



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMA İLE BULANIK KURAL KÜMESİNİN OTOMATİK OLARAK OLUŞTURULMASINDA YENİ BİR YAKLAŞIM

Ersin KAYA

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalını

Aralık-2014

KONYA

Her Hakkı Saklıdır

ÖZET

DOKTORA TEZİ

GENETİK ALGORİTMA İLE BULANIK KURAL KÜMESİNİN OTOMATİK OLARAK OLUŞTURULMASINDA YENİ BİR YAKLAŞIM

Ersin KAYA

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ahmet ARSLAN

2014, 86 Sayfa

Jüri
Prof. Dr. Ahmet ARSLAN
Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. İ. Öztuğ BİLDİRİCİ
Doç. Dr. Erkan ÜLKER
Yrd. Doç. Dr. Ali ERDİ

İnsanoğlu günlük hayatta karşısına çıkan bir problemi çözmek için sözel anlama ve belirsizlikle mücadele becerilerini kullanmak zorundadır. Bulanık mantık ve karar destek teknolojilerinin gelişmesiyle günümüzde bilgisayar sistemleri de belirsiz ve sözel verilerle işlem yapma kabiliyetine sahip olmuşlardır. Bu sayede bulanık mantık temelli pek çok uygulama ve yaklaşım ortaya konulmuştur. Bulanık mantık özellikle kontrol sistemlerinde, sınıflandırıcı sistemlerde ve karar destek sistemlerinde oldukça yaygın olarak kullanılmaktadır.

Bulanık mantığın sınıflandırıcı sistemlerde kullanılan modelleri bulanık kural tabanlı sınıflandırıcılar olarak adlandırılmaktadır. Bulanık sınıflandırıcı sistemlerde çıkarım mekanizması önceden tanımlanmış kural kümesini kullanarak örnek verileri sınıflandırmaktadır. Sınıflandırıcının doğru olarak sınıflandırdığı örnek sayısı sınıflandırma başarısını göstermektedir. Bu sınıflandırıcılarda sınıflandırma başarısının yüksek olmasının yanında sistemin yorumlanabilirliğinin de yüksek olması gerekmektedir. Bulanık kural tabanlı sınıflandırıcılarda yorumlanabilirlik bulanık kural kümesindeki kural sayısı ve bu kuralların uzunluğu ile ifade edilmektedir. Kural sayısının ve kural uzunluğunun minimum olması sistemin yorumlanabililiğini artırmaktadır. Bir bulanık kural tabanlı sınıflandırıcı için en ideal bulanık kural kümesi, sınıflandırma başarısını maksimize eden en az sayıdaki ve en kısa uzunluktaki kural kümesidir.

Bu tez çalışmasında, ideal bulanık kural kümesinin oluşturulması için 3 yeni yaklaşıma sahip bir genetik algoritma tasarlanmıştır. Bu yeni yaklaşımlarla verimliliği artırılan genetik algoritma, bulanık sınıflandırıcı için ideal bulanık kural kümesini etkin bir şekilde oluşturmaktadır. Bu yaklaşımlardan ilkinde, Pittsburgh ve Michigan yaklaşımlarının avantajlarını barındıran yeni bir genetik birey kodlama yöntemi sunulmuştur. İkinci yeni yaklaşım ise uygunluk fonksiyonunda gerçekleştirilmiştir. Uygunluk fonksiyonu sadece sınıflandırma başarısına bağlı olarak değil kural sayısı ve kural uzunluğuna bağlı olarakta hesaplanmaktadır. Böylece ideal bulanık kural kümesinin özelliklerine sahip bir arama gerçekleştirilmektedir. Üçüncü yeni yaklaşım ise genetik algoritmanın mutasyon operatöründe yapılmıştır. Klasik genetik algoritmalarda tek bir mutasyon oranı kullanılmakta ve bu orana göre

mutasyon işlemi gerçekleştirilmektedir. Üç farklı mutasyon oranı kullanılarak, arama sırasında kural çeşitliliği sağlanırken daha kısa uzunlukta kuralların oluşmasına yardımcı olmaktadır.

Tez çalışmasında ortaya konulan yöntem ile literatürde başarı göstermiş 5 farklı yöntem kıyaslamıştır. Kıyaslama işlemi Toronto Üniversitesi ve Irvine California Üniversitesine ait veri ambarlarından elde edilen 18 farklı veri kümesi üzerinde yapılmıştır. Kıyaslama işleminin tutarlılığı için 10-kez çapraz doğrulama yöntemi 3 defa tekrarlanarak kullanılmıştır. Elde dilen sonuçlar sınıflandırma başarısı, kural sayısı ve kural uzunluğu açısından karşılaştırılmış ve analiz edilmiştir. Elde edilen sonuçlar istatistiksel olarak incelenmiş ve önerilen yöntemin diğer yöntemler karşısındaki başarısı ortaya konulmuştur.

Anahtar Kelimeler: Bulanık kural tabanlı sınıflandırıcı sistemler, Bulanık mantık, Genetik algoritmalar.

ABSTRACT

Ph.D THESIS

A NEW APPROACH FOR AUTOMATIC CREATION OF FUZZY RULE SET BY USING GENETIC ALGORTHM

Ersin KAYA

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

2014, 86 Pages

Jury
Prof. Dr. Ahmet ARSLAN
Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. İ. Öztuğ BİLDİRİCİ
Assoc. Prof. Dr. Erkan ÜLKER
Asst. Prof. Dr. Ali ERDİ

The human beings have to use linguistic understanding skills and deal with uncertainties to solve problems that encountered in everyday life. By the technological advancement in fuzzy logic and decision support systems, today computer systems can also process linguistic data and deal with uncertainties. Thus, fuzzy logic is used in control systems, classification systems and decision support systems.

Fuzzy logic systems which are used in the classifier model are called fuzzy rule-based classifiers. In fuzzy classification systems, inference mechanism classifies the sample data by using a set of predefined rules. The number of samples of the classifier correctly classified indicates classification performance. In addition to need high classification performance of the system, interpretability of the system should also to be high. Interpretability is described by the number of rules in the rule set and the length of these rules in fuzzy rule-based classification systems. Interpretability described by rarity and shortness of rules in a fuzzy rule based system. Thus the best set of fuzzy rules for fuzzy rule-based classification systems is a set of rules, which maximizes the classification performance and which has minimum number of rules with minimum length.

In this study, genetic algorithm that contains three new approaches is designed to create ideal set of fuzzy rules. Genetic algorithm, which increased productivity with these new approaches, effectively creates a set of rules for fuzzy classifier. First of these approaches is a new coding method of genetic individual that contain the advantages of Pittsburgh and Michigan approaches. The second new approach is carried out in the fitness function. Instead of calculating fitness function only from classification performance, number and the length of rules also effected. Thus, a search of the ideal set of fuzzy rules is provided. The third new approach is applied in genetic algorithm mutation operator. In classical genetic algorithms, a single mutation rate is used and mutation operation is performed according to this ratio. Using three different mutation rates, while maintaining the diversity rules, causes shorter length rules appear.

Test results of proposed method in this thesis compared to 5 different successful methods in the literature. Benchmarking process was conducted on 18 different data sets obtained from the data repository of University of Toronto and University of California. For consistency of benchmarking process, 10-fold cross validation method is repeated 3 times. The obtained results compared and analyzed in terms of classification success, the number of rules and the length of rules. The obtained results were statistically analyzed and the success of the proposed method versus other methods has been introduced.

Keywords: Fuzzy logic, Fuzzy rule-based classification systems, Genetic algorithm

ÖNSÖZ

Doktora tez çalışmam boyunca değerli katkıları, yönlendirici desteği ve örnek hoca kişiliğiyle çalışmalarıma katkı sağlayan 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. İ. Öztuğ BİLDİRİCİ'ye ve Sayın Yrd. Doç. Dr. Ali ERDİ'ye;

Tüm hocalarım ve mesai arkadaşlarıma özellikle Yrd. Doç. Dr. Barış KOÇER, Uzman Sedat KORKMAZ'a, Uzman Sait Ali UYMAZ'a ve Okutman Havvagül KOÇER'e desteklerinden dolayı teşekkür ederim.

Çalışmalarım boyunca ve hayatımın her anında anlayışları ve destekleri ile yanımda olan sevgili eşim Neşe KAYA'ya ve tüm aileme;

İçtenlikle teşekkür eder ve şükranlarımı sunarım.

Ersin KAYA
KONYA-2014

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **vi**

ÖNSÖZ ..... **viii**

İÇİNDEKİLER ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **5**

3. BULANIK SİSTEMLER **10**

3.1. Bulanık Sistemlerin Genel Yapısı 10
3.1.1. Veri Tabanı 12
3.1.2. Bulanık Kural 16
3.1.3. Çıkarım Sistemi 17
3.1.4. Durulaştırma 18
3.2. Bulanık Kural Tabanlı Sınıflandırıcı Sistemler 21
3.2.1. Bulanık Kural Kümesi 21
3.2.2. Tek Kazanan Çıkarım Yöntemi 22
3.2.3. Ağırlıklı Oylama Çıkarım Yöntemi 23

4. GENETİK ALGORTİMALAR **24**

4.1. Problemin Genetik Olarak Kodlanması 25
4.2. Uygunluk Fonksiyonunun Belirlenmesi 26
4.3. Seçim 26
4.4. Çaprazlama 27
4.5. Mutasyon 29
4.6. Durma Kriteri 29

5. GENETİK BULANIK SİSTEMLER **30**

5.1. Genetik Ayarlama 31
5.2. Genetik Öğrenme 32
5.2.1 Kural Öğrenme Yaklaşımı 32
5.2.2 Kural Seçme Yaklaşımı 33
5.2.3. Veri Tabanı Öğrenmesi Yaklaşımı 34
5.2.4. Veri Tabanı ve Kural Tabanı Öğrenmesi Yaklaşımı 36

6. İDEAL BULANIK KURAL KÜMESİNİN OLUŞTURULMASI İÇİN TASARLANAN GENETİK ALGORİTMA **37**

6.1. Bulanık Kural Kümesinin Genetik Birey Olarak Sunulması 39
6.2. Uygunluk Fonksiyonu 44
6.3. Seçim 49

6.4. Çaprazlama....50
6.5. Mutasyon....50

7. DENEYSEL ÇALIŞMALAR VE SONUÇLARI 53

7.1. Kullanılan Veri Kümeleri....53
7.2. Yapılan Deneysel Çalışmada Kullanılan Yardımcı Yöntemler 54
7.2.1. *k*-kez çapraz doğrulama yöntemi 54
7.2.2. Wilcoxon işaretli sıralamalar testi 55
7.2.3. Kullanılan Programlar....57
7.3. Yapılan Deneysel Çalışma 58

8. SONUÇLAR VE ÖNERİLER 76

8.1. Sonuçlar 76
8.2. Öneriler 80

KAYNAKLAR 81

ÖZGEÇMİŞ....85

1. GİRİŞ

İnsanlar, günlük hayatta karşılaştığı problemleri ifade ederken ve çözmeye çalışırken sözsel ifadeler kullanmaktadır. İnsan düşünce yapısı sözsel bilgileri işlemeye daha yatkındır. Buna karşın bilgisayar sistemleri sayısal verileri işlemeye ve analiz etmede daha iyidirler. Bulanık mantık, bilgisayar sistemlerinin sözsel terimler içeren problemlerin çözümünde etkin olarak kullanılmasına olanak sağlamaktadır. Bulanık mantık ilk olarak 1965 yılında Azeri asıllı bilim adamı Lotfi A. Zadeh tarafından ortaya atılmıştır.(Zadeh, 1965) Klasik küme mantığında bir eleman bir kümeye aittir veya değildir. Bulanık küme mantığında ise bir eleman bir den fazla kümeye farklı üyelik değerleri ile ait olabilir. Üyelik değeri 1 ile 0 arasında bir değerdir ve kümeye ait olma derecesini göstermektedir. Bulanık mantık ve beraberinde bulanık küme mantığı sayesinde günlük hayatta kullanılan sözsel terimler ile işlem yapabilme kabiliyetine sahip olunmaktadır. Özellikle matematiksel modelinin ortaya konulması zor olan denetim ve sınıflandırıcı sistemlerde bulanık mantık kullanılarak geliştirilen modeller ile başarılı sonuçlar elde edilmiştir. Bu modellerin başarılı olmalarındaki en önemli faktör bulanık mantık sayesinde sözsel terimler ile işlem yapabilmeleridir. Bunun yanında sözel terimler ile çalışabilme yeteneğine sahip bu modellerin insanlar tarafından daha kolay yorumlanabilmesi de avantaj sağlamaktadır.

Günümüzde bulanık mantık temelli pek çok uygulama ve yaklaşım ortaya konulmuştur. Bulanık mantık özellikle denetim sistemlerinde, kontrol sistemlerinde, sınıflandırıcı sistemlerde, karar destek sistemlerinde oldukça yaygın olarak kullanılmaktadır. Bulanık mantık ticari anlamda ilk olarak Danimarka da bir çimento fabrikasının fırın sıcaklığının ayarlanmasında kullanılmıştır (Holmblad ve Ostergaard, 1982). Bu işlem çok dikkat isteyen ve uzun süreli bir işlemdir. Vardiya değişiminde fırın kontrolünü gerçekleştiren kişilerin değişmesi her zaman aynı kalitede ürün çıkarma imkânını zora sokmaktadır. Gerçekleştirilen bulanık denetim sistemi hem istenilen kalite standardının yakalanmasını hem de enerji tasarrufunun yapılmasını sağlamıştır. Japonya'nın Sendai metrosunda da başarılı bir bulanık denetim sistemi uygulaması gerçekleştirilmiştir (Hitachi, 1987). Metro trenlerinin hızlanma ve yavaşlama hareketleri bulanık mantık ile kontrol edilmiş ve oldukça sarsıntısız şekilde hareket etmeleri sağlanmıştır. Bulanık mantık ile denetim ve kontrol sistemlerinde oldukça başarılı uygulamalar gerçekleştirilmiştir.

Literatürde bulanık mantık kullanılarak sınıflandırma yapan pek çok başarılı yöntemde ortaya konulmuştur. Bulanık sınıflandırıcı sistemler sözsel terimlerden oluşan bulanık kuraları kullanarak sınıflandırma yapmaktadır (Mansoori ve ark., 2008). Bu nedenle literatürde isimleri bulanık kural tabanlı sınıflandırıcılar (BKTS) olarak da geçmektedir (Baykal ve Beyan, 2004a). BKTS'lerde iki önemli bileşen bulunmaktadır. Bunlar bilgi tabanı ve çıkarım sistemidir. Bilgi tabanı sınıflandırılacak örneğin kesin değerlerini bulanık değerlere dönüştürecek bilgileri ve bulanık kuralları içermektedir. Çıkarım sistemi ise bilgi tabanındaki bulanık kuralları kullanarak bulanık değerlere dönüştürülmüş veriyi sınıflandırmaktadır. Basit anlamda BKTS'ler bu şekilde çalışmaktadır.

BKTS'lerin başarısındaki en önemli aşama bilgi tabanının probleme uygun olarak oluşturulmasıdır. Bilgi tabanı bulanık kurallardan ve kuralların oluşturulmasında kullanılacak sözsel terimleri ve bu terimlere ait üyelik fonksiyonlarını içermektedir. Bilgi tabanın oluşturulmasında, çözülmesi amaçlanan problem konusunda uzman bir kişi tarafından bulanık kuralların ve üyelik fonksiyonlarının oluşturulması veya bilgisayar sistemlerindeki diğer akıllı sistemlerden yararlanarak bulanık kuralların ve üyelik fonksiyonlarının oluşturulmasıdır. Bilgi tabanını oluşturacak uzman kişi bulmak her zaman mümkün olmadığı gibi uzman kişinin çözüm konusunda ne kadar yeterli olduğu da bir tartışma konusu olmaktadır. Bunun yerine önceden gözlenmiş örnek veriler kullanarak akıllı bilgisayar sistemleri sayesinde bilgi tabanı oluşturulabilmektedir (Uebele ve ark., 1995). Bu konuda başarılı birçok model ortaya konulmuştur. Bu modellerde yapay sinir ağları, genetik algoritmalar ve birliktelik kuralları gibi akıllı sistemler yaygın olarak kullanılmaktadır.

Genetik algoritmalar evrimsel süreçlerden esinlenerek geliştirilmiş bir arama algoritmasıdır. Genetik algoritmalar problem için birden fazla çözüm üreterek bunlar arasında en iyi olanı aramaktadır. Bu nedenle geniş çözüm uzayına sahip arama problemlerinde etkin olarak arama yapabilmektedir. BKTS sistemlerde bulanık kuralların ve üyelik fonksiyonlarının oluşturulması da bir arama problemi olarak ele alınabilmektedir. Bu nedenle genetik algoritmalar ve BKTS sistemler birlikte fazlaca kullanılmaktadır. Genetik algoritmalarda çözümü etkileyen önemli faktörler bireyin genetik olarak kodlanması, yeni bireylerin oluşturulması yöntemi ve bireylerin kalitesini ölçeklendirecek olan uygunluk fonksiyonu seçimidir. Bu faktörler problem için ne kadar uygun seçilirse elde edilen çözümde aynı oranda daha başarılı olmaktadır. Bu göstermektedir ki oluşturulan genetik bulanık sistemin başarısı oluşturulan genetik

modelin başarısı ile doğru orantılıdır. Literatürde farklı yaklaşımlara sahip genetik algoritmalar ile oluşturulan genetik bulanık sistemler vardır.

Bu tez çalışmasında BKTS sistemler için ideal bulanık kural kümesini arayan bir genetik algoritma modeli ortaya konulmuştur. Tasalanan genetik algoritma, ideal bulanık kural kümesinin bulunması amacına ulaşmak için yeni operatörler ve yaklaşımlar içermektedir. BKTS sistemlerin değerlendirilmesinde iki önemli etken vardır. Bunlar sistemin sınıflandırma başarısı ve sistemin yorumlanabilirliğidir. Sınıflandırma başarısı sistemin doğru olarak sınıflandırdığı örnek sayısının, toplam örnek sayısına oranıdır. Yorumlanabilirlik ise sınıflandırıcı sistemin kullandığı bulanık kural kümesindeki kural sayısı ve bu kuralların ortalama uzunluğudur. İdeal bulanık kural kümesi sınıflandırma başarısını en yüksek seviyede tutarken kural sayısını ve ortalama kural uzunluğunu da en düşük seviyede tutan özellikte olmalıdır. Tez çalışmasında, ortaya koyduğumuz yeni yaklaşımları içeren genetik algoritma ile BKTS'ler için ideal bulanık kural kümesi oluşturulmaya çalışılmıştır. Tasarlanan model literatürde kabul görmüş farklı genetik bulanık sistemler ile farklı veri kümeleri kullanılarak kıyaslanmış ve başarısı ortaya konulmuştur. Tasarlanan modelin ve diğer modellerin sonuçları, hem sınıflandırma başarısı açısından hem de bulanık kural kümesinin yorumlanabilirliği açısından incelenmiştir.

Yapılan tez çalışması 8 ana bölümden oluşmaktadır. İlk bölümde, tezin konusu ve amacı hakkında temel bilgiler verilmiştir. İkinci bölümde konuyla ilgili kaynak araştırması sunulmuştur. Üçüncü bölümde, bulanık sistemler ve bulanık sistemlerin temel bileşenleri olan bilgi tabanı, bulanıklaştırma, durulaştırma ve çıkarım mekanizmaları hakkında detaylı olarak bilgi verilmiş ve bu bileşenlerin arasındaki ilişki ortaya konulmuştur. Dördüncü bölümde ise, efektif bir arama algoritması olan genetik algoritmalar hakkında bilgi verilmiştir. Genetik algoritmaların problemi çözmesi için gerekli aşamaların tasarlanması ve bu aşamaların nasıl uygulanacağı sunulmuştur. Beşinci bölümde, BKTS'ler ile genetik algoritmaların beraber kullanıldığı genetik bulanık sistemler tanıtılmıştır. Genetik algoritmalar bulanık sistemlerde farklı amaçlar için kullanılabilmektedir. Farklı bulanık genetik modellerin çalışma yapısı bu bölümde sunulmuştur. Altıncı bölümde ise, tez çalışması kapsamında ortaya konulan yeni yaklaşımlara sahip genetik bulanık model tanıtılmıştır. Sunulan modelde ortaya konulan yeni yaklaşımlar ve bu yaklaşımların ideal bulanık kural kümesinin oluşturulmasına yaptığı katkı açıklanmıştır. Yedinci bölümde; Ortaya konulan yeni yaklaşımlara sahip model ile literatürde kabul görmüş yöntemler arasındaki deneysel kıyaslama çalışmaları

sunulmuştur. Bu bölümde deneysel çalışmada kullanılan veri kümeleri, kıyaslanan yöntemler ve deney parametreleri açıklanmıştır. Ayrıca yapılan deneysel çalışmaların sonuçlarının elde edilmesinde ve yorumlanmasında kullanılan yöntemler açıklanmıştır. Yapılan deney sonuçları bu bölümde eğitim ve test kümesi sınıflandırma başarısı, kural sayısı ve ortalama kural uzunluğu açısından incelenmiş ve inceleme sonuçları ortaya konulmuştur. Son bölümde, elde edilen sonuçların yorumları, yöntemlerin literatüre katkısı ve gelecekte yapılabilecek çalışmalar hakkında bilgi ve öneriler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Bulanık kural tabanlı sistemlerin tasarlanmasında dikkat edilmesi gereken pek çok önemli aşama bulunmaktadır. Bu aşamalar, bulanık kural tabanlı sistemlerin üyelik fonksiyonlarının belirlenmesi, bulanık çıkarım sistemin belirlenmesi, bulanık kural kümesinin oluşturulması ve aynı zamanda bulanık kural ağırlıklarının hesaplanmasıdır. Bulanık kural tabanlı sistemlerin tasarlanmasında bu aşamalar uzman bir kişi tarafından yapılabileceği gibi akıllı bilgisayar sistemleri tarafından da yapılabilmektedir. Literatürde bu aşamaların akılı sistemlerle oluşturulduğu pek çok model ortaya konulmuştur. Bu tez çalışması kapsamında ortaya konulan modeller içerisinde bulanık kuralların oluşturulmasında genetik algoritmaların kullanıldığı modeller incelenmiş ve araştırılmıştır.

BKTS sistemlerde sınıflandırma başarısını etkileyen en önemli faktör bulanık kural kümesidir. BKTS sistemlerde ideal bulanık kural kümesinin elde edilmesi en önemli problemlerden biridir. Çünkü kullanılan üyelik fonksiyonlarının sayısına ve veri kümesindeki nitelik sayısına bağlı olarak oluşturulabilecek olası bulanık kural sayısı üssel bir şekilde artmaktadır. Olası bulanık kural sayısının fazla olması ideal bulanık kural kümesinin oluşturulmasını zorlaştırmaktadır. Bu noktada güçlü bir arama algoritmasına ihtiyaç duyulmaktadır. Genetik algoritmalar çok geniş arama uzaylarında etkili aramalar yapabilmektedir. Bu nedenle genetik algoritmalar BKTS sistemlerde bulanık kuralların oluşturulmasında sıklıkla kullanılmıştır. Genetik algoritmalarda problemin çözümünün bir birey olarak kodlanması, yeni bireylerin oluşturulması ve kullanılan uygunluk fonksiyonunun seçimine göre çözümün kalitesi değişmektedir. Literatürde bulanık kural kümesinin oluşturulmasında farklı yaklaşımlar içeren genetik algoritmaların kullanıldığı pek çok çalışma bulunmaktadır.

BKTS sistemlerde, genetik algoritma kullanarak bulanık kural kümesinin oluşturulması çalışmalarından ilki Ishibuchi ve ark. (1995) tarafından yapılmıştır. Bu çalışmanın amacı sınıflandırıcının en yüksek sınıflandırma başarısına ulaşmasının yanında bu sınıflandırma başarısını mümkün olan en az sayıda bulanık kural ile gerçekleştirmektir. Öncelikle tasarlanan BKTS'nin giriş değerlerine ait üyelik fonksiyonları belirlenmektedir. Bu üyelik fonksiyonları kullanılarak bulanık alt kümeler oluşturulmaktadır. Bu alt kümeler aynı zamanda bulanık kuralları ifade etmektedir. Kullanılan üyelik fonksiyonu sayısı aynı zamanda olası bulanık kural kümesi sayısını da belirlemektedir. Bir giriş değeri için belirlenen üyelik fonksiyonu sayısındaki artış, olası

bulanık kural sayısında üssel olarak artışa neden olmaktadır. Daha sonra olası bulanık kuralların sınıf değerleri ve kural ağırlıkları sezgisel bir yöntem ile bulunmaktadır. Bu sezgisel yöntemde olası bulanık kuralın her bir sınıfa dahil olma değeri hesaplanmaktadır. Hangi sınıfın değeri daha yüksek ise kural o sınıf kümesine dahil edilmektedir. Bütün sınıflar için aynı değer bulunursa kural oluşturulmaz. Yine bu değere bağlı olarak kurallın 0 ile 1 arasında bir ağırlık değeri hesaplanmaktadır. Sınıf değerleri ve ağırlıkları hesaplanan kurallar kümesi *aday kural kümesi* olarak adlandırılır. Genetik algoritma kullanılarak bu aday kural kümesi içerisinden *ideal kural kümesi* oluşturulmaktadır. Bu amaçla iki değerli (0,1) ve aday kural kümesinin uzunluğunda bireyler oluşturulmaktadır. Eğer bireydeki bir bitin değeri 1 ise bu bite karşılık gelen aday kural ideal bulanık kural kümesine dahil edilmiştir. Eğer 0 ise dahil edilmemiştir. Çalışmada oluşturulan bireylerin yarısı her hangi bir değişime uğramadan diğer nesillere aktarılmakta, diğer yarısı ise tek noktadan çaprazlama ve mutasyon operatörü ile oluşturulmaktadır. Bireylerin çözüme uygunluğu ise bireyin ifade ettiği bulanık kural kümesi ve yapılan sınıflandırmanın başarısı ile doğru, bulanık kural kümesindeki kural sayısı ile ters orantılı olacak şekilde tasarlanmıştır. Böylece sınıflandırma başarısını maksimize eden ve en az sayıda bulanık kuraldan oluşan, kural kümesi oluşturulmaya çalışılmıştır. Bu çalışmanın dezavantajı; giriş değerlerini artması ve/veya üyelik fonksiyonu sayısının artması durumunda olası bulanık kural sayısının üssel bir şekilde artmasıdır. Aday kural kümesinin oluşturulmasında hesaplama maliyetine neden olmaktadır. Bu nedenle çok boyutlu veri kümelerinde etkin bir şekilde kullanılamamaktadır.

Ishibuchi ve ark. (1997) 1995 yılında ortaya koyduğu genetik bulanık sistemi 1996 yılında çok amaçlı genetik algoritma kullanarak geliştirmiştir. Bu çalışmada ilk olarak bulanık kurallar oluşturulmuştur. Bulanık kurallar üyelik fonksiyonları kullanılarak oluşturulan bulanık alt kümeler şeklinde tanımlanmıştır. Elde edilen bulanık kuralların sınıf değerleri ve kural ağırlıkları ait oldukları bulanık alt küme içerisindeki örnekler aracılığı ile hesaplanmıştır. Oluşturulan bu olası bulanık kural kümesi içerisinden ideal bulanık kural kümesini genetik algoritma ile bulmaktadır. İdeal bulanık kural kümesini en yüksek sınıflandırma başarısını veren en az sayıdaki kural olarak tanımlamaktadır. Genetik algoritmada uygunluk fonksiyonu olarak; kural sayısını, kural sayısı ağırlık katsayısı ile çarpılmaktadır. Elde edilen bu değer sınıflandırma başarısı ve sınıflandırma başarı katsayısının çarpımından elde edilen değerden çıkarılmakta ve uygunluk değeri oluşturulmaktadır. Kural sayısı arttıkça

çarpıldığı katsayı oranında uygunluk fonksiyonun değerini düşürmektedir. Böylece iki amacı gerçekleştiren bir uygunluk fonksiyonu tanımlanmaktadır. Bu çalışmada kural sayısı ile sınıflandırma başarısı açısından birbirlerini baskılamayan farklı çözümler üretmek için genetik algoritmanın birey seçme aşamasında farklı yaklaşımlar ortaya konulmuştur. Her ebeveyn birey çiftinin seçilmesinde sınıflandırma başarı katsayısını ve kural sayısı katsayısını rastgele değiştirerek genetik algoritmanın çözüm uzayında farklı yönlere arama yapmasını sağlamaktadır. Böylece algoritma birbirinden bağımsız olarak çalıştırıldığında birbirini baskılamayan çözümler üretebilmektedir. Ayrıca bu çalışmada genetik algoritmanın her iterasyonunda bulanık kuralların ağırlıkları, ödül ve ceza yöntemi ile güncellenmektedir. Bulanık kuralların ağırlık değerleri 0 ile 1 arasında bir değer almaktadır. Bu değer 1'e yakın ise kural sınıflandırmada daha etkili, 0'a yakın ise sınıflandırmada daha az etkili olmaktadır. İterasyon sırasında bir kural doğru sınıflandırdığı her örnek için ödül alarak değeri 1'e yaklaştırılmakta, yanlış sınıflandırdığı her örnek için ceza alarak ağırlık değeri 0'a yaklaştırılmaktadır. Bu çalışmada da üyelik fonksiyonu sayısına ve veri kümesindeki nitelik sayısına bağlı olarak oluşabilecek aday kural kümesinin büyüklüğü genetik algoritmanın arama uzayını genişletmektedir.

Bulanık kural tabanlı sistemlerde sistemin başarısı kadar sistemin yorumlanabilirliği de önemlidir. Sistemin yorumlanabilirliği kural kümesindeki bulanık kuralların uzunluğu ile bağlantılıdır. Ortalama kural uzunluğu daha az olan sistemler insanlar tarafından daha kolay anlaşılabilir ve yorumlanabilir. Bu nedenle Ishibuchi ve ark. (2001a) 2001 yılında ortaya koydukları çalışmada sadece sınıflandırma başarısını yükselten en az sayıdaki kural kümesini değil bu iki özelliğe sahip aynı zamanda ortalama kural uzunluğu az olan kural kümesini aramaktadır. Böylece 3 amaçlı bir genetik algoritma yaklaşımı ortaya konulmuştur. Bu amaçlar sınıflandırma başarısını maksimize ederken, kural sayısını ve bulanık kural kümesinin ortalama kural uzunluğunu minimize etmektir. Genetik algoritmanın uygunluk fonksiyonu bu amaç için düzenlemiştir. Kısa uzunlukta kurallar oluşturabilmek için *don't care* isimli bir üyelik fonksiyonu kullanılmaktadır. Bu üyelik fonksiyonun özelliği fonksiyon kendisine gelen bütün giriş değerleri için 1 üyelik değerini üretmektedir. Bir kuralın uzunluğu *don't care* üyelik fonksiyonu ile değerlendirilmeyen niteliklerinin sayısıdır. Bu üyelik fonksiyonu sayesinde kısa kurallar oluşturma kabiliyeti kazanan model yukarda belirtilen 3 amacı gerçekleştiren genetik bulanık model olarak literatürde yer almaktadır. Ishibuchi ve Yamamoto (2004) 3 amaçlı yaptığı çalışma için aday kuralların

oluşturulmasında bir veri madenciliği tekniği olan birliktelik kurallarındaki güven ve destek terimleri kullanılmıştır. Bu çalışmada oluşturulan olası her kuralın her sınıf değeri için güven değeri hesaplanmakta ve en büyük değere sahip olan sınıf o kuralın sonuç değeri olmaktadır. Daha sonra bu kuralların güven ve destek değerleri çarpılarak bir sıralama kriteri oluşturulmaktadır. Aday kural kümesi bu olası kural kümesinden her bir sınıf için seçilen belirli sayıdaki kurallardan oluşmaktadır. Kuralın ağırlık değeri için güven terimi kullanılmaktadır. Ayrıca Ishibuchi ve Yamamoto (2005a) yaptığı çalışmada güven terimini kullanarak 4 farklı tipte bulanık kural ağırlıklandırma yöntemi ortaya koymuştur.

Ishibuchi'nin ortaya koyduğu genetik modellerde popülasyondaki bir birey bulanık sınıflandırıcının kural kümesinin tamamını temsil etmektedir. Berlanga ve ark. (2010) ortaya koyduğu genetik bulanık modelde ise bir birey sadece bir kuralı temsil etmektedir. İdeal bulanık kural kümesi popülasyondaki bireylerin bir alt kümesi olarak tanımlanmaktadır. Bu modelde iki adet uygunluk fonksiyonu kullanılmaktadır. İlk uygunluk fonksiyonu bireysel uygunluk fonksiyonu diğer uygunluk fonksiyonu ise çözümü niteleyen bireylerden oluşan alt kümenin uygunluk fonksiyonudur. Bireylerin oluşturulmasında rekabetçi bir yöntem (*Token Competition* (Wong ve Leung, 2000), (Leung ve ark., 1992)) kullanarak iterasyon sırasında hem bireylerin çeşitliliği korunmakta hem de gereksiz kurallardan kaçınılmaktadır. Bir kuralın bir bireyi ifade ettiği bu yöntemde genetik operatörler de bu amaca uygun olarak uyarlanmıştır. Ortaya konulan model mevcut literatürdeki 5 farklı model ile karşılaştırılmış ve başarılı sonuçlar elde edilmiştir.

