

**T.C.
TUNCELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YERÇEKİMSEL ARAMA ALGORİTMASIYLA NİCEL BİRLİKTELİK
KURALLARININ KEŞFİ**

YÜKSEK LİSANS TEZİ

Ümit CAN

Anabilim Dalı: Elektrik-Elektronik Mühendisliği

**DANIŞMAN
Doç. Dr. Bilal ALATAŞ**

ŞUBAT-2014

**T.C.
TUNCELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YERÇEKİMSEL ARAMA ALGORİTMASIYLA NİCEL BİRLİKTELİK
KURALLARININ KEŞFİ**

YÜKSEK LİSANS TEZİ

Ümit CAN

(102103104)

**Tezin Enstitüye Verildiği Tarih: 20.02.2014
Tezin Savunulduğu Tarih : 13.03.2014**

Tez Danışmanı : Doç. Dr. Bilal ALATAŞ (T.Ü.)
Diğer Jüri Üyeleri: Yrd. Doç. Dr. Zeki OMAÇ (T.Ü.)
Yrd. Doç. Dr. Ömer ÇELİK (T.Ü.)

ŞUBAT-2014

Ümit CAN tarafından hazırlanan YERÇEKİMSEL ARAMA ALGORİTMASIYLA NİCEL BİRLİKTELİK KURALLARININ KEŞFİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Bilal ALATAŞ

Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği/oy çokluğu ile Elektrik Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. Bu tez, Tunceli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

Başkan : Yrd. Doç. Dr. Zeki OMAÇ (T.Ü.)

Üye : Doç. Dr. Bilal ALATAŞ (T.Ü.)

Üye : Yrd. Doç. Dr. Ömer ÇELİK (T.Ü.)

Tarih : 13.03.2014

ÖNSÖZ

Tez çalışmam boyunca bilgisiyle, görüşleriyle ve sabrıyla bana yardımcı olan değerli tez danışmanım Sayın Doç. Dr. Bilal ALATAŞ'a bana sunduğu tüm desteklerden dolayı teşekkür ederim. Ayrıca yardımını benden esirgemeyen araştırma görevlisi Soner KIZILOLUK' a da şükranlarımı sunarım. Buna ek olarak YLTUB013-02 kodlu proje ile çalışmalarına maddi destek sağlayan Tunceli Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne teşekkür ederim.

Ümit CAN

TUNCELİ-2014

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	VI
SUMMARY.....	VII
ŞEKİLLER LİSTESİ.....	VIII
TABLolar LİSTESİ.....	IX
KISALTMALAR.....	X
SEMBOLLER LİSTESİ.....	XI
1. GİRİŞ.....	1
1.1. Çalışmanın Amacı.....	2
1.2. Bu Konuda Yapılan Çalışmalar.....	2
1.3. Optimizasyon.....	5
1.3.1 Sezgisel Algoritmalar.....	7
1.4. Fizik Tabanlı Sezgisel Optimizasyon Algoritmaları.....	9
1.4.1. Elektromanyetizma Benzeri Algoritması.....	9
1.4.2. Merkez Kuvvet Algoritması.....	10
1.4.3. Yüklü Sistem Arama Algoritması.....	10
1.4.4. Yapay Fizik Algoritması.....	11
1.4.5. Akıllı Su Damlaları Algoritması.....	11
1.4.6. Nehir Formasyonu Dinamikleri Algoritması.....	11
1.4.7. Uzay Yerçekimsel Algoritması.....	12
1.4.8. Parçacık Çarpışması Algoritması.....	12
1.4.9. Büyük Patlama Büyük Sıçrama Algoritması.....	13
1.4.10. Büyük Sıçrama Algoritması.....	13
1.4.11. Galaksi Tabanlı Arama Algoritması.....	13
1.4.12. Yerçekimsel Arama Algoritması.....	14
1.4.12.1. Yerçekimi Kanunu.....	14
1.4.12.2. Yerçekimsel Arama Algoritması (YAA).....	17
1.5. Veri Madenciliği.....	23

1.5.1.	Veri Madenciliğinin Uygulandığı Alanlar.....	24
1.5.2.	Veri Ambarları ve OLAP.....	25
1.5.2.1.	OLAP (On-Line Analytical Processing).....	26
1.5.3.	Veri Madenciliği İçin Verilerin Hazırlanması.....	26
1.5.3.1.	Verilerin Temizlenmesi.....	26
1.5.3.2.	Veri Bütünleştirilmesi.....	27
1.5.3.3.	Veri İndirgeme.....	27
1.5.3.4.	Veri Dönüştürme.....	27
1.5.4.	Veri Madenciliği Modelleri.....	28
1.5.4.1.	Değer Tahmini Modeli.....	28
1.5.4.2.	Bağlantı Analizi.....	29
1.5.4.3.	Birliktelik Kuralları.....	29
1.5.4.4.	Örüntü Tanıma.....	29
1.5.4.5.	Ardışık Zaman Örüntüleri.....	29
1.5.4.6.	Kümeleme Analizi.....	30
1.6.	Birliktelik Kuralları.....	30
1.6.1.	Destek ve Güven Kavramları.....	30
1.6.2.	Apriori Algoritması.....	32
2.	MATERYAL ve METOT.....	33
2.1.	Kullanılan Veritabanları.....	34
2.1.1.	Basketball Veritabanı.....	34
2.1.2.	Quake Veritabanı.....	35
2.1.3.	Thyroid Disease (New Thyroid) Veritabanı.....	36
2.1.4.	Stock Prices Veritabanı.....	37
2.2.	YAA ile Geliştirilen Uygulama.....	38
2.2.1.	Başlangıç Popülasyonunun Oluşturulması.....	38
2.2.2.	Her Bir Bireyin Uygunluğunun Hesaplanması.....	40
2.2.3.	G'yi Ve Popülasyonun En İyi ve En Kötüsünü Güncelle.....	41
2.2.4.	Her Bir Birey İçin M ve a Değerlerini Hesapla.....	41
2.2.5.	Hızı ve popülasyonu güncelle.....	44

2.2.6.	Sonlandırma kriteri ile karşılaşırsa programı sonlandır ve en iyi çözümü geri döndür değilse başa geri dön.....	44
3.	BULGULAR.....	45
3.1.	Basketball Veritabanı Sonuçları.....	45
3.1.1.	İlk Parametre Değerleri İçin Sonuçlar.....	45
3.1.2.	İkinci Parametre Değerleri İçin Sonuçlar.....	46
3.2.	Quake Veritabanı Sonuçları.....	47
3.2.1.	İlk Parametre Değerleri İçin Sonuçlar.....	47
3.2.2.	İkinci Parametre Değerleri İçin Sonuçlar.....	48
3.3.	Thyroid Disease (New Thyroid) Veritabanı Sonuçları.....	49
3.3.1.	İlk Parametre Değerleri İçin Sonuçlar.....	49
3.3.2.	İkinci Parametre Değerleri İçin Sonuçlar.....	50
3.4.	Stock Prices Veritabanı Sonuçları.....	51
3.4.1.	İlk Parametre Değerleri İçin Sonuçlar.....	51
3.4.2.	İkinci Parametre Değerleri İçin Sonuçlar.....	52
4.	SONUÇLAR.....	53
5.	TARTIŞMA.....	54
6.	ÖNERİLER.....	55
	KAYNAKLAR.....	
	ÖZGEÇMİŞ.....	

ÖZET

Bir problemin çözümünde optimum sonucu bulma işlemi veya bir başka deyişle en uygun çözümü bulma işlemi optimizasyon olarak adlandırılabilir. Matematiksel olarak bakıldığında belli bir aralıkta verilen değerler için bir fonksiyonun maksimize ya da minimize edilmesi işlemidir. Klasik optimizasyon algoritmaları zamanla karmaşık problemlerin çözümünde yetersiz kalmış ve sezgisel optimizasyon algoritmaları önerilmiştir. Bu algoritmalar oluşturulurken fizik, kimya ve biyoloji gibi birçok alandan esinlenilmiştir. Yerçekimi kanunu, kapalı evren teorisi ve genetik kodlama gibi alanlar bunlardan bazılarıdır.

Bu tez çalışmasında, bu çalışmanın amacı, bu konuda yapılan çalışmalar, optimizasyon, sezgisel optimizasyon, fizik tabanlı algoritmalar, veri madenciliği, birliktelik kuralları gibi konularda genel bilgilendirme yapılmış ve Yerçekimsel Arama Algoritması (YAA) detaylı olarak anlatılmıştır. Daha sonra YAA'nın nasıl bir metotla uygulandığı gösterilmiştir ve kullanılan veritabanları ile bu veritabanlarından elde edilen sonuçlar gösterilmiştir. Bunu takiben tartışma kısmında, bu konu ile ilgili daha önceden yapılan çalışmalar değerlendirilmiş ve bu tez çalışması ile karşılaştırılmıştır. Öneri kısmında ise bu çalışmanın nasıl daha iyi olabileceği konusunda fikirler belirtilmiştir. Bu çalışma sırasında Visual C# programı ile YAA'ya uygun bir program yazılmıştır. Bu program UCI ve Bilkent University Function Approximation Repository veri ambarlarından alınan 4 adet veri tabanında uygulanmış ve elde edilen birliktelik kuralları gösterilmiştir.

Anahtar Kelimeler: Yerçekimsel Arama Algoritması, Veri Madenciliği, Birliktelik Kurallarının Keşfi

SUMMARY

Finding optimum solution of a problem or looking for the best solution is named optimization. Looking in a mathematical aspect for given variables in a specific range optimization is making a function maximize or minimize. The classical optimization algorithms is not efficient in solving complexity problems. Thus, some heuristic optimization algorithms has been proposed. Many scientific field such as physics, chemistry and biology have been utilized for creating this algortihms. Gravitational law, closed universe theorem is some of them.

In this thesis study, general exposition was done about the subjects such as purpose of this thesis study, the studies about the thesis subject, optimization, heuristic optimization, physical based algorithm, data mining, association rules and gravitational search algorithm was taught in detail. Then it was demonstrated that in which and how the gravitational search algorithm was put into practice besides the databases that are used and the results that were received from these databases were also demonstrated. And then, in the argumentation section, the studies that had been done about this subject were evaulated and compared with this thesis study. In the suggestion section, ideas were taken into consideration about how this study could be better? In this study, a program which is convenient with the gravational search algorithm was written in the Visual C# program. This program was put into practice in 4 itemed databases that were received from the dataware houses of UCI and Bilkent Universty Function Approximation Reporsity and these received association rules were demonstrated.

Key Words: Gravitational Search Algorithm, Data Mining, Exploration of Association Rules

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Optimizasyon için matematiksel modeller.....	6
Şekil 1.2. Sezgisel Yöntemler.....	8
Şekil 1.3. Tüm kütleler, diğer kütlelerden etkilenen sonuç kuvvetine doğru hızlanır..	17
Şekil 1.4. YAA'nın genel çalışma prensibi.....	22
Şekil 1.5. Veri madenciliği ve bilgi keşfi süreci.....	24
Şekil 2.1. Basketball veritabanından bir kesit	34
Şekil 2.2. Quake veritabanından bir kesit.....	35
Şekil 2.3. Thyroid Disease (New Thyroid) veritabanından bir kesit.....	36
Şekil 2.4. Stock Prices veritabanından bir kesit.....	37
Şekil 2.5. Örnek birliktelik kural adayı	39
Şekil 2.6. Birliktelik kuralının ifadesi.....	39
Şekil 2.7. Programda oluşturulan gerçek adaylar	40
Şekil 2.8. Uygunluk fonksiyonu	40
Şekil 2.9. $F_{ij}^d(t)$ 'nin programdaki karşılığı.....	42
Şekil 2.10. Kütle ve boyutların gösterilmesi.....	43

TABLolar LİSTESİ

	<u>Sayfa No</u>
Tablo 1.1. Min-max normalleştirilmesi sonucu elde edilen değerler.....	28
Tablo 2.1. Basketball veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler...	35
Tablo 2.2. Quake veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler.....	36
Tablo 2.3. Thyroid Disease (New Thyroid) veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler.....	37
Tablo 2.4. Stock Prices veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler.....	38
Tablo 3.1. Basketball veritabanı için kullanılan ilk parametreler.....	45
Tablo 3.2. Basketball veritabanı ilk parametre sonuçları.....	45
Tablo 3.3. Basketball veritabanı için kullanılan ikinci parametreler.....	46
Tablo 3.4. Basketball veritabanı ikinci parametre sonuçları.....	46
Tablo 3.5. Quake veritabanı için kullanılan ilk parametreler.....	47
Tablo 3.6. Quake veritabanı ilk parametre sonuçları.....	47
Tablo 3.7. Quake veritabanı için kullanılan ikinci parametreler.....	48
Tablo 3.8. Quake veritabanı için kullanılan ikinci parametrele sonuçları.....	48
Tablo 3.9. Thyroid Disease (New Thyroid) veritabanı için kullanılan ilk parametreler.....	49
Tablo 3.10. Thyroid Disease (New Thyroid) veritabanı için kullanılan ilk parametrele sonuçları.....	49
Tablo 3.11. Thyroid Disease (New Thyroid) veritabanı için kullanılan ikinci parametreler.....	50
Tablo 3.12. Thyroid Disease (New Thyroid) veritabanı için kullanılan ikinci parametrele sonuçları.....	50
Tablo 3.13. Stock Prices veritabanı için kullanılan ilk parametreler.....	51
Tablo 3.14. Stock Prices veritabanı için kullanılan ilk parametre sonuçları.....	51
Tablo 3.15. Stock Prices veritabanı için kullanılan ikinciparametreler.....	52
Tablo 3.16. Stock Prices veritabanı için kullanılan ikinci parametre sonuçları.....	52

KISALTMALAR

ASD	: Akıllı Su Damlları Algoritması
BS	: Büyük Sıçrama
BP-BS	: Büyük Patlama Büyük Sıçrama
GTA	: Galaksi Tabanlı Arama Algoritması
KKO	: Karınca Koloni Optimizasyonu
MKA	: Merkez Kuvvet Algoritması
NFD	: Nehir Dinamikleri Formasyonu
OLAP	: On-Line Analytical Processing
PÇA	: Parçacık Çarpışma Algoritması
PSO	: Parçacık Sürü Optimizasyonu
UYA	: Uzay Yerçekimsel Algoritması
YAA	: Yerçekimsel Arama Algoritması
YFA	: Yapay Fizik Algoritması
YSAR	: Yüklü Sistem Arama Algoritması

SEMBOLLER LİSTESİ

a	: Parçacığın ivmesi
$a_i^d(t)$: i bireyinin t anında d 'ninci yöndeki ivmesi
$best(t)$: t anındaki en iyi uygunluk değeri
F	: Yerçekimi kuvvetinin büyüklüğü
$F_i^d(t)$: t anında d boyutunda i bireyine etkiye kuvvet
F_{ij}	: Kütle j 'den kütle i 'ye etki eden yerçekimsel kuvvet
$F_{ij}^d(t)$: t anında j kütlelerinden i kütlelerine etki eden kuvvet
$fit_i(t)$: t anında i bireyinin uygunluk değeri
G	: Yerçekimi kuvveti Sabiti
$G(t)$: t anında yerçekimi sabitidir
$G(t_0)$: t_0 zamanının ilk kozmik kuantum-aralığındaki yerçekimi sabitidir
$Kbest$: En iyi uygunluk değeri ve en büyük kütleli olan ilk K birey grubu
M_a	: Aktif yerçekimsel kütle
M_{aj}	: j nin aktif yerçekimsel kuvveti
M_i	: Eylemsiz yerçekimsel kütle
M_{ii}	: i 'nin eylemsiz yerçekimsel kuvveti
M_p	: Pasif yerçekimsel kütle
M_{pi}	: i nin pasif yerçekimsel kuvveti
M_1	: Birinci parçacığın kütlesi
M_2	: İkinci parçacığın kütlesi
R	: İki parçacık arasındaki mesafedir
$R_{ij}(t)$: i ve j bireyleri arasındaki Öklit mesafesi
$rand_j$: i bireyinin t anında d 'ninci yöndeki ivmesi
$sayı(A,B)$: A ve B ürün gruplarını birlikte içeren ilişki
$v_i^d(t)$: i bireyinin d boyutunda t anındaki hızı
$v_i^d(t + 1)$: i bireyinin d boyutunda $t+1$ anındaki hızı
$worst(t)$: t anındaki en kötü uygunluk değeri
X	: Tablodaki gerçek değer
X^*	: 0 ile 1 arasında dönüşüm sonucu bulunan değeri
X_{max}	: X 'in tablodaki en büyük değeri
X_{min}	: X 'in tablodaki en küçük değeri
$x_i^d(t + 1)$: Tablodaki gerçek değer

1. GİRİŞ

Yıllar boyunca bilim adamları birçok alanda çalışmalar yapmış ve karmaşık problemlerle karşılaşmışlardır. Özellikle günümüz dünyasında teknolojinin giderek gelişmesi ile kullanılan bilgi miktarının oldukça artmış olması sorunları çok boyutlu bir hale getirmiştir. Toplumların ihtiyaçları bununla paralel olarak artmış ve problemlere getirilen çözümün hızı ve kalitesi oldukça önem kazanmıştır. Bu problemlerin optimum çözümü ve bunların olabildiğince hızlı çözümü bilim adamlarının üzerinde uğraştığı bir durumdur. Bu noktada optimizasyon oldukça önemli bir alandır. Problemlerin çözümü için birçok algoritma geliştirilmiştir. Bilim adamları bunu yaparken doğal sistemlere ve bu sistemlerde cereyan eden olaylara yönelmişlerdir çünkü karmaşık zor doğal optimizasyon problemlerinin çözümü tabiatta var olan doğal yöntemlerle çözülmektedir (Karaboğa, 2011). Karın karınca koloni algoritması (Dorigo and Stutzle, 2004) buna örnek verilebilir. Optimizasyon algoritmaları içinde modern sezgisel algoritmalar karmaşık problemleri çözmek için kullanılan gelişmiş yöntemlerdir. Akıllı Su Damlacıkları Algoritması (ASD) (Kamkar vd., 2010) ve bu tez çalışmasında da kullanılan YAA (Rashedi vd., 2009) bu algoritmalara örnektir.

