	18.11
		mf2	0.00	24.97	29.24
		mf3	27.06	33.39	58.29
		mf4	36.95	60.00	60.00
	Giriş4	mf1	21.00	21.00	31.66
		mf2	24.36	39.92	59.96
		mf3	47.88	81.00	81.00

Appendicitis	Giriş1	mf1	0.00	0.00	0.36
		mf2	0.12	0.44	0.89
		mf3	0.53	1.00	1.00
	Giriş2	mf1	0.00	0.00	0.99
		mf2	0.05	1.00	1.00
	Giriş3	mf1	0.00	0.00	0.46
		mf2	0.02	0.46	0.83
		mf3	0.55	1.00	1.00
	Giriş4	mf1	0.00	0.00	0.18
		mf2	0.04	0.31	0.58
		mf3	0.39	0.58	0.77
		mf4	0.68	1.00	1.00
	Giriş5	mf1	0.00	0.00	0.73
		mf2	0.10	1.00	1.00
	Giriş6	mf1	0.00	0.00	0.96
		mf2	0.16	1.00	1.00
	Giriş7	mf1	0.00	0.00	0.10
		mf2	0.03	0.28	0.38
		mf3	0.31	0.45	0.60
		mf4	0.47	0.64	0.94
		mf5	0.64	1.00	1.00

Zoo	Giriş1	mf1	0.00	0.00	0.18
		mf2	0.00	1.00	1.00
	Giriş2	mf1	0.00	0.00	0.35
		mf2	0.00	1.00	1.00
	Giriş3	mf1	0.00	0.00	1.00
		mf2	0.00	1.00	1.00
	Giriş4	mf1	0.00	0.00	1.00
		mf2	0.00	1.00	1.00
	Giriş5	mf1	0.00	0.00	0.82
		mf2	0.00	1.00	1.00
	Giriş6	mf1	0.00	0.00	0.65
		mf2	0.00	1.00	1.00
	Giriş7	mf1	0.00	0.00	0.94
		mf2	0.00	2.00	3.80
		mf3	2.00	4.00	5.45
		mf4	4.00	6.00	8.00
		mf5	6.00	8.00	8.00

Ecoli	Giriş1	mf1	0.00	0.00	68.36
		mf2	29.00	89.00	89.00
	Giriş2	mf1	1.00	1.00	23.00
		mf2	2.52	43.41	88.00
		mf3	45.08	88.00	88.00
	Giriş3	mf1	1.00	1.00	10.90
		mf2	1.00	48.00	48.00
	Giriş4	mf1	1.00	1.00	5.00
		mf2	1.00	5.00	5.00
	Giriş5	mf1	0.00	0.00	8.08
		mf2	5.00	39.90	50.09
		mf3	41.67	53.58	68.91
		mf4	55.42	88.00	88.00
	Giriş6	mf1	1.00	1.00	68.44
		mf2	15.45	94.00	94.00
	Giriş7	mf1	0.00	0.00	12.48
		mf2	0.35	38.98	99.00
		mf3	56.85	99.00	99.00

Glass	Giriş1	mf1	1.00	1.00	1.51
		mf2	1.45	1.52	1.52
		mf3	1.52	1.52	1.52
		mf4	1.52	1.52	1.92
		mf5	1.52	2.00	2.00
	Giriş2	mf1	10.00	10.00	11.78
		mf2	11.56	13.14	15.34
		mf3	13.15	18.00	18.00
	Giriş3	mf1	0.00	0.00	1.48
		mf2	0.00	1.70	4.39
		mf3	2.90	5.00	5.00
	Giriş4	mf1	0.00	0.00	0.23
		mf2	0.04	0.59	2.01
		mf3	1.07	2.01	5.43
		mf4	2.17	7.00	7.00
	Giriş5	mf1	5.00	5.00	7.50
		mf2	6.51	8.38	9.65
		mf3	8.39	9.65	10.56
		mf4	10.17	11.39	15.00
		mf5	11.45	15.00	15.00

