

# Bulanık Birliktelik Kurallarının Genetik Algoritmalarla Keşfi

Bilal ALATAŞ\*, Ahmet ARSLAN\*\*

\* Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Fırat Üniversitesi 23119, ELAZIĞ

\*\* Bilgisayar Mühendisliği Bölümü, Mühendislik-Mimarlık Fakültesi, Selçuk Üniversitesi 42031, KONYA

## ÖZET

Birliktelik kurallarının keşfi veri madenciliğinde en çok çalışılan konulardan biridir. Bu çalışmada nitelikleri nicel değerler alabilen veritabanlarında nicel birliktelik kurallarının keşfi için yapay zeka ve yumuşak hesaplama konularından bulanık mantık ve genetik algoritma tabanlı yeni yöntemler geliştirilmiştir. Özellikle genetik algoritmanın ilk aşamasında gelişigüzel üretilen başlangıç popülasyonunun dezavantajlarını gideren etkili üç farklı yöntem daha denenmiş ve elde edilen sonuçlar karşılaştırılmıştır. Önerilen yöntemleri test için veritabanı olarak Fırat Üniversitesi Elektrik-Elektronik Mühendisliği lisans öğrencilerinin ders not kayıtları seçilmiş, kullanışlı ve ilginç kurallar etkili şekilde bulunmuştur.

**Anahtar Kelimeler:** Veri madenciliği, birliktelik kuralları, bulanık kümeler, genetik algoritma performansı.

## Mining of Fuzzy Association Rules with Genetic Algorithms

### ABSTRACT

Association rules mining is one of the most studied subjects in data mining. In this work, new methods based on fuzzy logic and genetic algorithm, topics of soft computing and artificial intelligence, have been developed for quantitative association rules mining from databases that can have quantitative attributes. Especially, three different initial population methods that have been eliminated, the disadvantages of random initial population method in the first step of genetic algorithm have been used and the obtained results have been compared. To test the proposed methods, recordings of the Fırat University Electrical and Electronic Engineering students' class grades have been selected as sample database and useful and interesting rules have effectively been mined.

**Keywords:** Data mining, association rules, fuzzy sets, performance of genetic algorithm.

### 1. GİRİŞ

Veri madenciliği (VM), eldeki büyük miktardaki veriden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin istatistik, makine öğrenmesi, yapay zeka ve örüntü tanıma yöntemleri kullanılarak çıkartılması süreci olarak tanımlanmaktadır.

VM'de en çok kullanılan tekniklerden birisi birliktelik kurallarıdır. Bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın almaya eğilimli olduğunun belirlenmesi, müşteriye daha fazla ürünün satılmasını sağlama yollarından biridir. Satın alma eğilimlerinin tanımlanmasını sağlayan birliktelik kuralları pazarlama amaçlı olarak pazar sepeti analizi adı altında VM' de yaygın olarak kullanılmıştır. Bununla birlikte bu teknikler, tıp, finans, mühendislik, web, telekomünikasyon ve farklı olayların birbirleri ile ilişkili olduğunun belirlenmesi sonucunda değerli bilgi kazanımının söz konusu olduğu ortamlarda da önem taşımaktadır.

Birliktelik kurallarının keşfi için en çok kullanılan algoritma Apriori algoritmasıdır ve bahsedildiği gibi ilk olarak market sepeti verisi üzerinde kullanılmıştır (1). Genel olarak kuralların bulunması iki aşamadan oluşur. İlk aşamada aday nesne kümeleri oluşturulur ve bunların kayıt verilerinde kapsadığı kayıt sayısı bulunur. Aday nesne kümelerinden kayıt verilerindeki kapsadığı kayıt sayısı önceden tanımlı eşik değerden (minimum destek) büyük olanlar yoğun nesne kümesi olarak gösterilir. Önce 1-nesne kümesi içeren nesne kümeleri işleme tabi tutulur. Sadece 2-nesne kümesi içeren yoğun nesne kümeleri, 1- nesne kümesi içeren aday nesne kümelerinden oluşturulur. Bu işlemler bütün nesne kümeleri bulunana kadar devam ettirilir. İkinci aşamada ise ilk aşamada bulunan nesne kümelerinden birliktelik kuralları oluşturulur. Bütün olası birliktelik kuralları her yoğun nesne kümesinin kombinasyonları şeklindedir ve hesaplanan güven değeri önceden tanımlı eşik değerden (minimum güven) büyük olmalıdır. Bu işlemin sonucunda birliktelik kuralları elde edilir.

İlk önerilen birliktelik kural madenciliği algoritmaları sadece ikili değer alabilen yani alındı/alınmadı şeklinde tutulan market sepeti verisi üzerinden çalışmaktaydı. Yani ayrık değerli veriler için önerilmiştir. Ancak günlük hayatta veriler ayrık olabileceği gibi sürekli değerli de olabilir. Bunun için önerilen metodun sürekli değerli verileri de içerecek şekilde genişletilmesi yapılmıştır (2). Burada sürekli değer içeren veri belli parçalara bölünmüş ve bu parçalar üzerinden Apriori algoritması çalıştırılmıştır. Yani sürekli değerler parçalanarak ayrık hale getirilmiştir. Bu durum nitelik sayısını arttırmış ve keskin şekilde veriyi ayırtmıştır. Bunu önlemek için farklı yaklaşımlar önerilmiştir. Bunlar yapay zeka ve yumuşak hesaplama tekniklerinden genetik algoritma (GA) yaklaşımı (3), ve bulanık mantık yaklaşımıdır (4).