Optimizasyon metotlarının birlikte kullanıldığı bir alan da veri madenciliğidir. Bilgisayarların hayatımıza girmesiyle beraber her yaptığımız işlem sayısal ortamda tutulmaya başlandı. Hastanelerde yaptığımız işlemler ya da bir alışveriş merkezinde aldığımız ürünler gibi milyonlarca verinin tutulduğu veri depoları oluşmuştur. Bu verilerin büyüklüğünün ve karmaşıklığının dışında bu veriler içinde keşfedilmeyi bekleyen birçok yeni bilgi barınmaktadır. Tam da bu nokta da var olanı değil, daha önceden bilinmeyen geçerli ve uygulanabilir bilgilerin geniş veritabanlarından çıkarılıp bulunmasına yarayan veri madenciliği devreye girmiştir (Silahtaroglu, 2008). Günümüzde veri madenciliği gelişen ve oldukça yaygın kullanılan bir dal haline gelmiştir. Veri madenciliği bankacılık, pazarlama, sigortacılık ve sağlık gibi birçok alanda kullanılmaktadır (Silahtaroglu, 2008). Veri madenciliği uygulamalarında birçok teknik, algoritma ve model kullanılır. Optimizasyon algoritmaları kullanılarak var olan ilişkinin optimum ya da en iyiye yakın durumu bulunabilir.

Bu tez çalışmasında, YAA kullanılarak ile ilk defa sadece numerik verilerden oluşan veritabanları üzerinde birliktelik kurallarının keşfi yapılmıştır. Bölüm 1.1.'de çalışmanın amacı kısaca anlatılıp, Bölüm 1.2'de bu konuda daha önceden yapılmış çalışmalar anlatılacaktır. Bölüm 1.3'te optimizasyon hakkında bilgi verilecektir. Bölüm 1.4.'te fizik tabanlı optimizasyon algoritmaları genel olarak ve YAA ise ayrıntılı bir şekilde anlatılacaktır. Devamında, Bölüm 1.5.'te veri madenciliğinden, Bölüm 1.6'da birliktelik kurallarından ayrıntılı biçimde bahsedilecektir. Daha sonra Bölüm 2.'de bu tez çalışması yapılırken kullanılan program ve veritabanları anlatılıp tez çalışması sırasında izlenen yol gösterilecektir. Bölüm 3.'te ise bu programın uygulandığı veritabanlarından elde edilen bulgular gösterilip; bu bulgular değerlendirilecektir. Bölüm 4.'te tez çalışması sırasında elde edilen sonuçlar sırasıyla özet şeklinde verilecektir. Bölüm 5.'te bu tez çalışması konusunda değerlendirme ve görüşler belirtilecektir. Son olarak Bölüm 6.'da ise bu çalışmanın nasıl geliştirilebileceği konusunda öneriler sunulacaktır.

1.1. Çalışmanın Amacı

İlk defa yapılan bu çalışmada, doğadan esinlenilerek oluşturulan fizik tabanlı optimizasyon algoritmalarından YAA kullanılarak, karmaşık ve zor bir problem olan, numerik verilerden oluşan veritabanlarında nicel birliktelik kurallarının keşfi amaçlanmıştır.

1.2. Bu Konuda Yapılan Çalışmalar

Agrawal ve Srikant (1994), satış hareketlerinden oluşan büyük veritabanlarında birliktelik kuralları bulmak için iki yeni algoritma önermişlerdir. Bu algoritmaların en iyi şekilde AprioriHibrid algoritması ile birleştirilmesi de gösterilmiştir. Hem sentetik hem de gerçek veriler üzerinde yaptıkları denemeler de AIS ve SETM algoritmalarından daha iyi sonuçlar elde etmişlerdir.

Chiu vd. (2006), bulanık birliktelik kuralları madenciliği için kümeleme tabanlı bir metod önermiştir. Bu metod tek bir kez veritabanını tarayarak kümeleme tabanlı bulanık-set (fuzzy-sets) tabloları oluşturur ve sonra hareketli (transaction) kayıtlar, kayıt uzunluğu olan k dikkate alınarak k 'ncü küme tablosuna kümelendir. Deney sonuçları bu metodun Apriori-tabanlı bulanık birliktelik kuralları madenciliği algoritmalarından daha iyi sonuç verdiğini göstermiştir.

Alatas vd. (2008), numerik birliktelik kurallarının keşfi için çok amaçlı differansiyel evrimsel algoritmayı önermiştir. Bu algoritma Pareto optimalliğini temel almış ve birliktelik kurallarının bulunmasını çok amaçlı bir optimizasyon problemi olarak düşünülmüştür. Bu algorithmada tek bir çalıştırmada son popülasyondan tüm doğru ve anlaşılır birliktelik kurallarının çıkarılması amaçlanmıştır. Bu algorithmada, işleme başlamadan önce oluşturulan aralık kuralları bulma sorunu engellenmiştir ve bu algoritma bir kurala uygun sayısal nitelikleri aralıklarda aramak için tasarlanmıştır. Her zaman kullanılan tekniklerin aksine, yüksek destek ve güven değerlerine sahip anlaşılır birliktelik kuralları, doğrudan sürekli verisetleri oluşturmadan bulunmuşlardır. Bu kurallar her veri tabanı için belirlenmesi zor olan en az düzeyde destek ve güven eşiğine bağlı olmadan ortaya çıkarılmıştır.

Yan vd. (2009), minimum destek değeri belirtmeden genetik algoritma kullanarak birliktelik kuralları bulmaya çalışmışlardır. Göreceli olarak güven değeri uygunluk fonksiyonu olarak kullanılmıştır. Çalışmanın daha etkili olabilmesi için FP ağaç yapısı kullanılmıştır. Çalışma sonucunda birliktelik kurallarının bulunmasında yüksek performans ve sistem otomasyonunda yararlar elde edilmiştir. Bu çalışmanın diğer çalışmalarla farkı minimum destek eşik değerine ihtiyaç duymamasıdır.

Kuo vd. (2011), birliktelik kurallarının keşfi için parçacık sürü optimizasyonunu önermiştir. Destek ve güven değerlerinin sınırlarının birliktelik kurallarının kalitesine etki etmesinden dolayı bu çalışma hesaplama etkinliğini arttırmak bu değerlerin otomatik olarak belirlenmesi için önerilmiştir. Bu çalışmada PSO ilk önce her bir parçacığın optimum uygunluk değerini ve sonra veriler ikili değerlere dönüştürüldükten sonra bunların minimum destek ve güven değerlerini bulmuştur. Çalışma gerçek veritabanlarına uygulandığında elde edilen sonuçlar oldukça umut vericidir.

Khademolghorani vd. (2011), birliktelik kurallarının keşfi için YAA algoritmasını kullanmıştır. Bu çalışmada sürekli(frequent) verisetleri üretmeden birliktelik kuralları keşfedilmiştir ve kullanılan bu metot minimum destek ve güven değerlerinden bağımsızdır. Bu çalışmada hareketli(transactions) veritabanları kullanılmıştır. Elde edilen sonuçlar PSO ile karşılaştırılmış ve çalışma zamanı ve global çözümlerin bulunması noktasında daha iyi sonuçlar elde edilmiştir.

Qodmanan vd. (2011), minimum destek ve güven değerlerini dikkate almadan birliktelik kurallarının keşfi için genetik algoritmayı kullanmıştır. Algoritmanın etkisinin artması için FP-ağaç algoritması uygulanmıştır. Bu metot destek ve güven arasında en iyi

korelasyonu sađlayan en iyi kuralları elde etmiştir. Apriori tabanlı algoritmanın tersine bu algoritma desteđe bađlı deđildir. Kullanılan operatörler deđişen uygunluk için esnektir.

Alvarez ve Vazquez (2012), nicel birliktelik kuralları keşfetmek için ayrık ve numerik veriler içeren büyük ve küçük veritabanları üzerinde çalışmışlardır. Bu kuralları bulmak için genetik algoritma kullanmışlardır. Bu çalışmada önemli bir nokta nicel ve kategorik veriler içeren veritabanları üzerinde çalışırken numerik niteliklerin domainlerinde öncelikle ayırıştırma işlemine ihtiyaç duyulmamıştır. Bu genetik algoritma destek, güven, aralık genişliđi ve nitelik sayısı gibi parametrelere dayanmıştır. Bu algoritmanın özelliđi geniş veritabanlarına uygulanabilmesidir.

Sarath ve Ravi (2013), birliktelik kurallarının keşfi için ikili parçacık sürüsü optimizasyonunu önermiştir. Önceki algoritmaların aksine minimum destek ve güven deđerlerini belirtmeden transactional(hareketli) veritabanlarından birliktelik kuralları üretmişlerdir. Veritabanından programa verilen bir M sayısından en iyi M tane kural üretilmiş ve kuralın kalitesi destek ve güven oranlarının çarpımıyla hesaplanan uygunluk fonksiyonuyla belirtilmiştir. Algoritmanın başarısı diđer algoritmalar gibi gereksiz kurallar üretmemesidir.

Minaei-Bidgoli vd. (2013), birliktelik kurallarının keşfi için çok amaçlı genetik algoritmayı kullanmıştır. Birçok metotta karşılaşılan numerik verilerde birliktelik kuralını bulma problemini aşmak için *güven*, *ilginçlik* ve *anlaşılrlık* gibi farklı deđerler kullanılarak çok amaçlı optimizasyon yapılmıştır. En iyi kuralları elde etmek için Pareto optimalliđi kullanılmıştır. Bu metot, belli aralıktaki deđerleri göstermek için üst ve alt aralıklar ile tanımlanmış kaba örüntü kavramını temel almıştır.

Mata vd. (2013), çalışmasında nicel nitelikler içeren veritabanlarında birliktelik kuralları keşfetmek için genetik algoritmayı kullanmıştır. Bu evrimsel algoritma farklı aralık deđerleri bularak sürekli aynı kuralı bulmamak için geliştirilmiştir. Bundan dolayı farklı birliktelik kuralları bulunmuştur.

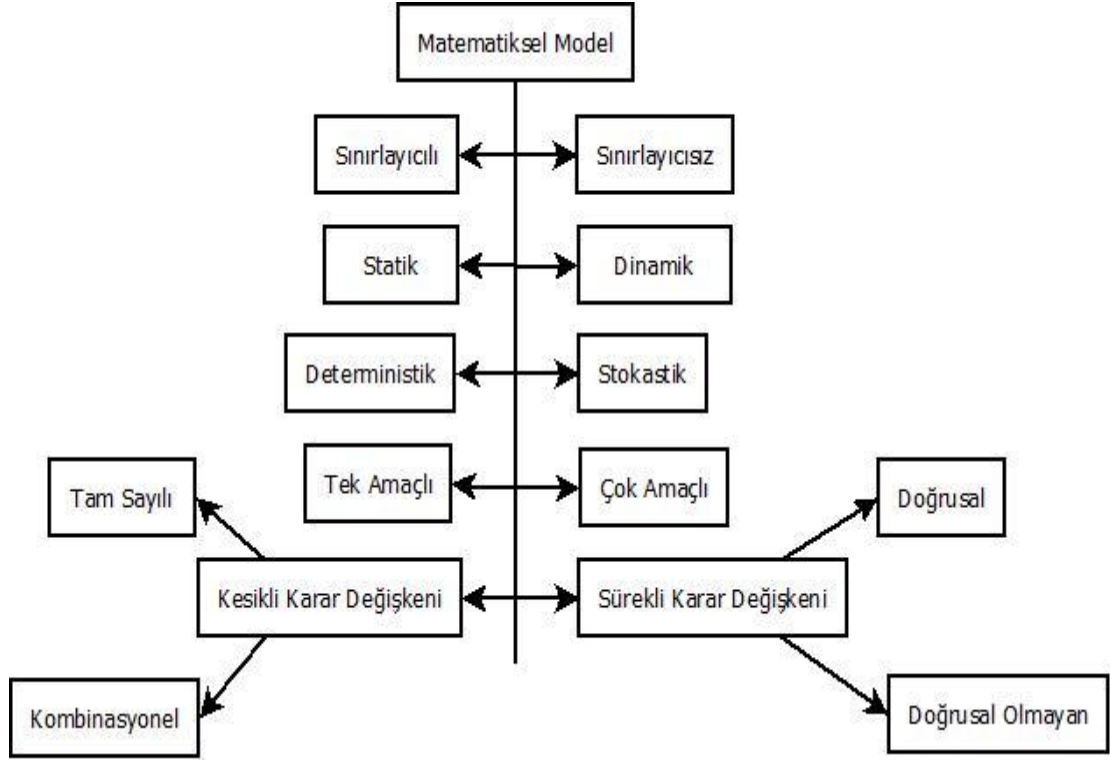
Beiranvand vd. (2014), numerik birliktelik kurallarının keşfi için çok amaçlı PSO algoritması önermiştir. Bu algoritmanın etkinliđini belirtmek için çok amaçlı optimizasyon yaklaşımına güven, anlaşılrlık ve ilginçlik özellikleri eklenmiştir. Buna ek olarak Pareto optimalliđi de kullanılmıştır. Bu metotun iyi sonuçlar verdiđi belirtilmiştir.

1.3. Optimizasyon

İnsan yaşamının giderek karmaşıklaştığı günümüzde kısıtlı kaynaklarla, karşılaşılan problemlerden maksimum ya da minimum fayda sağlamak giderek daha önemli hale gelmiştir. Bu önemin giderek artması optimizasyon kavramının değerini gün geçtikçe arttırmış ve ekonomiden teknolojiye birçok alanda kullanılmasını sağlamıştır.

En basit anlamı ile optimizasyon eldeki kısıtlı kaynakların optimum şekilde kullanılması olarak adlandırılabilir. Matematiksel olarak da optimizasyon, bir fonksiyonun minimize veya maksimize edilmesi için belli aralıkta tanımlı sayıların fonksiyona verilerek problemlerin incelenmesi ve çözülmesi olayıdır. Optimize edilecek büyüklük amaç fonksiyonu olarak adlandırılır. Optimizasyonun sonucunu etkileyen değişkenlere ise karar değişkenleri denir. Bu değişkenler problemin amacına göre belirli değerleri kullanır.

Eğer karar değişkenleri üzerinde hiçbir sınırlama yoksa sınırlayıcısız, en azından bir sınırlama olması durumunda sınırlayıcı olur. Eğer problem tek bir dönem için çözülecekse statik model, birden fazla dönem göz önüne alınarak çözülecekse dinamik model kullanılır. Modelin algoritmada işletilmesi esnasında belirli, kesin parametre veya girdiler kullanılıyorsa model deterministik, olasılık özelliği varsa model stokastiktir. Eğer birden fazla amaç varsa, problemler çok amaçlıdır. Eğer tüm karar değişkenleri pozitif reel (gerçel) değerler alıyorsa sürekli optimizasyon problemi söz konusudur. Tüm karar değişkenlerinin tamsayı değerler alması gerekiyorsa kesikli optimizasyon problemi ortaya çıkar (Alataş, 2007). Aşağıda optimizasyon modellerinin daha iyi anlaşılması için optimizasyon modellerini anlatan bir şekil gösterilmiştir (Şekil 1.1).



Şekil 1.1. Optimizasyon için matematiksel modeller

Klasik optimizasyon algoritmaları karmaşık ve doğrusal olmayan problemlerin çözümünde yetersiz kalmaktadır. Birçoğu problemlerin çözümünde matematiksel modele ihtiyaç duyduğundan buldukları sonuçlar yetersiz ya da anlamlandırılması zor olabilmektedir. Genellikle bu tip optimizasyon algoritmalarında problem algoritmaya uyarlanmakta bu da çeşitli yetersizliklere neden olmaktadır. Maliyet ve zaman gibi sorunlar bunlardan başlıcalarıdır. Farklı alanlardaki birçok optimizasyon problemi farklı tipteki amaç fonksiyonlarına, birçok sınırlayıcıya ve karar değişkenlerine ihtiyaç duyar. Bu tip problemlerin çözümünde günümüzde sezgisel optimizasyon algoritmaları kullanılmaktadır. Sezgisel optimizasyon algoritmaları kendini probleme uyarlayabilmektedir. Zaman ve maliyet açısından klasik optimizasyon algoritmalarına göre avantajlıdır. Sezgisel algoritmaların önemli bir avantajı ise bu algoritmaların esnek olmalarıdır. Esnek oluşları gerçek yaşam problemlerini çok daha doğru temsil etmelerini sağlayabilir. Bu da sezgisel algoritmaların buldukları çözümlerin gerçek hayat problemi için doğru ve iyi bir çözüm olmasını sağlayabilir (Cura, 2008).

1.3.1. Sezgisel Algoritmalar

Günümüzde çözüm uzayı çok büyük ya da tüm çözümlerin değerlendirilmesinin çok fazla zaman alacağı problemler vardır. Bu çok karmaşık problemleri çözmek için bilgisayarlar kullanılır. Fakat ileri derecede teknoloji kullanımı bile uygun çözümlerin istenen zaman ve düşük maliyette bulunması için yeterli değildir. Klasik optimizasyon algoritmalarının çok boyutlu ve karmaşık problemlerin çözümünde yetersiz kalması sezgisel algoritmaların geliştirilmesine neden olmuştur. Sezgisel algoritmalar tüm çözümleri arayıp değerlendirmek yerine bunların arasından olası çözümü bulan algoritmalarlardır. En iyi çözümü bulmayı garanti etmezler bu yüzden yaklaşık çözüm bulan algoritmalar olarak da düşünülebilir.

Sezgisel algoritmalar en uygun çözümü hızlı ve kolay bir şekilde bulmaya çalışırlar. Sezgisel algoritmalar problemlere basit bir şekilde uygulanabilir ve değişik problemlere uyarlanabilirler. Uyarlanabilirlik klasik algoritmaların karşılaştığı bir problemdir ve sezgisel algoritmalar bu yönleriyle bunu aşmıştır. Bununla birlikte sezgisel algoritmalar esnektir yani amaç fonksiyonunda yapılacak değişiklikleri karşılayabilir. Genellikle sezgisel algoritmalar karmaşık yapılarda değildir ve kaliteli çözümler bulmaya odaklıdır.