Tic Tac Toe	Giriş1	mf1	1.00	1.00	1.12
		mf2	1.00	2.00	2.57
		mf3	2.00	3.00	3.00
	Giriş2	mf1	1.00	1.00	1.19
		mf2	1.00	2.00	2.10
		mf3	2.00	3.00	3.00
	Giriş3	mf1	1.00	1.00	1.98
		mf2	1.00	2.00	2.14
		mf3	2.00	3.00	3.00
	Giriş4	mf1	1.00	1.00	1.83
		mf2	1.00	2.00	2.87
		mf3	2.00	3.00	3.00
	Giriş5	mf1	1.00	1.00	1.57
		mf2	1.00	2.00	2.99
		mf3	2.00	3.00	3.00

Magic	Giriş1	mf1	10.00	10.00	81.31
		mf2	27.82	81.31	271.00
		mf3	103.82	271.00	271.00
	Giriş2	mf1	0.00	0.00	16.93
		mf2	4.15	31.69	83.89
		mf3	37.99	229.00	229.00
	Giriş3	mf1	1.00	1.00	2.65
		mf2	2.20	3.03	5.02
		mf3	3.31	6.00	6.00
	Giriş4	mf1	0.00	0.00	0.45
		mf2	0.02	0.47	1.00
		mf3	0.52	1.00	1.00
	Giriş5	mf1	0.00	0.00	0.27
		mf2	0.01	0.31	1.00
		mf3	0.31	1.00	1.00
	Giriş6	mf1	-333.00	-333.00	-47.83
		mf2	-227.26	-47.83	11.98
		mf3	-17.55	13.00	409.35
		mf4	49.95	434.00	434.00
	Giriş7	mf1	-332.00	-332.00	-21.53
		mf2	-189.84	-21.53	3.02
		mf3	-19.61	19.49	59.86
		mf4	26.38	60.49	227.98
		mf5	63.48	228.00	228.00
	Giriş8	mf1	-206.00	-206.00	-33.70
		mf2	-178.20	-12.09	5.27
		mf3	-6.35	11.14	96.80
		mf4	16.03	144.00	144.00
	Giriş9	mf1	0.00	0.00	9.84
		mf2	6.48	37.18	89.50
		mf3	40.16	90.00	90.00
	Giriş10	mf1	9.00	9.00	135.41
		mf2	72.37	162.55	216.11
		mf3	196.75	231.34	432.00
		mf4	235.72	432.00	432.00

Mammograph	Giriş1	mf1	0.00	0.00	0.81
		mf2	0.00	2.88	3.18
		mf3	2.96	4.00	5.02
		mf4	4.00	5.03	30.59
		mf5	5.05	55.00	55.00
	Giriş2	mf1	0.00	0.00	0.34
		mf2	0.00	1.00	1.11
		mf3	1.00	2.00	2.76
		mf4	2.00	3.00	3.52
		mf5	3.00	4.00	4.00
	Giriş3	mf1	0.00	0.00	0.30
		mf2	0.00	1.00	2.88
		mf3	1.00	3.00	3.94
		mf4	3.00	4.00	4.41
		mf5	4.00	5.00	5.00

Page	Giriş1	mf1	1.00	1.00	56.58
		mf2	5.86	79.17	158.95
		mf3	89.64	311.00	311.00
	Giriş2	mf1	1.00	1.00	24.53
		mf2	16.26	119.05	206.75
		mf3	137.53	282.28	436.05
		mf4	305.57	538.00	538.00
	Giriş3	mf1	8.00	8.00	10038.23
		mf2	86.05	27228.80	86685.04
		mf3	31870.05	143993.00	143993.00
	Giriş4	mf1	0.00	0.00	9.91
		mf2	3.26	37.30	192.30
		mf3	39.34	279.00	279.00
	Giriş5	mf1	0.00	0.00	0.28
		mf2	0.05	0.34	0.59
		mf3	0.39	0.70	0.98
		mf4	0.70	1.00	1.00
	Giriş6	mf1	0.00	0.00	0.57
		mf2	0.06	0.65	0.94
		mf3	0.83	1.00	1.00
	Giriş7	mf1	1.00	1.00	31.72
		mf2	1.66	51.03	263.63
		mf3	58.11	277.00	277.00
	Giriş8	mf1	7.00	7.00	4392.06
		mf2	77.10	4837.40	9444.60
		mf3	6027.81	28093.00	28093.00
	Giriş9	mf1	7.00	7.00	855.44
		mf2	29.92	3244.16	12570.28
		mf3	4214.13	14170.17	39369.26
		mf4	14311.75	42821.00	42821.00
	Giriş10	mf1	1.00	1.00	273.60
		mf2	8.13	433.62	1918.49
		mf3	592.33	3212.00	3212.00

