

155115

KOCAELİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**BULANIK MANTIK TEMELLİ AKILLI BİR
SORGULAMA ARACI**

YÜKSEK LİSANS TEZİ

Bilgisayar. Müh. Sevinç İLHAN

Anabilim Dalı: Bilgisayar Mühendisliği

Danışman: Yrd. Doç. Dr. Nevcihan DURU

HAZİRAN 2004

**BULANIK MANTIK TEMELLİ AKILLI BİR
SORGULAMA ARACI**

YÜKSEK LİSANS TEZİ

Bilgisayar. Müh. Sevinç İLHAN

Tezin Enstitüye Verildiği Tarih : 25 Mayıs 2004

Tezin Savunulduğu Tarih : 2 Temmuz 2004

Tez Danışmanı

Üye

Üye

Yrd.Doç.Dr. Nevcihan DURU

Prof.Dr. Hüseyin KARABELLİ

Doç.Dr. Kadir ERKAN

(.....)

(.....)

(.....)

HAZİRAN 2004

BULANIK MANTIK TEMELLİ AKILLI BİR SORGULAMA ARACI

Sevinç İLHAN

Anahtar Kelimeler: Esnek Sorgu, Akıllı Veri Tabanları, Bulanık Sorgulama, Bulanık Veri Tabanları.

Özet: Bu çalışmada, bulanık mantık teorisi kullanılarak insan düşünce sistemine yakın akıllı bir sorgulama aracı geliştirilmiştir. Klasik veri tabanı sorgulama sisteminden farklı olarak konuşma dilindeki ifadeler yardımı ile sorgu cümleleri oluşturulmuştur. Bu doğrultuda geliştirilen akıllı sorgulama aracı ile, veri tabanında kayıtlı herhangi bir tablo üzerinde işlem yapılabilir. Geliştirilen sistemin standart sorgulama sistemine göre avantajlarını gösterebilmek için, öğrencilerin ekonomik durumlarına göre burs ihtiyaçlarını belirleyen bir veri tabanı üzerinde değerlendirme yapılmıştır. Bu karşılaştırma sonucunda, akıllı sorgu metodunun daha esnek bir yaklaşım sunduğu gözlenmiştir.

FUZZY LOGIC BASED INTELLIGENT TOOL

Sevinç İLHAN

Key words: Flexible Query, Intelligent Databases, Fuzzy Query, Fuzzy Databases

Abstract: In this study, a software tool, enabling fuzzy query from a classical database and behaving like a human thought system is developed. In a classical database query system, the query sentences are formed with SQL. However in our developed intelligent software tool the query sentences are formed with linguistic terms. And also any operation on any table registered in the database could be done. In order to see the advantages of our software system, an evaluation has been done in a database including data about students to determine scholarship needs. It is concluded that, the fuzzy query method is more flexible and the results of fuzzy query are more predictive.

ÖNSÖZ ve TEŞEKKÜR

Günümüzde veri işleme işinde, veri toplamak kadar, veriler üzerinde belirlenen amaca hizmet edecek sağlıklı sınıflandırma yapabilmek de önem kazanmıştır. Ancak bu şekilde, eldeki verilerden daha anlamlı bilgiler elde edilebilir. Standart veri tabanı sorgulama dili olan SQL' in yapısal formundan bağımsız olabilmek, kullanıcıya sorgu cümlelerini oluştururken maksimum esneklik sağlayacaktır. Bulanık mantığın çok boyutlu değerlendirmesinin, klasik mantığın iki boyutlu değerlendirmesine göre insan sezgilerine yakın sonuçlar verebildiği düşünüldüğünde; veri tabanlarındaki klasik sorgu kısıtlamalarından sıyrılıp daha esnek sorgulama yapabilecek ve insan sezgilerine daha yakın sonuçlar verebilecek akıllı bir sorgu aracı geliştirilmek istenmiştir.

Bu çalışmanın gerçekleştirilmesinde, tüm destek ve yardımlarından dolayı aileme ve arkadaşlarıma, fikirleri ve desteği ile yardımcı olan Tez Danışmanım Yrd. Doç. Dr. Nevcihan DURU' ya (KOÜ.M.F.) teşekkürlerimi sunarım.

Haziran 2003, KOCAELİ

Sevinç İLHAN

İÇİNDEKİLER

ÖZET.....	ii
ABSTRACT.....	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
İÇİNDEKİLER	v
SİMGELER DİZİNİ ve KISALTMALAR	vii
ŞEKİLLER DİZİNİ.....	viii
TABLolar DİZİNİ	x
BÖLÜM 1: GİRİŞ.....	1
BÖLÜM 2. KÜME TEORİSİ ve BULANIK MANTIK	5
2.1 Giriş.....	5
2.2 Klasik ve Bulanık Kümeler.....	5
2.2.1 Klasik Küme Teorisi	5
2.2.2 Bulanık Küme Teorisi	8
2.3 Üyelik Fonksiyonlarının Kısımları	13
2.4 Üyelik Fonksiyonlarının Özellikleri	14
2.5 Neden Bulanık Mantık?	15
2.6 Bulanık Sistemlerin Gelişimi	17
2.7 Bulanık Denetim	19
2.7.1 Bulanıklaştırma	20
2.7.2 Üyelik Fonksiyonlarının Oluşturulması.....	21
2.7.3 Bilgi Tabanı.....	27
2.7.4 Bulanık Kurallar ve Çıkartım.....	27
2.7.5 Durulaştırma.....	32
BÖLÜM 3: ESNEK SORGULAMA ARAÇLARI.....	37
3.1 Giriş.....	37

3.2 Bulanık Küme Teorsisi ile Geliştirilen Esnek Sorgu Araçları.....	38
3.3 Sonuç.....	43
BÖLÜM 4 : AKILLI SORGULAMA ARACI.....	45
4.1 Giriş.....	45
4.2 Neden Bulanık Sorgu	46
4.3 Programın Genel Yapısı.....	49
4.4 Niteliklerin Bulanıklaştırılması.....	50
4.5 Bulanık Sorgulama.....	56
4.5.1 Dilsel Değiştiriciler	62
4.5.2 Bölüm Kesmesi	65
4.6 Raporlama	67
4.7 Bulanık Sorgu Kaydetme	67
SONUÇLAR ve ÖNERİLER.....	68
KAYNAKLAR	69
EK-1	72
EK-2.....	74
EK-3	76
KİŞİSEL YAYINLAR ve ESERLER.....	80
ÖZGEÇMİŞ	81

SİMGELER DİZİNİ ve KISALTMALAR

ASA	Akıllı Sorgu Aracı
KDD	Knowledge Discovery
SQL	Structured Query Language
VTYS	Veri Tabanı Yönetim Sistemi



ŞEKİLLER DİZİNİ

Şekil 2.1a. Birim Aralıkta Üyelik Fonksiyonu.....	9
Şekil 2.1b. Üçgen Şeklinde Üyelik Fonksiyonu.	9
Şekil 2.1c. Yamuk Şeklinde Üyelik Fonksiyonu.	10
Şekil 2.1d. Çan Eğrisi Şeklinde Üyelik Fonksiyonu.	10
Şekil 2.2 Bir Üyelik Fonksiyonun Kısımları.	14
Şekil 2.3a. Normal Bulanık Küme.	15
Şekil 2.3b. Normal Olmayan Bulanık Küme.	15
Şekil 2.4. Bir Bulanık Denetim Sisteminin Blok Diagramı.....	20
Şekil 2.5. x_0 Değişkeninin Üyelik Gösterimi.	21
Şekil 2.6. Üyelik Fonksiyonlarının Sağ-Sol Genişlikleri ve Tepe Değeri.....	22
Şekil 2.7. Üyelik Fonksiyonlarında Kesişim Noktası.....	23
Şekil 2.8. Üyelik Fonksiyonları Arasındaki Kesişim Noktası Oranı Sıfır.....	23
Şekil 2.9. 0.7'den Kırpılmış bir Üyelik Fonksiyonu	24
Şekil 2.10. Asimetrik bir Üyelik Fonksiyonu.	25
Şekil 2.11a. e Değişkeninin PM ve PS Üyelik Fonksiyonları.	25
Şekil 2.11b. Aynı Üyelik Fonksiyonların Diğer bir Gösterimi.....	25
Şekil 2.12. Koşulun Sağlanması Durumu.....	26
Şekil 2.13. Koşulun Sağlanmaması Durumu	26
Şekil 2.14. Max-Min Çıkartım Yöntemi.....	28
Şekil 2.15. Max-Product Çıkartım Yöntemi.	29
Şekil 2.16a. Sıcaklık Giriş Değişkenine Ait Üyelik Fonksiyonları.....	30
Şekil 2.16b. Nem Giriş Değişkenine Ait Üyelik Fonksiyonları.....	30
Şekil 2.16c. Su Miktarı Çıkış Değişkenine İlişkin Üyelik Fonksiyonları	31
Şekil 2.17. Üyelik Fonksiyonları ve İki Kural Sontucu Yapılan Çıkartım.....	31
Şekil 2.18. Ağırlık Merkezi Dululaştırma Yöntemi.	33
Şekil 2.19. Toplamların Merkezi Yöntemi.	34
Şekil 2.20. Yükseklik Yöntemi.	35
Şekil 2.21. En Büyük Alanın Merkezi Yöntemi.	35

Şekil 2.22. İlk En Büyük Yöntemi.	36
Şekil 2.23. İlk En Büyük Orta Yöntemi.	36
Şekil 3.1a. “Uzun” Bulanık Kümesi.	42
Şekil 3.1b. “Şişman” Bulanık Kümesi.	42
Şekil 4.1. “Fnotu Mükemmel ve Devam İyi” Sorgusunun Sonuçları.	49
Şekil 4.2. Programa Ait Akış Diagramı.	50
Şekil 4.3. Burs Tablosu.	52
Şekil 4.4. Bulanık Küme Seçim Ekranı.	53
Şekil 4.5a. “Aylık Geliri” Nitelik Değerine Ait Üyelik Fonksiyonları.	53
Şekil 4.5b. “Kardeş” Nitelik Değerine Ait Üyelik Fonksiyonları.	54
Şekil 4.6a. Üçgen Üyelik Fonksiyonu.	54
Şekil 4.6b. Yamuk Üyelik Fonksiyonu.	55
Şekil 4.7a. Bulanık Küme Yardımı ile Sorgu Oluşturma.	57
Şekil 4.7b. Mantıksal İfadeler ile Sorgu Oluşturma.	57
Şekil 4.8. Sorgu Sonucu.	62
Şekil 4.9. “Aylık Geliri Çok Yüksek Değil” Bulanık Sorgu Sonucu.	64
Şekil 4.10. Eşik Değeri ile Bulanık Sorgulama.	66
Şekil 4.11. Sorgu Raporu.	67

TABLÖLAR DİZİNİ

Tablo 3.1. Sorgu Sonuçları.....	.41
Tablo 4.1. Laboratuar Tablosu47
Tablo 4.2. Bağlaç Dinamik Dizisi58
Tablo 4.3. Dinamik Dizi60



1. GİRİŞ

İnsan, çevresinde algıladıklarını tanımlarken ya da bunları sınıflandırırken kesin ifadelerden çok bulanık ifadelere başvurur. Bu yönüyle insan düşüncelerinin büyük çoğunluğu bulanıktır denilebilir. Örneğin, hissedilen sıcaklık, sıcak ve soğuk diye ifade edilmekten çok, “ılık”, “çok sıcak”, “çok soğuk” şeklinde ifade edilir. Bulanık ifadelerle başvurmanın bir nedeni, iki boyutlu mantık yerine çok boyutlu mantığın insan düşüncesine daha yakın olmasıdır, diğer bir nedeni de şu olabilir: zaman içerisinde veriler aynı kalsa dahi, insanın yaptığı değerlendirmelerde değişiklikler olabilir. Ya da veriler, insandan insana, sistemden sisteme farklılık gösterebilirler. Örneğin, bazı insanlar tarafından aylık geliri bir milyar olanlar zengin olarak kabul edilirken başka bir grup insan için aylık geliri beş milyar ve üzeri olanlar zengin olarak değerlendirilebilir. O halde, verilerin sınıflandırılması ve değerlendirilmesi işlemi tamamen sezgiseldir. İçinde bulunulan zamana ve değerlendirmeyi yapan kişiye göre değişiklikler gösterebilmektedir. Kesin değerlere dayanan düşünce yöntemi yerine, yaklaşık değerleri kullanan bulanık mantık, insan düşüncesine yakın olması nedeniyle şimdiye kadar birçok sistemde kullanılmıştır. Klasik mantık ile modellenmiş olan sistemlerden beklenen sonuçlar alınmadığı zamanlarda ya da klasik mantığın yetersiz kaldığı durumlarda, klasik mantık kurallarına göre daha üstün olmaları nedeniyle bulanık mantık kuralları ile sistemin modellenmesi yoluna gidilmiştir. Bulanık mantığın uygulanacağı sistemlerin, davranışının kuralları ile ifade edilebiliyor olması ve karmaşık matematiksel işlemler gerektirmiyor olması gerekmektedir. Bunun aksi bir sistemde bulanık mantık etkin çözümü sağlamayacaktır ve büyük olasılıkla beklenen değerleri vermeyecektir (Elmas 2003). Bu özellikleri nedeniyle bulanık mantık birçok farklı endüstriyel alanda uygulamalara dahil olmuştur. Çimento sanayi, otomotiv sanayi, otomatik tren sistemleri, borsa analizleri, akıllı çamaşır makineleri, asansör sistemleri, fotoğraf makineleri, mikrodalga fırınlar ve veri tabanı sistemleri gibi bir çok alandan söz

edebiliriz. Bulanık mantık uygulamaları, klasik mantık uygulamalarından daha optimum çözümler sunabildiği için gün geçtikçe artış göstermektedir.

Bulanık mantık ile gerçekleştirilen uygulamalar gözden geçirildiğinde, çok farklı alanlarda uygulandıkları gözlenmektedir. Bulanık küme teorisi, 1965 yılında Zadeh tarafından ortaya atıldıktan sonra bir çok bilim adamı tarafından araştırma konusu olmuş ve birçok endüstriyel uygulamada hayata geçirilmiştir. Mamdani, buhar türbininin hızının denetlenmesinde “Eğer türbin hızı çok hızlı artıyorsa ve basınç da çok düşük ise, buhar vanasını biraz aç” şeklindeki kurallar ile bir bulanık sistem geliştirmiştir. Bu çalışma bulanık denetim üzerinde yapılan ilk çalışmadır (Schwartz and Klir 1992). Endüstriyel alandaki ilk önemli uygulama, çimento sanayiinde olmuştur. Çimento sanayiinde, değirmen içerisindeki sıcaklık ve oksijen oranı, kaliteli üretim açısından oldukça önemlidir. Bir firma, doğrusal bir model üzerine kurulu geleneksel denetleyici yerine, bulanık mantık denetleyici kullanarak çok daha başarılı sonuçlar veren bir uzman sistem geliştirmiştir (Schwartz and Klir 1992). Bu uygulamadan sonraki en önemli aşama, Japonya’da 1987 yılında faaliyete geçen Sendai metrosudur. Yasunobu ve Miyamoto, Japonya’daki bir otomatik tren sisteminin denetiminde bulanık mantık kullanmışlardır. Sabit hız ve otomatik duruş denetiminin esas olarak sağlandığı bu sistem, 1987 yılından beri kullanılmaktadır (Berenji 1992). Japonya’daki bir diğer uygulama da, ATM makinalarında banknotların ayrılmasındaki hızın bulanık mantık ile artırılması uygulamasıdır Trafik denetimi ile ilgili birçok uygulamada da yine bulanık mantık kullanılmıştır. Örnek olarak, ABS fren sisteminde bulanık mantık kullanılmıştır. Aynı şekilde, motor hız denetimi uygulamalarında da bulanık mantık kullanılmıştır. Matsushita ve Hitachi'nin çamaşır makinası, çamaşırın cinsi ve kirliliği, suyun durumu gibi şartları özel sensörlerle tesbit ettikten sonra, deterjanı, devir sayısı ve süresini ayarlamakta ve bu şekilde mevcut şartlarda en optimum şekilde çalışmaktadır. Sanyo ve Matsushita'nın Video kamerası, Toshiba, Sharp, Sanyo ve Hitachi'nin mikrodalga fırınları, Canon'un fotoğraf makinası bulanık mantık uygulamaları arasında sayılabilir (Berenji 1992).

Veri tabanı sistemlerinde de bulanık mantık kullanılarak bir çok uygulama geliştirilmiştir. Veri tabanları, çok miktarda veri toplamak ve bu verilerden etkili

veriler elde edebilmek için kullanılmaktadırlar (Umano at al 1995). Veri tabanı sistemlerini düşündüğümüzde, verilerin kurallar ile değerlendirilebildiği basit sayısal ve mantıksal işlemler ile sorgulanabildikleri görülmektedir. O halde mevcut bir sistemde, kuralları sizin koymanız, verilerin istediğiniz doğrultuda değerlendirilmesini ve karar alma mekanizmasının tamamen sezgileriniz doğrultusunda olmasını sağlamaktadır. Bu anlamda bulanık mantık kuralları, veri tabanı sistemlerine rahatlıkla uygulanabilir. Veri tabanında saklanan verilerin sorgular ile değil de bulanık sorgular ile değerlendirilmesi elde edilen sonuçların insan sezgilerine daha yakın olmasını ve üzerinde çalışılan sistem hakkında daha sağlıklı kararlar alınmasını sağlamaktadır. Daha esnek ve sezgisel sorgulama yaparak istenilen bilgiye daha sağlıklı şekilde erişebilecek yeni nesil bir sorgu aracına ihtiyaç duyulması gerçeğinden sonra, bulanık mantık ve veri tabanı teknolojilerini birleştiren esnek sorgulamaların gerçekleştirilebileceği sorgu araçları geliştirilmiştir. Literatürde esnek sorgu araçlarından üç tanesi üzerinde yoğunlaşmıştır.

- Bulanık mantık küme teorisi ile esnek sorgulama sistemi ilk olarak Tahani (1977) tarafından geliştirilmiştir.
- 1986 yıllarında Kocprzyk ve Ziolkowski tarafından bir esnek sorgulama sistemi daha geliştirilmiştir.
- Bu yılların sonlarında Patrick Bosc (1980), bulanık olmayan(crisp) değerlerden oluşan veritabanları için esnek sorgulama sistemi geliştirdi.

Bu uygulama geliştirilmeye başlamadan önce incelenen diğer esnek sorgulama örnekleri şu şekilde sıralanabilir: Ribiero (1999), Cox (2000), Eminov, Galindo (2002). Bulanık küme yaklaşımı esas alınarak geliştirilen ve hedefi dilsel özetlemelerin bulanık sorgunun bir parçası olması şeklinde özetlenen bir diğer sorgulama aracı da Dan Rasmussen ve Ronald R.Yager (1997) tarafından geliştirilmiştir.

Yukarıda adı geçen çalışmalar incelendiğinde, klasik sorguların getirdiği sınırlamalardan sıyrılmış, daha esnek ve insan düşüncesine daha yakın olduğu için de akıllı diye adlandırılan sorgu araçlarının geliştirildiği görülmektedir.

Bu yüksek lisans tezinde gerçekleştirilmek istenen, klasik veri tabanlarında bulanık sorgulama yapabilecek bir sorgu arayüzü geliştirmektir. Standart sorgu dili olan SQL' in yapısal formundan bağımsız olarak tasarlanan bu sorgu arayüzünde, konuşma dilindeki dilsel ifadeler yardımı ile sorgulama gerçekleştirilmektedir. Tüm klasik ve bulanık veriler veri tabanında saklanmaktadır. Eldeki tüm verilerin sorguda aranan özelliklere uygunlukları, geliştirilen diğer uygulamalarda olduğu gibi [0,1] arasında derecelendirilerek elde edilmekte ve elde edilen sonuçlar da yine veri tabanında saklanmaktadır. Geliştirilen yazılım, özel bir amaca hizmet edecek şekilde tasarlanmamış olup, seçilen herhangi bir veri kümesi üzerinde sorgulama yapabilecek genel bir esnek sorgu arayüzüdür. Problemin temel olarak yukarıda sunulduğundan sonra, çalışmanın gelişimi aşağıda açıklanmaktadır.

Bu tez yukarıda anlatılan işlemlerin paralelinde dört bölüm halinde yazılmıştır. Birinci bölümde tez çalışmasında kullanılan bulanık mantık ve veri tabanı sistemleri hakkında genel bilgi verilmiş, literatür incelenmesi yapılmış ve uygulamanın amacı hakkında genel bilgi verilerek tezin genel bir tanımı yapılmıştır.

İkinci bölümde, klasik ve bulanık küme teorisi, bulanık mantığa ait genel tanımlar, bulanık denetim sistemlerindeki aşamalar hakkında bilgi verilmiştir.

Üçüncü bölümde, literatürde önemli kabul edilen esnek sorgu araçları incelenmiştir. Bunun yanında bu çalışmadan önce incelenmiş diğer bulanık sorgu araçları ve bunlara ait sorgulama yöntemleri hakkında bilgi verilmiştir.

Dördüncü bölümde, esnek sorgulamaya neden ihtiyaç duyulduğu anlatılmış ve belirlenen sorunlara geliştirilen sorgu arayüzünün nasıl çözüm getirdiği ayrıntıları ile incelenmiştir. Yazılımın gelişim aşamaları incelenmiştir.

Tezin son bölümünde genel olarak elde edilen sonuçlara ve bunların gelecekteki çalışmalara nasıl yön verebileceği konusunda bazı saptamalara yer verilmiştir.

2. KÜME TEORİSİ ve BULANIK MANTIK

2.1. Giriş

Bu bölümde ilk olarak, klasik küme teorisi, bulanık küme teorisi, bunlara ait temel kavramlar ve bulanık kümelerdeki işlemler anlatılmış; daha sonra bulanık mantığın önemine değinilmiş ve iki değerlilik prensibine dayanan klasik mantıktan farkı açıklanmıştır. Bu bölümün amacı, Bölüm 4' de tanıtılacak bulanık mantık temelli akıllı sorgu uygulamasının bulanık mantık kısmına ilişkin alt yapının oluşturulmasıdır.