Sezgisel algoritmalar oluşturulurken fizik, biyoloji, kimya gibi birçok alandan esinlenilmiştir. Yerçekimi kanunu, karıncaların yiyecek arama metodu, genetik kodlama gibi alanlardan esinlenilerek birçok algoritma oluşturulmuştur. Genel olarak sezgisel algoritmalar esinlendikleri alana göre fizik, biyoloji, kimya, müzik ve sosyal tabanlı olmak üzere beş ana başlık altında toplanılabilir. Sezgisel algoritma optimum çözümü bulma işlemi sırasında tek bir çözümden başlıyorsa tek noktali, birçok çözümden başlıyorsa bu çok noktali algoritma olarak adlandırılır. Bu ayrıca popülasyon tabanlı olarak da adlandırılır. Genetik algoritma, karınca koloni algoritması popülasyon tabanlı algoritmalara örnektir. Bunun dışında algoritmada amaç fonksiyonu değişmiyorsa buna sabit amaç fonksiyonlu eğer değişiyorsa algoritma değişken amaç fonksiyonlu olarak adlandırılır. Amaç fonksiyonunu değiştirmek yerel minimuma yakalanmayı engellemek için kullanılabilir. Algoritma çalışması sırasında tek bir komşuluk yapısında çalışıyorsa tek komşulu, farklı yöntemler kullanarak değişken komşuluk alanlarında çalışıyorsa değişken komşulu yapıdadır. Birçok sezgisel algoritma çalışması sırasında elde ettiği bilgileri saklar, yani hafıza kullanır. Çalışması sırasında hafıza kullanan algoritmalara hafızalı algoritmalar

kullanmayanlara ise hafızasız algoritmalar denir. Sezgisel algoritmalar genel olarak aşağıdaki gibi gösterilebilir (Şekil 1.2).



Şekil 1.2. Optimizasyon için matematiksel modeller

1.4. Fizik Tabanlı Sezgisel Optimizasyon Algoritmaları

Sezgisel algoritmalar fizik tabanlı, biyoloji tabanlı, kimya tabanlı, sosyal tabanlı ve müzik tabanlı olmak üzere çeşitli gruplara ayrılmıştır. Burada, bu çalışmada kullanılan Yerçekimsel Arama Algoritması'ndan detaylı olmakla birlikte diğer fizik tabanlı sezgisel optimizasyon algoritmalarından kısaca bahsedilecektir. Kütlelerin çekim kuvvetinden, su damlacıklarının uygun güzergâh bulmasına, parçacıkların nükleer reaksiyonundan, asteroitlerin uzayda en büyük kütleli cisimleri aramasına kadar birçok fizik konusu sezgisel yöntemler için kullanılmıştır. Fiziğin sezgisel optimizasyon algoritmalarında nasıl etkili bir şekilde kullanıldığı gösterilmiştir.

Bundan sonraki bölümlerde, Elektromanyetizma Benzeri Algoritması, Merkez Kuvvet Algoritması (MKA), Yüklü Sistem Arama Algoritması (YSAR), Yapay Fizik Algoritması (YFA), Akıllı Su Damaları Algoritması (ASD), Nehir Formasyonu Dinamikleri Algoritması (NFD), Uzay Yerçekimsel Algoritması (UYA), Parçacık Çarpışması Algoritması (PÇA), Büyük Patlama Büyük Sıçrama Algoritması (BP-BS), Büyük Sıçrama Algoritması (BS) ve Galaksi Tabanlı Arama Algoritması (GTA) kısaca ve Yerçekimsel Arama Algoritması (YAA) ise detaylı bir şekilde anlatılacaktır.

1.4.1. Elektromanyetizma Benzeri Algoritması

Olasılıksal global optimizasyonda, popülasyon tabanlı algoritmalar uygun bölgedeki gelişigüzel örneklenmiş noktalarla başlar. Bu örnek noktaların amaç fonksiyon değerlerine göre bölgenin çekimselliğine karar verilir. Daha sonra aday bölgelerin ilerdeki kullanma durumu için bir mekanizmaya başvurulur. Genetik algoritmada bu mekanizma yeniden üretime, çaprazlamaya ve mutasyon operatörlerine denk gelir (Birbil ve Fang, 2003).

Basit olarak, noktaları, yüksek çekimli vadilere yakınsamaya ve tersine daha dik tepelerden uzak durmaya teşvik eden bir mekanizma inşa edilir. Bu düşünce elektromanyetizma teorisindeki çekim ve itimden analogi yapmaya izin verir. Bu algoritma oluşturulurken basit elektromanyetizma kanunu ile benzerlik kurulmuştur. Buna göre her bir örnek nokta uzaya bırakılan bir parçacık olarak düşünülür ve bu noktaları yükü amaç fonksiyonuna göre belirlenir yani amaç fonksiyon ile ilişkilidir. Bu yük örnek popülasyonda ki her bir noktanın çekim ve itim gücüne karar verir. Bu algoritmada amaç fonksiyonunun değerini en iyi yapmak çekimi en yüksek yapmaktır (Birbil ve Fang, 2003).

Parçacıkların yükleri hesapladıktan sonra bu yükler her bir noktanın daha sonraki iterasyonlara hareketlerine bir yön bulmak için kullanılır. Bir noktaya diğer noktalardan etki eden bileşke kuvvet bulunarak bu yön tayin edilir. Elektromanyetik kuvvetler gibi bu kuvvette, her bir diğer noktadaki vektörel kuvvetlerin ayrı ayrı hesaplanması ile bulunur (Birbil ve Fang, 2003).

1.4.2. Merkez Kuvvet Algoritması

Merkez Kuvvet Algoritması (MKA), yerçekimsel alandaki klasik parçacık kinematikinden esinlenerek oluşturulmuş yeni bir meta-sezgisel algoritmadır. Çözüm için çok boyutlu bir karar uzayını arar. MKA'yı diğer sezgisel algoritmalarından örneğin Parçacık Sürü Optimizasyonu (PSO) ve Karınca Koloni Optimizasyonu (KKO) 'dan ayıran önemli bir özelliği deterministik olmasıdır. Algoritmanın kurucusu olan Formato (2007) MKA'dan genel olarak şöyle bahsetmiştir.

" Bir meta-sezgisel, farklı problemlere sezgisel metotların uygulanabilir olduğunu göstermek için kullanılabilen algoritmik konseptler kümesidir. Başka bir deyişle, bir meta-sezgisel farklı optimizasyon problemlerine göreceli olarak az değişiklik yaparak uygulanan genel amaçlı algoritmik bir çerçevedir. Genel bir metasezgisel herhangi bir matematiksel ispat olmadan önerilir ve çoğu kez biyolojiden alınmış bir metaforundan esinlenilir (KKO ve PSO önemli örneklerdir). MKA tamda bu biçimde önerilmiştir fakat biyolojinin yerine yerçekimsel alandaki kütlelerin hareketlerinin analogisine dayanmaktadır."

1.4.3. Yüklü Sistem Arama Algoritması

YSAR, fizik ve mekanik prensiplerini temel alan bir sezgisel algoritmadır. Coulomb yasasından ve Newton'un hareket kanunundan faydalanmıştır.

Uzayda bulunan elektrik yüklü parçacıkların birbirlerine uyguladıkları elektrik kuvvetinden kaynaklı oluşan çekim ve itim kuvveti vardır. Bu parçacıklar yükleri oranında birbirini çeker veya iterler. Oluşan elektriksel alanda parçacıkların uzaydaki yerlerine karar verir. Bu algoritmada da yükün miktarı amaç fonksiyonunun değeri düşünülerek tanımlanır. Deterministik bir algoritma değildir olasılıksak bir algoritmadır. YSAR bir yük hafızası kullandığı için hafızalı bir algoritma olarak tanımlanabilir (Kaveh ve Talatahari, 2010).

1.4.4. Yapay Fizik Algoritması

Doğal fiziksel kuvvetlerden motive olan William M.Spear, Diana F.Spear ve Rodney YFA algoritmasını oluşturmuşlardır. YFA'nın amacı altıgen şeklinde bir kafes-örgü biçimlendirerek etkili bir anten oluşturma olan robot formasyonuna uygulanmaktadır.

Bu yapılırken de kuvvet yasasından faydalanılarak robotlarının birbirlerini çekip itmesine karar verilir. YFA her bir varlığa fiziksel bir birey gibi kütlesi, pozisyonu ve hızının olduğunu öğretir. Kütle ile optimize edilmeye çalışılan uygunluk arasında bir ilişki vardır. Ayrıca diğer birçok algoritmada olduğu gibi popülasyon tabanlıdır. Birçok algoritmada olduğu gibi bireylerin kütlelerinden ve onların birbirlerine uyguladıkları itim ve çekim kuvvetinden yararlanılarak bireylerin pozisyonlarına karar verilir. Böylelikle en iyi uygunluk alanları bulunabilir (Xie vd., 2009).

1.4.5. Akıllı Su Damlaları Algoritması

ASD, su damlalarının doğada birlikte hareketlerinin kendi iç dinamiklerinden ve çevreyle olan etkileşiminden hareketle kurulmuş bir algoritmadır. Bir akarsuyun doğadaki hareketini, kendisini oluşturan su damlaları sürüsünün oluşturduğu düşünülür. Dolayısı ile her bir su damlacığı bireyinin hızı, taşıyacağı su miktarı gibi etkenler nehrin bir bütün olarak hareketini de etkileyecektir. Aynı şekilde içinde hareket ettikleri çevrede su sürüsünün hareketini etkileyecektir. Örneğin sert bir güzergâh yumuşak bir güzergâha göre daha dirençlidir. Su damlaların bu tür etmenleri düşünerek başlangıçtan varış noktasına olan en uygun ve de genelde en kısa güzergâhı bulması düşünülür. Bu da problemin çözümü olarak düşünülür (Kamkar vd., 2010).

1.4.6. Nehir Formasyonu Dinamikleri Algoritması

NFD, su damlacıklarının yüzeyi aşındırmasından ve yüzeydeki tortuları biriktirmesi olayından esinlenilerek oluşturulmuştur. Hareket halindeki su damlacıkları geçtikleri yerlerin mertebelerini azaltarak ya da yükselterek değiştirirler. Başlangıç ve hedef noktalar belirlenir. Hedef noktanın yüksekliği sifıra ayarlanır. NFD'nin analojisi bakımından, bu hedef nokta denizi belirtir ve bu damlalarının ulaşmak istedikleri yerdir. Diğer başlangıç noktaları ise aynı değeri almak koşulu ile belli bir yüksekliğe ayarlanır. Aranan çözümler

ise azalan mertebelerin güzergâhlarında bulunur. Bir başka deyişle azalan gradyanlar inşa edilir ve bu gradyanlar yeni gradyanlar oluşturmak üzere sonraki damlalar tarafından takip edilir ve yeni olanları kuvvetlendirir. Algoritma tüm damlalar aynı çözümü bulana kadar çalıştırılır (Rabanal vd., 2007).

1.4.7. Uzak Yerçekimsel Algoritması

UYA, oluşturulurken astrofizik algoritmasından faydalanılmıştır. Einstein'ın genel izafiye teorisine göre uzak yerçekimsel alan tarafından şekillendirilir. Bu teoriden hareketle UYA, uzayda bulunan birçok asteroitin en ağır kütleli cisim yani optimal çözümü bulmak için yaptıkları hareketleri örnek almıştır. Bir asteroit, uzak zaman geometrisindeki değişimlerle kendi etrafındaki en ağır kütleye doğru hareket eder ve ona doğru hızlanır. Daha sonrada bu asteroit kendini çeken bu ağır kütle tarafından tekrardan uzaya fırlatılacaktır. Sonra ağır kütleler aramaya devam edecektir.

Yukarıda bahsedilen, ağır kütleli cisim tarafından bir asteroitin tekrardan uzaya fırlatılması olayı bu algoritmaya önemli bir özellik sağlamaktadır. Böylece algoritmanın yerel optimum çözüme takılıp kalma olasılığı çok düşmektedir. Bu da algoritmanın durmadan diğer çözümleri araması anlamına gelmektedir (Hsiao vd., 2007).

1.4.8. Parçacık Çarpışması Algoritması

PÇA, Genetik Algoritmalar ve Isıl İşlem Algoritması gibi bazı optimizasyon algoritmalarının çeşitli eksikliklerini gidermek için geliştirilmiştir. Örneğin Isıl İşlem Algoritması parametrelerin seçiminde çok hassastır ve genetik algoritma ise yetersiz yakınsamaya karşı duyarlıdır.

PÇA oluşturulurken genel olarak nükleer çarpışma reaksiyonlarından esinlenmiştir. Özellikle saçılmadan (gelen nötronun hedef çekirdek ile çarpışmasından sonra saçılması) ve soğurmaktan (gelen nötronun hedef çekirdek tarafından soğurulması) temelde faydalanılmıştır. Isıl İşlem Algoritmasından temel farkı kullanıcı tanımlı parametrelere dayanmamasıdır (Sacco ve Oliveria, 2008).

1.4.9. Büyük Patlama Büyük Sıçrama Algoritması

BP-BS, algoritması oluşturulurken, evrenin oluşum teorisi olan Büyük Patlama teorisinden esinlenilmiştir. Bu algoritmada başlangıç popülasyonu Büyük Patlama safhası olarak adlandırılan durumda oluşturulur. Bu da genetik algoritmada olduğu gibi gelişigüzel oluşturulur. Büyük Patlama safhasını ise Büyük Sıçrama safhası izler. Büyük Sıçrama, sadece kütle merkezinin hesaplanmasından türetildiğinden dolayı 'kütle' merkezi olarak adlandırılan birçok girişi ve sadece bir çıkışı olan bir yakınsama operatörüdür. Büyük Patlama ve Büyük Sıçrama basamakları birbirini takip eder. İkinci patlamadan sonra kütle merkezi yeniden hesaplanır. Bu başarılı *patlama* ve *daralma* basamakları bir durdurma kriterine rastlayana kadar tekrar eder. Algoritmanın çalışması bittiğinde çözüm adayları kütle merkezi etrafında toplanmış olur (Erol ve Eksin, 2006).

1.4.10. Büyük Sıçrama Algoritması

BS, Kapalı Evren Teorisi'nden esinlenilerek oluşturulmuştur. Evren, ilk oluştuğunda büyük bir patlama gerçekleşmiş ve ortaya çıkan kinetik enerji sayesinde kütleler birbirinden uzaklaşmış ve daha sonra kütleler arasındaki çekim enerjisi kinetik enerjiden fazla olmaya başladığında daralma ya da çökme olarak nitelendirilen olay gerçekleşmiştir. Bu süreç evrende tek bir cismin kalması yani en uygun çözümün oluşması ile sonuçlanır. Bu tek bir noktaya yakınsama olarak düşünülür. Algoritma uzaydan tek bir cisim kaldığında bitirilebilir. Esinlendiği kuram olarak BS-BÇ algoritmasına benzese de çeşitli farklılıkları vardır (Kripta ve M.L.Kripta, 2008).

1.4.11. Galaksi Tabanlı Arama Algoritması

GTA, spiral kaotik hareketlerden esinlenilerek oluşturulmuştur. GTA spiral kaotik hareketlerden faydalanarak çözüm uzayını arar. GTA'da iyi bir çözüm bulmak için galaksinin kolları yavaşça açılır ve arama uzayını kapsar.

GTA yerel optimuma takılıp kalmayı engellemek için ya da bu durumu en aza indirmek için kaosu kullanır. Kaotik işleyişten kaynaklı GTA aynı çözümü geri dönmez ve bundan dolayı bulunan çözümlerin çeşitliliği yüksek oranda korunur. Çeşitliliği yüksek tutmak

özellikle arama uzaylarında birçok yerel optimumun var olduğu çok modlu problemlerde önemlidir (Shah-Hosseini, 2011).

1.4.12. Yerçekimsel Arama Algoritması

Büyük boyutlu bir arama uzayı ile optimizasyon problemlerinin çözümünde, klasik optimizasyon algoritmaları uygun çözüm bulmaz çünkü arama uzayı problem boyutu ile beraber üssel olarak artmaktadır. Bu yüzden bu problemlerin çözümünde kesin teknikler (örn: detaylı aramalar) pratik değildir.

Son on yıldan fazladır doğal fenomenlerden esinlenen algoritmalara büyük bir ilgi vardır. Birçok araştırmacı tarafından bu algoritmaların, amaç fonksiyonlarının optimizasyonu, örüntü tanıma, nesne kontrolleri, görüntü işleme, filtre modelleme gibi karmaşık problemlerin çözümüne uygun olduğu gösterilmiştir. Şu zamana kadar çeşitli sezgisel yaklaşımlar araştırmacılar tarafından kabul edilmiştir (Rashedi vd., 2009).

Burada, Yerçekimsel Arama Algoritması diye isimlendirilen ve yerçekimi kanununu temel alan bir algoritma gösterilmiştir. Bu algoritma Newton'nun :” Evrendeki her parçacık diğer parçacıklara bir kuvvetle etki eder. Bu kuvvet direkt olarak parçacıkların kütleleri ile doğru orantılı ve parçacıkların aralarındaki uzaklığın karesi ile ters orantılıdır”, yasası temel alınarak oluşturulmuştur (Rashedi vd., 2009).

1.4.12.1. Yerçekimi Kanunu

Yerçekimi kütlelerin birbirlerine hızlanarak yönelme eğilimidir. Bu, doğadaki en önemli dört etkileşimden birisidir (diğerleri: elektromanyetik güç, zayıf nükleer güç ve güçlü nükleer güç). Evrendeki her parçacık diğerlerini kendine doğru çeker. Yerçekimi her yerdedir. Yerçekiminin kaçınılmazlığı onu diğer doğal kuvvetlerden daha farklı kılar.