Cleveland	Giriş1	mf1	29.00	29.00	54.76
		mf2	30.34	55.58	76.66
		mf3	56.29	77.00	77.00
	Giriş2	mf1	1.00	1.00	1.27
		mf2	1.00	2.00	2.94
		mf3	2.00	3.00	3.49
		mf4	3.00	4.00	4.00
	Giriş3	mf1	94.00	94.00	122.02
		mf2	100.94	122.02	130.18
		mf3	124.54	135.35	152.61
		mf4	139.70	152.61	179.32
		mf5	161.60	200.00	200.00
	Giriş4	mf1	1.00	1.00	1.17
		mf2	1.00	2.00	2.15
		mf3	2.00	3.00	3.00
	Giriş5	mf1	0.00	0.00	0.41
		mf2	0.00	1.00	1.93
		mf3	1.00	2.00	2.07
		mf4	2.00	3.00	3.00

Spectfheart	Giriş1	mf1	36.00	36.00	62.96
		mf2	39.37	69.48	84.75
		mf3	69.64	86.00	86.00
	Giriş2	mf1	11.00	11.00	58.20
		mf2	13.44	58.57	72.17
		mf3	65.00	83.00	83.00
	Giriş3	mf1	23.00	23.00	50.35
		mf2	29.59	58.02	77.00
		mf3	60.30	77.00	77.00
	Giriş4	mf1	17.00	17.00	52.08
		mf2	27.30	57.20	68.94
		mf3	61.47	86.00	86.00

Ionosphere	Giriş1	mf1	-1.00	-1.00	-0.08
		mf2	-1.00	0.07	0.78
		mf3	0.13	1.00	1.00
	Giriş2	mf1	-1.00	-1.00	0.04
		mf2	-1.00	0.04	0.90
		mf3	0.07	1.00	1.00
	Giriş3	mf1	-1.00	-1.00	-0.10
		mf2	-0.93	-0.03	0.68
		mf3	0.33	1.00	1.00
	Giriş4	mf1	-1.00	-1.00	-0.44
		mf2	-1.00	0.02	1.00
		mf3	0.04	1.00	1.00

Vehicle	Giriş1	mf1	73.00	73.00	95.55
		mf2	78.15	95.67	119.00
		mf3	96.19	119.00	119.00
	Giriş2	mf1	40.00	40.00	70.30
		mf2	46.35	70.30	76.90
		mf3	72.33	83.50	112.00
		mf4	92.00	112.00	112.00
	Giriş3	mf1	47.00	47.00	62.17
		mf2	49.88	62.17	68.66
		mf3	62.26	69.14	133.00
		mf4	69.50	133.00	133.00
	Giriş4	mf1	2.00	2.00	8.09
		mf2	3.78	8.09	10.01
		mf3	8.78	10.70	16.60
		mf4	11.98	22.00	50.92
		mf5	24.91	55.00	55.00
	Giriş5	mf1	26.00	26.00	43.35
		mf2	28.97	43.35	49.05
		mf3	45.91	61.00	61.00
	Giriş6	mf1	17.00	17.00	20.09
		mf2	17.53	20.76	29.00
		mf3	21.33	29.00	29.00
	Giriş7	mf1	176.00	176.00	188.52
		mf2	181.00	188.52	204.00
		mf3	192.00	204.00	204.00
	Giriş8	mf1	181.00	181.00	195.71
		mf2	181.88	195.71	211.00
		mf3	196.18	211.00	211.00

