T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİANABİLİM DALI

BELGE BENZERLİĞİ SONUÇLARININ NSGA-II İLE ÇOK AMAÇLI OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Hüseyin AHMETOĞLU
DANIŞMAN: Doç. Dr. RıdvanSARAÇOĞLU

VAN-2018

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİANABİLİM DALI

BELGE BENZERLİĞİ SONUÇLARININ NSGA-II İLE ÇOK AMAÇLI OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Hüseyin AHMETOĞLU

VAN-2018

KABUL VE ONAY SAYFASI

Elektrik Elektronik Mühendisliği Anabilim Dalı'nda Doç. Dr. Rıdvan Saraçoğlu danışmanlığında, Hüseyin AHMETOĞLU tarafından sunulan "Belge Benzerliği Sonuçlarının NSGA-II İle Çok Amaçlı Optimizasyonu" isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince…./...../..... tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan:..... İmza:

Üye:..... İmza:

Üye:..... İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun…./..../….. tarih ve .....sayılı kararı ile onaylanmıştır.

İmza

……..……..

Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığınıbildiririm.

Hüseyin AHMETOĞLU

ÖZET

BELGE BENZERLİĞİ SONUÇLARININ NSGA-II İLE ÇOK AMAÇLI OPTİMİZASYONU

AHMETOĞLU, Hüseyin
Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
Tez Danışmanı:Doç. Dr. Rıdvan SARAÇOĞLU
Mart2018, 77 sayfa

Sınıflandırma algoritmalarının başarı performanslarının artırımı, veri madenciliğinin önemli amaçları arasındadır.Bu tez çalışmasında,veri madenciliği sınıflandırma başarısının sezgisel yöntemlerle arttırılması incelenmiştir. Sınıflandırmada kullanılan eğitim veri setihem benzerlik hesap sonuçları yönünden hem de sınıflandırma yeteneği yönünden optimize edilmiştir. Aynı sınıfta olan vektörlerin benzerlik sonuçlarınınmaksimizeedilmesi, aynı zamanda farklı sınıftaki vektörlerin benzerlik sonuçlarınınminimize edilmesi amaçlanmıştır.Bu çelişen iki durum için çok amaçlı sezgisel yöntemlerden olan, Sıralı Seçkin Bastırılamayan Genetik Algoritma (NSGA II) kullanılmıştır. Hatalı sınıflandırma oranlarının, optimizasyonun her iterasyonunda sıfıra daha çok yaklaştırılması hedeflenmiştir.

Bu çalışmada veri madenciliğinin tüm aşamalarının sırayla gerçekleştirilmesine özen gösterilmiştir. Ham veriler işlenerek öznitelikler çıkarılmıştır. Boyut azaltma işlemleri için ise Temel Bileşen Analizi (PCA) kullanılmıştır. Veri setleri üzerinde KEn Yakın Komşu Algoritması (KNN) kullanılarak yalın haldeki sınıflandırma başarıları ile optimizasyon sonrası sınıflandırma başarıları karşılaştırılmıştır. Optimizasyonun, eğitim veri setinin sınıflandırma yeteneğini arttırdığı görülmüştür. Optimize edilmiş veriler, eğitim kümesi olarak kullanıldığında sınıflandırma başarısında artış gözlemlenmiştir.

Anahtar kelimeler: Çok Amaçlı Optimizasyon, Genetik Algoritma, KNN, NSGA2, Sınıflandırma, Temel Bileşen Analizi, Veri Madenciliği

ABSTRACT

MULTI-OBJECTIVE OPTIMIZATION OF DOCUMENT SIMILARITY RESULTS WITH(VIA) NSGA-II

AHMETOGLU, Huseyin
M. Sc. Thesis, Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Rıdvan SARACOGLU
March2018, 77Pages

Increasing performance of classification algorithms is one of the important goals of data mining.In this thesis study, it has been investigated how to increase the data mining classification success with heuristic methods. The training data set used in the classification is optimized in terms of the both similarity calculation results and the classification ability. It is aimed to maximize the similarity results of the vectors in the same class and at the same time to minimize the similarity results of the vectors in different classes. For these two contradictory cases, Non Sorting Genetic Algorithm II (NSGA II), which is a multipurpose heuristic method, is used. It is aimed to approximate the false classification ratios zero in each iteration of the optimization.

In this study, care was taken in order to perform all phases of data mining in order. The raw data were processed and attributes were extracted. For size reduction operations, Principal Component Analysis (PCA) is used. Using the K Nearest Neighborhood (KNN) Algorithm on the data sets, the lean classification successes and the post-optimization classification successes are compared. Optimization has been shown to increase the ability to classify training data sets. An increase in classification success was observed when the optimized data were used as a training set.

Keywords: Classification, Data Mining, Genetic Algorithm, KNN, Multi Objective Optimization, NSGA2, Principal Component Analysis

ÖN SÖZ

Belge Benzerliği Sonuçlarının NSGA-II İle Çok Amaçlı Optimizasyonu isimli bu çalışma Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisans tezi olarak hazırlanmıştır.

Tez çalışmam süresince danışmanlığımı üstlenerek, değerli fikirleri ile bana rehberlik eden, tez konusunun önerilmesinden tamamlanmasına kadar geçen sürede benden desteğiniesirgemeyen Danışman Hocam Sayın Doç. Dr. Rıdvan SARAÇOĞLU'na saygı ve teşekkürlerimi sunarım.

Sevgisi ve desteği ile hep yanımda olan, varlığı ile güven veren sevgili eşim Betül Hanım'a vemasumluğuyla ilham kaynağım canım oğlum Ali Kerem'e bugünlere gelmemde büyük emek sahibi olan ve hep yanımda olan anneme, babama ve biricik kardeşime en içten şükranlarımı sunarım.

2018

Hüseyin AHMETOĞLU

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELER LİSTESİ

Çizelge **Sayfa**

ŞEKİLLER LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
$\boldsymbol{d}$	Öklid Mesafesi
$\boldsymbol{s}$	Kosinüs Benzerliği
$\boldsymbol{X}$	Parametre ve Değişken Uzayı
$\boldsymbol{x}$	Parametre veya Değişkenler
$\boldsymbol{E}$	Eğitim Kümesi
$\boldsymbol{k}$	Kfold veya KNN değeri
$\boldsymbol{m_j}$	Gen'in Bit Sayısı
$\boldsymbol{a_j}$	Parametrelerin Alt Sınır Değeri
$\boldsymbol{b_j}$	Parametrelerin Üst Sınır Değeri
$\boldsymbol{p_i}$	Bireyin Seçilme Olasılığı
$\boldsymbol{q_i}$	Bireyin Kümülatif Olasılığı
$\boldsymbol{p_c}$	Çaprazlama Oranı
$\boldsymbol{m_c}$	Mutasyon Oranı
$\boldsymbol{e}$	Kısıt Fonksiyonu
$\boldsymbol{P^*}$	Pareto Optimal Küme
$\boldsymbol{P_i}$	Başlangıç Popülasyonu
$\boldsymbol{Q_i}$	Yeni Popülasyon
$\boldsymbol{R_i}$	Birleşik Popülasyon
$\boldsymbol{F_i}$	Pareto Yüzeyler
$\boldsymbol{S_p}$	*P* bireyinin domine ettiği elemanların kümesi
$\boldsymbol{n_p}$	*P* bireyini domine eden elemanların sayacı

Kısaltmalar	Açıklama
NSGA2	Sıralı Seçkin Bastırılamayan Genetik Algoritma
GA	Genetik Algoritmalar
KNN	K En Yakın Komşu
PCA	Temel Bileşen Analizi
KFold	K Katlamalı Çapraz Doğrulama
VK	Veri Kümesi
RS	Rulet Çarkı Seçimi
TS	Turnuva Seçimi
NFE	Fonksiyon Değerlendirme Sayısı
CD	Yoğunluk Mesafesi
SCH	SCHaffer'in testfonksiyonu
ZDT	Zitzler-Deb-Thiele
TP	Topluluğun Toplam Puanı
SVM	Destek Vektör Makinesi
SF	SınıflandırmaFonksiyonu

1.GİRİŞ

Sürekli gelişen teknoloji ile birlikte her geçen gün kullanılmakta olan veri miktarı da hızla büyümektedir.Veri bilimi verinin analizine dayanır ve veri miktarının hızla artması da bu verilerin analiz edilmesini zorlaştırmaktadır. Verilerden yararlı ve anlamlı bilgiler çıkarılmasıiçin kaynağını istatistik, yapay zekâ ve makine öğrenmesinden alan bir disiplin olan veri madenciliği ortaya çıkmıştır. Veri madenciliği endüstri, ekonomi ve iş çevrelerinden gelen yoğun ilginin neticesinde akademik alanda da ilgi odağı haline gelmiştir. Veri Madenciliği artıkfinans, pazar araştırmaları, bankacılık, sağlık, temel bilimler gibi pek çok alanda kullanılmaktadır.

Yaşadığımız çağ bilgi ve iletişim teknolojisi çağıdır. Bu durum her geçen gün elektronik ortamda saklı bulunan bilgilerin katlanarak artmasına ve bilgi yığınlarının oluşmasına neden olmuştur. Bununla birlikte insanların, bu bilgi yığınları arasında ihtiyaç duydukları bilgilere en kısa zamanda erişmek istemeleri, birbirleriyle ilişkili bilgileri bir kategoriye göre sınıflandırma ihtiyacını da beraberinde getirmiştir. Ancak bu işlemin elle indekslenmesi çok zaman alacağından bunun sistematik hale getirilmesi zorunlu olmuştur.Veri yığınları içinden doğru verilerin bulunması, verilerin birbirleri arasındaki ilişkilerin sorgulaması işlemleri için veri madenciliği alanındaki teknikler tek başlarınaher zaman yeterli değildir. Bu nedenle veri madenciliğinin daha başarılısonuçlar vermesi için farklı yöntemler geliştirilmiştir.

1.1. Tezin Amacı

Verilerin sınıflandırılması işlemi makine öğrenmesinin önemli konuları arasındadır. Sınıflandırma sonuçları ne ölçüde başarılıysa, makine karar verme kabiliyetine aynı ölçüde sahip olur denilebilir. Bu çalışmanın temel amacı sınıflandırma başarısının sezgisel yöntemlerle artırılmasıdır. Sezgisel optimizasyon yöntemleri matematiksel olarak çözümü mümkün olmayan problemlerin çözümlerine, makinenin sezgisel yaklaşımlarıyla en kısa zamanda ve en yakın tahminlerle ulaşabilmesini sağlar. Birbirlerine benzeyen nesnelerin benzerliklerini artırmak ve farklı nesnelerin

benzerliklerini azaltmak çelişen iki durumdur. Bu noktada Sıralı Seçkin Bastırılamayan Genetik Algoritma (NSGA II) devreye girer. Optimizasyonun her döngüsünde sınıflandırma başarısını artıracak katsayılar üretmek, eğitim kümesinin sınıflandırma yeteneğini artıracaktır. Bu tez çalışmasının kapsamını benzerlik ve sınıflandırma olmak üzere iki optimizasyon oluşturmaktadır.

Uygulamanın üzerinde deneneceği ham veriler metin madenciliğinde kullanılan Classic 3 veri setidir. Bu veri seti 3 farklı sınıfa ait toplamda 4020 makaleden oluşmaktadır. Veri madenciliğinin her adımının bu veri seti üzerinde uygulanması amaçlanmaktadır. Bu durum göz önüne alındığında metin madenciliği ile ilgili bilgiler de tez kapsamında işlenmiştir.

Belgelerin sınıflandırılması işlemi birçok kullanım alanı bulmaktadır. Artık hayatımızda önemli bir yer tutan internet arama motorları, bu kullanıma güzel bir örnektir. Verimli, hızlı ve en önemlisi; kullanışlı ve faydalı sonuçlar getiren bir arama aracı, günlük hayatımızda önemli bir eksikliği dolduracaktır.

Belge sınıflandırması, soru cevaplandırma sistemlerinin ve bu alandaki çalışmalarında önemle üzerinde durduğu bir konudur. Bu konuda varılmak istenen son nokta, soru cevaplandırma sisteminin insana yakın bir şekilde cevap üretme yeteneğine sahip olmasıdır. Bu arama işlemi bir kütüphanede araştırma veya arama yaparken de kullanılabilir. Mevcut veri tabanı sorgularının ötesine geçerek kütüphanedeki tüm kitapların içindekiler bölümleri veya özetleri üzerinden daha detaylı ve zeki bir arama yapabilmek oldukça önemli bir konudur.

Sınıflandırma başarısının artırımı veri biliminde önemli bir konudur ve bu alanda yapılan çalışmalar veri madenciliği konusunda çok önemli bir yer teşkil eder. Bu çalışmanın kapsamı göz önüne alındığında, bu tezin sınıflandırma konusunda yapılacak sonraki çalışmalar için bir örnek teşkil edeceği düşünülmektedir.

2. LİTERATÜR BİLDİRİŞLERİ

Veri-Metin madenciliği, bilgi çıkarımı ve çok amaçlı optimizasyonla ile ilgili bugüne kadar yapılmış pek çok çalışma mevcuttur. Aşağıda bu çalışmaların bazılarına kısaca yer verilmiştir.

2.1.Metin Madenciliği

Liao ve ark. (2012), yaptıkları çalışmada; 2000-2011 yılları arasındaki veri madenciliğinin teknik ve uygulamalarını incelemişler, yayınlanmış makaleler ve yapılan çalışmaları kaynakları ile sunmuşlardır.

Pons-Porrata ve ark. (2007), yaptıkları çalışmada metin madenciliği ile metin konularının özetlenmesi ve benzer belgelerin belirlenmesini incelemişlerdir.Yüksek *idf* değerli sözcüklerle özetleme veKosinüs benzerliğiyle benzer yazıları belirleme işlemini gerçekleştirmişlerdir.

Yang and Liu (1999) ve Mengle ve ark. (2007) yaptıkları çalışmada; k-NN (k-Nearest Neighbor), Naive Bayes ve SVM (Support Vector Machines) yöntemleri kullanılarak metin sınıflandırma performanslarını karşılaştırmışlardır. Çalışmanın sonucuna göre SVM ve k-NN'in Naive Bayes'e göre daha başarılı sınıflandırma yaptığı görülmüştür.

Soucy ve Mineau (2001),k-NN ve Naive Bayes'in bit ağırlıklandırma kullanılarak gerçekleştirdikleri metin sınıflandırma işleminde, k-NN'in Kosinüs benzerliği ile birlikte uygulandığında, Naive Bayes'ten daha başarılı olduğunu görmüşlerdir.

Sanwaliya ve ark. (2010), farklı k değerlerine göre başarımı ölçümü gerçekleştirmişlerdir. KNN için k değerini 30,40, 50, 60, 70, 80 alarak sınıflandırma çalışması yapmışlar ve en yüksek sınıflandırmayı %90,64 başarıyla k=50 değerinden elde etmişlerdir.

Li ve ark. (2011), yaptıkları çalışmada; küçük ve büyük boyutlu veri setleri kullanarak, test dokümanlarının 6 farklı sınıftan birine atanmasını hedeflemişlerdir. tf

ağırlıklandırma, Naive Bayes ile birlikte kullanmışlardır. Büyük boyutlu veri setleri ile yapılan sınıflandırma işleminde, bütün sınıflarda daha başarılı sınıflandırma gerçekleştirmiştir.

Durmaz ve Bilge (2011), Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi konusunu işlemişlerdir. Çalışmada tf-idf olarak ağırlıklandırılmış sözcükler kullanılarak, iki farklı veri seti ile çalışılmıştır. Sınıflandırmada kullanılacak sözcük sayısı, orijinal metin sayısının %90,00'ından %10,00'una kadar kademeli olarak düşürülmüş ve sonuçlar gözlenmiştir. Toplam sözcüklerin %10,00'u ile yapılan sınıflandırma başarısının birinci veri setinde %6,25, ikinci veri setinde ise %11,39 arttığı gözlemlenmiştir.

Bozkurt ve ark. (2007), yazar tanıma alanında çalışmışlardır. Eğitimli ve eğitimsiz kümeleme ile parametrik ve parametrik olmayan metotlar gibi bilgi geri kazanım metotlarını incelemişlerdir. Gazete makalelerinden oluşan bir derlemoluşturmuşlardır.Bu derlemdeki dokümanların çeşitli özellik vektörleri çıkartılarak farklı sınıflandırıcılarla sonuçları incelemişlerdir.Kelime yığınlarında en yüksek başarıyı Destek Vektör Makinesi (SVM) ile, fonksiyonel kelime vektörünü de Gaussian sınıflandırıcısı ile işlediklerinde elde etmişlerdir. Bu çalışmada yazarların her bir dokümanından kelime zenginliği, fonksiyonel kelimeler, yazım stili (kullanılan noktalama işaretleri, cümle sayısı vb.) gibi vektörel özellikler çıkartmışlardır. Daha sonra bu özellikler Gaussian, Parzen Windows, Histogram Metodu, K En yakın Komşu(KNN), SVM, Yapay Sinir Ağı gibi sınıflandırıcılar ileişlemişlerdir. Ayrıca özellik uzayının boyutunu azaltmak için Temel BileşenlerAnalizi (PCA) kullanmışlardır. En düşük sınıflandırma başarısını Yapay Sinir Ağı yaklaşımınıkullandıklarında elde etmişlerdir. Kullandıkları veri setini üç bölüme ayırmışlardır.(Kelime yığını, fonksiyonel kelimeler, yazarın stil özellikleri). Yazarın stil özellikleriniHistogram metodu kullanarak işlediklerinde büyük boyutlu özellikuzayında elde ettikleri sonuçlar onları tatmin edici olmamıştır.Yazarın stilözellikleri vektörünü ve fonksiyonel kelimeler vektörünü KNN ile işlemişlerve bazı yazarlar için stil özelliğinin ayırt edici olduğunu fakat bazıları için de ayırtedici olmadığını gözlemlemişlerdir. Bundan yola çıkarak parametrik olmayanmetotların (Parzen Windows, Histogram

Metodu, KNN) yazar tanımada çok başarılısonuçlar vermediğini tespit etmişlerdir. Gaussian sınıflandırıcısını yazarın stilözellikleri vektörü ile kullanıldığında %60'lık, kelime yığını vektörü SVM ilekullanıldığında %95'lik bir başarı elde etmişlerdir.

Zhao ve Zobel (2005), çalışmalarında internet üzerindeki gazete ve dergilerde, yazar atıf metotlarını(anlama bağlı, sözdizimsel ve dil modeli) karşılaştırmayı amaçlamışlardır. Çalışmalarında özellik vektörü olarak fonksiyonel kelimelerin("the", "once", edat, bağlaç) sıklık sayılarını baz almışlardır. Sınıflandırıcı olarak Bayes Ağı, NB, KNN, Karar Ağacı algoritmalarını kullanmışlardır. Veri setini dokümanları yazarlarına göre gruplandırarak hazırlamışlardır. 10.918 adet yazarı bilinmeyen doküman ise tek bir grupta toplanmıştır. Özellik vektörü olarak 365 fonksiyonel kelime kullanmışlardır.

Weng ve Lin (2003), çalışmalarında benzer belge aranması için sınıflandırma konusu üzerine odaklanmanın yanı sıra kavram ve kavramın dağılımı faktörleri üzerinde de durmuşlardır. Benzerlik aranması için çoklu kavram mekanizması önermişlerdir. Benzerlik kalitesini artırmak için kavramın belge içindeki dağılımı faktörünü incelemişlerdir. Kümeleme işlemi için ise kNN algoritmasını kullanmışlardır. Deneysel sonuçlar, önerdikleri tekniğin geleneksel yaklaşımlardan daha verimli olduğunu ortaya koymuştur.

Dhillon ve ark. (2001), büyük miktardaki belgelerinkümeleme süreleri odaklı bir çalışma geliştirmişlerdir. Bunun için verimli bir hafıza yönetimi ve çoklu yollu bir ön işleme planı üzerinde durmuşlardır. Ayrıca veri kümesinde yer alan boşluk problemini çözmek için hızlı bir küresel k-ortalamalar algoritması önermişlerdir. Özet metinleri üzerinde yapılan deneysel sonuçların verimli olduğunu görmüşlerdir. Ayrıca belge sayısındaki artışla işlem süresinin doğrusal olarak arttığıgösterilmiştir.

Amasyalı ve Yıldırım (2004) ise çalışmalarında gazetelerin web sayfalarındaki Türkçe haber metinleri üzerinde otomatik sınıflandırmayı amaçlamışlardır. Belge tabanlı verilerdeki boyut azaltma işlemi için Bilgi Kazanım (Information Gain - IG) Ölçümleri ve Temel Bileşenler Analizi (Principal Components Analysis - PCA) kullanmışlardır.

Kou ve Gardarin (2002), çalışmalarında belge sınıflandırması için terim-kategori ve kategori-belge özelliklerini incelemişlerdir. Sınıflandırmada k yakın komsu (kNN) yöntemini kullanmışlardır. Sınıflandırmanın kalitesini artırmak için çalışmalarına terim ilişkisi faktörünü eklemişlerdir. Terim ilişkisi hesaplamasında e benzerlik modeliniönermişlerdir. Deneysel sonuçlar e benzerlik modelinin performansı artırdığını göstermiştir.