2.2. Klasik ve Bulanık Kümeler

2.2.1. Klasik küme teorisi

Klasik küme yaklaşımında bir eleman, ya bir kümenin elemanıdır, yani kümeye üyelik derecesi 1'dir; ya da kümenin elemanı değildir, yani kümeye üyelik derecesi 0'dır. Klasik küme yaklaşımında bir eleman bir kümeye ya 0 ya da 1 üyelik derecesi ile bağlıdır. Bu durumdan iki değerlilik prensibi diye bahsedilmektedir. Klasik mantıkta bir eleman birden fazla kümeye üye olabilmektedir. Örneğin "insanlar" evrensel kümesinde yer alan elemanlar, bu evrensel kümenin alt kümeleri olan öğrenci, öğretmen, mühendis kümelerinden birden fazlasına aynı anda üye olabilirler. Bir kişi mühendislik yaparken, aynı zamanda öğrencilik de yapıyor olabilir. Bu noktada bir problem yoktur. Klasik mantığın asıl problemi, bir elemanın ait olduğu tüm kümeler üyelik derecesinin 1 olmasıdır. Başka bir örnekte "Boy" evrensel kümesinde tanımlı alt kümelerin "kısa", "orta", "uzun", "çok uzun" olduğunu düşünelim. "Uzun" alt kümesi, $170 \text{ cm} \leq \text{Boyu} \leq 190 \text{ cm}$ olarak; "Çok uzun" alt kümesi de $190 \text{ cm} \leq \text{Boyu} \leq 220 \text{ cm}$ olarak belirlensin. Bu durumda boyu 190 cm olan bir insan aynı anda hem uzun boylu, hem de çok uzun boylu kabul

edilecektir. Ve iki farklı kümeye de üyelik derecesi 1 olacaktır. Başka bir değerlendirmede, boyu 190.1 cm olan bir kişi çok uzun boylu kabul edilecektir. Klasik kümelere ait sınırlar çok kesin olduğundan aralarında 0.1 cm boy farkı olan iki insandan biri uzun boylu, diğeri çok uzun boylu olarak tanımlanmaktadır.

Klasik mantıkta kümelerin sınırları ve elemanların tanımları kesindir. Oysa gerçek hayatta her kümenin sınırları ve dolayısıyla bu kümelere ait elemanların tanımları o kadar kesin değildir. Dolayısıyla klasik küme anlayışı böyle durumlarda yetersiz kalmaktadır (Öztürk at al 2003).

2.2.1.1. Klasik kümeler ile ilgili matematiksel işlemler

X bir evrensel küme ve küme elemanları, A kümesi ve x elemanı olsun. Bu durum şu şekilde ifade edilir:

$$X = \{A, x\}$$

X evrensel kümesinin bir alt kümesi olan A kümesi ise $A = \{a\}$ elemanından oluşsun. “a”, A kümesinin elemanıdır ve “a” ’nın A kümesine üyelik derecesinin 1 olduğu anlamına gelir. “x” elemanı ise A kümesinin elemanı değildir ve “x” ’in A kümesine üyelik derecesi 0’dır. Bu durum,

$$\mu_A = \left\{ \frac{1}{a}, \frac{0}{x} \right\} \quad (2.1)$$

şeklinde ifade edilir.

2.2.1.2. Klasik küme işlemleri

A ve B kümeleri, X evrensel kümesinin alt kümeleri olsun.

Tümleme: A kümesine dahil olmayan, X evrensel kümesine dahil olan kümeye A’nın tümleyeni denir.

$$\bar{A} = \{x | x \notin A \wedge x \in X\} \quad (2.2)$$

Alt Küme: A kümesi, B kümesinin alt kümesi ise A kümesinin tüm elemanları, B kümesine aittir.

$$A \subset B \quad (2.3)$$

Birleşim: A veya B kümelerine ait elemanların oluşturduğu kümedir.

$$A \cup B = \{x | x \in A \vee x \in B\} \quad (2.4)$$

Kesişim: Hem A, Hem de B kümesine ait olan elemanların oluşturduğu kümedir.

$$A \cap B = \{x | x \in A \wedge x \in B\} \quad (2.5)$$

Fark: Kesişen A ve B kümeleri için, A kümesinin B kümesine farkı, A kümesine ait, B kümesine ait olmayan elemaların kümesidir.

$$A - B = \{x | x \in A \wedge x \notin B\} \quad (2.6)$$

2.2.1.3. Klasik küme özellikleri

Klasik kümelere ait önemli bazı özellikler şunlardır.

Değişme Özelliği:

$$A \cup B = B \cup A \quad \text{ve} \quad A \cap B = B \cap A \quad (2.7)$$

Birleşme Özelliği:

$$A \cup (B \cap C) = (A \cup B) \cap C \quad \text{ve} \quad A \cap (B \cup C) = (A \cap B) \cup C \quad (2.8)$$

Dağılma Özelliği:

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned} \quad (2.9)$$

Tek Kuvvet Özelliği:

$$\begin{aligned} A \cup A &= A, \quad A \cap A = A \\ A \cup \emptyset &= A, \quad A \cup X = X, \quad A \cap X = A, \quad A \cap \emptyset = \emptyset \end{aligned} \quad (2.10)$$

2.2.2. Bulanık küme teorisi

Klasik kümelerin genelleştirilmiş bir şekli olan Bulanık Kümeler, günlük yaşamdaki belirsizliklerin matematiksel sunulma şeklidir (Bezdek 1993). Gerçek dünyada, nesne sınıfları, tam olarak tanımlanmış üyelik derecesine sahip değildir. Örneğin, hayvanlar kümesi köpekleri, atları, kuşları, vb kapsar; bitkileri, kayaları, sebzeleri kapsamaz. Bunun yanında deniz yıldızı, bakteri gibi canlıları, hayvanlar kümesine dahil etme konusunda belirsizlik vardır.

Açıkça, “1’den çok büyük olan reel sayılar”, “Güzel kadınlar”, “uzun insanlar” kümeleri, geleneksel matematik terimler ile ifade edilemeyen kümelelerdir. İnsan düşünce sisteminde, özellikle örüntü tanıma alanında gerçeklerin sınıflandırılması büyük önem taşır (Zadeh 1965).

Bulanık kümelerin sınırları klasik kümelerin aksine, keskin ve net değildir. Hatta kümelerin sınırları iç içe geçmiş durumdadır. Klasik kümelerde bir eleman bir kümeye aittir veya ait değildir. Yani üyelik derecesi 1’dir ya da 0’dır. Oysa bulanık kümelerde bir eleman bir bulanık kümeye 0 ile 1 aralığında üyedir. Başka bir deyişle “biraz aittir biraz değildir.” denebilir. Daha önce verdiğimiz boy örneğine dönersek, Boy uzayının alt kümeleri olarak; “kısa”, “orta”, “uzun”, “çok uzun” verilmişti. “Uzun” alt kümesi, $170 \leq \text{Boyu} \leq 190$ olarak belirlenmişti. Bu durumda boyu 169.9 cm olan bir kişi orta boylu, boyu 190.1 cm olan bir kişi ise çokuzun boylu kabul

edilecektir. Oysa bulanık küme anlayışına göre 170 cm boyundaki bir insan hem “uzun” hem de “orta” bulanık kümelerine farklı üyelik dereceleri kadar aittir.

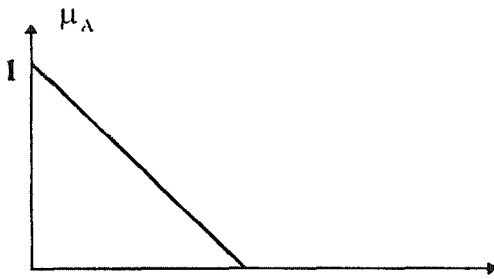
Gerçek hayata baktığımızda, bulanık küme mantığının gerçek hayata daha yakın olduğunu görürüz. Bulanık küme teorisinde bir elemanın tanımlanması için ait olduğu bulanık kümenin adının yanında o bulanık kümeye olan üyelik derecesinin de biliniyor olması gerekmektedir.

Bir bulanık kümedeki üyelik derecesi aşağıdaki gibi gösterilen bir üyelik fonksiyonu ile belirtilir.

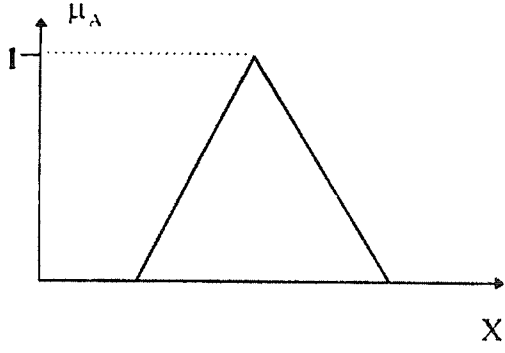
$$\mu_A(x) \rightarrow [0,1]$$

Buna göre, üyelik fonksiyonu gerçel bir sayıdır ve $0 \leq \mu_A \leq 1$ ile gösterilir. Üyelik fonksiyonuna ait ara değerler, üyelik derecesi olarak isimlendirilir. Burada 0 üyelik derecesi, kümenin üyesi olmama; 1 üyelik derecesi ise kümenin tam üyesi olma anlamındadır.

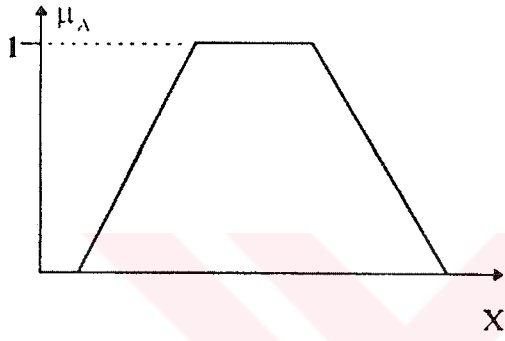
Şekil 2.1a,b,c,d’ de, bir A bulanık kümesine ait farklı üyelik fonksiyonları verilmiştir (Yager and Filev 1994).



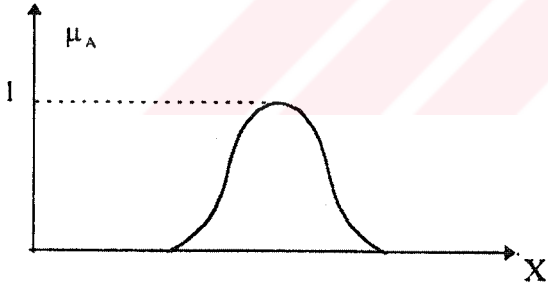
Şekil 2.1. a) Birim Aralıkta Üyelik Fonksiyonu



Şekil 2.1. b) Üçgen (Triangular) Şeklinde Üyelik Fonksiyonu



Şekil 2.1. c) Yamuk (Trapezoidal) Şeklinde Üyelik Fonksiyonu



Şekil 2.1. d) Çan Eğrisi Şeklinde Üyelik Fonksiyonu

X ayrık ve sonlu olmak üzere, bulanık kümeye ait üyeliklerin toplamı aşağıdaki gibi ifade edilir;

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots = \sum_i \frac{\mu_A(x_i)}{x_i} \quad (2.11)$$

X sürekli ve sonsuz ise bu gösterim aşağıdaki şeklini alır

$$A = \int \frac{\mu_A(x)}{x} \quad (2.12)$$

Yukarıdaki ifadelerde kullanılan $\frac{\mu_A(x)}{x}$ ifadesi, “ x’in A’daki üyelik derecesi $\mu_A(X)$ dir.” anlamına gelmektedir. Yani her iki ifadede de görülen yatay çizgiler bir “bölüm çizgisini değil” daha çok bir “sınırı” ifade etmektedir.

Denklem 3.1’deki “+” işaretleri, “cebrik bir toplamayı değil”; fonksiyonlar teorisindeki “birleşimi”, denklem 3.2’deki integral işareti ise “cebrik integrali değil”; daha çok sürekli değişkenler için “birleşim kümesini” göstermektedir (Öztürk at al 2003).

2.2.2.1 Bulanık küme işlemleri

X evrensel küme, \underline{A} ve \underline{B} , X evrensel kümesinde tanımlı bulanık kümeler olsun.

Tümleyen: Herhangi bir x elemanının \underline{A} bulanık kümesindeki üyelik derecesi 0.7 ise tümleyenindeki üyelik derecesi 0.3’tür.

$$\mu_{\underline{A}}(x) = 1 - \mu_{\underline{A}}(x) \quad (2.13)$$

Alt Küme: X evrensel kümesinin her bir elemanının bulanık \underline{A} kümesindeki üyelik derecesi, bulanık \underline{B} kümesindeki üyelik derecesinden küçük veya eşit ise \underline{A} kümesi, \underline{B} kümesinin alt kümesidir.

$$\underline{A} \subset \underline{B} \Leftrightarrow \mu_{\underline{A}}(x) \leq \mu_{\underline{B}}(x) \quad (2.14)$$

Birleşim: $\underline{A} \cup \underline{B}$ kümesinin, x elemanı için üyelik derecesi, \underline{A} ve \underline{B} kümelerinden üyelik derecesi büyük olana eşittir.

$$\mu_{\underline{A} \cup \underline{B}}(x) = \max(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) , x \in X \quad (2.15)$$

Kesişim: $\underline{A} \cap \underline{B}$ kümesinin, x elemanı için üyelik derecesi, \underline{A} ve \underline{B} kümelerinden üyelik derecesi küçük olana eşittir.

$$\mu_{\underline{A} \cap \underline{B}}(x) = \min(\mu_{\underline{A}}(x), \mu_{\underline{B}}(x)) , x \in X \quad (2.16)$$

Destek Keskin Kümesi: X uzayındaki bir \underline{A} bulanık kümesinin destek kümesi, keskin küme olup, X'in \underline{A} bulanık kümesinde 0'dan farklı üyelik derecesine sahip elemanların hepsini içermektedir.

$$\text{Supp}A = \{x \in X | \mu_A(x) > 0\} \quad (2.17)$$

α - Bölüm Kümesi : Bulanık A kümesinin α -bölüm keskin kümesi A_α ile gösterilir ve X evrensel kümesinin A kümesindeki elemanları arasında üyelik derecesi α değerinden büyük ve eşit olanları içerir.

$$A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\} \quad (2.18)$$

Seviye Kümesi : Bulanık \underline{A} kümesinin seviye keskin kümesi \wedge_α ile gösterilir ve X evrensel kümesinin \underline{A} kümesindeki elemanları arasında üyelik derecesi α değerine eşit olanları içerir.

$$\wedge_\alpha = \{x \in X | \mu_A(x) = \alpha\} \quad (2.19)$$

Eşitlik: \underline{A} ve \underline{B} kümeleri X uzayında tanımlı ve her iki kümenin de üyelik fonksiyonları eşiti ise \underline{A} ve \underline{B} eşittir.

$$\mu_A(u) = \mu_B(u) \quad (2.20)$$

2.2.2.2. Bulanık kümelerin özellikleri

A ve B, X uzayında tanımlı bulanık kümelerdir.

1. De Morgans Kuralı

$$(A \cup B)' = A' \cap B' \quad (2.21)$$

$$(A \cap B)' = A' \cup B'$$

2. Dağılma Özelliği

$$C \cap (A \cup B) = (C \cap A) \cup (C \cap B) \quad (2.22)$$

$$C \cup (A \cap B) = (C \cup A) \cap (C \cup B)$$

3. Aynı olma özelliği

$$\begin{aligned} A \cup A &= A \\ A \cap A &= A \end{aligned} \quad (2.23)$$

4. Özdeşlik özelliği

$$\begin{aligned} A \cup \emptyset &= A & A \cap X &= A \\ A \cap \emptyset &= \emptyset & A \cup X &= X \end{aligned} \quad (2.24)$$

5. Geçiş özelliği

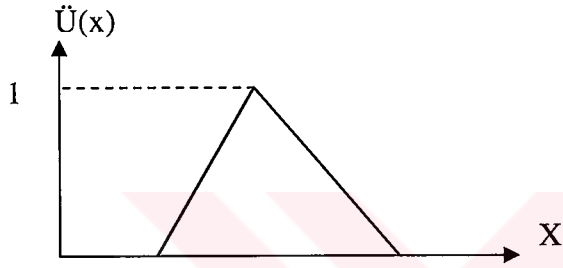
$$\text{Eğer } A \subseteq B \subseteq C \text{ ise } A \subseteq C \text{ 'dir.} \quad (2.25)$$

6.

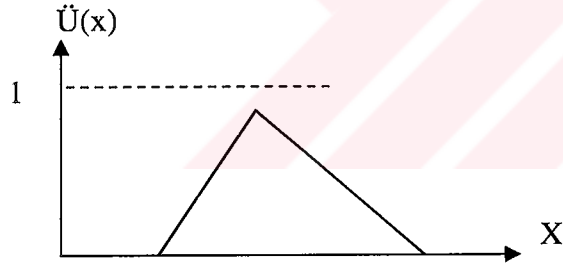
$$\begin{aligned} A \cup A' &\neq X \\ A \cap A' &\neq \emptyset \end{aligned} \quad (2.26)$$

2.4. Üyelik Fonksiyonlarının Özellikleri

Bulanık kümelerde genellikle üyelik derecesi 1'e eşit olan en az 1 eleman bulunmaktadır. Bu tür bulanık kümelere normal bulanık küme denir (Şekil 2.3a). Elamanların tümünün üyelik derecesi 1'den küçük ise bu tür bulanık kümelere normal olmayan bulanık küme denir (Şekil 2.3b). Eğer bulanık kümenin en yüksek üyelik derecesi 1'den küçük ise; yani normal olmayan bulanık kümeysen, normal hale dönüştürülür. Bunun için, her bir üyelik derecesi en büyük üyelik derecesine bölünerek normal bulanık kümeler haline dönüştürülür.



Şekil 2.3. a) Normal Bulanık Küme



Şekil 2.3. b) Normal Olmayan Bulanık Küme.

Üyelik fonksiyonlarının simetrik olma zorunluluğu yoktur.

2.5. Neden Bulanık Mantık ?

Bulanık mantık için matematiğin gerçek dünya uygulaması denilebilir. Çünkü gerçek dünyada her an değişen durumlarda değişik sonuçlar çıkabilir (Elmas 2003).

Sistemler için karar alma mekanizmaları geliştirebilmek için öncelikle sistem davranışının iyi analiz edilmesi gerekmektedir. Ancak bu şekilde sisteme ait

değişkenler doğru şekilde modellenebilir. Matematiksel modelleme ile sağlıklı tanımlamaları yapılamayacak değişkenler için uzman kişilerin de yardımıyla, günlük konuşma dilindeki bazı bulanık ifadeler ile modellemeler gerçekleştirilmeye çalışılır. Bu tür denetim ifadeleri tamamen insan sezgilerine bağlı olarak değişirler.

Bulanık mantık yaklaşımı, matematiksel modellemelerin yetersiz geldiği durumlarda insanın deneyimlerinden ve sezgilerinden yararlanarak çalışılabilmesine olanak tanıyan bir yaklaşımdır. İnsana ait deneyimler ya da düşünceler bilgisayar sistemlerine tanıtılırken ya da aktarılırken matematiksel bir temel söz konusudur. Bu da, Bulanık mantık kümeler kuramı ve buna dayanan bulanık mantık yaklaşımıdır.

Bu yaklaşım ilk Amerika Birleşik Devletleri'nde düzenlenen bir konferansta 1956 yılında duyurulmuştur. Ancak bu konuda ilk ciddi adım 1965 yılında Lotfi A.Zadeh tarafından yayınlanan bir makalede bulanık mantık veya bulanık küme kavramı adı altında ortaya konulmuştur. Zadeh bu makalesinde insan düşüncesinin büyük çoğunun bulanık olduğunu, kesin olmadığını belirtmiştir. İnsan mantığı, açık, kapalı, sıcak, soğuk, 0 ve 1 gibi değişkenlerden oluşan kesin ifadelerin yanında az açık, az kapalı, serin, ılık gibi ara değerleri de göz önüne almaktadır. Bulanık mantık klasik mantığın aksine iki seviyeli değil, çok seviyeli işlemleri kullanmaktadır. Klasik denetim uygulamalarında karşılaşılan zorluklar nedeniyle, bulanık mantık denetimi alternatif yöntem olarak çok hızlı gelişmiş ve modern denetim alanında geniş uygulama alanı bulmuştur (Öztürk at al 2003). Bulanık mantığın genel özellikleri Zadeh tarafından şu şekilde ifade edilmiştir:

- Bulanık mantıkta, kesin değerlere dayanan düşünme yerine, yaklaşık düşünme kullanılır.
- Bulanık mantıkta herşey $[0,1]$ aralığında belirli bir derece ile gösterilir.
- Bulanık mantıkta bilgi büyük, küçük, çok, az gibi dilsel ifadeler şeklindedir.
- Bulanık çıkarım işlemi dilsel ifadeler arasında tanımlanan kurallar ile yapılır.
- Her mantıksal sistem bulanık olarak ifade edilebilir.
- Bulanık mantık matematiksel modeli çok zor elde edilen sistemler için çok uygundur.

Bütün bunlardan sonra bulanık mantığın, geleneksel mantık kurallarından birçok yerde ayrıldığını görmekteyiz. Bu farklar Zadeh (1989) tarafından ayrıntılı olarak açıklanmıştır.

Doğruluk (truth): İki değerli mantıkta doğruluk sadece iki değer alabilir: Doğru veya Yanlış. Bulanık mantıkta ise bir önermenin doğruluk değeri $[0, 1]$ aralığındaki bir alt kümeden oluşur.

İfadeler (predicates): İki değerli mantıkta ifadeler kesindir. Ölü, çift, ..den büyük gibi. Bulanık mantıkta ise ifadeler bulanıktır. Uzun, hasta, yakın, ...den daha büyük gibi.

Belirti düzenleyiciler (predicate modifiers): Klasik sistemde, en çok kullanılanı, değil' dir. Bulanık mantıkta ise, birçok belirti düzenleyici vardır. Örneğin çok, çok veya az, oldukça, son derece gibi. Bunlar özellikle dilsel değişkenlere birlikte kullanımda büyük rol oynarlar. Örn: çok genç, çok genç değil gibi.

Niceleyiciler (quantifiers): Klasik sistemde niceleyiciler sadece iki tanedir: kesinlik ve ait olma. Bulanık mantıkta ise birçok niceleyici bulunmaktadır: birkaç, genellikle, çoğunlukla, hemen hemen her zaman, sıkça gibi.

Olasılık (probability): Klasik mantık sistemlerinde, olasılık sayısalıdır. Bulanık mantıkta ise dilsel bulanık ifadelerden faydalanılır. Örn: muhtemelen, çok muhtemel, yaklaşık 0.5 civarında, yüksek ihtimalle gibi.