Gonzalez ve Perez (2001) yaptıkları çalışmada SLAVE (*Structural learning algorithm on vague environment*) adında bir yöntem ortaya koymuşturlar. Bu yöntemde genetik algoritma kullanılarak BKTS için bulanık kural tabanı oluşturulmaktadır. Bu yöntemde genetik algoritmanın her bireyi, bir kuralı temsil etmektedir. Genetik algoritma çözüm olarak popülasyondaki bireylerden oluşan bir alt kümeyi sunmaktadır. Bu yöntemde birey iki parçadan oluşmaktadır. Her iki parçada ikili sayı sistemine göre kodlanmaktadır. İlk parça kural içerisinde kullanılan nitelikleri temsil etmektedir. İkinci parça ise birinci kısımda kurala dahil edilen niteliklere ait sözsel terimleri temsil etmektedir. Bu yöntemde genetik algoritma her seferinde sadece en iyi kuralı bulmak için çalıştırılmaktadır. Bulunan kurallar eğitim kümesindeki örnekler karşısında sınanmaktadır. Bu işlem bütün eğitim örneklerini kapsayan bir kural kümesi bulunana

kadar devam etmektedir. Bu kural kümesi sınıflandırıcının kural kümesi olarak kullanılmaktadır.

Mansoori ve arkadaşları (2008) ortaya koydukları genetik bulanık sistem ile nümerik veri kümesini kullanarak bulanık kural kümesini oluşturmaktadırlar. Öncelikle aday kurallar oluşturulmakta ve bu aday kuralların sonuç değerleri hesaplanmaktadır. Aday kurallar uygunluk değerine bağlı olarak sıralanmakta ve her bir sınıf değeri için belirli bir sayıda kural seçilerek bu kurallardan yeni kurallar üretilmektedir. Genetik algoritmadaki her birey bir kuralı temsil etmektedir.

Ishibuchi ve Yamamoto (2005b) 2005 yılında yaptıkları çalışmada hibrit bir yöntem ile bulanık kural kümesi oluşturulmaktadır. Bu yöntemde temel genetik algoritmanın içinde Pittsburgh yaklaşımı (Freitas, 2002) kullanılarak kural tabanı aranmaktadır. Pittsburgh yaklaşımında (Freitas, 2002) var olan aday kural havuzundan arama yapılmaktadır. Genetik algoritmada mutasyon operatöründen sonra sadece bir iterasyon Michigan yaklaşımına sahip bir genetik algoritma işletilerek yeni kurallar oluşturulmaktadır. Bu yöntemde Pittsburgh ve Michigan yaklaşımlarının güçlü yönleri kullanılarak etkili bir yöntem ortaya konulmaya çalışılmıştır.

Kızılkaya ve ark. (2012) yaptıkları çalışmada genetik algoritmanın kural seçme aşamasında sezgisel bir yöntem kullanarak kural seçimini gerçekleştirmiştirler. Genetik bireylerin her biri bir kuralı temsil etmektedir. Böylece yüksek boyutlu veri kümelerinde etkin olarak kullanılan bir model ortaya konulmuştur.

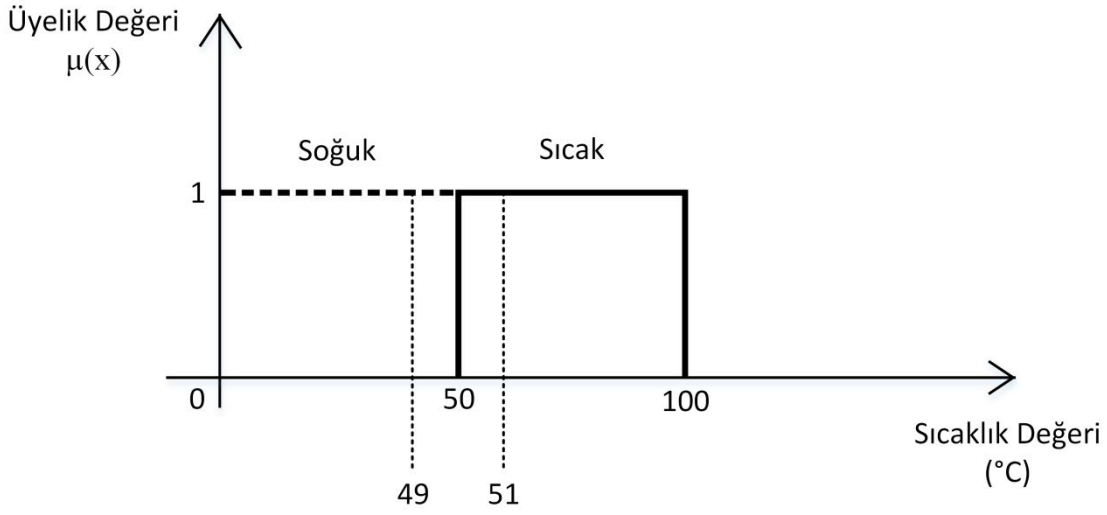
Lopez ve ark. (2013) çalışmalarında dengesiz sınıf dağılımına sahip veri kümelerinin, genetik bulanık modeller ile sınıflandırılması için bir model ortaya koymuşturlar. Kuralların veri kümesinden elde edildiği kural tabanlı sınıflandırıcılarda veri kümesindeki sınıf dağılımının dengesi önemlidir.

Dennis ve Muthukrishnan (2014), medikal verilerin sınıflandırmasında adaptif bir yapıya sahip genetik bulanık model sunmuşturlar. Modelde kural kümesi ve veri tabanı parametreleri süreç içerisinde otomatik olarak oluşturulmaktadır.

3. BULANIK SİSTEMLER

3.1. Bulanık Sistemlerin Genel Yapısı

Bilgisayar sistemleri kesin gerçekler üzerinde akıl yürütme ve işlem yapabilme yeteneğine sahiptir. İnsan beyni ise kesin olmayan bulanık gerçekler ile akıl yürütme yapabilmektedir. Bilgisayar sistemleri yüksek hız, soğuk hava, genç insan gibi değerler ile işlem yapabilme yeteneğine sahip değildir. 1965 yılında Azeri asıllı bilim adamı Lotfi A. Zadeh bulanık mantık kavramını geliştirmiştir. Bulanık mantık soğuk, hızlı, genç, küçük gibi günlük hayatta kullandığımız sözel terimleri bilgisayar sistemlerinde ve algoritmalarında kullanmamıza olanak sağlamaktadır. Bu sayede günlük hayatta karşımıza çıkan problemlerin çözümümde daha yorumlanabilir ve güçlü modeller ortaya konulabilir (Baykal ve Beyan, 2004b).

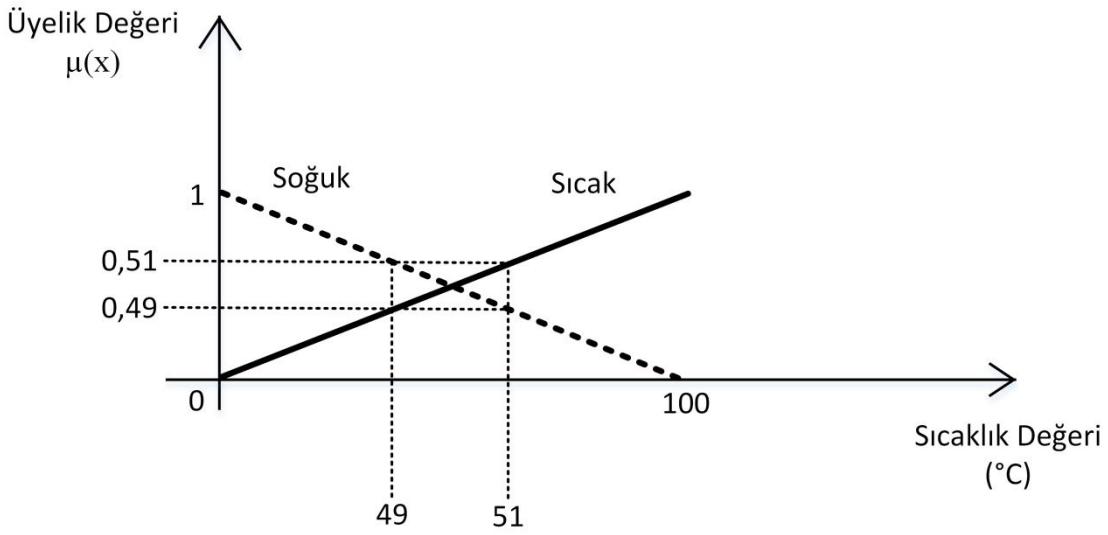


Şekil 3.1. Soğuk ve Sıcak kümelerine ait örnek klasik üyelik fonksiyonları

Bulanık mantığın temeli bulanık küme teorisine dayanmaktadır. Klasik küme mantığında bir eleman herhangi bir kümeye dahildir veya dahil değildir. Bulanık küme mantığında ise bir eleman birden fazla kümeye belirli bir üyelik değeri derecesinde ait olabilir. Örnek olarak elimizde "sıcak" ve "soğuk" olmak üzere iki kümemiz olsun. Klasik olarak bu kümeleri tanımlarsak 0-50 ^{0}C arasını soğuk, 50-100 ^{0}C arasını sıcak olarak tanımlayalım. Bu iki klasik kümenin üyelik fonksiyonları Şekil 3.1'de gösterilmiştir. Klasik küme mantığında 51 ^{0}C sıcak kümesinin bir elemanı 49 ^{0}C ise soğuk kümesinin elemanı olmaktadır. Aslında aralarında büyük sıcaklık farkları

olmamasına rağmen farklı iki kümede yer almaktadırlar. Klasik küme mantığında eğer bir eleman bir kümeye ait ise üyelik değeri 1 ait değilse üyelik değeri 0 dır.

Bulanık kümelerde ise üyelik değeri 0 ile 1 arasında değişen farklı değerler almaktadır. Bulanık küme mantığında eğer bir elemanın üyelik değeri 1'e yakın bir değer ise o kümeye daha fazla ait, 0'a yakınsa o kümeye daha az ait olmaktadır. Şekil 3.2'de "sıcak" ve "soğuk" bulanık kümelerine ait örnek üyelik fonksiyonları gösterilmektedir.

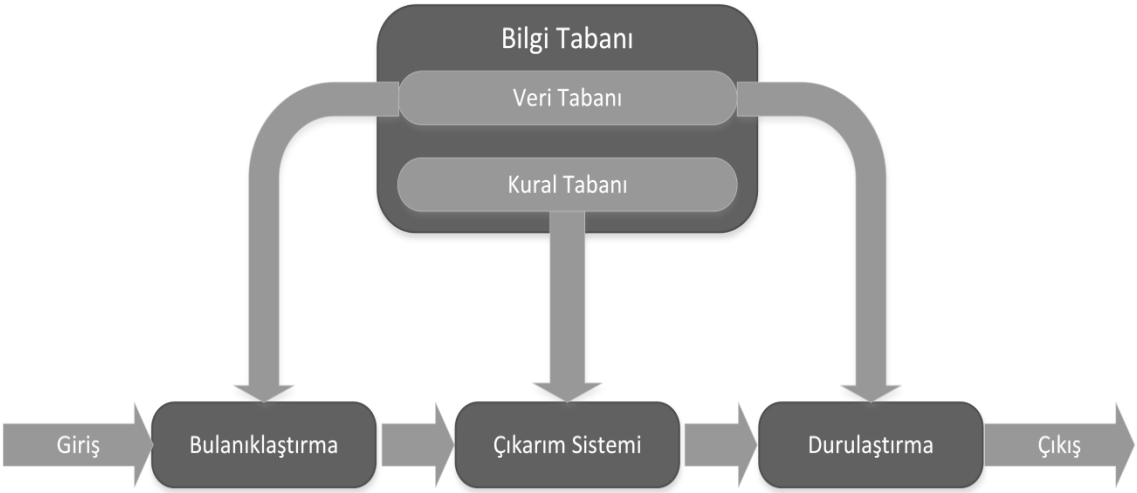


Şekil 3.2. Soğuk ve Sıcak kümelerine ait örnek bulanık üyelik fonksiyonları

Bir x değerinin bir kümeye ait olan üyelik değeri $\mu(x)$ şeklinde belirtilmektedir. Şekil 3.1 ve Şekil 3.2'de gösterilen örnek klasik ve bulanık üyelik fonksiyonlarına göre 49^0C ve 51^0C sıcaklıklarının üyelik değerleri klasik küme için $\mu_{soğuk}(49) = 1$, $\mu_{soğuk}(51) = 0$, $\mu_{sıcak}(49) = 0$ ve $\mu_{sıcak}(51) = 1$ bulanık küme için ise $\mu_{soğuk}(49) = 0.51$, $\mu_{soğuk}(51) = 0.49$, $\mu_{sıcak}(49) = 0.49$ ve $\mu_{sıcak}(51) = 0.51$ değerlerini almaktadır.

Bulanık sistemler kontrol, karar destek ve sınıflandırma sistemleri gibi pek çok alanda başarılı şekilde kullanılmaktadır. Kullanılan sitemlerin yapısı farklı olsa da temelinde bulanık mantık kavramı bulunmaktadır. Bulanık sistemlerin genel çalışma şekli şu şekildedir. İlk olarak sistemin giriş parametreleri bulanık değerlere dönüştürülmektedir. Bu işlem üyelik fonksiyonları aracılığı ile yapılmaktadır. Dönüştürülen bulanık değerler bir bulanık çıkarım mekanizması sayesinde tek bir bulanık çıkış değerine dönüştürülmektedir. Bulanık çıkarım mekanizmaları bu işlemi

sistemde daha önceden tanımlanmış bulanık kurallar sayesinde gerçekleştirmektedir. Elde edilen bu tek bulanık değer geçek bir değere dönüştürülerek sistemin çıkış değeri üretilmektedir. Bu yapıda çalışan bulanık sistemlere *Bulanık Kural Tabanlı Sistemler* de denilmektedir (Cordon, 1998). Bulanık kural tabanlı sistemleri detaylı olarak incelemek için bulanık sistemlerin yapısını detaylı olarak incelememiz gerekmektedir. Şekil 3.3'de bulanık sistemlerin genel yapısı, bulanık sistemleri oluşturan temel bileşenler ve bileşenler arasındaki ilişkiler gösterilmiştir. Bulanık sistemler; temel olarak 4 bileşenden oluşmaktadır. Bunlar bilgi tabanı, bulanıklaştırma birimi, çıkarım sistemi ve durulaştırma sistemidir. Bilgi tabanı; veri tabanı ve kural tabanı olmak üzere iki temel bileşenden oluşmaktadır (Cordon, 2011). Bu bileşenlerin işlevlerini ve diğer bileşenlerle olan ilişkilerini incelemek için her bir bileşeni ayrı bir başlık altında detaylandırabiliriz.



Şekil 3.3. Bulanık sistemlerin genel yapısı

3.1.1. Veri Tabanı

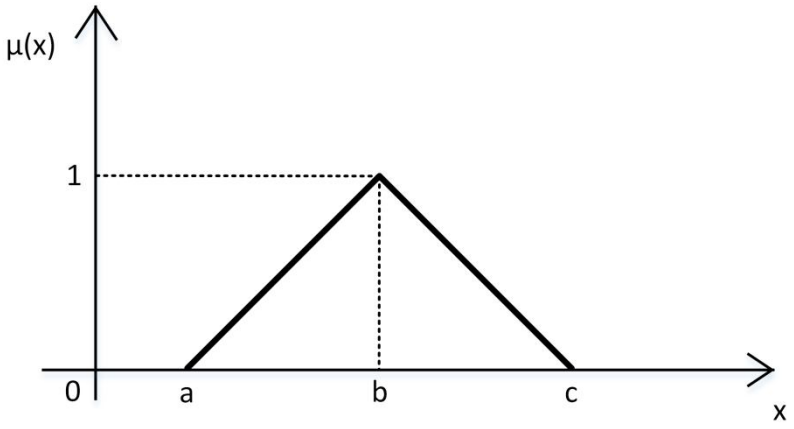
Veri tabanı bileşeni bulanık sistem içerisinde kullanılacak olan üyelik fonksiyonlarına ait bilgileri barındırmaktadır. Üyelik fonksiyonları bulanık sistemlerde gerçek verinin bulanık veriye dönüştürülmesinde veya bulanık verinin gerçek veriye dönüştürülmesinde kullanılmaktadır. Bu nedenle bulanıklaştırma ve durulaştırma birimleri veri tabanı ile ortak çalışmaktadır.

Bulanık küme mantığında bir eleman bir kümeye belirli bir üyelik değeri ile bağlı olmaktadır. Bu nedenle bulanık kümenin bir elmanı tanımlamak için elemanın

değeri ve ilgili kümeye ait üyelik değeri kullanılmaktadır. n elemanlı bir bulanık küme olan *B*, 3.1'deki gibi ifade edilmiştir.

$$B = \{ (\mu_B(x_1), x_1) , (\mu_B(x_2), x_2) , \dots , (\mu_B(x_n), x_n) \} = \{ \sum \mu_B(x_i), x_i \} \quad (3.1)$$

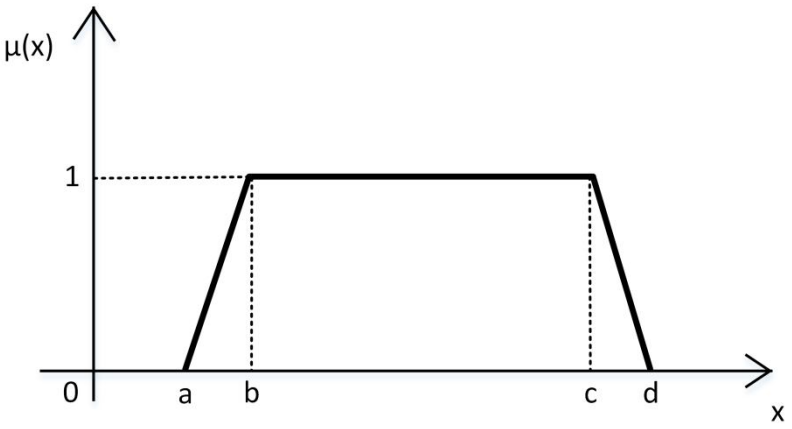
Küme elemanının değerindeki değişime bağlı olarak üyelik değerindeki değişimi gösteren eğriye üyelik fonksiyonu denir. Üyelik fonksiyonlarında *x* ekseni elemanın değerini, *y* ekseni ise üyelik değerini göstermektedir (Ross, 1995). Literatürde kullanılan pek çok üyelik fonksiyonu mevcuttur. Bunlardan yaygın olarak kullanılanlar üçgen, yamuk, Gaussian ve çan üyelik fonksiyonlarıdır (Baykal ve Beyan, 2004a).



Şekil 3.4. Üçgen üyelik fonksiyonu

Şekil 3.4'de örnek üçgen üyelik fonksiyonu gösterilmiştir. Üçgen üyelik fonksiyonunu tanımlamak için 3 adet parametreye (*a*, *b* ve *c*) ihtiyaç duyulmaktadır. Üçgen üyelik fonksiyonun formülü 3.2'de gösterilmiştir.

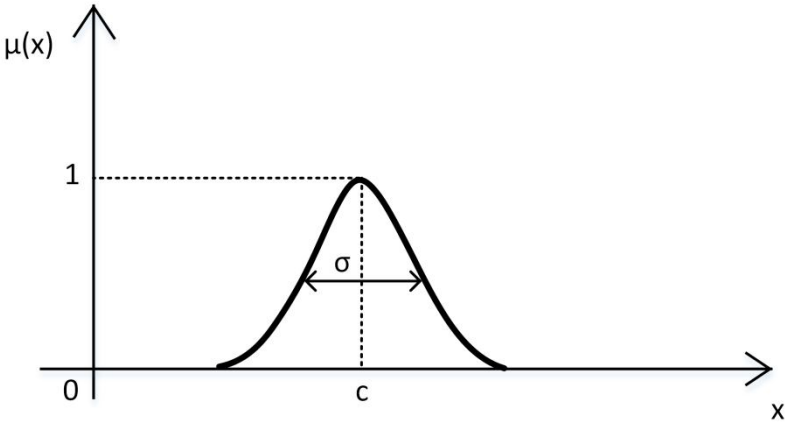
$$\mu(x) = \begin{cases} a \le x \le b \quad ise \quad (x-a)/(b-a) \\ b \le x \le c \quad ise \quad (c-x)/(c-b) \\ x < a \;\; veya \;\; x > c \;\; ise \;\; 0 \end{cases} \quad (3.2)$$



Şekil 3.5. Yamuk üyelik fonksiyonu

Yamuk üyelik fonksiyonunu tanımlamak için Şekil 3.5'de gösterildiği gibi 4 adet parametreye (*a*, *b*, *c* ve *d*) ihtiyaç duyulmaktadır. Yamuk üyelik fonksiyonun formülü 3.3'de görüldüğü gibidir.

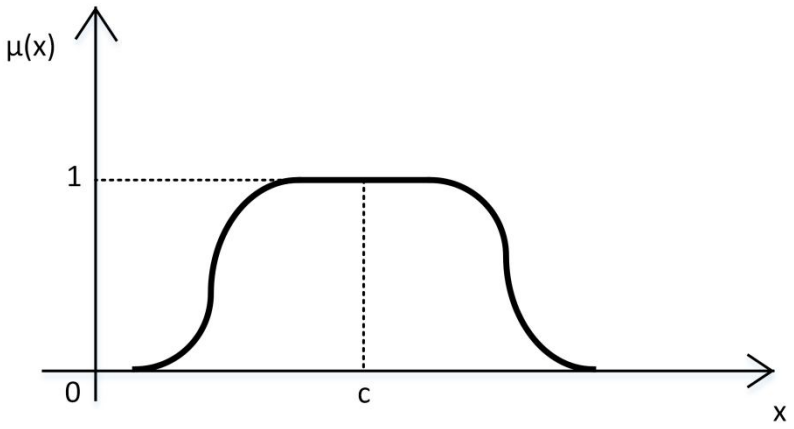
$$\mu(x) = \begin{cases} a \leq x \leq b \quad ise \quad (x-a)/(b-a) \\ b \leq x \leq c \quad ise \quad 1 \\ c \leq x \leq d \quad ise \quad (d-x)/(d-c) \\ x < a \;\; veya \;\; x > d \;\; ise \;\; 0 \end{cases} \tag{3.3}$$



Şekil 3.6. Gaussian üyelik fonksiyonu

Şekil 3.6'de Gaussian üyelik fonksiyonu gösterilmiştir. Bu üyelik fonksiyonunu tanımlayan *c* ve σ parametreleridir. *c* parametresi üyelik fonksiyonunun merkezini, σ parametresi ise fonksiyonun genişliğini ifade etmektedir. Fonksiyonun formülü 3.4'deki şekildedir.

$$\mu(x) = \exp\left(-\frac{-(x-c)^2}{2\sigma^2}\right) \tag{3.4}$$



Şekil 3.7. Çan üyelik fonksiyonu

Çan üyelik fonksiyonu *a, b* ve *c* parametreleri ile tanımlanmaktadır. *a* parametresi fonksiyonun genişliğini, *c* parametresi fonksiyonun merkezini, *b* parametresi ise fonksiyonun eğrilik derecesini belirlemektedir. Şekil 3.7'da gösterilen çan üyelik fonksiyonun formülü 3.5'de gösterilmektedir.

$$\mu(x) = \frac{1}{1 + \left[\frac{x-c}{a}\right]^{2b}} \tag{3.5}$$

Bulanık sistemlerde üyelik fonksiyonları sistemin girişini oluşturan gerçek değerleri bulanık değerlere dönüştürmektedir. Aynı zamanda sistemin bulanık çıkış değerini gerçek değere dönüştürerek sistemin çıkış değerini oluşturmaktadır. Bulanık sistemlerde hangi tip üyelik fonksiyonunun kullanılacağını belirleyen kesin bir yöntem yoktur. Üyelik fonksiyonlarının tipleri ve parametrelerinin belirlenmesi; deneme yanılma yöntemi ile veya problem konusunda uzman kişilerin görüşleri doğrultusunda yapılmaktadır.

3.1.2. Bulanık Kural

Bulanık kurallar, bulanık sistemlerin en önemli parçasını oluşturmaktadır. Bulanık sistemler girişlerine verilen değerleri bulanık kurallar sayesinde değerlendirebilmektedir. Bulanık kurallar *Eğer - ise* kuralları şeklinde gösterilmektedir. Eğer-ise kuralları koşul ve sonuç olmak üzere iki kısımdan oluşmaktadır. Örnek olarak *Eğer aracın hızı yüksek ise frene çok bas* bulanık kuralını ele alalım. Kuralın *aracın hızı yüksek* bölümü kuralın koşul kısmını, *frene çok bas* bölümü ise kuralın sonuç kısmını oluşturmaktadır. Bulanık kurallar sistemin giriş ve çıkış parametrelerini ve aynı zamanda bu parametrelerin sözsel karşılıklarını içermektedir. Örnek bulanık kuralın koşul kısmında "aracın hızı" giriş parametresi ve bu parametrenin sözsel değeri "yüksek" olarak ifade edilmiştir. Kuralın sonuç kısmında ise "frene basma şiddeti" çıkış parametresi ve bu parametrenin sözsel değeri "çok" olarak belirlenmiştir. Bulanık kurallarda kullanılan bu sözsel ifadeler bir üyelik fonksiyonu ile tanımlanmaktadır. Bulanık sistemlerde giriş ve çıkış parametrelerinin kaç üyelik fonksiyonu kullanılarak tanımlanacağı ve bu üyelik fonksiyonlarının hangi tipte olacağı belirlendikten sonra sistemin çalışmasındaki en önemli aşama olan bulanık kuralların oluşturulması aşaması gelmektedir.

Bulanık modellerde iki tip bulanık kural kullanılmaktadır. Bunlar Mamdani ve Takagi Sugeno Kaan (TSK) modelinde kullanılan bulanık kural tipleridir (Cordon, 1999b). *n* adet girişi ve 1 adet çıkışı olan bir Mamdani tipi bulanık modelin bulanık kuralı 3.6'da gösterilmiştir. x_i ile *i* inci girişe ait giriş parametresini temsil etmektedir. A_i ile *i* inci değere ait sözsel değişkeni temsil etmektedir. Başka bir deyişle giriş değerinin hangi üyelik fonksiyonu ile ifade edileceğini göstermektedir. *y* ile çıkış parametresi temsil edilmektedir. *C* ile çıkış parametresinin durulaştırılmasında kullanılacak üyelik fonksiyonu belirtilmektedir.

$$E\check{G}ER\ x_1 = A_1\ \ ve\ \ x_2 = A_2\ \ ve \ldots ve\ \ \ x_n = A_n\ \ \ \dot{I}SE\ \ y = C \tag{3.6}$$

TSK tipi bulanık modellerde kullanılan kural şekli Mamdani tipinde kullanılan kural tipinden farklıdır. Mamdani tipindeki kuralların sonuç kısmında çıkış değeri bir sözsel değişken ile ifade edilmektedir. TSK tipinde ise kuralın sonuç kısmı kesin bir fonksiyon ile ifade edilmektedir. 3.7'de TSK tipi bulanık modellerde kullanılan bulanık kural ifadesi gösterilmektedir.

$$EĞER\ x_1 = A_1\ \ ve\ \ x_2 = A_2\ \ ve \ldots ve\ \ \ x_n = A_n\ \ \ İSE\ \ y = f(x) \tag{3.7}$$

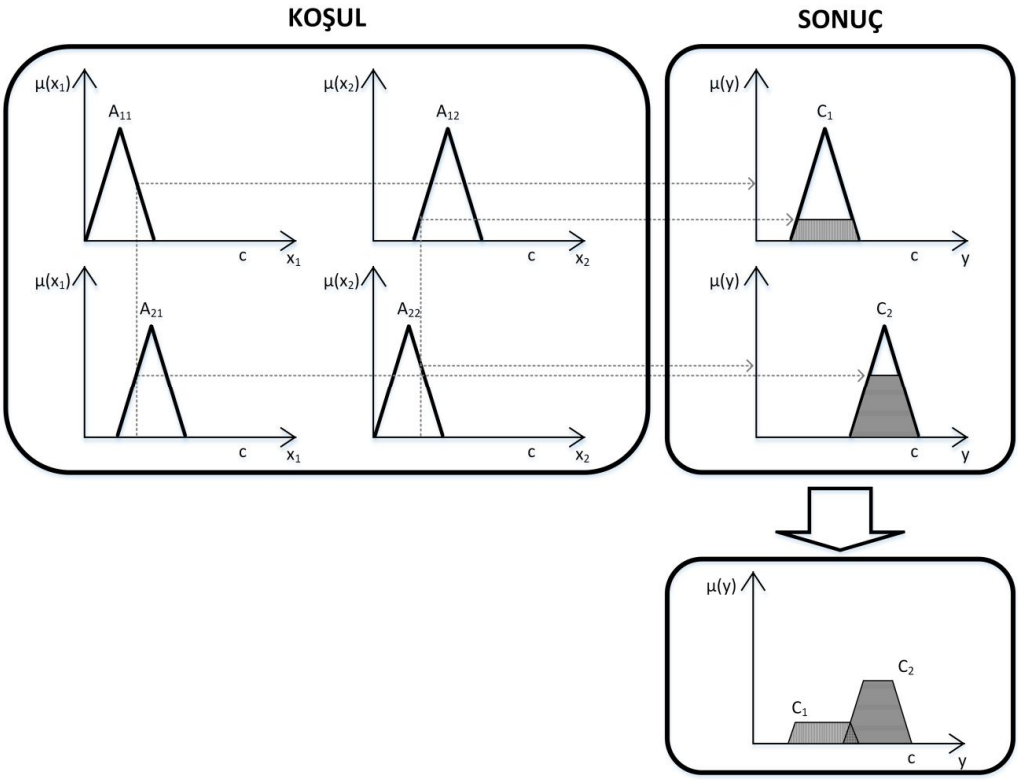
3.1.3. Çıkarım Sistemi

Çıkarım sistemi, bulanık sistemin girişlerine karşılık gelen çıkış değerlerini bulma görevini yapmaktadır. Çıkarım mekanizması modelin tipine göre oluşturulan bulanık kuralları kullanarak çıkış değerini üreten yöntemdir. Literatürde kullanılan farklı pek çok çıkarım mekanizması yöntemi vardır. Bulanık sistemlerde sıklıkla kullanılanlar ise Mamdani (Mamdani ve Assilian, 1975) ve TSK (Takagi ve Sugeno, 1985) tipi çıkarım yöntemleridir.

Mamdani tipi çıkarım yönteminde öncelikli olarak kural kümesindeki her bir kuralın koşul kısımları işlenmektedir. Bu işlem koşul kısmında kullanılan üyelik fonksiyonları ve koşul operatörlerine (VE / VEYA) bağlı olarak yapılmaktadır. Bu işlem sonunda her bir sonuç toplanarak bir bulanık çıkış elde edilmektedir. Bu bulanık çıkış, gerçek bir değere dönüştürülmesi için durulaştırma birimine gönderilmektedir. Örnek olarak iki adet Mamdani tipi bulanık kurala sahip bir sistem düşünelim. Bulanık kurallar 3.8'de ifade edilen şekilde olsun. Buna göre Mamdani çıkarım yöntemi Şekil 3.8'de gösterildiği gibi işlemektedir.

$$\begin{aligned} &\textbf{Kural 1}: EĞER\ x_1 = A_{11}\ \ ve\ \ x_2 = A_{12}\ \ İSE\ \ y = C_1 \\ &\textbf{Kural 2}: EĞER\ x_1 = A_{21}\ \ ve\ \ x_2 = A_{22}\ \ İSE\ \ y = C_2 \end{aligned} \tag{3.8}$$

TSK tipi kuralların sonuç kısımları giriş parametrelerine bağlı fonksiyonlardan oluşmaktadır. Bu nedenle kuralın sonuç değeri bulanık değil kesin bir değerden oluşmaktadır. Kuralın koşul kısmında kullanılan operatörlere bağlı olarak bu kurala ait bir ağırlık değeri oluşturulmaktadır. Daha sonra kuralların ağırlıkları ve kesin çıktıları, ağırlıklı ortalama yöntemi ile tek bir kesin sonuca dönüştürülmektedir. TSK tipi bulanık sistemlerde kural çıktılarının bulanık olmayan değerlerden oluşması nedeniyle durulaştırma işlemine ihtiyaç duymamaktadır. Örnek olarak iki adet TSK tipi bulanık kurala sahip bir sistem düşünelim. Bulanık kurallar 3.9'de ifade edilen şekilde olsun. Buna göre TSK çıkarım yöntemi Şekil 3.9'da gösterildiği gibi işlemektedir.