Market sepeti verisi ile başlayan birliktelik kuralları keşfi daha sonra genişletilmiş, farklı veriler üzerinde farklı algoritmalarla geliştirilmiştir. Bunlara örnek olarak; negatif birliktelik kuralları (5), nicel birliktelik kuralları (4), çok seviyeli birliktelik kuralları (6-7), zamana bağlı geçici birliktelik kuralları (8), döngüsel birliktelik kuralları (9), dağıtık ve paralel algoritmalarla birliktelik kural keşfi (10-11), çevrim içi birliktelik kurallarını verebiliriz.

Bulanık birliktelik kuralları madenciliği, bulanık küme kavramları kullanılarak birliktelik kurallarının keşfidir. Bulanık kümeler, bir kümenin aralığa üye olan kısımları ve aralığa üye olmayan kısımları arasında yumuşak bir geçiş sağlar. Ayrıca, bulanık birliktelik kuralı dilsel terimler içerdiğinden daha anlaşılardır. Bu çalışmada da bulanık kümeler ve etkili bir arama yöntemi olan GA'ların beraber kullanılması ve özellikle GA'larda çeşitli etkili yöntemlerle birliktelik kurallarının keşfi yapılmıştır.

Bu makalenin organizasyonu şu şekildedir: İkinci bölümde bulanık birliktelik kurallarını bulmak için önerilen yöntem anlatılmıştır. Özellikle, GA'larda gelişigüzel üretilen başlangıç popülasyonunun dezavantajlarını gidermek için kullanılan etkili üç farklı yöntem bu bölümde açıklanmıştır. Üçüncü bölümde simülasyon sonuçları sunulmuştur. Son olarak dördüncü bölümde makale sonuçlandırılmış ve kısaca gelecekte çalışılması düşünülen konulardan bahsedilmiştir.

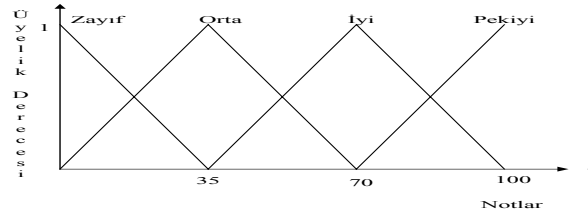
## 2. YÖNTEM

GA, kuralların keşfi için takip edilen iki aşamanın sadece ilkinde, yani yoğun nesne kümelerini bulma aşamasında kullanılmıştır. Çünkü kuralların, bulunan bu yoğun nesne kümelerinden oluşturulması düz bir adımdır.

### 2.1. Niteliklerin Bulanıklaştırılması

Sürekli değerli nitelikler önce bulanıklaştırılır. Daha sonra bu bulanıklaştırılmış değerler kullanılıp GA ile yoğun nesne kümeleri bulunur. Yani bulanıklaştırma için GA yerine kullanıcı tanımlı hazır üyelik fonksiyon-

ları kullanılmıştır. Daha sonra tanıtılacağı gibi kullanılan veri kümesindeki tüm nitelikler sürekli değer almaktadır ve bunların her biri, bir dilbilimsel terimle ilişkilendirilmiştir. Bu çalışmada ele alınan veritabanındaki tüm verilerin alacağı değerler 0-100 aralığında olduğundan hepsi için Şekil 1'de görülen üyelik fonksiyonu kullanılmıştır.



Şekil 1. Nitelikler için dilbilimsel terimlerin tanımı

## 2.2. Yoğun Nesne Kümelerinin Bulunması

Yoğun nesne kümeleri, önerilen GA'lar ile minimum destek eşliği kullanmadan keşfedilecek şekilde düzenlenmiştir. Bunun için önce en iyi yoğun nesne kümesi bulunur, sonra ikinci en iyi bulunur ve istenen sayıdaki en iyi yoğun nesne kümeleri GA'nın bu şekilde birden fazla çalıştırılmasıyla bulunur. Bu amaçla kullanılacak olan GA'nın özellikleri aşağıda verilmiştir.

### 2.2.1. Kodlama

Kromozomlar yoğun nesne kümelerini temsil etmektedir ve kromozomdaki her bir gen de, bir nitelik-değer çiftinden ve bir bayrak değerinden oluşmaktadır. Nitelik-değer çifti  $A_i=D_{ij}$  şeklindedir ve  $A_i$ , i. niteliği,  $D_{ij}$  de  $A_i$ 'nin j. değerini temsil eder. Kodlamayı basitleştirmek için i. niteliğin i. gende kodlandığı bir konumsal kodlama kullanılmıştır. Ayrık değerli nitelikler için  $D_{ij}$  değeri, ilgili niteliğin ayrık değeri olarak gösterilir. Sürekli değerli nitelikler için ise bu, bulanık kümelerdeki dilbilimsel değerlerden oluşur. Kodlamadaki bayrak değeri  $B_j$  de ilgili niteliğin yoğun nesne kümesinde var olup olmadığını göstermektedir. "1" değeri o niteliğin varlığını, "0" ise yokluğunu göstermektedir. Kromozom kodlaması Şekil 2'de gösterilmiştir.

Gen <sub>1</sub>		Gen <sub>2</sub>		...	...	Gen <sub>m</sub>	
$B_1$	$D_{1j}$	$B_2$	$D_{2j}$	...	...	$B_m$	$D_{mj}$

Şekil 2. Kodlama

### 2.2.2. Başlangıç Popülasyonu

Genetik süreç içerisinde başlangıç popülasyonu gelişigüzel üretildiğinden, her zaman çözüme ulaşmak yerine bazen yerel çözümde kalınırken bazen de çözümden uzaklaşma durumu ile karşılaşılabilir. Çünkü üretilen popülasyon gelişigüzel bir yapıya sahiptir ve çözümden uzakta bir araya toplanmış kromozomlardan oluşabilir. Hatta, bütün kromozomları bu özelliğe sahip olabilir. Bu durumda çözüme ulaşmak çok zor olabilir ve fazla zaman alabilir. Bu nedenle farklı yöntemler ele alınarak başlangıç popülasyonunun bu dezavantajlı durumu giderilmelidir. Başlangıç popülasyonu için

GA’larda kullanılan gelişigüzel başlangıç popülasyonu üretmenin dışında farklı yöntemler de kullanılmıştır. Bu yöntemler aşağıda, sırayla açıklanmıştır.