Olabilirlik (possibility): Bulanık mantıkta olabilirlik kavramı ikili mantıktakine göre çok daha derecelendirilmiş durumdadır. Olasılıkta olduğu gibi, olabilirlik ifadeleri de dilsel değişken işlevi görebilirler. Örn: mümkün, çok mümkün, imkansız gibi.

2.6. Bulanık Sistemlerin Gelişimi

Bulanık küme teorisi 1965 yılında Zadeh tarafından ortaya atıldıktan sonra bir çok bilim adamı tarafından araştırma konusu olmuş ve birçok endüstriyel uygulamada

hayata geçirilmiştir. Mamadani, buhar türbininin hızının denetlenmesinde “Eğer türbin hızı çok hızlı artıyorsa ve basınç da çok düşük ise, buhar vanasını biraz aç” şeklindeki kurallar ile bir bulanık sistem geliştirmiştir.

Endüstriyel alandaki ilk önemli uygulama, çimento sanayiinde olmuştur. Çimento sanayiinde değirmen içerisindeki sıcaklık ve oksijen oranı kaliteli üretim açısından oldukça önemlidir. Bir firma, doğrusal bir model üzerine kurulu geleneksel denetleyici yerine bulanık mantık denetleyici kullanarak çok daha başarılı sonuçlar veren bir uzman sistem geliştirmiştir (Öztürk at al 2003).

Bu uygulamadan sonraki en önemli aşama, Japonya’da 1987 yılında faaliyete geçen Sendai metrosudur. Yasunobu ve Miyamoto, Japonya’daki bir otomatik tren sisteminin denetiminde bulanık mantık kullanmışlardır. Sabit hız ve otomatik duruş denetiminin esas olarak sağlandığı bu sistem 1987 yılından beri kullanılmaktadır (Berenji 1992).

Bulanık mantık teorisinin endüstriyel uygulama alanları arasında beyaz eşyalar, asansör, tren, otomotiv sanayi sayılabilir. Beyaz eşya uygulamalarından olan çamaşır makinesindeki uygulama nedenini inçlersek; bu şekilde üretilen çamaşır makineleri, çamaşırın cinsi ve kirliliği, suyun durumu gibi şartları özel sensörlerle tesbit ettikten sonra deterjanı, devir sayısı ve süresini ayarlamakta ve bu şekilde mevcut şartlarda en optimum şekilde çalışmaktadır. Bunlara örnek olarak, Matsushita ve Hitachi'nin çamaşır makinası, Sanyo ve Matsushita'nın Video kamerası, Toshiba, Sharp, Sanyo ve Hitachi'nin mikrodalga fırınları, Canon'un fotoğraf makinası sayılabilir (Berenji 1992).

Bu sistemlerin yanında, şeker hastaları için vücuttaki insülin miktarını ayarlayarak yapay bir pankreas görevi yapan yapıların üretiminde, prematüre doğumlarda bebeğin ihtiyaç duyduğu ortamı devam ettiren sistemlerin hazırlanmasında, suların klorlanmasında, kalp pillerinin üretiminde, oda içerisindeki ışık miktarının ayarlanmasında ve bilgisayar sistemlerinin soğutulmasında bulanık mantık çok şeyler vaatmektedir.

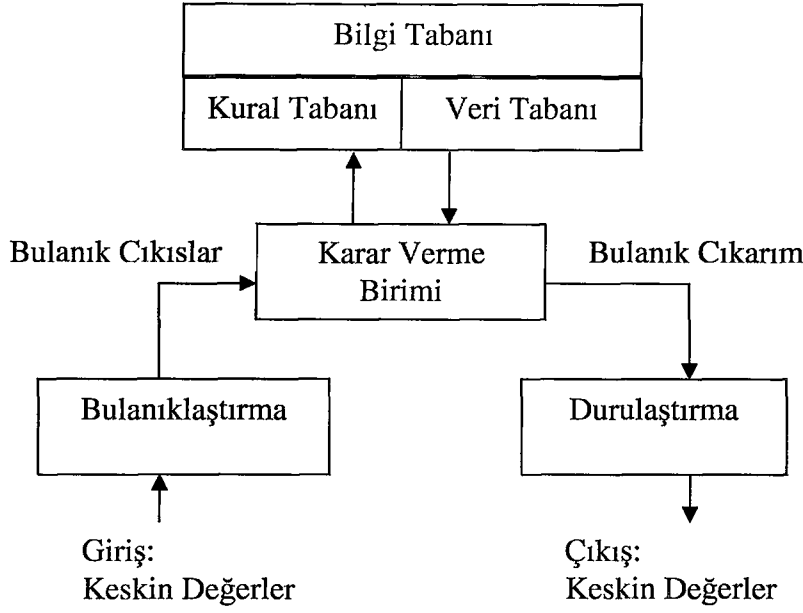
2.7. Bulanık Denetim

1973 yılında Zadeh, bulanık algoritmalar üzerine yaptığı bir çalışmada denetim algoritmasının mantıksal kurallar yardımıyla kurulabileceği fikrini ortaya atmıştır. O ana kadar, mantığın denetim algoritmalarına uygulanmasıyla ilgili öne sürülen fikirler mevcut olmasına karşın, Zadeh' in öne sürdüğü ile diğer yaklaşımlar arasındaki temel fark, karmaşık ve tam bir matematiksel modeli kurulamayan sistemlerin davranışlarını yaklaşık ve yeterince efektif olarak tanımlayabilmek, insan gibi düşünmeye ve insan gibi davranmaya dayalı denetim algoritmaları ortaya çıkarmaktır (Zadeh 1973).

Bulanık denetim sistemleri, bulanık mantık metotlarıyla tanımlanan kural-tabanlı sistemlerdir. Buradaki kural-tabanı, insanın tecrübe ve sezgilerine, denetlenen nesnenin (veya sistemin) pratik ve teorik davranışının anlaşılmasına dayalı olarak oluşturulur. Bulanık denetimi özel ve farklı kılan, bir analitik tanıma ihtiyaç duyulmamasıdır. Buradaki kurallar EĞER...O HALDE (IF ... THEN) şeklindeki önermelerden oluşmaktadır. EĞER A O HALDE B şeklindeki bulanık koşullu bir ifadede A ve B bulanık anlamlar taşırlar. Örneğin:

EĞER (Hava soğuk ise) O HALDE (Suyun ısını yüksek seviyeye getir). Burada Hava ve Su bir dilsel değişken, soğuk ve yüksek ise bu dilsel değişkenlerin dilsel değerleridir. Tezde, önermelerin “EĞER” ve “O HALDE” arasında kalan kısmı “birincil kısım”, O HALDE” den sonraki kısım ise “ikincil kısım” olarak isimlendirilecektir.

Şekil 2.4, bir bulanık denetim sistemindeki temel yapıyı açık olarak göstermektedir.

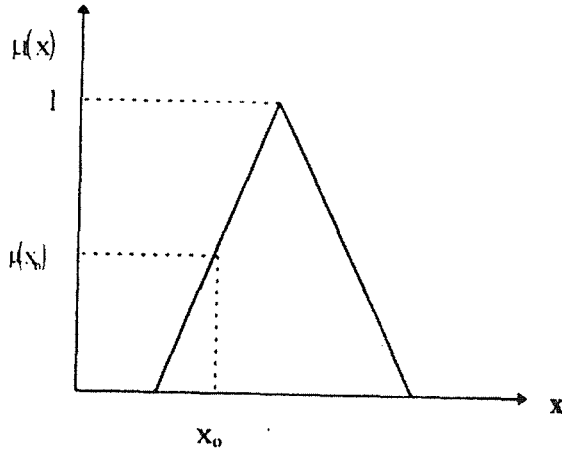


Şekil 2.4. Bir Bulanık Denetim Sisteminin Blok Diagramı.

Bulanık Denetim genel olarak, gelen keskin (crisp) değerlerin dilsel ya da bulanık üyelik fonksiyonlarına dönüştürüldüğü Bulanıklaştırma (Fuzzification) bölümü, buradan elde edilen bulanık değerlerin uygulandığı Bilgi-Tabanı (Rule-base) bölümü ve elde edilen bulanık çıktıların tekrar keskin değerlere dönüştürüldüğü Durulaştırma (Defuzzification) bölümlerinden oluşmaktadır.

2.7.1. Bulanıklaştırma

Bulanıklaştırma, giriş değişkenleri üzerine tanımlı üyelik fonksiyonlarının gerçek değerlere uygulanarak, her birine karşı bir bulanık değer elde edilmesidir. Şekil 2.5’de x değişkeninin, x_0 değerine karşı gelen üyelik değeri $\mu(x_0)$ olarak gösterilmiştir.



Şekil 2.5. x_0 Değişkeninin Üyelik Gösterimi.

Kullanılacak üyelik fonksiyonlarının tipi belirlendikten sonra, her dilsel değişkene karşı düşecek dilsel değerleri belirlemek gerekir. Örneğin {Küçük, Orta, Büyük} tekil terimlerinden oluşan dilsel değerler, kimi zaman değişkenin alabileceği değerler açısından yetersiz kalabilir. Bu durumda, { Çok küçük, Çok Büyük} gibi bileşik terimler eklemek gerekebilir.

Üyelik fonksiyonlarının oluşturulması, sonuca olan etkileri açısından önemli bir konudur. Aşağıdaki bölümde bununla ilgili kriterler incelenmiştir.

2.7.2. Üyelik fonksiyonlarının oluşturulması

x, y, z dilsel değişkenlerinin alabileceği dilsel değerleri gösteren kümeler DX, DY, DZ olsun. $DX=DY=DZ=(NB, NO, NK, Z, PK, PO, PB)$ olduğu varsayılırsa, herbir değişkenin yedi tane değeri olduğundan, yirmibir tane üyelik fonksiyonunun tanımlanması gerekir. Hesaplamadaki verimlilik, hafızanın verimli kullanımı ve performans gerekliliği açısından birbirine benzeyen üyelik fonksiyonlarının kullanımı gereklidir. Bu benzerlik, şekillerinin benzerliği, parametrik ve fonksiyonel benzerlik olarak algılanabilir.

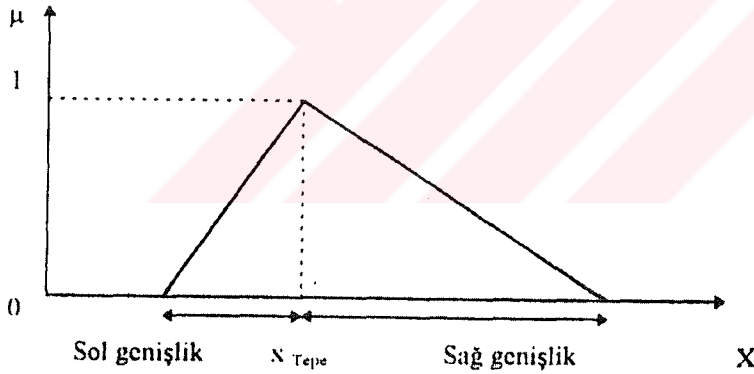
Üyelik fonksiyonlarından, parametrik ve fonksiyonel tanımı açısından en çok kullanılanı üçgen şeklinde olanıdır. Üyelik fonksiyonlarının tipi belirlendikten sonra, ilgili değişkene ait alana terimler yerleştirilmelidir.

Bir deęişkene ait dilsel deęerlerin yerleřtirilmesi bulanık denetim sistem, performansını birok bakımdan etkileyebilir. Bunlarda detaya inmeden nce yelik fonksiyonunu karakterize eden bazı parametreleri aıklamak yerinde olacaktır:

2.7.2.1. Tepe deęeri

x bir dilsel deęişken ve bu deęişkenin DX dilsel deęerine ait yelik fonksiyonu μ_{DX} ile vermek zere, $\mu_{DX} : X \rightarrow [0,1]$ olsun. μ_{DX} 'in tepe deęeri yani $\mu_{DX}(X_{tepe})=1$ dir (Őekil 2.7). Trapezoidal Őekilli yelik fonksiyonlarında tepe deęeri belli bir aralıktadır.

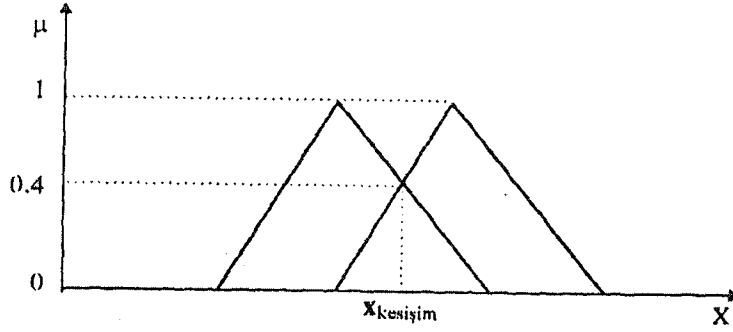
Sol ve Saę Geniřlikler: μ_{DX} 'in saę ve sol geniřlikleri Őekil 2.6' da gsterilmiřtir. Saę ve sol geniřliklerin birbirine eřit olduęu durumda yelik fonksiyonu simetrik, dięer durumda asimetrik olarak isimlendirilir.



Őekil 2.6. yelik Fonksiyonlarının Saę-Sol Geniřlikleri ve Tepe Deęeri,

2.7.2.2. Kesiřim noktaları

μ_{DX1} ve μ_{DX2} iki yelik fonksiyonu olmak zere, Őekil 2.7'de gsterildięi gibi kesiřim durumunda iseler bu kesiřim $\mu_{DX1}(x_k)=\mu_{DX2}(x_k)>0$ olarak belirtilir.

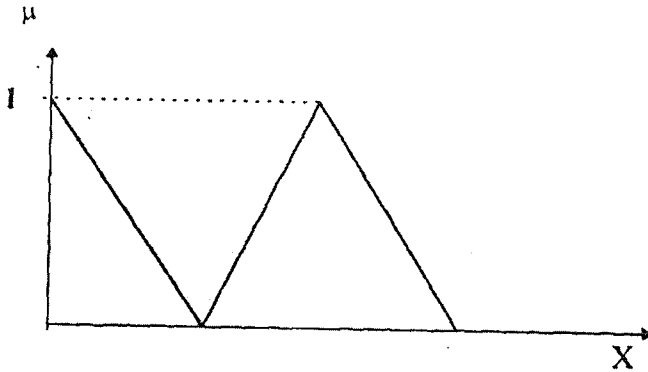


Şekil 2.7. Üyelik Fonksiyonlarında Kesişim Noktası.

İki farklı dilsel değeri tanımlayan iki üyelik fonksiyonunun birden fazla kesişim noktası olabilir. İki üyelik fonksiyonu arasındaki kesişim noktalarının oranına, kesişim-noktası oranı denir.

2.7.2.3. Kesişim noktası seviyesinin etkisi

İlk olarak, DE ile gösterilen terimler kümesinin yerleştirilmesi sırasında, her iki üyelik fonksiyonunun sıfırdan büyük bir kesişim noktası bulunmalıdır. Bu demektir ki, her keskin (crisp) değer, sıfırdan büyük bir üyelik derecesine sahip en azından bir üyelik fonksiyonuna ait olur. Bu koşul sağlanmadığı durumda bazı girişler hiçbir kuralı harekete geçirmeyecek ve çıkış için hiçbir değer ortaya çıkmayacaktır. Daha ötesi, iki üyelik fonksiyonu arasındaki kesişim noktası sıfır ise belli bir an için sadece bir -kural harekete geçecektir. Bu durum Şekil 2.8' de gösterilmiştir.

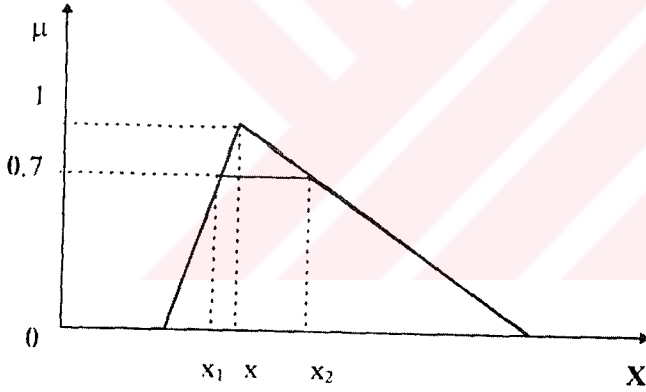


Şekil 2.8. Üyelik Fonksiyonları Arasındaki Kesişim Noktası Oranı Sıfır.

İkinci olarak, üçüncü dereceye kadar lineer sistemler için simetrik üyelik fonksiyonlarında kesişim noktası ve oranı için bazı “optimal” değerler bulunmaktadır. İki-komşu-üyelik fonksiyonunun kesişim noktası seviyesi 0.5 ve kesişim noktası oranı 1 ise bu, sonucun daha yavaş yükselmesini, zamandaki artışın daha hızlı olmasını ve daha yavaş bir inişi sağlar. Kesişim noktası ve oranı ile ilgili bu değerlerin kullanımı bir kurala bağlanmamış olsa da literatürde karşılaşıldığı üzere genelde yapılan bir seçimdir.

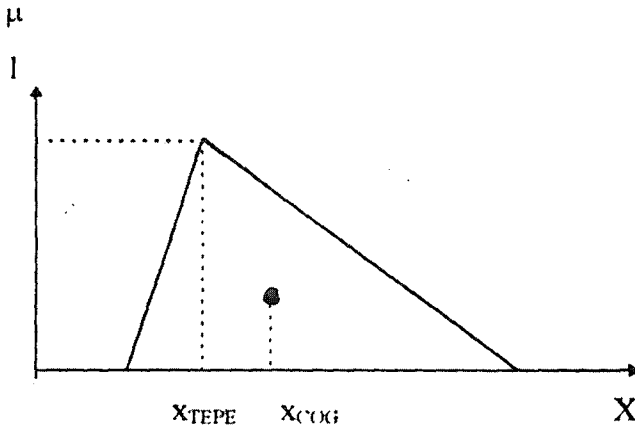
2.7.2.4. Simetrinin etkisi

Şekil 2.9’ da, 0.7 lik bir üyelik değerine karşılık X ekseninde iki değer bulunmaktadır. Buradan elde edilecek x değerinin bir sonraki işlemin girişi olduğu düşünüldüğünde alınacak olan x değeri $x = \frac{x_1 + x_2}{2}$ şeklinde hesaplanmalıdır.



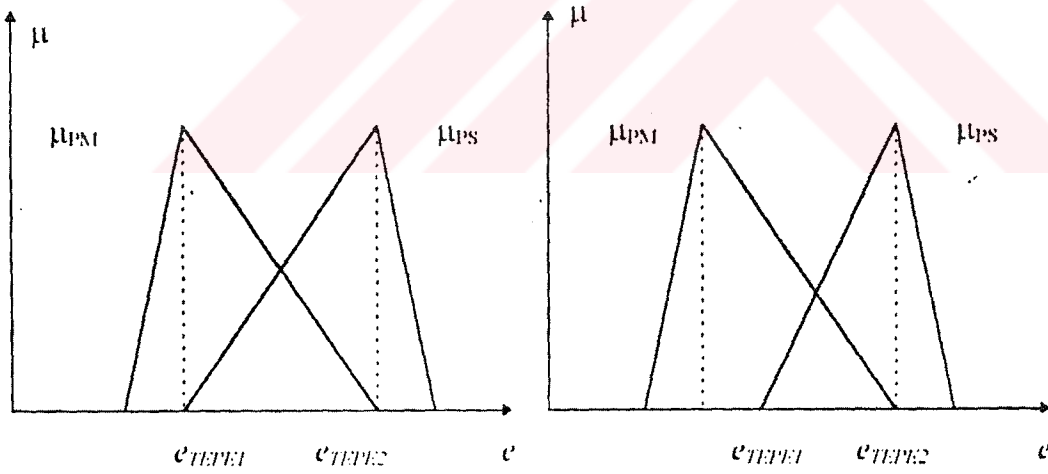
Şekil 2.9. 0.7’ den “kırpılmış” bir Üyelik Fonksiyonu

Şekil 2.10’ da, asimetrik bir üyelik fonksiyonunun, Center of Gravity (Ağırlık merkezi) metodu ile hesaplanan x_{COG} değeri ile X_{TEPE} değerinin eşit olmadığı görülmektedir.



Şekil 2.10. Asimetrik bir Üyelik Fonksiyonu

Simetrinin sağlanması durumunda $X_{TEPE}=X_{COG}$ olacaktır, 1 üyeliğini veren x değerinin, ağırlık merkezine karşılık düşen x değeri ile aynı olması her zaman tavsiye edilen bir sonuç olduğundan, üyelik fonksiyonlarında simetri, aranan bir özelliktir. Üyelik fonksiyonlarının genişliğinin yaratacağı etkiyi açıklayan şekiller, Şekil 2.11a ve b' de verilmiştir.



Şekil 2.11 a) e Değişkeninin PM ve PS Üyelik Fonksiyonları

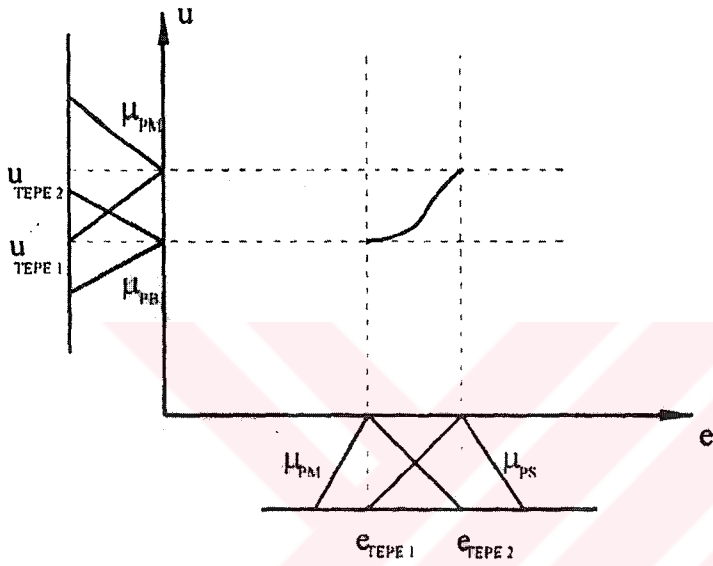
Şekil 2.11 b) Aynı Üyelik Fonksiyonlarının Diğer bir Gösterimi

“EĞER c' PM ise O HALDE // PB dir”

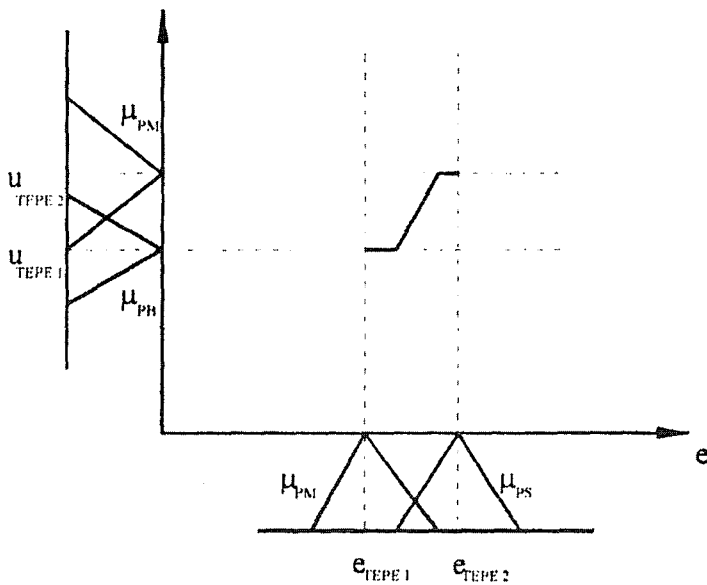
“EĞER c' PS ise O HALDE // PM dir”

şeklinde iki kuralın olduğu düşünülün.