2.2.Çok Amaçlı Optimizasyon ve Sıralı Seçkin Bastırılamayan Genetik Algoritma(NSGA II)

Optimizasyon problemleri, evrimsel algoritmalar ve çok amaçlı optimizasyon problemlerinin çözümü ile ilgili bugüne kadar yapılmış pek çok çalışma bulunmaktadır. Aşağıda bu çalışmaların bazılarına kısaca yer verilmiştir.

Osyczka(2002), evrimsel algoritmaların esas bakış açısını içeren bir çalışma gerçekleştirmiştir. Özellikle klasik optimizasyon yöntemleri kullanılarak çözülemeyen tasarım optimizasyon problemleri ele alınmış ve bu problemleri çözen ileri uygulamalar sunmuştur. Araştırmacılar ve başta makine, inşaat, endüstri ve bilgisayar mühendislikleri olmak üzere tüm mühendislik bölümlerinin öğrencileri için başvuru kaynağı niteliğinde bir kitap hazırlamıştır. Konuya basit tasarım optimizasyon problemlerinin tanıtımıyla başlamıştır.Daha gelişmiş GA'lar ve evrimsel algoritmalar, evrimsel işlemciler; çeşitli seçim, çaprazlama ve mutasyon türleri; sınırlanmamış-sınırlandırılmış optimizasyon, ceza fonksiyonları ve tarihsel sıralamayla çok-kriterli genetik algoritmalar hakkında çıkarılmış yayınların derlemeleri ve bunlarla ilgili örneklerden, çok-kriterli optimizasyon algoritmalarının diğer yapay zeka teknikleri ile hibritleştirilmesi konularına kadar ele almıştır.

Goldberg (1989), genetik algoritmalar (GA) – doğal seçim mekanizmasına ve doğal genetiğe dayalı arama işlemleri hakkında geniş bilgiler vermiştir. GA'ların matematiksel temelleri, bilgisayarda gerçekleştirim aşamaları, çeşitli örneklerle ele alınmıştır.Genetik aramada ileri işlemci ve teknikler, genetik temelli makine öğrenmesi

ve uygulamaları detaylı bir şekilde incelemiştir. Ayrıca bazı dillerde yazılmış kodlarla örnekler vermiştir.

Murata ve Ishibuchi (1995), Çok-kriterli optimizasyon problemlerinin Pareto-optimal çözümlerini araştırıp ortaya koyacak bir genetik algoritma çatısı önermişlerdir. Önerilen yaklaşımın tek-kriterli genetik algoritmalardan ayıran iki tarafı vardır, bunlar seçim işlemi ve seçkini koruma stratejisidir. Önerilen algoritmanın özgün yanı seçim işlemidir.Çok amaçlı fonksiyonların ağırlıklı toplamına dayanan bir çaprazlama işlemi ile seçim işlemi gerçekleşir. Seçim işleminin karakteristik özelliği, çok amaçlı fonksiyonlara iliştirilen ağırlıkların sabit değil her bir seçim için rasgele bir şekilde belirlenmesidir. Bu yönteme kısaca Seçkini koruma stratejisi denilebilir.Tek bir seçkin çözüm yerine, bir dizi seçkin çözüm kullanılır. Belli sayıda birey Pareto-optimal çözümlerin geçici kümesinden seçilir ve sonraki jenerasyona seçkin bireyler olarak aktarılır.

Deb (1999), Pareto-optimal yüzeye,doğru yakınsama için çeşitli problemleri ele almışlardır. Tek kriterli optimizasyon problemlerinden oluşturulan çok kriterli test problemleri geliştirilmiş ve bu problemler algoritmaların performans ölçümlerinde kullanılmıştır.

Murata (1997), önce tek-amaçlı optimizasyon problemlerini GA'lara uygulamıştır. GA'ların performansının geliştirilmesi amacıyla diğer algoritmalarla GA'ların hibritleştirilmesini denemiştir. Daha sonra, çok-amaçlı genetik algoritmalar bazı sezgisel yöntemlerle hibritleştirilmiştir.Çok amaçlı hibrit GA'lar zamanlama problemleri ve dilbilimsel karar seçim problemlerine uygulamıştır.

Elitist Non-dominated SortingGenetic Algorithm (NSGA-II) kullanılarak yapılan çalışmalarda ise Deb ve ark. (2006), Genel bir pareto cephesi yerine daha gerçek sonuçlar elde edilmek için yeni bir sağlamlık prosedürü geliştirmişlerdir. Küresel ve sağlam pareto cephelerin karşılaştırılması için NSGA II kullanmışlardır. Bulunan sonuçları simülasyon ile gösterebilmek için birçok kısıtlı ve kısıtsız test problemi geliştirmişlerdir.

Goel ve ark. (2007),NSGA II ile elde edilen Pareto Cephesi üzerinde yakınlaşmalar gerçekleştirmişlerdir. Yakınlaştırılmış Pareto optimal front (POF), uzlaşılmış tasarımların seçilmesi için amaçlar arasındaki dengelerin görselleştirilmesine ve değerlerinin belirlenmesine yardımcı olabildiğini göstermişlerdir.

Shokri ve ark. (2013), çok amaçlı optimizasyon problemlerinin evolutionary algorithms (EAs) ile çözümü sırasında harcanan zamanı azaltmak için NSGA-II ve Artificial Neural Networks (ANN) birleştirerek yeni bir metot önermişlerdir. Önerilen metotu üç standart ve bir gerçek hayat problemine uygulamışlardır. Pareto optimal sınırın bulunması için gereken zamanı, ANN olmadan kullanılan NSGA-II çözümlerine göre hatırı sayılır oranda azaltabilmişlerdir.

3.MATERYAL VE YÖNTEM

3.1. Metin Madenciliği

Metin madenciliği, veri madenciliğinin veya veritabanında bilgi keşfinin metin odaklı uzantısı olan farklı bir uygulaması olarak görülebilir. Metinlerin vektörel yani işlenebilir hale getirilmeleri için kullanılan bir uygulamadır. Metnin hazırlanması metin madenciliğinin, işlenmesi ise veri madenciliğinin konusudur.İnternet ortamındaki belge yığınlarından faydalı bilgiler elde etmek amacıyla bilgiye ulaşım ve bilgi çıkarımı alanlarında sürekli yenilikler yaşanmaktadır. Bu yenilikleri metin madenciliği adı altında incelemek mümkündür (Güven, 2007).

3.2. Metin Madenciliğinin Adımları

Bu bölümde metinlerden özellik çıkarımı konusuna değinilmiş ve metin madenciliğinin adımları başlıklar halinde incelenmiştir.

3.2.1.Metin koleksiyonu oluşturma

Veri seti, veriler arasından gerçekleştirilecek çalışmada kullanılacak olanların seçilmesiyle oluşturulur. Bilgisayarda kayıtlı dosyalar, E-postakutusundaki mailler, bir şirketin her ay sonunda hazırladığı raporlar, forum sitelerinde paylaşılan yazılar, hastaların tahlil sonuçları vs. veri setlerini oluşturan kaynaklar arasında gösterilebilir.

3.2.2. Metin ön işleme

Veriler uygulamaya geçilmeden önce temizlenmeli, bütünleştirilmeli, dönüştürülmeli ve indirgenmelidir. Bu adımlar veri madenciliğinin adımlarıdır doğru sonuca ulaşabilmek için dikkatle gerçekleştirilmelidir.Bu adımların gerçekleştirilme yöntemi eldeki verilere göre farklılıklar gösterebilir: Soru-cevap sistemlerinde soru

kelimeleri çok kullanıldığından bu kelimelerin dahil edilmemesi; web sayfaları üzerinde işlem yapılırken HTML etiketlerinin temizlenmesi; kelimelerin küçük harfe çevrilmesi, kelime köklerinin elde edilmesi; noktalama işaretlerinin kaldırılması vs. Gereksiz kelimelerden metni arındırmak ve kelime köklerini bulmak metin madenciliği çalışmalarında yapılan ön işlem aşamalarıdır.

3.2.3. Sözcük ağırlıklandırma

Kelime kökleri sözcük olarak ifade edilebilir. Sözcükler elde edildikten sonra sözcük ağırlıklandırma işlemine geçilir. Ağırlıklandırma işlemine sözcüklerin doküman üzerindeki etkisi de denilebilir. Sözcükler, dokümanlarda bulunma durumlarına göre değer alıp, sayısal olarak ifade edilerek vektörel yapısallık elde etmiş olurlar.

Bir sözcüğe değer verilirken, eğer sözcük doküman içerisinde bulunuyorsa sözcüğün ağırlıklandırılmış değeri, bulunmuyorsa 0 değeri verilir.

Çizelge 3.1'de, sözcüklerin ID değerlerinin nasıl verildiğine ilişkin, rasgele seçilen kelimeler üzerinde bir örnek verilmiştir.

Çizelge 3.1. Sözlük tablosu

Kelime No	***Kelime***
1	Van
2	Yüz
3	Yıl
4	Üniversite
5	Elektrik
6	Elektronik
7	Mühendis

3.2.3.1. Bit ağırlıklandırması

Bit ağırlıklandırma sözcüklerin dokümanda bulunup bulunmadığıyla ilgilenir. Bit, boolean ve binary ağırlıklandırma olarak da isimlendirilir.Sözcüğün alacağı değer 0 veya 1'dir. Eğer sözcük dokümanda yer alıyorsa 1, yer almıyorsa 0 değerini alır.

Dolayısıyla dokümanda bulunan bütün sözcükler eşit değerdedir (Jackson ve Moulinier, 2002). Bu durumda sözcüğün doküman içerisinde bir kez geçmesiyle birden çok geçmesi arasında bir fark yoktur denilebilir.

3.2.3.2 Sözcük frekansı ağırlıklandırması (*tf*)

Sözcük frekans ağırlıklandırılması bit ağırlıklandırma gibi sadece uygulandığı dokümanla ilgilenir. Sözcüğün diğer dokümanlarda geçmesi bu ağırlıklandırmanın konusu değildir. Sözcüğün doküman içerisinde kullanılma sayısıyla ağırlıklandırılır. Bu durumda bir sözcüğün doküman içerisinde birden fazla geçmesi, o doküman için değerli olduğu anlamına gelir.

3.2.3.3. Ters doküman ağırlıklandırması (*idf*)

Ters doküman ağırlıklandırmasında her sözcük, bütün eğitim dokümanlarındaincelenir ve geçtiği doküman sayısına göre ağırlıklandırılır. Sözcüğün doküman için ne kadar belirleyici olduğu ölçülür. Eğer sözcük, sadece bir dokümanda geçiyorsa yüksek değerli, birçok dokümanda geçiyorsa düşük değerli demektir. Bir sözcüğün bütün dokümanlarda geçmesiyle hiçbir dokümanda geçmemesi aynı değerdedir denilebilir ve ağırlık 0 olarak alınır.

3.3. Temel Bileşen Analizi (PCA)

Temel Bileşen Analizi ile bir veri seti öyle bir dönüşüme girer ki, elde edilen veri setinin yeni halinin bütün değerleri birbirinden olabildiğince ayrılır ve dağılır.Temel bileşen analiziverinin bağımlılık yapısının ortadan kaldırılması ve verinin boyutununminimuma indirgenmesi amaçları için kullanılır. Sınıflandırma, tanıma, boyut indirgenmesi ve boyut yorumlanması gibi işlevleri yerine getiren, çok değişkenli bir istatistik yöntemidir. Verinin içindeki en güçlü örüntü PCA ile bulunmaya çalışılır. PCA bu işlevi nedeniyleörüntü bulma tekniği olarak da kullanılmaktadır. Tüm boyut

takımından seçilen küçük bir boyut setiyle, verinin sahip olduğu bütün çeşitlilik, elde edilebilir. Veri setindeki gürültüler, güç yönünden örüntülerden daha zayıftır. Bu nedenle boyut küçültme sonucunda bu gürültüler PCA ile temizlenebilir.Verilerin boyutunu azaltma,tahminleme yapma ve veri setini bazı analizler için görüntülemek PCA'nın temel amaçları arasında sayılabilir.

PCA sonunda p boyutlu bir uzayın gerçek boyutu belirlenir. Oluşan bu gerçek boyuta temel bileşenler adı verilir. Temel bileşenlerin kolerasyonları yoktur. Birinci temel bileşen, diğer bileşenlere göre toplam değişkenliği en çok açıklayan bileşendir. Bir sonraki temel bileşen ise kalan değişkenliği en çok açıklayan bileşendir. Çok boyutlu verilere PCA ile doğru açıdan bakılarak çoğunlukla verideki ilişkiler doğru açıklanabilir. Temel bileşen analizinin amacı bu doğru açıyı saptayabilmektir.

PCA'nınen kritik adımı, problemi çözmek amacıyla görsel inceleme için en iyi açıyı yani uygun bir koordinat sistemini seçmektir. Verilere uygun bir açıdan bakmak, bulunan bir koordinat sistemini kullanarak verileri analiz etmek demektir.PCA ile uygun koordinat sistemi aranırken,ilk eksen olarak, verilerin değişiminin en fazla olduğu yön seçilir. İkinci eksen olarak, önceki eksene dikey olan ve yine verilerin değişiminin en fazla olduğu bir yön daha seçilir. Üçüncüeksen olarak ise önceki iki eksene dikey olan ve kalan verilerin değişiminin en fazla olduğu yön seçilir.Bu şekilde her zaman yeni eksen olarak verilerdeki en büyük kalan değişimde olan yön seçilmektedir. Seçilmiş dikey konumdaki en büyük değişimyönlerine temel bileşenler denir. PCA yönleri, verilerin değişiminde en büyük katkıya sahip olan yönü ilk önce belirtmekte, daha sonra da daha az katkısı olan yönleri açıklamaktadır.

Gerçek uygulamalar göz ününe alındığında çok boyutlu ve ilk bakışta çok karmaşık verilerin çok az temel etkiye sahip olduğu düşünülebilir. Uygulama gerçekleştiğinde bu etkiler sadece birkaç PCA yönü olarak bulunabilir. PCA'nın bulduğu yönler, çok boyutlu ve karmaşık verileri genellikleaz sayıda yeni nitelikle açıklayıp temsil edebilmesini sağlayabilmektedir.

"Tutulan varyans" kavramı temel bileşenlerin yeterli sayısını belirtmek için kullanılır. Kullanılacak ilk temel bileşenlerin toplam varyansı orijinal verilerin toplam varyansının %90'ı ile %95'i arasında olmalıdır. Genel uygulamalara bakıldığında, 1000

boyutlu veriler için genellikle 10 ile 20arasındaki ilk temel bileşen verilerin bu değerleri verdiği görülmektedir.Başka bir ifadeyle, orijinal verilerin %95 doğrululukla temsil edilebilmesi için 10 ile 20 arasında PCA bileşeni yeterli olabilir. Örneğin, 1000 özelliğe sahip bir veriyi kaydetmek için bütün 1000 özelliği ile kaydetme yerine 10 ile 20 arasında alınan ilk temel bileşenler kaydedilip diğer bileşenlerin de değerleri için ortalama olarak depolanabilir. Sonuç olarak, PCA, boyut azaltmada çok faydalı bir yöntemdir. Çok boyutlu verilerin daha az boyutlu yeni bir veri setiyle temsil edilmesinde çok etkilidir. PCA, orijinal veriler için dikolanenbüyükvaryansyönlerini bulup orijinal verileri bu koordinat sisteminde gösterebilir. Çok boyutlu verilerin görselleştirilmesi ve analiz edilmesi için kullanılabilir. Özellikle makine öğrenmesi ele alındığında, verilerin boyutu azaltılabilir. Değişimin az olduğu PCA özellikleri modelleme için önemsiz görülebilir, bu şekilde modelleme ile ilgili hesaplama hızlandırılabilir.Boyut azaltma ile PCA, veri sıkıştırılması işleminde kullanılabilir.

Özetlemek gerekirse PCA, istatistiksel bir yöntemdir. Bir veri setindeki örüntünün tanımlanmasında, veri setinin açıklanıp yorumlanmasında, veri içindeki benzer ve farklı desenlerin tanımlanmasında kullanılabilir. PCA verinin sıkıştırılmasına boyut azaltarak imkân sağlamaktadır. Bu rağmen boyut azaltılırken veri kaybı da yaşanmamaktadır. Bu teknik bilgisayar bilimleri içerisinde veri madenciliği ve görüntü işleme alanında sıkça kullanılmaktadır.

3.4. Benzerlik Hesaplamaları

Verilerin birbirlerine olan yakınlıkları, sınıflandırma ve doküman benzerliği bulma işlemlerinde verilerin birbirleriyle ne kadar ilişkili olduğunun tespit edilmesiyle ilgilidir. Verilerin benzerliklerini bulmak, vektörel ifadeleri kullanılarak gerçekleştirilir. Benzerlik, veriler arası mesafenin ölçülmesi ve değerlendirilmesidir (Nanopoulos ve ark., 2001), Öklid mesafe ölçümü ve Kosinüs benzerliği en çok kullanılan benzerlik tespit yöntemleridir.

3.4.1.Öklidmesafesi

$X=\{x_1,x_2,x_3,...,x_n\}$ ve $Y=\{y_1,y_2,y_3,...,y_n\}$ iki vektör olmak üzere bu iki vektör arasındaki mesafe olan *d*'nin hesaplanması Eşitlik 3.1'te formüle edilmiştir (Han and Kamber, 2006; Hand ve ark., 2001). *d* değerinin düşüklüğü vektörlerin birbirlerine benzerlik yönünden yakın olduğunu belirtir.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.1}$$

3.4.2.Kosinüsbenzerliği

$X=\{x_1,x_2,x_3,...,x_n\}$ ve $Y=\{y_1,y_2,y_3,...,y_n\}$ iki vektör olmak üzere bu iki vektörün benzerliği *s* ile gösterilmiştir. Eşitlik 3.2'de bu benzerlik formüle edilmiştir (Hand et al., 2001). *s*değerinin büyüklüğü vektörlerin birbirlerine yakın olup benzediğini belirtir. *s*değeri, iki doğru arasındaki açının kosinüsüdür. *s*değeri ile iki doğru arasındaki açı değeri ters orantılıdır.

$$s = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n}(X_i)^2}\sqrt{\sum_{i=1}^{n}(Y_i)^2}} \tag{3.2}$$

s benzerlik değeri, skaler çarpımların iki vektörün normlarının çarpımlarına bölünmesiyle bulunur (Salton ve Buckley, 1988). *s*, maksimum 1 değerini alır. *s* değerinin 1'e yakınlığı vektörlerin birbirine benzediğini gösterirken 0'a yakınlığı vektörler arasındaki ortaklığın az olduğunu ve vektörlerin benzerliğinin düşük değerde olduğunu gösterir.

3.5. Metin Sınıflandırma

Önceden belirlenmiş sınıflara doküman atamak metin sınıflandırmasının konusudur. Sınıflandırma yapılmadan önce sınıfların belirlenmesi gerekir.Dokümanların ağırlıklandırılmış değerli vektörel ifadeleri kullanılarak elde edilenbenzerlik ölçüm sonuçlarına ve uygulanan algoritmaya göre sınıflandırılmasıgerçekleştirilir. Metin sınıflandırma, doğal dil metinleriyle çalışan birsınıflandırmadır (Soucy and Mineau, 2001).

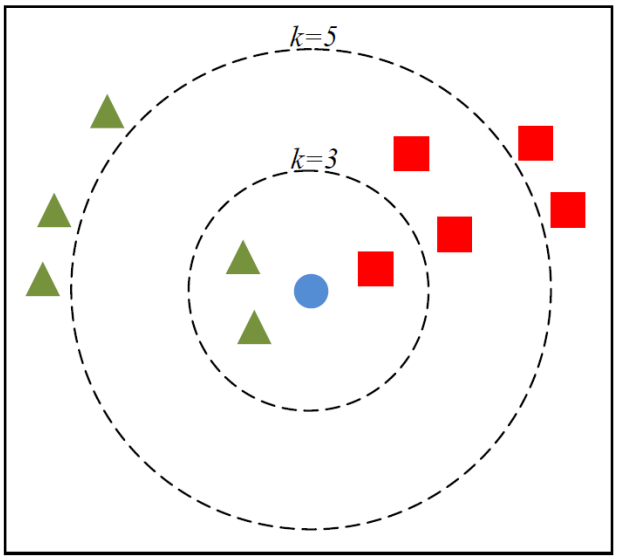
Dokümanların uzunluğu metin madenciliği çalışmalarındaki en büyüksorunlardan biridir. Dokümanların ön işlemlerden geçirilmesi ve özellik seçimi uygulaması ve dolayısıylaboyutun azaltılması bu sorunun giderilmesi için başvurulan adımlardandır.