Şekil 3.8. Mamdani tipi çıkarım mekanizmasının çalışma yöntemi

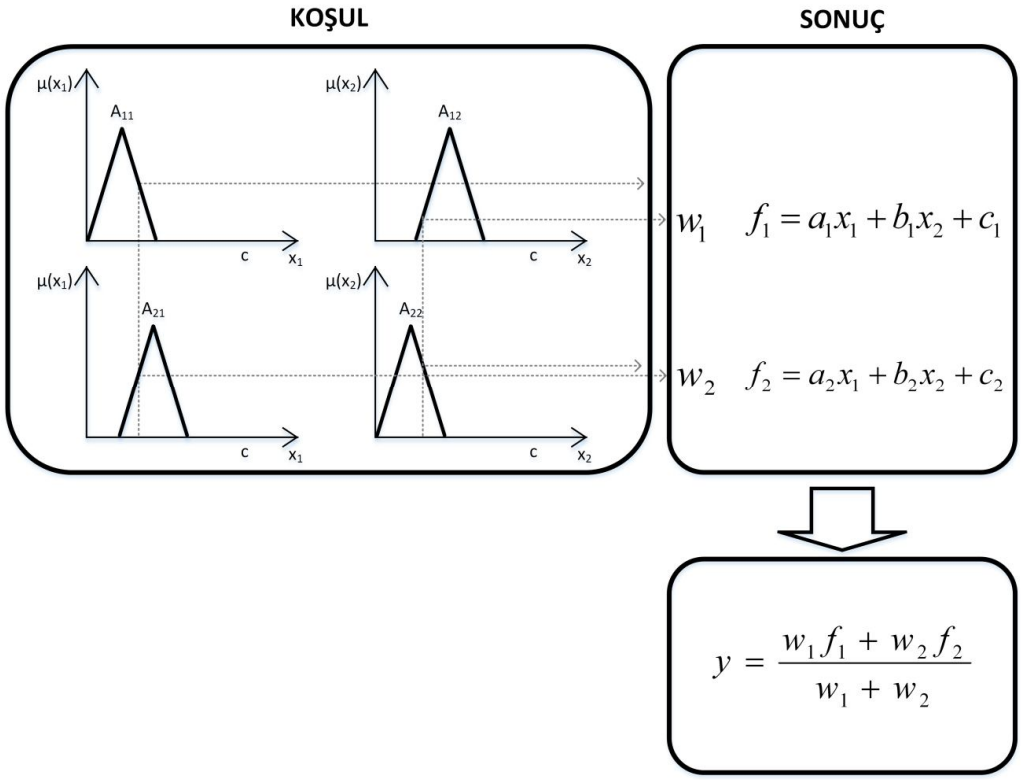
$$\textbf{Kural 1}: E\check{G}ER\ x_1 = A_{11}\ ve\ x_2 = A_{12}\ \text{İ}SE\ y = f_1(x) = a_1x_1 + b_1x_2 + c_1$$
$$\textbf{Kural 2}: E\check{G}ER\ x_1 = A_{21}\ ve\ x_2 = A_{22}\ \text{İ}SE\ y = f_2(x) = a_2x_1 + b_2x_2 + c_2 \quad (3.9)$$

Bunların dışında BKTS sistemlerde tek kazanan (single winner) yöntemi ve ağırlıklı oylama (weighted vote) yöntemi oldukça yaygın olarak kullanılmaktadır. Bu yöntemler özellikle sınıflandırıcı bulanık modellerde etkin olarak kullanılmaktadır. Tek kazanan ve ağırlıklı oylama yöntemleri BKTS sistemler bölümünde detaylı olarak sunulacaktır.

3.1.4. Durulaştırma

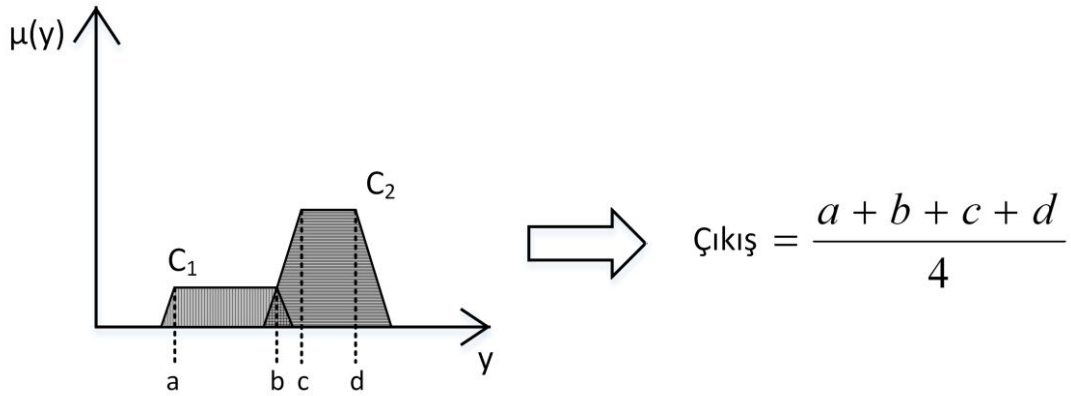
Çoğu kontrol ve sınıflandırma sisteminde sistem çıktısının gerçek bir değer olması istenir. Bulanık sistemlerde, bulanık çıkarım mekanizması tarafından sonuç olarak ortaya konulan bulanık değerin gerçek bir değere dönüştürülmesi gerekmektedir. Durulaştırma işlemi sonuç olarak elde edilen bulanık değeri sistem çıktısı olacak gerçek bir değere dönüştürmektedir (Elmas, 2003). Yaygın olarak kullanılan çıkarım

mekanizmaları; en büyüklerin ortası, ağırlık merkezi ve ağırlıklı ortalama yöntemleridir (Baykal ve Beyan, 2004a).



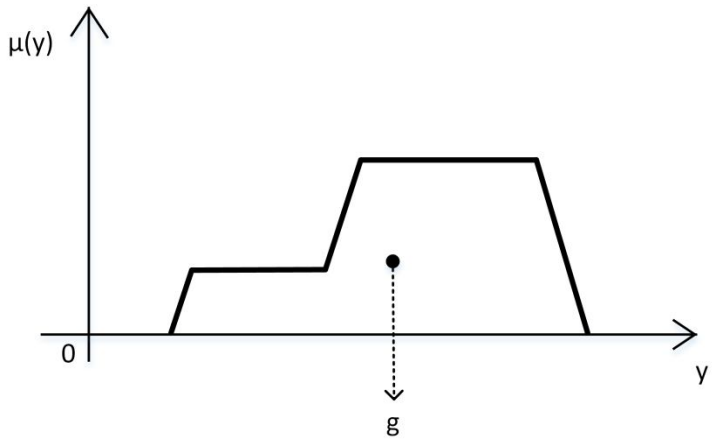
Şekil 3.9. TSK tipi çıkarım mekanizmasının çalışma yöntemi

En büyüklerin ortası yönteminde sonuç çıktısında en büyük üyelik değerini alan değerler toplanır ve ortalaması çıkış değeri olarak kullanılır. Şekil 3.10'da en büyüklerin ortası durulaştırma yöntemine ait bir örnek gösterilmiştir.



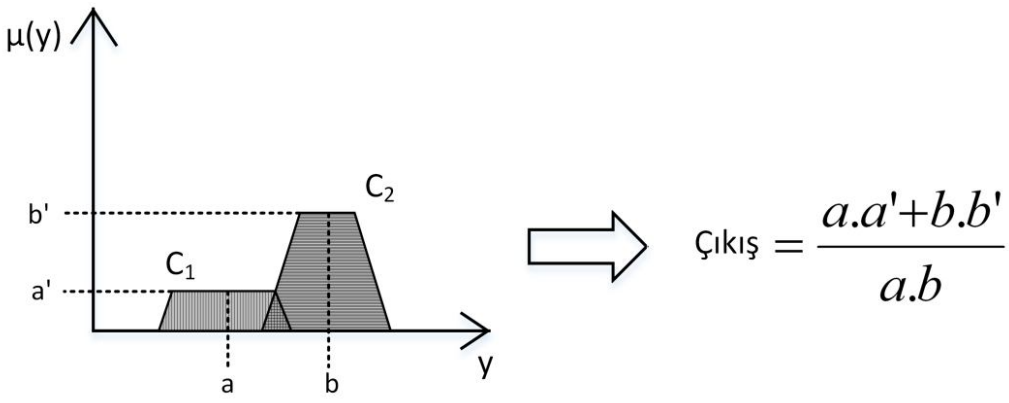
Şekil 3.10. En büyüklerin ortası durulaştırma yöntemi

Ağırlık merkezi yönteminde ise sonuçta oluşan çözüm yüzeyinin ağırlık merkezi çıkış değeridir. En yaygın kullanılan durulaştırma yöntemidir. Her türlü çözüm yüzeyinde kolaylıkla uygulanabilmektedir. Şekil 3.11'de örnek ağırlık merkezi durulaştırma yöntemi gösterilmiştir. "g" değeri yüzeyin ağırlık merkezini göstermekte ve çıkış değeri olarak kullanılmaktadır.



Şekil 3.11. Ağırlık merkezi durulaştırma yöntemi

Ağırlıklı ortalama yönteminde ise çözüm yüzeyini oluşturan her bir üyelik fonksiyonunun en büyük üyelik değeri ve bu değere ulaştığı noktanın değeri çarpılır ve bu değerlerin toplamına bölünerek gerçek çıkış üretilir. Bu yöntemde çözüm yüzeyini oluşturan üyelik fonksiyonları simetrik olmak zorundadır. Şekil 3.12'de örnek ağırlıklı ortalama durulaştırma yöntemi gösterilmiştir.



Şekil 3.12. Ağırlıklı ortalama durulaştırma yöntemi

Bu yöntemler dışında kullanılan farklı durulaştırma yöntemleri de literatürde mevcuttur (Zhang and Edmunds, 1991) (Yamakawa,1993) (Saade ve Diab, 2000). Hangi durulaştırma yönteminin kullanılacağına karar veren bir yaklaşım mevcut değildir. Durulaştırma yönteminin seçimi sistemi tasarlayan kişinin seçimine veya uzman bir kişinin görüşüne bağlı olmaktadır.

3.2. Bulanık Kural Tabanlı Sınıflandırıcı Sistemler

Bulanık sistemler kontrol ve denetim sistemlerinde yaygın olarak kullanılmaktadır. Bulanık sistemlerdeki gelişmelere paralel olarak sınıflandırıcı sistemlerde de kullanımı yaygınlaşmıştır. Sınıflandırıcı sistemler sınıflandıracakları örneği çalışma yapısına göre önceden belirlenmiş sınıflardan herhangi birine dahil etmektedir. Bulanık sınıflandırıcı sistemlerde sınıflandırma işlemini yapabilmek için yine bulanık sistemlerde olduğu gibi veri tabanı, kural tabanı ve çıkarım sisteminin önceden belirlenmesi gerekmektedir.

Bulanık kural tabanlı sistemlerde sınıflandırılacak örnek veri çıkarım sistemi tarafından veri tabanı ve kural tabanı kullanılarak bir sınıf değeri çıktısına dönüştürülmektedir. Bulanık sistemlerin genel yapısındaki bulanıklaştırma ve durulaştırma işlemleri çıkarım sistemi ile bütünleşik olarak çalışmaktadır. Bulanık sınıflandırıcılarda sınıflandırmayı etkileyen en büyük faktörün bulanık kurallar olması nedeniyle BKTS sistemler olarak da adlandırılmaktadırlar. Sınıflandırma işlemi sonunda örneğe karşılık gelen bir sınıf değeri bulunacaktır. Sınıf değerleri sözsel olarak ifade edilebilmektedirler. Bu nedenle BKTS'lerde çıkış değeri sürekli reel değerlerden oluşan TSK tipi bulanık kurallar yerine çıkış değeri sözsel değerlerden oluşan Mamdani tipi bulanık kural kullanılmaktadır. BKTS sistemlerde tek kazanan yöntemi ve ağırlıklı oylama yöntemi oldukça yaygın olarak kullanılmaktadır. BKTS sistemlerin temelini bulanık kural kümesi ve çıkarım mekanizması oluşturmaktadır.

3.2.1. Bulanık Kural Kümesi

BKTS sistemlerde bulanık kural kümesi 3.6'da gösterilen Mamdani tipi bulanık kurallardan oluşmaktadır. Fakat BKTS'lerde farklı olarak her bir bulanık kural için bir kural ağırlığı kavramı ortaya çıkmaktadır. Bulanık kuralın ağırlığı 0 ile 1 arasında bir reel değerdir. Bir bulanık kuralın ağırlık değeri 1'e yakın ise sınıflandırmadaki etkinliği

daha fazla, 0'a yakın ise etkinliği daha azdır. Bulanık kuralın ağırlık değeri 0 ise sınıflandırmada herhangi bir etkinlik değeri yoktur. n tane giriş verisi olan ve T adet sınıf değerine sahip bir çıkışı olan bir veri kümesi için bulanık kuralın gösterimi 3.10'da gösterilmiştir.

$$K_i\text{: } E\check{G}ER\ x_1 = A_{i1}\ \ ve\ \ x_2 = A_{i2}\ \ ve \ldots ve\ \ x_n = A_{in}\ \ \text{İ}SE\ \ Sınıf = C_t, CF \quad (3.10)$$

3.9'da gösterilen bulanık kuralda K_i bulanık kural kümesindeki i inci bulanık kuralı temsil etmektedir. x_k, k ıncı giriş parametresinin değerini ve A_{ik}, k ıncı giriş parametresinin hangi üyelik fonksiyonu ile temsil edileceğini göstermektedir. C_t, bulanık kuralın hangi sınıf değerine sahip olduğunu göstermektedir. CF ise bulanık kuralın ağırlık değerini temsil etmektedir (Ishibuchi, 2005a).

Bulanık kural kümesindeki kuralların oluşturulması bir uzman tarafından yapılabileceği gibi akıllı bilgisayar sistemleri tarafından da yapılabilmektedir. Literatürde bu konuda bir çok çalışma bulunmaktadır. Bu çalışmalar içerisinde genetik algoritmalarla yapılan pek çok başarılı çalışma bulunmaktadır. Genetik algoritmaların kullanımının çok yaygın olması nedeniyle bu çalışmalar Genetik Bulanık Sistemler başlığı altında toplanmıştır.

3.2.2. Tek Kazanan Çıkarım Yöntemi

Tek kazanan çıkarım yönteminde, sınıflandırılacak yeni örneğe ait her kural için bir performans indeksi bulunmaktadır. En büyük performans indeksine sahip kural hangi sonuç sınıfını temsil ediyorsa yeni örnek o sınıfa dahil edilir. Eğer kural kümesinde sonuç sınıf değerleri farklı ve en yüksek performans indeksine sahip birden fazla kural bulunursa sınıflandırıcı bu örneği sınıflandıramaz. Tek kazanan çıkarım yönteminde kuralın performans indeksi, yeni örneğin her bir giriş değerinin temsil edildiği üyelik fonksiyonu değerlerinin çarpımı ile kural ağırlığının çarpımı yapılarak bulunmaktadır. Buna göre tek kazanan çıkarım yöntemi 3.11 ve 3.12 de ifade edilmekte dedir (Ishibuchi, 1999a).

$$\mu(x) = \mu(x_1) \times \mu(x_2) \times \ldots \times \mu(x_n) \tag{3.11}$$

$$\mu_i(x) \times CF_i = \max\{\mu_k(x) \times CF_k \mid K_k \in S\} \tag{3.12}$$

3.11 de ki $\mu(x)$ ifadesi aynı zamanda bir kuralın uyumluluk değerini göstermektedir. S kural kümesini göstermektedir. $\mu_i(x) \times CF_i$ ifadesi elde edilen en yüksek performans indeksi değerinin *i* inci kurala ait olduğunu göstermektedir. Bu nedenle yeni örnek C_i sınıfına dahil edilmektedir.

3.2.3. Ağırlıklı Oylama Çıkarım Yöntemi

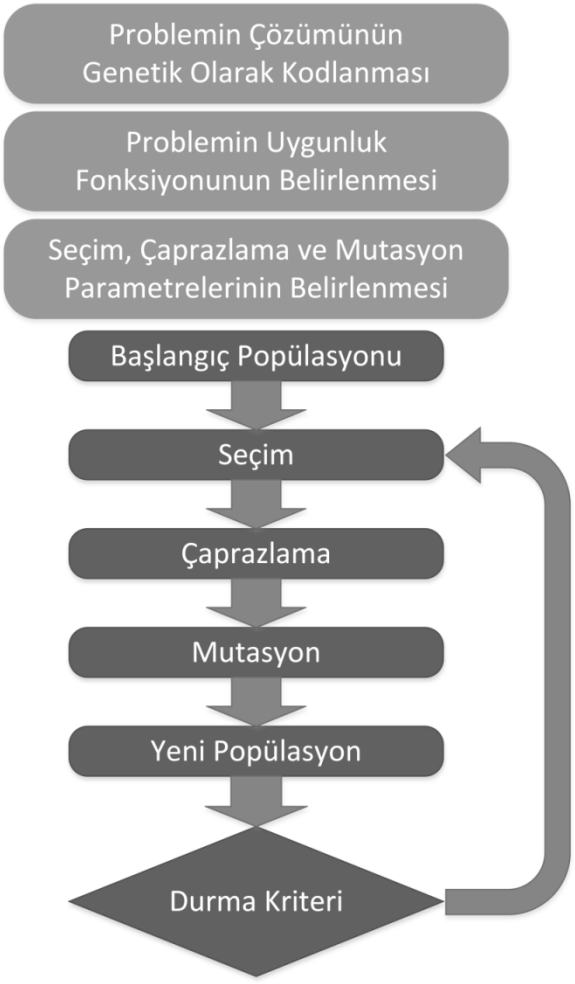
Ağırlıklı oylama yönteminde bulanık kural kümesindeki bulanık kurallar sınıf değerlerine göre gruplandırılmakta ve her sınıf grubunun toplam performans indeksi hesaplanmaktadır. En yüksek performans indeksine sahip gurup hangi sınıf değerini temsil ediyorsa yeni örnek o sınıfa dahil edilmektedir. İki farklı sınıf grubu en yüksek performans değerine sahip olursa sınıflandırıcı bu örneği sınıflandıramaz. Bir grubun performans indeksi o gruptaki kuralların uyumluluk değerleri ile ağırlık değerlerinin çarpımının toplamı ile elde edilmektedir. *T* sınıf değerinin oluşturduğu grubun performans indeksi 3.13 de gösterilmiştir (Ishibuchi, 1999a).

$$PI_T = \sum_{\substack{K_k \in S \\ C_t = T}} \mu_t(x) \times CF_t \tag{3.13}$$

Tek kazanan çıkarım yönteminde yeni örneğin sınıf değerini baskın olan kural belirlemektedir. Ağırlıklı oylama çıkarım yönteminde ise kuralın sınıf değerini baskın olan sınıf kural grubu belirlemektedir.

4. GENETİK ALGORTİMALAR

Genetik algoritmaların temelleri ilk kez Holland (1975) tarafından ortaya atılmıştır. Daha sonra Holland'ın öğrencisi olan Goldberg (1989) ortaya konulan bu teorik algoritmayı çeşitli uygulamalar ile desteklemiştir. Bu uygulamaları "Genetic Algorithms in Search, Optimization and Machine Learning" isimli bir kitapta toplamıştır. Bu başarılı uygulamalar genetik algoritmanın tanınırlığının artmasında önemli rol oynamıştır.



Şekil 4.1. Genetik Algoritmanın temel çalışma yapısı

Genetik algoritmalar Darwin'in doğal seçim ve evrim ilkelerini temel almaktadır. Bu anlamda algoritma içerisindeki bireylerden güçlü olanlar kullanılarak yeni bireyler topluluğu oluşturulmaktadır. Topluluktaki en iyi bireyler ayakta kalırken aynı zamanda iyi bireylerin özelliklerini taşıyan daha iyi bireyler oluşturulması

hedeflenmektedir. Genetik algoritmalar bilgisayar bilimleri, kimya, matematik, fizik gibi birçok bilim alanlarda etkin bir şekilde kullanılmaktadır. Genetik algoritmaların temel çalışma şekli; problem için tek bir çözüm üretmek yerine birden fazla çözüm üreterek daha geniş bir çözüm havuzu oluşturmak ve daha etkin bir arama yapmaktır (Li ve ark., 2009).

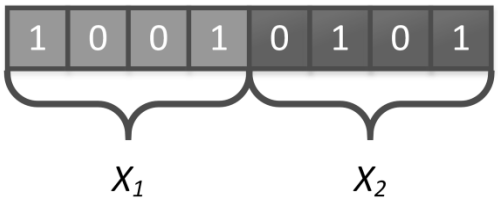
Genetik algoritmalar her biri bir çözümü temsil eden bireylerden oluşan bir topluluk üzerinde çalışmaktadır. Bu bireylerin oluşturduğu topluluğa popülasyon denilmektedir. Algoritmanın her adımda var olan popülasyondan daha iyi bir popülasyon üretmeyi amaçlamaktadır. Genetik algoritmada her bir adım veya diğer tanımıyla iterasyon, bir popülasyondan yeni bir popülasyonun oluşturulmasına kadar geçen süreçtir. Bu sürece başlamadan önce yapılması gereken ilk adım, uygunluk fonksiyonu, bireyin kodlanması ve operatör parametrelerinin belirlenmesidir. Daha sonra var olan popülasyon üzerinde sırasıyla seçim, çaprazlama ve mutasyon operatörleri kullanılarak yeni popülasyon üretilmektedir. Bu süreç, belirlenen bir durma kriteri sağlanana kadar devam ettirilmekte ve sonunda en iyi birey çözüm olarak kabul edilmektedir (Marwala ve Chakraverty, 2006). Şekil 4.1.'de genetik algoritmalara ait temel çalışma yapısı verilmiştir.

4.1. Problemin Genetik Olarak Kodlanması

Problemin genetik olarak kodlanması işlemi, problemin çözümünün genetik bir birey olarak sunulması işlemidir. Bir birey veya diğer adıyla kromozom genlerden oluşmaktadır. Bir gen bireyin belirli bir özelliğini temsil etmektedir. Örneğin denklem 4.1'deki gibi bir minimizasyon problemimiz olduğunu düşünelim. Problemin amacı y değerini minimum yapan x_1 ve x_2 değerlerini bulmaktır. Bu parametreler ikili kodlama yöntemi ile bir birey olarak sunulabilmektedir.

$$y = x_1^2 + x_1.x_2 + x_2^2 \quad x_1, x_2 \in Z, 0 < x_1 < 16, 0 < x_2 < 16 \tag{4.1}$$

x_1 ve x_2 parametrelerinin alabileceği maksimum değer 16 dan küçük olacağı için ikili olarak 4 bit ile ifade edilebilir ve iki parametremiz olduğundan dolayı bireyin toplam uzunluğu 8 bit olarak gerçekleşir. Problem için yapılan örnek kodlama Şekil 4.2.'de gösterilmiştir.



Şekil 4.2. Denklem 4.1.'deki problem için örnek birey kodlaması

Bireyin genetik olarak kodlanmasında ikili sayılar kullanılabildiği gibi gerçek değerli sayılarda kullanılabilmektedir. Çaprazlama ve mutasyon operatörleri seçilen kodlama yapısına uygun olmalıdır.

4.2. Uygunluk Fonksiyonunun Belirlenmesi

Uygunluk fonksiyonu bir bireyin çözüme ne kadar yaklaştığını ölçmektedir. Başka bir deyişle bireyin çözüm açısından kalitesini göstermektedir. Uygunluk fonksiyonu aynı zamanda genetik algoritmanın amacını da göstermektedir. Çözüme en yakın bireyin uygunluk değerinin en yüksek olması gerekmektedir. Denklem 4.1.'deki minimizasyon probleminde y değerini 0 yapan bireyin uygunluk değeri yüksek olmalıdır. Uygunluk fonksiyonunun verimli bir şekilde çalışması genetik algoritmanın başarısını doğrudan etkilemektedir.

4.3. Seçim

Popülasyondaki bireylerin uygunluk değerleri hesaplanır ve bireyler uygunluk değerlerine göre sıralanır. Bu noktada farklı yöntemler kullanılarak kaliteli bireylerden oluşan bir havuz oluşturulur. Kaliteli bireylerden oluşan bu havuz yeni bireyler oluşturmak için çaprazlama ve mutasyon işlemine tabi tutulmaktadır. Literatürde en sık kullanılan seçme yöntemleri rulet tekerleği yöntemi, rank yöntemi, turnuva yöntemi ve elitist seçim yöntemidir.

$$Rulet\ Tekerleği\ Değeri = \frac{Bireyin\ Uygunluğu}{Populasyonun\ Toplam\ Uygunluğu} \tag{4.2}$$

Rulet tekerleği seçme yönteminde bireylerin uygunluk değeri hesaplandıktan sonra büyükten küçüğe doğru sıralanmaktadır. Daha sonra her bireye ait rulet tekerleği değeri 4.2'deki denklem yardımı ile hesaplanmaktadır. Bu aşamadan sonra Rulet tekerleği değerlerinin kümülatif toplam değerleri oluşturulmaktadır. 0 ile 1 arasında rasgele bir sayı üretilmekte ve bu sayı sıralamanın aşağısından başlayarak kümülatif değerler ile karşılaştırılmaktadır. Rastgele değer hangi bireyin kümülatif değerinden düşük ise o birey seçim havuzuna aktarılır. Böylece uygunluğu yüksek bireylerin seçilme oranı artarken düşük bireylerin seçilme oranı daha düşük olmaktadır.

$$Rank\ Değeri = \frac{(T+1) - i}{(T \times (T+1))/2} \tag{4.3}$$

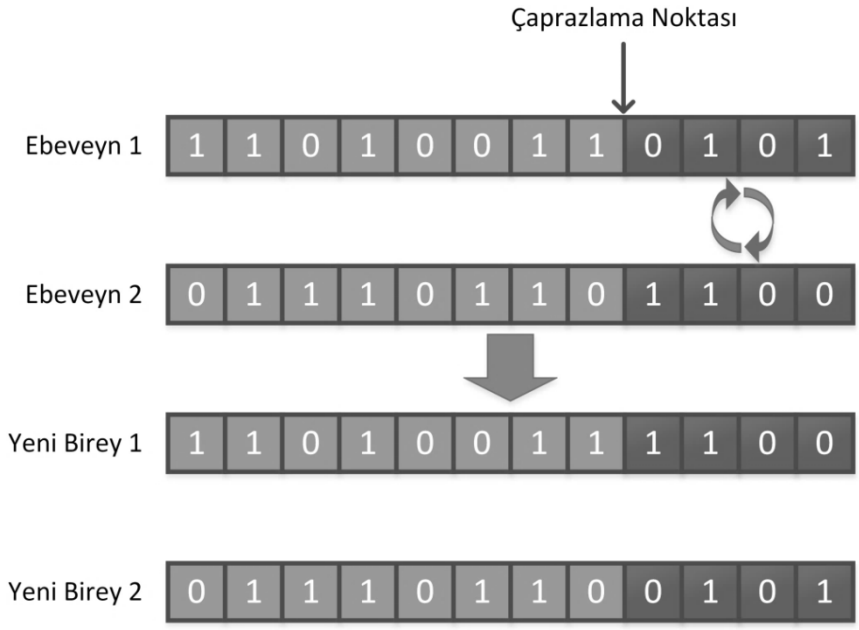
Rank yönteminde ise bireylerin uygunluk değerlerine bağlı olarak bir rank değeri hesaplanmaktadır. Rulet değeri 4.3'deki denklem ile hesaplanmaktadır. Denklemde T toplam birey sayısını, i ise bireyin sıralamadaki sayısıdır. Rank değerlerinin kümülatif değerleri hesaplanmakta ve rastgele üretilen değer ile kümülatif değerler kullanılarak birey seçimi yapılmaktadır.

Turnuva seçim yönteminde popülasyondan rasgele seçilen bireyler turnuvaya sokulur ve uygunluk değerine göre turnuvayı kazanan birey belirlenir. Belirlenen birey seçim havuzuna eklenir. Bu yöntemde en önemli parametre turnuvanın kaç defa yapılacağıdır.

Elitist seçim yönteminde uygunlukları hesaplanan bireylerden en iyi uygunluğa sahip birey veya belirli bir değerin üzerinde uygunluğa sahip bireyler seçim havuzuna doğrudan eklenmektedir.

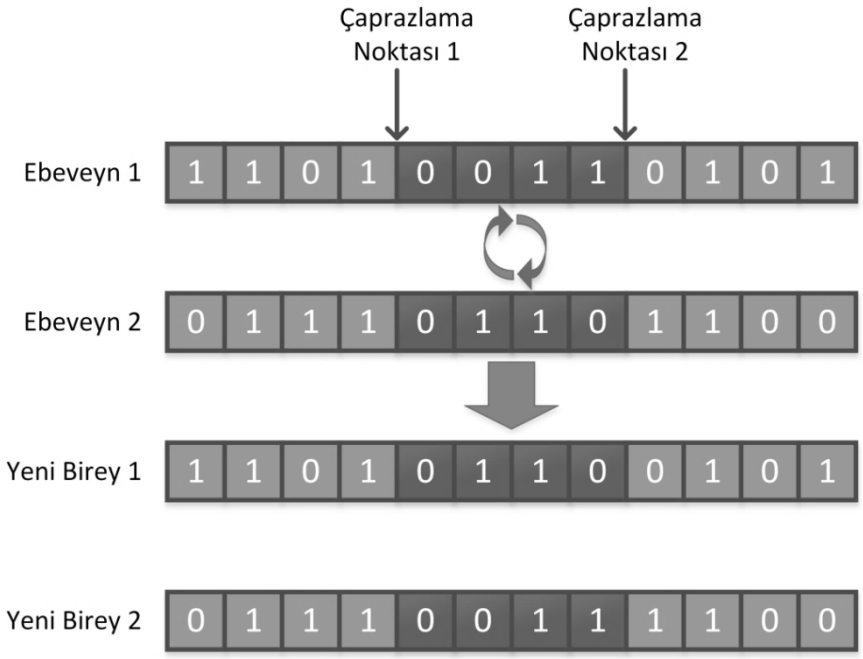
4.4. Çaprazlama

Seçim havuzunda bulunan bireyler yeni popülasyonu oluşturacak olan ebeveyn bireylerdir. Bu seçilmiş bireylerden yeni bireyler oluşturmadaki en önemli adım çaprazlama adımıdır. Çaprazlama işlemi eşleştirilen iki bireyin belirlenen yönteme göre genlerinin yer değiştirmesi işlemidir. Böylece seçilmiş güçlü bireylerin özeliklerini taşıyan daha güçlü bireyler oluşturulması amaçlanmaktadır. Literatürde yaygın olarak kullanılan çaprazlama yöntemleri tek noktalı ve iki noktalı çaprazlama yöntemleridir.



Şekil 4.3. Örnek tek noktalı çaprazlama işlemi

Tek noktalı çaprazlama işleminde belirlenen bir noktadan itibaren seçilen her iki bireyin genlerinin yer değişmesi işlemidir. Şekil 4.3'de örnek tek noktalı bir çaprazlama işlemi gösterilmiştir.

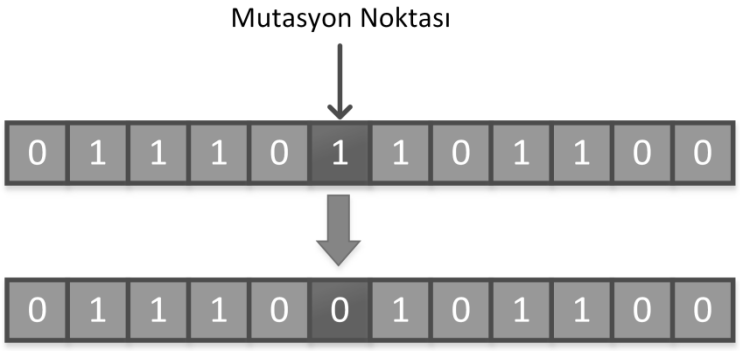


Şekil 4.4. Örnek iki noktalı çaprazlama işlemi

İki noktalı çaprazlama işleminde ise belirlenen başlangıç ve bitiş noktaları arasındaki bilgiler karşılıklı olarak değiştirilmektedir. Şekil 4.4'de örnek iki noktalı çaprazlama işlemi gösterilmiştir.

4.5. Mutasyon

Mutasyon işlemi popülasyondaki bir bireyin bir özelliğinin değişmesini ifade etmektedir. Bu değişim küçük bir ihtimal dahilinde olmaktadır. Mutasyon operatörü sayesinde genetik algoritmanın, arama işlemi sürecinde yerel minimum noktalara takılma ihtimali azaltılmaktadır. İkili kodlanmış bir bireyin bir özelliğinin mutasyon geçirmesi demek özelliğin değeri 1 iken 0'a veya 0 iken 1'e dönüşmesi demektir. Şekil 4.5'de örnek bir birey üzerinde mutasyon işlemi gösterilmiştir.