PM ve PS, f değişkenine ait, PB ve PM ise u değişkenine ait değerler olsun. Şekil 2.11a ve 2.11b' de iki μ_{PM} ve μ_{PS} verilmiştir, a'da aşağıdaki koşul sağlanmaktadır: μ_{PS} 'in sol genişliği, μ_{PM} 'in sağ genişliğine eşit ve ikisi de iki komşu üyelik fonksiyonunun tepe değerleri arasındaki mesafeye eşittir.



Şekil 2.12. Koşulun Sağlanması Durumu



Şekil 2.13. Koşulun Sağlanmaması Durumu

Şekil 2.11b' de bu koşul sağlanmamaktadır. Koşulun sağlanması durumunda e , e_{TEPE1} den e_{TEPE2} ye doğru yumuşak bir şekilde değişecek ve kuralın ikincil tarafının da uygulanmasından sonra elde edilecek u değişkeninin de, u_{TEPE1} den u_{TEPE2} ye olan değişimi yumuşak olacaktır. Şekil 2.12 ve Şekil 2.13' de bu durumlara karşı düşen çıkış değişimleri gösterilmiştir.

Yukarıda açıklandığı üzere, üyelik fonksiyonlarının şekli, simetrisi, birbirleriyle olan ilişkisi, çıkışa doğrudan etkileyen unsurlardır. Bu nedenle, çok iyi seçilmeleri en az kural tabanı kadar önem taşımaktadır.

2.7.3. Bilgi tabanı

Bilgi tabanı, karar verme biriminin bilgileri aldığı veri tabanı ve denetim amaçlarına uygun dilsel denetim kurallarının bulunduğu kural tabanından (rule base) oluşmaktadır. Bulanık mantık uygulamalarında denetim araçlarından ve değerlendirilecek bilgilerden oluşur.

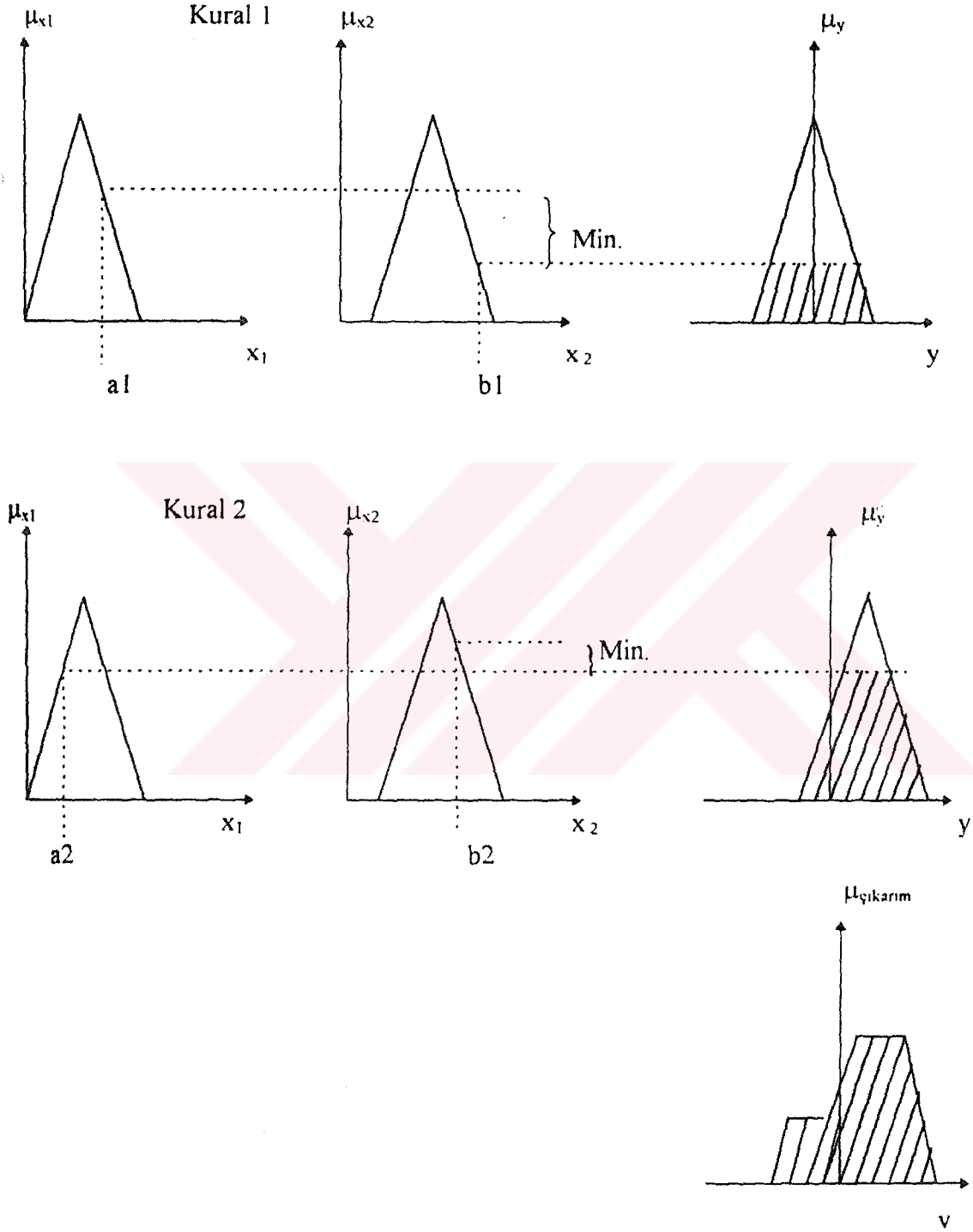
2.7.4. Bulanık kurallar ve çıkartım

Kuralların oluşturulmasında en çok kullanılan yöntem, uzman görüşü ve bilgisine başvurmaktır. Uzman kişinin sistemin denetiminde kullandığı bilgi ve tecrübeler tam olarak kurallara aktarılabildiğinde bu metod en etkin metoddur.

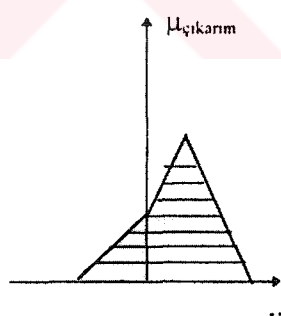
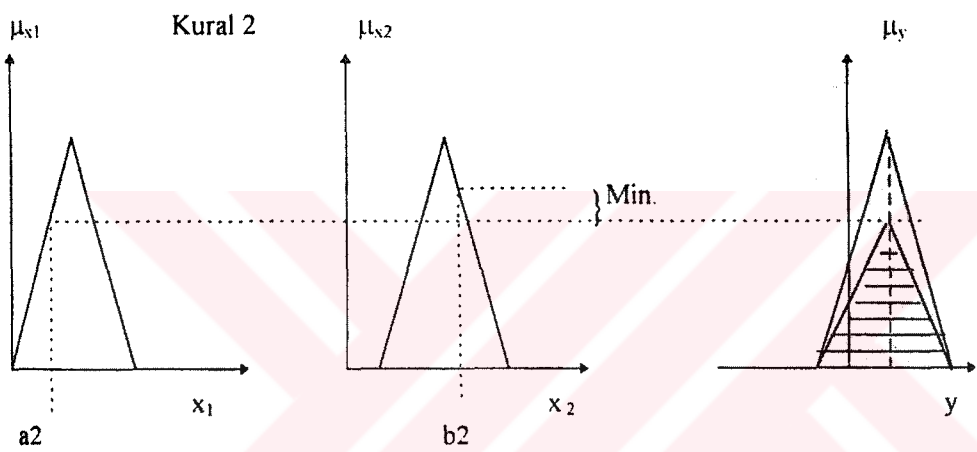
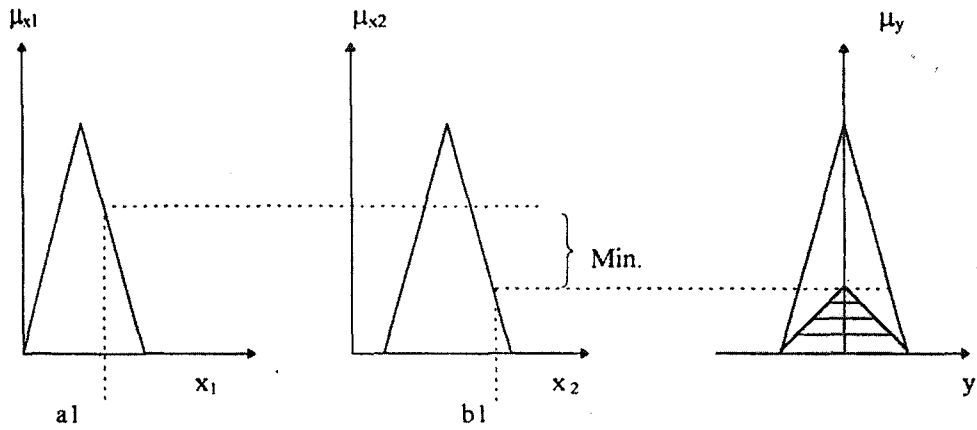
Giriş değerlerinin bulanıklaştırılması ve kuralların uygulanmasından sonra yapılan, çıkarımda bulunmaktır. Klasik mantıkta çıkarım (inference), verilen önermelerden bir sonuca varmaktır. Burada da, tüm girişlerin tüm kurallara uygulanmasından sonra varılan sonuç olarak tanımlanabilir. Aşağıda çıkarımla ilgili iki yöntem şekillerle açıklanmıştır. Bunlar:

- Max-min çıkarım yöntemi : Mamdani çıkarımı olarak da anılan bu yöntemde, çıkıştaki bulanık küme girişteki kümelerin VE (MIN) işlemine tabi tutulması sonucunda elde edilir. Yöntem, Şekil 2.14' de gösterilmiştir (Vadiee 1993).

- Max-product çıkarım yöntemi : Bu yöntemle ilgili grafiksel gösterim Şekil 2.15' de verilmiştir (Vadiee 1993).



Şekil 2.14. Max-Min Çıkarım Yöntemi



Şekil 2.15. Max-Product Çıkarım Yöntemi

Şimdiye kadar anlatılanların ışığında, olayları açıklamak bakımından bir örnek verilecektir.

Bir seradaki bitkilerin günlük sulama miktarı, bulanık mantık yardımıyla belirlenecektir. Burada hava sıcaklığı ve toprağın nemi ölçülerek su miktarı tespiti yapılacaktır.

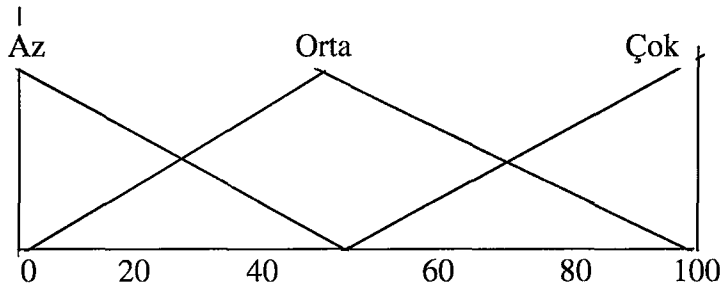
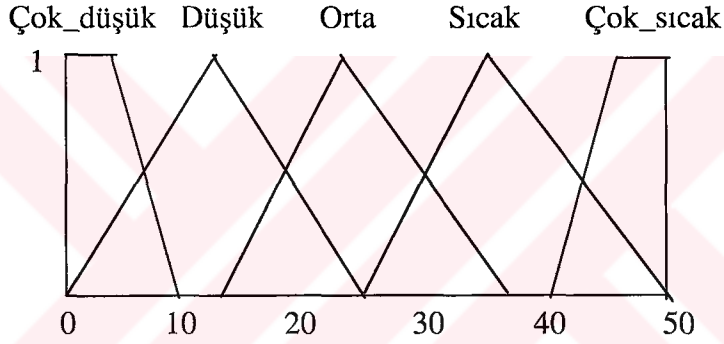
Sistemde giriş değişkenleri Sıcaklık ve Nem, çıkış değişkeni ise Su miktarıdır. Bu değişkenlerin alabileceği dilsel değerler,

$D(\text{Sıcaklık}) = \{ \text{Çok düşük, Düşük, Orta, Sıcak, Çok sıcak} \}$

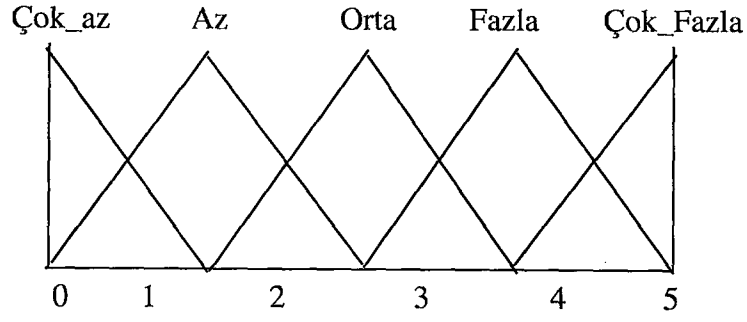
$D(\text{Nem}) = \{ \text{Az, Orta, Çok} \}$

$D(\text{Su miktarı}) = \{ \text{Çok az, Az, Orta, Fazla, Çok Fazla} \}$

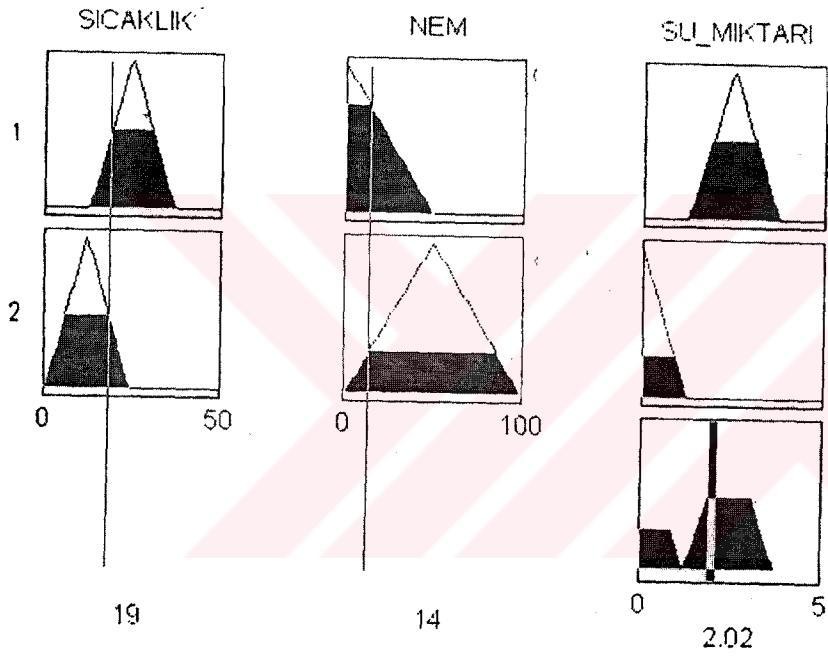
Şekil 2.16a, b, c'de, Sıcaklık, Nem ve Su miktarı değişkenlerinin üyelik fonksiyonları, Şekil 2.17' de ise, iki kural olması durumu için örnek bir çıkarım şekli gösterilmiştir.



Şekil 2.16. b) Nem Giriş Değişkenine İlişkin Üyelik Fonksiyonları



Şekil 2.16. c) Su Miktarı Çıkış Değişkenine İlişkin Üyelik Fonksiyonları



Şekil 2.17. Üyelik Fonksiyonları ve İki Kural Sonucunda Yapılan Çıkarım

Çıkarımın gösterildiği şekil, aşağıdaki gibi iki kuralın olması durumu içindir:

Sıcaklık Orta ve Nem Az ise Su miktarı Orta

Sıcaklık Düşük ve Nem Orta ise Su miktarı Çok Az

2.7.5. Durulaştırma

Bulanıklaştırılan giriş değerleri tüm kurallara tabi tutulduktan sonra, her giriş için bulanık bir çıkarım değeri oluşur. Bu bulanık değerlerin tekrar giriş değerleri gibi keskin değerler haline dönüştürülmeleri olayına durulaştırma denir. Aşağıda, en çok kullanılan durulaştırma yöntemleri şekillerle açıklanmıştır (Driankov, Hellendoorn 1996):

- Ağırlık merkezi ((Center-of-gravity/area) yöntemi,
- En büyük alanın merkezi (Center-of-largest-area) yöntemi,
- İlk En Büyük (First-of-maxima) yöntemi,
- En büyük orta (Middle-of-maxima) yöntemi,
- Yükseklik (Height) yöntemi.

2.7.5.1. Ağırlık merkezi yöntemi

Bu yöntem literatürde en çok karşılaşılan ve en iyi bilinen bir yöntemdir. Şekil 2.18, bu işlemin grafiksel olarak ne şekilde yapıldığını göstermektedir. Durulaştırılan çıkış u^* ile gösterilmek üzere,

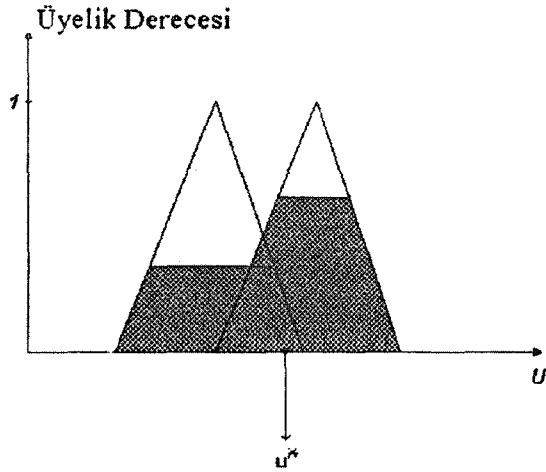
Ayrık zamanda ($U=[u_1.....u_2]$) :

$$u = \frac{\sum u_i \mu_u(u_i)}{\sum \mu_u(u_i)} = \frac{\sum_{i=1}^1 u_i \cdot \max(\mu_{KDU^{oko}}(u_i))}{\sum_{i=1}^1 \max(\mu_{KDU^{oko}}(u_i))} \quad (2.26)$$

şeklinde ifade edilir. Sürekli zamanda ise:

$$n = \frac{\int_u u \cdot \mu_u(u) du}{\int_u \mu_u(u) du} = \frac{\int_u u \cdot \max(\mu_{KDU^{oko}}(u)) du}{\int_u \max(\mu_{KDU^{oko}}(u)) du} \quad (2.27)$$

olarak ifade edilebilir.



Şekil 2.18. Ağırlık Merkezi Durulaştırma Yöntemi

2.7.5.2. Toplamların merkezi yöntemi

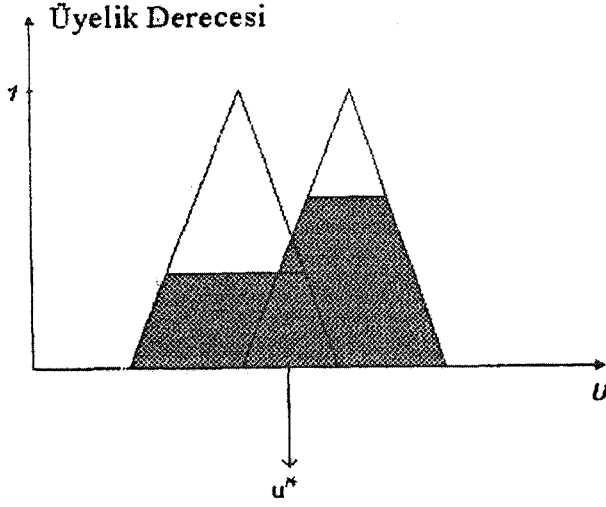
Basit fakat daha hızlı bir yöntemdir. Hızlı olması, en çok kullanılan yöntemlerden biri olmasının başlıca nedenidir.

Ayrık zamanda :

$$n = \frac{\sum_{i=1}^1 u_i \sum_{k=1}^n \mu_{KDO^{oko}}(u_i)}{\sum_{i=1}^1 \sum_{k=1}^n \mu_{KDO^{oko}}(u_i)} \text{ şeklinde,} \quad (2.28)$$

Sürekli zamanda ise :

$$n = \frac{\int_u \sum_{k=1}^n \mu_{KDU^{oko}}(u) du}{\int_u \sum_{k=1}^n \mu_{KDU^{oko}}(u) du} \text{ şeklinde ifade edilir.} \quad (2.29)$$



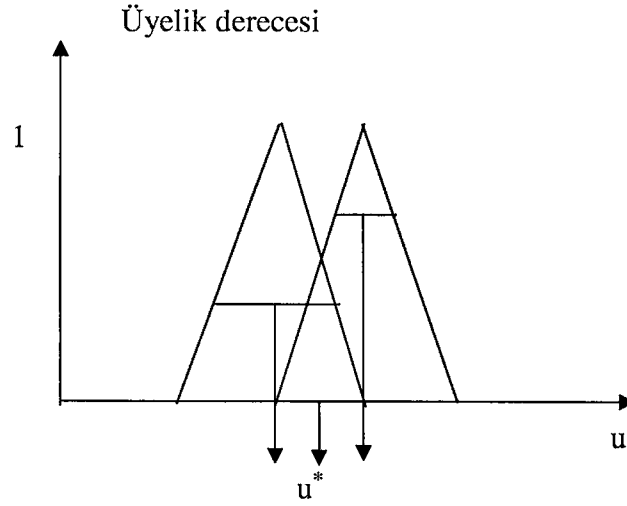
Şekil 2.19. Toplamların Merkezi Yöntemi

2.7.5.3. Yükseklik yöntemi

Bu yöntem kırılmış çıkışların birleşimini kullanmak yerine, her denetim çıkışının kırıldığı seviyeyi kullanır.