Sınıflandırmanın bilgisayar tarafından hızlı bir şekilde yapılabilmesi için makine öğrenmesine ihtiyaç duyulur. Çünkü el ile kategorizasyon pahalı vezaman tüketen bir iştir.Ayrıca elle sınıflandırmada, sınıflandırmayı yapanuzmanların vermiş oldukları kararlara bağlı olarak sonuçlar da değişmektedir (İlhan,2001). Bu sebeple otomatik metotlar, algoritmalar ve büyük miktarlardaki verilerleçalışan araçlar önemli bir hale gelmiştir.

3.5.1. K en yakın komşu algoritması (KNN)

KNN algoritması ile sınıflandırma, önceden belirlenmiş k değerine göre uzaklıkları hesaplanmış eğitim verileri içerisinden en yakın k verisine en yüksek frekansa sahip sınıfa göre test verisinin sınıfını belirleme işlemidir (Dasarathy, 1991; Han and Kamber, 2006). Bütün eğitim verileri ile test verilerinin uzaklıkları tek tek hesaplanır ve belirlenecek k değerine göre sınıflandırma sonucuna karar verilir.

Şekil 3.1'de görüldüğü gibi $k=3$ alındığında verinin sınıfı üçgen olarak belirlenecekken$k=5$ alındığında verinin sınıfı kare olmuştur. $k$değerinin yüksek seçilmesi benzemeyen dokümanların işleme dahil edilmesine, düşük seçilmesi benzeyenlerin dahil edilmemesine neden olabilir.



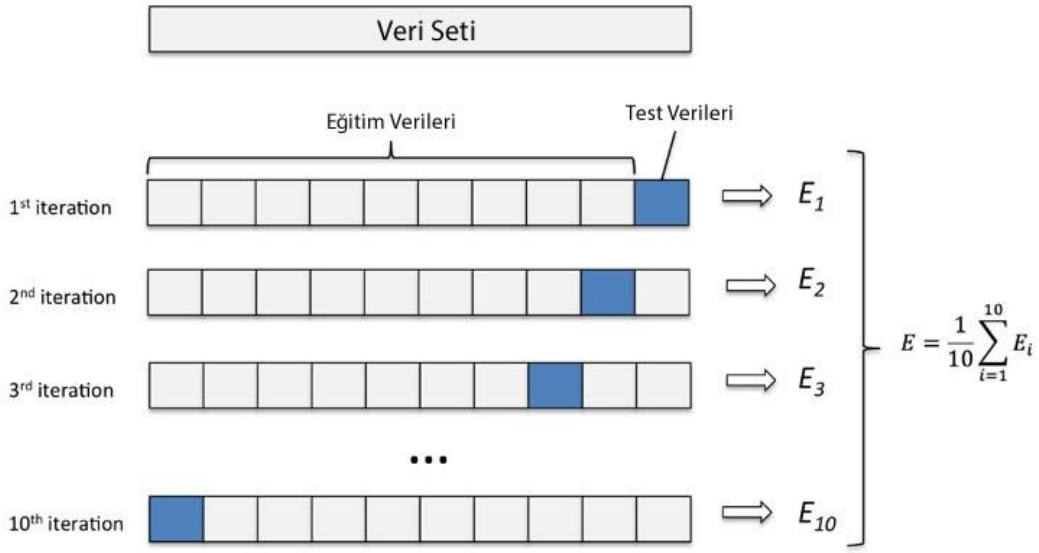
Şekil 3.1.*k* = 5 ve *k* = 3 için KNN sınıflandırma sonucu.

Çoğunluğun seçimi ilkesine dayanan KNN algoritmasının dezavantajı, eğitim verilerindeki dengesizliktir. Bir sınıfa ait eğitim verisi sayısının başka bir sınıftaki veri sayısından fazla olmasıdır. Bu durumda, diğerlerine göre daha fazla sayıda eğitim verisi bulunduran sınıf nesnelerininkuzunluklu küme içerisine girme olasılığı artmaktadır.

3.5.2.K katlamalı çapraz doğrulama (K fold cross validation)

Veri madenciliği uygulamalarında, bir yöntemin başarımının test edilmesiamacıyla, veri seti, eğitim ve test olmak üzereiki ayrı veri kümesine ayrılır. Bu işlem birkaç farklı şekilde gerçekleştirilebilir. Örneğin veri setinin %33'lük bir bölümünütest, %66'lık bir bölümü ise eğitim için ayrılır.Eğitim seti ile sistem eğitildikten sonra test seti ile başarısının test edilmesiyöntemi eğitim ve test verilerinin belirlenmesinde kullanılabilir. Aynı şekilde veri setinin tamamı eğitim seti olarak da kullanılabilir. Bu durum çok önerilen bir yöntem olmamakla beraber, veri seti optimizasyonu için kullanılabilecek bir yöntemdir. Eğitim ve test verilerininsırayla değil de rasgele olarak belirlenmesi de farklı bir yöntemdir. Bu durumda başarının her uygulamada değişimi ile karşılaşılabilir. Başarı ölçümlerinin ortalamalarının alınması ile genel bir başarı değeri elde edilmiş olur.

Kkatlamalı çapraz doğrulama yönteminin uygulaması incelendiğinde sırayla şu adımlar izlenir: İlk adım olarak bir k değeri seçilir.K= 10 için veri kümesi öncelikle 10 eşit parçaya bölünür. Örneğin veri setinde 3000kaydın bulunduğu varsayılsın. Bu veri setinden eşit olarak 300'er parçalık bölümler oluşturulur.



Şekil 3.2. Çapraz doğrulama örneği.

Verisetibelirlenen k değeri kadar eşit parçaya ayrıldıktan sonra kkatlamalı çapraz doğrulama sisteminin çalışması başlar. Şekil 3.2'deki örneğe10 katlamalı çapraz doğrulama yöntemi uygulanmış denilebilir. İlk olarak 10 eşit parçadan birisi test için seçilir, bu durumda veri setinin geriye kalanı eğitim için kullanılacaktır. Bu noktada uygulamaya hangi test parçasından başlandığının bir önemi yoktur.

Uygulamanın amacının sınıflandırma olduğu varsayılırsa, sistemin her adımında bir sınıflandırma algoritması çalıştırılır ve bir sonuç elde edilir.İlk çalışmanın ardından elde edilen birinci sonuca S1 denilsin. Ardından aynı işlem ikinci parça seçilerek tekrarlanır ve S2 sonucu bulunur. Üçüncü, dördüncü, beşinci ve onuncu adıma kadar toplamda 10 sonuç elde edilmiş olur.

Neticede 10 kere aynı yöntem, 10 farklı test ve eğitim kümelerinde çalıştırılmışolur. Sisteme ait genel başarı veya genel hata (error rate) hesaplanırken, bulunan10 sonucun ortalaması alınır. Uygulamanın son adımında ortalama alındığı görüldüğünde ve ortalama değer hesaplanırken kullanılan toplama işleminin yer değiştirme özelliği hesaba katıldığında, uygulamaya hangi parçadan başlandığının bir önemi yoktur.

Özetlemek gerekirse,k katlamalı çapraz doğrulama için k sefer yöntem çalıştırılır. Uygulamanın her adımında veri setinin$^{1}/_{k}$kadarlık, daha önce test kümesi için kullanılmamış parçası, test için alınır ve veri setinin geri kalan kısmı eğitim için kullanılır.Aşağıda bu yönetiminformül gösterimi verilmektedir.

$$t_i \in VK$$

$$Sonuç = \frac{\sum_{i=0}^{k} SF(t_i, VK - t_i)}{k} \qquad (3.3)$$

Eşitlik 3.3'te *SF*(test, eğitim), sınıflandırma fonksiyonunu temsil eder.*VK*, veri kümesini, *k* , uygulamada kaç parça katmanın kullanıldığını ve *t* ise veri kümesinden seçilen her bir test kümesi olarak tanımlanmıştır. Eşitlik 3.3'te formülü verildiği üzere, genel sonuç, bütün sınıflandırma fonksiyonlarının performans sonuçlarının toplamının, *k* sayısına bölünmesiyleyani ortalamanın alınmasıyla bulunur.

3.6. Genetik Algoritma (GA)

Genetik algoritmalar (GA), bir problemin en iyi çözümünün aranmasını biyolojik temelli bazı işlevleri (seçme, çaprazlama, mutasyon, elitizm vb.) taklit ederek yapar. Diğer optimizasyon yöntemlerinin ve klasik arama yöntemlerinin genetik algoritmaya göre geride kalmalarının nedeni, genetik algoritmanın tek bir çözüm yerine her adımda çözümlerden oluşan bir topluluk(popülasyon) kullanmasıdır. Her adımda yeni bir çözüm topluluğu kullanıldığı için, genetik algoritmayla elde edilen sonuçlar da

bir çözüm topluluğu olacaktır. Çok sayıdaki en iyi çözümün, tek bir adımda bulunabilmesi, genetik algoritmaların en önemliözelliklerinden biridir.

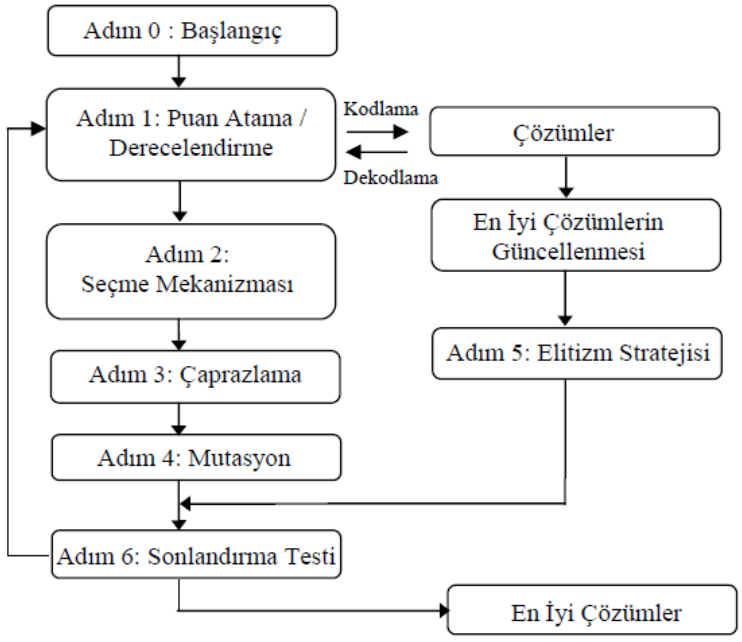
Değişik mühendislik konuları için, bilgisayar kullanılarak modellenme, benzetim, optimizasyon ve gelecek davranışlarının tahmini için, doğadaki olayların işleyiş ve davranış biçimlerinden esinlenerek ilgi çekici yöntemler geliştirilmiştir. Bunlardan biri canlıların genetik davranış biçimlerinin, modellenmesiyleortaya çıkmış olan Genetik algoritmalardır.

Genetik algoritmalar, doğada var olan bir yarışma ortamının ortaya çıkardığı, ancak daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildiği biyolojik olaylara benzetilerek geliştirilmiş paralel ve global bir arama tekniğidir. GA, arama uzayında aynı anda birçok noktayı değerlendirebilir.Bu sebepleglobal çözüme ulaşma olasılığı fazladır. Genetik algoritmalar (GA) temelinde kılavuzlanmış bir rasgele sayı üretme tekniğidir, yani parametreler için belirli sınırlar vardır. Rasgele sayılar bu sınırlar içinde üretilir. Genetik algoritmalar, ikili ya da gerçel sayı dizi yapıları içinde en iyi olanların bir sonraki nesle aktarılması şansı ile yapılandırılmıştır. GA rasgele bilgi alışverişi olaylarını birleştirerek bir arama algoritması oluştururlar. Genetik algoritmalar türevsel bir niteliğe sahip olmadığından analitik değildirler dolayısıyla tekrarlandığında aynı sonuçlaralınmayabilir. (Ergül, 2010)

GA konusunda ilk çalışmaları yapan kişi Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland'dır(Holland, 1975). Holland,mekanik öğrenme konusunda çalışmalar yapmış özellikle Darwin'in evrim kuramından bilişim tabanlı çıkarımlarda bulunmuştur.Canlılarda yaşanan genetik süreçleriyazılımsal temellerle problemler üzerinde modellemiştir. Öğrenme yeteneğini geliştirmek için sadece bir mekanik yapının seçilmesi yerine,mekanik yapılardan oluşan bir topluluğun çoğalma, eşleşme, mutasyon vb. genetik süreçlerden geçerek bir sonraki nesle aktarılan başarılı yeni bireyler oluşturabildiğini görmüştür. Çalışmalarını yaparken hedefini, arama ve en iyiyi bulma olarak belirlemiş vesonuca,doğal seçme,evrim ve genetikten yola çıkarak varmıştır. Biyolojik sistem incelendiğinde bireyin yaşadığı çevreye uyum sağlaması örnek alınmıştır.En iyi nesli bulma ve makine öğrenme problemlerinde evrimsel tabanlı bilgisayar yazılımı bu konulara odaklanarak modellenmiştir. Holland

çalışmalarının sonucunu 1975 yılında bir kitapta yayınlamış ve Holland'ın geliştirdiği yöntemin adı Genetik Algoritmalar olarak literatüre girmiştir.

Genetik algoritmaların genel yapısı Şekil 3.3'de verilmektedir. GA başlamadan önce optimizasyon problemini tanımlayan bir amaç işlevinin belirlenmesi gerekmektedir. Bu işlemden sonra GA adımları, çözüm önerilerini yani bireyleri içeren bir başlangıç topluluğunun belirlenmesiyle ve probleme ilişkin parametrelerin girilmesiyle başlar. Başlangıç topluluğu rasgele belirleneceği gibi dışarıdan kullanıcı tarafından da girilebilir. Daha sonra, amaç işlevinin hesabı parametrelere ve girişlere göre yapılır. En iyi birey/bireyler bellekte saklanır (Elitizm). Seçme mekanizmasıyla (RÇSveya TS) bir sonraki nesli yani çözüm önerilerini üretecek anne-baba bireyler seçilir. Çaprazlama ile çocuk bireyler (yeni çözüm önerileri) üretilir. Farklı bireylerin oluşturulması (çeşitlilik) için ise mutasyon işlemi uygulanır. En iyi birey/bireyler çözümlere eklenir. Girilen adım sayısına ve sonlandırma kriterine göre GA'nın bir adımı tamamlanmış olur.



Şekil 3.3. Genetik Algoritma'nın genel yapısı.

Genetik algoritmalar problem sayısına göre tek amaçlı veya çok amaçlı, parametrelerin kodlanma biçimine göre gerçel ya da ikili, problem tipine göre ise kısıtlamalı veya kısıtlamasız olarak adlandırılırlar.

3.6.1. Genetik algoritma operatörleri

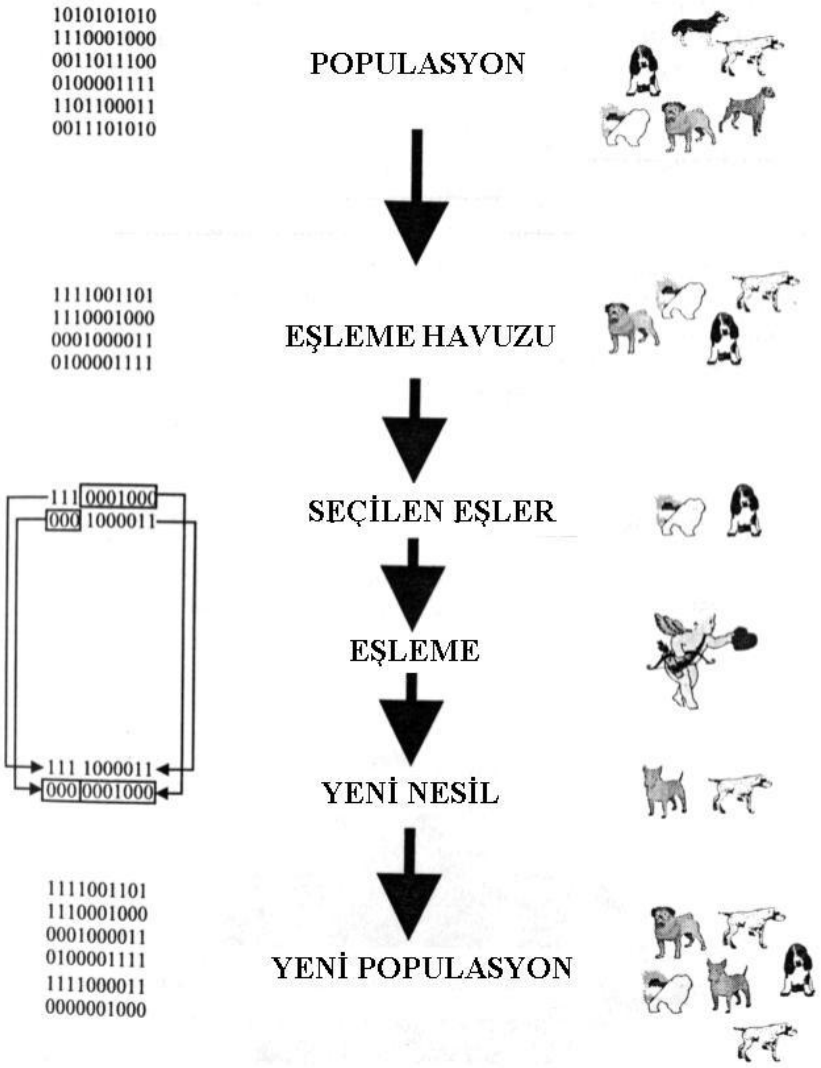
GA'larda üç temel işlev bulunmaktadır: seçme, çaprazlama ve mutasyon. Aşağıda bu işlevler ayrıntılı bir şekilde ele alınmaktadır. Bu işlevler dışında, genetik algoritmalarda kullanılan bazı parametreler de vardır. Örneğin topluluk boyutu ya da birey sayısı. Bu parametre, probleme göre belirlenir ve çok iyi seçilmesi gerekir. Parametrelerin çözünürlüğü ya da bit sayısı ise GA'nın başarımını etkileyen önemli bir faktördür.

3.6.1.1. Kodlama

Genetik algoritmalarda bireylerin kodlanması ikili sayı kodlama, gerçel sayı kodlaması veya gray kodlama olmak üzere üç faklı şekilde yapılabilir. Bu tezde, bireyler ikili sayı biçiminde kodlanmıştır. Verilen parametreler gerçel sayı ise, bu durumda gerçel sayılar ikili sayılara dönüştürülür ve uygunluk hesaplamalarında tekrardan gerçel sayılara dönüşüm yapılır.

Genetik Algoritmalar, bir veri kümesi için en iyi çözümü bulabilir.Bunu yaparken de belirli bir parametre uzayını kullanır.GA doğrusal olmayan optimizasyon aracıdır. GA, çözüm uzayında mümkün olan çözümler kümesinin rasgele üretilmesiyle başlayabileceği gibi ilk değerler önceden de belirlenebilir.Uygunluk fonksiyonu üzerindeki bulunacak her özel çözüm noktası arama uzayı içerisinden (kromozom veya nesil uzayı) bulunur. Her jenerasyonda üretilenbu nesiller kümesine popülasyon denir. Bir popülasyonun en iyi çözümünün bir parçası alınır. Bu parça popülasyonun yarısı veya dörtte biri olabilir. Bu parçadançocuklar (yeni nesil) üretilir. Bu yeni neslin eski nesillerden daha iyi uygunluk değerleri vermesi beklenir.

Biyolojik evrim ile ikili kodlarla çalışan GA arasındaki benzerlik Şekil 3.4'te görülmektedir. Her ikisinde de popülasyonun üyeleri rasgele başlar. Köpeklerin her birinin karakteristik özellikleri, sol taraftaki satırlarda verilmiştir.



Şekil 3.4. İkili kodlu GA ile biyolojik evrim arasındaki benzetim (Haupt, 1998).

Örneğin elde bulunan birkaç köpekle en iyi havlayan köpeklerin üretilmesiistensin. Bu durumda her bir köpeğin ikili sayı sistemiyle kodlanması gerekir. Popülasyondan rasgele iki köpek seçilerek eşleştirilir ve yeni yavruların üretilmesi sağlanır.Eşleştirme sonucunun iyi havlayan köpeği verme ihtimali yüksektir.

Oluşturulanyeni nesil, eşleştirme havuzuna tekrar atılır. Başta yapılan işlemler tekrarlanarakeşleştirme prosedürü tekrar uygulanır (Goldberg, 1993). Bu işlemelere son nesilde en iyi havlayan köpeklerin elde edilmesine kadar devam edilir.

Optimizasyon döngüsünü başlatmadan önce, optimize edilecek parametreler uygunşekillere dönüştürülmelidir. Bu işleme kodlama (encoding) denir. Kodlama işlemi genetik algoritmalar için büyük önem taşıyan bir konudur. Kodlama adımı doğru yapılmazsa sistemde gözlemlemek istenilen bilgiye bakış açısı büyük ölçüde daralabilir. Bu da çözümün yanlış yönde aranmasına neden olur.Kromozomlar ya da başka bir değişle gen stringleri problemlerin özel bilgilerini depolar. Değişken stringleri gen olarak da isimlendirilirler. Her problemin kendisine özgü bir değişken aralığı vardır ve bu değerler ikili veya reel sayı şeklinde gösterilebilir.