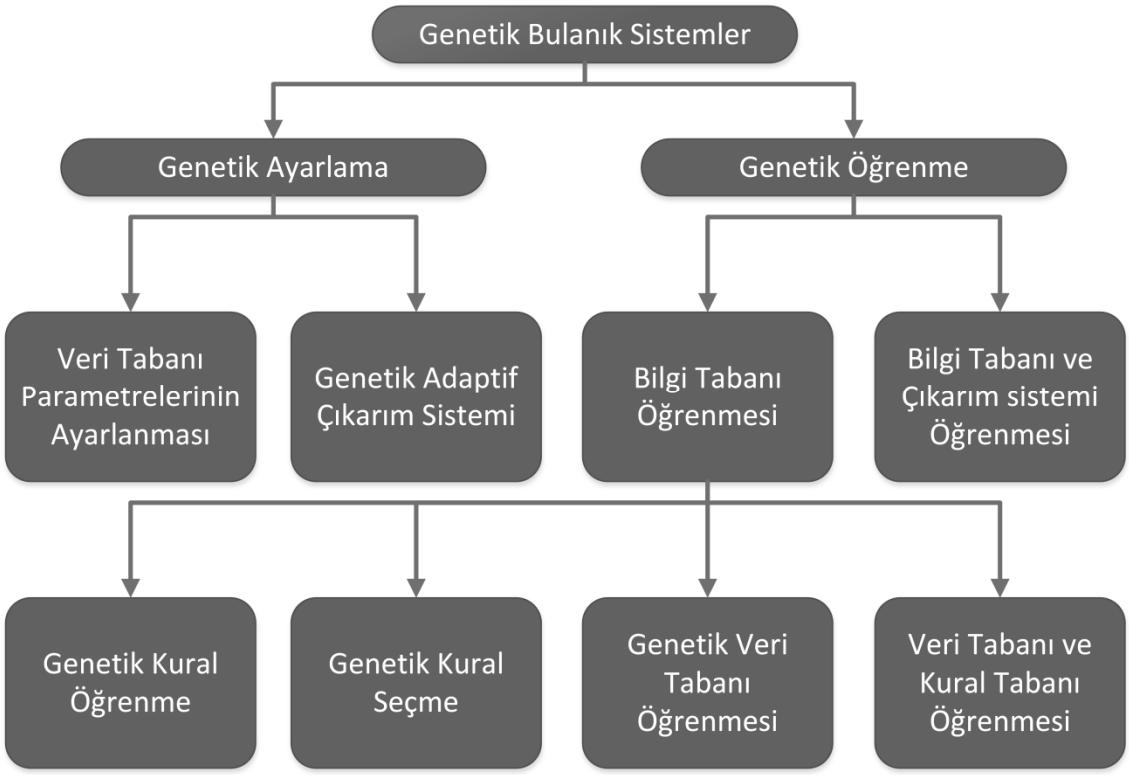
Şekil 4.5. Örnek mutasyon işlemi

4.6. Durma Kriteri

Genetik algoritma yukarıda bahsedilen adımları gerçekleştirerek daha iyi uygunluk değerlerine sahip yeni bir popülasyon oluşturmayı amaçlamaktadır. Bu işlemi tekrar tekrar yaparak problem için en iyi çözümü aramaktadır. Fakat bu işlem sonsuza kadar devam edemez. Bu nedenle bir durma kriterine ihtiyaç duymaktadır. Farklı problemler için farklı duruma kriterleri tanımlanabilmektedir. En yaygın kullanılan durma kriterleri; algoritmanın belirli bir adım sonrasında bulduğu en iyi çözümü çıkış olarak vermesi, belirlenen bir çözüm değerine ulaşılana kadar devam etmesi veya algoritmanın artık daha iyi bir çözüme ulaşamaması durumunda algoritma sonlandırılabilmektedir.

5. GENETİK BULANIK SİSTEMLER

BKTS sistemlerde sınıflandırmanın kalitesindeki en önemli etken sınıflandırıcıya ait bulanık kural kümesindeki kurallardır. Bulanık kural kümesinin oluşturulması, BKTS sistemlerin tasarlanmasındaki en önemli aşamadır. Bulanık kurallar, ilk başlarda sistemin çalışacağı alanda bir uzman yardımı ile oluşturulmaktaydı. Fakat ilgi alanda uzman bir kişinin bulunması her zaman mümkün olmadığı gibi kişinin uzmanlık derecesinin yeterliliği de tartışma konusu olabilmektedir. Bu nedenle bulanık kuralların oluşturulmasına yönelik çalışmalar araştırmacıların ilgisini çekmiştir. Bulanık kuralların oluşturulması ile alakalı pek çok sezgisel yöntem ortaya konulmuştur (Herrera, 2008). Yapay zekâ ve akıllı bilgisayar sistemlerindeki gelişmeler bu alanda da kendini göstermiştir. Özellikle bulanık kuralların oluşturulması bir optimizasyon problemi olarak ele alınmış ve etkili pek çok optimizasyon algoritması bu amaç için kullanılmıştır.



Şekil 5.1. Genetik bulanık sistemlerin temel sınıflandırma yapısı

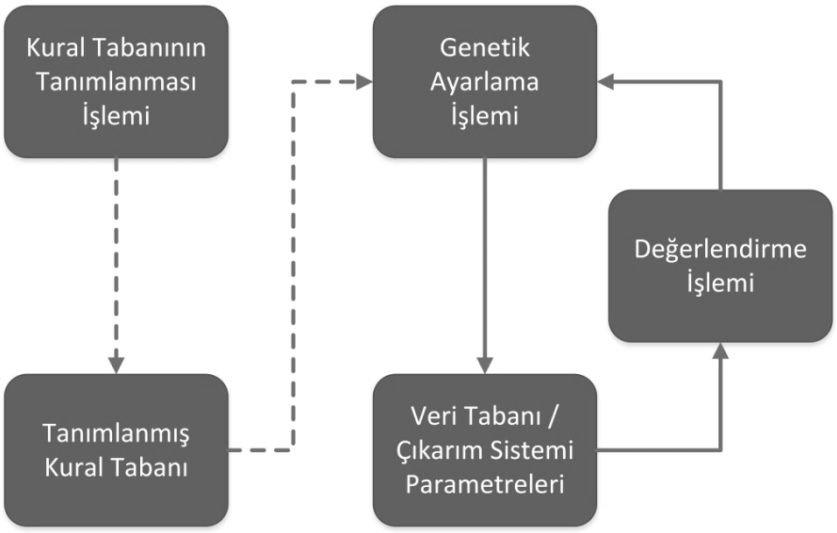
Genetik algoritmalar evrimsel süreçlerden esinlenerek geliştirilmiş bir optimizasyon algoritmasıdır ve geniş çözüm uzayına sahip optimizasyon

problemlerinde etkin olarak arama yapabilmektedir. Bulanık kuralların oluşturulması da bir optimizasyon problemi olduğu için genetik algoritmalar ve BKTS sistemler birlikte fazlaca anılmaktadır. Bu nedenle literatürde genetik algoritma kullanılan bulanık kural tabanı sistemler, genetik bulanık sistemler (*Genetic Fuzzy Systems-GFS*) adıyla anılmaktadır (Cordon ve ark., 2001).

Herrera (2008) genetik bulanık sistemler üzerine yaptığı bir değerlendirme çalışmasında genetik algoritmaların kullanılma şekillerine göre sınıflandırmasını yapmıştır. Şekil 5.1'de genetik algoritmaların kullanım amaçlarına göre genetik bulanık sistemler sınıflandırılmıştır. Genetik bulanık sistemler temel olarak iki amaç için kullanılmaktadır. Bunlar bulanık sistemlerdeki parametrelerin ayarlanması (*Genetic Tuning*) ve bulanık sistem üzerinde öğrenme (*Genetic Learning*) yaklaşımıdır.

5.1. Genetik Ayarlama

Genetik algoritmalar bulanık sistemlerin parametrelerinin ayarlanmasında kullanılabilmektedir. Bu yaklaşımlarda bulanık sistem önceden tanımlanmış bir bulanık kural kümesine sahiptir. Genetik algoritma değerlendirme işleminden dönen sonuca bağlı olarak veri tabanı ve çıkarım mekanizmasındaki sistem parametrelerini, uygunluk fonksiyonunda ortaya konulan amaca uygun şekilde ayarlamaktadır.



Şekil 5.2. Genetik ayarlama modellerinin genel çalışma yapısı

Şekil 5.2'de genetik algoritmaların bulanık sistemin parametrelerinin ayarlanmasında kullanılan modellerin genel çalışma yapısı gösterilmiştir. Genetik

algoritmanın, bulanık sistemin parametrelerinin ayarlanmasında kullanılması iki ana başlık altında toplanmaktadır. Bunlar veri tabanı parametrelerinin ayarlanması ve çıkarım sisteminin adaptif olarak ayarlanmasıdır. Veri tabanı parametrelerinin ayarlanması modellerinde genellikle bulanık sistemde kullanılacak üyelik fonksiyonlarının sayısı, şekli ve bu üyelik fonksiyonlarının parametrelerinin belirlenmesi amaçlanmaktadır. Parametrelere bağlı olarak kullanılan çıkarım sisteminin parametrelerin ayarlanmasında da kullanılmaktadır.

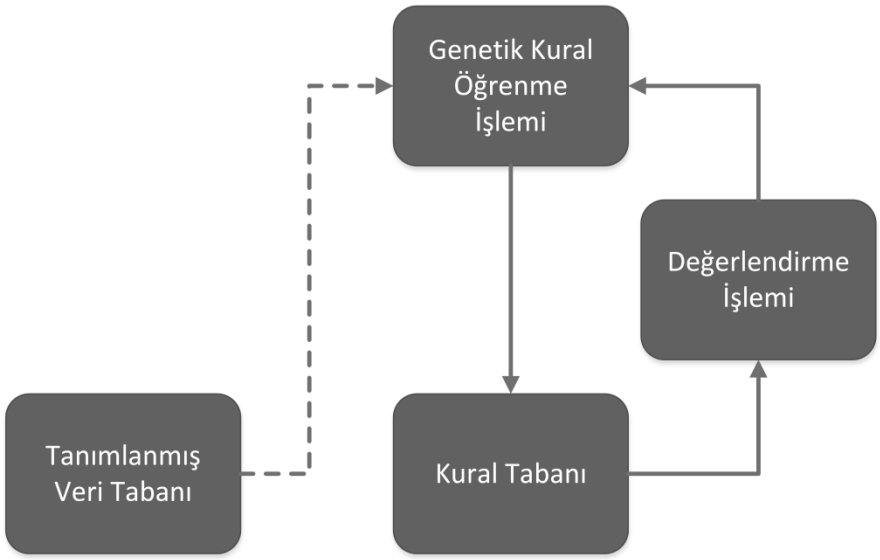
5.2. Genetik Öğrenme

Genetik algoritmanın öğrenme amaçlı kullanıldığı yaklaşım ise daha çok kural tabanı ve buna bağlı çıkarım sisteminin oluşturulması ile ilgilidir. İki başlık altında incelenmiştir. Bilgi tabanın öğrenilmesi ve bilgi tabanı ile beraber çıkarım sisteminin de dahil olduğu bütünleşik sistemlerin oluşturulmasıdır. Bilgi tabanı ve çıkarım sisteminin öğrenilmesi modeli aslında adaptif çıkarım sistemi ile bilgi tabanının öğrenilmesi yöntemlerinin birleştirilerek kullanılması yaklaşımıdır.

En yaygın kullanılan öğrenme yaklaşımı bilgi tabanın öğrenilmesi yaklaşımıdır. Bu yaklaşım da 4 alt yaklaşımdan oluşmaktadır. Kuraların öğrenilmesi, kuralların seçilmesi, veri tabanın öğrenilmesi ve veri tabanı ve kural tabanının öğrenilmesi yaklaşımlarıdır.

5.2.1 Kural Öğrenme Yaklaşımı

Genellikle bulanık kural tabanlı sistemlerde kullanılmaktadır. Sayısal veri kümesi kullanılarak bulanık sistem için ideal kural kümesinin oluşturulmasını sağlamaktadır. Bu modelde veri tabanı içerisinde bulunan giriş ve çıkış parametrelerine ait üyelik fonksiyonlarının şekli ve parametreleri önceden tanımlanmaktadır. Bu tanımlamalara bağlı olarak genetik algoritma olası kural kümeleri oluşturmakta ve Bu kural kümelerinin değerlendirilmesi yapılmaktadır. Genetik algoritma belirli bir durma kriterinin sonunda elde ettiği en iyi kural kümesini çözüm olarak sunmaktadır. Şekil 5.3'de genetik algoritmaların bulanık kural kümesinin öğrenilmesi yaklaşımının genel çalışma yapısı gösterilmiştir.

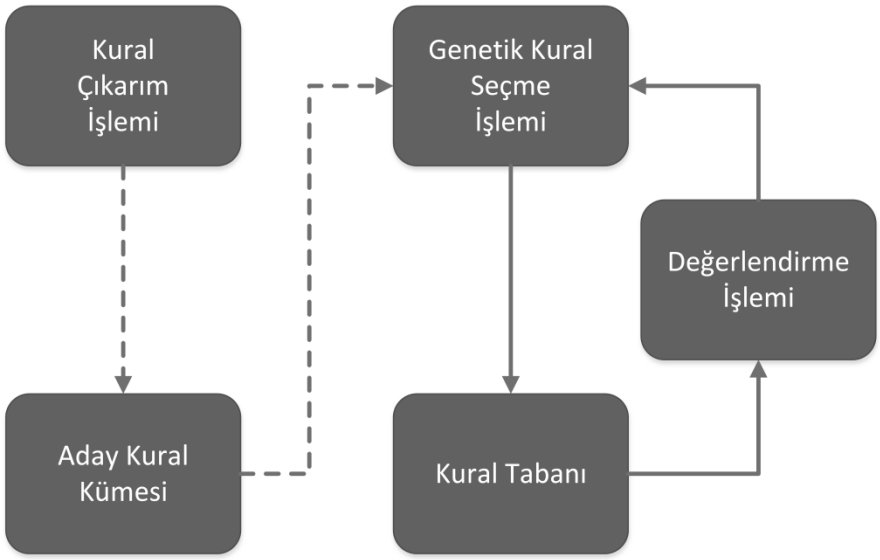


Şekil 5.3. Genetik kural öğrenme modellerinin genel çalışma yapısı

Kural öğrenme yaklaşımındaki en önemli nokta kural kümesinin genetik olarak nasıl sunulduğudur. Kuralın sunulmasının yanında kural öğrenme işlemi sırasında kural çeşitliliğini artırmak ve daha iyi bir arama yapabilmek için genetik algoritmaların operatörlerindeki modifikasyonlarda önemli bir adımı oluşturmaktadır. Literatürde bu özellikleri içerisinde barındıran başarılı pek çok kural öğrenme yöntemi mevcuttur (Kaya, 2006) (Thrift, 1991).

5.2.2 Kural Seçme Yaklaşımı

Bu yaklaşımda bulanık sistem için ideal kural kümesini bulma işini; önceden belirlenmiş bir çerçevedeki aday kurallar arasından seçme işlemi şeklinde yapmaktadır. Genetik bulanık sistemlerde, bulanık sistemin giriş parametre sayısı ve üyelik fonksiyonu sayısına bağlı olarak olası kural sayısı çok fazla olabilmektedir. Bu durumlarda olası kural kümesi içerisindeki bazı kurallar fazlalık veya diğer kurallar ile çakışan ve kararlı olmayan kurallar olabilmektedir. Kural seçme yaklaşımında bu kurallardan kurtulmak için ön kural çıkarım yöntemi kullanılmaktadır. Ön işlem sonrasında elde edilen aday kural kümesi içerisinde ideal kural kümesi genetik algoritmalar yardımı ile bulunmaktadır. Bu yaklaşımın genel çalışma yapısı Şekil 5.4'de gösterilmektedir.



Şekil 5.4. Genetik kural seçme modellerinin genel çalışma yapısı

Kural seçme yaklaşımında, arama uzayı ön işlem sayesinde küçüldüğü için genetik algoritma daha verimli bir şekilde çalışabilmektedir. Fakat aday kural kümesinin çıkarılması ön işlemi modele ek bir maliyet getirmektedir. Ayrıca aday kural kümesinin oluşturulması için seçilen yöntemin etkinliği, yöntemin başarısını direk olarak etkilemektedir.

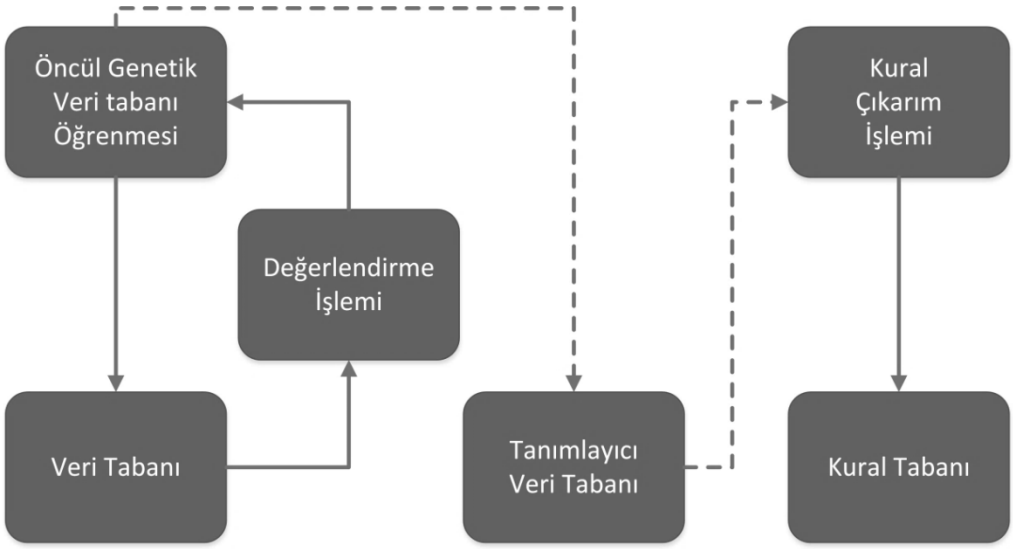
5.2.3. Veri Tabanı Öğrenmesi Yaklaşımı

Bu yaklaşımda; üyelik fonksiyonlarının sayısı ve şekli gibi veri tabanında bulunan bulanık modelin yapısına göre farklılıklar gösteren parametrelerin belirlenmesi yaklaşımıdır. Bu yaklaşımda iki farklı model kullanılmaktadır. Bunlar "Öncül genetik veri tabanı öğrenmesi" ve "Gömülü genetik veri tabanı öğrenmesi" modelleridir.

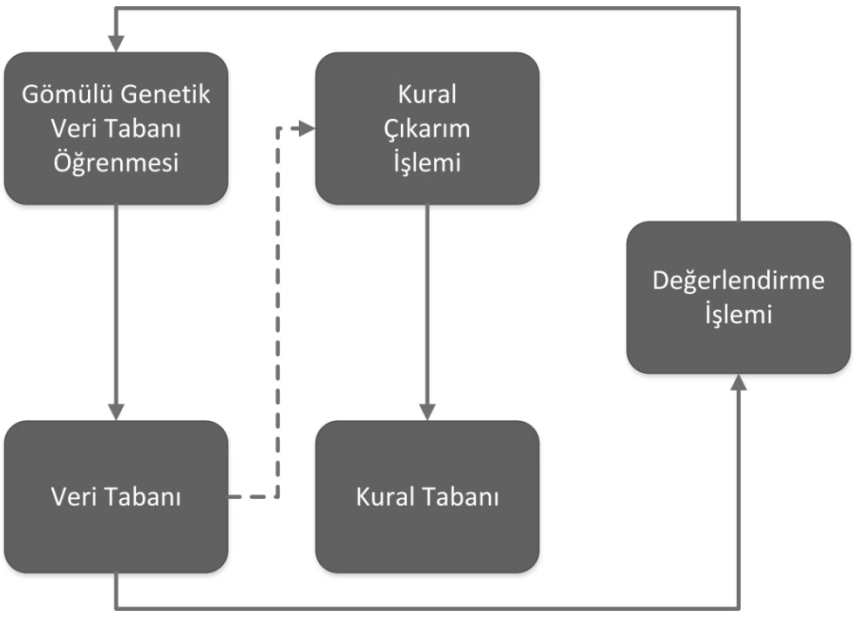
Öncül genetik veri tabanı öğrenme modelinde genetik algoritma ideal veri tabanını arama işlemini gerçekleştirmektedir. Genetik algoritma yalnızca veri tabanını değerlendirmeye almaktadır. Bulanık kurallar ise genetik algoritma tarafından oluşturulan belirleyici veri tabanı kullanılarak oluşturulmaktadır.

Gömülü genetik veri tabanı öğrenme modelinde ise genetik algoritma her veri tabanı çözümünde bu veri tabanına bağlı olarak bir kural tabanı oluşturma işlemi gerçekleştirmektedir. Değerlendirme işlemi bu veri tabanı ve kural tabanı kullanılarak gerçekleştirilmektedir.

İki model arasındaki en temel fark öncül modelde değerlendirme işlemi yalnızca veri tabanı için gerçekleştirilirken gömülü modelde elde edilen veri tabanı kullanılarak oluşturulan kural tabanı da değerlendirme işleminde dikkate alınmaktadır. Öncül modelde öncelikle genetik algoritma ideal veri tabanını oluşturmakta ve kural tabanı bu veri tabanı kullanılarak ortaya konulmaktadır. Gömülü modelde ise hem veri tabanı hem de buna bağlı kural tabanı beraber aranarak ideal veri tabanı oluşturulmaya çalışılmaktadır. Öncül ve gömülü genetik veri tabanı öğrenme modellerinin genel çalışma yapısı Şekil 5.5 ve Şekil 5.6'da sırasıyla gösterilmektedir.



Şekil 5.5. Öncül genetik veri tabanı öğrenmesi modelinin genel çalışma yapısı

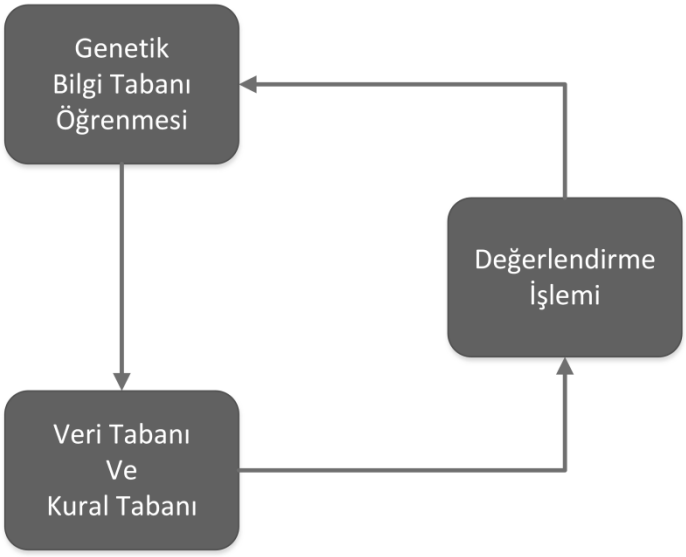


Şekil 5.6. Gömülü genetik veri tabanı öğrenmesi modelinin genel çalışma yapısı

5.2.4. Veri Tabanı ve Kural Tabanı Öğrenmesi Yaklaşımı

Bu yaklaşımda genetik algoritma eş zamanlı olarak hem kural kümesini hem de veri tabanını öğrenmektedir. Bu yaklaşımdaki en önemli nokta hem veri tabanının hem de kural tabanının aynı anda genetik algoritmaya sunulabilmesidir. Oluşturulacak bireylerin tasarımı buna imkân sağlamalıdır. Aynı zamanda genetik algoritma oluşan bu çok yönlü uzayda arama yapacak etkinlikte olmalıdır.

Fikir olarak cazip görünse de bu iki yaklaşımın beraber değerlendirilmesi çözüm uzayını oldukça büyütmektedir. Bu nedenle çözümün istenilen zamanda ve kalitede bulunmasını zorlaştırmaktadır. Bu modelin genel çalışma yapısı Şekil 5.7'de gösterilmiştir.



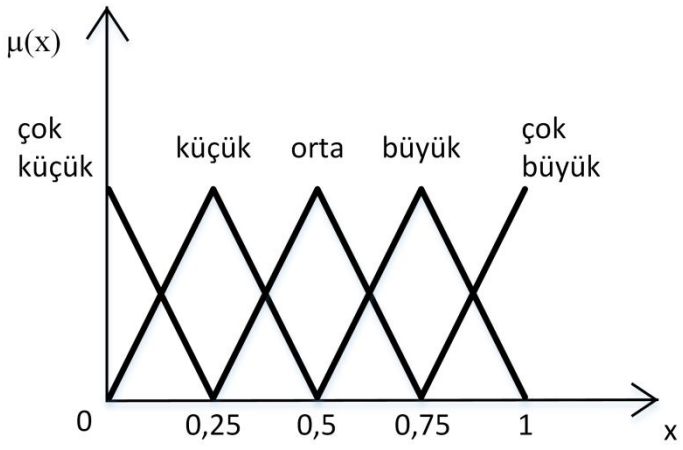
Şekil 5.7. Genetik veri tabanı ve kural tabanı öğrenmesi modelinin genel çalışma yapısı

6. İDEAL BULANIK KURAL KÜMESİNİN OLUŞTURULMASI İÇİN TASARLANAN GENETİK ALGORİTMA

BKTS sistemlerde genetik algoritmaların, bulanık kural kümesinin oluşturulmasında etkin bir şekilde kullanıldığı pek çok başarılı model ortaya konulmuştur (Cordon, 2001). Bulanık sistemin başarısını etkileyen en önemli adım bulanık kural kümesinin oluşturulmasıdır. Bulanık sınıflandırıcıda istenilen özellikler sınıflandırma başarısının yüksek olması ve kolay anlaşılabilir olmasıdır. Sınıflandırma başarısı sınıflandırıcının doğru olarak sınıflandırdığı örnek sayısı ile doğru orantılıdır. BKTS'nin kolay anlaşılabilir olması sınıflandırıcı sistemdeki bulanık kurallarının sayısı ve bu kuralların içerdiği sözsel değişken sayısı ile ters orantılıdır. Bu bilgiler ışığında ideal bulanık kural kümesi, sınıflandırma başarısını maksimize eden, en az sayıda kural sayısı ve kural uzunluğuna sahip kümedir.

Amacı ideal bulanık kural kümesini bulmak olan bir genetik algoritma tasarlanırken, genetik algoritmanın aşamalarında farklı yaklaşımlar kullanmak gerekmektedir. Örneğin çözümün bir birey olarak nasıl sunulacağı, uygunluk fonksiyonunun seçimi, çaprazlama ve mutasyon işlemlerinin nasıl yapılacağı çözüme ulaşmada belirleyici olmaktadır. Bu tez çalışmasında bulanık kural kümesinin oluşturulması için farklı yaklaşımları içeren genetik algoritma tasarlanmıştır. Tasarlanan bu genetik algoritmanın ayrıntıları alt başlıklar halinde sunulmuştur. Bireyin genetik olarak sunulması, uygunluk fonksiyonu ve mutasyon alt başlıklarında tez çalışması kapsamında ortaya konulan yeni yaklaşımlar detaylı olarak açıklanmıştır. Tasarlanan genetik algoritmanın seçim ve çaprazlama adımlarında geleneksel yöntemler kullanılmıştır. Aşağıda tez çalışmasında kullanılan, bulanık kuralların ifade edilmesi için gerekli üyelik fonksiyonları ve bu üyelik fonksiyonlarına ait üyelik değerlerinin hesaplanması hakkında detaylı bilgi verilmektedir.

Bulanık kuralların ifade edilebilmesi için sınıflandırıcı sistemde kullanılacak sözsel değişkenleri yani her bir giriş parametresi için üyelik fonksiyonlarının belirlenmiş olması gerekmektedir. Bu tez çalışmasında yapılan bütün deneysel çalışmalarda Şekil 6.1'deki 5 üyelik fonksiyonu (1 – "Çok küçük", 2 – "Küçük", 3 – "Orta", 4 – "Büyük", 5 – "Çok büyük") kullanılmıştır.



Şekil 6.1. Giriş parametreleri için kullanılacak üyelik fonksiyonları

Bütün üyelik fonksiyonları simetrik ve üçgen üyelik fonksiyonlarıdır. Bu üçgen üyelik fonksiyonlarının *a*, *b* ve *c* parametre değerleri Çizelge 6.1'de gösterilmiştir.

Çizelge 6.1. Üyelik fonksiyonlarının parametre değerleri

Fonksiyon No	Üyelik Fonksiyonu	*a*	*b*	*c*
1	Çok küçük	-0.25	0.00	0.25
2	Küçük	0.00	0.25	0.50
3	Orta	0.25	0.50	0.75
4	Büyük	0.50	0.75	1.00
5	Çok büyük	0.75	1.00	1.25

Her bir giriş parametresi farklı aralıklarda değer almaktadır. Bütün giriş parametreleri için Şekil 6.1'deki üyelik fonksiyonları kullanılacaksa giriş parametrelerinin değerleri 0 ile 1 arasında normalize edilmelidir. Bu sayede giriş parametreleri hangi aralıkta değer alırsa alsın algoritmanın programlaması yapılırken değerler 0 ile 1 arasında değerler alıyormuş gibi kodlama yapılarak farklı veri kümeleri aynı program kodları için kolaylıkla çalıştırılabilmektedir. Normalizasyon işlemi 6.1'deki formalizasyonla ifade edilmektedir.

$$Norm(a) = \frac{a - a_{min}}{a_{min} - a_{max}} \tag{6.1}$$

$Norm(a)$ normalize edilmiş değeri, a normalize edilecek değeri, a_{min} ve a_{max} normalize edilecek değerler içerisindeki minimum ve maksimum değeri ifade

etmektedir. Normalizasyon işlemi ile değerler 0 ile 1 arasında sınırlandırıldığı için Şekil 6.1'deki üyelik fonksiyonları gösterilirken bu aralıkta gösterilmiştir. "Çok küçük" ve "Çok büyük" üyelik fonksiyonlarının sadece 0 ile 1 arasında kalan kısımları kullanılmaktadır.

Bu üyelik fonksiyonları kullanılarak normalize edilmiş giriş değerlerine ait üyelik değeri hesaplanmaktadır. *i* inci üyelik fonksiyonu için *x* değerine karşılık gelen üyelik değeri $\mu_i(x)$ şeklinde gösterilmektedir ve 6.2'deki denklem ile ifade edilmektedir. *K* toplam üyelik fonksiyonu sayısıdır.

$$\mu_i(x) = \max\left\{1 - \frac{\left|x - \frac{i-1}{K-1}\right|}{\frac{1}{K-1}}, 0\right\}, \qquad i = 1,2,\dots,K \tag{6.2}$$

Örnek olarak *x* değerimiz 0.375 olsun ve "Küçük" üyelik fonksiyonu için üyelik değerini hesaplayalım. "Küçük" üyelik fonksiyonu 2. üyelik fonksiyonumuz ve *i* değişkeninin değeri 2, *K* değişkeninin değeri ise 5 dir. Buna göre 6.2'deki denklemi kullanarak 6.3'deki ifade de $\mu_2(0.375)$ değeri 0.5 olarak gerçekleşmektedir.

$$\mu_2(0.375) = \max\left\{1 - \frac{\left|0.375 - \frac{2-1}{5-1}\right|}{\frac{1}{5-1}}, 0\right\} \Rightarrow \{0.5, 0\} \Rightarrow 0.5 \tag{6.3}$$

6.1. Bulanık Kural Kümesinin Genetik Birey Olarak Sunulması

Bu adım problemin çözümünün genetik algoritma tarafından nasıl ifade edileceğinin belirlendiği adımdır. Bir bulanık kural kümesinin genetik algoritmanın bireyleri olarak nasıl ifade edileceğini göstermektedir. Genetik bulanık sistemlerin ideal bulanık kural kümesinin bulunmasını amaçlayan modellerinde bulanık kural kümesinin genetik birey olarak sunulmasında temel iki yaklaşım bulunmaktadır. Bunlar Pittsburgh (Smith, 1983) ve Michigan(Holland, 1986) yaklaşımlarıdır.

- *Pittsburgh Yaklaşımı:* Bu yaklaşımda her bir birey bir bulanık kural kümesini temsil etmektedir. Bu yaklaşımda bireyin her bir bit değeri bir kuralı ifade etmektedir. Genellikle önceden oluşturulmuş aday kurallar

kullanılarak ideal bulanık kural kümesinin oluşturulduğu modellerde ikili kodlama ile bu yaklaşım kullanılmaktadır. Bireyin uzunluğu aday kural sayısı kadardır. Bireyin her bir biti sıralanmış olan o kuralın kümede olup olmadığını göstermektedir. Örneğin 5. bit değerinin 1 olması, 5. sıradaki aday kuralın kural kümesinde olduğunu göstermektedir. 0 değerine sahip bitlere karşılık gelen aday kurallar bulanık kümeye dahil değildir. Bir bireyin temsil ettiği bulanık kural kümesindeki kural sayısı, 1 değerine sahip bitlerinin sayısıdır.