#### a- Düzenli Popülasyon

Bu yöntemin bir  $r$  parametresi vardır ve  $r=1$  durumunda bir kromozom gelişigüzel üretilir, bu kromozomun tümleyeni alınır ve bu tümleyen kromozom da popülasyon içerisinde bir kromozom olarak yerini alır.  $r=2$  olduğu durumda popülasyon biraz daha düzenli hale gelmektedir. Bu durumda gelişigüzel üretilen kromozomun iki parçaya bölünür. İlk olarak sağ yarının tümleyeni ile sol yarının birbirine eklenmesi ile bir kromozom elde edilir. Ondan sonra sol yarının tümleyeni ile sağ yarının birbirine eklenmesi ile başka bir kromozom elde edilir. Son olarak gelişigüzel üretilen kromozomun tümleyeni alınarak başka bir kromozom elde edilir. Kısaca gelişigüzel olarak üretilen bir kromozomdan 3 tane daha kromozom elde edilmektedir (12-14). İlk seçilen birey  $\vec{x}$  ise  $\vec{y}, \vec{z}, \vec{t}$  türetilen bireyler

$$\begin{aligned}\vec{x} &= (x_1, x_2, \dots, x_n), \vec{y} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) \\ \vec{z} &= (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n, x_{n+1}, \dots, x_n) \\ \vec{t} &= (x_1, x_2, \dots, x_n, \bar{x}_{n+1}, \dots, \bar{x}_n)\end{aligned}$$

şeklinde temsil edilir. İndislerde görülen  $n/2$  ifadesinde bölme işleminin tam kısmı alınmıştır.

$$(2^m - 1) \times \phi + \phi = \phi \times 2^m \quad (1)$$

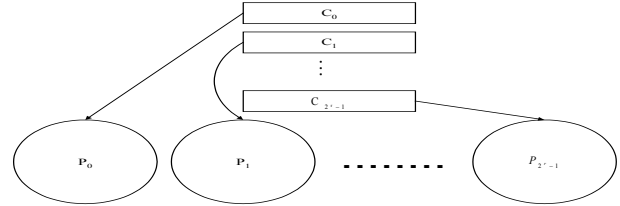
Tümleme işlemi kodlamadaki genin bayrakları için 1’den 0’a ve 0’dan 1’e geçiş olarak tanımlanırken, değerlerin tümlenmesi işlemi için *zayıfın* tümleyeni *iyi*, *ortanın* ise *pekiyi* seçilmiştir.

#### b- Adalı Düzenli Popülasyon

Düzenli popülasyonda gelişigüzel üretilen kromozom  $r$  değerine bağlı olarak parçalara ayrılıp daha sonra bütün kromozomun ve parçaların belli bir düzen içerisinde tümleyenleri alınarak yeni kromozomlar türetilmektedir. Adalı düzenli popülasyonda ise, gelişigüzel üretilen kromozomlar bir adaya atılırken, bu kromozomların tümleyenleri başka bir adaya atılır; ilgili parçaların tümlenmesiyle oluşturulan kromozomlar da başka bir adaya atılır ve bu şekilde  $2^r$  tane ada oluşturulur (15-16). Şekil 3’te gelişigüzel üretilen bir kromozomdan adalara kromozomların türetilmesini ve gelişigüzel kromozomların adalara eklenişi görülmektedir.

Gelişigüzel üretilen kromozom  $C_0$  olmak üzere  $C_1, C_2, \dots, C_{2^r-1}$  tane kromozom  $C_0$  kromozomundan türetilir. Bu kromozomlar Şekil 3’te görüldüğü gibi her kromozom kendisi ile aynı endekse sahip adaya yerleştirilir. Bu şekilde gelişigüzel üretilen bir kromozom durumu gösterilmiştir. Her gelişigüzel kromozom üretildi-

ğinde aynı durum söz konusudur. Dikkat edildiğinde görülecek bir gerçek vardır; o da eğer gelişigüzel üretilen bütün  $C_0$  kromozomları birbirinden farklı ise adaların hepsinde bulunan kromozomlar birbirinden farklıdır. Eğer  $C_0$  kromozomların hepsi aynı ise, bu durumda sadece her adadaki kromozomlar birbirinin aynısı olacak ve adalar arasında kromozom benzerliği olmayacaktır. Adalı düzenli popülasyonun bu özelliği, gelişigüzel popülasyondan daha iyidir. Çünkü bu adalı popülasyonlar gelişigüzel olarak inşa edilecek olursa, popülasyondaki bütün kromozomların aynı olma ihtimali vardır.



Şekil 3. Adalı düzenli popülasyonda kromozomların yerleşimi

Adalı popülasyonun diğer önemli bir özelliği de paralel veya dağıtık mimariler üzerinde kullanılmaya çok müsait bir yapıya sahip olmasıdır. Daha önce vurgulandığı gibi adalar arasında başlangıçta kromozom benzerliği olmadığından her makine farklı bir popülasyon ile aynı problemi çözebilecektir.

#### c- Genelleştirilmiş Düzenli Popülasyon

Bu yöntemle GA’lardaki başlangıç popülasyonu üretme aşamasındaki gelişigüzellik tamamen kaldırılmıştır. Kromozomdaki genlerin sınırları bilindiğinde tüm genleri üst sınırdaki ve tüm genleri alt sınırdaki olan iki kromozom üretilir. Bu kromozomların  $C_0$  ve  $C_1$  olduğunu varsayalım.  $C_1, C_0$ ’ın tümleyeni olarak düşünülür ve diğer kromozomlar da bu tümleme esasına göre üretilir.