Eğer bir $x_j$değişkeni için istenilen kesinlik değeri biliniyorsa (örneğin 1000), o zaman gerekli bit sayısı m_jşöyle hesaplanabilir:

$$2^{m_j} - 1 < (b_j - a_j) \times 10^3 \leq 2^{m_j} - 1 \qquad (3.4)$$

Burada;

a_j = Değişkenlerin alt sınır değeri,

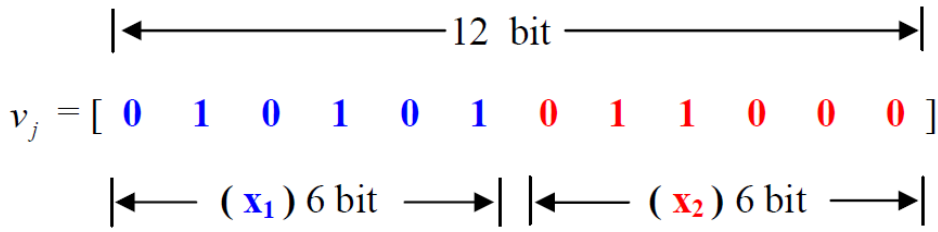
b_j = Değişkenlerin üst sınır değeri,

m_j =Gen'in bit sayısıdır.

İkili bir sayıyı gerçel sayıya çevirmek için aşağıdaki işlem yapılır (xj parametrelerdir):

$$x_j = a_j + decimal(100110 \ldots 01) \times \frac{(b_j - a_j)}{2^{m_j} - 1} \qquad (3.5)$$

Sınırlara göre temsil edilen bit sayısı değişmelidir. Sınır büyükse daha fazla bit, küçükse daha az bit parametrenin temsili için kullanılır. Aşağıdaki örnekte her parametre 6 bitlik ikili sayılar olarak kodlanmıştır. Yalnızca iki parametre olduğu için, bir bireyin (ya da çözüm önerisinin) boyutu toplam 12 bit olarak gösterilir:



Şekil 3.5.İki parametreli bir bireyin ikili kodlama gösterimi.

x_1 ve x_2'nin değerleri aşağıdaki gibidir:

	İkili Sayı	Ondalık Sayı
x_1	0 1 0 1 0 1	21
x_2	0 1 1 0 0 0	24

Şekil 3.6. İkili kodlanmış genlerin onluk sistemde sıra değerleri.

x_1 ve x_2'nin gerçek değerleri aşağıdaki gibi hesaplanır:

$$x_1 = 10 + 21 \times \frac{(10 - (-10))}{2^6 - 1} = -3.333 \quad x_2 = 10 + 24 \times \frac{(10 - (-10))}{2^6 - 1} = -2.381$$

Parametrelerin alt sınırının -10 üst sınırının 10 olduğu varsayıldığında bir birey için gerçek değerler yukarıdaki gibi hesaplanır.

3.6.1.2. Seçme

Seçme işlemi, en basit şekliyle puan değerine bağlı olarak, sonraki topluluğun yani ebeveyn (anne-baba) adayları arasından bazı bireylerin seçilmesi olarak tanımlanır. Başka bir deyişle, bir ebeveyn topluluğu bir seçme mekanizması tarafından seçilir. Sonuçta, hangi bireylerin çaprazlama ve mutasyon işlevlerine uğrayacağı

belirlenmektedir. Bu bölümde sık kullanılan iki seçme mekanizmasından bahsedilecektir:

Rulet Çarkı Seçimi: Bu yöntem, puan orantılı bir seçme mekanizmasıdır.Puan değerlerine bağlı olarak olasılık dağılımına göre ebeveyn topluluğunu seçmektedir. Rulet çarkı, aşağıda belirtilen adımlar kullanılarak uygulanır:

Adım 1. Her birey için puan değerleri *(puan(i))* hesaplanır *(i=1,2,...,N).* Burada N topluluk boyutu ya da birey sayısıdır.

Adım 2. Her bireyin puan değerleri birbirine eklenerek topluluğun toplam puanı (TP) hesaplanır:

$$TP = \sum_{i=1}^{N} puan(i) \tag{3.6}$$

Adım 3. Her birey için, seçilme olasılığı (p_i) hesaplanır:

$$p_i = \frac{puan(i)}{TP} \tag{3.7}$$

Adım4. Her birey için, toplam (kümülatif) olasılık (q_i) hesaplanır:

$$q_i = \sum_{j=1}^{k} p_j \tag{3.8}$$

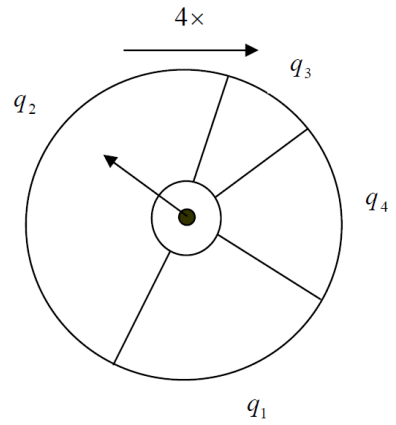
Seçme işlemi rulet çarkının *N* kez döndürülmesiyle başlar; çarkın her döndürülmesinde aşağıda belirtilen şekilde tek bir birey yeni topluluk için seçilir.

Adım 5. [0,1] aralığında rasgele bir r sayısı oluşturulur (rulet bir kez döner).

Adım6. Eğer $r \leq q_1$ ise o zaman ilk bireyi seç (v_i); aksi takdirde k. birey $v_k(2 \leq k \leq N)$için $q_{k-1} < r < q_k$ ile seçilir.

5. ve 6. adımlar yeni topluluğunoluşması için *N* kez tekrarlanır. Rulet Seçimi büyük bir seçme hatasının oluşmasına neden olabilir. Bu yöntemde, aynı bireyin *N* kez

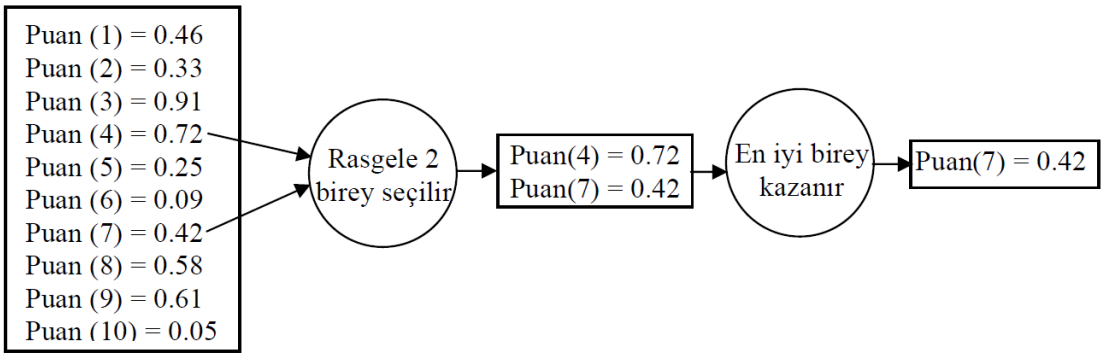
seçilebilme olasılığı vardır. Rulet Çarkı'nın gösterilimi (dört bireylik topluluk için) Şekil 3.7'de verilmektedir.



Şekil 3.7. Rulet Çarkı Seçimi.

Turnuva Seçimi: Bu seçim çok kullanılan seçme mekanizmalarından birisidir. Turnuva seçiminde, birkaç adet birey o anki topluluktan rasgele seçilir ve bu bireylerden amaca göre en iyisi (probleme göre en düşük puanlı ya da en yüksek puanlı) yeni toplulukta yerini alır. Seçilme baskısının miktarı turnuvanın boyutu ayarlanmasıyla kontrol edilebilir ve böylece yakınsama hızı da ayarlanmış olur. Turnuva seçimi, temelde her çalışmasında yalnız bir kazanan olacağını varsayar. Seçme işlemi, gelecek nesillerin tamamlanmasına yetecek miktarda tekrarlanır. Eğer seçme mekanizması turnuva olan bir GA tasarlamışsanız ve turnuvanın boyutu bireylerin sayısına eşitse, o zaman seçilen birey o anki jenerasyonun en iyi bireyi olacaktır. Örneğin, 500 bireyden 50 tanesi seçilecekse, rasgele olarak bir grup birey seçilir (10 tane), bunlar bir torbaya konulur, bu 10 tane birey içinden en iyisi alınır ve 1. sıraya konulur. Daha sonra ikinci bir 10 birey seçilir ve bunun en iyisi de 2. sıraya konulur, bu işlem oluşturulacak birey sayısı tamamlanıncaya kadar devam eder.Daha çok bireyden en iyi olanını seçmek,uygunluğu en iyi olan bireyin bulunmasını kolaylaştırabilir, dolayısıyla torbaya konulacak birey sayısı değişebilir. Turnuva seçimi puan bazlı bir seçilme olasılığı kullanmaz, sadece o anda torbada bulunan bireylerden en iyi olanı seçilir. Şekil 3.8' de, 10 bireylik bir topluluktaki bireylerin puanları verilmiştir. Bu problem minimizasyonproblemidir, dolayısıyla uygunluk değeri en küçük olan birey en iyi bireydir.Şekilde 3.8'de bu

bireyler arasından ikili TS kullanılarak yapılan bir seçme işlemine ilişkin örnek verilmektedir.

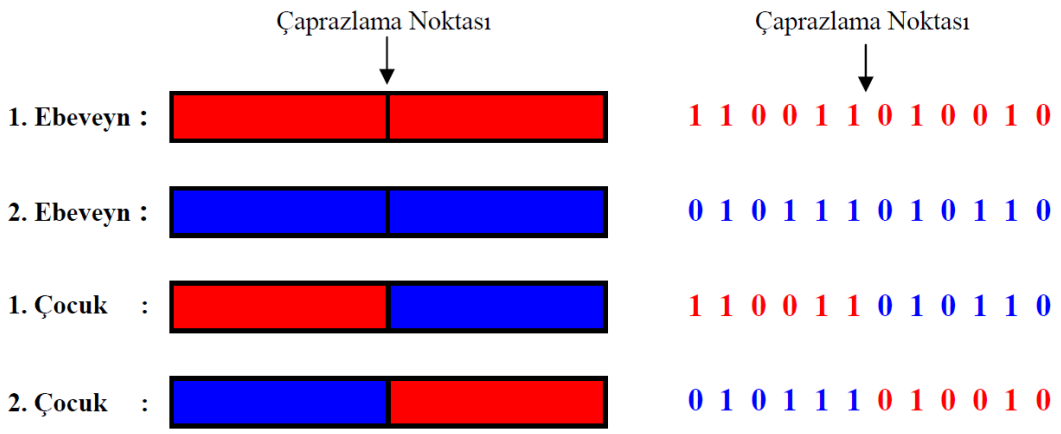


Şekil 3.8.İkili turnuva seçim örneği (min problemi).

3.6.1.3. Çaprazlama

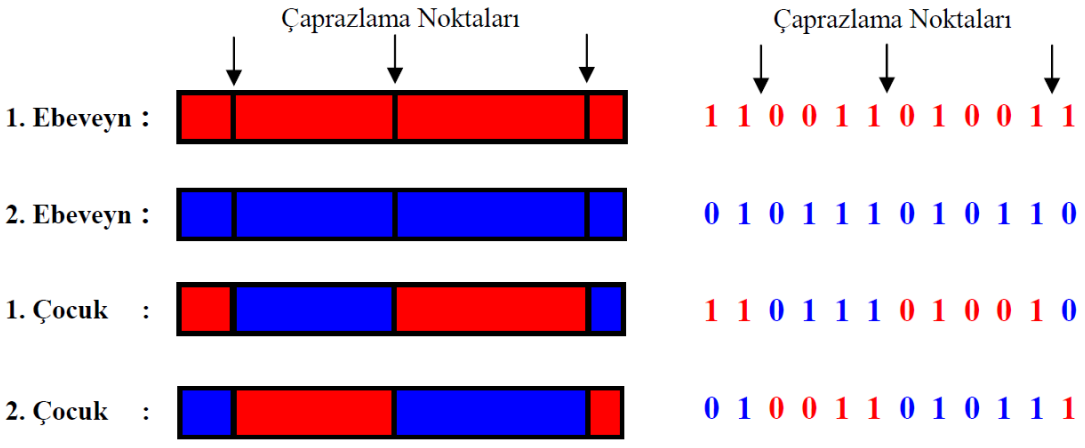
Çaprazlama, ebeveyn eşleşme havuzundan seçme mekanizması kullanılarak seçilen bireylerden yeni bireyler (çözümler) üretilmesi için kullanılır. Literatürde birçok çaprazlama işlevi önerilmiştir, ancak hemen hemen tüm çaprazlama işlevlerinde, ebeveyn havuzundan rasgele seçilen iki birey, ikili kodları bir noktadan kesilerek bölümlere ayrılır ve bu iki bireyin kodları karşılıklı olarak yer değiştirilerek iki yeni birey elde edilir. Çaprazlama işlemi tek noktadan yapılabileceği gibi daha fazla noktadan, düzgün (uniform) veya karışık (shuffled) yapılabilir. Tek noktalı çaprazlamada bireyler rasgele seçilmiş bir noktadan ikiye ayrılır ve karşılıklı ikili kodlar yer değiştirir. Bu tezde tek-noktalı çaprazlama işlemi kullanılmıştır. Tek noktalı çaprazlama işlemine ilişkin bir örnek, Şekil 3.9’ da verilmektedir.

Çaprazlama işlemi sonrasında elde edilen bireylerin çaprazlamaya uğrayan bireylerden daha iyi uygunluk sonuçları üretmeleri beklenmektedir, ancak bu her zaman mümkün değildir. Çaprazlamayla ebeveynlere kendilerinden daha iyi bireyler üretebilme şansı verilmektedir. Eğer kötü bireylerin üretildiği varsayılırsa,bu bireylerin büyük olasılıkla bir sonraki GA adımında elenecekleri söylenebilir.



Şekil 3.9. Tek noktalı çaprazlama örneği.

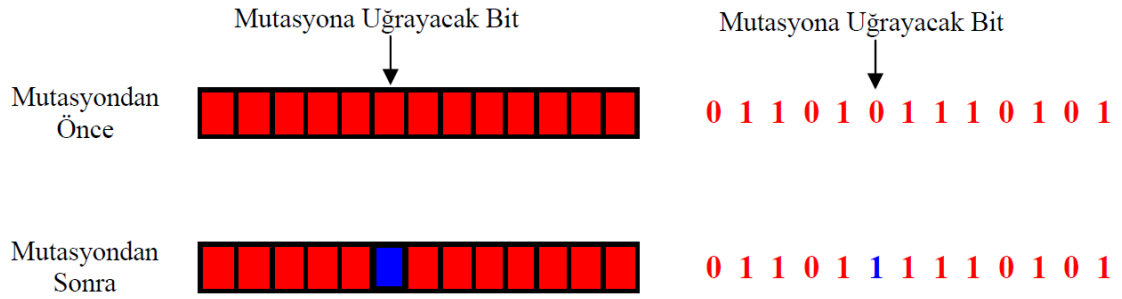
Çaprazlamada çeşitliliğin azalması ihtimali vardır. Eğer çaprazlamaya uğrayacak bireylerin ikili kodları birbirine çok benzerse, çaprazlamadan sonra elde edilen çocuk bireyler de birbirine çok benzeyecektir, bu da topluluktaki çeşitliliği azaltacaktır. İkili kodları birbirinden farklı bireylerin çaprazlaması çeşitliliği artırır ve daha iyi sonuçlar üretilebilir.Eğer çaprazlama olasılığı p_c=1 (%100) seçilirse, bu tüm bireylerin çaprazlamaya uğrayacağı anlamına gelir. Çaprazlama işlevi birden çok noktadan da uygulanabilir. Şekil 3.10'da çok-noktalı bir çaprazlama örneği verilmektedir.



Şekil 3.10. Çok noktalı çaprazlama örneği.

3.6.1.4. Mutasyon

Mutasyon işlevi de çaprazlama işlevi gibi genetik algoritmanın arama yönüne odaklanır ve bu işlevde birey çeşitliliği amaçlanır. İkili kodlama ile kodlanmış bireyler bit-tabanlı mutasyona uğrayarak, p_m mutasyon olasılığına bağlı olarak bireyin ikili kodundaki rasgele bir bit 0 ise 1, 1 ise 0 yapılır. Mutasyonun uygulanması için rasgele seçilmiş sayının, mutasyon olasılığından düşük olması gerekir. Mutasyonun en önemli işlevi popülasyonun çeşitliliğinin sağlamasıdır. Yapılan çalışmalar mutasyon olasılığının düşük seçilmesi gerektiğini göstermiştir. Mutasyon olasılığı yüksek seçilirsepopülasyonun çeşitliliği azalır, yani popülasyona farklı bireylerin dahil olma olasılığı azalır. Genellikle mutasyon olasılığı $1/Binary\_Dizi\_Uzunluğu$ olarak seçilir. Örneğin, 10 bit uzunluklu 10 bireylik bir topluluk için, mutasyon olasılığı 0.01 (%1) seçilirse, bununanlamı en azından iki bitin mutasyona uğrayacağıdır. Şekil 3.11'de mutasyon işlevinin nasıl bir değişikliğe yol açtığı gösterilmektedir.



Şekil 3.11. Mutasyon örneği.

3.6.1.5. Elitizm

Elit birey, bir popülasyondaki uygunluk değeri istenilene en yakın olan, yani puanı en iyi olan birey demektir. Seçkinciliğin veya elitizminyani en iyi birey/bireylerin saklanması ve bir sonraki topluluğa eklenmesinin GA'nın başarımına önemli ölçüde katkı sağladığı görülmüştür (Zitzler, 1999). Elitizm dikkatli bir biçimde uygulanmalıdır. Eğer elitizm kontrollü bir biçimde uygulanmazsa popülasyonda çeşitlilik kaybı söz

konusu olabilir. Bu da aynı uygunluk değeri veren bireylerin çoğalmasına neden olur. Tek amaçlı GA'larda en iyi birey sayısı birdir, dolaysıyla elitizmin uygulanması kolaydır. Elitizim uygulanırken elit birey saklandıktan sonra, bir sonraki adımda ya en kötü bireyle yer değiştirilmekte ya da basitçe topluluğun en sonundaki bireyle yer değiştirilmektedir. Ancak çok amaçlı GA'larda amaç fonksiyonların fazlalığından kaynaklanan tek bir en iyi bireylerden oluşan bir küme söz konusudur. Bu durumda elitizmin uygulanması zorlaşır ve elitizm tek amaçlı durumda olduğu gibi kolay ve tek bir biçimde gerçekleşmez.

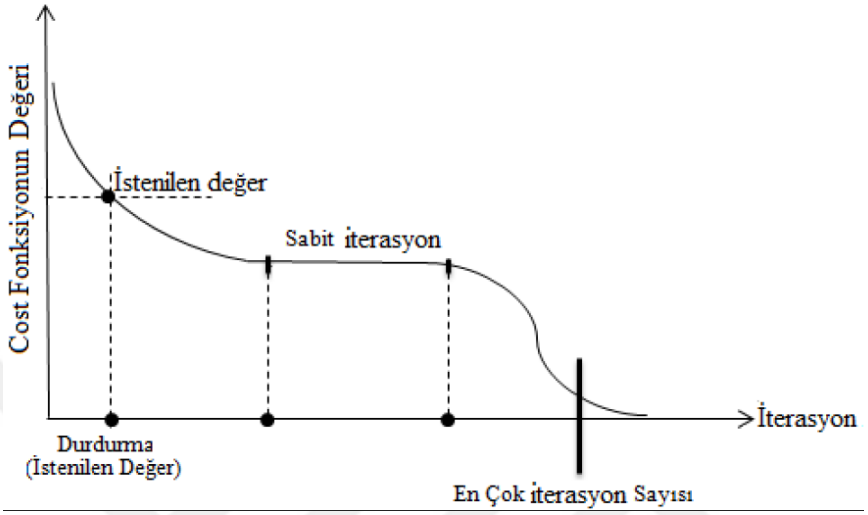
3.6.1.6. Durdurma kriteri

İterasyon ve zaman kriterlerine göre iki tür sonlandırma durumu vardır. Şekil 3.12'de gösterildiği gibi eğer sabit iterasyon sayısı belli bir miktardan fazla olursaozaman durdurma sırası yani algoritmanın sonlandırılmasının vakti gelmiştir. Algoritma zamanabağlı olarak da çalıştırılabilir. Örneğin eğer belirli bir süreden fazlaçalışması durumunda ve uygunluk değerlerinde bir değişiklik tespit edilmemesi halinde ya da değişimlerin genliğinin çok az olması durumunda algoritmanın sonlandırılmasının vakti gelmiştir ve programın daha fazla çalışmasının bir yararı olmayacaktır.