Newton'un yerçekiminin davranışı, “belli mesafedeki etki”, olarak adlandırılır. Bu yerçekiminin farklı parçacıklar arasında herhangi bir aracı ve herhangi bir gecikme olmadan etki ettiği anlamına gelir. Newton'un yerçekimi kuvveti kanununda her bir parçacık diğer bir parçacığı bir “yerçekimi” kuvveti ile çeker. İki parçacık arasındaki yerçekimi kuvveti, parçacıkların kütleleri ile direkt olarak doğru orantılı ve aralarındaki uzaklığın karesi ile ters olarak orantılıdır (Rashedi vd., 2009).

$$F = G \frac{M_1 M_2}{R^2} \quad (1.1)$$

F yerçekimi kuvvetinin büyüklüğüdür. G yerçekimi kuvveti sabitidir. M_1 ve M_2 sırası ile birinci ve ikinci parçacığın kütlesidir. R , iki parçacık arasındaki mesafedir. Newton'un ikinci kanunu der ki: bir F kuvveti, bir parçacığa uygulandığında onun ivmesi sadece onun kuvvetine ve onun kütlesine bağlıdır.

$$a = \frac{F}{M} \quad (1.2)$$

Denklem (1.1) ve (1.2)'yi temel alırsak, evrende daha büyük ve daha yakın parçacıklarda yüksek olacak şekilde tüm parçacıklar arasında bir yerçekimi kuvveti vardır. Şekil 1.3'de gösterildiği üzere iki parçacık arasındaki mesafenin artması aralarındaki yerçekimi kuvvetinin azalması anlamına gelmektedir. Bu şekilde F_{1j} , M_1 'e M_j 'ye etki eden kuvvettir ve F_1 , M_1 'e etki eden tüm kuvvetlerin toplamıdır ve a_1 ivme vektörüne neden olur (Rashedi vd., 2009).

Buna ek olarak, azalan yerçekiminin etkisinden dolayı, “yerçekimi sabitinin” güncel değeri evrenin yaşının güncel değerine bağlıdır. Denklem (1.3) yerçekimi sabitinin azalmasını verir. G , yaş ile:

$$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^\beta, \quad \beta < 1 \quad (1.3)$$

$G(t)$, t anında yerçekimi sabitidir. $G(t_0)$, t_0 zamanının ilk kozmik kuantum-aralığındaki yerçekimi sabitidir. Üç çeşit kütle teorik fizikte tanımlanmıştır.

- Aktif Yerçekimsel Kütle:

M_a , belirli bir nesneden kaynaklı yerçekimsel alanın kuvvetinin miktarıdır. Küçük aktif yerçekimsel kütle, daha büyük aktif yerçekimsel kütlesi olan nesneden daha zayıftır (Rashedi vd., 2009).

- Pasif Yerçekimsel Kütle:

M_p , yerçekimsel alanla etkileşim halinde olan bir nesnenin kuvvetinin ölçümüdür. Aynı yerçekimsel alanda daha küçük pasif yerçekimsel kütleyle sahip olan bir nesne, daha büyük pasif yerçekimsel kütleyle sahip olan nesneden daha az kuvvete maruz kalır.

- Eylemsiz Kütle:

M_i , bir kuvvet uygulandığında kendi hareket durumunu değiştirmeye direnen bir nesnenin ölçümüdür. Büyük bir eylemsiz kütleyle sahip olan nesne kendi hareketini daha yavaş değiştirir ve daha küçük eylemsiz nesneye sahip olan ise daha hızlı değiştirir (Rashedi vd., 2009).

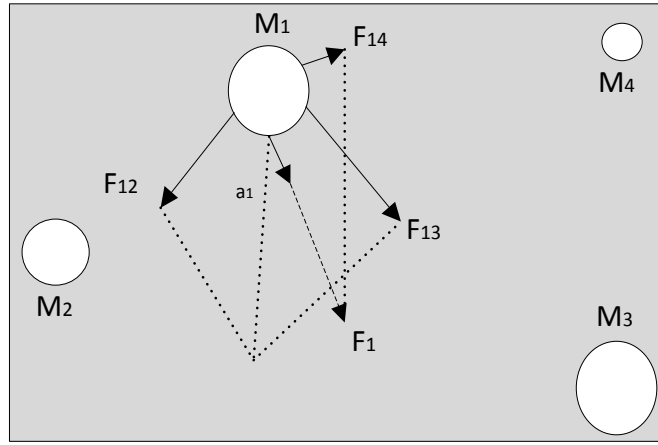
Şimdi yukarıda bahsedilen bakış açılarını düşünerek Newton'un yasalarını tekrardan yazalım.

Kütle j ' den kütle i ' ye etki eden yerçekimsel kuvvet F_{ij} , j ' nin aktif yerçekimsel kuvveti ve i 'nin pasif yerçekimsel kuvveti ile doğru orantılı ve aralarındaki mesafenin karesi ile ters orantılıdır. a_i , F_{ij} ile doğru orantılı ve i ' nin eylemsiz kütlesi ile ters orantılıdır. Denklem (1.1) ve (1.2)' yi şöyle yazabiliriz:

$$F_{ij} = G \frac{M_{aj} \times M_{pi}}{R^2} \quad (1.4)$$

$$a_i = \frac{F_{ij}}{M_{ii}} \quad (1.5)$$

M_{aj} ve M_{pi} sırası ile j ' nin aktif yerçekimsel kuvvetini ve i ' nin pasif yerçekimsel kuvvetini gösterir. M_{ii} ise parçacık i 'nin eylemsiz kütlesini gösterir. Aşağıda M_1 , M_2 , M_3 ve M_4 kütlelerin arasındaki kuvvet etkileşimini gösterilmiştir (Şekil 1.3).



Şekil 1.3. Tüm kütleler, diğer kütlelerden etkilenen sonuç kuvvetine doğru hızlanır.

Eylemsiz, pasif yerçekimsel ve aktif yerçekimsel kütlelerin kavramsal olarak birbirlerinden ayrık olmalarına rağmen şimdiye kadar hiçbir tecrübe açık bir biçimde aralarındaki farkı gösterememiştir. Bu, zayıf denklik prensibi olarak bilinir. Genel standart izafiyet, eylemsiz kütle ve yerçekimsel kütlenin denkliğini farz eder; bu denklik bazen güçlü denklik prensibi olarak adlandırılır (Rashedi vd., 2009).

1.4.12.2. Yerçekimsel Arama Algoritması (YAA)

Bu bölümde, yerçekimi yasasını temel alan optimizasyon algoritması tanıtılacaktır. Bu algorithmada bireyler nesnelere olarak düşünülmüş ve performansları kütleleri ile ölçülmüştür. Tüm bu nesnelere birbirlerini yerçekimi kuvveti ile çekerler ve bu daha ağır kütleli nesnelere doğru global bir harekete neden olur. Bu yüzden, yerçekimsel kuvvet boyunca nesnelere iletişimin direkt bir formunu kullanırlar. İyi çözümlere uygun olan ağır kütleler, hafif olandan daha yavaş hareket ederler. Bu algoritmanın *kullanma* basamağını garanti eder.

YAA'da her bir kütlenin (bireyin) 4 tane spesifikasyonu vardır: pozisyon, eylemsiz kütle, aktif yerçekimsel kütle ve pasif yerçekimsel kütle. Kütlenin pozisyonu problemin çözümüne uygundur ve onun yerçekimsel ve eylemsiz kütleleri uygun bir fonksiyon kullanmaya karar verir (Rashedi vd., 2009).

Başka bir deyişle her bir kütle bir çözümü belirtir ve algoritma doğru olarak ayarlanmış yerçekimsel ve eylemsiz kütleler tarafından yönlendirilir. Zamanın bitmesi ile kütlelerin en ağır kütle tarafından çekilmesi beklenilir. Bu kütle, arama uzayında bir optimum çözümü gösterecektir.

YAA kütlelerin izole edilmiş bir sistemi olarak düşünülebilir. Bu kütlelerin Newton'un yerçekimi ve hareket kanunlarına uyduğu küçük bir yapay dünyaya benzer. Kütleler şu kurallara uyarlar:

- Yerçekimi Kanunu: Her bir parçacık diğer parçacıkları çeker ve yerçekimsel kuvvet iki parçacığın kütleleri ile doğru ve aralarındaki uzaklığın karesi ile ters orantılıdır. Burada R , R^2 yerine kullanıldı. Çünkü R , R^2 'ye göre tüm deneysel durumlarda iyi sonuçlar vermiştir (Rashedi vd., 2009).
- Hareket Kanunu: Herhangi bir kütlenin mevcut hızı, onun önceki hızının bölümlerinin toplamına ve değişimine eşittir. Kütlenin hızındaki veya ivmesindeki değişim eylemsiz kütle tarafından bölünmüş sisteme etki eden kuvvete eşittir.

Şimdi N tane bireyli (kütleli) bir sistem düşünelim. i 'nci bireyin pozisyonu şöyle tanımlanabilir.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, \dots, N, \quad (1.6)$$

x_i^d , d 'nci boyuttaki i 'nci bireyin pozisyonunu gösterir.

Belirli bir ' t ' anında, ' j ' kütesinden ' i ' kütesine etki eden kuvveti şöyle tanımlanır:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij(t)} + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (1.7)$$

M_{aj} , j bireyi ile ilişkilendirilmiş aktif yerçekimsel kütle, M_{pi} , i bireyi ile ilişkilendirilmiş pasif yerçekimsel kütle, $G(t)$, t anındaki yerçekimsel sabit, ε küçük bir sabit ve $R_{ij}(t)$, i ve j bireyleri arasındaki Öklit mesafesi:

$$R_{ij}(t) = ||X_i(t), X_j(t)||_2 \quad (1.8)$$

Algoritmaya olasılıksal bir karakter vermek için birey i 'ye d boyutunda etki eden toplam kuvvetin, diğer bireylerden etki eden kuvvetlerin d 'nci bileşenlerinin toplamının rastgele ağırlıklandırılması olabileceği düşünülür (Rashedi vd., 2009).

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (1.9)$$

$rand_j$, $[0,1]$ aralığında gelişigüzel bir sayıdır.

Bu yüzden hareket kanunu ile birlikte, birey i 'nin t anında ve d 'ninci yöndeki ivmesi, $a_i^d(t)$, şöyle bulunur:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (1.10)$$

M_{ii} , i 'ninci bireyin eylemsiz kütlesidir.

Dahası, bireyin bir sonraki hızı, şimdiki hızının bölümünün kendi hızına eklenmiş hali olarak düşünülür. Bundan dolayı pozisyonu ve hızı şöyle hesaplanabilir.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (1.11)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (1.12)$$

$rand_i$, $[0,1]$ aralığında benzer rastgele bir değerdir. Gelişigüzel numara, aramaya gelişigüzel bir karakter vermek için kullanılır (Rashedi vd., 2009).

Yerçekimsel sabit G , başlangıçta ayarlanır ve arama doğruluğunu kontrol etmek için zamanla birlikte azaltılacaktır. Başka bir deyişle, G , başlangıç değeri (G_0) ve zaman (t)'nin fonksiyonudur.

$$G(t) = G(G_0, t) \quad (1.13)$$

Yerçekimsel ve eylemsiz kütleleri uygunluk değerlendirme tarafından basit bir şekilde hesaplanır. Daha ağır bir kütle daha etkili bir birey anlamına gelir. Bu daha iyi bireylerin daha büyük çekime sahip olduğu ve daha yavaş hareket ettiği anlamına gelmesidir. Yerçekimsel kütle ile eylemsiz kütlenin eşitliği farz edilerek, kütlelerin değerleri uygunlukları eşlenerek hesaplanır. Yerçekimsel ve eylemsiz kütlelerini şu denklemlerle güncellenir:

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad i = 1, 2, \dots, N, \quad (1.14)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (1.15)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}, \quad (1.16)$$

$fit_i(t)$, t anında ki birey i ' nin uygunluk değerini gösterir ve $worst(t)$ ve $best(t)$ şöyle tanımlanır (minimizasyon problemi için):

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (1.17)$$

$$worst(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (1.18)$$

Bir maksimizasyon problemi için Denklem (1.17) ve (1.18), Denklem (1.19) ve (1.20) ile değiştirilmelidir:

$$best(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (1.19)$$

$$worst(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (1.20)$$

Keşif ve kullanma arasında iyi bir anlaşma uygulamanın bir yolu Denklem (1.9)' daki bireylerin sayısını azaltmaktır. Bu yüzden sadece kendi kuvvetlerini başkasına uygulayan bir grup büyük kütleli bireyler önerilir. Fakat bunu yaparken dikkatli olunmalıdır çünkü bu keşif kuvvetini azaltabilir ve kullanma yeteneğini arttırabilir.

Algoritmanın yerel optimumda kalmasını engellemek için başlangıçta keşfi kullanması gerektiği hatırlatılır. Son iterasyonlara doğru keşif zayıflayıp kaybolmalı ve kullanma yavaş yavaş görünür olmalıdır. YAA' nın performansını, keşfi ve kullanmayı kontrol ederek geliştirmek için sadece $Kbest$ bireyleri diğerlerini çekecektir. $Kbest$, başlangıçtaki K_0 başlangıç değeri ve azalan zamanla birlikte zamanın fonksiyonudur. Belli bir yolla, başlangıçta tüm bireyler bir kuvvet uygular ve zamanın geçmesi ile $Kbest$ lineer olarak

azaltılır ve sonuçta diğer bireylere kuvvet uygulayan sadece bir birey olacaktır. Bu yüzden, Denklem (1.9) şöyle değiştirilebilir:

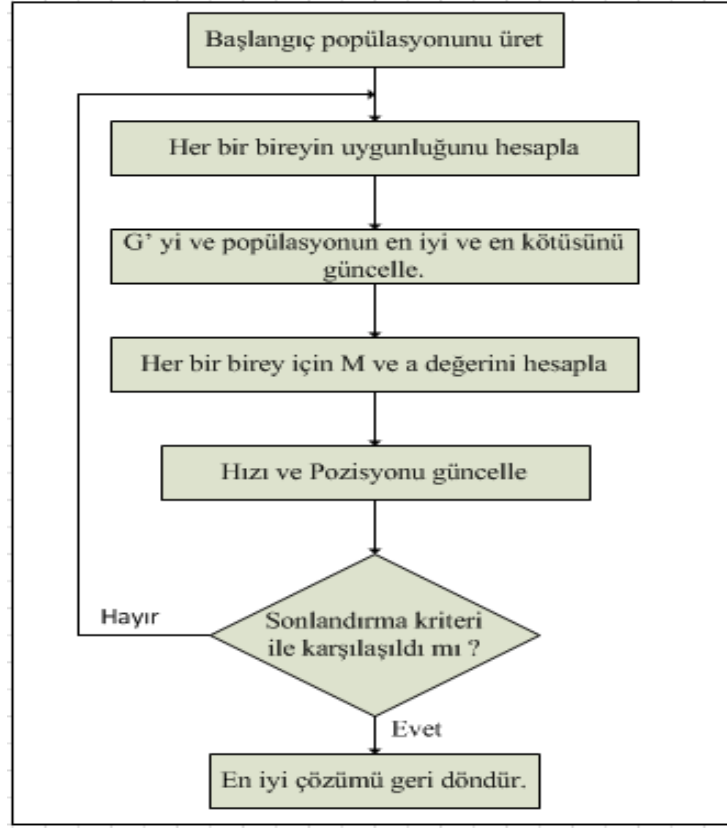
$$F_i^d(t) = \sum_{j \in K_{best}, j \neq i} rand_j F_{ij}^d(t) \quad (1.21)$$

Kbest, en iyi uygunluk değeri ve en büyük kütleli olan ilk K birey grubudur. Algoritmanın farklı basamakları şunlardır:

- a) Arama uzayını belirleme
- b) Gelişigüzel yapılan başlangıca ayarlama.
- c) Bireylerin uygunluk değerlendirilmesi
- d) $G(t)$, $best(t)$, $worst(t)$ ve $M_i(t)$ 'i güncelle ve $i = 1, 2, \dots, N$.
- e) Farklı yönlerdeki kuvvetlerin toplamının hesaplanması
- f) Hızın ve ivmenin hesaplanması.
- g) Bireylerin pozisyonunu güncelleme
- h) Durdurma kriterine ulaşana kadar c'den g'ye kadar basamakları tekrar et.
- i) Son

YAA'nın genel çalışma prensibi aşağıda gösterilmiştir (Şekil 1.4). Önerilen algoritmanın nasıl etkili olduğunu göstermek için bazı önemli noktalar kaydedilir. Her bir bireyin diğer bireylerin performansını gözlemlemesinden dolayı, yerçekimsel kuvvet bir bilgi transfer aracıdır (Rashedi vd., 2009).

- Ağır bir kütleli daha geniş bir etki yarıçapı vardır. Bundan dolayı büyük bir çekim yoğunluğu olur. Bu yüzden yüksek performanslı bireylerin daha büyük yerçekimsel kütlesi vardır. Bunun sonucu olarak, bireyler en iyi bireye doğru hareket etme eğilimi gösterirler.



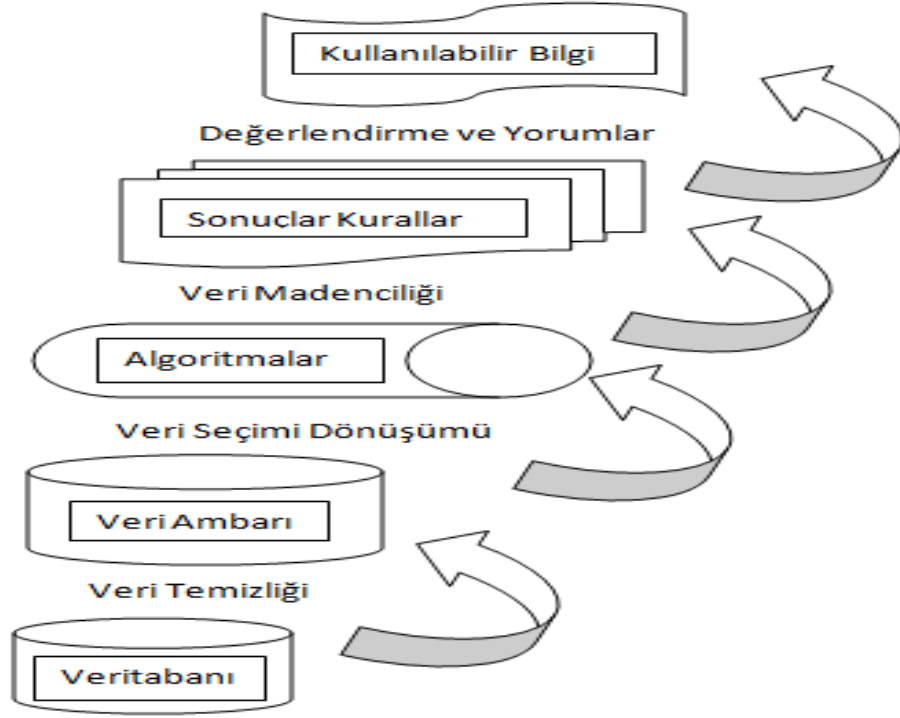
Şekil 1.4. YAA'nın genel çalışma prensibi.