Düzenli popülasyonda bölme parametresi  $r$  olduğunda gelişigüzel üretilen bir kromozomdan  $2^r-1$  yeni kromozom üretilmektedir. Basitlik için ikili kodlamaya göre anlatırsak, bu yöntemde eğer popülasyon sayısı sabitse,  $2^r-2$  tane yeni kromozom üretilir. Popülasyon sayısı  $|P|$  olsun. O zaman  $2^r \leq |P|$  eşitliğini sağlayan en büyük  $r$  değeri bulunur. Sonra,  $r$  bir azaltılır ve  $2^r-2$  tane yeni kromozom bulunan  $r$  değerine göre üretilir. Bu işlem kalan kromozom sayısı ikiye ulaşmaya kadar yani,  $r$  değeri 2 olana kadar devam eder. Son aşamada istenenden daha fazla sayıda kromozom üretilmiş olabilir. Bu durumda popülasyon boyutunu sabit tutmak için bunlardan sadece uygun sayıda seçilir ve başlangıç popülasyonuna eklenir. Daha ayrıntılı bilgi için referanslar bölümündeki (17)’de gösterilen çalışmaya bakılabilir.

#### 2.2.3. Uygunluk Fonksiyonu

Yoğun nesne kümelerini bulacağımız için GA’daki kromozomların veritabanında sık olarak geçen

birlikteliklerin ne kadar sıklıkla geçtiğini gösteren bir uygunluğa sahip olmaları gerekmektedir. Bunun için uygunluk olarak aşağıdaki ifade seçilmiştir:

$$\text{Uygunluk} = \alpha \times \text{kapsama} + \beta \times \text{nitelik sayısı} - \gamma \times \text{işaret} \quad (2)$$

*kapsama*, veritabanında kromozomdaki ilgili bayrak ve değerlerin içerdiği tüm kayıtlar için üyelik değerlerinin toplamıdır. Üyelik derecesi bir kayıt için

$$\min_{i=1}^z \mu_i \quad (3)$$

olur. Burada  $\mu_i$ , i. nitelik için derece; z, nitelik sayısı; min ise minimum operatördür. i'nin aldığı değerler direk olarak bayrak değerlerinin ilgili konumundaki değerlerdir. Bir kromozomun herhangi bir kayıttan alacağı *kapsama* değeri, kaydın ilgili nitelik değerlerinin üyelik derecelerinin min operatörüne tabi tutulmasıyla elde edilecek değerdir. Mesela bir kromozomun şu iki genden oluştuğunu düşünelim: (*Türk\_Dili=Orta*) VE (*İngilizce=İyi*). Verilen bir kaydın *Türk\_Dili* niteliği için 38, *İngilizce* niteliği için 69 değerlerine sahip olduğunu varsayalım. Aynı zamanda *Türk\_Dili*'nin *Orta* dilbilimsel terimi için 0.4, *İngilizce*'nin *İyi* dilbilimsel terimi için 0.6 değerini verdiğini varsayalım. O zaman bu kaydın bu kromozoma vereceği kapsama değeri  $\min(0.4, 0.6)=0.4$  olur.

*nitelik sayısı* terimi, kromozomdaki nitelik sayısıdır. Yani kromozomda bayrak değeri 1 olan gen sayısıdır. Birliktelik için en az iki bayrağın 1 olması gerekmektedir. Tek 1 olan kromozomların uygunluğu çok küçük bir değere atanır. Bu şekilde kromozomun hayatta kalması engellenir. Bu terim yoğun nesne kümesinin boyutu anlamına gelir ve ne kadar büyüğe o kadar nesnenin sık olarak veritabanı içinde yer aldığını gösterir. Uygunluğun değerini artırıcı yönde etki etmektedir.

*işaretli* terimi ise, GA'nın daha sonraki çalışmalarda sürekli aynı en iyi yoğun nesne kümesini bulmasını önlemek için kullanılan bir terimdir. Bilindiği gibi GA'daki kromozomlar sürekli en iyi aynı noktaya doğru ilerler. Burada ise amaç farklı en iyi yoğun nesnelere bulmaktır, yani tek bir sonuç yoktur. GA'nın ilk çalışmasından sonra kapsanan ilgili nitelik değerleri işaretlenir ve daha sonraki en iyilerin bulunması aşamasında, bunları içeren kromozomların uygunluk değerine negatif yönde etki etmektedir.

$\alpha$ ,  $\beta$  ve  $\gamma$  ise kullanıcı tanımlı katsayılarıdır. En etkili terim *kapsama* olacak şekilde ayarlanabilir. Programda  $\alpha$  ve  $\gamma$  1,  $\beta$  ise 0.001 olarak seçilmiştir.

### 2.2.4. Genetik Operatörler

Seçim için rulet tekerleği kullanılmış ve elitist strateji izlenmiştir. Çaprazlama olarak iki noktali

çaprazlama kullanılmıştır ve oranı %50 seçilmiştir. Mutasyon oranı da %50 seçilmiş, hem bayraklara hem de değerlere ayrı ayrı uygulanmıştır.