Şekil 6.2. Pittsburgh yaklaşımı ile kodlanmış bir genetik birey

Şekil 6.2'de Pittsburgh yaklaşımı ile kodlaması yapılmış örnek bir genetik birey gösterilmektedir. Örnek modelde toplam 12 aday kural bulunduğu için bireyin uzunluğu 12 bit kadardır. Bu aday kuralların yapısı, içerdiği sözsel değişkenler, sınıf değerleri ve ağırlık değerleri bir ön işleme adımı ile belirlenmektedir. Belirlenen bu aday kurallar belirli bir düzene göre sıralanmakta ve genetik birey içerisinde bu sıradaki bit ile ifade edilmektedir. Örnek genetik bireyde 6 adet değeri 1 olan bit bulunmakta ve bu bitlere karşılık gelen 6 adet kural bu bireyin temsil ettiği bulanık kural kümesini oluşturmaktadır.

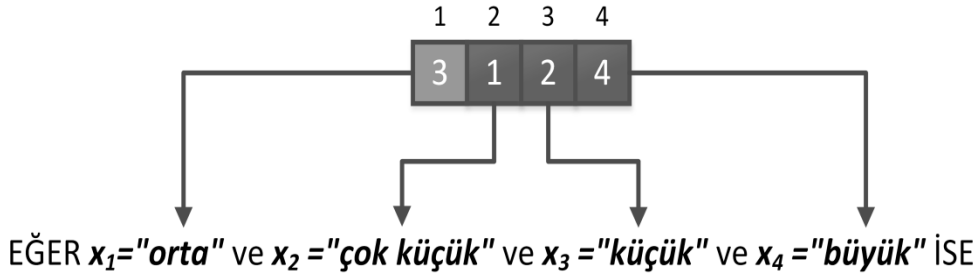
$$(K_1) \times (K_2) \times \ldots \times (K_n), \quad K_i = K \Rightarrow K^n \tag{6.4}$$

Pittsburgh yaklaşımı bazı avantaj ve dezavantajlara sahiptir. Bazı dezavantajlarını sıralayacak olursak. Bu yaklaşımda birey içerisinde kurala ait herhangi bir detay bilgi mevcut değildir. Sadece belirlenen kuralların, bulanık kümeye dahil edilme veya edilmeme bilgisini içermektedir. Bu

nedenle genetik algoritmanın oluşturacağı çözümler bu aday kural kümesi ile kısıtlanmış olmaktadır. Giriş parametresinin ve giriş parametrelerine ait üyelik fonksiyonlarının sayısı doğrusal olarak arttığında oluşturulabilecek olası aday kural kümesi üssel bir artış göstermektedir. n tane giriş parametresi olan ve i inci parametresi için K_i adet sözsel değişen tanımlanmış bir veri kümesi için oluşturulabilecek olası kural sayısı 6.4'de ifade edilmiştir. Örnek olarak 4 giriş parametresi olan ve her bir giriş parametresi 5 üyelik fonksiyonu ile tanımlanan bir modelde oluşabilecek olası kural sayısı $5^4 = 625$ dir. Aynı model için eğer üyelik fonksiyonu sayısı 5 olursa olası kural sayısı $5^5 = 3125$ veya giriş parametre sayısı 6 olursa olası kural sayısı $6^4 = 1296$ şeklinde olacaktır. Örnekte görüldüğü üzere giriş parametresinin veya üyelik fonksiyonlarının sayısındaki küçük artışlar olası kural sayısında büyük artışlara neden olmaktadır. Özellik sayısı fazla olan veri kümeleri için sınıflandırma modelleri tasarlanırken bazen bu rakam milyonları hatta milyarları bulabilmektedir. Bu nedenden dolayı aday kural kümesi oluşturulamaz hale gelmektedir veya milyarları bulan aday kural kümesinin genetik bir birey olarak kodlanması sonucunda uzunluğu milyarları bulan bireyler ortaya çıkmaktadır. Birey uzunluğunun büyük olması ideal çözümün bulunmasını zorlaştırmakta ve çalışma süresini olumsuz etkilemektedir. Bütün bunların yanında Pittsburgh yaklaşımının avantajları da vardır. Bireyin genetik kodlama mantığı basit olduğundan klasik mutasyon, çaprazlama ve seçme işlemleri kolaylıkla modele uygulanabilmektedir. Ayrıca aday kural kümesi belirlenerek çözüm uzayının sınırları önceden belirlenerek belirli bir uzayda daha verimli arama yapması sağlanabilmektedir.

- *Michigan Yaklaşımı:* Bu yaklaşımda her bir birey yalnızca bir kuralı temsil etmektedir. Çözüm ise popülasyonu oluşturan bireylerin bir alt kümesi olarak düşünülmektedir. Pittsburgh yaklaşımında bir birey ideal kural kümesini temsil ederken, Michigan yaklaşımında seçilen bireyler topluluğu çözümü ifade etmektedir. Michigan yaklaşımında bir birey kural hakkında detaylı bilgiler içermektedir. Bu yaklaşımda her bir bit, kendisine karşılık gelen giriş parametresinin hangi üyelik fonksiyonu ile ifade edildiğini göstermektedir. Her bir bitin bir üyelik değerine karşılık geldiğini

düşünürsek Michigan yaklaşımında bir bireyin uzunluğu giriş parametresinin sayısı kadardır.

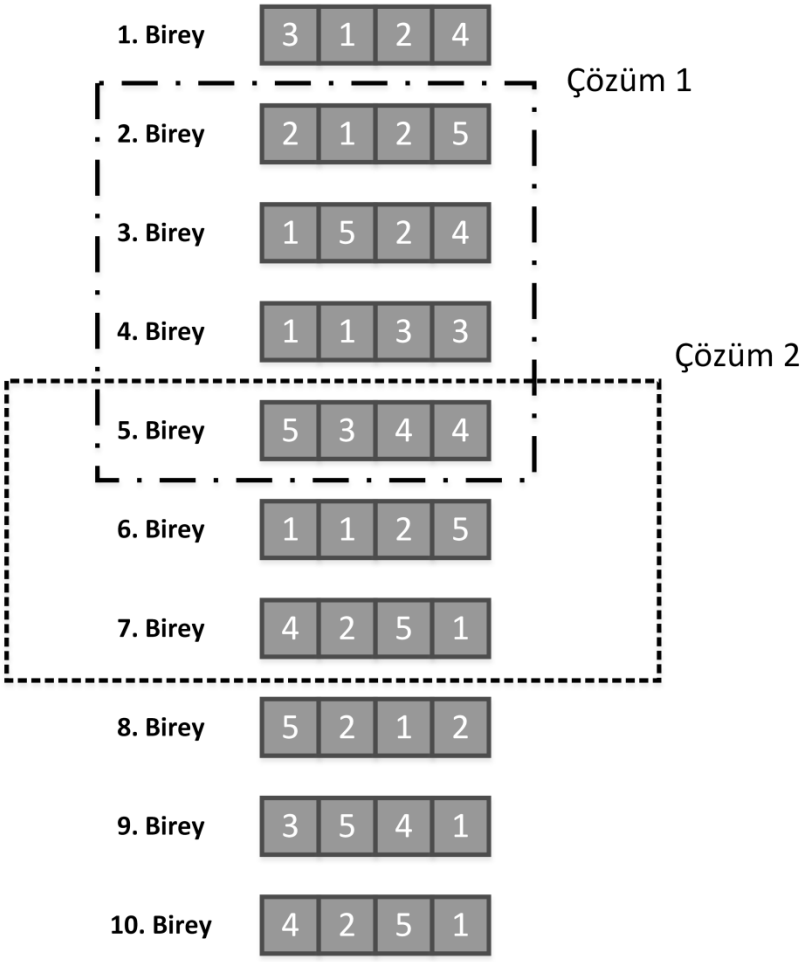


Şekil 6.3. Michigan yaklaşımı ile kodlanmış bir genetik birey

Şekil 6.3 de Michigan yaklaşım ile kodlanmış bir birey ve bu bireyin temsil ettiği bulanık kural gösterilmektedir. Örnek birey 4 giriş parametresi olan bir veri için tasarlanmıştır. Giriş parametrelerinin hepsi, Şekil 6.1'deki 5 üyelik fonksiyonu (1 – "Çok küçük", 2 – "Küçük", 3 – "Orta", 4 – "Büyük", 5 – "Çok büyük") ile ifade edilmektedir. 4 giriş parametresi olduğu için bireyin uzunluğu 4 bit ve her bit 1 ile 5 arasında üyelik fonksiyonlarını ifade eden tam sayılardan oluşmaktadır. Michigan yaklaşımında popülasyondaki birey sayısı aslında bulanık kural kümesinin maksimum kural sayısını göstermektedir.

Şekil 6.4'de toplam popülasyon sayısı 10 olan bir model üzerinde genetik algoritmanın bulduğu iki çözüm gösterilmiştir. 1. çözüm, 2, 3, 4, 5 ve 6. bireylerin temsil etiği bulanık 5 kuraldan oluşmaktadır. 2. çözüm ise 5, 6 ve 7. Bireylerin temsil ettiği 3 bulanık kuraldan oluşmaktadır.

Michigan yaklaşımında bulanık kurallar bireylerde detaylı olarak ifade edildiği için arama uzayında daha etkin bir arama yapabilmektedir. Ayrıca bireyin uzunluğu veri kümesindeki nitelik sayısı ile doğru orantılı olarak değişmektedir. Bu da Michigan yaklaşımının büyük veri kümelerinde bu yaklaşım etkin şekilde kullanılmasını sağlamaktadır. Bütün bunların yanında Michigan yaklaşımının bazı dezavantajları da mevcuttur. Örnek olarak her bir bireyi ölçeklendirmek için bireye ait bir uygunluk fonksiyonuna bunun yanında çözümlerin kalitesini belirleyecek bir uygunluk fonksiyonuna ihtiyaç duyulmaktadır. Bunun yanında çözümü oluşturmak içinde bir stratejiye gereksinim vardır.



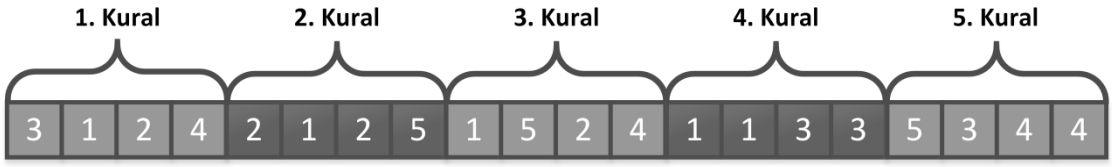
Şekil 6.4. Michigan yaklaşımında genetik çözümün gösterimi

Pittsburgh yaklaşımında genetik birey, bulanık kurala ait detaylı bilgi içermezken daha basit bir yapı sunarak uygulanması kolay olmaktadır. Buna karşın Michigan yaklaşımında bulanık kural detaylı olarak genetik bireyde temsil edilmektedir. Fakat çalışma yapısı daha karmaşık bir yapıdadır. Bu tez çalışmasında tasarlanan genetik bulanık sistemde bulanık kural kümesinin genetik olarak ifade edilmesinde hem Pittsburgh hem de Michigan yaklaşımlarının avantajlarını içeren bir yaklaşım ortaya konulmuştur.

$$Bireyin\ uzunluğu = Maksimum\ kural\ sayısı \times giriş\ parametre\ sayısı \qquad (6.5)$$

Ortaya konulan bu yeni yaklaşım temelinde Pittsburgh yaklaşımı gibi davranmaktadır. Yani bir birey bulanık kural kümesini temsil etmektedir. Fakat birey aynı zamanda bulanık kural kümesindeki kurallara ait giriş parametrelerine karşılık

gelen üyelik fonksiyonlarının hangisi olduğu bilgisini içermektedir. Bu yanı ile de Michigan yaklaşımına benzemektedir. Birey Michigan yaklaşımındaki gibi ifade edilen kuralların yan yana eklenmesi şeklinde ifade edilmektedir. Bireyin uzunluğu, maksimum kural sayısına ve giriş parametresinin sayısına bağlı olmaktadır. Bireyin uzunluğu 6.5 deki ifadede gösterilmiştir. Giriş parametre sayısı 4 olan ve maksimum kural sayısı 5 olan bir sistem düşünelim. Şekil 6.5'de örnek bir genetik birey kodlaması gösterilmiştir.



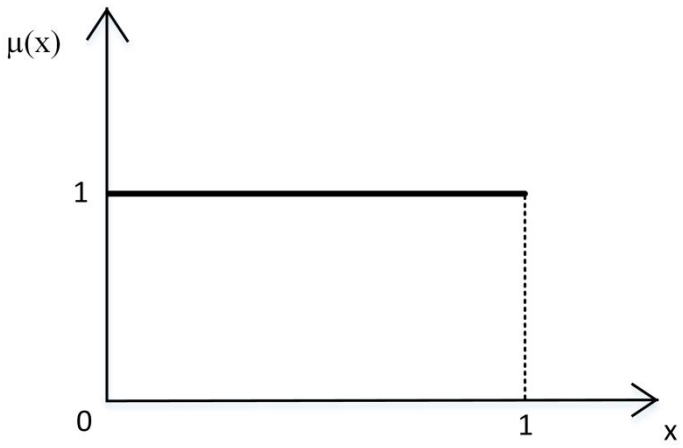
Şekil 6.5. Örnek genetik birey gösterimi

Bu yaklaşım sayesinde hem kurallar hakkında daha fazla bilgiye sahip bireyler oluşturulmakta hem de bir birey ile bulanık kural kümesi ifade edilmektedir. Bir birey ile bulanık kural kümesinin ifade edilmesi sayesinde klasik genetik algoritma operatörleri kolaylıkla kullanılmakta ve algoritmanın çözümü arama işlemi daha basit bir hale gelmektedir. Bununla birlikte giriş parametre sayısının ve/veya kullanılan üyelik fonksiyonu sayısının artması, Pittsburgh yaklaşımında bireyin uzunluğunu üssel olarak artırmakta ve bu durumda algoritmanın çalışma zamanını olumsuz olarak etkilemektedir. Tez çalışması kapsamında yeni bir yapı olarak ortaya konulan genetik bireyin uzunluğu giriş parametrelerinin sayısının artmasından doğrusal olarak etkilenmektedir. Kullanılan üyelik fonksiyonun sayısının artmasından ise etkilenmemektedir.

6.2. Uygunluk Fonksiyonu

Uygunluk fonksiyonu genetik algoritmanın amacını belirleyen fonksiyondur. Bu fonksiyon aracılığı ile popülasyon da bulunan bireylerin çözüme ne kadar yaklaştıkları belirlenmektedir. Uygunluk fonksiyonu ile iyi bireyler belirlenerek yeni popülasyonun oluşturulması sağlanmaktadır. Bu nedenden dolayı uygunluk fonksiyonunun belirlenmesi çözüm stratejisini belirlemektedir.

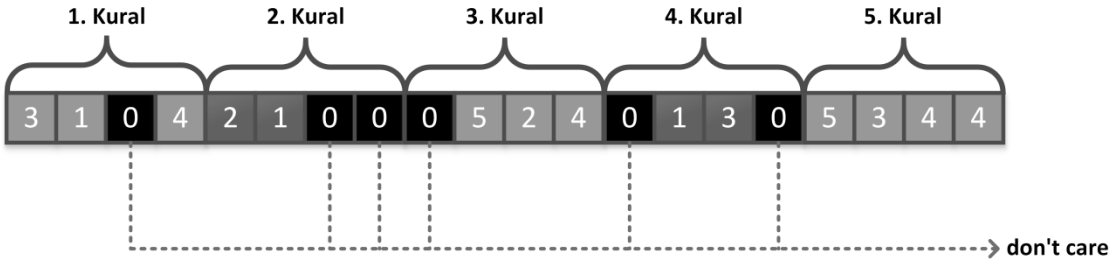
Bu tez çalışmasında amacımız BKTS'ler için ideal bulanık kural kümesinin bulunması işlemidir. BKTS sistemlerin sınıflandırma başarısının ve yorumlanabilirliğinin yüksek olması istenmektedir. Sınıflandırma başarısı, sistemin doğru olarak sınıflandırdığı örneklerin sayısının toplam örnek sayısına oranıdır. Yorumlanabilirlik ise sistemin basitliğini ve kolay anlaşılabilirliğini temsil etmektedir. BKTS sistemlerde bulanık kural kümesindeki kural sayısı ve bu kuralların uzunlukları sistemin yorumlanabilirliğini belirleyen önemli faktörlerdir. Bu bağlamda ideal bulanık kural kümesi en az sayıda ve en kısa uzunluktaki bulanık kurallar ile en yüksek sınıflandırma başarısını elde etmelidir. Genetik algoritmanın amacı sınıflandırma başarısını maksimize ederken bulanık kural sayısını ve kural uzunluklarını minimize etmelidir.



Şekil 6.6. "*don't care*" üyelik fonksiyonunu

Bölüm 6.1'de bulanık kuralların genetik birey olarak nasıl sunulacağı açıklanmıştır. Bu yaklaşımda her bir kuralın uzunluğu giriş parametresinin sayısı kadardır. Bu durumda bütün kuralların uzunluğu giriş parametre sayısı kadar olacaktır. Bu noktada daha kısa kuralların tanımlanmasına imkân veren "*don't care*" üyelik fonksiyonu kullanılmaktadır (Ishibuchi, 1999b). "*don't care*" üyelik fonksiyonu giriş parametresinin bütün değerleri için 1 ($\mu_{don'tcare}(x) = 1$) değerini döndüren bir üyelik fonksiyonudur. "*don't care*" üyelik fonksiyonu Şekil 6.6'da gösterilmiştir. Bu özel fonksiyon bir kural içerisinde herhangi bir parametreye karşılık gelirse, bu parametrenin değeri bu kural için herhangi bir anlam ifade etmemektedir. Bu tez çalışmasında Şekil 6.1'de gösterilen 5 adet üyelik fonksiyonu 1 ile 5 arasında kodlanırken, "*don't care*" üyelik fonksiyonu 0 ile kodlanmaktadır. Bulanık bir kuralın uzunluğu; "*don't care*"

üyelik fonksiyonu ile tanımlanmayan giriş parametre sayısıdır. Şekil 6.7'de "*don't care*" üyelik fonksiyonunu içeren örnek bir genetik birey gösterilmektedir. Genetik birey 5 adet bulanık kuraldan oluşmakta ve her giriş parametre sayısı 4 dür. Şekil 6.7'deki örnek genetik bireyin kurallarının uzunlukları sırasıyla 3, 2, 3, 2 ve 4 şeklindedir.



Şekil 6.7. "*don't care*" üyelik fonksiyonunu içeren örnek genetik birey

Bu tez çalışmasında sunulan yaklaşımda genetik bireyde sadece bulanık kuralın koşul kısmı gösterilmektedir. Kuralın sonuç sınıf değeri ve kuralın ağırlığı genetik birey içerisinde gösterilmemektedir. Bulanık kuralın sonuç değerini ve bulanık kuralın ağırlığının bulunması işlemi, Ishibuchi ve Yamamoto 2004 yılında yaptıkları çalışmadaki yöntem ile gerçekleştirilmiştir. Bu çalışmada bulanık mantık ile birliktelik kuralları birlikte kullanılmıştır (Ishibuchi ve ark., 2001b). Bulanık kuralların sınıf değerinin belirlenmesinde ve ağırlık değerinin bulunmasında birliktelik kurallarındaki güven kavramı kullanılmıştır. *A* bulanık kuralın koşul kısmını, *C* kuralın sonuç kısmını temsil ettiğinde, bulanık kural kümesinin q ıncı kuralının güven değeri $c(A_q \Rightarrow C_q)$ ifadesi ile gösterilmektedir (Hong 2001).

$$c(A_q \Rightarrow C_q) = \frac{\sum_{p \in Class\ C_q} \mu_{A_q}(x_p)}{\sum_{p=1}^{m} \mu_{A_q}(x_p)} \tag{6.6}$$

$$\mu_{A_q}(x_p) = \mu_{A_{q1}}(x_{p1}) \times \mu_{A_{q2}}(x_{p2}) \times \ \ldots \times \ \mu_{A_{q1n}}(x_{pn}) \tag{6.7}$$

n giriş parametresi ve *M* tane farklı sonuç sınıf değeri olan bir model için güven değerini 6.6 ve 6.7 deki denklemler ile hesaplanmaktadır. Koşul kısmı belirlenen bulanık kuralların, sonuç sınıf değeri bulmak için her bir farklı sınıf değeri için bulanık

kuralın güven değeri hesaplanmaktadır. En büyük güven değerine sahip sınıf değeri kuralın sonuç sınıf değerini göstermektedir. Aynı zamanda bu değer bulanık kuralın güven değeri olarak kullanılmakta ve 6.8 de ifade edilmektedir.

$$c(A_q \Rightarrow C_q) = max\{c(A_q \Rightarrow Sınıf\ 1), c(A_q \Rightarrow Sınıf\ 2), \dots, c(A_q \Rightarrow Sınıf\ M)\} \quad (6.8)$$

Bazı durumlarda en büyük güven değerine birden fazla sınıf sahip olabilmektedir. Bu durumda bulanık kuralın sınıf değerine karar verilememektedir. Bu tip bulanık kurallar fazlalık kural olarak değerlendirilmekte ve bulanık kural kümesinden çıkarılmaktadır. Bütün sınıflara ait güven değeri sıfır olursa bu kural da fazlalık kural olarak değerlendirilir ve bulanık kural kümesinden çıkarılır.

Bulanık kuralın sonuç sınıf değeri belirlendikten sonra kuralın ağırlığının belirlenmesi gerekmektedir. Cordon ve ark (1999a) tarafından 1999 yılında yaptığı çalışmada bulanık kuralların ağırlık değerinin belirlenmesinde kuralın güven değeri kullanılmıştır. 6.9 da bulanık kural kümesindeki q ıncı kuralın ağırlık değeri ifade edilmiştir.

$$CF_q = c(A_q \Rightarrow C_q) \quad (6.9)$$

Buna ek olarak Ishibuchi ve ark. da 2001 yılların da yaptıkları çalışmada 2 farklı ağırlık belirleme yöntemi ortaya koymuştur. Bu yöntemlerden ilki 6.10 ve 6.11 de gösterilmiştir. Bu yöntemde kuralın ağırlığı ait olduğu sınıf değerindeki güven değerinden diğer sınıflara ait güven değerlerinin ortalamasının çıkarılması ile elde edilmektedir.

$$CF_q = c(A_q \Rightarrow C_q) - \bar{c} \quad (6.10)$$

$$\bar{c} = \frac{1}{M} \sum_{\substack{t=1 \\ t \neq C_q}}^{M} c(A_q \Rightarrow Sınıf\ t) \quad (6.11)$$

Ishibuchi ve arkadaşlarının ortaya koyduğu diğer ağırlık yaklaşımı ise 6.12 ve 6.13 de gösterilmiştir. Bu yöntemde bulanık kuralın ait olduğu sınıfın güven değerinden diğer sınıf değerlerinin en büyüğü çıkarılmaktadır.

$$CF_q = c(A_q \Rightarrow C_q) - c(A_q \Rightarrow C_{q^{**}}) \tag{6.12}$$

$$c(A_q \Rightarrow C_{q^{**}}) = \max\{(A_q \Rightarrow Sınıf\ t) \mid t = 1,2,\dots,M; t \neq C_q\} \tag{6.13}$$

6.10 ve 6.12 de ifade edilen ağırlık yöntemleri sonuç sınıf sayısı iki olan veri kümeleri için aynı sonucu vermektedir. BKTS sistemlerde bulanık kural kümesindeki kuralların ağırlık değerlerinin hepsine 1 değeri atanırsa bu kuralların ağırlıksız olarak kullanılacağı anlamına gelmektedir. Eğer herhangi bir kuralın ağırlık değeri 0 ise bu kural sınıflandırma işlemine etki etmemektedir. Bu tez çalışmasında bütün bulanık kurallar için 6.9 daki ifade yani bulanık kuralın ait olduğu sınıfın güven değeri kullanılmıştır.

Genetik birey olarak sunulan bulanık kural kümesindeki bulanık kuralların sınıf değerleri ve ağırlık değerleri oluşturulmaktadır. Bu bilgiler aracılığı ile seçilen çıkarım yöntemi kullanılarak kural kümesinin sistemi sınıflandırma başarısı elde edilebilir. Genetik algoritmamızın amacı en yüksek sınıflandırma başarısını en az sayıda ve en kısa kurallar ile gerçekleştirmektir. Aslında genetik algoritmamızın 3 amacı bulunmaktadır. Tez çalışmasında bu üç amacı gerçekleştirecek ve tek bir uygunluk değerini verecek bir uygunluk fonksiyonu ortaya konulmaktadır.

Uygunluk fonksiyonunu F olarak tanımlarsak, f_1 sınıflandırma başarısı, f_2 bulanık kural kümesindeki kural sayısı ve f_3 de bulanık kural kümesindeki kuralların uzunluğunun ifade eden fonksiyonudur. f_3 değeri bulanık kural kümesindeki kuralların uzunluğunun ortalamasını ifade etmektedir. f_1 fonksiyonunu maksimize ederken f_2 ve f_3 fonksiyonlarını minimize eden uygunluk fonksiyonun sözde kodu şu şekildedir.

1. *F:Uygunluk Fonksiyonu Değeri;*
2. *f_1=Sınıflandırma Başarısı;*
3. *f_2=Maksimum Kural Sayısı – Kural Sayısı;*
4. *f_3=Maksimum Kural Uzunluğu – Ortalama Kural Uzunluğu*
5. *$F = f_1 + f_2 + f_3$;*

5. adımdaki "+" işlemi matematiksel toplama değil karakter olarak birleştirme anlamına gelmektedir. 3. ve 4. adımlarda kural sayısı ve ortalama kural uzunluğu minimize edilmek için sırasıyla maksimum kural sayısından ve maksimum kural uzunluğundan çıkarılmıştır. Bu sayede sınıflandırma başarısını maksimize ederken, kural sayısını ve

ortalama kural sayısını minimize edecek tek bir amaç fonksiyonu ortaya konulmuştur. Sonuç olarak genetik algoritma en yüksek *F* değerini aramaktadır.

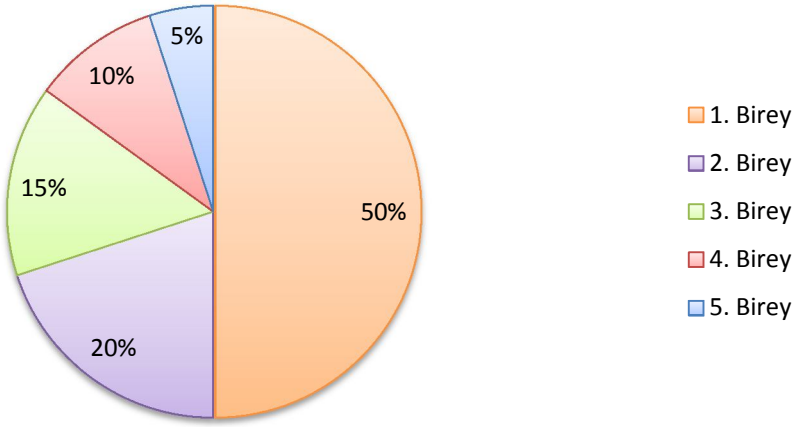
6.3. Seçim

Seçim işlemi popülasyondaki bireyler arasından güçlü bireylerin yeni bireyler oluşturmak amacıyla seçilmesidir. Seçim işlemi ile belirlenen bireyler önce çaprazlama işlemine daha sonra mutasyon işlemine tabi tutularak yeni ve güçlü bireyler oluşturulmaktadır. Seçim stratejisi oluşturulacak yeni bireylerin kalitesi açısından önemlidir. Tez çalışmasında ortaya konulan genetik algoritma modelinde seçim stratejisi olarak rulet tekerleği yöntemi kullanılmıştır.

Çizelge 6.2. Örnek rulet tekerleği seçim değerleri

	Uygunluk Değeri	**Rulet Tekerleği Seçimi**	**Rulet Kümülatif**
1.Birey	50	0.50	0.50
2. Birey	20	0.20	0.70
3. Birey	15	0.15	0.85
4. Birey	10	0.10	0.95
5. Birey	5	0.05	1.00

Rulet tekerleği yönteminde bütün bireylerin uygunluk fonksiyonu değerleri hesaplanarak büyükten küçüğe doğru sıralanmaktadır. Bu uygunluk değerleri kullanılarak her bir birey için rulet tekerleği değeri hesaplanmaktadır. Rulet tekeri değeri 4.2'deki ifade ile hesaplanmaktadır. Bu aşamadan sonra bireylerin kümülatif rulet değerleri oluşturulmaktadır. Bireyin seçiminde bu kümülatif değerler kullanılmaktadır. Rasgele 0 ile 1 arasında bir sayı üretilmekte ve sıralamanın en üstünden başlayarak bu değer kümülatif değerler ile karşılaştırılmaktadır. Hangi bireyin kümülatif değerinden düşük ise o birey seçilmiş olmaktadır. 5 bireyden oluşan bir popülasyon için uygunluk değerleri ve rulet tekerleği değerleri Çizelge 6.2'de gösterilmektedir. 0 ile 1 arasında üretilen rasgele değerin 0.65 olduğunu düşünelim. Üstten başlayarak sıra ile karşılaştırıldığında ilk olarak 0.70 den düşük olmaktadır. Bu nedenle 2. Birey seçilmektedir. Bireylerin seçilme olasılığını grafiksel olarak Şekil 6.8'de gösterilmiştir. Rulet tekerleği yönteminde uygunluk fonksiyonu yüksek olan bireyin seçilme ihtimali çok yüksek olmaktadır.



Şekil 6.8. Örnek rulet tekerleği grafiksel gösterimi

6.4. Çaprazlama

Çaprazlama işlemi seçilme işlemi ile oluşturulan bireyler havuzundan eşleştirilen iyi özelliklere sahip iki bireyin genlerinin yer değiştirmesi işlemidir. Amaç iyi bireylerin özelliklerinin taşıyan yeni ve daha iyi bireyler oluşturmaktır. Bireylerin özelliklerinin çaprazlanması için pek çok farklı yöntem kullanılmaktadır. Literatürde en yaygın kullanılan yöntemler tek noktalı ve iki noktalı çaprazlama yöntemleridir. Tek noktalı çaprazlama yönteminde belirlenen bir noktadan sonraki birey bilgileri karşılıklı olarak değiştirilmektedir. İki noktalı çaprazlama yönteminde ise belirlenen iki nokta arasındaki birey bilgileri yer değiştirilmektedir. Bu iki çaprazlama yöntemi bölüm 4.4 de detaylı olarak açıklanmıştır. Bu tez çalışmasında yeni bireylerin oluşturulmasında önemli bir adım olan çaprazlama için iki noktalı çaprazlama yöntemi kullanılmıştır.

6.5. Mutasyon

Mutasyon genetik algoritmanın arama yaparken yerel minimumlara takılmamasını sağlayan çok önemli bir operatördür. Mutasyon işleminde; çok küçük bir ihtimal ile bireyin herhangi bir geninde ki bilgi değişime uğramaktadır. Mutasyon operatörünün çalışma yöntemi, genetik bireyin yapısına bağlı olmaktadır. İkili kodlama şeklinde bir yapıya sahip bir genetik bireyin her hangi bir bitinin mutasyona uğraması durumunda, mutasyona uğrayan bitin değeri 1 ise 0'a veya bitin değeri 0 ise 1'e

dönüşmesi demektir. Fakat genetik bireyin bitleri her zaman ikili tabanda sayılardan oluşmayabilir. Örneğin bireyin bitleri 1 ile 5 arasında 5 adet farklı tamsayı değeri alabiliyor olsun. Mutasyon geçirecek olan bitin mevcut değerinin 3 olduğunu düşünürsek, mutasyondan sonra bu bittin muhtemel değerleri 1, 2, 4 ve 5 dir.

Bölüm 6.1'de ideal bulanık kural kümesinin bir genetik birey olarak nasıl sunulacağı ortaya konulmuştur. Ortaya konulan genetik birey bulanık kuralları giriş parametrelerine karşılık gelen uygunluk fonksiyonu bilgisini tutmaktadır. Bu tez çalışmasında Şekil 6.1'deki 5 üyelik fonksiyonu (1 – "Çok küçük", 2 – "Küçük", 3 – "Orta", 4 – "Büyük", 5 – "Çok büyük") kullanılmış ve genetik birey içerisinde de bu üyelik fonksiyonlarına karşılık gelen 1 ile 5 arasındaki tamsayı değerleri kullanılmıştır. Bunun yanında daha kısa uzunlukta kurallar oluşturmamıza imkan veren "*don't care*" üyelik fonksiyonu kullanılmakta ve 0 ile genetik birey içerisinde ifade edilmektedir. Bu noktada genetik bireyin bir biti 0 ile 5 arasında değişen 6 farklı değer alabilmektedir. Mutasyon geçiren bit bu değerlerden diğer 5 ine belirli bir ihtimal dahilinde dönüşmektedir. Bir bitin mutasyon geçirme ihtimali p_m ile ifade edilmektedir. p_m değeri 0 ile 1 arasında değişmektedir. Değer 0'a yakın olursa mutasyon ihtimali azalır, 1'e yakın bir değer ise mutasyon ihtimali yükselir. Genellikle mutasyon ihtimali 0'a yakın değerler seçilir. Mutasyon işleminin fazla olması sağlıklı bireylerin gereksiz mutasyonlar geçirerek sürekli olarak iyi bireylerin bozulmasına neden olabilmektedir.