Sonar	Giriş1	mf1	0.00	0.00	0.24
		mf2	0.10	0.28	0.57
		mf3	0.30	1.00	1.00
	Giriş2	mf1	0.00	0.00	0.43
		mf2	0.01	0.46	0.99
		mf3	0.50	1.00	1.00
	Giriş3	mf1	0.00	0.00	0.49
		mf2	0.34	0.49	0.85
		mf3	0.63	1.00	1.00
	Giriş4	mf1	0.00	0.00	0.12
		mf2	0.03	0.24	0.35
		mf3	0.31	0.45	0.71
		mf4	0.45	0.71	1.00
		mf5	0.71	1.00	1.00
	Giriş5	mf1	0.00	0.00	0.13
		mf2	0.03	0.21	0.30
		mf3	0.21	0.30	0.43
		mf4	0.34	0.43	1.00
		mf5	0.50	1.00	1.00
	Giriş6	mf1	0.00	0.00	0.02
		mf2	0.01	0.03	0.36
		mf3	0.03	1.00	1.00

Not: Çizelge 6.5'de nitelik seçimi uygulanan veritabanları ve seçilen niteliklerin numaraları verilmiştir. Bu veritabanlarının girişleri, sırası ile *seçilen nitelik no.* sütununda verilen niteliklere karşılık gelmektedir.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Ayşe Merve (Şakiroğlu) ACILAR
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	14.04.1980 - Konya
Telefon	**:**	+90 535 788 78 22
Faks	**:**	+90 332 241 06 35
e-mail	**:**	msakiroglu@selcuk.edu.tr

EĞİTİM

Derece		**Adı**	**Bitirme Yılı**
Lise	:	Konya Karatay Anadolu Lisesi - Konya	26.06.1998
Üniversite	:	Selçuk Üniversitesi Mühendislik Mimarlık Fak. Bilgisayar Mühendisliği - Konya	28.06.2002
Yüksek Lisans	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. A.B.D. - Konya	17.08.2005
Doktora	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. A.B.D. - Konya	

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2002-2003	Ilgın Meslek Yüksek Okulu	Öğretim Görevlisi
2003-devam	Selçuk Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI: Bulanık Mantık, Evrimsel Algoritmalar, Optimizasyon

YABANCI DİLLER: İngilizce

YAYINLAR

Index yada TUBİTAK Yayın Teşvik Listelerinde Yer Alan Yayınlar – Araştırma Makaleleri:

1. Acılar A.M, Arslan A., Optimization of multiple input–output fuzzy membership functions using clonal selection algorithm, Expert Systems with Applications, vol: 38, 1374-1381, 2011. (Doktora Tezinden)

2. Acılar A.M., Arslan A., A Collaborative Filtering Method Based On Artificial Immune Network, Expert Systems with Applications, vol: 36, pp: 8324-8332 , 2009. (Yüksek Lisans Tezinden)
3. Şakiroğlu M., Arslan A., "Optimization of Fuzzy Membership Function Using Clonal Selection" , Lecture Notes in Computer Science, 4431, 1, 694 - 701 , 2007. (Doktora Tezinden)
4. Tuğ E., Şakiroglu M., Arslan A., Automatic Discovery of the Sequential Accesses from Web Log Data Files via a Genetic Algorithm, Knowledge-Based Systems, vol:19, pp:180-186, 2006. (Yüksek Lisans Tezinden)

Uluslararası Bilimsel Toplantılarda Sunulan Bildiriler:

1. Acılar A.M., Arslan A., *Optimization of Multiple Input Single Output Fuzzy Membership Functions Using Clonal Selection Algorithm*, Proceedings of the 8th WSEAS International Conference on APPLIED COMPUTER SCIENCE (ACS'08), 49-53, Venice/Italy, 2008 (Doktora Tezinden)
2. Şakiroğlu M., Arslan A., *Tavsiye Sistemlerindeki Bilgi Eksikliği Problemine Veri Madenciliği Yaklaşımı*, 4. International Advanced Techologies Symposium, Konya,2005 (Yüksek Lisans Tezinden)

Ulusal Bilimsel Toplantılarda Sunulan Bildiriler:

1. Acılar A.M, Arslan A., Klonal Seçme Prensibiyle Bulanık Üyelik Fonksiyonlarının Optimizasyonu, Bilimde Modern Yöntemler Sempozyumu-BMYS'2008, sf: 121-128, Editörler: İ.B. Topçu, A. Öztopal, Z. Şen, 15-17 Ekim 2008, Eskişehir Osmangazi Üniversitesi, Eskişehir,Türkiye (Doktora Tezinden)