Yoğun nesne kümelerini bulan GA'nın adımları Şekil 4'te gösterilmiştir. Burada N, istenen yoğun nesne küme sayısıdır.

```

Yoğun_Nesne_Küme_Sayısı = 0
While (Yoğun_Nesne_Küme_Sayısı < N) do
  Nesil = 0
  İlk_Nesil P(Nesil)'i üret
  While (Nesil < NESILSAYISI) do
    P(Nesil)'i işle
    P(Nesil + 1) = P(Nesil)'in kromozomlarından seç
    P(Nesil + 1)'i çaprazlamayla tamamla
    P(Nesil + 1)'de mutasyon yap
    Nesil ++
  End_While
  I[Yoğun_Nesne_Küme_Sayısı] = P(Nesil)'in en iyisini seç
  I[Yoğun_Nesne_Küme_Sayısı] tarafından kapsanan kayıtlara ceza ver
  Yoğun_Nesne_Küme_Sayısı ++
End_While

```

Şekil 4. Yoğun nesne kümelerini bulan GA'nın adımları

### 2.3. Kuralların Bulunması

GA ile bulunan yoğun nesne kümelerinin her biri birçok ata kısım ve sonuç kısım oluşturabilir. Eğer ata kısım ve sonuç kısmı yeterince desteğe sahipse ve kural yüksek güvene (ve yüksek korelasyon değerine) sahipse, bu kural ilginç sayılır. Bulanık birliktelik kuralı için şu form kullanılır:

Eğer  $X = \{x_1, x_2, \dots, x_p\}$   $A = \{f_1, f_2, \dots, f_p\}$  ise o zaman  $Y = \{y_1, y_2, \dots, y_q\}$   $B = \{g_1, g_2, \dots, g_q\}$  olur. Burada  $f_i \in \{x_i$  niteliğine bağlı bulanık kümeler}  $g_j \in \{y_j$  niteliğine bağlı bulanık kümeler}'dir ve X, Y nesne kümeleridir ve birbiriyle ortak nitelikleri yoktur. A ve B, X ve Y'de ilgili niteliklerle ilişkili bulanık kümeleri içerir. İkili birliktelik kuralında olduğu gibi "X = A" kuralın atası "Y = B" ise kuralın sonucu olarak adlandırılır.

Eğer bir  $\langle X, A \rangle$  yoğun nesne kümesi elde edersek, "Eğer X = A ise o zaman Y = B'dir" şeklinde bulanık birliktelik kuralları elde etmek isteriz. Burada  $X \subset Z$ ,  $Y = Z - X$ ,  $A \subset C$  ve  $B = C - A$ 'dır. Yoğun nesne kümesi elde etmişsek bunun alt kümelerinin de yoğun olduğunu biliyoruz demektir. Bulanık güven değeri kayıtlar tarafından verilen desteğin derecesinin bir ölçüsü olduğundan güveni, üretilen bulanık birliktelik kuralının ilginçliğini tahmin etmede bize yardım etmesi için kullanırız. Güven için  $\langle Z, C \rangle$ 'nin bulanık desteğini  $\langle X, A \rangle$ 'nin bulanık desteğine böleriz. Bulanık destek ve bulanık güven değerleri (18)'de önerilen şekilde kullanılmıştır.

### 3. SİMÜLASYON SONUÇLARI

Bu bölümde, önce kullanılan veri ve bu veride yapılan ön işlemler anlatılmış, sonra önerilen GA'ların bulunduğu sonuçlar verilmiştir. Önerilen yöntemler için Delphi 5.0 ile SQL sunucu kullanılmıştır.

#### 3.1. Veri Kümesi ve Yapılan Ön İşlemler

Veri olarak, Fırat Üniversitesi Elektrik-Elektronik bölümü lisans öğrencilerinin ders notları kullanılmıştır. Derslerle ilgili ayrıntılı bilgiler <http://www.firat.edu.tr/> adresinden alınabilir. Program ortak olarak alınan dersler üzerinde çalıştırılmıştır. Kayıt dondurma, öğrencilerin ayrılması, seçmeli dersler, derslerin değiştirilmesi vb. nedenlerden dolayı çoğu öğrenci tarafından alınmayan ders problemini ortadan kaldırmak için kurallar, dönem dönem çıkarılmıştır. Yani birinci sınıfın birinci dönem kuralları, üçüncü sınıfın ikinci dönem kuralları gibi.

Ders notu olarak geçme notu kullanılmıştır. Nitelikler, dersler; değerler de vize ve final notlarından hesaplanan geçme notudur. Tüm nitelikler 0-100 arasında sayısal değer alabilir. Bu değerlerin bulanık kümelerden aldığı değerler de ilgili alana yazılmış ve bu değerler üzerinden algoritma çalıştırılmıştır.

Her dönemdeki öğrenci kayıt sayısı da Tablo 1'de gösterilmiştir.

Tablo 2. Yoğun nesne kümeleri

Dönem	En İyi ve İkinci En İyi	Yoğun Nesne Kümeleri	Uygunluk Değeri	Gen. Düz. Pop. İterasyon Sayısı	Adalı Düz. Pop. İterasyon Sayısı	Düz. Pop. İterasyon Sayısı	Gelişigüzel Pop. İterasyon Sayısı
1	1	TRD109=iyi VE ENF101=iyi	174.2425	19	27	34	43
	2	YDİ107=iyi VE ENF101=iyi	129.6119	17	21	29	37
2	1	TRD110=iyi VE KİM106=iyi	141.6002	21	25	31	33
	2	FİZ112=iyi VE TRD110=iyi	123.8657	21	24	30	36
3	1	AİT209=iyi VE MAT265=iyi	150.9524	25	29	33	42
	2	YDİ207=iyi VE MMÜ207=iyi	119.934	19	23	30	39
4	1	AİT210=iyi VE YDİ208=iyi	66.4391	17	22	31	38
	2	EMÜ210=iyi VE EMÜ232=iyi	59.397	17	21	29	36
5	1	EMÜ351=iyi VE EMÜ333=iyi	46.1855	25	30	39	47
	2	EMÜ313=iyi VE EMÜ331=iyi	45.984	32	37	48	56
6	1	EMÜ312=iyi VE EMÜ316=iyi	51.9311	17	22	30	38
	2	EMÜ314=iyi VE EMÜ332=iyi	43.489	23	29	37	45
7	1	EMÜ451=iyi VE EMÜ413=iyi	98.472	17	23	29	36
	2	EMÜ425=iyi VE EMÜ413=iyi	73.526	17	22	31	35
8	1	EMÜ408=iyi VE KİM300=iyi	86.2023	19	25	36	44
	2	EMÜ400=iyi VE EMÜ422=iyi	77.987	17	23	31	36