Ishibuchi ve Yamamoto 2004 yılında yaptıkları çalışmada ikili kodlardan oluşan ve Pittsburgh yaklaşımına sahip genetik bireyi için tek bir mutasyon olasılığı yerine iki mutasyon olasılığı kullanmıştırlar. Bu olasılıklar $p_m\ (1 \rightarrow 0)$ ve $p_m\ (0 \rightarrow 1)$ şeklinde gösterilmektedir. $p_m\ (1 \rightarrow 0)$ olasılık değeri genetik bireydeki 1 değerine sahip bir bittin 0 değerine dönüşmesinde kullanılacak olan olasılık değeridir. $p_m\ (0 \rightarrow 1)$ olasılık değeri ise değeri 0 olan bir bitin 1'e dönüşmesini sağlayacak mutasyon olasılık değeridir. İki farklı olasılık değerinin kullanılmasındaki temel amaç daha az sayıda kuraldan oluşan bulanık kural kümeleri oluşturmaktır. Ishibuchi ve Yamamoto (2004) yaptığı çalışmada genetik bir bireyin bitinin 1 olması demek bu bite karşılık gelen kuralın bulanık kural kümesine dahil edilmesi demektir. Eğer bitin değeri 0 ise bu bite karşılık gelen kural bulanık kural kümesine dahil edilmemektedir. Dolaysıyla 0 değerinde bitlerin çok olduğu bireylerde kural sayısı daha az olmaktadır. Kural sayısı az olan bireyler oluşturmak için $p_m\ (1 \rightarrow 0)$ değerini büyük, $p_m\ (0 \rightarrow 1)$ değeri küçük seçilir. Böylece algoritma daha az sayıda kural içeren bulanık kural kümeleri oluşturma eğiliminde olur.

Bu tez çalışmasında, genetik birey için farklı bir mutasyon operatörü yaklaşımı ortaya konulmuştur. Bu yeni mutasyon yaklaşımında Ishibuchi ve arkadaşlarının yaklaşımından esinlenerek 3 farklı mutasyon olasılık değeri kullanılmıştır. Bu mutasyon olasılık değerleri p_m $([1-K] \rightarrow 0)$, p_m $(0 \rightarrow [1-K])$ ve p_m $([1-K] \rightarrow [1-K])$ şeklinde ifade edilmektedir. K bir giriş parametresini ifade eden üyelik fonksiyonu sayısıdır. p_m $([1-K] \rightarrow 0)$ değeri, genetik bireydeki "*don't care*" değerinde olmayan bir bitin "*don't care*" değerine dönüşme olasılığını ifade etmektedir. p_m $(0 \rightarrow [1-K])$ değeri ise *"don't care"* değerine sahip bir bitin *"don't care"* değeri dışında bir değere dönüşme olasılığını ifade etmektedir. p_m $([1-K] \rightarrow [1-K])$ değeri *"don't care"* değerinde olmayan bir bitin yine *"don't care"* olmayan ve kendinden farklı bir değere dönüşme olasılık değeridir. Olasılık değerleri belirlenirken p_m $(0 \rightarrow [1-K])$ olasılık değeri diğer iki olasılık değerine göre daha küçük belirlenmelidir. İdeal kural kümesinin en önemli özelliklerinden biride kısa uzunlukta kuralardan oluşmasıdır. Bu yaklaşım sayesinde *"don't care"* değerine sahip bit sayısını artırarak daha kısa kural uzunlukları oluşturma eğilimi gerçekleştirilmektedir.

7. DENEYSEL ÇALIŞMALAR VE SONUÇLARI

7.1. Kullanılan Veri Kümeleri

Bölüm 6'da sunulan genetik bulanık sistem üzerinde deneysel çalışmaların yapılması için 18 farklı veri kümesi kullanılmıştır. Bu veri kümelerinden "*Ringnorm*" ve "*Twonorm*" veri kümeleri Toronto Üniversitesi tarafından yürütülen DELVE projesi kapsamında oluşturulan veri ambarından elde edilmiştir (http://www.cs.utoronto.ca/~delve/data/datasets.html, erişim tarihi: 10.03.2014). Geriye kalan 16 adet veri kümesi Irvine California Üniversitesi Veri Ambarından (UCI Machine Learning Repository) elde edilmiştir (Blake and Merz, 1998). Çizelge 7.1'de tez çalışmasına kullanılan veri kümelerinin isimleri, örnek sayıları, nitelik sayıları ve sınıf değer sayıları gösterilmiştir.

Çizelge 7.1. Tez Çalışmasında Kullanılan Veri Kümeleri ve Özellikleri

Veri Kümesi	Örnek Sayısı	Nitelik Sayısı	Sınıf Sayısı
Bupa	345	6	2
Cleveland	297	13	5
Ecoli	336	7	8
Glass	214	9	7
Iris	150	4	3
Magic	1902	10	2
Page-blocks	548	10	5
Pen-Based	10992	16	10
Pima	768	8	2
Ringnorm	7400	20	2
Satimage	6435	36	6
Segmentation	2310	19	7
Sonar	208	60	2
Spambase	4597	57	2
Twonorm	7400	20	2
Wine	178	13	3
Wisconsin (Diagnostic)	569	30	2
Yeast	1484	8	10

BKTS sistemlerde bulanık kural kümesinin bulunması işleminde veri kümesindeki örnek sayısı, nitelik sayısı ve sınıf değer sayısı önem arz etmektedir.

Özellikle veri kümesindeki nitelik sayısı oluşturulacak en uzun kuralı belirlemektedir. Ayrıca nitelik sayısının artması oluşturulabilecek olası kural sayısını artırmaktadır. Örnek sayısının artması ise eğitim ve test aşamasında hesaplama maliyeti getirmektedir. Ortaya konulan yöntemin tutarlılığı açısından tez çalışmasında kullanılan veri kümeleri farklı örnek, nitelik ve sınıf değer sayılarına sahip veri kümelerinden oluşmaktadır. Deneysel çalışmada kullanılan bütün veri kümeleri sürekli değerli niteliklere sahiptir.

7.2. Yapılan Deneysel Çalışmada Kullanılan Yardımcı Yöntemler

7.2.1. *k*-kez çapraz doğrulama yöntemi

k-kez çapraz doğrulama yöntemi sınıflandırıcı modellerin bir veri kümesi üzerinde yapılan sınıflandırma işleminin sonuçlarının tutarlı olması için kullanılmaktadır. Metodun uygulanmasından önce k parametresinin belirlenmesi gerekmektedir. k parametresi veri kümesinin kaç parçaya bölüneceğini belirtmektedir.

	1. Parça	2. Parça	3. Parça	4. Parça	5. Parça	6. Parça	7. Parça	8. Parça	9. Parça	10. Parça
1. Adım	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
2. Adım	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
3. Adım	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
4. Adım	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
5. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
6. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim
7. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim
8. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim
9. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim
10. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test

Şekil 7.1. Örnek *10*-kez çapraz doğrulama işlemi

k adet sınıflandırma işlemi yapılmaktadır ve her adımda bölünen parçalardan bir tanesi test işlemi için ayrılmakta geriye kalan k-1 tanesi sınıflandırıcının eğitimi için kullanılmaktadır. k adım sonra elde edilen sınıflandırma sonuçlarının ortalaması

alınarak genel sınıflandırma sonucu elde edilmektedir (Breiman ve ark., 1984, Kohavi, 1995). Şekil 7.1'de k parametresi 10 olarak belirlenen bir çapraz doğrulama işlemi görsel olarak ifade edilmiştir. 10 parçaya bölünen veri kümesindeki 9 parça eğitim kümesi ve geriye kalan diğer parça test kümesi olarak kullanılmakta ve bu işlem 10 adımda ve her adımda farklı bir parça test kümesi alınarak gerçekleştirilmektedir.

7.2.2. Wilcoxon işaretli sıralamalar testi

Wilcoxon işaretli sıralamalar testi birbirine bağımlı iki küme arasında anlamlı bir fark olup olmadığını test etmektedir (Wilcoxon, 1945). Wilcoxon işaretli sıralamalar testi, eşlenik t-testinin parametrik olmayan alternatifidir. Bu testin kullanılabilmesi için veri kümeleri bağımlı olmalıdır. Örneğin iki farklı tansiyon aleti ile yapılan ölçümler arasındaki farklılığa bakmak istiyorsak ölçüm yapılan kişiler aynı olmalıdır. Bu tez çalışmasında bu test aynı veri kümeleri üzerinde farklı sınıflandırma metotlarından elde edilen sonuçlarında anlamlı bir farklılığın araştırılmasında kullanılmıştır.

Wilcoxon işaretli sıralamalar testinde öncelikli olarak iki veri kümesinin karşılıklı gelen değerleri arasındaki fark hesaplanmaktadır. Bu farkların mutlak değeri alınır ve küçükten büyüğe doğru sıra değerleri verilir. Sıra değerleri verilirken 1. Veri kümesinin değerinin büyük olduğu durumlarda pozitif işaretli sıra, küçük olduğu durumlarda negatif işaretli sıra değeri verilir. Pozitif işaretli sıra değerleri ve negatif işaretli sıra değerleri ayrı ayrı toplanır. Veri küme değerleri arasındaki farkın 0 olduğu durumlar pozitif ve negatif işaretli sıralar için eşit şekilde dağıtılır. Bulunan toplam pozitif ve negatif işaret değerlerinden mutlak değer olarak küçük olan çıkış değeri olarak kullanılır. 7.1, 7.2 ve 7.3'deki ifadelerde pozitif ve negatif işaret sıra değerlerinin ve çıkış değerlerinin hesaplanması gösterilmiştir (Alcala ve ark., 2011). d_i , i'inci örnekler arasındaki farkı ifade etmektedir. R^+ ve R^- pozitif ve negatif işaretli sıra değerlerinin toplamını ifade etmektedir. T ise çıkış değerini göstermektedir.

$$R^+ = \sum_{d_i>0} sıra(d_i) + \frac{1}{2}\sum_{d_i=0} sıra(d_i) \quad (7.1)$$

$$R^- = \sum_{d_i<0} sıra(d_i) + \frac{1}{2}\sum_{d_i=0} sıra(d_i) \quad (7.2)$$

$$T = \min(R^+, R^-) \quad (7.3)$$

Çizelge 7.2. Wilcoxon İşaretli Sıralamalar Testi için Kritik Değer Tablosu

Gözlem Sayısı	α=0.10	α=0.05	α=0.01	Gözlem Sayısı	α=0.10	α=0.05	α=0.01
1				16	35	29	19
2				17	41	34	23
3				18	47	40	27
4				19	53	46	32
5	0			20	60	52	37
6	2	0		21	67	58	42
7	3	2		22	75	65	48
8	5	3	0	23	83	73	54
9	8	5	1	24	91	81	61
10	10	8	3	25	100	89	68
11	13	10	5	26	110	98	75
12	17	13	7	27	119	107	83
13	21	17	9	28	130	116	91
14	25	21	12	29	140	126	100
15	30	25	15	30	151	137	109

Bu çıkış değeri kritik değer ile karşılaştırılır. Eğer çıkış değeri kritik değere eşit veya kritik değerden küçük ise sıfır hipotezi reddedilir. Çıkış değerinin, kritik değerden büyük olduğu durumlarda sıfır hipotezi kabul edilir (Demsar, 2006). Sıfır hipotezi H_0 şeklinde gösterilir ve "İki veri kümesi arasında anlamlı bir fark yoktur" şeklindedir. Çizelge 7.2'de farklı anlamlılık derecelerine ait gözlem sayılarına göre kritik değerler verilmiştir. α anlamlılık derecesini göstermektedir. α değerinin 0.05 olarak belirlenmesi %95 anlamlılık seviyesinde bir test yapılacağını göstermektedir.

Wilcoxon işaretli sıralamalar testini bir örnek ile açıklayalım. Çizelge 7.3'de iki sınıflandırıcının 18 adet veri seti ile yapılan sınıflandırma sonuçları gösterilmektedir. Amacımız bu iki sınıflandırıcının sonuçları arasında %95 anlamlılık derecesinde bir farklılık olup olmadığını incelemektir. H_0 hipotezimiz "sınıflandırıcı 1 ile sınıflandırıcı 2'ye ait sınıflandırma sonuçları arasında anlamlı bir fark yoktur" şeklindedir. d_i kolonunda sınıflandırıcılar arasındaki fark gösterilmiştir. İşaret kolonu farklara bağlı olarak oluşturulmuştur. Bu farklar mutlak değerlerine göre sıralanır ve işaret kolonu ile işaretli sıra değer kolonu oluşturulur. R^+ ve R^- değerleri 7.1 ve 7.2'deki ifadeler kullanılarak hesaplanır. $R^+ = 9 + 8 + 11 + 17 + 4 + 13 + 1 + 15 + 16 + 2 + 18 + 12 + 10 + 3 = 139$ ve $R^- = (-14) + (-5) + (-6) + (-7) = -32$ şeklinde hesaplanır. Çıkış değerini 7.3'deki ifade ile hesaplarsak $T = \min(139, -32) = 32$ olarak bulunur. Çizelge 7.2'de görüldüğü üzere 18 veri kümesi ve %95 anlamlılık seviyesinde kritik değer 40 olmaktadır. Çıkış değerimiz kritik değerden küçük olduğu

için H_0 hipotezi reddedilir. Bunun anlamı: sınıflandırıcı 1 ile sınıflandırıcı 2 arasında %95 önem seviyesinde anlamlı bir fark vardır.

Çizelge 7.3. Wilcoxon işaretli sıralamalar testi için örnek hesaplamalar

Veri	Sınıflandırıcı 1	Sınıflandırıcı 2	d_i	İşareti	İşaretli Sıra Değerleri
1	65,39	67,64	-2,252512	-	-14
2	55,89	54,53	1,362720	+	+9
3	81,54	82,36	-0,821100	-	-5
4	60,76	59,49	1,274810	+	+8
5	94,44	95,56	-1,120000	-	-6
6	82,11	80,46	1,650000	+	+11
7	95,05	91,51	3,540000	+	+17
8	84,44	84,27	0,170000	+	+4
9	77,34	75,43	1,909390	+	+13
10	92,74	92,71	0,025380	+	+1
11	86,72	83,64	3,080000	+	+15
12	71,03	67,94	3,090000	+	+16
13	78,35	78,25	0,100000	+	+2
14	85,26	61,56	23,700000	+	+18
15	96,72	94,96	1,760000	+	+12
16	94,76	93,38	1,380000	+	+10
17	53,78	54,97	-1,190000	-	-7
18	81,48	81,36	0,120000	+	+3

Bu tez çalışmasında, Wilcoxon işaretli sıralamalar testi Matlab R2011b versiyonu kullanılarak yapılmıştır. p = signrank(x,y,as) komutu kullanılarak hesaplama yapılmıştır. x ve y değerleri veri kümleri, as ise 0 ile 1 arasında bir değer alarak anlamlılık seviyesini belirtmektedir. p çıkış değeri anlamlılık değerinden büyük olursa sıfır hipotezi kabul edilir, küçük ise hipotez reddedilir.

7.2.3. Kullanılan Programlar

Tez çalışmasında ortaya konulan model, Matlab R2011b ve Microsoft Visual Studio 2010 programları kullanılarak gerçekleştirilmiştir. Modelin genetik algoritmalar kısmı C# programlama dili kullanılarak Microsoft Visual Studio 2010 ortamında geliştirilmiştir. Bulanık sınıflandırıcı model ise Matlab R2011b programında geliştirilmiştir.

Geliştirilen modelin farklı genetik bulanık sistemler ile kıyaslanması için KEEL (http://www.keel.es/, erişim tarihi: 04.06.2013) adında ticari olmayan bir yazılım kullanılmıştır. KEEL yazılımı sınıflandırma, kümeleme ve ayrıklaştırma gibi farklı algoritmaları içeren ve farklı çalışma grupları tarafından geliştirilen açık kaynak kodlu

bir yazılımdır. KEEL yazılımında bulunan 4 farklı algoritma karşılaştırma için kullanılmıştır. Bu algoritmalar Çizelge 7.4'de gösterilmiştir.

Çizelge 7.4. KEEL yazılımda bulunan ve kıyaslama için kullanılan algoritmalar

Kısa Adı	Algoritma	Referansı
FH-GBML	Fuzzy Hybrid Genetics-Based Machine Learning	Ishibuchi ve ark., 2005b
GFS-GCCL	Fuzzy rule approach based on a genetic cooperative-competitive learning	Ishibuchi ve ark., 1999b
SGERD	Steady-State Genetic Algorithm for Extracting Fuzzy Classification Rules From Data	Mansoori ve ark., 2008
SLAVE	Structural Learning Algorithm in a Vague Environment with Feature Selection	Gonzalez ve Perez, 2001

Bu algoritmalar dışında 2001 yılında Ishibuchi ve ark. (2001a) tarafından geliştirilen MOGA (*Multi Objective Genetic Algorithm*) algoritması da karşılaştırma işlemi için kullanılmıştır.

7.3. Yapılan Deneysel Çalışma

Bölüm 7.1'deki veri kümeleri üzerinde bölüm 6'da sunulan BKTS sistem ile deneysel çalışmalar gerçekleştirilmiştir. Yapılan deneysel çalışmada belirlenmesi gereken parametreler vardır. Öncelikli olarak bulanık kural kümesindeki maksimum kural sayısının belirlenmesi gerekmektedir. Bu parametre aynı zamanda genetik bireyin uzunluğunu da belirlemektedir. Maksimum kural sayısının belirlenmesinde veri kümesinin sınıf değer sayısı dikkate alınmıştır. Çünkü BKTS'nin sahip olduğu bulanık kuralların veri kümesindeki bütün sınıf değerlerini kapsaması gerekmektedir. Bu nedenle sınıf değerleri yüksek olan veri kümelerinde maksimum kural sayısı daha fazla olacak bir politika benimsenmiştir. Veri kümesinin sınıf değer sayısı T ile ifade edildiğinde maksimum kural sayısı 7.1' deki ifade ile hesaplanmaktadır.

$$Maksimum\ Kural\ Sayısı = \begin{cases} 20, & T \leq 4 \\ T * 4, & T > 4 \end{cases} \tag{7.1}$$

Maksimum kural sayısı belirlendikten sonra her bir veri kümesi için tasarlanacak genetik algoritmadaki bireyin uzunluğu 6.5'deki ifade yardımı ile hesaplanabilmektedir.

Çizelge 7.4'de veri kümeleri için kullanılacak maksimum kural sayısı ve oluşacak genetik bireyin uzunluğu gösterilmektedir. Örnek olarak "Segmentation" veri kümesi 7 farklı sınıf değerine sahiptir. 7.1'deki ifadeye göre maksimum kural sayısı 7*4=28 olarak belirlenmektedir. Genetik bireyin uzunluğu ise maksimum kural sayısının nitelik sayısı ile çarpımı sonucunda elde edersek, bireyin uzunluğu 28*19=532 olarak belirlenir.

Çizelge 7.4. Tez Çalışmasında Kullanılan Veri Kümeleri İçin Belirlenen Maksimum Kural Sayısı ve Genetik Bireyin Uzunluğu

Veri Kümesi	Nitelik Sayısı	Sınıf Sayısı	Maksimum Kural Sayısı	Genetik Bireyin Uzunluğu
Bupa	6	2	20	120
Cleveland	13	5	20	260
Ecoli	7	8	32	224
Glass	9	7	28	252
Iris	4	3	20	80
Magic	10	2	20	200
Page-blocks	10	5	20	200
Pen-Based	16	10	40	640
Pima	8	2	20	160
Ringnorm	20	2	20	400
Satimage	36	6	24	864
Segmentation	19	7	28	532
Sonar	60	2	20	1200
Spambase	57	2	20	1140
Twonorm	20	2	20	400
Wine	13	3	20	260
Wisconsin (Diagnostic)	30	2	20	600
Yeast	8	10	40	320

Deneysel çalışmada ortaya konulan model Tek Kazanan çıkarım yöntemi ve Ağırlıklı Oylama çıkarım yöntemlerinin ikisi için de incelenmiştir. Yapılan deneysel çalışmada karşılaştırma yapılan yöntemler ve sunulan yöntemler 10-kez çapraz doğrulama işlemi 3 defa tekrarlanarak yapılmıştır. Elde edilen 30 sonucun ortalaması alınarak çıkış sonuç değerleri elde edilmiş ve sonuçların tutarlı olması sağlanmıştır. Yapılan deneysel çalışmada kullanılan yöntemler ve bu yöntemlere ait parametreler Çizelge 7.5'de verilmiştir.

Çizelge 7.5. Deneysel Çalışmada Kullanılan Algoritmaların Çalıştırılma Parametreleri

Algoritmalar	Parametreler
FH-GBML	Popülasyon sayısı : 200 Çaprazlama oranı : 0.9 İterasyon Sayısı :200
GFS-GCCL	Üyelik Fonksiyonu Sayısı : 5 Popülasyon sayısı : 200 Çaprazlama oranı : 1 Mutasyon Oranı : 0.1 İterasyon Sayısı :200
SGERD	Üyelik Fonksiyonu Sayısı : 14 Popülasyon sayısı : 200 İterasyon Sayısı :200
SLAVE	Üyelik Fonksiyonu Sayısı : 5 Popülasyon sayısı : 200 Mutasyon Oranı : 0.01 İterasyon Sayısı :200
MOGA	Üyelik Fonksiyonu Sayısı : 5 Popülasyon sayısı : 200 Mutasyon Oranı $(1 \rightarrow 0)$: 0.1 Mutasyon Oranı $(0 \rightarrow 1)$: 0.001 İterasyon Sayısı :200
SUNULAN YÖNTEM	Üyelik Fonksiyonu Sayısı : 5 Popülasyon sayısı : 200 Mutasyon Oranı $([1-K] \rightarrow 0)$: 0.1 Mutasyon Oranı $(0 \rightarrow [1-K])$: 0.01 Mutasyon Oranı $([1-K] \rightarrow [1-K])$: 0.1 İterasyon Sayısı :200

Yapılan deneysel çalışmada elde edilen sonuçlar sadece sınıflandırma başarısı açısından değil kural sayısı ve bulanık kural kümesinin ortalama kural uzunluğu açısından da dikkate alınmıştır. Bu sayede elde edilen bulanık kural kümesinin, ideal bulanık kural kümesine ne kadar yaklaştığının analizi yapıla bilmektedir. Sınıflandırma başarıları ise eğitim kümesi ve test kümesi üzerindeki sınıflandırma başarısı olarak ayrı ayrı elde edilmiştir. Eğitim ve Test kümelerine ait sınıflandırma başarı sonuçları Çizelge 7.6 ve Çizelge 7.7'de gösterilmektedir. Çizelge 7.6 ve Çizelge 7.7'de verilen değerler yüzde olarak tanımlanmakta ve yöntemin test veya eğitim kümesindeki örneklerin yüzde kaçını doğru sınıflandırdığını göstermektedir. Çizelgelerde MOGA ve Sunulan Yöntemlere ait hem tek kazanan (TKÇ) hem de ağırlıklı oylama (AOÇ) çıkarım yöntemi için sınıflandırma başarısı sonuçları verilmiştir.

Çizelge 7.6. Deneysel Çalışma Sonucu Elde Edilen Eğitim Kümesi Sınıflandırma Sonuçları

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	72,90	60,03	59,83	65,66	69,87	73,16	70,98	**75,09**
Cleveland	58,38	57,93	55,39	**86,83**	63,80	72,15	60,83	60,23
Ecoli	75,80	67,74	75,82	**89,19**	84,59	88,02	84,29	84,29
Glass	64,76	67,58	62,87	73,96	72,79	72,33	72,55	**74,07**
Iris	**98,64**	95,57	94,64	98,00	97,53	98,47	97,68	98,57
Magic	81,09	73,61	71,97	80,16	79,35	80,70	81,63	**82,13**
Page-Blocks	94,44	90,42	91,12	93,98	92,25	91,72	94,61	**95,20**
Pen-Based	40,31	78,43	67,23	**93,63**	77,63	84,55	84,52	84,56
Pima	77,57	69,83	72,42	77,87	75,84	78,04	80,56	**80,81**
Ringnorm	88,71	90,87	70,37	85,88	88,48	93,09	93,13	**93,37**
Satimage	75,02	64,89	62,99	66,95	80,89	81,92	44,92	**81,97**
Segmentation	69,41	81,51	77,74	**87,71**	83,01	84,39	87,03	87,61
Sonar	52,86	77,22	73,94	88,10	76,21	84,56	86,63	**88,71**
Spambase	78,69	69,16	71,27	82,78	70,52	78,23	**84,27**	78,32
Twonorm	88,02	**88,41**	72,96	87,25	59,96	62,40	85,00	85,72
Wine	94,61	97,77	92,74	97,42	97,75	99,41	99,46	**99,98**
Wisconsin	92,86	91,96	91,38	93,89	94,31	96,39	97,88	**98,34**
Yeast	50,96	48,98	40,03	53,90	57,03	**56,95**	55,14	56,71
Ortalama	**75,28**	**76,22**	**72,48**	**83,51**	**78,99**	**82,03**	**81,17**	**83,65**
En İyi Sayısı	**1**	**1**	**0**	**4**	**0**	**1**	**1**	**10**

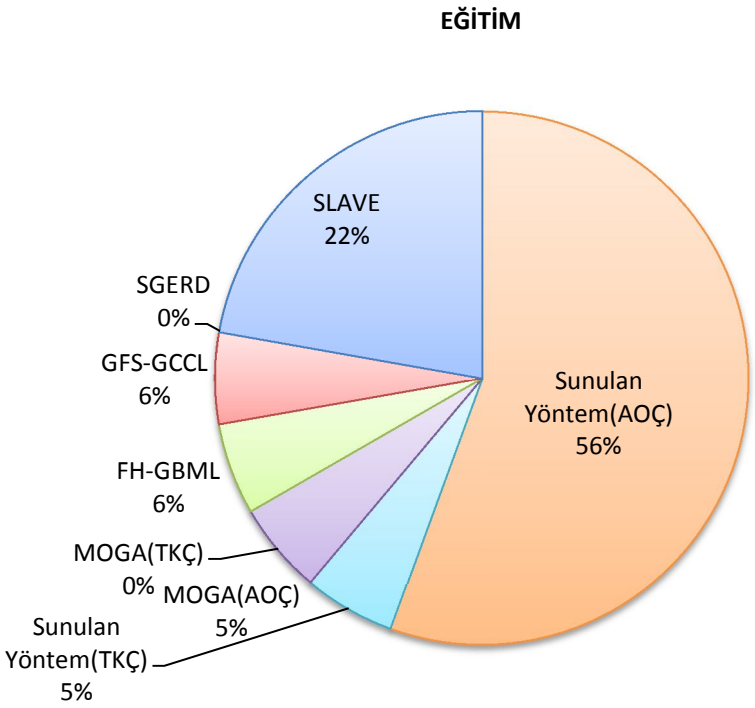
Her bir veri kümesi için eğitim ve test sonuçlarından en iyi olan sonuç kalın olarak gösterilmektedir. Çizelge 7.6 ve Çizelge 7.7'nin son iki satırında ise elde edilen eğitim ve test sonuçların ortalama değerleri ve her bir yöntemin eğitim ve test kümeleri için kaç adet en iyi sonuç elde ettikleri bilgisi bulunmaktadır. Çizelge 7.6'da elde edilen sonuçlar incelendiğinde ortalama sınıflandırma başarıları açısından hem eğitim hem de test kümeleri için en iyi sonuçlar Sunulan Yöntem (AOÇ) yaklaşımına ait olmaktadır. Eğitim kümeleri için ortalama sınıflandırma başarısı %83,65 ve test kümeleri için ortalama sınıflandırma başarısı %79,88 olarak gerçekleşmiştir.

Çizelge 7.7. Deneysel Çalışma Sonucu Elde Edilen Test Kümesi Sınıflandırma Sonuçları

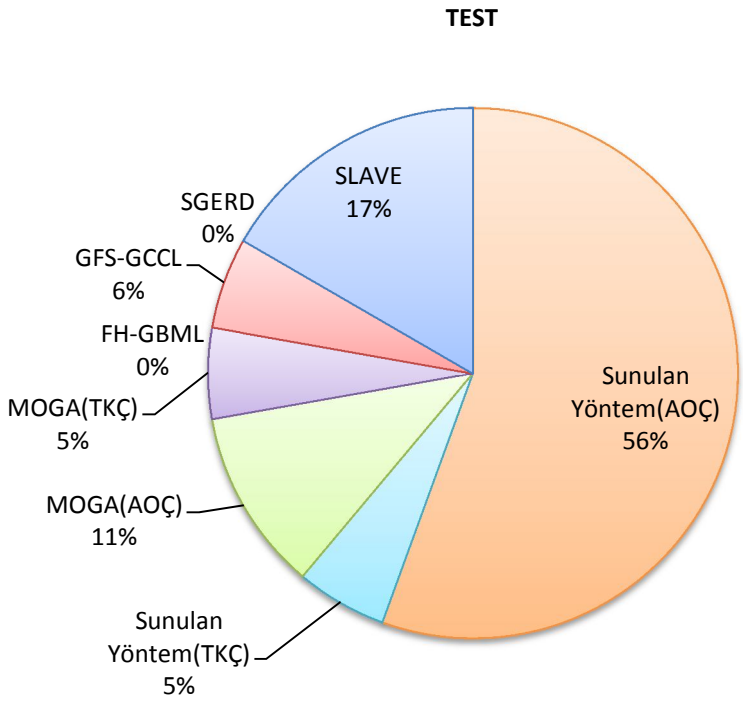
Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	64,44	58,27	58,39	59,62	61,08	**67,64**	64,17	65,39
Cleveland	53,54	52,51	47,55	54,85	46,93	54,53	54,21	**55,89**
Ecoli	72,77	64,86	74,25	**82,79**	77,88	82,36	80,34	81,54
Glass	59,73	59,83	60,98	59,96	**63,02**	59,49	60,74	60,76
Iris	93,78	91,44	95,11	**96,44**	94,22	95,56	94,44	94,44
Magic	81,45	73,57	72,06	79,91	79,28	80,46	81,36	**82,11**
Page-Blocks	94,32	90,30	91,05	93,60	92,13	91,51	94,35	**95,05**
Pen-Based	40,14	78,14	67,04	**92,03**	77,04	84,27	84,33	84,44
Pima	71,79	68,75	70,96	73,26	73,12	75,43	77,03	**77,34**
Ringnorm	88,03	90,47	72,06	85,77	88,41	92,71	92,46	**92,74**
Satimage	74,89	64,71	62,80	65,82	80,10	81,36	44,86	**81,48**
Segmentation	68,99	81,01	73,39	86,09	82,81	83,64	86,51	**86,72**
Sonar	41,15	66,02	70,28	69,71	58,66	67,94	69,40	**71,02**
Spambase	78,14	67,26	70,60	80,03	70,15	78,25	**83,68**	78,35
Twonorm	87,23	**88,34**	72,58	85,12	59,05	61,56	83,09	85,26
Wine	88,60	89,69	90,43	89,52	89,51	93,38	94,17	**94,76**
Wisconsin	90,06	90,86	91,42	91,98	92,55	94,96	94,85	**96,72**
Yeast	50,74	47,93	39,12	50,54	54,57	**54,97**	52,70	53,78
Ortalama	**72,21**	**73,55**	**71,12**	**77,61**	**74,47**	**77,78**	**77,37**	**79,88**
En İyi Sayısı	**0**	**1**	**0**	**3**	**1**	**2**	**1**	**10**

Eğitim kümesi üzerinde ortalama sınıflandırma başarısı açısından değerlendirildiğinde başarı sırası yüksekten düşüğe doğru; Sunulan Yöntem(AOÇ), SLAVE, MOGA(AOÇ), Sunulan Yöntem(TKÇ), MOGA(TKÇ), GFS-GCCL, FH-GBMLL ve SGERD şeklinde sıralanmaktadır. Şekil 7.2'de eğitim kümeleri üzerinde elde dilen başarıların grafiksel gösterimi verilmiştir.

Test kümleri ise ortalama sınıflandırma başarısı açısından yüksekten düşüğe doğru; Sunulan Yöntem(AOÇ), MOGA(AOÇ), SLAVE, MOGA(TKÇ), Sunulan Yöntem(TKÇ), GFS-GCCL, FH-GBMLL ve SGERD şeklinde sıralanmaktadır. Test kümeler üzerinde elde edilen başarıların grafiksel gösterimi Şekil 7.3'de verilmiştir. Hem eğitim hem de test kümelerinin %56'sında Sunulan Yöntem(AOÇ) yaklaşımı en iyi sonuçları elde etmiştir.