$c^{(k)}$ DU denetim çıkışına ait tepe değeri ve f_k , $KDU^{(k)}$ kırılmış çıkışının yüksekliği olmak üzere, m kurallı bir sistemin Yükseklik yöntemi ifadesi.

$$u = \frac{\sum_{k=1}^m e^{(k)} \cdot f_k}{\sum_{k=1}^m f_k} \text{ şeklinde verilir.} \quad (2.30)$$

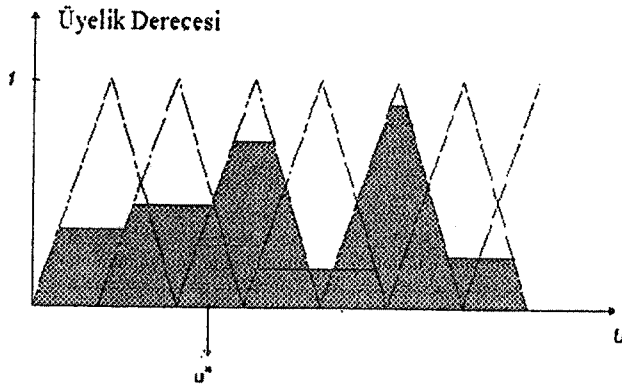


Şekil 2.20. Yükseklik Yöntemi

2.7.5.4. En büyük alanın merkezi yöntemi

Bu yöntem, U' nun konvex olmadığı durumlarda kullanılabilir. Yöntem en büyük alanlı konvex bulanık alt kümeyi hesaplar.

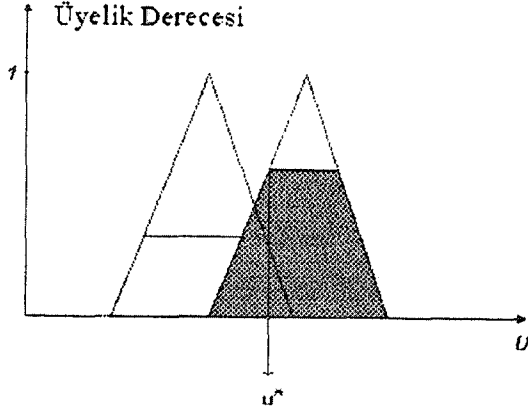
Burada u^* ilgili bulanık alt küme alanının merkezindedir. Bu yöntem diğerlerine göre biraz daha karmaşıktır çünkü, ilk olarak konvex bulanık alt kümelerinin bulunması gerekmektedir. Yöntemle ilgili grafik gösterim Şekil 2.21' de verilmiştir.



Şekil 2.21. En Büyük Alanın Merkezi Yöntemi

2.7.5.5. İlk en büyük yöntemi

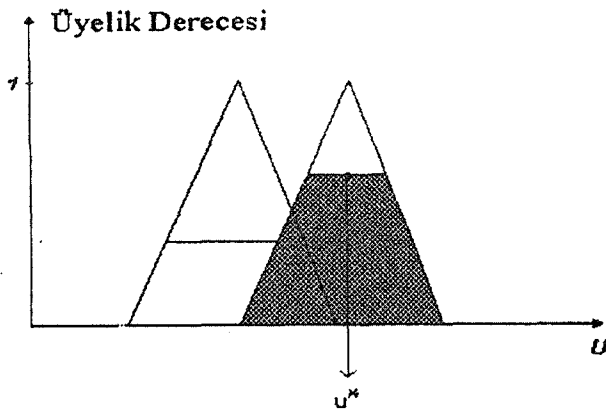
Bu yöntem U denetim çıkışını kullanır ve maksimum üyelik derecesine sahip (T nin en küçük değerini alır. Yöntemle ilgili grafik gösterim Şekil 2.22' de verilmiştir.



Şekil 2.22. İlk En Büyük Yöntemi

2.7.5.6. En büyük orta yöntemi

İlk en büyük yöntemine benzer şekilde, maksimum üyelik derecesini veren aralığın orta noktası u değerini verir. Yöntemle ilgili grafik gösterim Şekil 2.23' de verilmiştir.



Şekil 2.23. En Büyük Orta Yöntemi

3. ESNEK SORGULAMA ARAÇLARI

3.1. Giriş

Veri tabanı sistemleri alanında, bugüne kadar pek çok uygulama geliştirilmiştir. Bir veri tabanı sistemi, büyük boyutlu veri kümelerinin yönetilebilmesi için kullanılan araç şeklinde tanımlanmaktadır. Veri tabanı uygulamalarında amaç, verinin güvenilir bir ortamda saklanıp; daha sonra bu verilere hızlı ve doğru erişimi gerçekleştirmek olduğu için; veri tabanı yönetim sistemlerindeki gelişmeler de bu amaç doğrultusunda ilerlemiştir. Veri tabanlarında klasik sorgulama yapabilmek için en yaygın kullanılan araç, SQL sorgulama dilidir. SQL dili ile sorgulama yapabilmek için, bu dil hakkında bilgi sahibi olmak, kurallarını bilmek ve matematiksel ifadelerden yararlanmak gerekmektedir. Sistemde keskin sınırlar ile tam olarak ifade edilemeyen değişkenler olduğunda klasik sorgulama yöntemi yetersiz kalmaktadır. Bunun nedeni klasik mantığın 2 değerli olmasıdır. Bir değişken aynı anda yalnızca bir kümeyle ait olabilmektedir. Günlük yaşantımızda ise veriler sınıflandırılırken çoğu kez sezgiler doğrultusunda hareket edilmektedir. Yani kesin ifadelerle, veriler tam düşünüldüğü gibi ifade edilemeyebilir. Bu tür, matematiksel modeli çıkartılmayan sistemler modellenirken, bulanık mantık teorisinden yararlanılmaktadır.

Bu gereksinimler doğrultusunda, gerçek dünyadaki belirsiz veriler üzerinde de işlem yapabilecek ya da veriler üzerinde daha esnek sorgulama yapabilecek sorgu araçları geliştirilmeye başlanmış ve bu durum bir çok araştırmaya konu olmuştur.

Bu bölümde bugüne kadar geliştirilmiş olan bulanık mantık temelli esnek sorgu araçlarından bazıları incelenmiştir.

3.2. Bulanık Küme Teorisi ile Geliştirilen Esnek Sorgu Araçları

Bulanık mantık küme teorisi temel alınarak bir çok esnek sorgulama sistemi geliştirilmiştir. İlk olarak bulanık mantık küme teorisi ile esnek sorgulama sistemi Tahani(1970) tarafından geliştirilmiştir. Tahani tarafından geliştirilen sistemde, veri tabanındaki kayıtların bulanık kümelere olan üyelik dereceleri üzerinden bağıntılar tanımlanmaktadır.

$X = \{ X_1, X_2, \dots, X_n \}$ kümesi üzerinde tanımlı bir ilişki:

$\{x / \mu | x \in X, \mu \text{ x'in bağıntıya ait olma derecesidir}\}$ şeklinde tanımlanmaktadır. Bir ilişkinin bulanık olabilmesi için $\mu = 1$ olması gerekmektedir. Bulanıklaştırma işlemi dilsel değişkenler yardımı ile sağlanmaktadır. “genç, yaklaşık 100” gibi bulanık ifadeleri örnek olarak verebiliriz. Oluşturulan bulanık sorgu cümleleri “AND” ve “OR” bağlaçları ile birleştirilebildiği gibi; mantıksal değer üreten bir sorgu cümlesi ile bulanık sorgu cümlesi aynı bağlaçlar ile birleştirilerek sonuç alınabilmektedir.

1980 yıllarında Kacprzyk ve Ziolkowski tarafından fquery1 adında bir bulanık sorgu aracı geliştirildi. Bu çalışmada bir bulanık sorgu “Satışı çok iyi ve yaşı çok genç olmayan çalışanların hemen hemen hepsini bul” şeklinde oluşturulabilmektedir. Basit sorgular için eşleşme derecesinin hesaplanması eğer sorgu bulanık bir operatör içermiyorsa, yalnızca mantıksal operatörler içeriyorsa;

$$\text{Eşleşme Derecesi} = \begin{cases} 1, \text{ sorgu doğru ise} \\ 0, \text{ sorgu yanlış ise} \end{cases} \quad (3.1)$$

Eğer sorgu cümlesi basit ve bulanık bir sorgu cümlesi ise eşleşme derecesi aşağıdaki şekilde hesaplanır.

$$ED = \mu_{\text{Bulanikoperator}} (\text{BulanıkKume}) \quad (3.2)$$

Bileşik sorgu cümlelerinde eşleşme derecesi, cümleyi oluşturan basit sorgu cümlelerinin eşleşme dereceleri ile hesaplanır.

1980' li yılların sonlarında Patrick Bosc, bulanık olmayan değerlerden oluşan veritabanları için esnek sorgulama sistemi geliştirilmiştir. Bu uygulamada standart sorgulama dili SQL' e uzantı eklenerek SQLf diye isimlendirilen bir bulanık sorgulama dili geliştirilmiştir. SQLf , mümkün olduğunca SQL sorgulama diline benzer oluşturulmaya çalışılmıştır. Klasik veritabanlarındaki SQL sorgularında, istenen sonucu almak için veri tabanındaki kayıtlarda mantıksal işlemler gerçekleştirilir. Örneğin “Ortalama ücreti 3500'den çok olan mühendislerin çalıştıkları bölümleri bul” şeklindeki bir sorgu için:

```
Select #bolum from Calisan  
Where meslek='Mühendis'  
Group By #bolum Having avg(ucet) ≥ 3500
```

şeklinde bir standart SQL cümlesi çalıştırılır.

SQLf 'de ise, aynı sorgu cümlesi şu şekilde oluşturulur. “Ücreti yaklaşık 3500 olan mühendislerin çalıştıkları bölümleri bul” Bu sorgu için:

```
Select #bolum from Calisan  
Where meslek='Mühendis'  
Group By #bolum Having avg(ucet) ≈ 3500
```

şeklinde bir standart SQL cümlesi çalıştırılır.

Yukarıdaki sorgu cümlelerinde görüldüğü üzere, geliştirilen SQLf esnek sorgulama sistemi, bulanık şartları “Having” kısmında içermektedir.

Bulanık sorguya dilsel niteleyiciler eklendiği zaman SQLf yapısı değişikliğe uğramaktadır. Örnek verecek olursak; “genç mühendislerin çoğunun iyi ücret aldığı bölümleri bul” şeklindeki bir sorgu için SQLf cümlesi:

```
Select #bolum from Calisan
```

Group By #bolum

Having çoğu (yaşı='genç') and (ücret='iyi')

şeklinde yapılandırılmaktadır. Dilsel niteleyiciler ile nitelenmiş olan önerme "Having" kısmından sonra yazılmaktadır.

Bosc, Bazı metotların neden bulanık küme yaklaşımı ile sorgulama yapmadıklarının nedeninin, bulanık sorgularda tanımlamaların zorlukları olduğunu öne sürmüştür. SQLf sorgulama dilinin kapasitelerini destekleyen bir sistem, bazı özel stratejilere ihtiyaç duymaktaydı (Bosc 1997). Bunların yanında sıradan veri tabanı sistemlerine bulanık sorgu uygulamak daha fazla kayıtları geri dönmek demektir. Bu da sistemin beraberinde getirdiği bir diğer zorluktu.

Bulanık küme yaklaşımı esas alınarak geliştirilen bir diğer sorgulama aracı da 1997 yılında Dan Rasmussen ve Ronald R.Yager tarafından tasarlanmış ve uygulanmıştır (Rasmussen 1997). Hedefleri, dilsel özetlemelerin bulanık sorgunun bir parçası olmasıydı. SummarySQL' de dilsel özetler, bulanık sorgunun doğruluk ölçüsünü (τ), bulmak için kullanıldıkları gibi yüklem gibi de kullanılabilirler. Dilsel özetler, standart SQL'deki toplama fonksiyonları ile (AVG(), COUNT() gibi) ilişkilidir. Fakat bu fonksiyonlardan farklıdır. Bu farklılık dilsel özetler, birim aralıkta daima doğru değerler almasındandır. SummarySQL' in standart şekli aşağıdaki gibidir.

Select NitelikListesi

From TabloListesi

Where Koşullar

Bu formatta oluşturulan bulanık sorgu sonuçları, bulanık olmayan değerlerin saklandığı veri tabanındaki kayıtlar ve bunların doğruluk derecelerinin yer aldığı bir tablo ile verilmektedir. Sorgunun genel yapısı tanımlanırken belirtilen TabloListesi kısmı, bir alt sorgu içerebilmektedir. Koşullar kısmında yazılan bulanık koşul formu : "Nitelik IS BulanıkTerim" şeklindedir; koşullar AND(Ve), OR(Veya) ile birleştirilebilmektedir ya da koşula NOT işlemi uygulanabilmektedir.

SummarySQL’ de yazılan bir bulanık sorgu cümlesi ve sonuçları Tablo3.1’ de gösterilmektedir:

Sorgu, “find all tall persons from the table persons”.

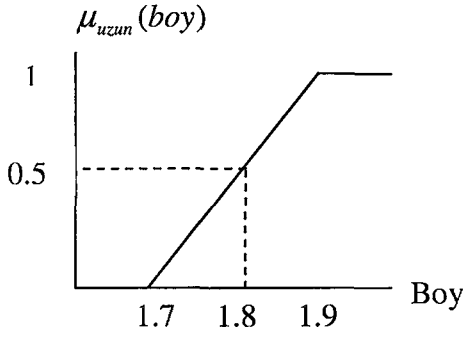
```
Select *  
From Persons  
Where height is tall ;
```

Tablo 3.1 Sorgu Sonuçları

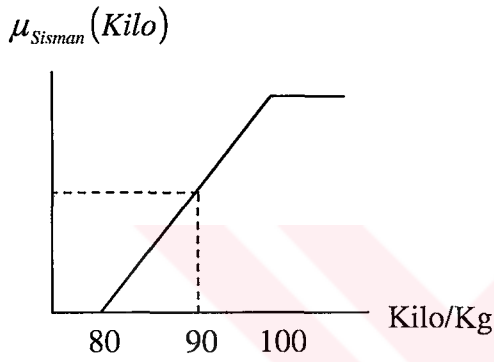
Name	height	weight	μ
John	195	96	1
Ben	193	101	1
Jake	191	97	1
Harry	190	99	1
Jerry	187	96	0,85
Sam	185	97	0,75
Burt	185	96	0,75
Dan	180	93	0,5
Dustin	177	87	0,35
Adam	173	83	0,15
Michael	170	78	0
Eric	168	75	0
Paul	166	70	0

Yukarıdaki bulanık sorguda, ‘*’, veri tabanındaki tüm alanları temsil etmektedir. Sonuç kümesinde yer alan μ sütunu ise, kayıtların sorguya olan uygunluk derecelerini vermektedir.

SummarySQL ‘de bir özet nasıl değerlendirilir? Bunu görebilmek için Şekil 3.1a ve 3.1b’ deki “Uzun” ve “Kısa” bulanık kümelerinden yararlanılmıştır.



Şekil 3.1. a) “Uzun” Bulanık Kümesi



Şekil 3.1. b) “Sisman” Bulanık Kümesi

Aşağıdaki örnekte “Uzun insanların çoğu şismandır” özeti değerlendirilmiştir (Schwartz and Klir 1992).

Summary Çoğu

From (Select *

From Persons

Where boy is uzun)

Where kilo is sisman

Result: 0.943878

Bulanık sorgu cümlesinin “Summary” kısmı, dilsel özet diye adlandırılmaktadır. Burada özetlenen bilginin, veri tabanındaki kayıtlar arasında geçerli olup olmadığı sonucu elde edilmektedir. Yukarıdaki “boyu uzun olanlar şisman olur” bulanık sorgusunun, “çoğu” dilsel özeti için, 0.943878 derecesinde doğru olduğu

anlaşılmaktadır. Dilsel özetlerin $[0,1]$ aralığında bir doğruluk derecesine atandığı görülmektedir.

Bir dilsel özetin doğruluk derecesi τ olsun. Dilsel özet için standart form “ Q nesne in R are S” şeklindedir (Raschia and Mouaddib 2002).

Burada, Q bulanık niteleyici, R veri tabanındaki ilişki kümesi, S de özetleyici olarak kullanılan bulanık kavramdır. τ hesaplanması şu şekilde gerçekleştirilir.

- 1- Veri tabanındaki (R'deki) her kayıt (t) için, S(t), S özetleyicisini belirten bir derecedir.
- 2- $r = \frac{1}{n} \sum_{i=1}^n S(t)$, Veri tabanında S özetini sağlayan kayıtların oranı bulunur.
- 3- $\tau = Q(r)$, r'nin özetle ilgili sürülen miktara uygunluk derecesi hesaplanır.

Bundan başka, yazarlar “Summary” düşüncesini, “Q S1 objects in R are S2” örneğin “Most tall people in R are Young” şeklindeki dilsel özetler genel formuna genişletmektedirler. Bu durumda, r, R'deki S1 nesnelere oranı diye hesaplanır ve bu da S2 değerini karşılar. Daha sonra r, olayı dilsel özetleri doğruluk dereceleri ile birlikte bulanık yüklem gibi hesap etmeye dönüştürür. Sonuç olarak, yaklaşım veri tabanlarından bulanık kurallar yaratmak olarak ele alınabilir. Gerçekte, “Q S1 objects in R are S2” özetinin doğruluk derecesi, veri tabanından çıkarılan varsayıma dayanan “S1 in R s2” bulanık kuralının doğruluğunun kanıtlanmasını sağlar. Bu işlemi gerçekleştirmek için Summary-SQL diye adlandırılan bir bulanık sorgulama dili geliştirilmiştir (Raschia and Mouaddib 2002).

3.3. Sonuç

Yukarıdaki çalışmalar esnek sorgu literatüründe üzerinde en çok yoğunlaşılacak örnek çalışmalardır. Bunlar incelendiği zaman şimdiye kadar oluşturulan bulanık sorgulama dillerinin standart SQL sorgulama dilinin bir uzantısı olarak oluşturulduğu görülmektedir.

Bu alıřmalardan farklı olarak, tez kapsamında geliřtirilen bulanık sorgu aracı, standart SQL cümlelerinin uzantısı olarak oluşturulmamıřtır. Bu yazılımda bulanık sorgu cümleleri, gündelik konuşmadaki ifadelerin yardımıyla oluşturulmuřlardır. 4. bölümde geliřtirilen bulanık sorgu aracı ayrıntılı řekilde anlatılacaktır.



4 AKILLI SORGULAMA ARACI

4.1 Giriş

Veri tabanlarında gerçekleştirilen klasik sorgular ile, yalnızca sorgu cümlesi ile birebir eşleşen kayıtlara ulaşılabilir. Örneğin mağazadaki en yüksek maliyetli ürünü bulmak için sorgulama yaptığımızda eğer bu ürünün maliyeti 5 milyon lira ise; klasik sorgular ile sadece maliyeti 5 milyon lira olan ürünlere erişilebilir. Bu durumda 4.99 milyon lira maliyetindeki ürünler de çok yüksek maliyetli olmalarına rağmen dikkate alınmamış olacaklardır. Elde edilen sonuç matematiksel olarak istenen sonucu verse de, sezgisel olarak gerçeği tam anlamıyla yansıtmamaktadır. Çok küçük bir matematiksel farktan dolayı, 4.99 milyon lira olan ürün yüksek maliyetli sayılmamaktadır. Bu tür durumlarda bulanık mantık alternatif çözüm yöntemleri üretilmesine olanak vermektedir ve bu alternatif yöntemlerde, söz konusu probleme, herşeye bir doğruluk derecesi vererek çözüm getirir.

Son yıllarda bulanık sorgular üzerinde yapılan çalışmaların sayısında artış gözlenmektedir (Rasmussen and Yager 1997, Bosc and Pivert 1997, Rasmussen And Yager 1997, Cox 2000, Eminov, Galindo at al 2002, Ribiero and Moreira 1999).

Bu tez çalışması kapsamında, bulanık mantığın veri tabanı sorgulamalarında mevcut yöntemlere nasıl alternatif olabileceği anlatılmış ve klasik veri tabanları üzerinde esnek sorgulama yapabilen bir sorgu arayüzü geliştirilmiştir. Esnek sorgulama işlemi bulanık mantık küme teorisine dayanarak gerçekleştirilmiştir. Bu bağlamda, tez kapsamında geliştirilen sorgulama aracına bir bulanık sorgulama aracıdır. Bundan sonraki bölümlerde sorgu arayüzünden “Akıllı Sorgulama Aracı” (ASA) diye bahsedilecektir.

ASA ile veritabanlarından, daha insan-merkezli; başka bir ifade ile konuşma dilindeki ifadeler ile sorgular yapılabilmektedir. Bu ifadeler, bulanık ifadeler diye adlandırılmaktadırlar, “iyi”, “çok”, “ortalama”, “ağır”, “başarılı” gibi. Konuşma dilinde kullanılan kelimeler ile sorgu geliştirilebilmesinin yanında, bahsedilen bulanık ifadeler, tamamen kullanıcı tarafından sezgisel olarak oluşturulabilmektedir. Örneğin, fiyatı 5 milyon ve üzerindeki ürünler pahalı sayılırken; kullanıcılar, 4-5 milyon arasındaki ürünleri de belirli oranlarda pahalı ürün kümesine dahil edebilmektedirler. Yani kullanılan bulanık ifadeler ve sınırları kullanıcının seçimine bağlı olarak sınırsız şekilde gerçekleştirilebilmektedir.

Bulanık kümelerin sınırlarının belirlenmesinde, bu konuda uzman olan ya da tecrübeli insanlara görev düşmektedir. Çünkü, bulanık kümelerin optimum tanımlanmasıyla orantılı olarak, bulanık sorgunun doğruluğu ve işlevselliği artmaktadır. Bu da elde edilen sorgu sonuçları doğrultusunda daha sağlıklı kararların alınmasına neden olacaktır. Bu anlamda bulanık kümelerin sınırlarını iyi belirlenmesi elde edilen sonuçların gerçeği yansıtmasında önemli bir etkidir. Bunun için bulanıklaştırma aşamasında, bulanık kümelerin sınırlarının belirleyen kişilerin alanında uzman ya da konu hakkında tecrübeli insanlar olmaları gerekmektedir.