Tablo 1. Her dönemdeki kayıt sayısı

Dönem	Kayıt Sayısı
1. Sınıf 1. Dönem	279
1. Sınıf 2. Dönem	272
2. Sınıf 1. Dönem	278
2. Sınıf 2. Dönem	150
3. Sınıf 1. Dönem	170
3. Sınıf 2. Dönem	166
4. Sınıf 1. Dönem	157
4. Sınıf 2. Dönem	161

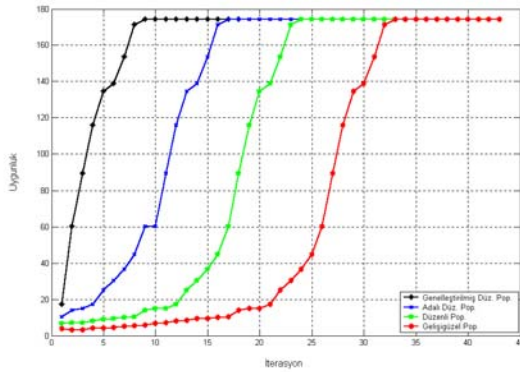
#### 3.2. Yoğun Nesne Kümelerinin Bulunması

Sekiz dönem için farklı başlangıç popülasyonu üretme yöntemi kullanan GA ile bulunan en iyi ve ikinci en iyi yoğun nesne kümeleri karşılaştırmalı olarak Tablo 2'de görülmektedir. Tüm yöntemler için başlangıç popülasyonu 48 seçilmiş, en iyi uygunluk art arda 10 iterasyon boyunca değişmediği anda bitime gidilmiş ve bu şartın sağlandığı iterasyon sayısı, kullanılan farklı GA'lara göre son dört sütunda gösterilmiştir. İlk sütun dönemi göstermektedir. İkinci sütundaki "1" ve "2", en iyi ve ikinci en iyi yoğun nesne kümesi anlamındadır. Üçüncü sütun bulunan yoğun nesne kümelerini, dördüncü sütun ise hesaplanan uygunluk değerlerini göstermektedir.

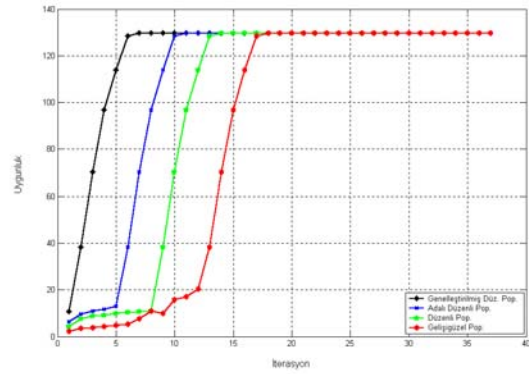
Şekil 5'te, birinci dönem için en iyi yoğun nesne kümesini minimum destek kullanmadan bulan GA'ların iterasyon sayısına göre uygunluk değerleri grafiği gösterilmiştir. Farklı başlangıç popülasyonu üretme tekniği kullanan GA'ların bulduğu sonuçlar toplu olarak görülmektedir. Burada her iterasyonda, popülasyondaki bireylerin uygunluk değerlerinin ortalaması alınmıştır. Değerler, GA 10 kez çalıştırılıp bulunan bu ortalama uygunlukların ortalaması alınarak elde edilmiştir. Birinci dönemin ikinci en iyi yoğun nesne kümesini bulan GA için bulunan değerler de Şekil 6'da gösterilmiştir.

Birinci sınıfın ilk döneminin en iyi ve ikinci en iyi yoğun nesne kümesini bulmaya çalışan, farklı başlangıç popülasyonu üretme yöntemi kullanan GA'nın her iterasyondaki en iyi kromozomlarının 10 çalışma boyunca ortalaması alınmasıyla elde edilen sonuçlar da sırayla Şekil 7 ve Şekil 8'de gösterilmiştir.

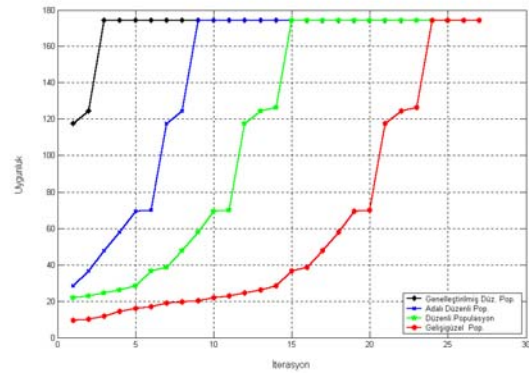
Bu çalışmada, başlangıç popülasyonunda yer alan kromozomların kromozom uzayına düzgün bir şekilde dağıtılmasını sağlayan yöntemler kullanılmıştır. İlk yöntem olan düzenli popülasyon yöntemi ile başlangıç popülasyonunun uzaya düzgün dağıtılması amaçlanmıştır, ancak bu yöntemde de kısmen gelişigüzellik bulunmaktadır. Adalı düzenli başlangıç popülasyonunun bu yöntemde daha iyi performans gösterdiği görülmektedir. Başlangıç popülasyonun gelişigüzel-liğini tamamen ortadan kaldıran genelleştirilmiş düzenli popülasyon yönteminin ise GA için en uygun başlangıç popülasyonu oluşturma yöntemi olduğu açık şekilde görülmektedir.