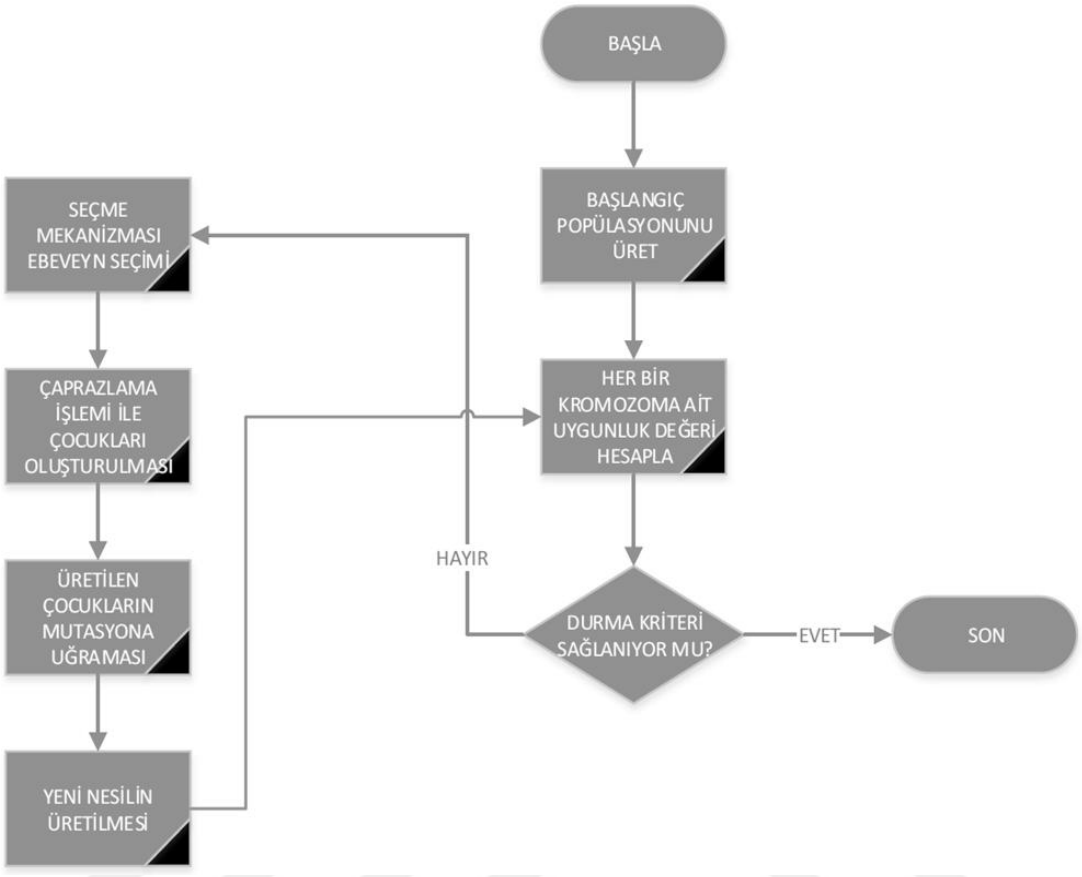
NFE (number of function evaluation) (Ali ve ark. 2009) genetik algoritmaların performans ölçümünde oldukça güvenilir bir yöntemdir. Önceki koşullar yani zaman ve iterasyon sonlandırmaları bilgisayarın performansına göre değişim gösterebilir. Ama NFE bu durumdan bağımsız bir kriterdir.Aşağıdaki eşitlik kullanılarak NFE hesaplanabilmektedir.

NFE: Amaç fonksiyonun çağırılma sayısı
APS: Ana popülasyonun sayısı
ÇS: Çocukların sayısı
MS: Mutasyonluların sayısı
İS: İterasyon sayısı

NFE = APS + (ÇS + MS) * İS (3.9)



Şekil 3.12. Durdurma kriteri örneği.



Şekil 3.13. Genetik algoritma akış şeması.

3.7. Çok Amaçlı Optimizasyon

Gerçek hayatta karşımıza çıkan problemlerin çoğunun çözümü için birden fazla amacın optimize edilmesi gerekmektedir. Çok amaçlı optimizasyon problemlerinde çoğu zaman amaçlar çelişir. Bu nedenle çok amaçlı optimizasyon problemlerinin çözümü, tek amaçlı optimizasyon problemlerine göre çok daha zordur. Tek amaçlı optimizasyon problemlerinde tek bir optimum çözüm üretmek mümkündür ancakçok amaçlı optimizasyon problemlerinde bu durum mümkün olmamaktadır. Çok amaçlı optimizasyon problemlerinde çözüme ulaşmak, sonlu uzunluktaki bir çözüm kümesinden,üzerinde bütün amaçların mutabık kaldığı bir çözümün seçilmesi ile mümkün olabilmektedir.

Çok amaçlı problemler ile tek amaçlı problemler arasındaki en önemli fark; tek amaçlı problemlerde tek bir optimal çözüm veya onun alternatifi olan optimal çözümler kümesi söz konusu iken, çok amaçlı problemlerde aynı anda her amacınüzerinde mutabık kaldığı en iyi uygunluk değerlerini veren bir çözümün olmamasıdır. Bu problemlerde tüm amaçlar değerlendirmeye katılırve sadece bir amaca göre çözüm aramak yanlıştır. Amaçlar çoğuzaman birbirleriyle çelişir. Bu yüzden birindeki iyileşme diğer amaçlardan en az birinde kötüleşmeye yol açabilir. Çelişen amaçların ortaya çıkardığı bu sorun nedeniyle bulunan çözümler arasında amaç fonksiyon değerlerine bakılarak amaçların ödünleşim miktarları elde edilir. Kabul edilebilir ödünleşim miktarları karar verici yöntemlere bağlıdır. Tüm bu durumlar göz ününe alındığında çok amaçlı problemlerde genelde kesin olarak bir optimal çözüme ulaşmak mümkün değildir. Elde edilen ve aralarında belirli bir ödünleşim olan çözümlerden bazıları seçilir ve en iyi çözümler olarak karar vericiye sunulur.

Genel birçok amaçlı optimizasyon problemi, *n* adet parametre (karar değişkeni), *k* tane amaç fonksiyonu ve *m* adet bir grup kısıt içermektedir. Amaç fonksiyonları ve kısıtlar karar değişkenlerinin fonksiyonu şeklinde yazılırlar. Optimizasyon işlemi eşitlik 3.10'da formüle edilmiştir.

$$\begin{aligned} &\min y = f(x) = (f_1(x), f_2(x), f_3(x), \dots., f_k(x)) \\ &Koşul: e(x) = (e_1(x), e_2(x), e_3(x), \dots., e_m(x)) \leq 0 \\ &x = (x_1, x_2, x_3, \dots, x_n) \in X \\ &y = (y_1, y_2, y_3, \dots, y_k) \in Y \end{aligned} \tag{3.10}$$

Eşitlik 3.10'da kullanılan *x* karar vektörü, *y* amaç vektörü, *X* karar uzay ve *Y* ise amaç uzayı olarak adlandırılmaktadır. Kısıtlar $e(x) \leq 0$ tanımlanan problem için erişilebilir çözümler kümesini göstermektedir (Zitzler 1999).

3.7.1.Erişilebilir küme

Erişilebilir Küme, X_f, x karar vektörü kümesi olup $e(x)$kısıtlarını sağlamaktadır.

$$X_f = \{ x \in X \mid e(x) \leq 0\} \tag{3.11}$$

Eşitlik 3.10'daki gibi tanımlanabilir. X_f'nin imajı yani amaç uzayındaki erişilebilir bölge ise Eşitlik 3.12'de tanımlanmıştır.

$$Y_f = f(X_f) = \cup\, x \in X_f\{f(x)\} \tag{3.12}$$

3.7.2.Dominantlık

Herhangi iki karar vektörü (Örn: x_1, x_2) için dominantlık kavramı aşağıdaki gibi verilmektedir. Problem bir minimizasyon problemidir.

- $x_1 \prec x_2$ x_1, x_2'ye göre dominanttır ve bütün amaçlarda: $f(x_1) < f(x_2)$
- $x_1 \preccurlyeq x_2$ x_1, x_2'nin zayıf dominantıdır ve en az bir amaçta: $f(x_1) \leq f(x_2)$
- $x_1 \cong x_2$ $x_1\ ve\ x_2$birbirlerini domine edemez:$f_1(x_1) \leq f_1(x_2)\ ve\ f_2(x_2) \leq f_2(x_1)$

3.7.3.Pareto-Optimalküme

Pareto-Optimal kavramı 1900'lü yılların başlarında, İtalyan bir iktisatçı ve sosyolog olan Vilfredo Pareto tarafından bulunmuştur. Pareto-Optimal kavramı, evrimsel algoritmalara uyarlanmış ve çok amaçlı evrimsel optimizasyon algoritmaları geliştirilmiştir. Buna göre, bir çözüm amaç, değerine göre, en iyi, en kötü ve diğer çözümlere eşit olabilir. En iyi çözümkavramı, amaçların herhangi birinde en kötü olmayan ve en azından bir amaçta diğerlerinden daha iyi olan çözüm anlamındadır. En iyi kavramı problemin en küçükleme ya da en büyükleme problemi olmasına göre

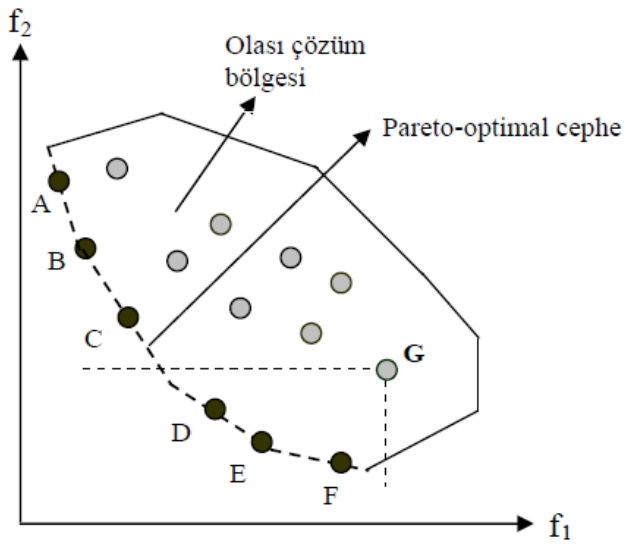
farklılık gösterir. Pareto-Optimal çözüm, arama uzayında, herhangi bir diğer çözüm tarafından bastırılamayan çözümdür (Deb, 2001).

Tüm amaç fonksiyonları göz önünde bulundurularak, bastırılamayan/domine edilemeyen tüm çözümlerin kümesi, Pareto Ön Yüzü ya da daha yaygın kullanılan tabirle Pareto Cephesi olarak adlandırılır.

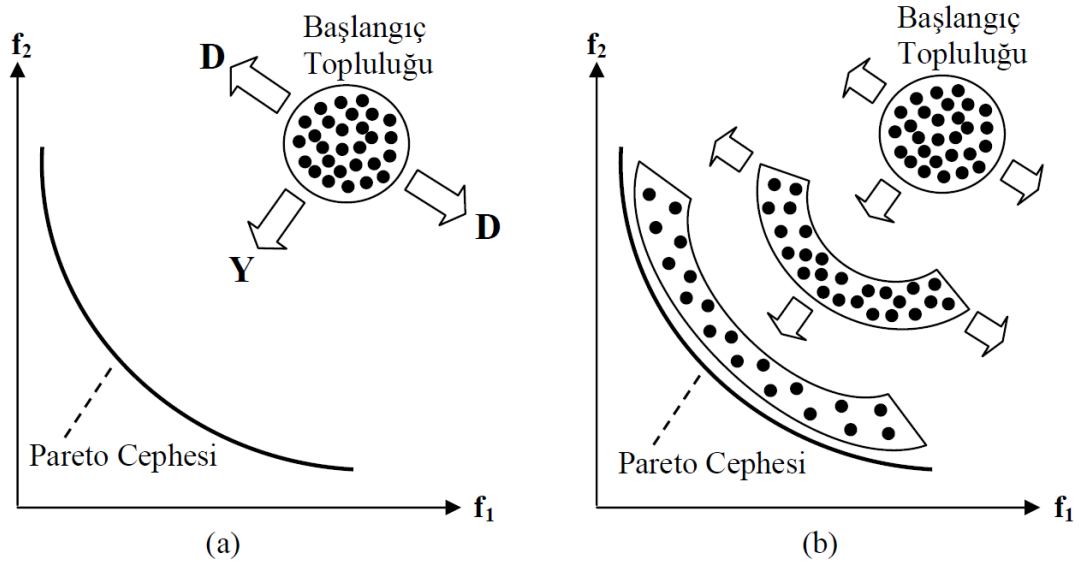
Şekil 3.14'te iki amaçlı bir en küçükleme problemi üzerinde, Pareto-optimal (veya kısaca Pareto) bireyler, bu bireylerden oluşan P^*={A,B,C,D,E,F} Pareto-optimal küme ve Pareto-optimal cephe gösterilmektedir. Çizgili bölge bir minimizasyon problemi için Pareto-optimal cepheyi göstermektedir. Siyah noktalar Pareto bireyleri, gri noktalar Pareto olmayan bireyleri göstermektedir. Örneğin, "G" bireyi, "D, E" ve "F" Pareto bireylerinin etki alanına girmektedir (Ergül, 2010).

Çok amaçlı problemlerin çözümünde elde edilecek çözümler bazı özelliklere sahiptir. Elde edilecek son çözüm kümesinin, problemin Pareto optimal kümesine yakınsanması ya da karar vericinin kabul edebileceği çözümlere ulaşması istenir. Bu durumun gerçekleşmesi için çözüm algoritmasının, çözüm arama sırasında uyguladığı seçme ve seçtiği çözümlerin değerlendirilme şekli büyük önem taşımaktadır.

Tek amaçlı GA'larda tek bir en iyi sonuç vardır ve bireyler (çözüm önerileri) en iyiden en kötüye doğru tekil bir biçimde sıralanabilir. Ancak çok amaçlı GA'larda tek bir en iyi çözüm yerine bir grup en iyi çözüm (Pareto-optimal çözümler) vardır ve dolayısıyla en iyiden en kötüye doğru bireylerin sıralanması tekil değildir. Bu yüzden, birçok çok amaçlı genetik algoritma yöntemi geliştirilmiş ve geliştirilmeye devam edilmektedir. Sıralama yapmanın amacı, bir sonraki nesildeki bireylerin (çözüm önerilerinin) nasıl elde edileceğini belirleyen seçme mekanizmasına bilgi üretmektir. Yani seçme mekanizmasından önce bir sıralama yöntemine ilişkin hesaplamalar yer almaktadır.



Şekil 3.14.Pareto-optimal cephe örneği (Ergül, 2010).



Şekil 3.15.(a) Başlangıç topluluğu ve Pareto cephesi, (b) İstenilen durum.

Amaçlar tek tek hesaba katıldığında bireylerin sıralaması farklı olacaktır, bir amaca göre en iyi olan çözüm diğerlerine göre en kötü olabilir. Ancak tüm amaçlar hesaba katıldığında Pareto-optimal cephedeki bireylerin birbirlerine göre bir üstünlüğü

olmayacaktır. Çok amaçlı optimizasyonun asıl hedefi, Pareto-cephesini bulmak veya ona yaklaşmak ve bu cephe üzerinde düzgün bir dağılım sağlamaktır. Var olan tüm yöntemler bunu sağlamaya çalışmaktadırlar. Bu hedefler, Şekil 3.15'de bir en küçükleme problemi üzerinde görsel olarak gösterilmektedir. Bu şekilde D-düzgün dağılım ya da çeşitliliği, Y-yakınsamayı temsil etmektedir. Düzgün dağılımın artırılması için, başlangıç topluluğu D ile gösterilen oklar yönünde genişletilmelidir. Pareto cephesine yakınsama için ise başlangıç topluluğu Y ile gösterilen ok yönünde ilerlemelidir. Düzgün dağılım – yakınsama dengesi doğru olarak ayarlandığı zaman, Şekil 3.15b' de gösterildiği gibi iyi bir çözüm topluluğu bulunabilir. Bununla birlikte, her problem ve her çok amaçlı GA yöntemine göre böyle bir çözümü bulmak kolay değildir (Ishibuchi ve Shibata, 2004).

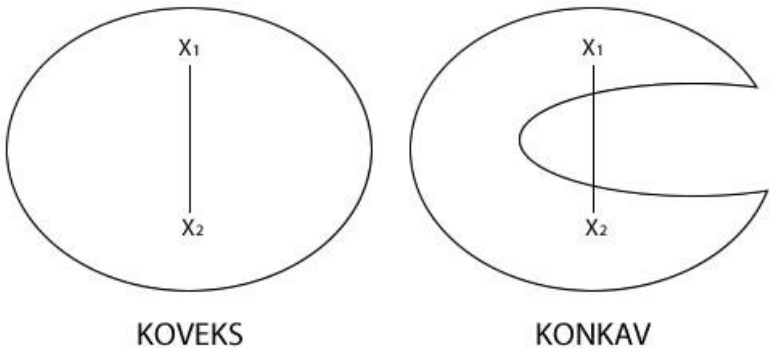
3.8. Çok Amaçlı Problemler ve Sezgisel Yöntemler

Karar verme problemleri tek ya da çok amaçlı olarak ikiye ayrılır. Tek amaçlı problemlerde hedef, kısıtlar altında en iyi çözümü bulmaktır. Çok amaçlı problemlerde ise amaç fonksiyon değerlerinin birbirlerine üstünlük sağlamadığı birden çok uygun çözümün bulunması hedeflenmektedir. Tek amaçlı optimizasyon problemleri üzerinde II. Dünya Savası sırasında çalışılmaya başlanmış ve bu konuda çözüme ulaşmak için çeşitli algoritmalar geliştirilmiştir. Doğrusal Programlama problemlerinin çözümünde Simpleks Algoritmasından, Doğrusal Olmayan Programlama problemlerinin çözümünde Kuhn-Tucker teoreminden faydalanılarak geliştirilen ve en iyi çözümü garantileyen teknikler bulunmuştur. Bu teknikler bazı varsayımlar gerektiriyordu ve uygulamada değişken sayısı arttıkça optimal çözüm bulunma maliyeti üstel olarak artıyordu. Bu sebepler hem tek hem de çok amaçlı problemlerin çözümünde sezgisel (heuristic) yöntemlerin yaygınlaşmasını sağlamıştır (Jaszkiewicz, 1998).

Çok amaçlı karar verme problemlerinde amaç uzayının şekli önemlidir. Amaç uzayı konveks ya da konkav olabilmektedir. Bu tip uzaylara basit bir örnek Sekil 3.16'daverilmiştir. Çok amaçlı bir problemin konveks bir problem olabilmesi için tüm amaç fonksiyonlarının ve uygun çözüm bölgesinin konveks olması gerekir. Bunu

sağlamayan, yani amaç fonksiyonları ve çözüm bölgesi konveks olmayan problemler ise çok amaçlı konkav problemlerdir. Problemler hem konveks hem de konkav olabilir ve bu durum problemin çözümünü zorlaştırabilir ama konkav problemlerin çözümünde daha da dikkatli olmak gerekir. Bunun sebebi amaç uzayının parçalı ya da birbirinden ayrık olabilmesidir. Çözüm arama sırasında sadece bir bölgede takılı kalmamak, diğer bölgelerde de araştırmalar yapmak çözüm kümesi çeşitliliğini sağlamak için önemlidir.

Problemin çözümü sırasında kendisinden yararlanılacak algoritma uygun olmayan çözümler üretebilir. Uygun olmayan çözümlerden yararlanmak bir avantaja da dönüşebilir, çözüm yönünün başka bölgelere sıçramasını ve belki de daha iyi sonuçların bulunmasını sağlayabilecektir. Çok fazla uygun olmayan çözüm ile ilgilenmek ise çözüm arayışında gereksiz oyalanmaya yol açıp belki de doğru çözüm bölgesine yakınsamanın hiç sağlanamamasına da neden olabilir. Bu sebeple uygun olmayan çözümlerden yararlanma kontrol altında ve kısıtlı yapılmalıdır. Özetle, konveks problemlerin çözümünü araştırmak konkav problemlere göre daha kolaydır. Hem karar hem de amaç alanlarının ikisinin ya da en azından birinin konkav olması o problem için optimal çözüm bulmayı güçleştirir.



Şekil 3.16. Konveks ve Konkav çözüm bölgeleri.

Çok amaçlı problemlerinin çoğu için optimal çözümü bulmak çok zordur ve gelişmiş bilgisayarlara rağmen zaman ve bellek maliyeti gerektiren bir iştir. Bunun sebebi çok amaçlı problemlerin çoğunun NP-zor (NP-hard) olması yani; en iyi çözüm kümesini bulmak için kullandığımız algoritmanın üreteceği çözüm sayısının, problemin

büyüklüğünün artması ile üstel olarakfazlalaşması ve buna bir sınırlamanın getirilememesidir. Böylece Pareto optimal kümenin bulunması makul bir zaman aralığında gerçekleşmez. Çözüm için gerekli zaman maliyeti problem büyüklüğünün üstel bir fonksiyonu olarak artar.