- -Eylemsiz kütle harekete karşıdır ve kütlelerin hareketini yavaşlatır. Bu yüzden ağır eylemsiz kütleli bireyler yavaş hareket ederler ve uzayı daha yerel ararlar. Böylece bu adaptif öğrenme oranı olarak düşünülebilir.
- -Yerçekimsel sabit aramanın doğruluğunu ayarlar ve bu yüzden zamanla beraber azalır (Isıl İşlem Algoritması'ndaki sıcaklık gibi).YAA hafızası az olan bir algoritmadır. Fakat hafızası olan algoritmalar gibi etkili çalışır.
- -Burada yerçekimsel ve eylemsiz kütlelerin aynı olduğunu farz edilmiştir. Fakat bazı uygulamalar için farklı değerler kullanılabilir. Büyük bir eylemsiz kütlesi arama uzayında daha yavaş birey hareketi sağlar ve bu yüzden daha hassas aramalara neden olur. Tersine daha büyük yerçekimsel kütle daha büyük birey çekimine neden olur. Bu daha hızlı yakınsama demektir (Rashedi vd., 2009).

1.5. Veri Madenciliđi

Bilgisayar kullanımının yaygınlařması ile paralel olarak teknolojik depolama aygıtları genişlemiřtir. Geliřen ve karmařıklařan insan yařamı beraberinde depolanacak milyonlarca veriyi getirmiřtir. Büyük market zincirlerinden tutunda giyim mađazalarına, hastane veri stoklarından üniversitelerin bilimsel çalışmalarına kadar birçok bilgi artık depolanabilmektedir. Milyonlarca veri keřfedilmeyi bekleyen yeni kuralları, bađlantıları ve yani yeni bilgileri barındırmaktadır.

Veri madenciliđi tam da bu noktada işlevsel hale gelmekte ve her geçen gün önemini arttırmaktadır. Veri madenciliđi çalışmalarını 1990'lı yıllardan beri veri depolama araçları, barkod ve RFDI teknolojilerine paralel olarak geliřmekte olan bir konudur. Veri madenciliđi geniş veri yığınlarının bulunduđu veritabanlarından daha önceden keřfedilmemiş bilgilerin çıkarılıp kullanılması olarak en yaygın haliyle tanımlanabilir. Burada gözden kaçırılmaması gereken husus önceden keřfedilmemiş yani bilinmeyen bilgilerin bulunmasıdır. Veri madenciliđi önceden bilinen bilgilerin çeřitli yollarla ispatlanmasını sađlayan bir disiplin deđildir. Bilinen en klasik örnek ise bira-çocuk bezi örneđidir. Bir perakende marketler zincirinin yaptıđı veri madenciliđi arařtırmasında özellikle cuma günleri çocuk bezi alan kiřilerin çođunluđun kendilerine bira aldıkları da görülmüřtür. Bu klasik örnek veri madenciliđinin sonuçlarından küçük bir örnektir (Silahtarođlu, 2008). Ařađıda veri madenciliđi ve bilgi keřif sürecini daha iyi anlamak için bir Őekil gösterilmiřtir (Őekil 1.5).



Şekil 1.5. Veri madenciliği ve bilgi keşfi süreci

1.5.1. Veri Madenciliğinin Uygulandığı Alanlar

Veri madenciliği pazarlama, bankacılık, sağlık gibi birçok alanda kullanılmaktadır. Bu alanlar geniş veri ambarları oluşturduğundan veri madenciliği çalışmaları ile ilgili uygun imkânlar vermektedir. Aşağıda pazarlama yönetimi ile ilgili belli uygulamalardan bahsederek veri madenciliğinin uygulama alanları hakkında bilgi verilmektedir.

Pazarlama Yöntemi:

Pazarlama yönteminde müşteri pazar olgusu üzerinden çeşitli ilişkiler geliştirilmektedir. Müşterilerin satın aldığı ürünlerden, yaşlarından cinsiyetlerine kadar çeşitli bilgiler çıkarılmaktadır. Müşterinin satın alma örüntüsünün çıkarılması işleminde, örneğin makarna ve pizza alanların % 40'nın ketçap alıyor bilgisi deneyimli bir işletmenin kolayca çıkarabileceği bir bilgidir. Oysa veri madenciliğinde kolayca bulunabilecek bir bilginin çıkarılmasından ziyade önceden kestirilemeyen bir bilginin çıkarılması esastır. Makarna ve pizza alan erkeklerin %15'inin çamaşır makinesi deterjanı da aldığı bilgisi gibi kestirilmesi zor bilgiler. Bunun dışında müşterilerin aldığı ürünler ile müşterilerin yaşları arasında bir ilişki olup olmadığı bilgisine bakılır. Örneğin yaşı 25-40 arasında olan ve makarna alan erkeklerin %12'si çocuk bezi alır. Buna ek olarak müşterilerin pazar kampanyalara

katılımının sağlanması ya da hangi özellikteki bireylerin bu kampanyalara katılacağını belirlenmesi içinde kullanılır. Pazar sepeti analizi kullanılarak A ürünü alan müşteri bunun yanında D ve M ürünü alıyor mu? A ürünün alan kişinin bir hafta içinde M ürünü alma olasılığı nedir? Bütün bunlar veri madenciliği yöntemleri ile belirlenebilir. Pazarlama alanının dışında tıp, biyoloji gibi birçok uygulama alanı bulunmaktadır (Silahtaroglu, 2008).

1.5.2. Veri Ambarları ve OLAP

Veri madenciliği çalışmaları için bize içinde milyonlarca verinin depolandığı veritabanları gerekmektedir. Kullanıcılar bu veritabanlarını doğrudan kullanamazlar. Bu veritabanlarının kullanıcının amacı doğrultusunda düzenlenmesi gerekir. Yani veri madenciliği yapılabilmesi için uygun hale getirilmelidir. İşte bu belli amaca göre düzenlenmiş veritabanlarına veri ambarları denir (Silahtaroglu, 2008).

Veri ambarlarının aşağıda belirtilen özellikleri vardır:

- Konu odaklıdır: Veri ambarının önemli özelliklerinden biridir. Veri ambarı hazırlanırken bir amaca veya konuya yönelik olmalıdır. Örneğin bir veri ambarı müşteri, satış, ürün gibi konulara yönelik olarak hazırlanabilir.
- Bütünleşiktir: Bu özellik veri ambarında istisnası kabul edilmeyen bir özelliktir. Veri ambarında veriler bütünleşik olmalıdır. Bütünleşiklik kavramı ölçü birimlerinin tutarlılığı, veri kodlamalarının aynı olması gibi durumları kapsar. Küçük bir örnek verilirse, belli veri tabanlarında erkek ve kadın "E" ve "K" olarak belirtilmekte, bazılarında ise 1 ve 0 değerleri ile belirtilmektedir. Bunun aynılaştırılması gerekmektedir.
- Zaman boyutu: Veri ambarındaki zaman kavramı günlük veri tabanı işlemlerinde kullanılan anlık zaman kavramından farklıdır. Kısa dönemli ya da anlık zaman kavramından ziyade yılları kapsayan belli anlara ait bilgiler kullanılır. Veri ambarı bir bilginin en az beş yıllık değerinin içerir.
- Sadece okunabilir olması: Veri ambarından bilgiler sadece okunabilir yani silinemez. Günlük işlemlerde kullanılan uygulamalarda veri tabanına veri eklenebilir, silinebilir fakat bu veri ambarında farklıdır. Veri ambarında ki veri belli gereksinimlere yanıt vermek için tasarlanmıştır bu yüzden bu tür işlemler yapılmaz.

1.5.2.1. OLAP (On-Line Analytical Processing):

Çeşitli stratejik ve taktik konular hakkında karar vermek için veri ambarları üzerinde yapılan veri analizi ve sorgulama işlemlerine OLAP denir.

OLAP normal sorgulamalardan farklıdır. Örneğin bir market veritabanını düşünelim. Bu veritabanında günlük malların satışı, depolarda kalan miktarları gibi veriler saklanır. Normal bir sorgulama ile günlük satılan kola miktarını ya da herhangi bir ürüne ait haftalık satılan mal miktarını bulabiliriz. Fakat OLAP sorgulaması ile kola alanların %20'sinin patates alma olasılığına ulaşırız (Silahtaroglu, 2008).

1.5.3. Veri Madenciliği İçin Verilerin Hazırlanması

Günümüzde büyük veri tabanlarının oluşması bunların ihtiva ettiği milyonlarca verinin var olması anlamına gelmektedir. Bu verilerin istenilen amaca uygun olarak hazırlanması gerekmektedir.

Veritabanları tutarlı olmayabilir. Örneğin birçok veri tekrar ediyor olabilir. Ya da bir veritabanında yaş ve doğum tarihi birlikte girilmiş olabilir. Buna ek olarak eksik girilmiş veriler (missing data) olabilir. Tüm bu eksikliklerin giderilmesi için veritabanlarının belirli işlemlerden geçirilmesi gerekmektedir. Bu işlem dört başlık altında incelenebilir.

1.5.3.1. Verilerin Temizlenmesi

Kirli veri olarak adlandırılan kayıp ve gürültünün giderilmesi işlemine genel olarak verilerin temizlenmesi denir (Örn bir kişinin doğum tarihi 1078 olarak girilmişse buna gürültü veya gürültülü bilgi denir).

Verilerin temizlenmesi için aşağıdan başlıklar halinde sunulmuş teknikler kullanılabilir:

- a) Veritabanında toplam kayıt listesine oranla kayıp veri oranı sonucu etkilemeyecek kadar az ise bu kayıtlar iptal edilebilir.
- b) Veritabanı küçükse kayıp veriler elle girilebilir.
- c) Kayıp olan verilere aynı bilgi girilebilir. Fakat bunun genel sonucu etkileyebileceği akıldan çıkarılmamalıdır.
- d) Veriler kayıpsa bu kısma diğer değerlerin ortalaması yazılabilir.

e) Regresyon yöntemi kullanılarak kayıp veriler tahmin edilebilir.

1.5.3.2. Veri Bütünleştirilmesi

Farklı veritabanlarından alınan kaynakların aynı türde olmaması durumunda oluşabilecek sorunların engellemek için verilerin bütünleştirilmesi gerekmektedir. Gerçi veri ambarının alt yapısı oluşturulurken yapılması gereken bir işlemdir fakat bu yapı kurulmamış olabilir.

1.5.3.3. Veri İndirgeme

Veri madenciliği işlemlerinde bazen çözümlene süresi uzun sürmektedir. Bu noktada sonuçta elde edilecek sonucun değişmeyeceği düşünülüyorsa veri sayısı azaltılabilir. Bu tür işlemlere veri indirgeme denir. Kısaca belirtmek gerekirse veri birleştirme, boyut indirgeme, veri sıkıştırma, örnekleme ve genelleme olarak çeşitli biçimlerde yapılır.

1.5.3.4. Veri Dönüştürme

Veri madenciliği uygulamalarında ya da çözümlerinde veriyi olduğu gibi kullanmak bazen doğru sonuçlar bulma açısından uygun olmayabilir. Örneğin değişkenlerin aldığı çok büyük ya da çok küçük değerler çözümlerin sağlıklı şekilde yapılmasının engelleyebilir. Bundan dolayı çeşitli normalleştirme ya da standartlaştırma yöntemleri vardır. Normalleştirmede çok kullanılan yöntemlerden birine örnek verirsek:

- Min- Max Normalleştirilmesi

Verilerin 0 ile 1 arasındaki sayısal değerlere dönüştürmek için kullanılan yöntemdir. Bu yöntem verilerin içindeki en büyük ve en küçük değer belirlenerek diğer değerlerin bunlara göre dönüştürülmesini baz alır. Aşağıda örnek bir tablo üzerinden gösterilmiştir. Kullanılan dönüştürme formülü şu şekildedir:

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1.22)$$

X^* değeri 0 ile 1 arasından dönüşüm sonucu bulunan değeri, X ise tablodaki değeri, X_{min} en küçük değeri ve X_{max} ise en büyük değeri ifade etmektedir.

Birinci değer için: $X_{min} = 25$ ve $X_{max} = 75$

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} = \frac{25 - 25}{75 - 25} = 0$$

Aşağıdaki X değerlerinin normalleştirilmesi sonucunda elde edilen sonuçlar gösterilmiştir (Tablo 1.1)

Tablo 1.1. Min-max normalleştirilmesi sonucu elde edilen değerleri

X	X^*
25	0,000
36	0,172
60	0,645
75	1,000

1.5.4. Veri Madenciliği Modelleri

Veri madenciliği kullanıldıkları alanlara göre çeşitli modellere ayrılırlar.

1.5.4.1. Değer Tahmini Modeli

Değer tahmin modeli bir insanın öğrenmesine benzeyen bir modeldir. İnsanlar etraflarındaki nesnelere isimlendirirken aslında onları sınıflandırmış olurlar. Bunu yaparken de onlara ait özellikleri yani parametreleri kullanmış olurlar. Örneğin kanatları olan ve uçabilen bir canlıyı kuş olduğu bize öğretilir. Bizde bundan hareketle çeşitli boylarda ya da renklerde olan fakat kanatları olup uçabilen canlıları kuş olarak adlandırırız. Böylelikle bir tahminde bulunmuş ve onu sınıflandırmış oluruz. Veritabanlarında da gerçek hayattaki nesnelere karşılık gelen her bir kayıttır. Tasarladığımız algoritma ya da program nesnelere özelliklerini değerlendirerek bu nesnelere isimlerini tahmin eder ve bunların birbirlerine

olan benzerliklerini ya da farklılıklarını bularak bunları ilişkilendirir. Bu modelde kullanılan en temel iki teknik sınıflandırma ve karar ağaçlarıdır (Silahtaroglu, 2008).

1.5.4.2. Bağlantı Analizi

Tahmini model veri tabanını bir bütün olarak incelerken bağlantı analizi kayıtlar arasında çokça rastlanan kuralları ortaya çıkarır. Bağlantı analizi; birliktelik kuralları, örüntü tanıma, ardışık zaman örüntüleri ve benzer zaman keşfi başlıklarında incelenebilir.

1.5.4.3. Birliktelik Kuralları

Belli türlerdeki ilişkileri tanımlayan bir modeldir. Herhangi bir ürün alındığında bununla beraber alınan başka bir ürünün bilgisini veren bir modeldir. Yani birliktelik kurallarını bulur. Bir süpermarkette satın alınan bir ürünle beraber hangi ürünün alındığı birliktelik kurallarının alanına girer. Örneğin makarna alınan %40 alışverişin %5'i ketçap içerir. Burada %40 güven seviyesini %5 ise destek seviyesini belirtir (Silahtaroglu, 2008).

1.5.4.4. Örüntü Tanıma

Daha önceden şekillendirilmiş çok boyutlu bir örüntünün veri tabanında benzerinin aranmasıdır. Bir nevi sınıflandırma olarak da görülebilir çünkü elimizde hali hazırda bulunan bir örüntünün aynısı veya benzeri aranmaktadır. Parmak izi aramalarında, ses, yüz tanımlarında kullanılabilir. Örüntü tanıma için birçok algoritma geliştirilmiştir. Bunlar: "K-en yakın komşu algoritması", "doğrusal sınıflayıcı" bunlara birkaç örnektir (Silahtaroglu, 2008).

1.5.4.5. Ardışık Zaman Örüntüleri

Örüntü tanımlama modeli ses, parmak izi ya da görüntü gibi örüntülerin kullanıldığı bir modeldir. Buna ek olarak bir müşterinin aldığı ekmek, peynir de bir örüntü oluşturabilir. Süt olan bir müşterinin iki gün sonra ekmek alması ve bir hafta sonra bardak alması da bir örüntü oluşturur. Farklı olarak birbirini izleyen yani zaman içinde ardışık olan bir örüntü oluşturacaktır.

1.5.4.6. Kümeleme Analizi

Kümeleme analizi veri madenciliğindeki önemli alanlardan birisidir. Kümeleme analizi nesnelere birbirine benzerliklerine göre gruplama ayırma işlemidir. Kümeleme analizinde dikkate alınması gereken hususlardan biri sınıfların önceden belli olmamasıdır. Bu yüzden sınıfların belli olmamasından kaynaklı kümeleme analizi ile yapılan öğrenmeye denetimli öğrenme denmektedir. Kayıtlar arasındaki benzerlik sınıf veya etiket ismine dayandırılmayacağı için kayıtlar arasındaki mesafe bulunarak aradaki benzerlik çıkartılır. Kümeleme analizi bilgisayar, matematik ve istatistik gibi alanlarda kullanılır (Silahtaroglu, 2008).

1.6. Birliktelik Kuralları

Veritabanında bulunan kayıtların birbirleri ile olan ilişkilerinin inceleyerek, bu kayıtların birlikte olabilme durumlarını bulan veri madenciliği yöntemine birliktelik kuralları denir.

Birliktelik kuralları başta pazarlama alanı olmak üzere birçok alanda kullanılır. Pazarlama alanında kullanılan sepet analizi adı verilen uygulamalar birliktelik kurallarına dayanmaktadır. Birliktelik kurallarının daha da anlaşılır olması bakımından örnek vermek gerekirse müşterilerin alışveriş alışkanlıkları analiz edilerek müşterinin bir ürünü alırken yanında sepetine koyduğu başka ürünlerde belli bir olasılığa göre ortaya konulur. Böylece müşteri alışkanlıklarına göre mağazanın ya da marketin rafları bu olasılıklara göre düzenlenir. Bu çıkarım iş sahibine satış açısından ciddi avantajlar sağlayacaktır. Aşağıda birliktelik kuralları ile ilgili kavramlar anlatılacaktır.