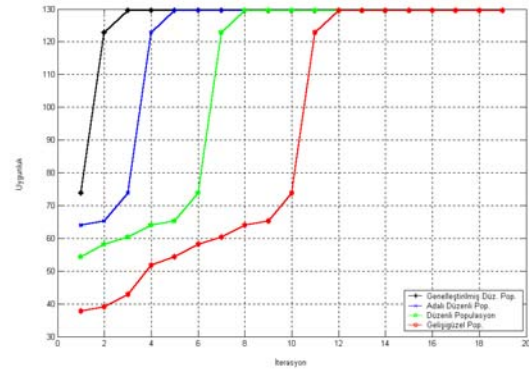
Şekil 5. Birinci dönemin en iyi yoğun nesne kümesi için iterasyon-ortalama uygunluk



Şekil 6. Birinci dönemin ikinci en iyi yoğun nesne kümesi için iterasyon-ortalama uygunluk



Şekil 7. Birinci dönemin en iyi yoğun nesne kümesi için iterasyon-en iyi uygunluk



Şekil 8. Birinci dönemin ikinci en iyi yoğun nesne kümesi için iterasyon-en iyi uygunluk

### 3.3. Kuralların Bulunması

GA ile bulunan yoğun nesne kümeleri için bulanık güven değerleri hesaplanmıştır. Bulunan sonuçlar Tablo 3'te gösterilmiştir. Burada, mesela birinci dönemin ikinci en iyi yoğun nesne kümesinden güven değeri %70 verildiğinde sadece “YDİ107=iyi  $\Rightarrow$  ENF101=iyi” kuralı oluşturulur, “ENF101=iyi  $\Rightarrow$  YDİ107=iyi” kuralı, yeterli güven değerini sağlayamadığından oluşturulamaz.

Tablo 3. Bulunan kurallar ve güven değerleri

Dönem	En İyi ve İkinci En İyi	Kurallar	Güven
1	1	TRD109=iyi $\Rightarrow$ ENF101=iyi	0.7426
		ENF101=iyi $\Rightarrow$ TRD109=iyi	0.7541
	2	YDİ107=iyi $\Rightarrow$ ENF101=iyi	0.7373
		ENF101=iyi $\Rightarrow$ YDİ107=iyi	0.5633
2	1	TRD110=iyi $\Rightarrow$ KİM106=iyi	0.65
		KİM106=iyi $\Rightarrow$ TRD110=iyi	0.6857
	2	FİZ112=iyi $\Rightarrow$ TRD110=iyi	0.683
		TRD110=iyi $\Rightarrow$ FİZ112=iyi	0.589
3	1	AİT209=iyi $\Rightarrow$ MAT265=iyi	0.702
		MAT265=iyi $\Rightarrow$ AİT209=iyi	0.6778
	2	YDİ207=iyi $\Rightarrow$ MMÜ207=iyi	0.6694
		MMÜ207=iyi $\Rightarrow$ YDİ207=iyi	0.5576
4	1	AİT210=iyi $\Rightarrow$ YDİ208=iyi	0.5499
		YDİ208=iyi $\Rightarrow$ AİT210=iyi	0.6506
	2	EMÜ210=iyi $\Rightarrow$ EMÜ232=iyi	0.659
		EMÜ232=iyi $\Rightarrow$ EMÜ210=iyi	0.6944
5	1	EMÜ351=iyi $\Rightarrow$ EMÜ333=iyi	0.6557
		EMÜ333=iyi $\Rightarrow$ EMÜ351=iyi	0.5713
	2	EMÜ313=iyi $\Rightarrow$ EMÜ331=iyi	0.726
		EMÜ331=iyi $\Rightarrow$ EMÜ313=iyi	0.7119
6	1	EMÜ312=iyi $\Rightarrow$ EMÜ316=iyi	0.5543
		EMÜ316=iyi $\Rightarrow$ EMÜ312=iyi	0.6608
	2	EMÜ314=iyi $\Rightarrow$ EMÜ332=iyi	0.6722
		EMÜ332=iyi $\Rightarrow$ EMÜ314=iyi	0.6913
7	1	EMÜ451=iyi $\Rightarrow$ EMÜ413=iyi	0.5724
		EMÜ413=iyi $\Rightarrow$ EMÜ451=iyi	0.6908
	2	EMÜ425=iyi $\Rightarrow$ EMÜ413=iyi	0.708
		EMÜ413=iyi $\Rightarrow$ EMÜ425=iyi	0.6908
8	1	EMÜ408=iyi $\Rightarrow$ KİM300=iyi	0.6818
		KİM300=iyi $\Rightarrow$ EMÜ408=iyi	0.6543
	2	EMÜ400=iyi $\Rightarrow$ EMÜ422=iyi	0.713
		EMÜ422=iyi $\Rightarrow$ EMÜ400=iyi	0.6701

### 4. SONUÇ

Bu çalışmada; bilime, mühendisliğe, tıp sahasına, eğitime ve bilhassa ticari hayata yeni uygulamalar kazandıran bir disiplin olarak ortaya çıkmaya başlayan VM için yapay zeka ve yumuşak hesaplama metodlarından bulanık mantık ve GA tabanlı yeni yöntemler geliştirilmiştir.