Kısaca esnek sorgulamada, veri tabanında yer alan her kayıt, kullanıcı tarafından tanımlanmış olan her bulanık küme üyesidir. Bu üyelikler $[0,1]$ arasında derecelendirilmiş durumdadırlar. Yani kullanıcıya döndürülen sonuç kümesinde, klasik SQL sorgularındaki gibi sorguya uyan kayıtlar yerine; veri tabanındaki her kayıt, bulanık sorguya olan uygunluk derecesi ile birlikte yer almaktadırlar.

Bu bölümde, buraya kadar olan bölümlerde geliştirilme amacı anlatılmaya çalışılan tez çalışması süresince üzerinde çalışılan esnek sorgulama arayüzü ayrıntıları ile anlatılacaktır.

4.2. Neden Bulanık Sorgu ?

Klasik sorgular, insan düşüncesini tam anlamıyla yansıtamayabilir. Bu da, aynı sınıfta düşünülen bazı kayıtların farklı sonuç kümelerinde değerlendirilmesine neden

olmaktadır. Uygulamada, genellikle kullanılan sorgulama dili, ilişkisel veri tabanları için standart hale geldiğinden beri SQL sorgulama dilidir (Elmasri and Navathe 2000). SQL, sorgular için kesin kurallar ve komutlar içeren bir dildir. Sorgu sonuçları, sorgu kriterlerini sağlayan kayıtlardan oluşmaktadır. Bulanık sorgular ise kayıtların, sorguda aranan koşullara uygunluk derecelerine [0,1] arasında değerler vererek değerlendirme yapılmasını sağlar.

Bulanık mantık, insan düşüncesine olan yakınlığı ve bulanık ifadeleri işleyebilmesi ile çözüme en uygun yöntem olarak belirlenmiştir (Zadeh 1965). Yukarıda anlatılmak istenenler aşağıda bir sorgu üzerinde incelenmektedir.

Tablo 4.1. Laboratuar Tablosu

NUMARA	AD	FNOTU	RAPOR	DEVAM
1	SEVINC	84	80	9
2	SERPİL	89	30	7
3	AYSEL	20	67	2
4	ENİS	80	70	3
5	ONUR	68	50	8
6	HARUN	40	50	9
7	OZCAN	83	65	10
8	GUNAY	30	20	4
9	BANLI	81	90	9
10	LEVENT	90	85	2
11	NURAY	81	85	7
12	EYLEM	95	95	9

Tablo 4.1' deki örnek tablo üzerinde sorgulama işlemini gerçekleştiririm. Tablodaki kayıtlardan başarılı öğrencileri elde edelim. 'Notu ≥ 85 ' ve 'Devam ≥ 8 ' olan öğrenciler başarılı olarak tanımlanmaktadır. Bu ölçüte göre geleneksel bir sorgu ve sonuçları şu şekilde olacaktır :

```
SELECT NUMARA, AD, FNOTU, DEVAM FROM LABORATUAR WHERE
FNOTU $\geq$ 85 AND DEVAM $\geq$ 8
```

NUMARA	AD	FNOTU	DEVAM
2	SERPİL	89	8
12	EYLEM	95	10

Yukarıdaki sonuçlar dikkatlice incelendiğinde, Tablo 4.1'deki 1 numaralı öğrencinin devam durumunun çok iyi olmasına rağmen, final notu 84 olduğu için listede yer almadığı gözlemlenir. Belirlenen sorgu kriterlerine göre çok yakın durumda olan kayıtlar farklı sonuç kümelerinde yer almaktadırlar. Bu da değerlendirmeyi esneklikten uzaklaştırır ve değerlendirmede kesin kurallar dışına çıkılmasına izin vermez. Sorgu aralığının genişletilmesi bu probleme çözüm olarak sunulabilir. SQL sorgusunun WHERE bloğunda FNOTU>=83 VE DEVAM>=7 şeklinde bir değişiklik yaptığımız zaman elde edilen kayıtlar aşağıda görülmektedir.

NUMARA	AD	FNOTU	DEVAM
1	SEVINC	84	10
2	SERPİL	89	8
7	OZCAN	83	8
12	EYLEM	95	10

Bu kez de 9 ve 11 numaralı öğrencilerin notları ve devam durumları seçilen kriterlere çok yakın olsalar dahi, yine sonuç kümesinin dışında kalmışlardır. 12 kayıtlı bir veri tabanında sorgu aralığını genişletmek, bir çözüm olarak düşünülebilir olsa da; çok daha büyük boyutlu veri tabanlarında bu çözüm yöntemi kabul edilemez. Klasik sorgu cümlesi ile gerçekleştirmeye çalıştığımız sorguyu, geliştirilen ASA ile gerçekleştirmeye çalışalım. "FNOTU MÜKEMMEL VE DEVAM İYİ " şeklinde başarılı öğrencileri listelemek için oluşturulan bir bulanık sorgu cümlesini değerlendirelim. Bulanık sorgu sonucunda her kaydın hangi sonuç kümesine üye olduğunu sonuç kümelerine olan üyelik dereceleri ile birlikte gözlemleyebiliriz. Böylece başarı durumunu değerlendirebilmek için derece farklılık gösteren çok daha geniş bir başarılı öğrenci kümesi elde edebiliriz ve çok küçük not farklılıklarından dolayı başarılı ya da başarısız diye birbirinden ayrılan öğrencilerin, başarılı kavramına yakınlık durumlarını elde edip, öğrenciler için daha sağlıklı kararlar alabiliriz. Klasik sorguda sadece belirlenen aralıktaki öğrencilerin başarı durumları hakkında bilgi sahibi olunabilirken; bulanık sorgu sayesinde veri tabanındaki tüm kayıtların başarı durumları hakkında bilgi sahibi olunabilir. Geliştirilen yazılımla tüm kayıtları başarı durumları ile birlikte veren ekran görüntüsü Şekil 4.1'de verilmiştir.

NUMARA	AD	FNOTU	DEVAM	SONUC
1	SEVINC	84	9	0,4
2	SERPIL	89	7	0,5
3	AYSEL	20	2	0
4	ENIS	80	3	0
5	ONUR	68	8	0
6	HARUN	40	9	0
7	OZCAN	83	10	0,3
8	GUNAY	30	4	0
9	BANU	81	9	0,1
10	LEVENT	90	2	0
11	NURAY	81	7	0,1
12	EYLEM	95	9	1

Şekil 4.1. “Fnotu Mükemmel Ve Devam İyi” Sorgusunun Sonuçları

4.3. Programın Genel Yapısı

Program Delphi 6.0 ortamında hazırlanmış, VTYS olarak SQL Server 7.0 kullanılmıştır. Temelde iki kısımdan oluşmaktadır.

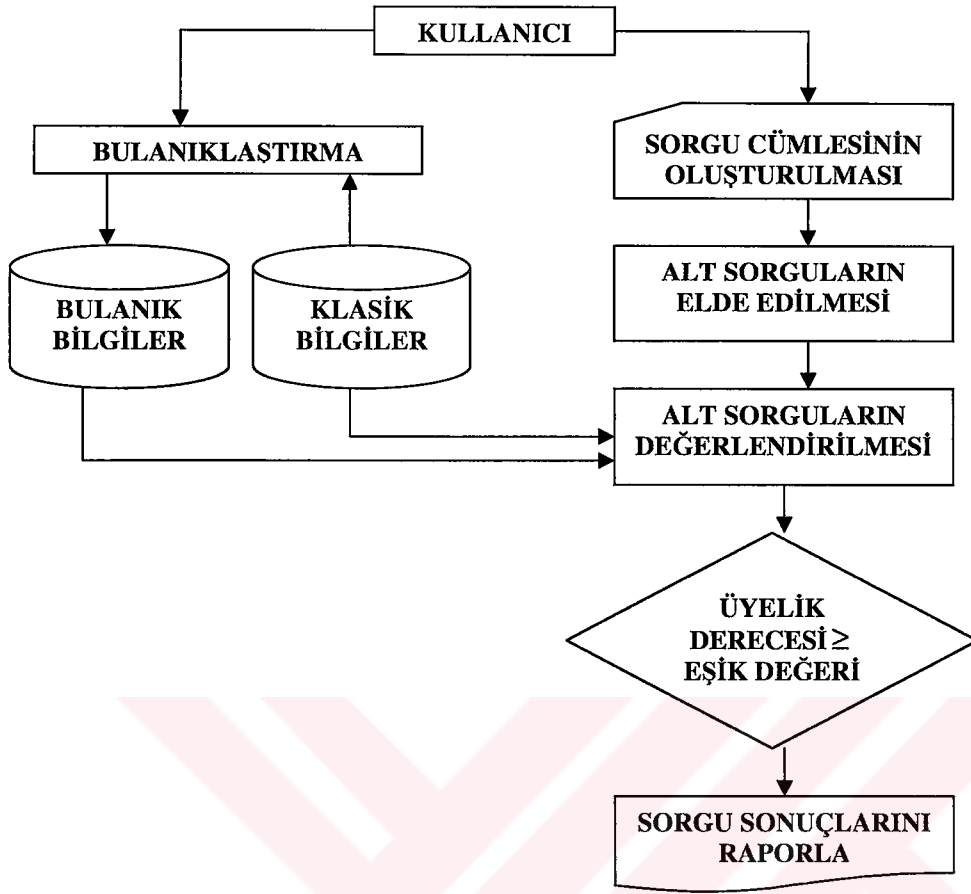
- Nitelik(attribute) değerlerinin bulanıklaştırıldığı kısım
- Esnek sorgulamanın gerçekleştiği kısım.

Programda, öncelikle kullanıcı istekleri doğrultusunda bulanık ifadeler tanımlanmış, bu işlemin sonrasında ise tanımlanan bulanık ifadeler yardımı ile esnek sorgulama işlemi gerçekleştirilmiştir.

Geliştirilen ASA, burs başvurusu yapmış olan öğrencilerin durumlarının değerlendirilmesinde ve burs alacak öğrencilerin belirlenmesinde kullanılmıştır.

Bundan sonraki bölümlerde programın işleyişi ‘BURS’ veri tabanı üzerinde anlatılacaktır.

Programa ait genel yapı, Şekil 4.2’ de özetlenmiştir.



Şekil 4.2. Programa Ait Akış Diagramı

4.4. Niteliklerin Bulanıklaştırılması

Bulanık küme, değişik üyelik dereceleri olan elemanların oluşturduğu kümedir. Bir bulanık kümenin elemanları, üyelik derecelerinden oluşan bir evrensel küme üzerinde dağılmış durumdadırlar. Elemanların, aynı evrensel kümede farklı bulanık kümelerine üyelikleri bulunabilir. Yani bulanık olmayan (crisp) bir değer ancak 1 kümenin elemanı olabilirken, bulanık değer farklı bulanık kümelere üye olabilir.

Bulanık küme elemanları, “üyelik dereceleri” evrensel kümesine teorik fonksiyonlar ile yerleşirler. Örneğin A bulanık kümeyi ifade etsin. Bahsedilen fonksiyonlar, A kümesinin elemanlarının $[0,1]$ aralığında değerler almalarını sağlar. Varsayalım x, A bulanık kümesinin bir elemanı olsun. Bu durum ve A bulanık kümesi aşağıdaki gibi ifade edilir.

$$\mu_A(x) \in [0,1] \quad (4.1)$$

$$A = (x, \mu_A(x) / x \in X) \quad (4.2)$$

$$A = \sum \frac{\mu_A(x_i)}{x_i} = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n \quad (4.3)$$

Programda, üyelik fonksiyonlarının belirlenmesinde yukarıdaki denklem (Denklem 4.3) kullanılmıştır. Yani üyelik fonksiyonu, eleman ve üyelik derecesi çifti ile belirlenmiştir (Jamshidi 1993).

Bulanıklaştırma işlemlerini gerçekleştirebilmek için, öncelikle üzerinde çalışılacak olan verilerin depolandığı tabloya erişmek gerekmektedir. Bulanık olmayan tüm değerlere ulaşıldıktan sonra bu değerlerin bulanıklaştırılmasına geçilebilir. Bu aşamada, seçilen tablo niteliklerinin üzerinde bulanık kümelerin tanımlanması işlemi gerçekleştirilir. Hangi nitelik bulanık olarak tanımlanmak isteniyorsa, ilgili evrensel küme üzerinde üyelik fonksiyonları yardımıyla bulanık kümeler tanımlanır.

SQL Server üzerinde programa ait iki veri tabanı yer almaktadır. Bunlardan bir tanesi kullanıcının amacına hizmet edecek bulanık olmayan verilerin depolandığı, “KlasikDB” adındaki veri tabanı; bir diğeri ise daha sonra anlatılacak olan bulanık bilgi tabanının yer aldığı “BulanıkBilgiler” veri tabanıdır.

Program ilk çalıştığında MS SQL Server’a bağlanarak, daha önceden oluşturulmuş olan “KasikDB” altındaki kullanıcı tablolarına erişir. Bunlar, kullanıcının üzerinde işlem yapabilmesi için aktif form üzerinde listelenir ve üzerinde sorgulama yapılacak tablo listeden seçilir. Sonuçlar arasında görünmesi istenen nitelik değerleri de yine kullanıcı tarafından nitelik değerleri listesinden seçilir.

Numarası	AdiSoyadi	Cinsiyet	AylikAileGeliri	KiraBedeli	AylikGeliri	KardesSayisi	OkuyanSayisi	TopBursMik
201010	Mustafa Gokcelik	E	250	<NULL>	100	10	5	0
201011	Sevinc Ilhan	K	500	100	200	2	2	0
201012	Ozcan	E	150	<NULL>	60	7	2	0
201013	Muhsin Aler	E	335	0	100	2	0	0
201014	Gozde Giray	K	100	120	100	4	3	0
201015	Silen Timur	K	350	50	0	4	1	0
201016	Zeynel Onder	E	150	0	100	9	2	0
201017	Riza Kurban	E	200	0	200	3	2	0
201018	Muharrem Demir	E	590	0	150	5	3	0
201019	Ahmet Ugur	E	300	0	0	5	2	0
201020	Sevda Ergin	K	330	0	200	5	1	0
201021	Emine Turan	K	300	0	80	8	2	0
201022	Aydin Ozbey	E	420	0	150	14	6	0
201023	Murat Yurukcu	E	350	0	0	2	2	0

Şekil 4.3. BURS Tablosu

Örnek olarak, veri tabanındaki, Şekil 4.3’ de gösterilmekte olan “BURS” tablosuna ait kayıtlar ile bulanık sorgulama gerçekleştirilmek istenmektedir. BURS tablosunun nitelik değerleri aşağıdaki gibidir:

BURS={Numarası, AdiSoyadi, Cinsiyet, AylikAileGeliri, KiraBedeli, AylikGeliri, KardesSayisi, OkuyanSayisi, AldigiBursMiktari}

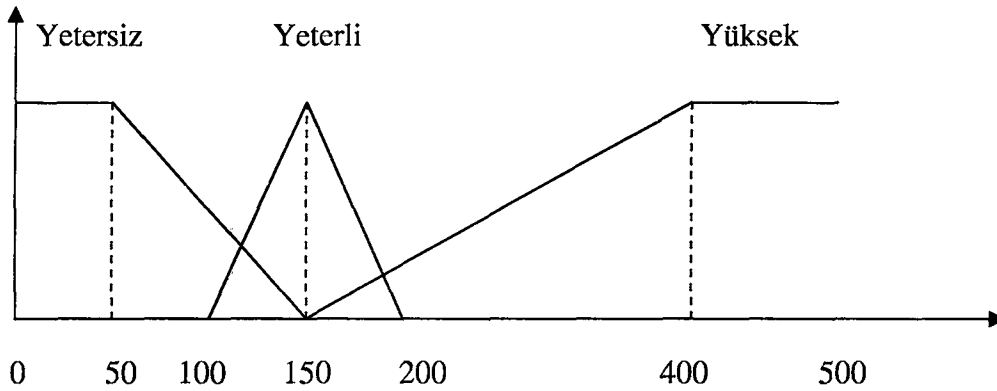
Bu alanlardan “AylikGeliri” üzerinde bulanıklaştırma işlemi gerçekleştirilecektir. Bunun için, bulanık kümenin değer aralığı ve sınırları belirtilmelidir. Değer aralığı oluşturulan model hakkında tecrübeli kişilerce belirlenmelidir. “AylikGeliri” niteliği için değer aralığı 0-500 milyon olarak belirlenmiştir (Bkz. Şekil 4.4). Bu da veri tabanında yalnızca bu aralıktaki kayıtların aylık gelir durumları incelenecek anlamına gelmektedir.

Eğer “AylikGeliri” niteliği üzerinde ilk kez bulanıklaştırma işlemi gerçekleşiyor ise min ve max alanları değer aralığını belirleyebilmek için aktif durumdadır. Buna karşın daha sonra yine aynı nitelik üzerinde üyelik fonksiyonu tanımlanmak istenirse min ve max alanları tekrar düzenlenemez. Program böyle bir değişikliğe izin vermez. Yani bir nitelik için bu aralık saptaması bir kez ve başlangıçta gerçekleştirilir. Bu bulanık kümeleri isimlendirirken herhangi bir kısıtlama yoktur. Yalnızca aynı nitelik değeri üzerinde üyelik fonksiyonları isimlendirilirken farklı isimler kullanılmalıdır. Farklı nitelik değerleri üzerinde aynı isimle üyelik fonksiyonları tanımlamakta sakınca yoktur. Örneğin “KiraBedeli” alanı üzerinde birden çok üyelik fonksiyonunun adı “Yüksek” olamaz. Ama, “KiraBedeli” alanında “Yüksek” üyelik

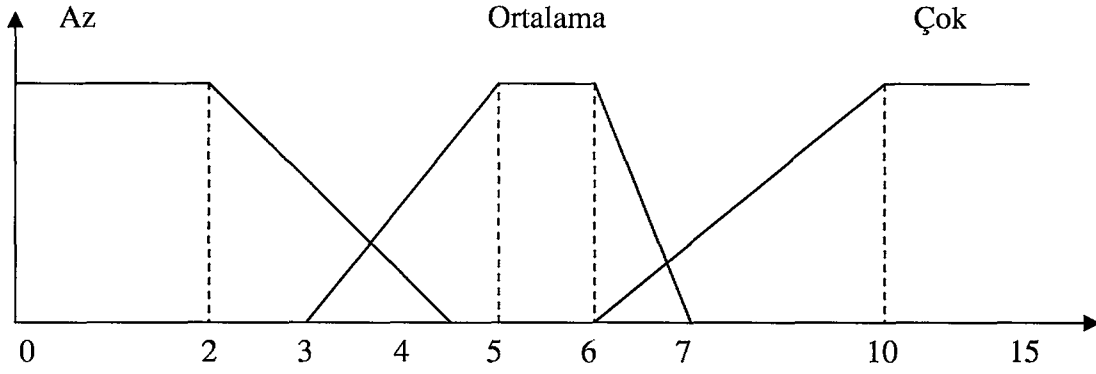
fonksiyonu tanımlıyken; “AylıkGeliri” alanında da aynı isimli üyelik fonksiyonu tanımlanabilir.

Şekil 4.4. Bulanık Küme Seçim Ekranı

“BURS” tablosundaki “AylıkGeliri” ve “KardesSayisi” alanları üzerinde kullanıcı tarafından belirlenen üyelik fonksiyonları Şekil 4.5a, b’ de belirtilmiştir.



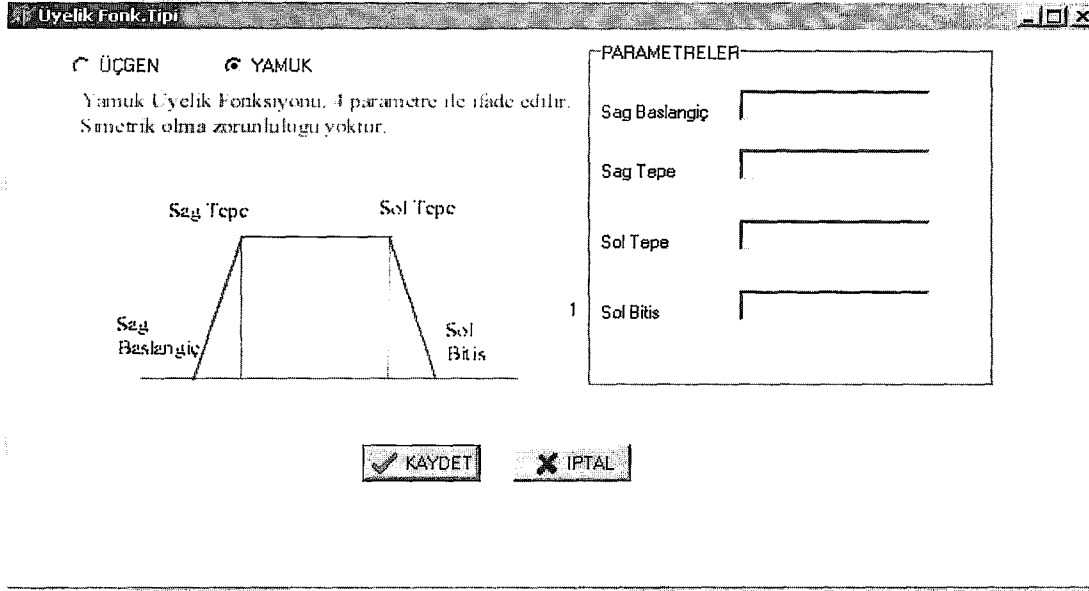
Şekil 4.5. a) “AylıkGeliri” Nitelik Değerine Ait Üyelik Fonksiyonları



Şekil 4.5. b) "Kardeş" Nitelik Değerine Ait Üyelik Fonksiyonları

Burs değerlendirmesi için belirlenen bu üyelik fonksiyonu tanımlarını veri tabanındaki bulanık bilgi tablosuna aktarmak gerekmektedir. Bu işlem için bulanık küme üzerinde tanımlanacak üyelik fonksiyonunun adı ve tipi yani teorik formu belirtilmelidir. Üyelik Fonksiyonu triangular (üçgen), trapezoid (yamuk), exponential, S-function etc ... şeklinde olabilmektedir. Geliştirilen bu uygulamada üyelik fonksiyonu tipi olarak triangular (üçgen) form (Şekil 4.6a) ve trapezoid (yamuk) form (Şekil 4.6b) seçenekleri sunulmuştur. Yazılan bulanık sorguların konuşma diline yakın olabilmesi için bu fonksiyonlar, konuşma dilinden alınan ifadeler ile adlandırılmıştır. Örneğin, "AylıkGeliri" bulanık kümesi üzerinde, "Düşük", "Ortalama", "Yüksek" üyelik fonksiyonları tanımlanmıştır.