1.6.1. Destek ve Güven Kavramları

Birliktelik kuralları yöntemi ile olaylar arasındaki ilişkiler bulunurken "destek" ve "güven" gibi iki tane ölçütten faydalanılır. "Destek" ölçütü, bir ilişkinin tüm ilişkiler içinde hangi oranda tekrar ettiğini belirler. "Güven" ölçütü ise örneğin A ürün grubunu alan müşterilerin B ürün grubunu alma olasılığını ortaya koyar. A ürün grubunu alanların B ürün grubunu alma durumu $A \rightarrow B$ formatında gösterilir. "Destek" ve "Güven" ölçütlerini aşağıdaki gibi simgesel olarak göstermek gerekirse:

$$destek(A \rightarrow B) = \frac{sayı(A,B)}{N} \quad (1.23)$$

sayı(A,B) kavramı A ve B ürün gruplarını birlikte içeren ilişkiler anlamına gelir.

$$güven(A \rightarrow B) = \frac{sayı(A,B)}{sayı(A)} \quad (1.24)$$

Birliktelik kuralları bulunurken destek ve güven ölçütlerinin belli bir eşik değeri vardır. Bu eşik değerleri ne kadar yüksek tutulursa kuralların o kadar güçlü olacağı anlamına gelir.

Yukarıdaki ölçütleri basit bir örnekle açıklayalım:

Bir markette 20 müşterinin bir defada yaptığı alışveriş bilgilerini kullanarak birliktelik kurallarına örnek verelim.

güven (makarna, yağ → ketçap)

Yukarıdaki kural bize makarna ve yağ alan müşterilerin ketçap alma olasılığını gösterir. Bunun destek oranını bulmak demek, bu üç ürünün birlikte satın alınma olayının; örneğin bu 5 ise bunun müşteri sayısına (tüm alışveriş sayısı) bölünmesi ile bulunur. Denklem (1.3) uygulanırsa:

$$\begin{aligned} destek(makarna, yağ \rightarrow ketçap) &= \frac{sayı(makarna, yağ, ketçap)}{müşteri sayısı} \\ &= \frac{5}{20} = 0,25 = \%25 \end{aligned}$$

Güven ölçütü bulunurken de bu iki ürünü birlikte satın alma sayısının 10 olduğunu düşünersek, makarna, yağ ve ketçap ürünlerinin aynı anda alınma sayısının makarna ve

yağ ürünlerinin aynı anda alınma sayısına oranı bize güven ölçütünü verir. Denklem (1.4) uygulanırsa:

$$\begin{aligned} \text{güven}(\text{makarna}, \text{yağ} \rightarrow \text{ketçap}) &= \frac{\text{sayı}(\text{makarna}, \text{yağ}, \text{ketçap})}{\text{sayı}(\text{makarna}, \text{yağ})} \\ &= \frac{5}{10} = 0.5 = \%50 \end{aligned}$$

1.6.2. Apriori Algoritması

Birliktelik kurallarının bulunmasında kullanılan en yaygın yöntemlerden biri apriori algoritmasıdır. Burada bu algorithmadan kısaca bahsedilerek mevcut uygulamaların daha da anlaşılması amaçlanmıştır.

Apriori Algoritmasının Aşamaları:

- a) Öncelikle destek ve güven ölçütlerinin eşik değerleri belirlenir. Elde edilen değerlerin bu eşik değerlerine eşit ya da büyük olması beklenir.
- b) Veritabanının bütününe bakılarak ürünlerin destek oranları bulunur. Eşik değerinin altında bulunan satırlar çıkarılır.
- c) Yukarıda anlatılan adımlardan sonra ürünler bu sefer ikişerli gruplandırılıp destek oranları elde edilir. Eşik değerin altında olanlar çıkartılır.
- d) Bu işlem üçerli, dörderli ve artan grup sayıları ile tekrarlanır ve eşik değerlere uygun olduğu sürece tekrarlanır.
- e) Daha sonra destek oranlarına bakılarak birliktelik kuralları çıkartılır ve bu kuralların her biri ile ilgili güven oranları hesaplanır.

2. MATERYAL ve METOT

Bu tez çalışmasında YAA esas alınarak birliktelik kurallarının keşfi için Visual C# programından yararlanılmıştır. Bu program, UCI veri ambarında bulunan New Thyroid (URL-4, 2013) veritabanına ve Bilkent University Function Approximation Repository veri ambarından alınan Basketball (URL-1, 2013), Quake (URL-3, 2013) ve Stock Prices (URL-2, 2013) veritabanlarına uygulanarak birliktelik kurallarının keşfi uygulaması gerçekleştirilmiştir. Bu veritabanlarının içerdiği tüm veriler numeriktir.

Bölüm 1.5.'te anlatıldığı üzere veri tabanları tutarlı olmayabilir. Örneğin birçok veri tekrar ediyor olabilir. Ya da bir veritabanında yaş ve doğum tarihi birlikte girilmiş olabilir. Tüm bu eksikliklerin giderilmesi için veritabanlarının belirli işlemlerden geçirilmesi gerekmektedir. Bu işlemler sırasıyla verileri temizleme, veri bütünleştirme, veri indirgeme ve veri dönüştürmedir. Bu çalışmada kullanılan veritabanlarında herhangi eksik veya bozuk veri olmadığından bu veritabanları olduğu gibi çalışmada kullanılmıştır.

Bu bölümde öncelikle uygulamanın daha iyi anlaşılabilmesi için kullanılan veritabanları anlatılmıştır. Veritabanlarından örnek kesitler sırası ile gösterilmiş ve program için önemli olan veritabanındaki niteliklerin aldıkları en küçük değer ve en büyük değer tablolarında verilmiştir. Daha sonra bu veritabanları dikkate alınarak birliktelik kuralları için nasıl bir kodlama yapıldığı ve YAA'nın birliktelik kuralları için nasıl kullanıldığı; bunun yapılırken hangi yöntemlerin izlendiği anlatılmıştır. Genelde YAA'ya bağlı kalınmıştır fakat bazı noktalarda algoritmanın adımları birebir izlenmemiştir. Bu durum uygulama sırasındaki gerekliliklerden kaynaklanmıştır.

2.1. Kullanılan Veritabanları

Bu tez çalışması sırasında UCI veri ambarından bulunan New Thyroid (URL-4, 2013) ve Bilkent University Function Approximation Repository veri ambarlarından alınan Basketball (URL-1, 2013), Quake (URL-3, 2013) ve Stock Prices (URL-2, 2013) veritabanları kullanılmıştır. Aşağıda bu veritabanlarının özellikleri anlatılmıştır.

2.1.1. Basketball Veritabanı

Basketball veritabanı tamamı gerçek verilerden oluşan bir veritabanıdır. Basketball veritabanı toplamda 96 kayıt içermektedir. Her bir kayıt beş tane nitelikten oluşmaktadır. Basketball veritabanının daha iyi anlaşılabilmesi için aşağıda bu veritabanına ait bir şekil gösterilmiştir (Şekil 2.1).

id	n1	n2	n3	n4	n5
1	0.0888	201	36.02	28	0.5885
2	0.1399	198	39.32	30	0.8291
3	0.0747	198	38.8	26	0.4974
4	0.0983	191	40.71	30	0.5772
5	0.1276	196	38.4	28	0.5703
6	0.1671	201	34.1	31	0.5835
7	0.1906	193	36.2	30	0.5276
8	0.1061	191	36.75	27	0.5523
9	0.2446	185	38.43	29	0.4007
10	0.167	203	33.54	24	0.477
11	0.2485	188	35.01	27	0.4313
12	0.1227	198	36.67	29	0.4909
13	0.124	185	33.88	24	0.5668
14	0.1461	191	35.59	30	0.5113
15	0.2315	191	38.01	28	0.3788
16	0.0494	193	32.38	32	0.559
17	0.1107	196	35.22	25	0.4799
18	0.2521	183	31.73	29	0.5735
19	0.1007	193	28.81	34	0.6318

Şekil 2.1. Basketball veritabanından bir kesit

Algoritmanın veritabanlarına uygulanması sırasında basketball veritabanında ki nitelik sayıları ve o niteliklere ait alt ve üst sınır değerleri belirlenmiştir. Basketball veritabanındaki nitelikler ve o niteliklerin alt ve üst sınır değerleri aşağıda Tablo 2.1'de gösterilmiştir.

Tablo 2.1. Basketball veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler

Nitelikler	Niteliğin aldığı en küçük değer	Niteliğin aldığı en büyük değer
N1	0.0494	0.3437
N2	160	203
N3	10.08	40.71
N4	22	37
N5	0.1593	0.8291

2.1.2. Quake Veritabanı

Quake veritabanı da tamamı gerçek verilerden oluşan bir veritabanıdır. Quake veritabanı toplamda 2178 kayıt bulunmaktadır. Her bir kayıt dört tane nitelikten oluşmaktadır. Quake veritabanının daha iyi anlaşılabilmesi için aşağıda bu veritabanına ait bir şekil gösterilmiştir (Şekil 2.2).

id	n1	n2	n3	n4
1	33	-52.26	28.3	6.7
2	36	45.53	150.93	5.8
3	57	41.85	142.78	5.8
4	67	29.19	141.15	6.2
5	30	-21.66	169.81	6
6	0	23.09	120.58	6.2
7	139	-20.7	169.92	6.1
8	60	22.33	93.58	6
9	50	-13.64	165.96	6
10	119	-16.31	-71.66	6
11	197	36.48	70.95	6.3
12	13	55.72	-155.95	6.1
13	26	55.75	-155.88	5.9
14	58	-5.14	151.77	6
15	33	-44.6	37.34	6
16	0	-44.2	167.96	5.8

Şekil 2.2. Quake veritabanından bir kesit

Algoritmanın veritabanlarına uygulanması sırasında Quake veritabanında ki alt ve üst sınır değerleri belirlenmiştir. Quake veritabanındaki niteliklerin alt ve üst sınır değerleri aşağıdaki Tablo 2.2' de gösterilmiştir.

Tablo 2.2. Quake veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler

Nitelikler	Niteliğin aldığı en küçük değer	Niteliğin aldığı en büyük değer
N1	0	656
N2	-66.49	78.15
N3	180	359.96
N4	6.9	1.1

2.1.3. Thyroid Disease (New Thyroid) Veritabanı

New Thyroid veritabanı toplamda 215 kayıttan oluşmaktadır. Bu kayıtların tamamı gerçek verilerden oluşmaktadır. Her bir kayıt kendi içinde beş ayrı özellikten oluşmaktadır. Aşağıda New Thyroid veritabanının kısmi görüntüsü verilmiştir (Şekil 2.3).

id	t3resin	thyroxin	triiodothyronine	thyroidstimula...	tsh_value
1	107	10.1	2.2	0.9	2.7
2	113	9.9	3.1	2	5.9
3	127	12.9	2.4	1.4	0.6
4	109	5.3	1.6	1.4	1.5
5	105	7.3	1.5	1.5	-0.1
6	105	6.1	2.1	1.4	7
7	110	10.4	1.6	1.6	2.7
8	114	9.9	2.4	1.5	5.7
9	106	9.4	2.2	1.5	0
10	107	13	1.1	0.9	3.1
11	106	4.2	1.2	1.6	1.4
12	110	11.3	2.3	0.9	3.3
13	116	9.2	2.7	1	4.2
14	112	8.1	1.9	3.7	2
15	122	9.7	1.6	0.9	2.2
16	109	8.4	2.1	1.1	3.6
17	111	8.4	1.5	0.8	1.2
18	114	6.7	1.5	1	3.5
19	119	10.6	2.1	1.3	1.1
20	115	7.1	1.3	1.3	2
21	101	7.8	1.2	1	1.7
22	103	10.1	1.3	0.7	0.1
23	109	10.4	1.9	0.4	-0.1

Şekil 2.3. Thyroid Disease (New Thyroid) veritabanından bir kesit

New Thyroid veritabanının değer aralıkları yani özelliklerin alt ve üst sınırları birliktelik kurallarının keşfi sırasında algoritmanın optimum değeri arayacağı aralığı belirlediğinden önemlidir. Bu yüzden bu özelliklerin ismi ve değer aralıkları aşağıdaki Tablo 2.3' te gösterilmiştir.

Tablo 2.3. Thyroid Disease (New Thyroid) veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler.

Nitelikler	Niteliğin aldığı en küçük değer	Niteliğin aldığı en büyük değer
T3resin	65	144
Thyroxin	0.5	25.3
Triiodothyronine	0.2	10
Thyroidstimulating	0.1	56.4
TSH_value	-0.7	56.3

2.1.4. Stock Prices Veritabanı

Stock Prices veritabanı toplamda 950 kayıttan oluşmaktadır. Bu kayıtların tamamı gerçek verilerden oluşmaktadır. Her bir kayıt kendi içinde 10 ayrı nitelikten oluşmaktadır. Aşağıda Stock Prices veritabanının kısmi görüntüsü verilmiştir (Şekil 2.4).

id	n1	n2	n3	n4	n5	n6	n7
1	17.219	50.5	18.75	43	60.875	26.375	67.75
2	17.891	51.375	19.625	44	62	26.125	68.125
3	18.438	50.875	19.875	43.875	61.875	27.25	68.5
4	18.672	51.5	20	44	62.625	27.875	69.375
5	17.438	49	20	41.375	59.75	25.875	63.25
6	18.109	49	19.5	41.875	59.625	26.625	66.25
7	18.563	49.375	19.125	42.5	60.75	27.25	65.75
8	18.672	50.125	19.25	43	61.75	28	66
9	18.563	49.75	19	43.25	61.75	29	65.75
10	19.063	50.5	19.125	43.875	61.875	29.625	66.875
11	19	50.25	19.625	44	62.125	30	66.5
12	19.063	49.75	20	44.375	61.25	29.875	66.5
13	18.719	49.25	19	43.5	60.375	29	65.875
14	18.438	49.25	18.375	43.375	60.375	29	65
15	19.063	50.25	18.375	43.5	60.375	29.125	65.75
16	20	50.25	18.125	44	60.75	30	67

Şekil 2.4. Stock Prices veritabanından bir kesit

Stock Prices veritabanının değer aralıkları yani özelliklerin alt ve üst sınırları birliktelik kurallarının keşfi sırasında algoritmanın optimum değeri arayacağı aralığı belirlediğinden önemlidir. Bu yüzden bu özelliklerin ismi ve değer aralıkları aşağıdaki Tablo 2.4'te gösterilmiştir.

Tablo 2.4. Stock Prices veritabanındaki nitelikler ve niteliklere ait alt ve üst değerler

Nitelikler	Niteliğin aldığı en küçük değer	Niteliğin aldığı en büyük değer
N1	17.219	61.5
N2	19.25	60.25
N3	12.75	25.125
N4	34.375	60.125
N5	27.75	94.125
N6	14.125	35.25
N7	58	87.25
N8	16.375	29.25
N9	31.5	53
N10	34	62

2.2. YAA ile Geliştirilen Uygulama

Bu bölümde YAA ile geliştirilen uygulamanın adımları ve nasıl yapıldığı anlatılacaktır. Detaylı anlatıma geçmeden önce YAA algoritmasının işleyiş basamakları bölümün sağlıklı takip edilmesi açısından bir kez daha gösterilmiştir. Tez çalışmasında bu işleyiş genellikle sadık kalmış fakat programın daha iyi çalışması açısından çeşitli eklemelerde yapılmıştır. YAA algoritmasının işleyiş basamakları:

- 1- Başlangıç popülasyonunu üret
- 2- Her bir bireyin uygunluğunu hesapla
- 3- G 'yi ve popülasyonun en iyi ve en kötüsünü güncelle
- 4- Her bir birey için M ve a değerlerini hesapla
- 5- Hızı ve popülasyonu güncelle
- 6- Sonlandırma kriteri ile karşılaşırsa programı sonlandır ve en iyi çözümü geri döndür değilse başa geri dön.

2.2.1. Başlangıç Popülasyonunun Oluşturulması

Birliktelik kuralı bir eylemin birlikte olma durumunu araştırdığından buna uygun kodlama yapılmıştır. Başlangıçta rastgele bir başlangıç popülasyonu oluşturmuştur.

YAA'nın uygulandığı bu programda, Basketball, Quake ve New Thyroid veritabanları için başlangıç popülasyonu 200 olarak belirlenmiştir. Fakat Stock Prices veritabanında nitelik sayılarının fazla olmasından dolayı daha iyi sonuçlar almak için başlangıç popülasyonu 300 olarak belirlenmiştir. Herbir birey bir aday kuralı temsil etmektedir. Yani başka bir deyişle 200 veya 300 tane rastgele aday kural elde edilmiştir. Fakat bu yapılmadan önce algoritmanın uygulanacağı veritabanı içerisindeki her bir kayıttaki nitelik sayısı dikkate alınmıştır. Örneğin uygulanan veritabanında her bir kayıt beş tane nitelikten oluşuyor ise oluşturulan aday kuralda beş nitelikten oluşmalıdır. Aşağıda kullanılan kodlama şekli ve buna ait bir birey gösterilmiştir (Şekil 2.5).