Çözüm zamanın üstel olarak artmasına bir örnek vermek gerekirse; çok amaçlı gezgin satıcı probleminde her şehre yalnızca bir kez uğranmak koşulu ile gidilen toplam uzaklığın minimum yapılması istenirken, satıcının gezdiği yerlere vardığı zamana göre değişen kâr miktarları da maksimum yapılmak istenir. 10 şehir için yaklaşık 181,000 çözüm sayısı olacak ve bu belirli bir donanıma sahip bir bilgisayarda 3 dakikada çözülebilecektir. 20 şehir olduğunda ise 10^{16} çözüm olacak ve buaynı bilgisayar ile 320,000 yılda çözülebilecektir. Şehir sayısındaki %100 artış, çözüm sayısını yaklaşık 55 milyar kat arttırmıştır ve 20 şehir için bile problemin çözümü imkânsız hale gelmiştir. Bunun sebebi çözüm sayısının, *n*şehir sayısı olmak üzere $n(n-1)!/2$ ile üstel olarak büyümesidir. Buna göre çok amaçlı gezgin satıcı probleminin NP-hard olduğu ve en iyi çözüm kümesini bulmanın çok zor olduğu görülmektedir (Jaszkiewicz, 1998). Bu tip problemlerde optimal çözüm kümesi tam olarak elde edilememesine rağmen, bu çözüm kümesine yakınsayan başka çözümlerin oluşturacağı bir kümeye ulaşmak imkânsız değildir. Bu kümedeki çözümlere kabul edilebilir bir zamanda ulaşılabiliniyorsa, elde edilen yaklaşık optimal çözümler karar vericiye problemin çözümü olarak verilebilir. Böylece problemin optimal çözümü, karar vericinin kabul edebileceği amaç fonksiyon değerlerinin bulunması ile sağlanmış olur. Yaklaşık optimal kümenin bulunması için kullanılabilecek yöntemler sezgisel yöntemlerdir(Deb, 2001).

Klasik yöntemler konveks çok amaçlı karar verme problemlerde ve amaçların tek amaca indirgenmesinde bir sakınca olmayan problemlerde en iyi çözümü garanti etmektedir. Bu özelliği ile bu yöntemlerin problemlere uygulanmaları günümüzde devam etmektedir. Çok amaçlı karar verme problemlerinin çoğu için Pareto optimal çözüm kümesine ulaşmak klasik yöntemlerle hemen hemen imkânsızdır. Bunun sebebi çok amaçlı problemlerinin çoğunun NP-zor özelliği göstermesidir. Değişkenlerin birkaçı ve hatta bazen hepsi sadece tamsayılı değerler alabilir. Bu tip kesikli değişkenli problemlerin çözüm uzayı dahi belirlenemeyebilir. Tamsayılı problemler genelde çok

karışık ve konkav arama uzayına sahiptir ve en iyi çözüm kümesini bulmak zordur. Ayrıca çok sayıda matematik ve mühendislik uygulamalarında karşılaşılan çok amaçlı problemler NP-zor problemlerdir ve NP-zor sınıfı problemlerinin çözümü için kullanılan geçerli bir yöntem yoktur. Bunlar için optimal çözümleri araştırmak ancak sezgisel yöntemlerle olabilmektedir.

3.9. Sıralı Seçkin Bastırılamayan Genetik Algoritma (NSGA II)

- p bireyinin domine ettiği popülasyon elemanlarının kümesi → S_p
- p bireyini domine eden popülasyon elemanlarının sayacı → n_p
- Tüm popülasyon elemanları için S_p ={} ve n_p = 0
- p popülasyonun her bir bireyi
- q popülasyonun her bir bireyi
- Eğer p, q bireyini domine ederse, q bireyini S_p kümesine ekle.
- Eğer q, $p$bireyini domine ederse n_p sayacını bir arttır.
- Tüm n_p = 0 olan popülasyon elemanlarını F_1 cephesine ekle.
- Cephelerin sayacı için k=1
- * Q kümesini F_{k+1}cephesinin bir taslağı olarak sakla.
- F_k cephesinden p elemanı için, S_p kümesinden q elemanı için (p bireyinin domine ettiği tüm $q$bireyleri) n_p sayacını bir birim eksilt.
- Eğer n_p = 0 ise q bireyini Q kümesine ekle. Eğer Q boş küme ise sıralama işlemi sona ermiştir.
- Eğer Q boş küme değilse F_{k+1}cephesini Q kümesine kaydet.
- k cephe sayacını bir arttır ve * adımına geri dön.

Şekil 3.17. NSGA II için sözde kodlar.

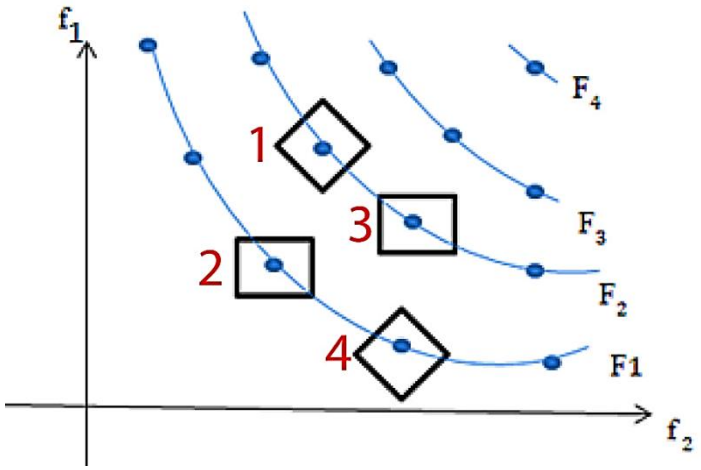
Sıralı Seçkin Bastırılamayan Genetik Algoritma (Elitist Non-dominated Sorting Genetic Algorithm, “NSGA II”) Deb ve Goel tarafından geliştirilmiştir (Deb ve Goel, 2001). Algoritma rastgele seçilen N tane çözümün bulunduğu P_1 popülasyonu ile başlar

ve bu çözümlere çaprazlama ve mutasyon operatörleri uygulanarak N tane çözümü olan yeni Q_1 popülasyonu yaratılır. P_1veQ_1birleştirilip R_1birleşik popülasyonu elde edilir. Bütün çözümler birbirleriyle karşılaştırılır ve alt edilip edilmeme durumuna göre gruplanarak $F_i$alt kümeleri elde edilir. F_1'de alt edilemeyen, F_2'de ise F_1'deki çözümlerden sonra alt edilemeyen çözümler vardır. Bu şekilde devam edilerek tüm çözümler gruplanır. Bir adımdaki orijinal popülasyon $P_t$olarak gösterilirse, N tane çözümün olması gerektiği bilindiğine göre, bir adım sonrasına gerekli P_{t+1}popülasyonunun oluşturulması,F_1kümesindeki çözümlerle baslar. F_1'deki çözümlerin sayısı N'den az ise F_2'ye geçilir. Bu şekilde devam edilerek Nadet çözüm bulununcaya kadar kümedeki tüm çözümler seçilir. N adet çözüme ulaşıldığında sonraki F kümelerine bakılmaksızın işleme devam edilir. P_{t+1}popülasyonunda$N$ tane çözüm olması gerektiği için, en son çözüm alınan kümenin eleman sayısı taşmaya neden olursa bu sefer kalabalık uzaklık atama prosedürü (crowded distance assignment procedure) ile gerekli sayıda çözüm alınır. Bu teknikte, seçim yapılacak alt kümede bulunan çözümler, önce 1. amaca verdiği değere göre iyiden kötüye sıralanır ve çözümlere ait uzaklıklar hesaplanır. En iyi ve en kötü çözümlerin uzaklığı sonsuz olarak belirlenir sonra arada kalan tüm değişkenlerin uzaklıkları hesaplanır. Daha sonra aynı kümedeki çözümler 2. amaç fonksiyonuna verdikleri değerlere göre sıralanır ve Eşitlik3.11'deki formül ile uzaklıklar bulunur. Çözümlerin 1. amaca göre bulunan uzaklık değerleri ve ikinci amaca göre bulunan değer toplanır ve böylece çözümün toplam uzaklık değeri bulunur. M tane amaç fonksiyonu varsa, bu işlem$M$ kez tekrarlanır ve bulunan toplam uzaklıklar çözümlerin son uzaklık değerleri olur. P_{t+1}için gerekli olan çözüm sayısı kadar çözüm, bulunan uzaklık değerlerine göre seçilir. Öncelikle büyük uzaklık değerine sahip çözümler alınır. Elde edilen P_{t+1} popülasyonunda en son alt kümeden seçilen çözümlerin uzaklık değerleri bilinmektedir ancak önceki seçilenlerin de uzaklık değerlerinin hesaplanması gerekir. Bu uzaklık değerleri kalabalık turnuva seçimi (crowded tournament selection) için kullanılır. Bu seçime göre rastgele seçilen iki çözümün önce alt etme sıra sayılarına bakılır. Hangisi büyük ise, o çözüm seçilir. Eğer aynı ise, bulunan uzaklık değeri büyük olan çözüm seçilir ve böylece eşleme kümesi oluşturulur. Artık bu kümedeki çözümlere çaprazlama ve mutasyon uygulanabilir.

Sonuçta yeni Q_{t+1}popülasyonu oluşturulur. Durdurma kriteri sağlanıncaya kadar algoritmanın çalışması devam eder (Deb, 2001; Deb ve ark., 2002)

3.9.1. Rütbeleme

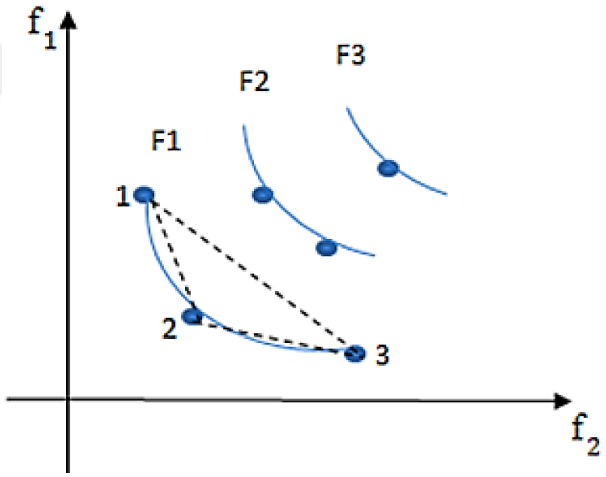
Rütbelemenin amacı popülasyon elemanlarının değerlerininbirbirleri ile kıyaslanmasıdır. NSGA II'de her elemanın kaç kere mağlup olduğu bir sayaçta tutulur. Bu duruma göre eğer diğerleri tarafından mağlup edilemeyen eleman ya da cevaplar bulunursa bunlar F_1 ya da birinci cepheyi (Front) oluşturmaktadırlar, böylece bu cephenin içindeki cevapların rütbeleri 1 olmaktadır. Sonra popülasyon elemanlarından F_1'ler kaldırılır böylelikle sonraki cephenin bulunması sırasında önceki aşamada bulunan F_1 cevaplarının bozucu etkisi, sonraki F_2 adayların üzerinden kaldırılmış olur. Böylece popülasyonun geriye kalan bireylerinden mağlup edilemeyenler F_2'yi oluşturmaktadırlar. Bu işlemlerbu şekilde sürdürülür. F_1 ve F_2 kümelerindeki bireyler popülasyondan kaldırılır ve geriye kalanların mağlup edilemeyenleri F_3eklenir. Bu şekilde devam edilerek F_4 ve F_5 yüzeyleri de ortaya çıkmış olacaktır.



Şekil 3.18. Rütbeleme örneği.

Rütbelemenin önemi yukarıdaki Şekil 3.18'de gösterilmiştir. f_1 ve f_2 uzayında rütbeleme olmadığı halde 2 ve 3 çözümleri arasında seçim yapılabilirdi ve 2 noktası 3'ü domine ettiğinden dolayı 2 seçilebilirdi. Fakat 1 ve 4 cevaplarının birbirlerine göre galip ya da mağlup olma durumları yoktur. Yani aralarında baskınlık durumu yoktur. Bu çözümler arasında seçim rütbeleme ile gerçekleşir ki 4numaralı çözüm noktasının, rütbesi 1'dir ve F_1 kümesindedir. 1 noktasının rütbesi 2'dir ve F_2 kümesindedir. Dolayısıyla 1 ve 4 arasında bir seçim yapılacağı zaman rütbesi bir olan4numaralı çözüm seçilir. Düşük rütbelerin önemi çoktur ve öncelik düşükten yükseğe doğrudur.

Bazı durumlarda rütbeleme çözümler arasındaki seçimde belirleyici olmayabilir. Şekil 3.19'a göre 1 ve 2 ve 3 noktaların arasında nasıl seçim yapılacağı incelendiğindegörüleceği üzere bu noktalar F_1 cephesinin içindedirler ve rütbeleri aynı olduğundan dolayı rütbe açısından seçim yapılamamaktadır.

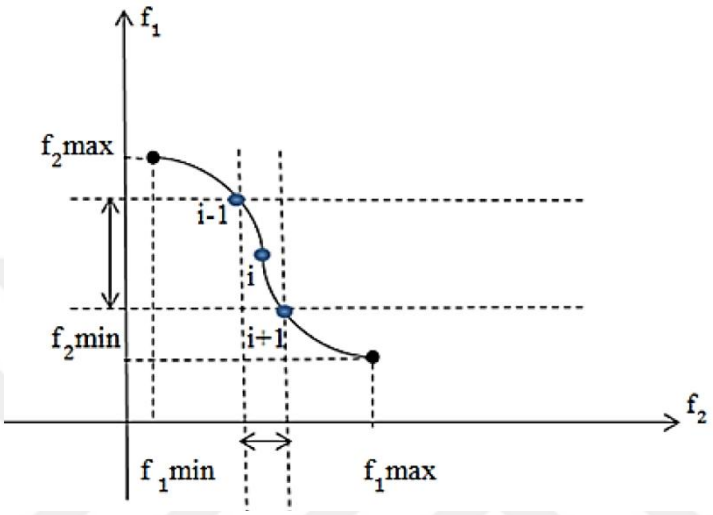


Şekil 3.19. Aynı rütbeliler arasında seçim.

Bu noktada ikinci kritere ihtiyaç duyulur. 1 ve 2, 2 ve 3 noktaların örtüştükleri alan 1 ve 3 noktalarına göre daha azdır. 1 ve 3 yüksek örtüşme alanına sahip oldukları için bu noktaların yüzey içerisinde kalmalarında bir mahsur görülmez. Geri kalan 2 noktasının örtüşmede düşük katkısı olduğundan bu çözüm elenmek zorunda kalır. Bu işlem Yoğunluk Mesafesi(Crowding Distance) Hesabı (Coello, 2007) olarak isimlendirilmektedir.

3.9.2. Yoğunluk mesafesi

Şekil 3.20'de gösterildiği gibi yoğunluk mesafesi cevabın bir önceki ve sonraki komşusuile popülasyonun ilk ve son elemanlarına göre hesaplanır.



Şekil 3.20. Yoğunluk mesafesi örneği.

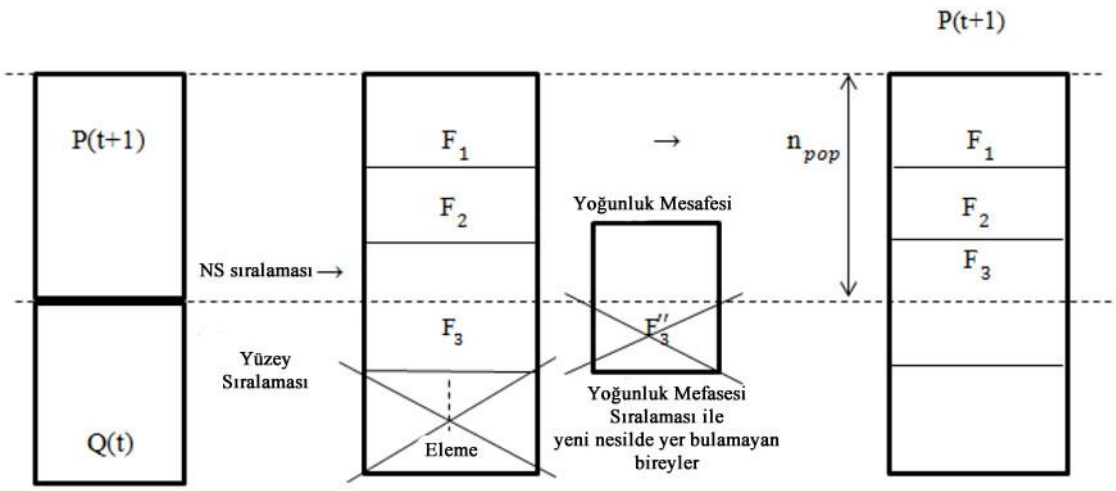
$$d_i^1 = \frac{\left|f_1^{i+1} - f_1^{i-1}\right|}{f_1^{max} - f_1^{min}}$$
$$d_i^2 = \frac{\left|f_2^{i+1} - f_2^{i-1}\right|}{f_2^{max} - f_2^{min}} \quad (3.13)$$
$$d_i = d_i^1 + d_i^2$$

Bir bireyin bir sonraki nesle aktarılması için rütbesinin yani yüzey numarasının düşük olması gerekir. Rütbeleri eşit bireyler arasında seçim yapılacaksa bu durumda yoğunluk mesafesi devreye girer. Eşitlik 3.13'e göre yoğunluk mesafeleri hesaplanan bireyler büyükten küçüğe doğru sıralanır. Listenin başında yer alan bireylerin, diğer bireylerle örtüşme miktarları daha yüksektir ve bir sonraki nesle aktarılmaları gerekir.

3.9.3. Elitizm

Seçkinciliğin veya elitizmin yani en iyi bireylerin saklanması ve bir sonraki topluluğa eklenmesinin GA'nın başarımına önemli ölçüde katkı sağladığı görülmüştür (Zitzler, 1999). Elitizm dikkatli bir biçimde uygulanmalıdır. Eğer elitizm kontrollü bir biçimde uygulanmazsa popülasyonda çeşitlilik kaybı söz konusu olabilir. Bu da aynı uygunluk değeri veren bireylerin çoğalmasına neden olur.

Elitizmi, Şekil 3.21 üzerinde incelemek gerekirse F_1 ve F_2yüzeylerindeki bireyler yeni popülasyon P_{t+1}içine sığmaktadırlar, fakat F_3 kümesinin boyutu P_{t+1}'in boyutunu aşmaktadır. F_3 yüzeyindeki bireylerin rütbeleri eşit durumda ve yeni popülasyon boyutu aşıldığı için F3'ün bir kısmı elenmek zorundadır. Bu durumda ikinci kriter olarak Yoğunluk Mesafesi (CD) devreye girer.



Şekil 3.21. Yeni popülasyon seçimi.

F_3yüzeyindeki bireylerin elemesi şu şekilde gerçekleşir: Elemanları kendi içlerinde yoğunluk mesafelerine(CD) göre kıyaslanarak büyükten küçüğe doğru sıralanır.Yoğunluk mesafesi yüksek olanlar yeni popülasyona geçerken yoğunluk mesafesi düşük olanlar elenmektedirler.

3.9.4. Seçim

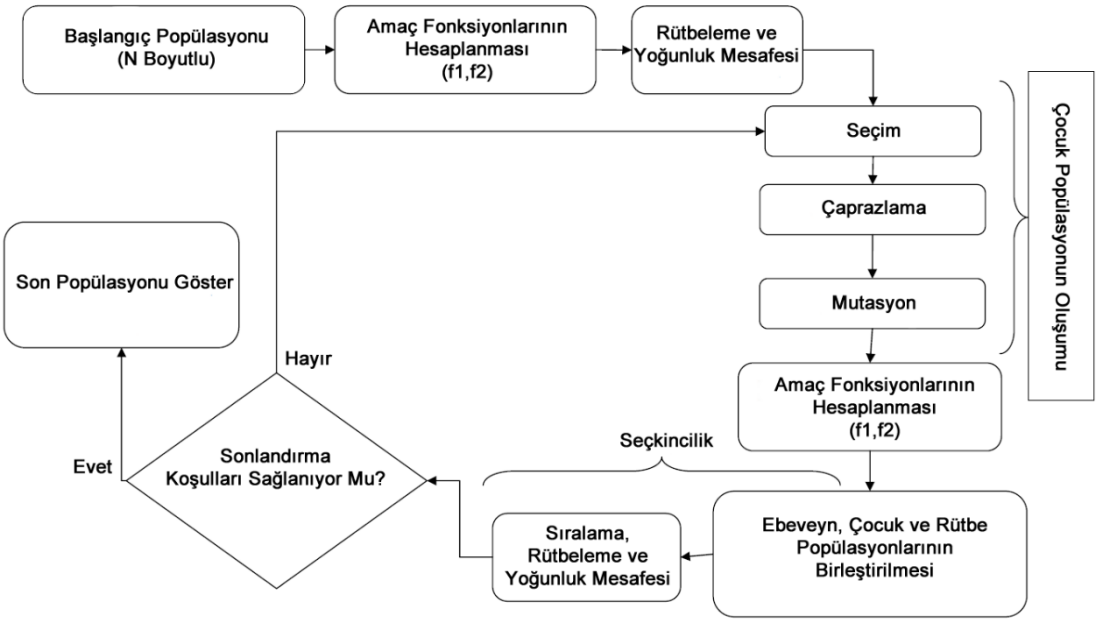
İkili turnuva seçimi (Binary Tournament Selection) klasik genetik algoritmada ebeveyn seçimlerinden biridir. Çok amaçlı genetik algortimalarda da ikili turnuva seçimi kullanılabilir. Klasik genetik algoritmalarda bireylerin uygunluk değerleri karşılaştırılır ve problemin istediği sonucaen yakın çözümü vermiş birey seçilir. NSGA II'de ise baskın mantığı ile seçim gerçekleşir. Bireylerin önce yüzeyleri karşılaştırılır, eşitlik durumunda yoğunluk mesafelerine(CD) bakılır.İkili turnuva seçimi için $x\ dom\ y$operatörü kullanılmaktadır.

$$Eğer\ F_x < F_y\ ise \rightarrow\ x\ dom\ y$$
$$Eğer\ F_x =\ \ F_y\ ise\ ve\ (CD)_x > (CD)_y\ ise\ \rightarrow\ x\ dom\ y \tag{3.14}$$

Yukarıda verilen F_i her bireyin rütbesini yani yüzey numarasını vermektedir. $(CD)_i$ise her bireyin yoğunluk mesafesi değeridir.