Şekil 7.2. Eğitim Kümeleri Üzerinde Elde Edilen Sınıflandırma Başarıların Grafiksel Gösterimi



Şekil 7.3. Test Kümeleri Üzerinde Elde Edilen Sınıflandırma Başarıların Grafiksel Gösterimi

Çizelge 7.6, Çizelge7.7, Şekil 7.2 ve Şekil 7.3 incelendiğinde Sunulan Yöntem (AOÇ) sınıflandırma başarısı açısından diğer yöntemlere nazaran daha başarılı sonuçlar elde etmektedir. Fakat Çizelge 7.6'da ortalama sınıflandırma başarılarına bakıldığında eğitim kümelerinde Sunulan yöntem (AOÇ) (%83,65) ve SLAVE (%83,51) arasındaki ve Çizelge 7.7'deki test kümeleri açısından Sunulan Yöntem (AOÇ) (%79,88) ve MOGA (AOÇ) (%78,99) arasındaki fark küçük gibi görünmektedir. Bu nedenle elde edilen sonuçlar arasında anlamlı bir fark olup olmadığını test etmemiz gerekmektedir. Bunun için bölüm 7.2.2'de detaylı olarak anlatılan Wilcoxon işaretli sıralamalar testi yapılmıştır. Test %95 anlamlılık seviyesinde yapılmıştır. Wilcoxon işaretli sıralamalar testi, Sunulan Yöntem (AOÇ) ile diğer yöntemleri için eğitim ve test sonuçları arasında anlamlı bir fark olup olmadığını araştıracak şekilde yapılmıştır. Eğitim ve test kümeleri için elde edilen sonuçlar kapsamında Sunulan Yöntem (AOÇ) ile diğer yöntemlerin Wilcoxon işaretli sıralamalar testi sonuçları Çizelge 7.8'de gösterilmiştir. Test %95 anlamlılık seviyesinde yapıldığı için p değeri 0,05 den küçük ise H_0 hipotezi reddedilir. Yani sonuçlar arasında anlamlı bir farklılık vardır. Eğitim kümeleri üzerinde yapılan Wilcoxon işaretli sıralamalar testinde Sunulan Yöntem (AOÇ) ile SLAVE yöntemi arasında p değeri 0,3720 olarak bulunmuş ve %95 anlamlılık seviyesinde bir fark olmadığı saptanmıştır. Diğer bütün p değerleri 0,05 den küçük olduğu için anlamlı bir fark olduğu ortaya konulmuştur.

Çizelge 7.8. Eğitim ve Test Kümelerinde Sınıflandırma Sonuçları için Wilcoxon İşaretli Sıralamalar Testi Sonuçları

	Algoritmalar	***p* değeri**
Eğitim	SUNULAN YÖNTEM(AOÇ) – SUNULAN YÖNTEM(TKÇ)	0,0148
	SUNULAN YÖNTEM(AOÇ) – MOGA(AOÇ)	0,0347
	SUNULAN YÖNTEM(AOÇ) – MOGA(TKÇ)	0,0014
	SUNULAN YÖNTEM(AOÇ) – SLAVE	**0,3720**
	SUNULAN YÖNTEM(AOÇ) – GFS-GCCL	0,00038618
	SUNULAN YÖNTEM(AOÇ) – FH-GBML	0,001
	SUNULAN YÖNTEM(AOÇ) – SGERD	0,00019644
Test	SUNULAN YÖNTEM(AOÇ) – SUNULAN YÖNTEM(TKÇ)	0,0042
	SUNULAN YÖNTEM(AOÇ) – MOGA(AOÇ)	0,0198
	SUNULAN YÖNTEM(AOÇ) – MOGA(TKÇ)	0,00053567
	SUNULAN YÖNTEM(AOÇ) – SLAVE	0,0429
	SUNULAN YÖNTEM(AOÇ) – GFS-GCCL	0,00038618
	SUNULAN YÖNTEM(AOÇ) – FH-GBML	0,00062827
	SUNULAN YÖNTEM(AOÇ) – SGERD	0,0032701

Bilindiği gibi Wilcoxon işaretli sıralamalar testi sadece sonuçlara arasında anlamlı bir fark olup olmadığını ortaya koymaktadır. Hangi yöntemin daha iyi olduğuna dair bir bilgi vermemektedir. Hangi yöntemin daha iyi olduğunu belirlemek için başarı sıra değerleri yöntemi kullanılmıştır. Bu yöntemde bütün algoritmaların her veri kümesi için elde ettiği sınıflandırma sonucu bir sıra değeri verilmektedir. Bir veri kümesi için en iyi sınıflandırma sonucunu elde eden algoritmaya 1 sıra değerini en iyi ikinci sınıflandırma sonucunu elde edene 2 değerini ve bu şekilde bütün algoritmaların her veri kümesi için sıra değerleri bulunur. Elde edilen sıra değerlerinin ortalaması alınarak ortalama sıra değeri hesaplanır. Bu ortalama başarı sıra değeri algoritmanın başarısını göstermektedir. Ortalama sıra değerinin düşük olması algoritmanın veri kümelerinde daha iyi sonuçlar elde ettiği anlamına gelmektedir.

Çizelge 7.9. Yöntemlerin Eğitim Kümeleri için Sınıflandırma Başarılarına ait Sıra Değerleri

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	4	7	8	6	5	2	3	1
Cleveland	6	7	8	1	3	2	4	5
Ecoli	7	8	6	1	3	2	4,5	4,5
Glass	7	6	8	2	3	5	4	1
Iris	1	7	8	4	6	3	5	2
Magic	3	6	8	5	7	4	2	1
Page-Blocks	3	8	7	4	5	6	2	1
Pen-Based	8	5	7	1	6	3	4	2
Pima	5	8	7	5	6	3	2	1
Ringnorm	5	4	8	7	6	3	2	1
Satimage	4	6	7	5	3	2	8	1
Segmentation	8	6	7	1	5	4	3	2
Sonar	8	5	7	2	6	4	3	1
Spambase	3	8	6	2	7	5	1	4
Twonorm	2	1	6	3	8	7	5	4
Wine	7	4	8	6	5	3	2	1
Wisconsin	6	7	8	5	4	3	2	1
Yeast	6	7	8	5	1	2	4	3
Ortalama	**5,17**	**6,11**	**7,33**	**3,61**	**4,94**	**3,50**	**3,36**	**2,03**

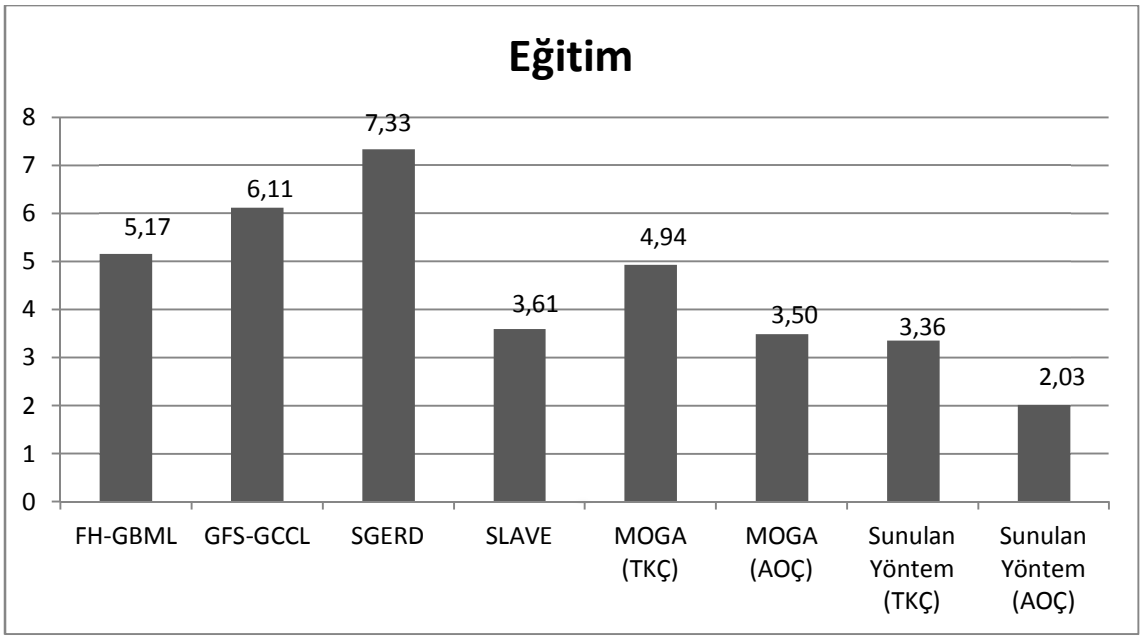
Çizelge 7.9 ve Çizelge 7.10'da kıyaslanan yöntemlerin eğitim ve test kümeleri için elde ettikleri sıra değerleri gösterilmiştir. Başarı sıra değerleri tam sayılardan oluşmaktadır. Fakat Sunulan Yöntem (AOÇ) ve Sunulan Yöntem (TKÇ)'de Ecoli veri

kümesinin eğitim kümesi ve Iris veri kümesinin test kümesi için tam sayı olmayan değerler kullanılmıştır. Bunun nedeni eğer başarı sıralaması yapılırken aynı sonuca sahip yöntemlerin sıralama değerleri sıralamada olacakları sıra değerlerinin ortalaması alınarak bulunur ve bu değer aynı sonuca sahip bütün yöntemler için kullanılır.

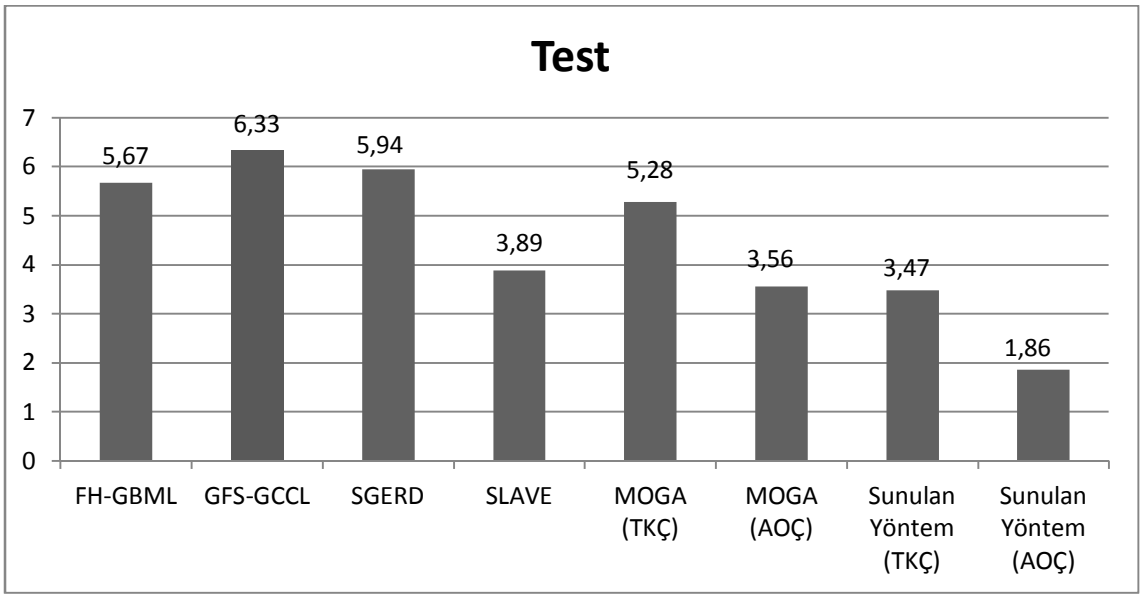
Çizelge 7.10. Yöntemlerin Test Kümeleri için Sınıflandırma Başarılarına ait Sıra Değerleri

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	3	8	7	6	5	1	4	2
Cleveland	5	6	7	2	8	3	4	1
Ecoli	7	8	6	1	5	2	4	3
Glass	7	6	2	5	1	8	4	3
Iris	7	8	2	1	6	3	4,5	4,5
Magic	2	7	8	5	6	4	3	1
Page-Blocks	3	8	7	4	5	6	2	1
Pen-Based	8	5	7	1	6	4	3	2
Pima	6	8	7	4	5	3	2	1
Ringnorm	6	4	8	7	5	2	3	1
Satimage	4	6	7	5	3	2	8	1
Segmentation	8	6	7	3	5	4	2	1
Sonar	8	6	2	3	7	5	4	1
Spambase	5	8	6	2	7	4	1	3
Twonorm	2	1	6	4	8	7	5	3
Wine	8	5	4	6	7	3	2	1
Wisconsin	8	7	6	5	4	2	3	1
Yeast	5	7	8	6	2	1	4	3
Ortalama	**5,67**	**6,33**	**5,94**	**3,89**	**5,28**	**3,56**	**3,47**	**1,86**

Örnek olarak Sunulan Yöntem (AOÇ) ve Sunulan Yöntem (TKÇ), Ecoli veri kümesinin eğitim kümesinde 84,29 sonucu ile aynı değere sahiptir. Yöntemler 4. ve 5. Sırada değerlendirilecekleri için ortalamaları alınır ve her iki yöntem içinde 4,5 sıra değeri belirlenir. Bu sıra değerlerinin ortalamaları alınarak Şekil 7.4 ve Şekil 7.5'de yöntemlerin eğitim ve test veri kümelerindeki başarılarını gösteren diyagram oluşturulmuştur. Diyagrama bakıldığında hem eğitim hem de test kümeleri için en küçük değer Sunulan Yöntem (AOÇ)'ye ait olmaktadır. Bu değerler Sunulan Yöntem (AOÇ)'nin diğer yöntemlerden sınıflandırma başarısı açısından daha başarılı olduğunu göstermektedir.



Şekil 7.4. Kıyaslanan Yöntemlerin Eğitim Kümeleri için Sınıflandırma Başarı Sıralarını Gösteren Diyagram



Şekil 7.5. Kıyaslanan Yöntemlerin Test Kümeleri için Sınıflandırma Başarı Sıralarını Gösteren Diyagram

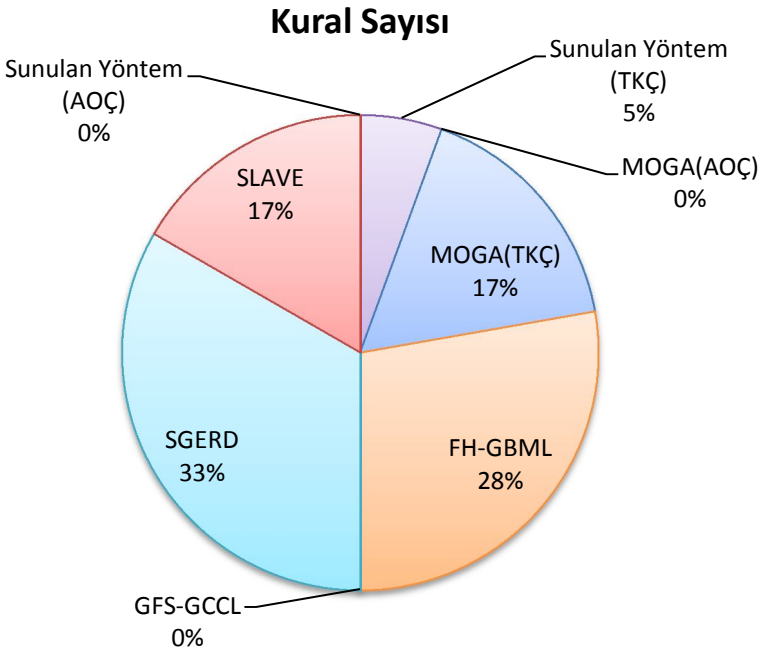
Yapılan deneysel çalışmada sadece yöntemlerin veri kümeleri üzerindeki sınıflandırma başarıları değil bu sınıflandırma başarılarını elde eden bulanık kural kümesindeki kural sayısı ve bulanık kural kümesinin ortalama kural uzunluğu da incelenmiştir. Bu tezin amacı sadece en iyi sınıflandırma başarısını elde eden bulanık

kural kümesini ortaya koyan bir model değildir. En iyi sınıflandırma başarısını en kolay yorumlanabilen bulanık kural kümesi ile elde etmektir.

Çizelge 7.11. Deneysel Çalışma Sonucu Elde Edilen Bulanık Kural Kümesine ait Kural Sayıları

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	8,47	33,90	**3,27**	5,57	6,77	9,10	10,77	12,00
Cleveland	5,33	55,67	**7,40**	36,10	16,70	25,03	12,20	9,67
Ecoli	**6,93**	59,97	8,83	12,50	11,20	24,53	10,50	12,53
Glass	**6,57**	47,27	6,67	15,63	13,00	17,47	12,97	13,80
Iris	8,73	11,77	3,27	**3,00**	4,70	6,37	4,13	5,33
Magic	8,07	81,37	**3,43**	19,97	6,93	7,63	8,00	8,00
Page-Blocks	5,50	42,57	5,63	9,73	4,90	**5,00**	12,73	13,37
Pen-Based	**9,33**	85,00	16,20	37,97	17,87	35,53	35,07	35,03
Pima	7,63	55,30	**2,63**	10,70	5,17	8,33	11,50	13,60
Ringnorm	8,37	85,40	5,97	**2,20**	11,53	11,63	14,90	15,23
Satimage	**7,33**	80,43	8,67	18,33	10,27	15,70	15,27	16,21
Segmentation	6,50	49,37	**3,30**	8,43	8,63	14,07	12,10	13,00
Sonar	5,20	53,20	2,17	30,53	**2,00**	4,00	5,20	4,00
Spambase	8,87	83,87	**3,13**	35,20	2,00	5,00	9,00	9,00
Twonorm	6,53	52,90	**3,47**	5,33	4,20	6,17	11,50	14,20
Wine	7,67	63,10	3,97	**3,60**	8,73	10,33	9,20	8,83
Wisconsin	**5,77**	74,57	11,37	23,80	16,87	20,93	16,80	21,57
Yeast	6,27	54,27	19,83	33,87	11,03	15,68	**2,00**	15,43
Ortalama	**7,17**	**59,44**	**6,62**	**17,36**	**9,03**	**13,47**	**11,88**	**13,38**
En İyi Sayısı	**5**	**0**	**7**	**3**	**1**	**1**	**1**	**0**

Çizelge 7.11'de Sunulan Yöntem ve kıyaslama için kullanılan yöntemlerle yapılan deneysel çalışma sonucunda elde edilen bulanık kural kümelerine ait kural sayısı değerleri gösterilmektedir. Çizelge 7.11'in son iki satırında yöntemlerin bulanık kural sayısı için elde ettikleri ortalama değerler ve en iyi sonuç değerlerinin sayısı gösterilmektedir. Kural sayısı açısından değerlendirildiğinde SGERD ortalama 6,62 kural sayısı ile kural sayısı açısından en efektif görünmektedir. Bulanık kural sayısı açısından değerlendirildiğinde en iyi sonuçtan en kötü sonuca doğru sıralama: SGERD, FH-GBML, MOGA(TKÇ), Sunulan Yöntem(TKÇ), Sunulan Yöntem(AOÇ), MOGA(AOÇ), SLAVE ve GFS-GCCL şeklinde olmaktadır. Şekil 7.6'de yöntemlerin kural sayılarına göre elde ettikleri en iyi sonuçların grafiksel gösterimi verilmektedir. SGERD yöntemi veri kümelerinde %33 oranında en iyi sonucu elde etmiştir.



Şekil 7.6. Yöntemlerin Kural Sayıları Açısından Elde Ettikleri En İyi Sonuçların Dağılımı

Hem ortalama kural sayısına, hem de elde edilen en iyi sonuç sayısına bakıldığında SGERD yöntemi kural sayısı açısından en başarılı yöntem olarak öne çıkmaktadır.

Çizelge 7.12. Bulanık Kural Sayıları için Wilcoxon İşaretli Sıralamalar Testi Sonuçları

Algoritmalar	***p* değeri**
SGERD – SUNULAN YÖNTEM(AOÇ)	0,00045523
SGERD – SUNULAN YÖNTEM(TKÇ)	0,0029
SGERD – MOGA(AOÇ)	0,00019644
SGERD – MOGA(TKÇ)	0,0079
SGERD – SLAVE	0,00033316
SGERD – GFS-GCCL	0,00019644
SGERD – FH-GBML	**0,372**

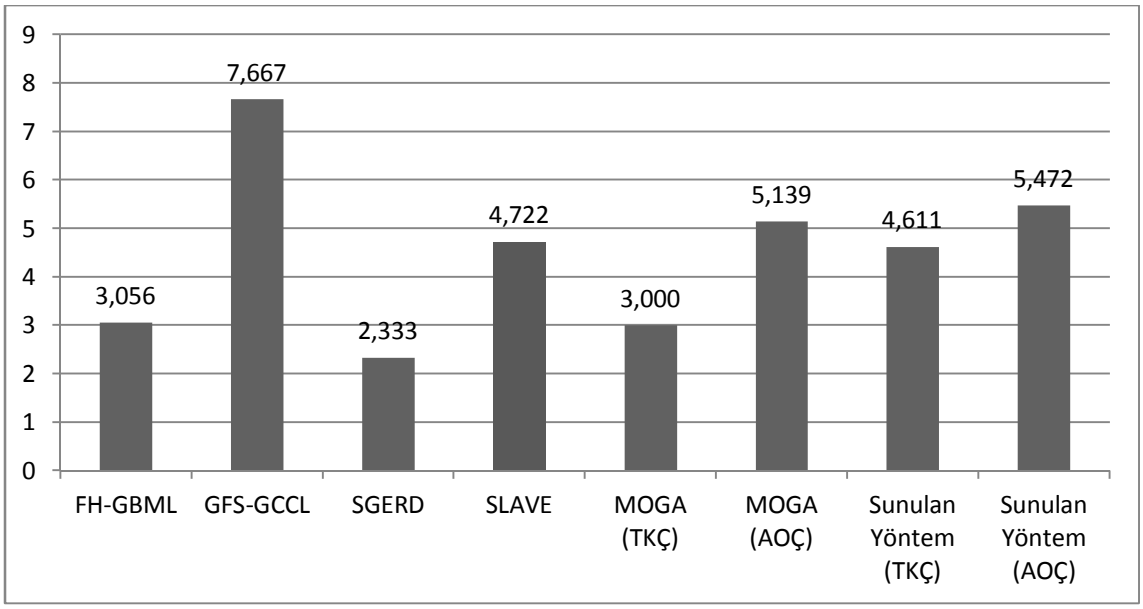
Wilcoxon işaretli sıralamalar testi kullanarak SGERD yönteminin sonuçları ile diğer yöntemlerin sonuçları arasındaki farkın anlamlı olup olmadığı ortaya konulmalıdır. Bu kapsamda yine sınıflandırma başarılarında olduğu gibi %95 anlamlılık seviyesinde Wilcoxon işaretli sıralamalar testi yapılmıştır. Çizelge 7.12'de SGERD yöntemi ile diğer yöntemler arasında %95 anlamlılık seviyesinde yapılan Wilcoxon işaretli sıralamalar testi sonuçları gösterilmektedir. *p* değerlerine bakıldığında SGERD yöntemi FH-GBML yöntemi hariç diğer bütün yöntemlerle kural sayıları açısından %95

seviyesinde bir anlamlı farka sahiptir. *p* değeri 0,372 olduğu için yani 0,05 den büyük olduğundan dolayı FH-GBML yöntemi ile anlamlı bir farka sahip değildir.

Çizelge 7.13. Kıyaslanan Yöntemlerin Kural Sayılarına Göre Başarı Sıra Değerleri

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	4	8	1	2	3	5	6	7
Cleveland	2	8	1	5	6	7	4	3
Ecoli	1	2	8	5	4	7	3	6
Glass	1	8	2	6	4	7	3	5
Iris	7	8	2	1	4	6	3	5
Magic	6	8	1	7	2	3	4,5	4,5
Page-Blocks	3	8	4	5	1	2	6	7
Pen-Based	1	8	2	7	3	6	5	4
Pima	3	8	1	4	2	5	6	7
Ringnorm	3	8	2	1	4	5	6	7
Satimage	1	8	2	7	3	5	4	6
Segmentation	2	8	1	3	4	7	5	6
Sonar	6	8	2	7	1	3,5	5	3,5
Spambase	4	8	2	7	1	3	5,5	5,5
Twonorm	5	8	1	3	2	4	6	7
Wine	3	8	2	1	4	7	6	5
Wisconsin	1	8	2	7	3	5	4	6
Yeast	2	8	6	7	3	5	1	4
Ortalama	**3,06**	**7,67**	**2,33**	**4,72**	**3,00**	**5,14**	**4,61**	**5,47**

Çizelge 7.13'de kural sayılarına göre başarı sıra değer tablosu verilmiştir. Kural sayısı açısından hangi yöntemin daha başarılı olduğunu Çizelge 7.13'deki ortalama değerler kullanılarak oluşturulan, Şekil 7.7'deki diyagram ile gösterilmiştir. Buna göre kural sayıları açısından en başarılı yöntem SGERD olarak belirlenmiştir. SGERD yöntemi ile sonuçları arasında anlamlı bir fark olmayan FH_GBML yöntemi de başarı sırası diyagramında en iyi ikinci yöntem olmaktadır. En başarısız yöntem ise GFS-GCCL olarak belirlenmektedir. SGERD yöntemi her ne kadar kural sayıları açısından başarılı bir performans gösterse de sınıflandırma başarıları açsından hem test hem de eğitim kümelerinde çok başarılı bir yöntem değildir. Yine buna benzer olarak MOGA (TKÇ) yöntemi de kural sayısında başarılı olmasına karşın sınıflandırma başarı performansı iyi değildir.



Şekil 7.7. Kıyaslanan Yöntemlerin Kural Sayılarına Göre Başarı Sıralarını Gösteren Diyagram

Buna ek olarak Sunulan yöntem ve MOGA yöntemi dikkate alındığında; Tek Kazanan Çıkarım yöntemi her Sunulan Yöntemde hem de MOGA yönteminde Ağırlıklı Oylama Çıkarım yönteminden kural sayıları açısından daha başarılı sonuçlar elde etmektedir.

Sunulan ve Kıyaslanan yöntemleri bulanık kural kümesinin ortalama kural uzunluğuna göre değerlendirelim. Bulanık kural kümesinin yorumlanabilirliğini bulanık kural kümesindeki kural sayısı ve bulanık kuralların uzunlukları belirlemektedir. Bu nedenle bulanık kural kümesinin ortalama kural uzunluğunun düşük olması bulanık kural kümesinin yorumlanabilirliğini artırmaktadır. Çizelge 7.14'de Sunulan Yöntem ve kıyaslanan yöntemlere ait bulanık kural kümesinin ortalama kural uzunlukları bilgisi verilmiştir. Çizelgenin son iki satırında yine yöntemlere ait ortalama kural uzunluklarının ortalama değerleri ve elde edilen en iyi sonuç sayısı gösterilmiştir.

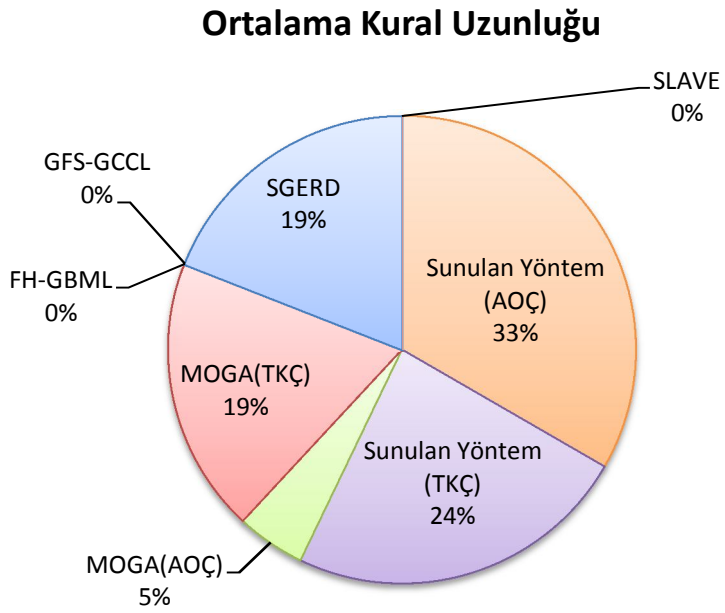
Ortalama kural uzunluklarının ortalaması dikkate alındığında Sunulan Yöntem (AOÇ) en kısa uzunluk ortalamasına sahip olmaktadır. Ortalama kural uzunluklarının ortalaması için sıralama yapacak olursak en iyi sonuçtan en kütü sonuca doğru; Sunulan Yöntem (AOÇ), Sunulan Yöntem (TKÇ), MOGA (TKÇ), SGERD, MOGA (AOÇ), GFS-GCCL, FH-GBML ve SLAVE şeklinde sıralanmaktadır. SGERD yöntemi ve MOGA (AOÇ) yönteminin her ikisinde 1,93 ortalama değerine sahip olmalarına rağmen SGERD yönteminin daha fazla en iyi sonuca sahip olması nedeniyle 4. sıraya yerleştirilmiştir.

Çizelge 7.14. Deneysel Çalışma Sonucu Elde Edilen Bulanık Kural Kümesine ait Ortalama Kural Uzunlukları

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	4,65	1,56	1,95	10,03	1,78	1,88	**1,40**	1,52
Cleveland	6,16	3,23	2,00	12,73	2,74	2,75	1,72	**1,68**
Ecoli	4,82	2,57	1,57	9,61	2,53	2,55	1,55	**1,34**
Glass	5,74	2,36	2,00	9,58	2,58	2,58	**1,59**	1,66
Iris	3,03	1,11	2,00	3,43	1,32	1,52	**1,07**	1,23
Magic	6,84	1,99	1,88	9,93	**1,52**	1,63	2,04	2,04
Page-Blocks	6,55	1,91	2,00	9,94	2,43	2,45	**1,67**	1,73
Pen-Based	8,20	2,86	**2,00**	18,99	2,66	2,66	2,67	2,67
Pima	4,80	1,88	2,00	9,72	1,61	1,64	**1,34**	1,35
Ringnorm	12,40	3,14	2,00	23,64	1,08	1,06	1,06	**1,05**
Satimage	10,61	2,25	**1,96**	11,81	2,51	2,60	2,00	2,08
Segmentation	31,31	3,74	2,00	16,00	**1,01**	1,99	1,62	1,63
Sonar	29,69	5,52	**2,00**	12,59	**2,00**	**2,00**	4,33	**2,00**
Spambase	10,98	1,69	2,00	16,19	**1,12**	1,14	2,40	2,00
Twonorm	14,79	1,76	2,00	8,77	1,26	1,32	1,38	**1,25**
Wine	7,65	7,65	2,00	10,46	1,54	1,48	1,21	**1,12**
Wisconsin	5,61	2,97	**1,45**	9,61	2,71	2,72	1,78	1,75
Yeast	20,17	2,82	2,00	17,81	1,73	1,69	3,10	**1,68**
Ortalama	**10,78**	**2,83**	**1,93**	**12,27**	**1,90**	**1,93**	**1,89**	**1,65**
En İyi Sayısı	**0**	**0**	**4**	**0**	**4**	**1**	**5**	**7**

Ortalama kural uzunluğu için elde edilen en iyi sonuçlar değerlendirildiğinde, Sunulan Yöntem (AOÇ) 7 veri kümesinde en iyi sonucu elde ederek veri kümelerinin %33'ün de en iyi sonuca ulaşmıştır. Sunulan Yöntemin (TKÇ) ise 5 veri kümesinde en iyi sonucu elde etmiş ve veri kümelerinin %24'ün de en iyi sonuca ulaşmıştır. Şekil 7.8'de bütün yöntemler için elde edilen ortalama kural uzunlukları için en iyi sonuçların dağılımı grafiksel olarak gösterilmiştir.

Ortalama kural uzunlukları içinde elde edilen sonuçlar arasında anlamlı bir fark olup olmadığını araştırmak için Wilcoxon işaretli sıralamalar testi yapılmıştır. Test hem ortalama değerler hem de en iyi sonuçlar açısından başarılı görünen Sunulan yöntem (AOÇ) ile diğer yöntemler arasında yapılmıştır. Test işlemi yine diğer testlerde olduğu gibi %95 anlamlılık seviyesinde yapılmıştır.



Şekil 7.8. Yöntemlerin Ortalama Kural Uzunluğu Açısından Elde Ettikleri En İyi Sonuçların Dağılımı

Çizelge 7.15'de Sunulan Yöntem (AOÇ) ve diğer yöntemler arasında ortalama kural uzunlukları sonuçları için yapılan Wilcoxon işaretli sıralama testi sonuçları gösterilmektedir. Sunulan Yöntem (AOÇ) ile Sunulan Yöntem (TKÇ) ve MOGA (TKÇ) yöntemleri arasındaki sonuçlara arasında %95 anlamlılık seviyesinde bir fark olmadığı gözlenmiştir. Diğer yöntemler ile Sunulan Yöntem (AOÇ) sonuçları arasında anlamlı bir fark bulunmaktadır.