Yöntemler insana özgü tecrübe ile öğrenmeyi kolayca modelleyen ve belirsiz kavramları bile matematiksel olarak ifade edebilen bulanık mantık ve doğal seçim ilkelerine dayanan bir arama ve optimizasyon algoritması olan GA'yı içermektedir. Bu çalışmada GA'da bazen rastlanan yerel çözümden kalma ve çözümden uzaklaşma durumlarının önlenmesi ve gerçek çözüme hızlı şekilde ulaşılabilmesi için başlangıç popülasyonu üretme aşamasında üç farklı yöntem denenmiş ve en etkilisinin geliştirilmiş düzenli popülasyon adı verilen yöntem olduğu görülmüştür.

Çalışmada veritabanı olarak tüm niteliklerin nicel değerler aldığı Fırat Üniversitesi Elektrik-Elektronik Mühendisliği lisans öğrencilerinin not kayıtları seçilmiş ve önerilen yöntemlerin performansı incelenmiştir. Nicel değerler alan nitelikler bulanık kümelerle temsil edilerek bulanıklaştırılmış, daha sonra bulanıklaştırılan bu nitelikler üzerinden birliktelik kurallarını bulmak için GA çalıştırılmıştır.

Yeni veri tiplerinin madenciliği için yeni algoritma, teknik ve sistemler sürekli geliştirilmektedir. İlerleyen çalışmalarda yeni yapay zeka ve yumuşak hesaplama teknikleri kullanarak özellikle birliktelik kuralları, ilginç tahmin kuralları, sınıflama ve kümeleme kurallarının keşfi düşünülmektedir. Birliktelik kural madenciliğinden özellikle çevrim içi, negatif, nicel, geliştirilmiş ve ağırlıklı birliktelik kurallarının keşfi için önerilen yöntemler kullanılacaktır. Bunların birbiriyle karşılaştırılarak performans değerlendirilmesi de yapılacaktır. Algoritmaların paralel ve dağıtık versiyonlarıyla ve GA'nın genetik operatörlerindeki ince ayarlamalarla da daha hızlı şekilde kuralların bulunması hedeflenmektedir.

### 5. KAYNAKLAR

1. Agrawal, R., Imielinski, T., Swami, A., Mining Association Rules Between Sets of Items in Large Databases, ACM SIGMOD Conf. Management of Data, 207-216, 1993.
2. Srikant, R. and Agrawal, R., Mining Quantitative Association Rules in Large Relational Tables, Proceedings of 25<sup>th</sup> ACM SIGMOD International Conference on Management of Data, 1-12, 1996.
3. 10. Vázquez, J.M., Macías, J.L.A., Santos, J.C.R., Discovering Numeric Association Rules via Evolutionary Algorithm, PAKDD 2002, 40-51, 2002.
4. Zhang, W., Mining Fuzzy Quantitative Association Rules, 11<sup>th</sup> IEEE International Conference on Tools with Artificial Intelligence, 99-102, 1999.

5. Wu, X., Zhang, C., Zhang, S., Mining Both Positive and Negative Association Rules, The 9<sup>th</sup> International Conference on Machine Learning (ICML-2002), 658-665, 2002.
6. Han, J., Fu, Y., Mining Multiple-Level Association Rules in Large Databases, IEEE Transactions on Knowledge and Data Engineering, vol:11 no:5, 798-805, 1999.
7. Srikant, R. and Agrawal, R., Mining Generalized Association Rules, Proc. of VLDB95, 407-419, 1995.
8. Li, Y., Ning, P., Wang, X.S., Jajodia, S., Discovering Calendar-Based Temporal Association Rules, Eighth International Symposium on Temporal Representation and Reasoning TIME'01, 111-118, 2001.
9. Ozden, B., Ramaswamy, S., Silberschatz, A., 1998, Cyclic Association Rules. Proc. 14<sup>th</sup> Int. Conf. on Data Engineering, 412-421, 2001.
10. Zaki M.J., et al., Parallel Algorithms for Fast Discovery of Association Rules, Data Mining and Knowledge Discovery: An Int'l J., vol:1, no:4, 343-373, 1997.
11. Cheung D., et al., A Fast Distributed Algorithm for Mining Association Rules, Proc. 4<sup>th</sup> Int'l Conf. Parallel and Distributed Information Systems, IEEE Computer Soc. Press, Los Alamitos, Calif., 31-42, 1996.
12. Karcı, A., Arslan, A., Uniform Population in Genetic Algorithms, Journal of Electrical and Electronics, Istanbul Univ., vol:2, no:2, 495-504, 2002.
13. Alataş, B., Arslan, A., Association Rule Mining with Genetic Algorithms, Mühendislik Bilimleri Genç Araştırmacılar 1. Kongresi MBGAK'2003, İstanbul, 81-88, 2003.
14. Alataş, B., Karcı, A., Genetik Sürecin Düzenlilik Operatörüyle Global Çözüme Doğru Harekete Zorlanması, ELECO'2002, Tübitak / Bursa, 364-368, 2002.
15. Alataş, B., Arslan, A., Mining of Interesting Prediction Rules with Uniform Two-Level Genetic Algorithm, International Journal of Computational Intelligence (IJCI) Proceedings TAINN'2003, vol:1 no:1, 65-70, 2003.
16. Gündoğan K.K., Alataş, B., Karcı, A., Tatar, Y., Comprehensible Classification Rule Mining with Two-Level Genetic Algorithm, 2<sup>nd</sup> FAE International Symposium, TRNC, 373-377, 2002.
17. Gündoğan K.K., Alataş, B., Karcı, A., Generalized Uniform Population in Genetic Algorithm for the Task of Comprehensible Classification Rule Mining, Mühendislik Bilimleri Genç Araştırmacılar 1. Kongresi MBGAK'2003, İstanbul, 97-104, 2003.
18. Gyenesei, A., A Fuzzy Approach for Mining Quantitative Association Rules, Univ. Turku, Dept. Comput. Sci., Lemminkisenkatu 14, Finland, TUCS Tech. Rep. 336, 2000.