İkili turnuva seçimi yöntemi uygulanırken, üretilmiş yeni nesilpopülasyon bireylerindenrasgele ikisi seçilir. İlk etapta bu iki bireyin rütbeleri karşılaştırılır.Rütbesi daha düşük olan birey bir sonraki popülasyona eklenir.Eğer bireylerin rütbelerieşit değerdeyseyoğunluk mesafesi(CD) yüksek olan birey bir sonraki nesle aktarılır. Eğer P_{t+1}için popülasyon boyutu *n* ise bu durumdaİkili Turnuva seçimi *n*sefer tekrarlanacak demektir.

Şekil 3.22'de NSGA II için akış diyagramı verilmiştir. N boyutlu başlangıç popülasyonu ile amaç fonksiyonları hesaplanmıştır. Rütbeleme ve yoğunluk mesafesi hesaplamalarının ardından genetik operatörlerin uygulanmasına geçilmiştir. Genetik uygulamaların ardından amaç fonksiyonları tekrar hesaplanmış ve sonrasında ebeveyn ve çocuk bireyler birleştirilmiştir. Yoğunluk mesafesi ve rütbeleme ile tekrardan bir sıralama yapılmış, sonlandırma kriterine bakılmıştır. Koşul sağlanıyorsa algoritma sonlanır aksi durumda seçim işlemine tekrar dönülür.



Şekil 3.22. NSGA II akış diyagramı.

3.10. Test Fonksiyonları

Çok amaçlı genetik algoritmalar için önerilmiş ve önerilmeye devam eden birçok test fonksiyonu vardır. Bir yöntemin test edebildiği problem ne ölçüde zorsa, o yöntem aynı ölçüde etkindir denilebilir. Bu amaçla, çok amaçlı genetik algoritma yöntemlerinin karşılaştırma çalışmalarında veya bir çok amaçlı genetik algoritma yönteminin etkinliğinin tespit edilmesinde kullanılmak amacıyla birçok yapay test işlevi önerilmiştir. Bu test fonksiyonları karşılaştırma çalışmalarında kullanılmışlardır (Deb, 1999; Deb, 2001; Coello, 2007). Bu test fonksiyonların uygulanmasıiçin genetik algoritmaların nasıl çalıştığı konusu hakkında bilgi sahibi olmaya gerek yoktur. Bu analitik problemler, çok amaçlı genetik algoritma yöntemine uygulandığında elde edilecen çözümler, yöntemin geçerliliğine dair bilgiler sunar. Pareto-optimal cepheye mümkün olduğunca yakınsama ve bu cephe üzerinde çözümlerin düzgün dağılımıbir çok amaçlı genetik algoritmadan beklenen iki işlevdir.Çok amaçlı genetik algoritma test işlevleri, genellikle bu iki amaca ulaşılmasını zorlaştıracak biçimlerde oluşturulmuştur.

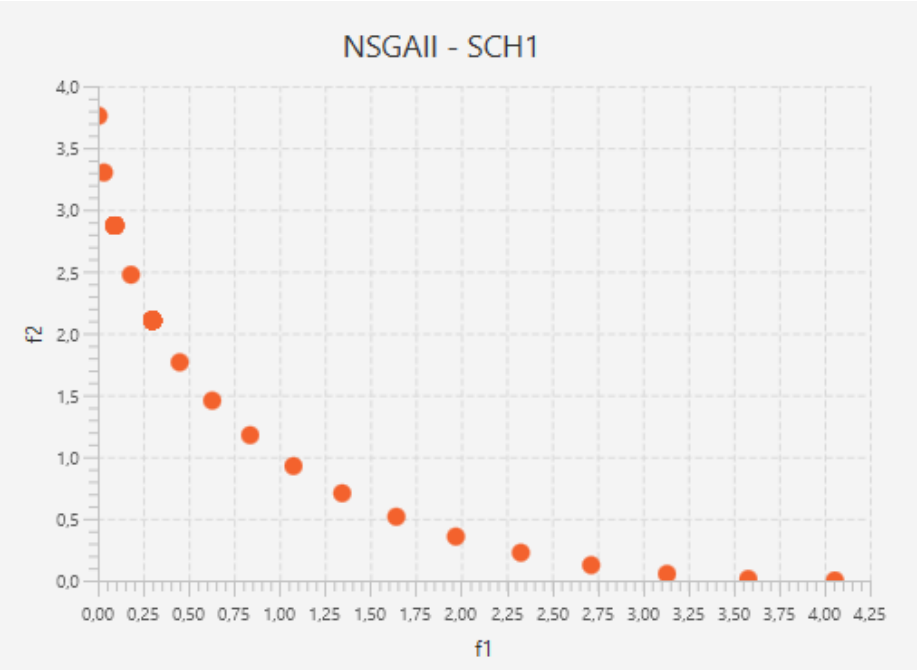
Örneğin, çok modluluk, ayrık Pareto-optimal cepheler, içbükeylik, dışbükeylik, düzgün dağılmamış Pareto-optimal cephe vb.

3.10.1.SCH1testi

Basit olmasına rağmen, en çok kullanılan tek parametreli (değişkenli) test işlevidir. Schaffer tarafından önerilmiştir [Schaffer, 1984]. Bir en küçükleme problemidir ve matematiksel ifadesi aşağıdaki eşitlikte verilmektedir:

$$f_1(x) = x^2$$
$$f_2(x) = (x-2)^2 \qquad (3.15)$$
$$-A \leq x \leq A$$

Bu işlev, $x \in [0,2]$aralığında Pareto-optimal çözümlere sahiptir ve Pareto-optimal çözüm kümesi $0 \leq f_1 \leq 4$ aralığındadır. Parametre sınırları için (*A*), farklı çalışmalarda farklı değerler kullanılmıştır. A değeri yükseldikçe, Pareto-optimal cepheye olan yakınsama zorlaşmaktadır. Bu test işlevinin A=6 değeri için grafiği Şekil 3.23' de verilmektedir.



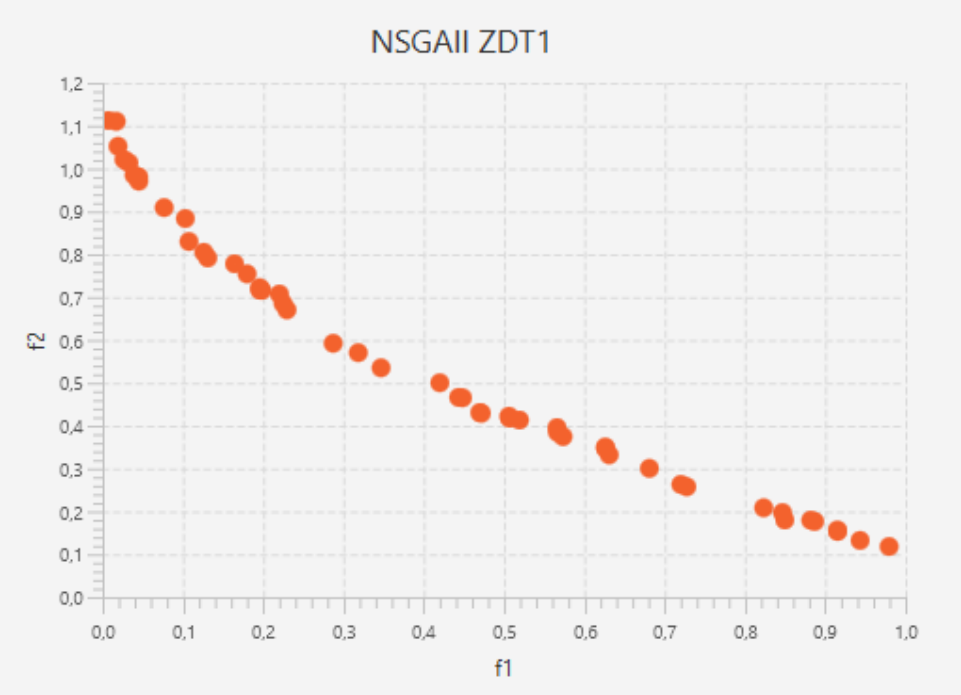
Şekil 3.23. SCH1 testi.

3.10.2. ZDT1 testi

ZDT1 test işlevi dışbükey bir Pareto-optimal cepheye sahiptir:

$$f_1(x_1) = x_1$$
$$g(x_2, \dots, x_m) = 1 + 9\sum_{i=2}^{m} \frac{x_i}{(m-1)} \qquad (3.16)$$
$$f_2(f_1, g) = 1 - \sqrt{f_1/g}$$

Burada m=30 ve $x_i \in [0,1]$ 'dir. Pareto-optimal cephe $g(\vec{x}) = 1$ alınarak oluşturulur. Bu test işlevinin Pareto-optimal cephesi Şekil 3.24'te verilmektedir.



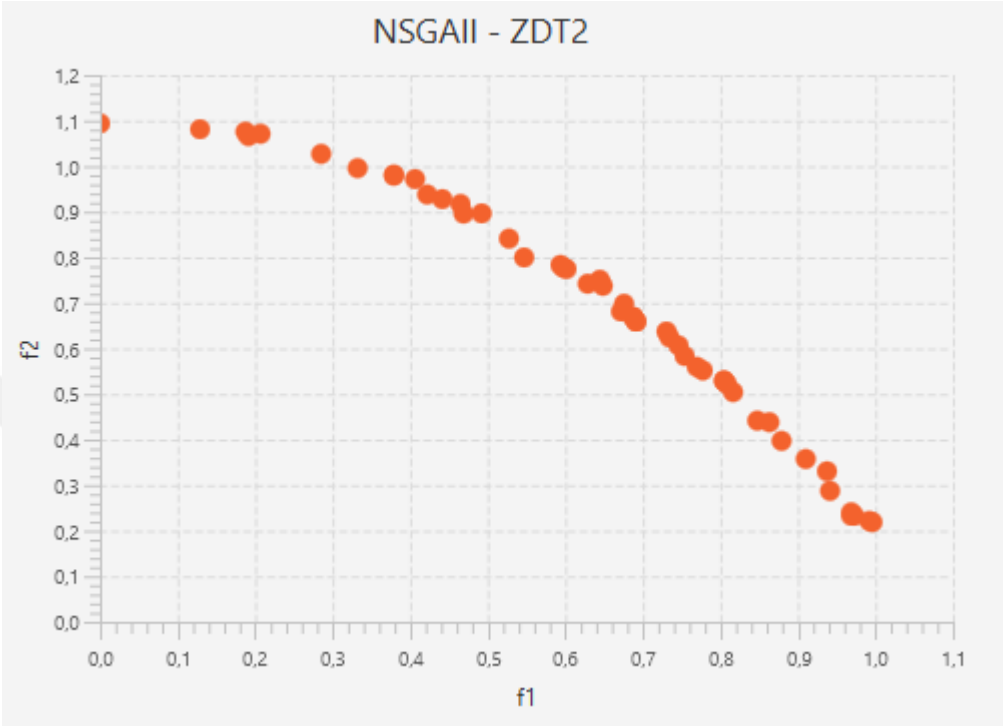
Şekil 3.24. ZDT1 testi.

3.10.3. ZDT2 testi

Bu test işlevi, ZDT1 test işlevinin içbükey bir biçimidir:

$$f_1(x_1) = x_1$$
$$g(x_2, \dots, x_m) = 1 + 9\sum_{i=2}^{m}\frac{x_i}{(m-1)} \quad (3.17)$$
$$f_2(f_1, g) = 1 - \left(f_1/g\right)^2$$

Burada m=30 ve $x_i \in [0,1]$ 'dir. Pareto-optimal cephe $g(\vec{x}) = 1$ alınarak oluşturulur. Bu test işlevinin Pareto-optimal cephesi Şekil 3.25' te verilmektedir.



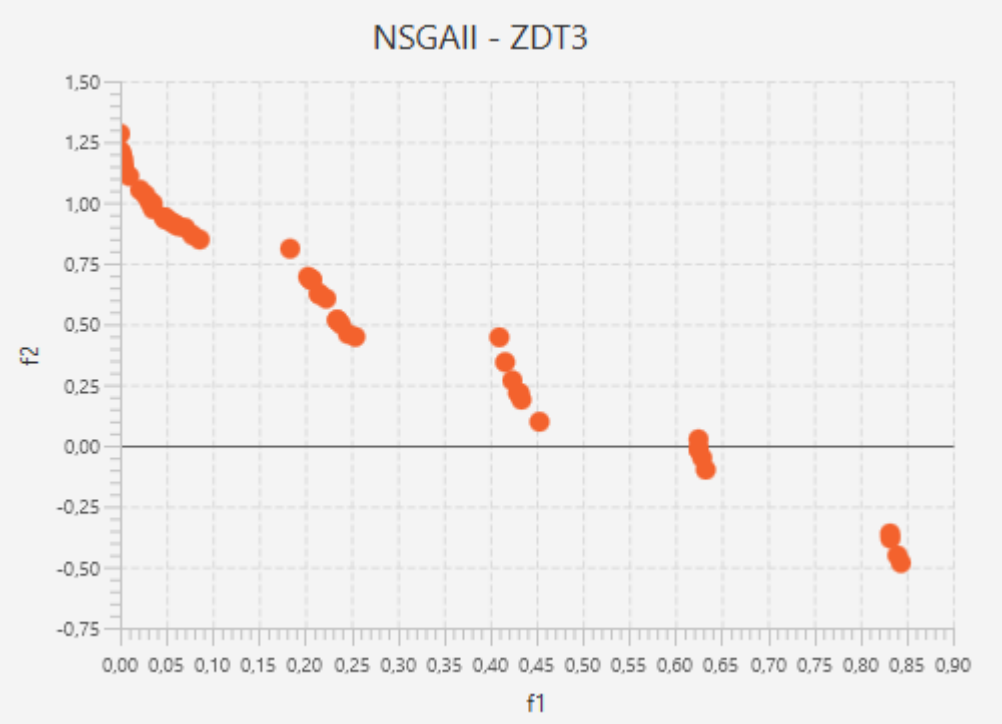
Şekil 3.25. ZDT2 testi.

3.10.4. ZDT3 testi

Bu test işlevi ayrık bir işlev özelliği gösterir, Pareto-optimal cephesi birkaç adet sürekli olmayan (ayrık) dışbükey parçadan oluşur:

$$f_1(x_1) = x_1$$
$$g(x_2, \dots, x_m) = 1 + 9\sum_{i=2}^{m} \frac{x_i}{(m-1)} \tag{3.18}$$
$$f_2(f_1, g) = 1 - \sqrt{{}^{f_1}/_{g}} - ({}^{f_1}/_{g}) \sin(10\pi f_1)$$

Burada m=30 ve $x_i \in [0,1]$ 'dir. Pareto-optimal cephe $g(\vec{x}) = 1$ alınarak oluşturulur. $f_2$ifadesindeki sinüs işlevi Pareto-optimal cephede ayrıklığa neden olmaktadır. Bununla birlikte, parametre uzayında hiçbir ayrıklık yoktur. Bu test işlevinin grafiği Şekil 3.26'da verilmektedir.



Şekil 3.26. ZDT3 testi.

4. UYGULAMA VE BULGULAR

4.1. Kullanılan Veri Seti

Bu çalışmada metin madenciliğinde iyi bilinen bir kıyas veri koleksiyonu olan Classic veri koleksiyonu kullanılmıştır. Veri seti 4 faklı doküman koleksiyonundan oluşur. CACM, CISI, CRAN ve MED.Koleksiyonun kompozisyonu şöyledir: CACM: 3204 belge, CISI: 1460 belge, CRAN: 1398 belge ve MED: 1033 belge. Bu veri kümesinin yalnızca 3 sınıftan oluşan (CISI, CRAN ve MED) versiyonu Classic3 olarak adlandırılır. Bu çalışmada Classic3 versiyonu kullanılmıştır. Bu versiyonda bu kategorilere ait belge sayıları ise sırasıyla 1589, 1398 ve 1033'tür. Bu belge koleksiyona veri madenciliği adımlarının uygulanmasının ardından rasgele seçilmiş ve her sınıftan 50 belge olacak şekilde 150 boyutlu 5 veri seti oluşturulmuştur.Optimizasyonlar bu veri setleri üzerinde uygulanmıştır.

Kullanılan veri kümeleri öncelikle ön işlemeye tabi tutulmuştur. İlk olarak 350 kelimeden oluşan stop-words kelimeleri bu belgelerden çıkarılmıştır. Kelimelerin gövdeleme işlemi için, metin madenciliğinde yaygın bir şekilde kullanılan Porter Stemmer algoritması seçilmiştir. Bunun sonucunda belgeler içerdikleri gövde kelimelere yani terimlere göre modellenmiştir.

Modellenmiş veri seti, kelime frekansları üzerinden yeniden düzenlenmiştir. Buna göre her makaleye ve her kelimeye birer *Kelime No* atanmış ve her makalede hangi kelimeden kaç adet yer aldığı belirlenmiştir. Aşağıda Çizelge 4.1'de örnek gösterimi verilmiştir.

Çizelge 4.1.Kelime Frekans Tablosu

BelgeNo	***KelimeNo***	***Kelimenin Sayısı***
1	346	1
1	2985	3
2	319	1
2	2408	1
3	8215	10
3	2259	3

Her bir belgenin kaç adet kelime barındırdığı ve hangi kategoriye ait olduğunun bilgisinin yer aldığı bir doküman daha oluşturulmuştur. Bunun örneği ise Çizelge 4.2'de sunulmuştur.

Çizelge 4.2. Kelime Frekans Tablosu

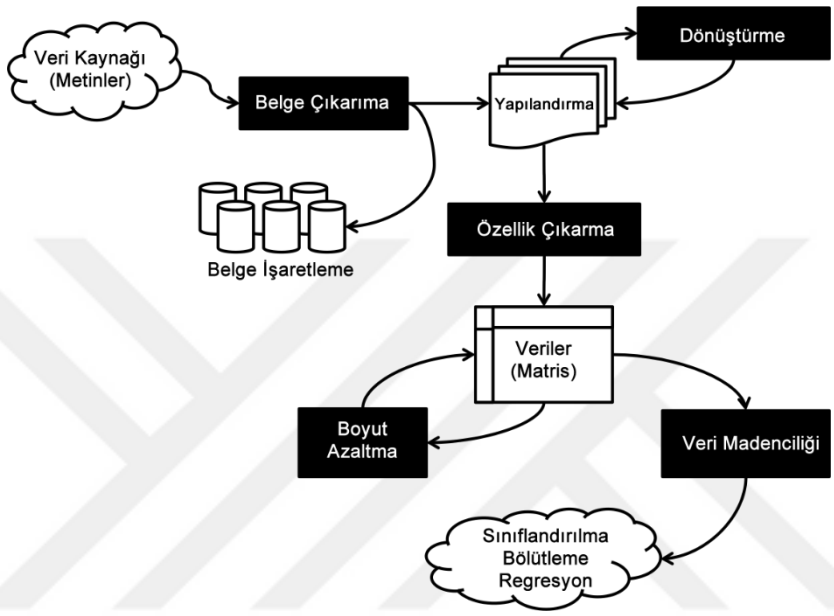
Belge No	***KelimeSayısı***	***Kategori***
1	19	1
2	14	1
3	100	1
1588	75	2
1589	116	2
1590	17	2
2988	55	3
2989	115	3
2990	72	3

Bu işlemlerden sonra tüm veri seti için Makale-Kelime matrisi çıkarılmıştır. Bu matris çoğu hücresi sıfır olan ve her bir makaleye karşılık hangi kelimeden kaç adet yer aldığının bilgisini tutan bir yapıya sahiptir. Sonuçta elde ettiğimiz matris, makale sayısının yer aldığı 4020 satırdan ve kelime sayısının yer aldığı 11398 sütundan oluşmaktadır. Bu matris çoğu sıfırlardan oluşan bir matristir. Bu halde bu matrisin veri madenciliğinde kullanılması mümkün görünmemektedir. Bu noktada boyut azaltma işlemlerini kullanmak gerekir. Bu çalışmada bu işlem için PCA kullanılmıştır.