Nitelik 1			Nitelik 2			Nitelik 3			Nitelik 4			Nitelik 5		
0	40	70	1	20	90	1	80	175	0	60	200	2	126	400

Şekil 2.5. Örnek birliktelik kuralı aday

Programda oluşturulan bu birey aday bir kuralı temsil etmekte ve YAA'da bir bireyin pozisyonunu belirtmektedir. Yukarıda oluşturulan birey New Thyroid veritabanına göre oluşturulduğundan veritabanındaki nitelik sayısı olan beşe göre oluşturulmuştur. Her bir nitelik üç boyuta bölünmüştür. Yani her üçlü bölüm bir niteliğe ait sırasıyla kuralda var olma durumunu, alt sınırını ve üst sınırını belirtmektedir. Aday kuralda üretilen bu alt ve üst sınırların veritabanındaki gerçek alt ve üst sınırları geçmemesi için sınır kontrolleri yapılmıştır. Buna ek olarak "0" sayısı o niteliğin kuralın dışında olduğunu, "1" sayısı o niteliğin kuralın sol tarafında olduğunu ve "2" sayısı ise o niteliğin kuralın sağ tarafında olduğunu göstermektedir. Yani "1" sayısına sahip nitelikler solda "2" sayısına sahip nitelikler sağda olacak anlamındadır. Yukarıdaki kural şu şekilde gösterilir (Şekil 2.6).

$$20 < \text{Nitelik 2} < 90 \wedge 80 < \text{Nitelik 3} < 175 \Rightarrow 126 < \text{Nitelik 5} < 400$$

Şekil 2.6. Birliktelik kuralının ifadesi

Yani nitelik 2 ve 3 belirtilen aralıkta iken nitelik 5 ise belirtilen aralıktadır. Yukarıda ki kodlama şekli programın temelini oluşturur ve programın sağlıklı işlemesi açısından önemlidir. Programda oluşturulan gerçek aday kurallara aşağıda örnek verilmiştir (Şekil 2.7).

```
0) 1 112 120 2 0.748 6.14944 2 7.6 9.16 1 46.266 46.57002 0 5.57 21.8036
1) 2 118 137 2 6.7 14.512 1 8.9 9.043 0 16.99 52.459 0 19.25 24.437
2) 0 125 139 1 22.324 24.76432 0 5.6 9.56 2 34.443 37.29741 2 37.49 47.2712
3) 0 114 115 1 24.804 25.09664 0 9.3 9.587 1 55.837 56.24236 0 19.25 45.5555
4) 0 130 135 1 12.9 13.52 0 6.3 7.114 1 31.628 34.84836 2 47.75 53.222
5) 0 124 128 1 1.244 17.60208 2 4.6 6.598 1 49.644 55.3866 0 1.01 50.2181
6) 0 70 83 0 13.892 22.79024 2 2.6 3.784 1 0.663 1.22037 2 3.86 25.8848
7) 1 96 98 2 3.476 8.71376 1 5.4 6.872 0 27.124 38.24888 0 54.02 55.4108
8) 0 139 141 0 10.42 17.5624 2 6.6 7.62 2 29.939 43.1695 0 6.14 7.1432
9) 1 65 93 2 2.732 13.56464 2 5.7 9.613 1 32.191 45.26386 2 8.42 11.2928
10) 1 123 129 1 7.444 8.51536 1 5.1 8.383 2 27.687 49.22175 2 51.74 54.3848
11) 2 103 119 0 9.428 16.88784 0 0.8 8.712 0 22.62 35.4564 1 38.06 53.0168
12) 2 72 106 0 11.412 22.5224 0 5 9.2 2 12.486 34.443 2 -0.13 33.728
13) 2 67 115 0 23.068 23.24656 0 5.8 9.16 1 21.494 44.88102 1 22.67 51.9281
14) 2 126 140 2 25.052 25.15368 0 5.4 7.608 2 28.813 28.813 1 18.68 23.5706
15) 2 91 93 0 10.42 15.7768 2 3.5 8.505 2 44.014 50.45472 2 35.78 36.806
16) 0 127 128 2 7.196 15.16176 2 8.5 8.515 0 25.435 41.22715 0 30.08 40.3058
17) 0 113 119 1 15.628 16.1116 1 1.8 1.882 1 51.896 53.87776 0 5 15.773
18) 2 112 140 1 16.62 22.8696 2 5.9 6.146 2 15.864 35.72664 0 2.72 53.0852
19) 2 112 137 1 16.62 18.0956 0 0.3 0.494 0 28.25 45.703 0 25.52 35.9852
20) 0 116 136 2 7.94 22.8696 1 7.5 9.7 1 8.545 9.98065 0 54.59 55.1372
21) 1 111 114 0 16.62 19.3976 1 6.8 8.24 0 32.754 46.46868 2 22.67 27.3782
```

Şekil 2.7. Programda oluşturulan gerçek adaylar,

İlerleyen aşamalarda program, bu aday kuralları kullanarak veritabanındaki kayıtları bu aday kurallarla algoritmanın işleyişi içinde karşılaştırarak en uygun kuralları bulmaktadır.

2.2.2. Her Bir Bireyin Uygunluğunun Hesaplanması

Rastgele oluşturulan her bir bireyin uygunlukları bu çalışmada esnek bir uygunluk fonksiyonu dikkate alınarak hesaplanmıştır. Bulunan uygunluk değerleri en iyiden başlamak üzere en kötüye doğru sıralanmış ve bu değerler kullanılarak algoritmada her bir bireyin kütle M olarak gösterilen değerleri hesaplanmıştır. Sonuç olarak program bittiğinde en yüksek uygunluk değerine sahip olan kural bizim en iyi kuralımızı temsil eder. Aşağıda programda kullanılan uygunluk fonksiyonu gösterilmiştir (Şekil 2.8).

$$\text{Uygunluk Fonksiyonu} = \text{Destek} * \text{alfa1} + \text{Güven} * \text{alfa2} - \text{Ns} * \text{alfa3} + \text{alfa4} * \text{Kuraloranı} - \text{alfa5} * \text{Aralık}$$

Şekil 2.8. Uygunluk fonksiyonu

Yukarıda uygunluk fonksiyonunda kullanılan güven ve destek kavramları 1.6.1. Destek ve Güven Kavramları adlı bölümde anlatılmıştır. alfa1 , alfa2 , alfa3 , alfa4 ve alfa5 parametre değerleri toplamda 1 olacak şekilde seçilmiştir. Ns bir kayıttaki "1" ve "2" 'lerin sayısını belirtmektedir. Uygunluk fonksiyonunda kural oranı olarak belirtilen parametre ise

kötü kuralları cezalandırmak yani uygunluklarını düşürmek ve iyi kuralları ise ödüllendirmek için kullanılmıştır. Böylelikle sadece 1'lerden ya da sadece 2'lerden oluşan kuralların uygunlukları her iterasyonda düşürülmüştür. Birliktelik kurallarında sadece 1'lerden ya da 2'lerden oluşan kurallar hiçbir anlam ifade etmemekte olur birliktelik kavramına terstir. Aralık ise bulmak istediğimiz kurallardaki niteliklere ait aralıkların küçük olmasını sağlayacak bir formüldür. Amaç aralıkları, küçük bir eksi değerdeki katsayı ile çarparak daha küçük aralıklardaki birliktelik kurallarını bulmaktır. Ayrıca uygunluk fonksiyonundaki aralık değeri, kuraldaki aralıkların farkının, gerçek üst değerinden gerçek alt değerinin farkına bölünmesi ile bulunur.

2.2.3. *G*'yi ve Popülasyonun En İyi ve En Kötüsünü Güncelle

YAA'nın bu basamağında *G*'nin ve popülasyonun en iyi ve en kötüsünün güncellenmesi olayı bu tez çalışmasında YAA'nın işleyiş basamaklarındaki sıra takip edilerek gerçekleştirilmemiştir. Programın başlangıcında, başlangıç popülasyonun yani aday kuralların rastgele oluşturulmasından sonra popülasyonun en iyi ve en kötü uygunluk değerlerine sahip olanları hesaplanmıştır. Daha sonra her iterasyon da en iyi ve en kötü uygunluk değerlerine sahip olanlar güncellenmiştir. Yerçekimi sabiti olan *G* değişkeni ise zamanla azalan bir fonksiyon olarak düşünülmüştür. Algoritmada ilk iterasyonda 1 olarak alınmış ve her iterasyonda 0.002 düşürülmüştür.

2.2.4. Her bir birey için *M* ve *a* değerlerini hesapla

Algoritmada kütle yani *M* olarak geçen değerler her bir kayıt için yukarıda, 2.2.2. Her Bir Bireyin Uygunluğunun Hesaplanması bölümünde gösterilen uygunluk fonksiyonu kullanılarak hesaplanır. Fakat büyük *M*'ler hesaplanmadan önce küçük *m*'lerin hesaplanması gerekmektedir. Algoritmada küçük *m* harfi ile gösterilen *m*'in hesaplanması 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formülle (1.15) yapılır.

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$$

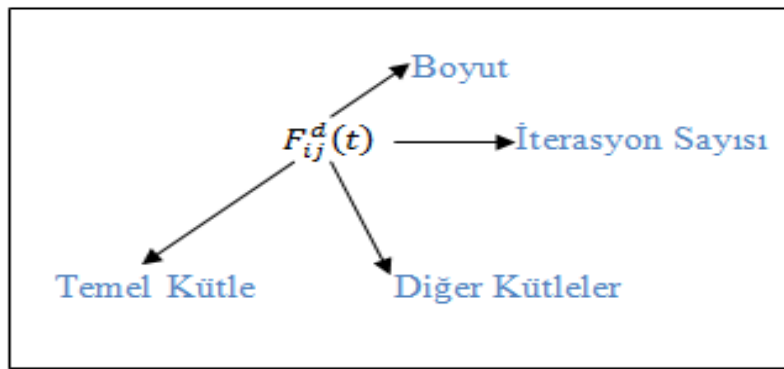
i'ninci bireyin m_i 'i yukarıdaki formüle göre hesaplanır. Bundan sonraki aşamada ise bireyin büyük M değeri, 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen formüle (1.16) göre hesaplanmıştır.

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}$$

Her bir bireyin kütlelerinin hesaplanmasından sonra algoritmaya uygun olarak her bireye etki eden toplam kuvvet aşağıdaki formüle göre hesaplanır. Burada dikkat edilmesi gerek husus her bir bireyin her bir boyutunun ayrı ayrı dikkate alınarak, 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formüle (1.7) göre bu kuvvetin hesaplanmış olmasıdır:

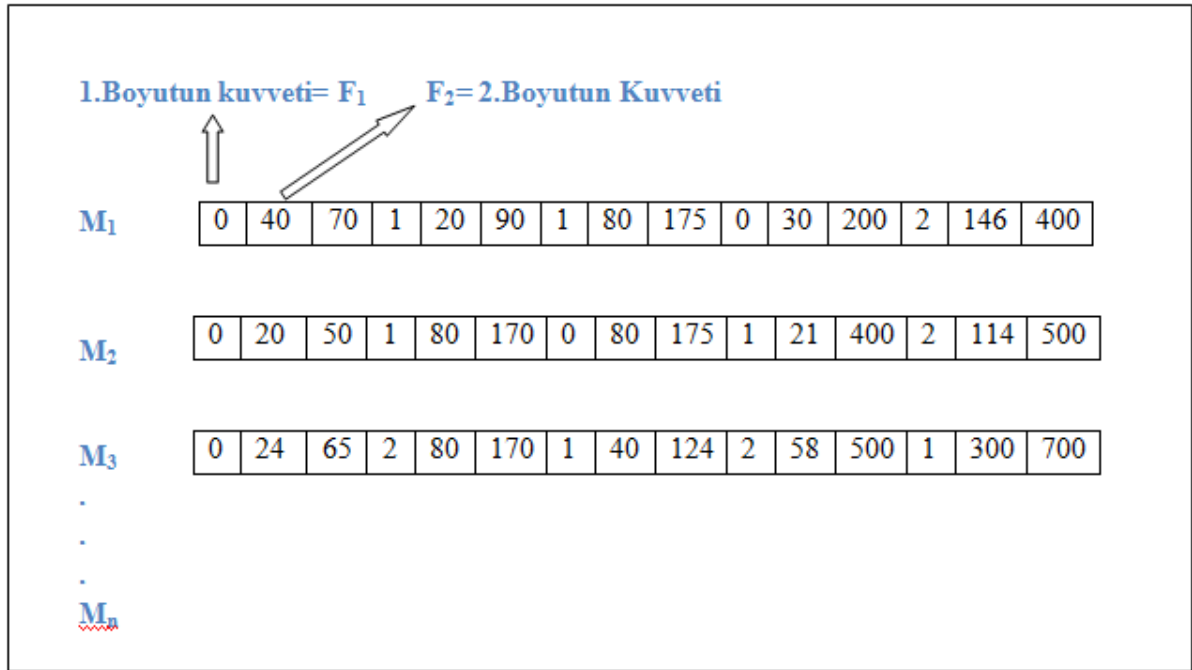
$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij(t)} + \varepsilon} (x_j^d(t) - x_i^d(t))$$

Aşağıda algoritmanın işleyiş mantığının anlaşılması için şekil üzerinde fizik kuralındaki kavramların kullandığımız algoritmada nelere denk geldiği gösterilmiştir (Şekil 2.9).



Şekil 2.9. $F_{ij}^d(t)$ 'nin programdaki karşılığı

Burada aslında boyut bazlı bir hesaplama yapıldığından işlemler oldukça karmaşıktır. Yukarıdaki formülde (1.7), d ile boyut temsili yapılmaktadır. Bu durum daha da anlaşılır olması için aşağıdaki şekil ile gösterilmiştir (Şekil 2.10).



Şekil 2.10. Kütle ve boyutların gösterilmesi

Şekil 2.10'da herbir bireye etki eden kuvvet hesaplanırken birinci bireyin ilk boyutuna etki eden kuvvet F_1 olarak bulunur. İlk bireyin onbeş boyuttan oluştuğu düşünülürse toplamda onbeş tane kuvveten bahsedilir. Bulunan bu F kuvveti algoritmaya olasılıksal bir karakter kazandırmak için rastgele (0-1 aralığında) bir sayı ile çarpılmıştır. Bu olasılıksallık 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formüle (1.9) göre yapılmıştır:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t),$$

Daha sonra i 'ninci bireye etki eden toplam kuvvet bulunur. Bu kuvvet kullanılarak algoritmada a simgesi ile gösterilen ivme her bir boyuta göre ayrı ayrı hesaplanır. Bu hesaplamada yukarıda 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formüle (1.10) göre yapılmıştır:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$$

2.2.5. Hızı ve popülasyonu güncelle

Her bir boyutun ivmesi yukarıda ki formüle göre hesaplanır. Her bir boyutun ivmesi hesaplandıktan sonra her bir boyutun hızı hesaplanır. Hız hesaplanması yukarıda 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formüle (1.11) göre yapılmıştır:

$$v_i^d(t + 1) = rand_i \times v_i^d(t) + a_i^d(t)$$

Burada ilk hız değeri "0" olarak alınmıştır. Böylece boyutların hızları hesaplanmış olur. Hesaplanan bu hız değerleri daha sonra yeni pozisyonların hesaplanmasında kullanılır. Yeni pozisyonların hesaplanması da yukarıda 1.4.12.2. Yerçekimsel Arama Algoritması (YAA) bölümünde gösterilen şu formüle (1.12) göre yapılır:

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1)$$

Böylelikle her bir pozisyondaki değerler değiştirilmiş olur.

2.2.6. Sonlandırma kriteri ile karşılaşırsa programı sonlandır ve en iyi çözümü geri döndür değilse başa geri dön.

Bu tez çalışmasında sonlandırma kriteri olarak iterasyon sayısı esas alınmıştır. Bu programda kullanılan iterasyon sayısı Basketball, Quake ve New Thyroid veritabanlarında 100 olarak belirlenmiştir. Fakat Stock Prices veritabanında nitelik sayısının fazla olması küçük iterasyon değerlerinde kötü sonuçlar verdiğiinden sonlandırma kriteri olan iterasyon sayısı ilk denemede 200 ve sonraki demede 300 olarak belirlenmiştir. Program iterasyon sayısı kadar çalıştıktan sonra en iyi uygunluk değerine sahip birliktelik kuralını döner ve sonlanır. Elde edilen sonuç bizim en iyi uygunluk değerine sahip sonucumuzdur.

3. BULGULAR

Bu bölümde tez çalışması sonucunda elde edilen sonuçlar programın uygulandığı veritabanlarına göre sırası ile verilmiştir. İlk başta Basketball veritabanı (URL-1, 2013) sonuçları olmak üzere sırası ile Quake (URL-3, 2013), Thyroid Disease (New Thyroid) (URL-4, 2013) ve Stoke Prices (URL-2, 2013) veritabanlarından elde edilen sonuçlar gösterilmiştir. Bu sonuçlar uygunluk fonksiyonunda kullanılan iki farklı *alfa1*, *Alfa2*, *Alfa3* ve *Alfa4* parametre değerleri için gösterilmiştir. Ayrıca kullanılan bu katsayılar sırası ile destek, güven, kayıttaki toplam nitelik sayısı ve kural oranı gibi değişkenlerin çarpıldığı katsayı parametreleridir. Uygulama sonucunda elde edilen kurallara birer tane örnek verilerek sonuçların daha iyi anlaşılması amaçlanmıştır.

3.1. Basketball Veritabanı Sonuçları

3.1.1. İlk Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıda Tablo 3.1.'de gösterilmiştir.

Tablo 3.1. Basketball veritabanı için kullanılan ilk parametreler

Kullanılan Parametre Değerleri	
Alfa1	0.3
Alfa2	0.1
Alfa3	0.1
Alfa4	0.4

YAA kullanılarak 100 iterasyon sonucunda Basketball veritabanından elde edilen sonuçlar aşağıda Tablo 3.2.'de gösterilmiştir.

Tablo 3.2. Basketball veritabanı ilk parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	60/73	%82
2.Çalıştırmada	1	5/5	
Toplamda	2	60/73	

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması

1.Çalıştırma sonucunda elde edilen kural şudur:

1 0.075887 0.21514976 **2** 175.05 196.292 **0** 26.0076 34.240944 **1** 22 36.25 **0** 0.25977
0.4590355

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$0.075887 < N1 < 0.21514976 \wedge 22 < N4 < 36.25 \rightarrow 175.05 < N2 < 196.292$

3.1.2. İkinci Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıda Tablo 3.3.'te gösterilmiştir.

Tablo 3.3. Basketball veritabanı için kullanılan ikinci parametreler

Kullanılan Parametre Değerleri	
alfa1	0.35
alfa2	0.25
alfa3	0.05
alfa4	0.25

YAA kullanılarak 100 iterasyon sonucunda Basketball veritabanından elde edilen sonuçlar aşağıda Tablo 3.4.'de gösterilmiştir.

Tablo 3.4. Basketball veritabanı için ikinci parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	60/62	%96
2.Çalıştırmada	1	4/4	
Toplamda	2	64/66	

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması

1.Çalıştırma sonucunda elde edilen kural şudur:

1 0.099431 0.2704193 **1** 183.65 196.034 **0** 22.0257 37.159983 **0** 35.35 36.8845 **2** 0.219582
0.7986241

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

0.099431 < N1 < 0.2704193 \wedge 183.65 < N2 < 196.034 \rightarrow 0.219582 < N5 < 0.7986241

3.2. Quake Veritabanı Sonuçları

3.2.1. İlk Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıdaki Tablo 3.5.'te gösterilmiştir.