Şekil 4.6. a) Üçgen Üyelik Fonksiyonu



Şekil 4.6. b) Yamuk Üyelik Fonksiyonu

Bulanıklaştırma için gereken bilgiler kullanıcı tarafından eksiksiz doldurulduğu zaman bulanıklaştırma işleminin gerçekleştirildiği tablo adı, nitelik adı, bulanık kümenin sınır değerleri, tanımlanan üyelikfonksiyonu adı, tipi gibi bulanıklaştırma ile ilgili tüm “BulanikBilgiler” veri tabanındaki “Bulanikkume“ tablosuna eklenir. Tanımlanan her üyelik fonksiyonu “BulanikKume” tablosuna yeni bir kayıt eklemek demektir.

Üyelik fonksiyonun tipine göre, mevcut kayıtlar üzerinde farklı işlemler uygulanacak demektir. Bir bulanık küme üzerinde üçgen bir üyelik fonksiyonu tanımlamak demek üyelik fonksiyonunu oluşturan elemanlara aşağıda belirtilen işlemlerin uygulanması demektir. Benzer işlemler yamuk üyelik fonksiyonu için de uygulanacaktır.

$$\mu_A(x) = \begin{cases} 0 & \text{..... } x < a \text{ yada } c < x \\ \frac{x-a}{b-a} & \text{..... } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{..... } b \leq x \leq c \end{cases} \quad (4.4)$$

Bu denklemde, üçgen üyelik fonksiyonunu oluşturan a, b, c değerleri şu noktaları temsil etmektedir.

a: Sağ başlangıç, b: Tepe, c: Sol bitiş.

Bulanık küme için tanımlanan üyelik fonksiyonu tipi trapezoid(yamuk) seçilir ise üyelik fonksiyonu için kullanılacak denklem şu şekilde tanımlanmaktadır.

$$\mu_A(x) = \begin{cases} 0 & \text{..... } x < a \text{ yada } d < x \\ \frac{x-a}{b-a} & \text{..... } a \leq x \leq b \\ 1 & \text{..... } a \leq x \leq c \\ \frac{d-x}{d-c} & \text{..... } c \leq x \leq d \end{cases} \quad (4.5)$$

Bu denklemde, yamuk üyelik fonksiyonunu oluşturan a,b,c,d değerleri şu noktaları temsil etmektedir.

a:Sağ başlangıç, b: Sağ tepe, c:Sol tepe, d:Sol bitiş

Üyelik fonksiyonunun tipinin üçgen ve yamuk seçilmesine göre hesaplama işlemi “UyderBul” prosedüründe yapılmaktadır (Ek-2).

Sonuç olarak, bulanıklaştırma işleminde, üzerinde sorgulama yapılacak verilerin bulunduğu tablodaki nitelikler üzerinde istenildiği kadar aynı isimli üyelik fonksiyonu tanımlanabilir. Bu anlamda bulanık küme ve üyelik fonksiyonlarının belirlenmesinde, yani bulanıklaştırma işleminde kullanıcıya maximum esneklik sunulmuştur. Gerçekleştirilen bu işlemler sonucunda, sorgulama yaparken bakılacak olan bulanık bilgi tabanı oluşturulmuş olmaktadır.

4.5. Bulanık Sorgulama

Programın ikinci kısmında bulanık sorgulama işlemi gerçekleştirilmektedir. Bulanık sorgu cümlesi oluşturulmadan önce yapılması gereken ilk işlem, üzerinde çalışacağınız bulanık olmayan verilere erişmektir. Bunun için kesin değerlerin

saklandığı KlasikDB veri tabanındaki tabloların listesi aracılığı ile, istenen tabloya bağlantı kurulur. Sonraki adımda tablonun alanlarından sorgu sonuç listesinde görüntülenmesi istenenler seçilerek yeni bir listeye aktarılır. Nitelik seçimi de tamamlandıktan sonra sıra sorgu yapılandırma kısmına gelir. Daha önceden tanımlanmış olan bulanık kümelerin ya da mantıksal operatörlerin yardımı ile bulanık sorgu cümlesi oluşturulur. Bunun için gerekli olan bulanık küme ve operatörlerin seçimi Şekil 4.7a, b’de gösterilmektedir. Sorguya istenen sayıda özellik eklenebilir. Birden fazla kriter ile sorgulama gerçekleştirilmek istenirse, standart SQL sorgularında kullanılan Ve/Veya bağlaçlarından yararlanır. İstenen sorgu koşulları Ve/Veya bağlaçları ile birleştirilmiş olur.

Şekil 4.7. a) Bulanık Küme Yardımı ile Sorgu Oluşturma

Şekil 4.7. b) Mantıksal İfadeler ile Sorgu Oluşturma

Sorgu cümlesi oluşturulduğunda programın gerçekleştirdiği ilk işlem sorgu cümlesinin yazımını kontrol etmektir. Programda belirlenmiş bulanık sorgu cümlesinin genel formatı aşağıda belirtildiği gibidir. Bu formata uymayan bulanık sorgular hatalı olarak değerlendirilecektir.

NİTELİK DEĞİSTİRİCİ ÜYELİK FONKSİYONU DEĞİL (VE / VEYA)

NİTELİK “=, <, >, <=, >=,” SABİT SAYI (VE / VEYA)

Bulanık sorgu cümlesi oluşturulup, işlenmeye başlandığında; öncelikle cümle içerisindeki bağlaçlar (VE/VEYA) bulunur. Bulunan bağlaçlar dinamik bir diziye yerleştirilirler (Tablo 4.2). Bulanık sorgu cümlesi bağlaçlar ile parçalanarak alt sorgu cümleleri oluşturulur. Oluşan her alt sorgu cümlesi yine dinamik bir dizi yapısında saklanır. Dizi elemanları tek tek programın “syntax prosedürü”ne gönderilir (Ek-1). Eğer yazımları üstte belirtilen sorgu formatına uyuyorsa “doğru” sonucu ile prosedürden geri dönerler. Oluşan alt sorgulardan bir tanesi dahi kurala uymaz ise yazılan bulanık sorgu cümlesinin “Hatalı” yazıldığı mesajı kullanıcıya gider ve bulanık sorgunun yeniden oluşturulması istenir. Tüm alt sorguların yazımı istenen formatta ise bulanık sorgu cümlesinin yazımı doğrudur ve bu da sorgu değerlendirilebilir demektir.

Tablo 4.2. Bağlaç Dinamik Dizisi

VE	VEYA	VE
----	------	-------	----

Yazımı test edildikten sonraki aşama, sorguyu işleme aşamasıdır. Sorgu işleme aşamasının işleyişi şu şekildedir. Programın “sorguislem” prosedürüne gelen alt sorgu cümlesinin içerisinde geçen üyelik fonksiyonunun adı araştırılır. Sorgu içinde üyelik fonksiyonunun adını bulmak demek aynı zamanda sorgunun bulanık sorgu olduğunun belirlenmesi demektir. Bu durumda “BulanikKume” tablosundan, sorguda adı geçen üyelik fonksiyonuna ait bilgiler alınır ve böylece oluşturulan bulanık küme üzerinde işlem yapabilmek için ihtiyaç duyulan tüm veriler elde edilmiş olur.

”KlasikDB” veri tabanında bulanık olamayan deęerleri ieren tablo kayıtları tek tek incelenir. Her kaydın ilgili nitelik deęerinin üyelik fonksiyonuna olan üyelik derecesi hesaplanır. Bunu yaparken, üyelik fonksiyonunun tipine göre (ügen ya da yamuk) üyelik fonksiyonları için tanımlanmış denklemlerden yararlanılır. İlgili denklemlerden elde edilen deęerler [0,1] aralığındadır ve veri tabanındaki kaydın alt sorguda adı geen bulanık kümeye aitlik derecesini verir. Bu işlem her alt sorgu cümlesi için tekrarlanır.

Bu şekilde veri tabanındaki her kayıt için tek tek alt sorgulara aitlik dereceleri hesaplanmış olur ve bu deęerlerin tümü iki boyutlu dinamik bir dizide saklanır. Son aşama olarak ilk kayıttan başlayıp son kayda kadar, sorgu cümlesine toplam uygunluk derecesi hesaplanır. Ve bu hesaplanan deęerler, ekranda kullanıcıya gösterilir.

Buraya kadar anlatılan işlemleri bir örnek üzerinde inceleyelim. “BURS” tablosu üzerinde “AYLIKGELİRİ YETERSİZ VEYA KARDESSAYISI FAZLA” kuralına uyan öğrencileri bulmak için bir bulanık sorgu cümlesi oluşturulduğunu varsayalım.

Sorgu cümlesi iki alt sorgu cümlesine parçalanır. Öncelikle “AYLIKGELİRİ YETERSİZ” alt sorgusundaki nitelik deęeri ve üyelik fonksiyonu aranır. Bulanık olmayan deęerlerin yani öğrencilere ait burs başvuru bilgilerinin depolandığı “BURS” tablosundaki “AYLIK GELİRİ” alanına ait kayıtlardan bir kayıt kümesi oluşturulur. Bu kayıt kümesindeki her elemanın “YETERSİZ” bulanık kümesine olan üyelik derecesi hesaplanır. Bunu gerçekleştirebilmek için “YETERSİZ” bulanık kümesine ait bilgiler çekilmelidir. Bulanık bilgilerin saklandığı “BulanıkBilgiler” tablosundan Klasik tablo adı, “BURS” ve üyelik fonksiyonu adı, “YETERSİZ” olan kayda ait bilgiler çekilir. Elde edilen kayıt kümesinde üyelik fonksiyonunun tipine göre kullanılacak denklem belirlenir ve BURS tablosundaki ilk kaydın AylıkGeliri alanındaki deęer, üyelik fonksiyonuna ait denklemde yerine koyularak kaydın, kaydın üyelik fonksiyonuna üyelięi bulunmuş olur. [0..1] aralığında hesaplanan bu deęer, bulanık olmayan deęerlerin tutulduğu tablodaki ilk kaydın “AYLIKGELİRİ YETERSİZ” alt sorgusuna uygunluk derecesidir. Aynı yöntem ile tüm kayıtların sıra ile “AYLIKGELİRİ YETERSİZ” alt sorgusuna olan uygunluk dereceleri hesaplanır.

Daha sonra bu işlemler aynen “KARDESSAYISI FAZLA” alt sorgusu için de tekrarlanır. Ve tablodaki her kaydın ikinci alt sorguya olan uygunluk derecesi yukarıda anlatıldığı gibi hesaplanır.

Elde edilen uygunluk dereceleri iki boyutlu dinamik bir dizi yapısında saklanır. Dizinin ilk boyutu tablodaki kayıtların indisini; ikinci boyutu hangi alt sorgu cümlesine uygunluk derecesinin araştırıldığı yani alt sorgu cümlelerinin indisini tutar. Bu şekilde iki boyutlu dinamik bir dizide tüm üyelik dereceleri saklanmış olmaktadır (Tablo 4.3).

Tablo 4.3. Dinamik Dizi (n Kayıtlı bir Veri tabanı, m Parçalı bir Bulanık Sorgu)

	1.Alt Sorgu	2.Alt Sorgu	m.Alt Sorgu
1. Kayıt	$\mu_{1,1}$	$\mu_{1,2}$		$\mu_{1,m}$
2.Kayıt	$\mu_{2,1}$	$\mu_{2,2}$		$\mu_{2,m}$
.....				
n.Kayıt	$\mu_{n,1}$	$\mu_{n,2}$		$\mu_{n,m}$

Böylece tablodaki her kaydın (0'dan n'e kadar), m adet alt sorguya uygunluk dereceleri bilinmektedir. Bu işlemlerden sonra toplam bulanık sorguya olan uygunluk derecesinin yani sorguya ait olama derecesinin hesaplanması gerekmektedir. Sorguya olan toplam uygunluk derecesinin nasıl hesaplandığı aşağıda açıklanmaktadır.

Sorgu cümlesinde yer alan bağlaçların da dinamik bir diziye yerleştirildiğinden söz edilmişti. Bağlaç dizisinin “i”.elamanı “VE” ise üyelik derecelerinin saklandığı dizinin $\mu_{j,i-1}$ ve $\mu_{j,i+1}$.elemanlarının sayı değerleri , yani “j”.elemanın (i-1) ve (i+1). alt sorgulara uygunluk derecesi “Min” işlemine tabi tutulur. Bağlaç dizisinin “i”.elamanı “VEYA” ise $\mu_{j,i-1}$ ve $\mu_{j,i+1}$.elemanlarının sayı değerleri “Max” işlemine tabi tutulur.

1. ve 2. alt sorgular “VE” bağlacı ile bağlanmış ise,

$$\text{Sonuc} = \min(\mu_{\text{Sorgu1}}(x), \mu_{\text{Sorgu2}}(x)) \quad (4.6)$$

1. ve 2. alt sorgular “VEYA” bağlacı ile bağlanmış ise,

$$\text{Sonuc} = \max(\mu_{\text{Sorgu1}}(x), \mu_{\text{Sorgu2}}(x)) \quad (4.7)$$

Kayıtların sorguya toplamda uygunluk derecesi bulunurken, öncelik “VE” işlemindedir. Sorgu cümlesi boyunca, alt sorgular arası “VE” işlemi gerçekleştirilir. Daha sonra bulunan sonuçlar ile yenilenen dizide “VEYA” işlemi uygulanır.

Tüm bu işlemlerin sonunda veri tabanındaki her kayıt için bulanık sorguya uygunluk derecesi hesaplanmış olur (Şekil 4.8). Yazılan bulanık sorgunun veri tabanındaki bulanık olmayan verilere ne kadar etkideği, elde edilen [0,1] aralığındaki sayı değeri kadardır.

Klasik sorgu ile gerçekleşmesi durumunda elde yalnızca kesin olarak sorguya uyan kayıtların görüntüleneceği bir çıkış ekranı yerine, tüm verilerin oluşturulan sorgu cümlesine ne derece uygunluk gösterdiğini gözleyebildiğimiz bir çıkış ekranı elde edilmiş olur.

AKILLI SORGU ARACI

DOSYA BULANIK DEĞİŞTİRİCİ BAĞLAC RAPOR YARDIM HAKKINDA

OK AZ YAKLAŞIK VE NEHA DEĞİL

TABLO SEÇİM

BULANIK DEĞİŞTİRİCİ

NİTELİK SEÇİM

Numarası Adı Soyadı
Adı Soyadı Numarası
Cinsiyet Cinsiyet
Aylık Geliri Aylık Geliri
Kira Bedeli Kira Bedeli
Aylık Geliri Aylık Geliri
Kardes Sayisi Kardes Sayisi
Okuyan Sayisi
Top Burs Mik
Sonuc

SORGU YAPILANDIR

FAZLA

ESK DEĞERİ

0

BULANIK SORGU CÜMLESİ

AYLIK GELİRİ YETERLİ VE YA KARDES SAYISI FAZLA

TEMİZLE ÇALIŞTIR

SORGU SONUÇLARI

Adı Soyadı	Numarası	Cinsiyet	Aylık Geliri	Kardes Sayisi	SONUÇ
Mustafa Gökçelik	201010	E	110	10	1
Servinc İlhan	201011	K	200	2	0
Dücan	201012	E	60	7	0,9
Muhsin Alar	201013	E	100	2	0,5
Gözde Girey	201014	K	120	4	0,3
Silen Timur	201015	K	0	4	1
Zeynel Onder	201016	E	100	9	0,75
Rıza Kutban	201017	E	200	3	0
Muharrem Demir	201018	E	150	5	0
Ahmet Uğur	201019	E	0	5	1
Sevda Ergin	201020	K	200	5	0
Emine Turan	201021	K	80	8	0,7
Aydin Ozbey	201022	E	150	14	1

Sorgu Sonuçları

Şekil 4.8. Sorgu Sonucu

Programda oluşturulan sorgu cümlesinin bulanık sorgu olmadığı tesbit edildiğinde klasik yöntemler ile istenen koşullara uyum sağlayan kayıtların üyelik derecesi 1, uymayan kayıtların üyelik derecesi ise 0 olarak hesaplanır. Bundan sonra gerçekleşecek aşamalar yukarıda anlatılmıştır. Program bulanık sorgu ve klasik sorguyu aynı cümlede gerçekleştirebilmektedir.

4.5.1. Dilsel değıştiriciler

Değıştirici fonksiyonları, $[0,1]$ 'den $[0,1]$ 'e tanımlıdırlar ve bulanık küme üyelik fonksiyonlarına uygulanabilirler. Bunun için daha önceden belirlenmiş modeller "çok", "az", "yaklaşık", "az çok" vb. kullanılır (Wang 1994, Bosc and Prade 1993, Nakajima at al 1993).

Bizim sorgulama modelimizde, sorgulama konuşma diline yakın bir kullanıma olanak verecek biçimde geliştirilmiştir. Bunun için “çok”, “az”, “yaklaşık” dilsel değiştiriciler kullanılmıştır (Nakajima at al 1993).

Eğer bulanık sorgu “NİTELİK DEĞİŞTİRİCİ ÜYELİK FONKSİYONU”, kısaca “A is m MF” şeklinde ise; burada a, veri tabanındaki ilgili niteliğin değerini, m sorguda kullanılan dilsel değiştiriciyi, MF de bulanık kümeyi ifade etmektedir. O zaman değişikliğe uğrayan kural aşağıdaki şekildedir (Maelanin 1998).

$$K = \mu_m(\mu_{MF}(x)) \quad (4.8)$$

μ_m , $\mu_{MF}(x)$ üyelik derecesinin, m değiştirici fonksiyonuna olan üyelik derecesidir.

Örneğin “A1 m1 MF1 VE/VEYA A2 m2 MF2” şeklindeki bir sorgu cümlesi için yapılacak işlem şu şekildedir.

$$K = \min(\mu_{m1}(\mu_{F1}(x)), \mu_{m2}(\mu_{F2}(y))) , \text{bağlaç VE ise} \quad (4.9)$$

$$K = \max(\mu_{m1}(\mu_{F1}(x)), \mu_{m2}(\mu_{F2}(y))) , \text{bağlaç VEYA ise} \quad (4.10)$$

Örneğin, bulanık sorgu cümlesi “AYLIK GELİRİ ÇOK YÜKSEK DEĞİL” olsun. Sorgu sonuçları Şekil 4.9’ da görülmektedir.

Buradaki basit bulanık sorgu “AYLIK GELİRİ YÜKSEK” sorgusudur.

Veri tabanındaki herhangi bir kaydın bu sorguya uygunluk derecesinin [0,1] aralığında bir “ud” sayısı olduğu durumda,

ÇOK değiştiricisinin sorguya etkisi:

$$\text{ÇOK}(ud) = ud^2 , \text{şeklinde hesaplanır.} \quad (4.11)$$

DEĞİL değiştiricisinin sorguya etkisi:

$$\text{DEĞİL}(ud^2) = 1 - ud^2 , \text{şeklinde hesaplanır.} \quad (4.12)$$

O halde kaydın sorguya uygunluk derecesi $1 - ud^2$ şeklinde $[0,1]$ aralığında hesaplanmış olur.

Numarası	AdıSoyadı	AylıkGeliri	SONUÇ
201010	Mustafa Gokcelik	110	1
201011	Sevinc Ilhan	200	0,96
201012	Ozcan	60	1
201013	Muhsin Aler	100	1
201014	Gozde Giray	120	1
201015	Silen Timur	0	1
201016	Zeynel Onder	100	1
201017	Riza Kurban	200	0,96
201018	Muharem Demir	150	1
201019	Ahmet Ugur	0	1
201020	Sevda Ergin	200	0,96
201021	Emine Turan	80	1
201022	Aydin Ozbey	150	1

Şekil 4.9. “Aylık Geliri Çok Yüksek Değil” Bulanık Sorgu Sonuçları

Yaklaşık dilsel değıştiricisi kullanılarak “AYLIK GELİRİ =100” klasik sorgusu “AYLIK GELİRİ YAKLAŞIK 100” bulanık sorgu cümlesi haline getirilmiş olmaktadır. Yaklaşık dilsel değıştiricisinin sorguya olan etkisi şu şekilde hesaplanmaktadır.

Veri tabanındaki kesin değerler arasında AYLIK GELİRİ değışkeninin aldığı minimum ve maximum değerler bulunur. Veri tabanındaki kayıt sayısı bulunur.

$$\varepsilon = (\max \text{Değer} - \min \text{Değer}) / \text{Kayıt Sayısı} \quad (4.13)$$

gibi bir değer elde edilir ve, $80 - \varepsilon \leq \text{AYLIKGELİYİ} \leq 80 + \varepsilon$ aralığında olmayan kayıtların sorguya uygunluk dereceleri 0'dır kabul edilir.

Bu aralıktaki bir AYLIKGELİRİ değeri için ise sorguya uygunluk derecesi :

$$1 - \left(\frac{80 - \text{NOTU}}{\varepsilon} \right) \quad (4.14)$$

şeklinde hesaplanır.

Bir bulanık sorguda dilsel değiştirici ifadeler kullanılmış ise, yukarıda bahsedilen matematiksel işlemler üyelik derecelerine uygulanacak demektir. Ve bu işlemlerin yapılabilmesi için söz konusu olan verinin bulanık sorguya toplam üyelik derecesinin bulunmuş olması gerekmektedir.

4.5.2. α – Bölüm kesmesi (Eşik değeri)

Bulanık A kümesinin α – Kesme kümesi, A_α ile gösterilir ve X evrensel kümesinin A kümesindeki bütün elemanlarında üyelik derecesi α özel değerinden büyük ve eşit olanları içerir (Elmas 2003). Bu şu şekilde ifade edilebilir.

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad (4.15)$$

Örneğin Başarılı kümesinin elemanları $\{0.1, 0.2, 0.5, 0.6, 0.7, 0.9, 1\}$ iken;

$\alpha = 0.2$ için α -bölüm kümesi : $\{0.2, 0.5, 0.6, 0.7, 0.9, 1\}$,

$\alpha = 0.8$ için α -bölüm kümesi : $\{0.9, 1\}$ olarak elde edilir.