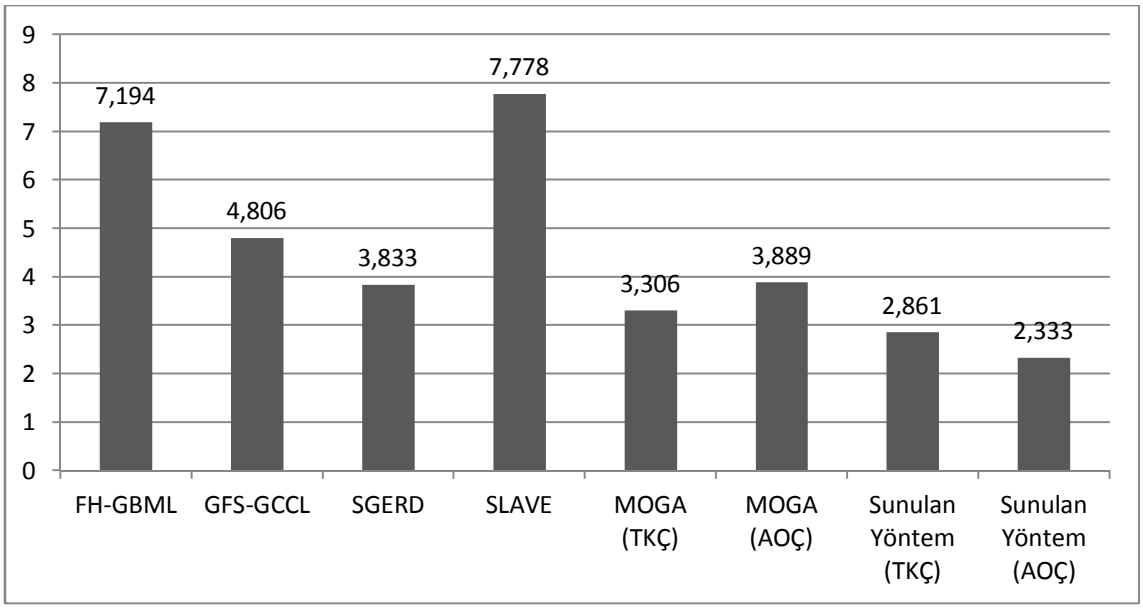
Çizelge 7.15. Ortalama Kural uzunluğu için Wilcoxon İşaretli Sıralamalar Testi Sonuçları

Algoritmalar	***p* değeri**
SUNULAN YÖNTEM(AOÇ) – SUNULAN YÖNTEM(TKÇ)	**0,2772**
SUNULAN YÖNTEM(AOÇ) – MOGA(AOÇ)	0,0493
SUNULAN YÖNTEM(AOÇ) – MOGA(TKÇ)	**0,0522**
SUNULAN YÖNTEM(AOÇ) – SLAVE	0,00019644
SUNULAN YÖNTEM(AOÇ) – GFS-GCCL	0,0130
SUNULAN YÖNTEM(AOÇ) – FH-GBML	0,0014
SUNULAN YÖNTEM(AOÇ) – SGERD	0,00019644

Çizelge 7.16. Kıyaslanan Yöntemlerin Ortalama Kural Uzunluklarına Göre Başarı Sıra Değerleri

Veri Kümeleri	FH-GBML	GFS-GCCL	SGERD	SLAVE	MOGA (TKÇ)	MOGA (AOÇ)	Sunulan Yöntem (TKÇ)	Sunulan Yöntem (AOÇ)
Bupa	7	3	6	8	4	5	1	2
Cleveland	7	6	3	8	4	5	2	1
Ecoli	7	6	3	8	4	5	2	1
Glass	7	4	3	8	5,5	5,5	1	2
Iris	7	2	6	8	4	5	1	3
Magic	7	4	3	8	1	2	5,5	5,5
Page-Blocks	7	3	4	8	5	6	1	2
Pen-Based	7	6	1	8	2,5	2,5	4,5	4,5
Pima	7	5	6	8	3	4	1	2
Ringnorm	7	6	5	8	4	2,5	2,5	1
Satimage	7	4	1	8	5	6	2	3
Segmentation	8	6	5	7	1	4	2	3
Sonar	8	6	2,5	7	2,5	2,5	5	2,5
Spambase	7	3	4,5	8	1	2	6	4,5
Twonorm	8	5	6	7	2	3	4	1
Wine	6,5	6,5	5	8	4	3	2	1
Wisconsin	7	6	1	8	4	5	3	2
Yeast	8	5	4	7	3	2	6	1
Ortalama	**7,19**	**4,81**	**3,83**	**7,78**	**3,31**	**3,89**	**2,86**	**2,33**

Kıyaslanan yöntemlerin ortalama kural uzunluğu açsından başarılarını sıralayabilmek için Çizelge 7.16'da yöntemlerin her bir veri kümesi için başarı sıra değerleri verilmiştir. Başarı sıra değerlerinin ortalamaları kullanılarak oluşturulan diyagram, yöntemlerin ortalama kural uzunluğu açısından başarısı hakkında bilgi vermektedir. Şekil 7.9'da ortalama kural uzunlukları açısından başarı sıralarına göre oluşturulmuş diyagram sunulmaktadır. Diyagrama göre Sunulan Yöntem (AOÇ) en başarılı yöntem olmaktadır. Sunulan Yöntem (AOÇ) ile Sunulan Yöntem (TKÇ) ve MOGA (TKÇ) başarı sıra değerleri oldukça yakın görünmektedir. Zaten bu yöntemler arasında Wilcoxon işaretli sıralamalar testinde anlamlı bir fark çıkmadığı hatırlanmalıdır. Şekil 7.9'deki diyagram incelendiğinde Ortalama kural sayısı açısından en başarılı yöntemden en başarısıza doğru; Sunulan Yöntem (AOÇ), Sunulan Yöntem (TKÇ), MOGA (TKÇ), SGERD, MOGA (AOÇ), GFS-GCCL, FH-GBML ve SLAVE şeklinde sıralanmaktadır.



Şekil 7.9. Kıyaslanan Yöntemlerin Ortalama Kural Uzunluklarına Göre Başarı Sıralarını Gösteren Diyagram

Yapılan deneysel çalışmada sınıf değer sayıları, nitelik sayıları ve örnek sayıları birbirinden farklı 18 veri kümesi, tez çalışması kapsamında geliştirilen yöntem ve yaygın olarak kullanılan 5 farklı BKTS yöntem ile sınıflandırılmıştır. Veri kümelerindeki değerler, Şekil 6.1'deki 5 adet simetrik üçgen üyelik fonksiyonu kullanılarak bulanık değerlere dönüştürülmektedir. Ayrıca Kısa uzunlukta kurallar tanımlayabilmek için bütün *x* değerleri için 1 üyelik değerini veren *don't care* üyelik fonksiyonu kullanılmıştır. Üyelik fonksiyonlarının değer aralığı 0 ile 1 arasında olduğu için veri kümelerindeki değerler 0 ile 1 arasında normalize edilmiştir. Deneysel çalışmada sunulan yöntem ve MOGA yöntemi, yaygın olarak kullanılan Tek Kazanan ve Ağırlıklı Oylama çıkarım yöntemlerinin ikisi ile de denenmiştir. Yapılan deneysel çalışma sonuçları 10 kez çapraz doğrulama yönteminin 3 defa tekrarlanması ile elde dilmiştir. Yani elde edilen sonuçlar bu 30 denemenin ortalaması alınarak elde edilmiştir. Deneysel çalışma sonuçları eğitim, test kümeleri üzerindeki sınıflandırma başarısı ve bu sınıflandırma başarılarının elde edildiği bulanık kural kümesindeki kural sayısı ve ortalama kural uzunluğu açısından incelenmiş ve sonuçlar detaylı grafikler ile sunulmuştur. Elde dilen sonuçlar arasında anlamlı bir farkın olup olmadığı Wilcoxon İşaretli Sıralamalar testi ile araştırılmış ve yöntemleri başarılarını belirlemek için ortalama başarı sıra değerleri diyagramından yararlanılmıştır.

8. SONUÇLAR VE ÖNERİLER

8.1. Sonuçlar

Bilgisayar sistemlerin çoğu sayısal verilerle çalışma yeteneğine sahiptir. Günlük hayatımızda ise çoğu zaman kesin olmayan veriler kullanılmaktadır. Gerçek dünyada karşılaştığımız problemler kesin olmayan veriler içermektedir. Bulanık mantık günlük hayatta kullanılan kesin olmayan verilerin bilgisayar sistemlerinde kullanılacak şekilde tanımlanmasına imkân vermektedir. Bulanık mantık ve buna bağlı olarak geliştirilen yaklaşımlar bilgisayar sistemlerinin kesin olmayan verilerle çalışmasına olanak sağlamıştır.

BKTS'ler, bulanık mantık çerçevesinde sınıflandırma yapmaktadır. BKTS'lerin tasarımındaki en önemli aşama bulanık kural kümesinin oluşturulmasıdır. Bulanık kural kümesi, problem konusunda uzman kişiler tarafından oluşturulmaktadır. Bulanık kural kümesinin uzman tarafından oluşturulması beraberinde konusunda yeterli birikime sahip uzman bulmak gibi bir dezavantajları da getirmektedir. Bu noktada bulanık kural kümesinin oluşturulmasında akıllı sistemler kullanılmaktadır. Özellikle efektif bir optimizasyon algoritması olan genetik algoritmalar sıklıkla bulanık kural kümesinin oluşturulmasında kullanılmaktadır. Genetik algoritmaların işleme sürecindeki problemin kodlanması, seçim, çaprazlama ve mutasyon adımları farklı yaklaşımlarla geliştirilerek daha verimli hale getirilmektedir.

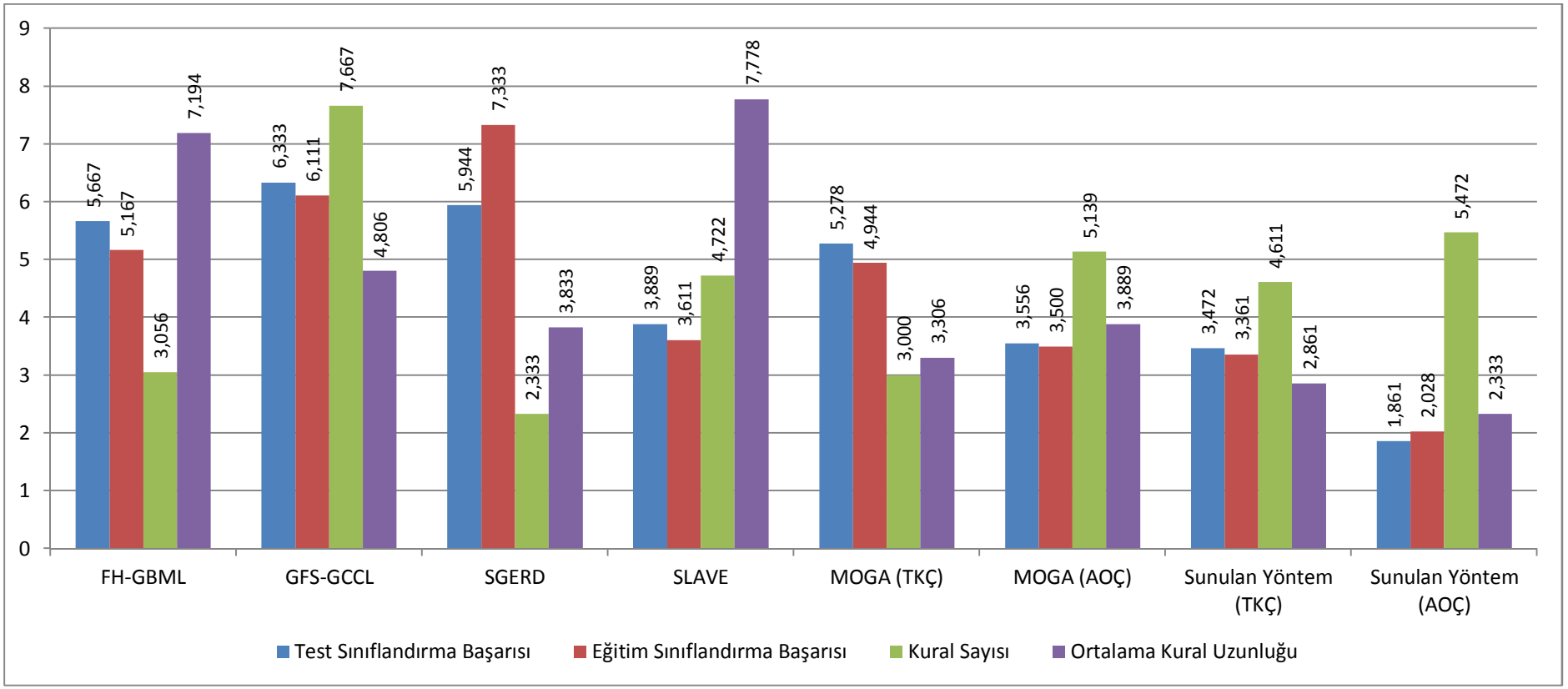
Bu tez çalışmasında farklı yaklaşımlara sahip bir genetik algoritma ortaya konulmuş ve bu model kullanılarak BKTS'nin bulanık kural kümesini oluşturan bir yaklaşım sunulmuştur. Yeni yaklaşımlara sahip genetik algoritmanın temel amacı ideal bulanık kural kümesinin oluşturulmasıdır. BKTS'lerde sınıflandırma başarısının yüksek olmasının yanında bulanık kural kümesinin yorumlanabilir olması da istenmektedir. Bulanık kural kümesinin yorumlanabilirliği bulanık kural kümesindeki kural sayısına ve bu kuralların kural uzunluğuna bağlı olmaktadır. Bu nedenle ideal bulanık kural kümesini; sınıflandırma başarısını en yüksek tutan en az sayıdaki kural ve en kısa kurallara sahip bulanık kural kümesi şeklinde tanımlayabiliriz. Bu tanım bağlamında genetik algoritmanın 3 amacı aynı anda gerçekleştirecek yapıda olması gerekmektedir. Bunlar sınıflandırma başarısını maksimize ederken bulanık kural sayısını ve bulanık kural uzunluğunu minimize etmektir.

Tez çalışmasında ortaya konulan genetik model 3 farklı yaklaşıma sahiptir. Bu yaklaşımlar problemin genetik birey olarak kodlanmasında, uygunluk fonksiyonunun tasarımında ve mutasyon operatöründe gerçekleştirilmiştir. Bulanık kural kümesinin genetik birey olarak sunulmasında genel iki yaklaşım bulunmaktadır. Bunlar Pittsburgh ve Michigan yaklaşımlarıdır. Pittsburgh yaklaşımında her bir birey bir bulanık kural kümesini ifade ederken, Michigan yaklaşımında her bir birey bir kuralı temsil etmekte ve bulanık kural kümesi bireylerden oluşan bir küme olarak ifade edilmektedir. Pittsburgh yaklaşımı ön işlem oluşturulan aday kural kümeleri arasında seçim yaparak bulanık kural kümesini oluşturmaya çalışmaktadır. Nitelik sayısı büyük veri kümelerinde olası aday kural kümesi çok büyük olduğundan ön işlem maliyeti artmaktadır. Michigan yaklaşımında ise birey bulanık kural hakkında daha fazla bilgi içermektedir. Fakat bu yaklaşımda hem bireyler için hem de çözüm için iki farklı uygunluk hesaplaması yapılmaktadır. Sunulan yaklaşımda ise her iki yaklaşımında avantajlarını kullanan temelde Pittsburgh yaklaşımını taşıyan fakat bulanık kuralların daha detaylı olarak sunulduğu bir yaklaşım benimsenmiştir. Tasarlanan uygunluk fonksiyonu ile ideal bulanık kural kümesinin sahip olması gereken özeliklere ulaşılmıştır. Uygunluk fonksiyonu sınıflandırma başarısının maksimizasyonunu, kural sayısının ve kural uzunluğunun minimizasyonunu gerçekleştiren bir yapıya sahiptir. Uygunluk fonksiyonu üç farklı amacı tek bir amaca dönüştürmektedir. Aynı zamanda kullanılan mutasyon operatörü ile genetik algoritmanın daha kısa uzunlukta kurallar arasında arama yapması sağlanmıştır. Bunun için 3 farklı mutasyon parametresi belirlenmiştir. Bunlardan birincisi p_m $([1-K] \rightarrow 0)$ değeri, genetik bireydeki "*don't care*" değerinde olmayan bir bitin "*don't care*" değerine dönüşme olasılığını ifade etmektedir. İkincisi p_m $(0 \rightarrow [1-K])$ değeri ise *"don't care"* değerine sahip bir bitin *"don't care"* değeri dışında bir değere dönüşme olasılığını ifade etmektedir. Üçüncüsü ise p_m $([1-K] \rightarrow [1-K])$ değeri *"don't care"* değerinde olmayan bir bitin yine *"don't care"* olmayan ve kendinde farklı bir değere dönüşme olasılık değeridir. Bu üç mutasyon parametresinin değerleri aynı zamanda algoritmanın hangi karakterde bireyler oluşturulacağını da belirlemektedir. Bu yaklaşımlara sahip tasarlanan genetik algoritma ile BKTS için bulanık kural kümesi oluşturulmuş ve tasarlanan model deneysel çalışmaya tabi tutulmuştur.

Deneysel çalışma farklı nitelik, örnek ve sınıf değer sayılarına sahip 18 farklı veri kümesi üzerinde yapılmıştır. Veri kümelerindeki değerler 0 ile 1 arasında normalize edilerek kullanılmıştır. Deneysel çalışmada sunulan yöntem hem Tek Kazanan hem de

Ağırlıklı Oylama çıkarım mekanizmaları ile yapılmıştır. Sunulan yöntem, literatürde başarı göstermiş 5 adet genetik BKTS ile kıyaslanmıştır. Deneysel çalışmanın tutarlılığı için karşılaştırılan yöntemlerin sonuçları, 10 kez çapraz doğrulama yönteminin 3 defa tekrarlanması sonucunda elde edilmiştir. Bu 30 sonucun ortalaması alınarak yöntemin başarısı bulunmaktadır. Deneysel çalışmada yöntemlerin başarısı yalnızca sınıflandırma başarısı açısından değil bulanık kural kümesindeki kural sayısı ve kural uzunlukları açısından da değerlendirilmiştir. Bulunan sonuçlar Wilcoxon İşaretli sıralamalar testi ile test edilerek sonuçlar arasında anlamlı bir fark olup olmadığı incelenmiştir.

Sınıflandırma başarıları, kural sayıları ve ortalama kural uzunlukları açısından her bir yöntemin başarı sıra değerleri hesaplanarak yöntemlerin başarı sıraları ortaya konulmuştur. Şekil 8.1'de deneysel çalışma sonucunda elde edilen sonuçların sınıflandırma başarıları, kural sayıları ve ortalama kural uzunlukları açısından başarı değerleri gösterilmiştir. Sunulan Yöntem (AOÇ) sınıflandırma başarısı açısından hem eğitim hem de test veri kümelerinden 18 tanesinden 10 tanesinde en iyi sonucu elde ederek veri kümelerinin %56'sında en iyi sonucu elde etmiştir. Başarı sıraları hesaplandığında ise Sunulan Yöntem (AOÇ), sınıflandırma başarıları açısından en iyi yöntem olarak belirlenmiştir. Kural sayısı açısından sunulan yöntem diğer yöntemlerden daha başarılı değildir. Bunun nedeni yöntem sınıflandırma başarısını en yüksek olacak minimum kural kümesini elde etmeye çalışmaktadır. Sonuçlar incelendiğinde kural sayısı açısından başarılı olan SGERD, MOGA(TÇK) ve FH-GBML yöntemlerinin sınıflandırma başarısındaki sonuçlar iyi değildir. Ayrıca MOGA ve Sunulan Yöntemlerin kural sayısı sonuçları incelendiğinde Tek Kazanan çıkarım yöntemi, Ağırlıklı Oylama çıkarım yönteminden kural sayısı açısından daha başarılı gözükmektedir. Ortalama kural uzunluğu açısından veri kümelerinin 7 tanesinde en iyi sonucu elde etmiştir. Ortalama kural uzunluğu açısından Sunulan Yöntem (AOÇ) en iyi başarı sıra değerine sahiptir.



Şekil 8.1. Kıyaslanan Yöntemlerin Sınıflandırma Başarıları, Kural Sayıları ve Ortalama Kural Uzunlukları Açısından Başarı Sıralarını Gösteren Diyagram

Tez çalışması kapsamında BKTS'ler için ideal bulanık kural kümesini oluşturan bir model ortaya konulmaya çalışılmıştır. İdeal bulanık kural kümesi sınıflandırma başarısını en yüksek yapan ve yorumlanabilirliği en yüksek kural kümesi olarak tanımlanmaktadır. Bulanık kural kümesi için yorumlanabilirlik, kural sayısı ve kuralların uzunluğuna bağlı olmaktadır. BKTS sistemlerde sınıflandırma başarısı ile yorumlanabilirlik arasında ters bir bağlantı bulunmaktadır. Yorumlanabilirliğin yüksek olduğu durumlarda sınıflandırma başarısı düşük olmakta veya sınıflandırma başarısının yüksek olduğu durumlarda yorumlanabilirlik düşük olmaktadır. Ortaya konulan model bu çok amaçlı problemi tek bir amaca dönüştürerek çözüme ulaşmaktadır. Yapılan deneysel çalışma sonuçları ışığında Sunulan Yöntem (AOÇ) en yüksek sınıflandırma başarısını elde eden ve yorumlanabilirliği yüksek kural kümelerini oluşturma yeteneğine sahiptir.

8.2. Öneriler

BKTS sistemlerde sınıflandırma işlemi temel olarak bulanık kural kümesi ve çıkarım mekanizmasına bağlı olarak gerçekleştirilmektedir. Çıkarım mekanizması bulanık kuralları kullanarak sınıflandırma işlemini gerçekleştirmektedir. Bu tez çalışmasında tasarlanan genetik algoritma, ideal bulanık kural kümesini aramaktadır. Arama işlemi çok büyük bir arama uzayında gerçekleşmektedir. Genetik algoritmalarda başlangıç popülasyonunun sezgisel yöntemlerle amaca uygun olarak oluşturulması algoritmanın çözümü bulunma suresini ve bulunan çözümün kalitesini olumlu yönde etkileyebilir.

Yeni bir araştırma konusu olan transfer öğrenme, genetik algoritmalarda başarı ile uygulanabilmektedir. Genetik transfer öğrenmenin amacı; benzer makine öğrenmesi problemlerinin çözülmesi sürecinde genetik algoritmalar arasında genetik bireyler transfer edilerek çözüm süresi kısaltılmaya ve çözüm kalitesi artırılmaya çalışılmaktır. Transfer edilecek genetik bireylerin oluşturulmasında farklı sezgisel yöntemler kullanılmaktadır. Bu yöntemler genetik bulanık sistemlerde, genetik popülasyonun oluşturulmasında kullanılabilir. Böyle etkin bir popülasyon oluşturma yöntemi istenilen sonuca daha hızlı bir şekilde yaklaşmayı sağlayabilir.

KAYNAKLAR

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.

Baykal, N. ve Beyan, T., 2004a, Bulanık Mantık İlke ve Temeleri, *Bıçaklar*, 413 sayfa.

Baykal, N. ve Beyan, T., 2004b, Bulanık Mantık Uzman Sistemler ve Denetleyiciler, *Bıçaklar*, 509 sayfa.

Berlanga, F.J., Rivera, A.J., Del Jesus, M.J. and Herrera, 2010, F., GP-COACH: Genetic Programming-Based Learning of Compact and Accurate Fuzzy Rule-Based Classification Systems for High-Dimensional Problems, *Information Sciences*, 180, 1183-1200.

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

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J., 1984, Classification and regression trees. *Taylor & Francis*, 368 sayfa.

Cordon, O., Del Jesus, M.J. and Herrera, 1998, Genetic Learning of Fuzzy Rule-Based Classification Systems Cooperating with Fuzzy Reasoning Methods, *International Journal of Intelligent Systems*, 13, 1025-1053.

Cordon, O., Del Jesus, M.J. and Herrera, F., 1999a, A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems, *International Journal of Approximate Reasoning,* 20(1), 21-45.

Cordon, O., Del Jesus, M.J., Herrera, F. and Lozano, M., 1999b, MOGUL: A Methodology to Obtain Genetic Fuzzy Rule-Based Systems Under the Iterative Rule Learning Approach, *International Journal of Intelligent Systems*, 14, 1123-1153.

Cordon, O., Herrera, F., Gomide, F. and Hoffmann, H., 2001, Ten years of genetic fuzzy systems: current framework and new trends, *IFSA World Congress and 20th NAFIPS International Conference*, 3, 1241-1246.

Cordon, O., 2011, A Historical Review of Evolutionary Learning Methods for Mamdani-Type Fuzzy Rule-Based Systems: Designing Interpretable Genetic Fuzzy Systems, *International Journal of Approximate Reasoning*, 52, 894-913.

DELVE projesi http://www.cs.utoronto.ca/~delve/data/datasets.html, [erişim tarihi: 10.03.2014].

Demsar, J., 2006, Statistical Comparisons of Classifiers over Multiple Data Sets, *Journal of Machine Learning Research,* 7,1-30.

Dennis. B. and Muthukrishnan S., 2014, AGFS: Adaptive Genetic Fuzzy System for medical data classification, *Applied Soft Computing,* 25, 242-252.

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

Freitas A.A., 2002, Data Mining and Knowledge Discovery with Evolutionary Algorithms, *Springer Science & Business Media,* 264 pages.

Goldberg, D. E., 1989. Genetic algorithms in search, optimization and machine learning. *Addison-Wesley*, 412 sayfa.

González, A. and Perez, R., 2001, Selection of relevant features in a fuzzy genetic learning algorithm, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 31(3), 417-425.

Herrera, F., 2008, Genetic fuzzy systems: taxonomy, current research trends and prospects, Evolutionary Intelligence, 1(1), 27-46.

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

Holland, J., 1975, Adaptation in Natural and Artificial Systems, *The University of Michigan Press, Ann Arbour,* 183 sayfa.

Holland, J., 1986, Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. I*n: Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann Publishers, 2, 593-623.

Holmblad, P. and Østergaard, J.-J., 1982, Control of a Cement Kiln by Fuzzy Logic, Gupta M. And Sanchez E. (ads.), *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, 398-399

Hong, T. P., Kuo, S. C. and Chi, C. S., 2001, Trade-off between Computation Time and Number of Rules for Fuzzy Mining from Quantitative Data, *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 9(5), 587-604.

Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H., 1995, Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Transaction on Fuzzy Systems,* 3(3), 260-270.

Ishibuchi, H., Nakashima, T. and Murata, T., 1997, Comparison of the Michigan and Pittsburgh Approaches to the Design of Fuzzy Classification Systems, *Electronics and Communications in Japan*, 80(12), 379-387.

Ishibuchi, H., Murata, T. and Türkşen, I.B., 1997, Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems, *Fuzzy Set and Systems*, 89, 135-150.

Ishibuchi, H., Nakashima, T. and Morisawa, T., 1999a, Voting in Fuzzy Rule-Based Systems for Pattern Classification Problems, *Fuzzy Set and Systems*, 103(2), 223-238.

Ishibuchi, H., Nakashima, T. and Murata, T., 1999b, Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems, *IEEE Transactions on Systems, Man and Cybernetics*, 29(5), 601-618.

Ishibuchi, H., Nakashima, T. and Murata, T., 2001a, Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences*, 136, 109-133.

Ishibuchi, H., Yamamoto, T. and Nakashima, T., 2001b, Fuzzy Data Mining: Effect of Fuzzy Discretization, *Proc. 1st IEEE International Conference on Data Mining,* 59-88.

Ishibuchi, H. and Yamamoto, T., 2004, Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining, *Fuzzy Set and Systems*, 141, 59-88.

Ishibuchi, H. and Yamamoto, T., 2005a, Rule Weight Specification in Fuzzy Rule-Based Classification Systems, *IEEE Transaction on Fuzzy Systems,* 13(4), 428-435.

Ishibuchi, H., Yamamoto, T. and Nakashima, T., 2005b, Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems, *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics,* 35(2), 359-365.

Kaya. M., 2006, Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules, *Soft Computing,* 10(7), 578–586.

KEEL yazılım aracı, http://www.keel.es/, [erişim tarihi: 04.06.2013].

Kızılkaya. E., Karaoğan. I. and Pardalos., P. M., 2012, hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems, *Applied Soft Computing,* 12(2), 800-806.

Kohavi, R., 1995, A study of cross validation and bootstrap for accuracy estimation and model selection. The Fourteenth International Joint Conference on Artificial Intelligence, San Francisco, CA:Morgan Kaufman , 1137-45.

Leung, K.S., Leung, Y., So, L. and Yam, K. F., 1992, Rule Learning in Expert Systems Using Genetic Algorithms: 1, concepts. *In Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks*, 201-204.

Li, Y., Yang, Y., Zhou, L. and Zhu, R., 2009, Observations On Using Problem-Specific Genetic Algorithm For Multiprocessor Real-Time Task Scheduling, *International Journal of InnovativeComputing, Information and Control*, 5(9), 1349-4198.

Lopez. V., Fernandez. A., Del Jesus, M.J. and Herrera, F., 2013, A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets, *Knowledge-Based Systems,* 38, 85-104.

Mamdani, E.H. and Assilian, S., 1975, An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7, 1-13.

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 Transaction on Fuzzy Systems,* 16(4), 1061-1071.

Marwala, T. and Chakraverty, S., 2006, Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm, *Curr Sci India*, 90, 542-548.

Ross, T.J., 1995, Fuzzy Logic with Engineering Applications, *McGraw-Hill Inc*, United States of America, 600 sayfa.

Saade, J. J. and Diab, H. B., 2000, Defuzzification techniques for fuzzy controllers, *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics,* 30(1), 223-229.

Smith, S. F., 1983, Flexible Learning of Problem Solving Heuristics Through Adaptive Search, *Proc, 8th IJCAI*, 1, 422,425.

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.

Thrift. P., 1991, Fuzzy logic synthesis with genetic algorithms, *In: Proceedings of 4th international conference on genetic algorithms (ICGA'91)*, 509–513.

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.

Wilcoxon, F., 1945, Individual comparisons by ranking methods, *Biometrics Bulletin,* 1(6), 80-83.

Wong, M. and Leung, K., 2000, Data Mining using Grammar based Genetic Programming and Applications, *Kluwer Academics Publishers,* 213 sayfa.

Yamakawa, T., 1993, A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic controller, *IEEE Trans. Neural Networks*, 4, 496-522.

Zadeh, L.A. 1965, Fuzzy Sets, *Information and Control*, 8, 338-353

Zhang, B. S. and Edmunds, J. M., 1991, On Fuzzy Logic Controller, *Proceedings of International Conference Control '91,* 961-965.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Ersin KAYA
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	01.09.1980 - Erzincan
Telefon	**:**	+90 505 503 76 37
Faks	**:**	+90 332 241 06 35
e-mail	**:**	ersinkaya@selcuk.edu.tr

EĞİTİM

Derece		**Adı**	**Bitirme Yılı**
Lise	:	Erzincan Nevzat Ayaz Fen Lisesi - Erzincan	30.06.1997
Üniversite	:	Selçuk Üniversitesi Mühendislik Mimarlık Fak. Bilgisayar Mühendisliği - Konya	29.06.2001
Yüksek Lisans	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. A.B.D. - Konya	01.09.2005
Doktora	:	Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. A.B.D. - Konya	

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2001-2008	Selçuk Üniversitesi	Araştırma Görevlisi
2008-devam	Selçuk Üniversitesi	Uzman

UZMANLIK ALANI: Bulanık Mantık, Evrimsel Algoritmalar, Kaba Kümeler

YABANCI DİLLER: İngilizce

YAYINLAR

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

1. Kaya E., Koçer B.,Arslan A., A single-objective genetic-fuzzy approach for multi-objective fuzzy problems, Journal of Intelligent and Fuzzy Systems, vol: 25(3), 557-5661, 2013. (Doktora Tezinden)

2. Kaya E., Koçer B., Arslan A., Learning Weights of Fuzzy Rules by Using Gravitational Search Algorithm, International Journal of Innovative Computing Information and Control, vol: 9(4), pp: 1593–1601 , 2013.

3. Kaya E., Fındık O., Babaoğlu İ., Arslan A., Effect of Discretization Method on the Diagnosis of Parkinson's Disease, International Journal of Innovative Computing Information and Control, vol: 7(8), pp: 4669–46782 , 2011.

Uluslararası Bilimsel Toplantılarda Sunulan Bildiriler:

1. Kaya E., Oran B., Arslan A., *A Diagnostic Fuzzy Rule-Based System for Congenital Heart Disease,* International Conference on Computer, Electrical, and Systems Sciences, and Engineering (ICCESSE 2011), vol: 5 pp: 210-213, Amsterdam / Netherlands, 2011 (Doktora Tezinden)

2. Kaya E., Oran B., Arslan A., *A rough sets approach for diagnostic M-mode evaluation in newborn with congenital heart diseases,* 2th International Conference on Human System Interaction (HSI 2010), 119-123, Rzeszow / Poland, 2010 (Yüksek Lisans Tezinden)

Ulusal Bilimsel Toplantılarda Sunulan Bildiriler:

1. Bildirici İ.Ö., Yıldız F., Babaoğlu İ., Kaya E., *DIGITAL CHART OF THE WORLD COĞRAFİ VERİ TABANININ KARTOGRAFİK OLARAK KULLANIMI*, Selçuk Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Öğretiminde 30. Yıl Sempozyumu, sf: 364-372, 15-16-18 Ekim 2002, Selcuk Üniversitesi, Konya,Türkiye