Tablo 3.5. Quake veritabanı için kullanılan ilk parametreler

Kullanılan Parametre Değerleri	
alfa1	0.35
alfa2	0.25
alfa3	0.05
alfa4	0.25

YAA kullanılarak 100 iterasyon sonucunda Quake veritabanında elde edilen sonuçlar aşağıda Tablo 3.6.'da gösterilmiştir

Tablo 3.6. Quake veritabanı ilk parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	68/72	
2.Çalıştırmada	1	778/819	
3.Çalıştırmada	1	524/616	
4.Çalıştırmada	1	237/238	
5.Çalıştırmada	1	89/146	
Toplamda	5	1696/1891	%89

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

4. Çalıştırma sonucunda elde edilen kural şudur:

0 322.1 489.1 **2** 28.9724 78.15 **1** -169.1612 -141.228304 **0** 6.394 6.80386

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$-169.1612 < N1 < -141.228304 \rightarrow 28.9724 < N2 < 78.1$

3.2.2. İkinci Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıdaki Tablo 3.7.'de gösterilmiştir

Tablo 3.7. Quake veritabanı için kullanılan ikinci parametreler

Kullanılan Parametre Değerleri	
alfa1	0.3
alfa2	0.2
alfa3	0.1
alfa4	0.3

YAA kullanılarak 100 iterasyon sonucunda Quake veritabanında elde edilen sonuçlar aşağıda Tablo 3.8'de gösterilmiştir

Tablo 3.8. Quake veritabanı ikinci parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	172/235	
2.Çalıştırmada	1	987/991	
3.Çalıştırmada	1	341/391	
4.Çalıştırmada	1	40/41	
5.Çalıştırmada	1	77/90	
6.Çalıştırmada	1	5/5	
7.Çalıştırmada	1	34/63	
8.Çalıştırmada	1	6/6	
Toplamda	8	1662/1882	%88

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

2. Çalıştırma sonucunda elde edilen kural şudur:

0 310.1 455.1 **2** -46.2404 78.15 **1** 75.6116 153.9029 **0** 6.262 6.84258

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$75.6116 < N3 < 153.9029 \rightarrow -46.2404 < N2 < 78.15$

3.3. Thyroid Disease (New Thyroid) Veritabanı Sonuçları

3.3.1. İlk Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıdaki Tablo 3.9.'da gösterilmiştir.

Tablo 3.9. Thyroid Disease (New Thyroid) veritabanı için kullanılan ilk parametreler

Kullanılan Parametre Değerleri	
alfa1	0.3
alfa2	0.1
alfa3	0.1
alfa4	0.4

YAA kullanılarak 100 iterasyon sonucunda Thyroid Disease veritabanında elde edilen sonuçlar aşağıda Tablo 3.10.'da gösterilmiştir.

Tablo 3.10. Thyroid Disease (New Thyroid) veritabanı ilk parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	152/177	
2.Çalıştırmada	1	23/30	
Toplamda	2	175/207	%84

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

4. Çalıştırma sonucunda elde edilen kural şudur:

0 130 134 **0** 17.364 19.03056 **2** 1 3.43 **0** 2.915 28.05295 **1** -0.13 8.3345

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$$-0.13 < N5 < 8.3345 \rightarrow 1 < N3 < 3.43$$

3.3.2. İkinci Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıda Tablo 3.15.'te gösterilmiştir.

Tablo 3.11. Thyroid Disease (New Thyroid) veritabanı için kullanılan ikinci parametreler

Kullanılan Parametre Değerleri	
alfa1	0.3
alfa2	0.2
alfa3	0.1
alfa4	0.3

YAA kullanılarak 100 iterasyon sonucunda Thyroid Disease veritabanında elde edilen sonuçlar aşağıda Tablo 3.12.'de gösterilmiştir.

Tablo 3.12. Thyroid Disease (New Thyroid) veritabanı ikinci parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	171/194	
2.Çalıştırmada	1	1/1	
Toplamda	2	172/195	%88

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

1. Çalıştırma sonucunda elde edilen kural şudur:

$$1 \ 91 \ 136 \ 0 \ 8.932 \ 22.8448 \ 2 \ 1.1 \ 8.754 \ 0 \ 30.502 \ 32.83282 \ 0 \ 14.69 \ 15.1061$$

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$$91 < N1 < 136 \rightarrow 1.1 < N3 < 8.754$$

3.4. Stoke Prices Veritabanı Sonuçları

3.4.1. İlk Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıda Tablo 3.13.'de gösterilmiştir.

Tablo 3.13. Stoke Prices veritabanı için kullanılan ilk parametreler parametreler

Kullanılan Parametre Değerleri	
alfa1	0.4
alfa2	0.3
alfa3	0.02
alfa4	0.18

YAA kullanılarak 200 iterasyon sonucunda Stoke Prices veritabanında elde edilen sonuçlar aşağıda Tablo 3.14.'te gösterilmiştir.

Tablo 3.14. Stoke Prices veritabanı ilk parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	247/468	
2.Çalıştırmada	1	234/234	
Toplamda	2	481/702	%69

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

2. Çalıştırma sonucunda elde edilen kural şudur:

0 25.63239 61.5 **0** 40.98 52.7347 **1** 19.55625 22.563375 **0** 37.7225 41.75495 **0** 42.3525
49.60065 **1** 22.36375 33.3170625 **0** 58.8775 81.00805 **0** 20.36625 20.36625 **2** 40.745
51.89705 **1** 36.52 50.534

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

$19.55625 < N3 < 22.563375 \wedge 22.36375 < N6 < 33.3170625 \wedge 22.36375 < N10 < 33.3170625 \rightarrow 28.9724 < N9 < 78.1$

3.4.2. İkinci Parametre Değerleri İçin Sonuçlar

Uygunluk fonksiyonunda kullanılan parametreler aşağıda Tablo 3.15.'te gösterilmiştir.

Tablo 3.15. Stoke Prices veritabanı için kullanılan ikinci parametreler

Kullanılan Parametre Değerleri	
alfa1	0.4
alfa2	0.3
alfa3	0.05
alfa4	0.15

YAA kullanılarak 300 iterasyon sonucunda Stoke Prices veritabanında elde edilen sonuçlar aşağıda Tablo 3.16.'da gösterilmiştir.

Tablo 3.16. Stoke Prices veritabanı ikinci parametre sonuçları

	Bulunan Kural Sayısı	Toplam doğru veri sayısı / tüm verilerin sayısı	Elde edilen başarı yüzdesi
1.Çalıştırmada	1	628/644	
2.Çalıştırmada	1	3/3	
3.Çalıştırmada	1	75/77	
Toplamda	3	706/724	%97

Uygulama sonucunda elde edilen örnek birliktelik kuralı ve açıklaması:

1. Çalıştırma sonucunda elde edilen kural şudur:

1 28.73206 61.5 **0** 47.54 55.166 **0** 18.9375 21.78375 **2** 34.89 59.1156 **0** 64.92 81.8589 **0**
24.265 25.8029 **0** 85.7875 85.7875 **0** 22.04 22.04 **0** 51.28 52.8108 **1** 37.36 58.7968

Yukarıda elde edilen birliktelik kuralından şu sonuç çıkarılır:

28.73206 < N1 < 61.5 \wedge 37.36 < N10 < 58.7968 \rightarrow 34.89 < N4 < 59.1156

4. SONUÇLAR

“Yerçekimsel Arama Algoritmasıyla Nicel Birliktelik Kurallarının Keşfi” başlıklı çalışma kapsamında, algoritmanın Basketball, Quake, Thyroid Disease (New Thyroid) ve Stoke Prices veritabanlarına uygulanmasından sonra elde edilen sonuçlar kullanılan farklı uygunluk fonksiyonu parametre değerlerine ve programın çalışma sayısına göre aşağıda verilmiştir.

1. Basketball veritabanında ilk parametre (Tablo 3.1.) değerlerine göre programın iki kez çalıştırılması sonucu iki kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %82 olduğu belirlenmiştir.
2. Basketball veritabanında ikinci parametre (Tablo 3.3.) değerlerine göre programın iki kez çalıştırılması sonucu iki kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %96 olduğu belirlenmiştir.
3. Quake veritabanında ilk parametre (Tablo 3.5.) değerlerine göre programın 5 kez çalıştırılması sonucu 5 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %89 olduğu belirlenmiştir.
4. Quake veritabanında ikinci parametre (Tablo 3.7.) değerlerine göre programın 5 kez çalıştırılması sonucu 8 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %88 olduğu belirlenmiştir.
5. Thyroid Disease veritabanında ilk parametre (Tablo 3.9.) değerlerine göre programın 2 kez çalıştırılması sonucu 2 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %84 olduğu belirlenmiştir.
6. Thyroid Disease veritabanında ikinci parametre (Tablo 3.11.) değerlerine göre programın 2 kez çalıştırılması sonucu 2 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %88 olduğu belirlenmiştir.
7. Stoke Prices veritabanında ilk parametre (Tablo 3.13.) değerlerine göre programın 2 kez çalıştırılması sonucu 2 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %69 olduğu belirlenmiştir.
8. Stoke Prices veritabanında ikinci parametre (Tablo 3.14.) değerlerine göre programın 3 kez çalıştırılması sonucu 3 kural elde edilmiş ve bu kuralların başarı oranının kapsadığı veriler baz alındığında %97 olduğu belirlenmiştir.

5. TARTIŞMA

Numerik verilerden oluşan veritabanlarında birliktelik kurallarının keşfi oldukça kompleks bir problemdir. YAA ile numerik verilerden oluşan veritabanlarında nicel birliktelik kurallarının keşfi ilk kez bu çalışmada yapılmıştır. Kullanılan veriler üzerinde temizleme, ayrıştırma vb. herhangi bir işlem yapılmamıştır. Buna rağmen elde edilen sonuçlar oldukça umut vericidir. Ayrıca yapılan çalışma sırasında her veritabanı için ayarlanması gereken minimum destek ve güven değerlerinin belirlenme ihtiyacı kaldırılmıştır. Yani minimum ve maksimum destek ve güven değerlerinden bağımsız olarak algoritma uygulanmıştır. Bunun dışında kullanılan uygunluk fonksiyonu esnektir. Karşılaşılan probleme göre herhangi bir parametre eklenip çıkartılabilir. Aralık değerlerinin belirlenmeside otomatik olarak program tarafından belirlenmiştir.

Bizim yaptığımız çalışmaya yakın literatürde bir çalışma vardır. YAA ile birliktelik kurallarının keşfi çalışması, Khandemolghorani vd. (2011) tarafından yapılmıştır. Fakat bu çalışma kategorik veriler üzerinde yapılan bir çalışmadır. Khandemolghorani vd. (2011) tarafından yapılan çalışma nispeten daha kolay bir çalışmadır. Bizim yaptığımız çalışma tamamen numerik veriler üzerinde yapılmış ve oldukça kompleks bir probleme tekabül etmektedir.

6. ÖNERİLER

Bu tez kapsamında, YAA ile nicel birliktelik kuralları keşfedilmeye çalışılmıştır. Tamamen numerik verilerle çalışılmış ve elde edilen sonuçlar paylaşılmıştır. YAA ile yapılan bu çalışmada uygunluk fonksiyonuna ilginçlik vb. farklı amaçlar esnek bir şekilde eklenebilir. Bu çalışmanın, kurallar bulunduğundan sonra destek ve güven değeri azalmayacak şekilde kuralların içerdikleri niteliklerin aralıklarının azaltılması işlemi ile daha kaliteli kuralları bulması sağlanabilir. Ayrıca tek çalıştırmada çakışan farklı amaçları aynı anda optimize edecek şekilde kurallar kümesinin bulunmasıyla çok amaçlı hale getirilebilir. Ayrıca YAA, veri madenciliğinin diğer alt dalları olan kümeleme kural madenciliği, sınıflandırma kural madenciliği ve ardışık örüntü madenciliği gibi alanlarda da etkili şekilde kullanılabilir.

KAYNAKLAR

- Agrawal, R. and Srikant, R.**, 1994. Fast algorithms for mining association rules, *Proceedings of 20th International Conference on Very Large Data Bases*, Santiago, Chile, September 12-15.
- Alataş, B.**, 2007. Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, *Doktora Tezi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- Alatas, B., Akin, E. and Karci, A.**, 2008. MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules, *Applied Soft Computing*, **8**, 646-656.
- Beiranvand, V., Mobasher-Kashani, M. and Abu Bakar, A.** 2014. Multi-objective PSO algorithm for mining numerical association rules without a priori discretization, *Expert Systems with Applications*, **41**, 4259–4273.
- Birbil, I. and Fang, S.**, 2003. An Electromagnetism-like Mechanism for Global Optimization, *Journal of Global Optimization*, **25**, 263-282.
- Cura, T.**, 2008. Modern Sezgisel Teknikler ve Uygulamaları, Papatya Yayıncılık Eğitim, İstanbul.
- Dorigo, M. and Stutzle, T.**, 2004. Ant Colony Optimization, The MIT Press, London.
- Formato, R. A.**, (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics, *Progress In Electronagnetics Research*, **77**, 425-491.
- Han J. and Kamber M.**, 2001. Data mining: Concepts and techniques, Morgan Kaufmann Publishers, San Francisco.
- Hsiao, Y. T., Chuang, C., L., Jiang, J. A., Chien, C. C.**, 2005. A Novel Optimization Algorithm:Space Gravitational Optimization, 2005 IEEE International Conference on Systems, Man and Cybernetics, **3**, 2323 - 2328.
- Hung-Pin Chiu, Yi-Tsung Tang, and Kun-Lin Hsieh.**, 2006. A Cluster-Based Method for Mining Generalized Fuzzy Association Rules, *First International Conference on Innovative Computing, Information and Control*, Beijing, Aug. 30-Sept. 1, 519-522 (in China)
- Karaboğa, D.**, 2004. Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayın Dağıtım.
- Kamkar, I. Akbarzadeh-T, M.-R. Yaghoobi, M.**, 2010. Intelligent Water Drops a new optimizationalgorithm for solving the Vehicle Routing Problem, 2010 IEEE

International Conference on SystemsMan and Cybernetics (SMC), Istanbul, 10-13 Oct, 4142-414.

- Kaveh, A., Talatahari, S.,** 2010a. A novel heuristic optimization method: charged system Search, *Acta Mechanica*, **213**, 267-289
- Khademolghorani, F., Baraani, A. and Zamanifar, K.,** 2011. Efficient mining of association rules based on gravitational search algorithm, *International Journal of Computer Science Issues*, **8**, 2.
- Kripta, M. M.L.Kripta. R.,** 2008. Big Crunch Optimization Method, International Conference on Engineering Optimization, 01 - 05 June, Rio de Janeiro
- Kuo, R.J., Chao, C.M. and Chiu, Y.T.,** 2011. Application of particle swarm optimization to association rule mining, *Applied Soft Computing*, **11**, 326-336.
- Mata, J., Alvarez, J.L. and Riquelme, J.C.,** 2001. Mining Numeric Association Rules with Genetic Algorithms, Artificial Neural Nets and Genetic Algorithms, 264-267.
- Minaei-Bidgoli, B., Barmaki, R. and Nasiri, M.,** 2013. Mining numerical association rules via multi-objective genetic algorithms, *Information Science*, **22**, 15-24.
- Osman K. Erol, Eksin, I.,** 2006. A new optimization method: Big Bang–Big Crunch, *Advances in Engineering Software*, **37**, 106–111.
- Özkan, Y.,** 2008. Veri Madenciliği Yöntemleri, Papatya Yayıncılık Eğitim.
- Pachón Álvarez, V. and Mata Vázquez. J.,** An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an a priori discretization, *Expert Systems with Applications*, **39**, 585-593.
- Rabanal, P., Rodriquez, I. and Robio, F.,** 2007. Using River Formation Dynamics to Design Heuristic Algorithms, 6th International Conference, Kingston, Canada, August 13-17.
- Rashedi, E., Nezamabadi, H., Saryazdi, S.,** (2009). Gravitational Search Algorithm, *Information Sciences*, **179**, 2232-2248.
- Reza Qodmanan, H., Nasiri, M. and Minaei-Bidgoli, B.,** 2011. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence, *Expert Systems with Applications*, **38**, 288-298.
- S, K.N.V.D. and Ravi, V.,** 2013, Association rule mining using binary particle swarm optimization, *Engineering Applications of Artificial Intelligence*, **26**, 1832-184.

Shah-Hosseini, H., (2009a). Optimization with the Nature-Inspired Intelligent Water Drops Algorithm, In W. P. Dos Santos (Eds.), *Evolutionary computation* (pp. 297-320). Austria: I-Tech, Vienna.

Shah-Hosseini, H., (2011b). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, *Int. J. Computational Science and Engineering*, **6**, 132-140.

Silahtaroglu, G., 2008. *Kavram ve Algoritmalarıyla Temel Veri Madenciliği*, Papatya Yayıncılık Eğitim, İstanbul.

Tayarani M. H., Akbarzadeh, M.R., (2008). Magnetic Optimization Algorithms a new synthesis, in *IEEE Congress on Evolutionary Computation*, 2659 – 2664.

Xie, L., Zeng J., Cui, Z., (2009). General framework of Artificial Physics Optimization Algorithm, *World Congress on Nature & Biologically Inspired Computing*, Coimbatore, Page(s): 1321 – 1326.

Yan, X., Zhang, C. and Zhang, S., 2009. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support Original Research Article, *Expert Systems with Applications*, **36**, 3066-3076.

Wagner F.Sacco, Cassiano R.E.de Oliveria, (2005). A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic, *6th World Congresses of Structural and Multidisciplinary Optimization*. Rio de Janeiro, Brazil

URL-1, <http://funapp.cs.bilkent.edu.tr/DataSets/> Basketball veritabanı. 15 Haziran 2013.

URL-2, <http://funapp.cs.bilkent.edu.tr/DataSets/> Stock Prices veritabanı. 3 Ağustos 2013.

URL-3, <http://funapp.cs.bilkent.edu.tr/DataSets/> Quake veritabanı, 20 Ağustos 2013.

URL-4, <http://sci2s.ugr.es/keel/dataset.php?cod=66> New Thyroid veritabanı, 5 Eylül 2013

ÖZGEÇMİŞ

Ümit Can, 31.03.1982'de Tunceli'de doğdu. İlk, orta eğitimini Tunceli'de ve lise eğitimini de Malatya'da tamamladı. 2001 yılında Malatya Fen Lisesi'nden mezun oldu. Bahçeşehir Üniversitesi Bilgisayar Mühendisliği bölümünü bitirdi. 2010 yılında, Tunceli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak işe başladı. Halen aynı görevi sürdürmektedir.