Programda, bulanık sorgu cümlesi oluşturulduktan sonra, sorgu sonuçları için α bölüm kümesi oluşturulmak isteniyorsa; α kesme değeri yani bir eşik değeri girilmelidir.

Kullanıcının seçimi doğrultusunda oluşturulan bulanık sorgular, değerlendirilmiş ve kayıtların sorguya üyelik dereceleri istenen eşik değeri belirtilerek listelenmiştir. Bu

şekilde veri tabanında sorguya uygunluk derecesi istenen sınırın altında ya da üstünde olan kayıtların listelenip listelenmemesi gibi bir seçim yapılabilir. Veri tabanındaki kayıtlar $[0,1]$ aralığında sonuç değerleri alırlar. Eşik değeri olmadan veri tabanındaki tüm kayıtlar listelenirken; eşik değeri kullanarak sadece α bölüm kümesini listelenmiş olmaktadır.

Örneğin, Şekil 4.8’ de “AYLIK GELİRİ YETERSİZ VEYA KARDESSAYISI FAZLA” sorgusunun sonuçları bir eşik değeri girilmeden listelenmiştir (Bkz. Şekil 4.8). Aynı sorgu cümlesi, eşik değerini 0,5 olarak seçilip işletildiğinde, sadece sorguya 0,5 ve üzeri derecesinde uygunluk gösteren kayıtlar listelenmiş olmaktadır (Şekil 4.10).

AdıSoyadı	Numarası	Cinsiyet	AylıkGeliri	KardesSayisi	SONUÇ
Mustafa Gökçek	201010	E	110	10	1
Özcan	201012	E	60	7	0,9
Muhsin Aler	201013	E	100	2	0,5
Silen Timur	201015	K	0	4	1
Zeynel Önder	201016	E	100	9	0,75
Ahmet Uğur	201019	E	0	5	1
Emine Turan	201021	K	80	8	0,7
Aydın Düzbey	201022	E	150	14	1
Murat Yurukcu	201023	E	0	2	1

Şekil 4.10. Eşik Değeri ile Bulanık Sorgulama

Çok büyük ölçekli veri tabanlarında çalışıldığı varsayılır ise, eşik değerini kullanmak, gerçekleştirilen sorguya en çok uyan kayıtları göstermek açısından elverişli bir yöntemdir. Aksi takdirde kayıtların sorguya olan uygunluk dereceleri

[0,1] arasında ne deęer alırsa alsın tümü listelenmiř olur ve bu da bulanık sorguyu verimli bir çözümler olmaktan uzaklařtırır.

4.6. Raporlama

Oluřturulan bulanık sorgu sonucu elde edilen kayıt kümesi programda listelendikten sonra raporlama iřlemi gerçekteřtirilebilir. Elde edilen sonuç listesi excel sayfasına aktarılmak suretiyle raporlama iřlemi gerçekteřtirilmiř olmaktadır (řekil 4.11).

	A	B	C	D	E
1	NUMARA	AD	FNOTU	DEVAM	SONUC
2	1	SEVINC	84	9	0,4
3	2	SERPIL	89	7	0,5
4	3	AYSEL	20	2	0
5	4	ENIS	80	3	0
6	5	ONUR	88	8	0
7	6	HARUN	40	9	0
8	7	OZCAN	83	10	0,3
9	8	GUNAY	30	4	0
10	9	BANU	81	9	0,1
11	10	LEVENT	90	2	0
12	11	NURAY	81	7	0,1
13	12	EYLEM	95	9	1

řekil 4.11. Sorgu Raporu

4.7. Bulanık Sorgu Kaydetme

Bulanık sorgular diske kaydedilip, daha sonradan kullanıcı isteęine baęlı olarak açılıp çalıřtırılabilmektedir.

5. SONUÇLAR ve ÖNERİLER

Bu tez çalışması kapsamında, konuşma dilindeki ifadeleri kullanarak çeşitli sorgulara cevap verebilen esnek ve akıllı bir sorgulama modeli geliştirilmiştir. Esnek sorgulama işleminin yapılabilmesi için klasik sorgu mantığının dışına çıkılması gerekmektedir. Uygulama dahilinde oluşturulan bulanık kümeler ile sezgisel sorgulama gerçekleştirilmektedir. Bu da gerçek dünyada klasik mantık ile sınıflandıramadığımız ancak, sezgisel olarak sınıflandırabildiğimiz verilerin daha sağlıklı değerlendirilmesine olanak tanımaktadır. Ayrıca bulanık mantık yardımı ile konuşma dilindeki ifadeler matematiksel olarak modellenabilmektedir ve bunun sonucu olarak konuşma dili ile sorgulama yapılabilmektedir. Çok daha geniş ölçekli kayıt kümeleri arasında insan-merkezli sorgular ile gerçek dünyada var olduğu düşünülen kümelere uygun kayıtlar izlenebilmektedir. Geliştirilen ASA' nın, üniversitenin burs işlerinde kullanılabilirliği ve değerlendirmede sağladığı avantajlar gösterilmiştir. Bulanık mantık ile değerlendirildiğinde daha sağlıklı sonuçlar verecek sistemler üzerinde de uygulanabilmektedir. Bu anlamda herhangi bir kısıtlama yoktur.

Yukarıda anlatılan özelliklerinden dolayı, bulanık sorgulama, günümüzde bilgi keşfine dayanan teknolojilerde (KDD) ya da veri madenciliği teknolojilerinde de sıkça başvurulan yaygın bir yöntem haline gelmiştir. İlerleyen dönemlerdeki çalışmalarda, veri tabanlarından bilgi keşfinin giderek önem kazandığı göz önünde bulundurulduğunda; bulanık sorgulamanın veri madenciliğinde uygulanması, daha büyük veri kümeleri arasından anlamlı verileri çekebilmek için bulanık sorgudan yararlanılması, ya da veri madenciliği tekniklerinin bulanık mantık teorisine dayandırılarak daha esnek bilgiye erişilmesi gibi konuların seçilmesi önerilebilir.

KAYNAKLAR

1. ELMAS, Ç., 2003. Bulanık Mantık Denetleyiciler. Seçkin Yayıncılık, 53-92, Ankara.
2. SCHWATRZ, D.G. and KLIR, G.J., 1992. Fuzzy Logic Flowers in Japan. IEEE Spectrum, July, 32-34 Vol. 1, 128-133, California.
3. BERENJI, H.R., 1992. in: R.R. Yager, L.A. Zadeh (Editors), An Introduction to Fuzzy Logic Applications in Intelligent Systems. Kluwer Academic Publishers, 69-95.
4. UMANO, M., HATONO, I., TAMURA, H., 1995. Fuzzy Database Systems, International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems, Vol. 5, 35.
5. RASMUSSEN, D., YAGER, R.R., 1997. SummarySQL – A Fuzzy Tool For Data Mining. Intelligent Data Analysis, Elsevier Science Inc.
6. ÖZTÜRK A., MERCAN, D.E., TOPRAK, F., KİŞİ, Ö., ŞAHİN, U., 2003. Bulanık Mantık Kurs Notları. İTÜ Bulanık Mantık ve Teknolojisi Klubü, 1-51, İstanbul.
7. BEZDEK J.C., 1993. Fuzzy Models-What are They and Why?. IEEE Trans. Fuzzy Systems, Vol. 1, No.1, 1-6.
8. ZADEH, L.A., 1965. Fuzzy Sets. Information and Control, 8, 338-353.
9. YAGER, R.R. and FILEV, D.P., 1994. Essentials of Fuzzy Modelling and Control. John Wiley&Sons, Inc, Canada, 1-387.
10. ZADEH, L.A., 1989. Knowledge Representation in Fuzzy Logic. IEEE Trans. Knowledge and Data Engineering, Vol. 1, No. 1, 89-100.
11. ZADEH, L.A., 1973. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-3, No. 1, 28-44.
12. VADIEE, N., 1993. Fuzzy Rule Based Expert System II., Fuzzy Logic Control. Prentice Hall, 1-389, New Jersey.
13. DRIANKOV, D., HELLENDORRN, H., REINFRANK, M., 1996. An Introduction to Fuzzy Control. Second Edition, Springer, 103-144, Germany.

14. KACPRZYK, J., ZIOLKOWSKI A., 1986. Database Queries with Fuzzy Linguistic Quantifiers, IEEE Trans. on Systems, Vol. SMC-16, May/June, 475-478.
15. BOSC, P. and PIVERT, O., SQLf: 1997. A Relational Database Language for Fuzzy Querying, IEEE Transactions on Fuzzy Systems, Vol. 3.
16. RASCHIA, G., MOUADDIB N., 2002. SAINTETIQ: A Fuzzy Set Based Approach to Database Summarization. Fuzzy Sets and Systems, 129, 137-162.
17. RASMUSSEN, D. and YAGER, R.R., 1997. Fuzzy Tool For Datamining. Intelligent Data Analysis, Vol. 1, Elsevier Science Inc.
18. COX, E., 2000. FuzzySQL-A Tool for Finding The Truth-The Power of Approximate Database Queries. PC AI-Intelligent Applications, Vol. 14.
19. Eminov, M., Querying Database by Fuzzification of Attribute Values. <http://idari.cu.edu.tr/sempozyum/bil46.htm>.
20. GALINDO, J., ARANDA, M.C., CARO, J.L., GUEVERA, A., AGUAYO, A., 2002. Applying Fuzzy Databases and FSQL to the Management of Rural Accommodation. Tourism Management.
21. RIBIERO, R., MOREIRA, A., 1999. Intelligent Query Model for Business Characteristics. Computational Intelligence and Applications, Mastorakis, N.E., (Editor).
22. ELMASRI, R., NAVATHE, S., B., 2000. Fundamentals of Database Systems. Addison Wesley.
23. JAMSHIDI, M., 1993. Fuzzy Logic and Control: Software and Hardware Applications. PTR Prentice-Hall, Englewood Cliffs, 16-18, New Jersey.
24. WANG, Li-Xin., 1994. Adaptive Fuzzy Systems and Control: Design and Stability Analysis. PTR Prentice-Hall, Englewood Cliffs, 9-14, New Jersey.
25. BOSC, P., PRADE, H., 1993. An Introduction to The Fuzzy Set and possibility Theory-Based Treatment of Soft Queries and Uncertain or Imprecise Databases. Proceedings of the 2nd Workshop on Uncertainty Management in Information Systems.
26. NAKAJIMA, H., SOGOH, T., ARAO, M., 1993. Fuzzy Database Language and Library: Fuzzy Extension to SQL. Proc. Second IEEE International Conference on Fuzzy Systems, 479-480.
27. MAELANIN, S., A., BENSALD, A., 1998. Fuzzy Data Minig Query Language. Second International Conferance on Knowledge-Based Intelligent Electronic Systems, 336.

28. DURU, N., 1997. Bulanık Mantık Temelli Gürültü Azaltma Sistemi. KOÜ. Fen Bil. Ens. Doktora Tezi, Kocaeli.
29. KARA, M., 2002. Bilgisayar Ağlarında Bulanık Mantık Yardımıyla Çok Yollu Yönlendirme. KOÜ. Fen Bil. Ens. Doktora Tezi, Kocaeli.
30. ASAR, B., 1999. Standart Veri tabanı Sistemlerinde Bulanık Küme Yaklaşımı ile Esnek Sorgulama, İTÜ. Fen Bil. Ens. Yüksek Lisans Tezi, İstanbul.
31. CANTU, M., 1999. Delphi 4 Uygulama Geliştirme Klavuzu, Alfa Yayıncılık, 425-520.



EK-1

Sorgu cümlesinin yazımını kontrol eden prosedüre ait program kodları:

```
procedure syntax (sira:integer; parcastr:string; var kontrol:boolean; islem:array of string);
var
  i,say,k:integer;
  parca,gecici:string;
  A:array [1..4] of string;
  kont:boolean;
begin
  kontrol:=false;
  i:=1; say:=1;
  while i<=length(parcastr) do
  begin
    parca:=parca+parcastr[i];
    if (parcastr[i]=' ') or (i=length(parcastr)) then
      begin A[say]:=parca; say:=say+1; parca:=''; end;
  i:=i+1; end;//while
//burdan sonra syntax kontrol ediliyor
if say=2 then
  kontrol:=false
else
begin
  k:=0; i:=1; kont:=false;
  while i<=say-1 do
  begin
    if i=1 then k:=k+1;
    if i=2 then
      begin gecici:=A[i];
      if A[i]='COK ' then k:=k+1
      else if A[i]='AZ ' then k:=k+1
      else if(A[i]='YAKLASIK ') then k:=k+1;
      if (gecici[1]='$') then
        begin k:=k+1; kont:=true; end;
      If (A[i]='< ')or (A[i]='<= ')or (A[i]='= ')or (A[i]='> ')or (A[i]='>= ') then k:=k+1;
    end; //i=2
    if i=3 then
      begin gecici:=A[i];
      if (gecici[1]='$') then
```

```
begin k:=k+1; kont:=true; end;
if ((A[i]='DEGIL')) and (kont=true) then
    k:=k+1;
if ((A[i]='DEGIL ') and (kont=true) then k:=k+1;
if (gecici[1]!='#) then k:=k+1;
end; //if i=3
if i=4 then
begin
if (a[i]='DEGIL')and (kont=true) then k:=k+1;
if (a[i]='DEGIL ')and (kont=true) then k:=k+1;
end; //if i=4
i:=i+1;
end; //while
```



EK-2

Üyelik derecelerinin hesaplandığı prosedüre ait program kodları:

```
procedure uyderbul (x:real; p1,p2,p3,p4:string; var uyder:real);  
//üçgen ve yamuk üyelik formülüne göre X'in üyelik derecesini bulur.
```

```
var
```

```
    a,b,c,d:integer;
```

```
begin
```

```
uyder:=0;
```

```
    a:=strtoint(p1);
```

```
    b:=strtoint(p2);
```

```
    c:=strtoint(p3);
```

```
    if p4="" then d:=0
```

```
    else d:=strtoint(p4);
```

```
if a>=strtoint(min) then
```

```
begin
```

```
    if (a<=x) and(x<b) then
```

```
        uyder:=(x-a)/(b-a)
```

```
    else
```

```
    begin
```

```
        if d=0 then
```

```
            if (b<x) and(x<=c) then
```

```
                uyder:=(c-x)/(c-b);
```

```
            if c<=strtoint(max) then
```

```
            begin
```

```
                if d<>0 then
```

```
                    if (d<x) and(x<=c) then        uyder:=(c-x)/(c-d)
```

```
            end;// if c<=max
```

```
        end; //else
```

```
end// a>min
```

```
else
```

```
begin
```

```
    if d<>0 then //yamuk üyelik ise
```

```
        if (X>d) and (x<=c) then        uyder:=(c-x)/(c-d);
```

```
    if d=0 then
```

```
        if (X>b) and (x<=c) then        uyder:=(c-x)/(c-b);
```

```
end; //else
```

```
if d<>0 then
```

```
begin
```

```
    if (x=b)or (x=d) then      uyer:=1;
    if (X>b) and (x<d) then    uyer:=1;
end;
if d=0 then
    if (x=b) then             uyer:=1;
end;
```



EK-3

Sorgu cümlesi işletilmeye başladığında çalışan prosedüre ait program kodları:

```
procedure Tfrmsorgu.Button1Click (Sender: TObject);
var
  sorgustr,SQLstr:string;
  kontrol:boolean;
  i,k1,k2,j:integer;kontrol:boolean;
  baglac:array[0..4] of string;
  ed:real;
  gecicied:integer;
  yenidizi:array of array of real;
  kayitsay,y:integer;
  sonuc:array of real;
  ss,g,uygunkisisay:Integer;
  sondizi:array of real;
  qq:integer; SSql,ssql2:string;
  indis:array[1..10]of integer;
begin
  statusbar1.SimpleText :='Query is Runnig';
  if edit6.Text ='' then
    edit6.Text :='0';
  sorgustr:=edit1.text;
  sorguislem(sorgustr,sqlstr,baglac,y);
  k1:=0;k2:=0;
  for i:=1 to y do
    begin
      if sqlstr[i]<>'' then
        begin
          k1:=k1+1;
          syntax(i,sqlstr[i],kontrol,sqlstr);
          if kontrol=true then
            k2:=k2+1;
          end;//if
        end;//for
    end

  if k1=k2 then
    begin
      showmessage('SQL yazim denetimi basarili');
      for i:=1 to y do
        sorguislemci(sqlstr[i],i,y,kayitsay);
      end //k1=k2
    end
  else
    begin  showmessage('Hatali SQL');  exit;edit1.Clear; end;
  setlength(yenidizi,kayitsay,y);
  for i:=0 to kayitsay-1 do
    for j:=0 to y-1 do //baglac
      yenidizi[i,j]:=eslesmeder(uyderdizi[i,j],sqlstr[j+1]);
  setlength(sonuc,kayitsay);
  if y<=2 then
    begin
      for i:=0 to kayitsay-1 do
        begin
          for j:=0 to y-1 do
            begin
              if y=1 then
```

```

        ed:=yenidizi[i,j];
        if baglac[j+1]='VE' then
        if yenidizi[i,j]<=yenidizi[i,j+1] then
            ed:=yenidizi[i,j]
            else ed:=yenidizi[i,j+1];
        if baglac[j+1]='VEYA' then
        if yenidizi[i,j]>=yenidizi[i,j+1] then
            ed:=yenidizi[i,j]
            else ed:=yenidizi[i,j+1];
        end;
        sonuc[i]:=ed;
    end;//for
end;//y=2
setlength(sondizi,kayitsay);
for ss:=1 to kayitsay do
begin
    if sonuc[ss-1]>=strtofloat(edit6.Text) then//treholddan büyük olan listelenir
        begin
            sondizi[ss-1]:=sonuc[ss-1];
        end
        else
            sondizi[ss-1]:=-1;
    end;

if y>2 then
begin
for i:=0 to kayitsay-1 do
begin
    gecicied:=1;
    for j:=0 to y-1 do
    begin
        if baglac[j+1]='VE' then
        begin
            indis[gecicied]:=j+1;
            gecicied:=gecicied+1;
        end;
    end;// for y-1

if (gecicied=y) OR (gecicied=1) then
begin
    for j:=0 to y-1 do
    begin
        if baglac[j+1]='VE' then
        begin
            if yenidizi[i,j]<=yenidizi[i,j+1] then
                ed:=yenidizi[i,j]
                else ed:=yenidizi[i,j+1];
            yenidizi[i,j]:=ed;
            yenidizi[i,j+1]:=ed;
            end;// Baglac VE ise
            if baglac[j+1]='VEYA' then
            begin
                if yenidizi[i,j]>=yenidizi[i,j+1] then
                    ed:=yenidizi[i,j]
                    else ed:=yenidizi[i,j+1];
                yenidizi[i,j]:=ed;
                yenidizi[i,j+1]:=ed;
                end;// Baglac VEYA ise
            end;//for

```



```

end;// sadece VE ve VEYA varsa
if (0<gecicied) and (gecicied<=y-1) then //Hem VE hem VEYA varsa
begin
  j:=1;
  while j<gecicied do
  begin
    if yenedizi[i,indis[j]-1]<=yenedizi[i,indis[j]] then
      ed:=yenedizi[i,indis[j]-1]
    else
      ed:=yenedizi[i,indis[j]];
    yenedizi[i,indis[j]-1]:=ed;
    yenedizi[i,indis[j]]:=ed;
    j:=j+1;
  end;//while VE islemleri bitti Veya baslayacak
  for j:=0 to y-1 do
  begin
    if baglac[j+1]='VEYA' then
      begin
        if yenedizi[i,j]>=yenedizi[i,j+1] then
          ed:=yenedizi[i,j]
        else ed:=yenedizi[i,j+1];
        yenedizi[i,j]:=ed;
        yenedizi[i,j+1]:=ed;
      end;// Baglac VEYA ise
    end;
  end; //if Ve yada VEya karisik ise
  sonuc[i]:=yenedizi[i,1];
end;//for kayitsay

setlength(sondizi,kayitsay);
for ss:=1 to kayitsay do
begin
  if sonuc[ss-1]>=strtofloat(edit6.Text) then
  begin
    sondizi[ss-1]:=sonuc[ss-1];
  end
  else
    sondizi[ss-1]:=-1;
  end;
end;//y>2
adotable2.Close ;
adotable2.TableName :=combobox1.Text ;
adotable2.Active :=true;
adotable2.First ;
for ss:=0 to kayitsay-1 do
begin
  adotable2.edit ;
  frmsorgu.ADOTable2.fieldbyname('Sonuc').AsFloat :=sondizi[ss];
  adotable2.UpdateRecord;
  adotable2.Next ;
end;

dbgrid1.Color :=clcream ;
adodataset1.Close;
ssql:='select ';
ssql2:="";
for ss:=0 to listbox2.Items.Count-1 do
  begin
    ssl2:=ssql2+listbox2.Items.Strings[ss];
  end;
end;

```

```
        ssql2:=ssql2+'';  
    end;  
    ssql2:=ssql2+'SONUC';  
    ssql:=ssql+ssql2;  
    ssql:=ssql+' from '+adotable2.TableName+' where sonuc>=Edit6.txt';  
    adoDataSet1.CommandText:=ssql;  
    adodataset1.Active :=true;  
    AdoDataSet1.Open ;  
    datasource1.DataSet :=adodataset1;  
    dbgrid1.DataSource :=datasource1;  
    edit1.ReadOnly :=true;  
    edit7.ReadOnly :=true;  
    statusBar1.SimpleText :='Sorgu Sonuçlari';  
end;
```



KİŞİSEL YAYINLAR ve ESERLER

1. S. İLHAN, N. DURU, 2004. Öğrenci Başarısının Değerlendirilmesinde Kullanılacak Akıllı bir Sorgulama Aracı. Asyu-Inista 2004 Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu. 23-25 Haziran, İstanbul.



ÖZGEÇMİŞ

23 Aralık 1979'da Isparta'da doğdu. İlk öğrenimini Hürriyet İlkokulu'nda, orta öğrenimini Bahçelievler İlköğretim Okulu'nda, Burdur'da tamamladı. 1994 yılında başlayan lise öğrenimini, Burdur Anadolu Öğretmen Lisesi'nde tamamladı. 1997 yılında girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümünden 2001 yılında mezun oldu. 2001 yılı ekim ayından bu yana Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü, Bilgisayar Yazılımı A.B.D'de Araştırma Görevlisi olarak çalışmaktadır.