4.2.Veri Madenciliği Aşamaları

Veri madenciliği uygulaması bir süreci gerektirmektedir. Madencilik kavramı veri yığınları arasında soyut kazıların yapılmasıyla bilginin ortaya çıkarılmasını temsil eder.Bilgi keşfi süreciörüntülerin ayrıştırılarak süzülmesi ve bir sonraki adıma hazır hale getirilmesidir. Veri madenciliği yöntemlerinin uygulanabilmesi için veri ambarlarında veya veri tabanlarında tutulan verilerin belirli aşamalardan geçmesi şarttır.

Bu çalışmada veri madenciliğinin tüm aşamalarının gerçekleştirilmesine özen gösterilmiştir. Başarılı bir veri madenciliği çalışmasında uygulanması gereken aşamalar Şekil 4.1'de verilmiştir.



Şekil 4.1. Veri madenciliği aşamaları.

4.2.1.Veri seçimi

Veri seçimi, veri madenciliği aşamalarında en fazla zaman harcanan kısımlardandır. Bu aşamada bilgi sistemlerinde oluşan bilgiyi iyi analiz etmek gerekir.Bu analiz sonucunda bilgi-problem ilişkilendirmesi yapılır. Veri kalitesinin ölçülmesi bu aşamada yapılır ve çok büyük önem arz eder. Büyük miktardaki veriler tek bir veri tabanı veya veri ambarında birleştirilir. Bu veri madenciliğinde sonucun bütünlüğü bakımından önemlidir. Veri seçimi aşaması veri filtreleme olarak da isimlendirilebilir.

Bu çalışmada amaç sınıflandırma başarısının artırımıdır. Bu amaç doğrultusunda benzerlik ve sınıflandırma optimizasyonu yapılmıştır. Üzerinde çalışılacak veri seti seçilirken bu durumlar göz önüne alınmıştır.

4.2.2. Ön işleme

Ön işleme aşaması veri madenciliğinin başarısını büyük ölçüde etkiler. Ön işleme aşamasında veri, sonraki aşamalarda kullanılabilmesi için daha elverişli hale getirilir. Ön işleme aşamasındaki her bir adım sonuçtaki başarıyı doğrudan etkiler. Ön işleme aşamasındaki başarıyla kesin ve net sonuçlara ulaşmak mümkündür.

Bu çalışmada kullanılan 4020 makale kelime frekanslarına göre yeniden modellenmiştir. Elde edilen 11398 faklı kelimenin her birinin, her makalede kaç adet geçtiğini gösteren bir matris oluşturulmuştur.

4.2.3. İndirgeme ve PCA uygulaması

Veri üzerinden faydalı ve doğru sonucudaha ekonomik ve daha doğru elde etmek için kullanılacak verinin indirgenmesi gerekir. Eldeki verinin büyük bir kısmı istenilen problemi çözmede gereksiz olabilir. Veri her ne kadar ön işleme aşamasından geçmiş olsa bile sonraki aşamalarda kullanılabilecek durumda olmayabilir. Dolayısıyla verinin kullanılabilecek duruma indirgenmesi gerekir.

Veri önişleme aşamasında elde edilen makale-kelime matrisinde hangi kelimeden hangi makalede kaç adet bulunduğunun bilgisi yer almaktadır. Bu matrisin büyük kısmı sıfırlardan oluşmaktadır. Toplamda 11398 sütun veri düşünüldüğünde bu veri seti hem optimizasyon için hem de veri madenciliği için uygun değildir. NSGA2 ile optimizasyon işlemi hem zaman hem de bellek maliyeti bakımından külfetli bir iştir. Hal böyleyken veri indirgeme adımı bu çalışmanın en önemli aşamalarından biri haline gelmiştir.

Temel bileşen analizi bağımlılık yapısını ortadan kaldırma ve boyut indirgeme amaçları için kullanılmaktadır. Tanıma, sınıflandırma, boyut indirgenmesi ve boyut yorumlanmasını sağlayan, çok değişkenli veri yapılarında kullanılan bir istatistik yöntemidir. Bu yaklaşım verinin içindeki en güçlü örüntüyü ortaya çıkarmaya çalışır.

Çoğunlukla verinin sahip olduğu çeşitlilik bozulmadan, tüm boyut takımından seçilen küçük bir boyut setle tüm veri temsil edilebilir.

Bu çalışmada veri indirgeme işlemi için PCA kullanılmıştır. PCA uygulaması için ise MATLAB programından yararlanılmıştır.Makale-Kelime matrisi PCA'ya sunulmuştur. PCA sonrası elde ettiğimiz yeni matris ise bize ana bileşen puanını vermektedir. Elde ettiğimiz yeni matrisin temel bileşen alanındaki temsili, bize üzerinde veri madenciliğini uygulayacağımız yeni bir set verecektir. Bu yeni veri setinden seçilecek sütun miktarı çok önemlidir. Bu seçim yapılırken dikkat edilecek kriter, uygulanacak veri madenciliği aşamasının başarılı olmasıdır. Bu çalışma kapsamında kodladığımızbasit KNN uygulamasındaki denemelerde 10 sütunluk bir seçimle elde ettiğimiz veri setinin farklı Kfold değerlerine göre sınıflandırma başarısı Çizelge 4.3'te verilmiştir. KNN algoritmasındaki k değeri 5 olarak alınmıştır.

Çizelge 4.3.Tüm veri setinin farklı K-Fold değerine göre KNN sınıflandırma başarıları (%)

Veri Seti	***k=2***	***k=3***	***k=5***	***k=7***	***k=9***
Doğru sınıflandırma	93.8	94.4	94.4	94.7	94.3

4.2.4 Veri madenciliği

Veri madenciliğinin tam olarak uygulanması bu aşamada gerçekleşir. Veri bu aşamada son halindedir. Çalışmanın amacına göre veri madenciliği yöntemlerinden bir ya da birkaçı seçilerek, uygun ve kullanılabilir veri üzerinde uygulama gerçekleştirilir.Problemin durumuna göre yöntemler sadece yalın hal yerine birleştirilerek de kullanılabilir.

Çalışmanın bu aşamasında modellenmiş ve PCA ile indirgenmiş veri setindeki tüm değerler ait oldukları kategorilere göre ayrı listelere kaydedilmiştir. Nesne tabanlı programlamada kullanılan koleksiyonlar yardımıyla listeler karıştırılmıştır. Her listeden rasgele seçilen 50 veri, 5 ayrı veri setine saklanmıştır. Bu veri setleri üzerinde K En Yakın Komşu algoritmasıyla bir sınıflandırma başarısı ölçümü gerçekleştirilmiştir.

Benzerlik hesaplamaları için Öklid Mesafesi kullanılmış ve sınıflandırma sonrası farklı çapraz doğrulama(*k*) değerlerine göre elde edilen bu sonuçlar Çizelge 4.4'te verilmiştir.

Çizelge 4.4.Rasgele belirlenmiş veri setlerinin farklı K-Fold değerine göre KNN sınıflandırma başarıları (%)

Veri Setleri	***k=2***	***k=3***	***k=5***	***k=7***	***k=9***
SET 1	96.6	91.3	96.0	94.0	86.0
SET 2	89.3	85.3	91.3	89.3	82.0
SET 3	94.6	91.3	94.0	94.0	85.0
SET 4	92.6	87.3	92.6	90.6	84.0
SET 5	88.0	81.3	88.6	86.6	80.6

4.2.5 Yorumlama ve doğrulama

Veri üzerinde uygulanan veri madenciliği aşamasından sonra alınan sonuçlar yorumlanır ve çalışmanın nasıl bir sonuca ulaştığı araştırılır. Eğer farklı yöntemler uygulanmışsa onların karşılaştırması yapılır. En önemlisi de elde edilen sonuçlar yapılmış olan diğer çalışmaların sonuçlarıyla karşılaştırılıp doğrulanır.

Veri madenciliği adımıyla elde edilen sonuçlar incelendiğinde Çizelge 4.4'e göre elde edilen sınıflandırma başarıları %81 ile %97 arasında değişim göstermiştir. Bu değişimin nedeni her veri setinin farklı karakteristik özelliği ve faklı çapraz doğrulama parametreleridir. Tablo incelendiğimde her veri setin için en yüksek başarı k=5 ve k=2 için gerçekleşmiştir. Bu değerler bize daha sonra gerçekleşecek optimizasyon adımları için yol gösterici olmuştur. Bu çalışma da iki tür optimizasyon işlemi gerçekleştirilmiştir. Bunlardan biri benzerlik optimizasyonu diğeri ise sınıflandırma optimizasyonu. Sınıflandırma optimizasyonu için belirlediğimiz k değeri bu tabloya göre belirlenmiştir.

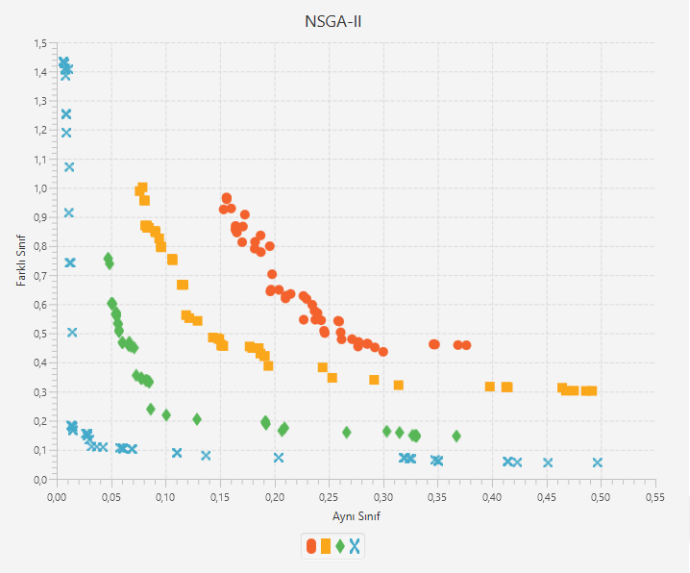
4.3 Benzerlik Optimizasyonu

Benzerlik iki nesnenin birbirlerine ne kadar benzediğini gösteren sayısal bir değerdir. Bu değer ne kadar büyükse ele alınan nesneler birbirlerine aynı ölçüde

benziyor denilebilir. Uzaklık ise benzerliğin tam tersidir ve iki nesnenin birbirlerinden ne ölçüde farklı olduklarının sayısal göstergesidir. Veri madenciliğinde sınıflama ve kümeleme işlemleri bu değerler üzerinden gerçekleştirilir. Veri nesneleri arasındaki uzaklıkları hesaplamak için çok farklı benzerlik ve uzaklık ölçüm yöntemi mevcuttur. Biz bu çalışmada Öklid uzaklık ve Kosinüs benzerlik yöntemlerini kullandık.

Sınıflandırmada kullanılan test ve eğitim verilerinin karakteristik yapıları sınıflandırma başarısını doğrudan etkilemektedir. Test ve eğitim verilenin seçiminde birden çok yöntem mevcuttur. Tüm veri eğitim seti olarak kullanılabilir. Çapraz doğrulama ya da veri setinin belirli bir yüzde kısmı eğitime ayrılabilir. Aynı özelliklere sahip dışardan bir veri seti eğitim verisi olarak kullanabilir. Bu çalışmada benzerlik optimizasyonu uygulanmış bir veri setinin, optimizasyon öncesi ve sonrası sınıflandırma başarısı incelenmiştir.

Aynı sınıfa ait nesnelerin benzerlikleri yüksek, farklı sınıftaki nesnelerin ise düşüktür. Bu durum incelendiğinde aynı sınıftaki nesnelerin benzerliklerini arttırmak ve aynı zamanda farklı sınıftaki nesnelerin benzerliklerini düşürmekçelişen iki durumdur. Çok amaçlı optimizasyon bu durumda devreye girer. Çok amaçlı optimizasyon için amaç fonksiyonlarımızdan birisi Öklid uzaklık mesafesi ile aynı sınıftaki nesnelerin uzaklıklarını düşürmektir. İkinci amaç fonksiyonumuz ise farklı sınıftaki nesnelerin benzerliklerini düşürmektir. Minimum-Minimum çok amaçlı optimizasyon problemine göre Şekil 4.2’de farklı iterasyonlardaki Pareto optimal düzeyler verilmiştir.



Şekil 4.2.Pareto optimal yakınsama.

Şekil4.2'de sırayla kırmızı, turuncu, yeşil ve mavi renkle gösterilen yüzeylerde, sırayla artan iterasyon sayılarına göre uygulama sonundaki Pareto optimal cepheler verilmiştir. Pareto optimal cephelerin oluşumu NSGA2 de kullanılan parametrelere bağlıdır. NSGA2 temelde GA operatörlerini kullanır. Çaprazlama ve mutasyon oranları, popülasyon ve iterasyon sayıları bu cephelerin bulunmasında etkilidir. İterasyon sayısının seçimi önemlidir ve doğru seçilmeyen sonlandırma kriteri bazen istenmeyen sonuçlarındoğmasına neden olabilmektedir.

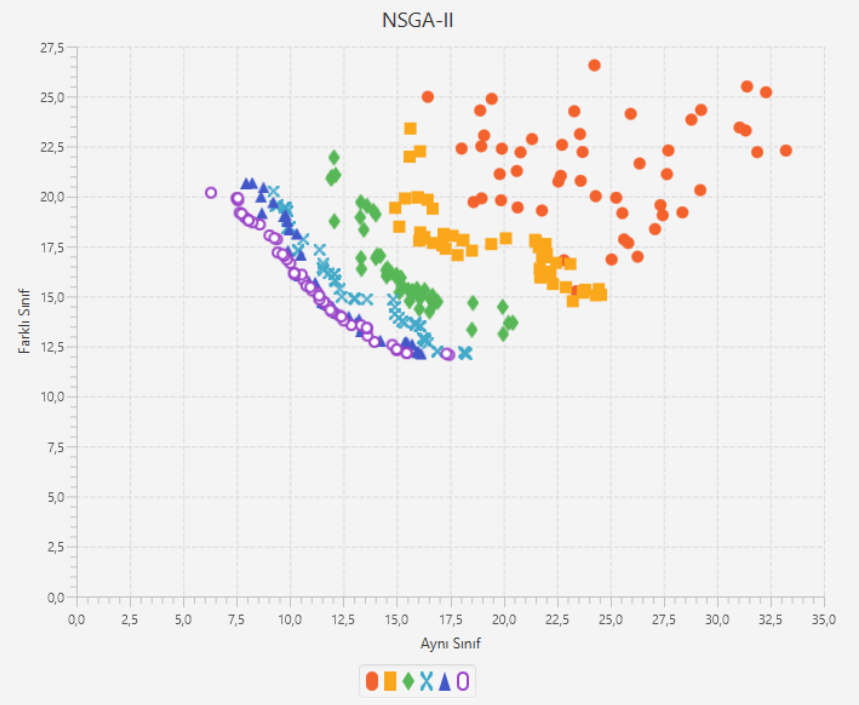
4.3.1.Rasgele seçilen veri setlerindebenzerlik optimizasyonu

Bu çalışmada her sınıftan rasgele seçilen 50 nesne ile 150 nesneden oluşan toplamda 5 veri seti oluşturulmuştur. Bu 5 veri setine ayrı ayrı benzerlik optimizasyonu uygulanmıştır. Bu optimizasyon uygulaması için genetik operatörler belirlenirken çaprazlama oranı 0.5, mutasyon oranı 0.05 ve popülasyon sayısı 50 olarak seçilmiştir. Aşağıdaki şekillerde her veri seti için ayrı ayrı pareto optimal cepheler verilmiştir. Bu cephelere ulaşmak için her set için durdurma kriteri olarak iterasyon sayısı 100 olarak

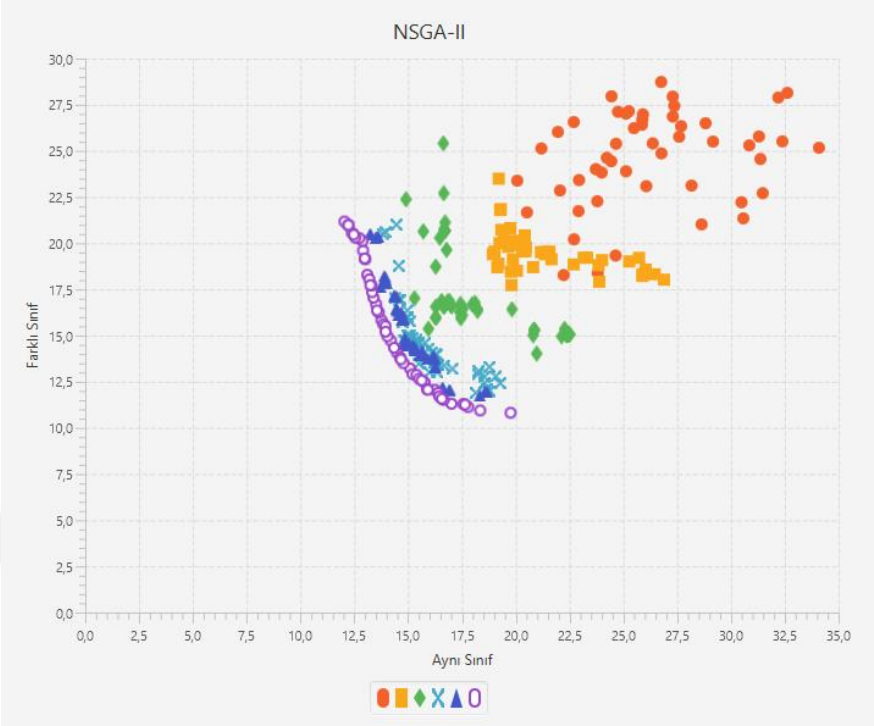
seçilmiştir. Cephelere yakınsamanın görülebilmesi için optimizasyon işlemi 6 sefer çalıştırılmıştır. Her uygulamada iterasyon sayıları sırasıyla 5, 10, 20, 40, 80 ve 100 seçilerek optimizasyon gerçekleştirilmiştir. Şekil 4.3'te Set1 için, Şekil 4.4'te Set2 için, Şekil 4.5'te Set3 için, Şekil 4.6'da Set4 için ve Şekil 4.7'de Set5 için, Pareto optimal cephelerin oluşumları verilmiştir.

Her şekil farklı iterasyon sayılarını göstermektedir. Her iterasyında popülasyon sayıları eşittir. Grafiklerde farklı iterasyonların çözüm sayılarının farklı görünmesi normaldir, bunun sebebi bazı çözümlerin çakışmasıdır.

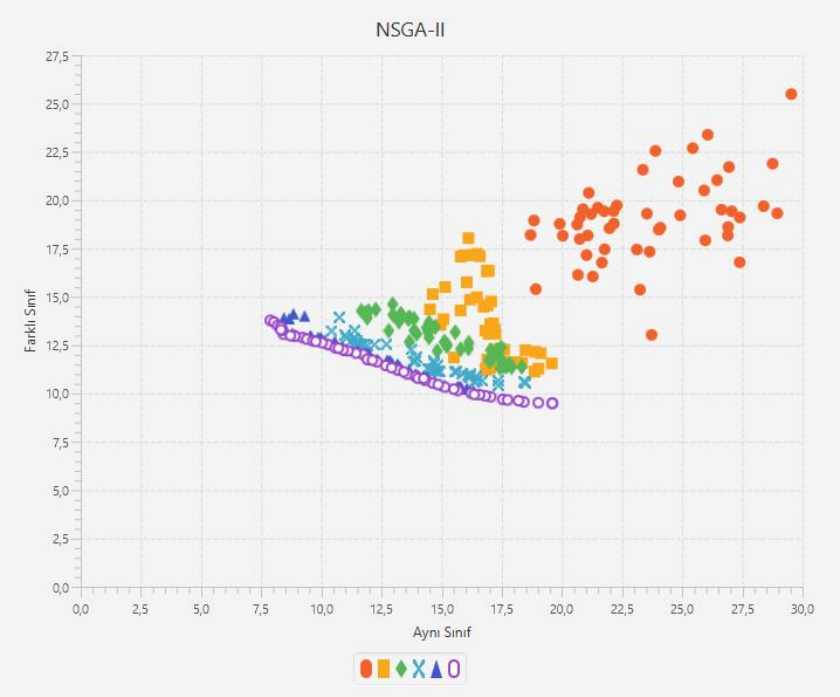
- şekli 5 iterasyonu
- şekli 10 iterasyonu,
- şekli 20 iterasyonu,
- şekli 40 iterasyonu,
- şekli 80 iterasyonu,
- şekli 100 iterasyonu temsil etmektedir.



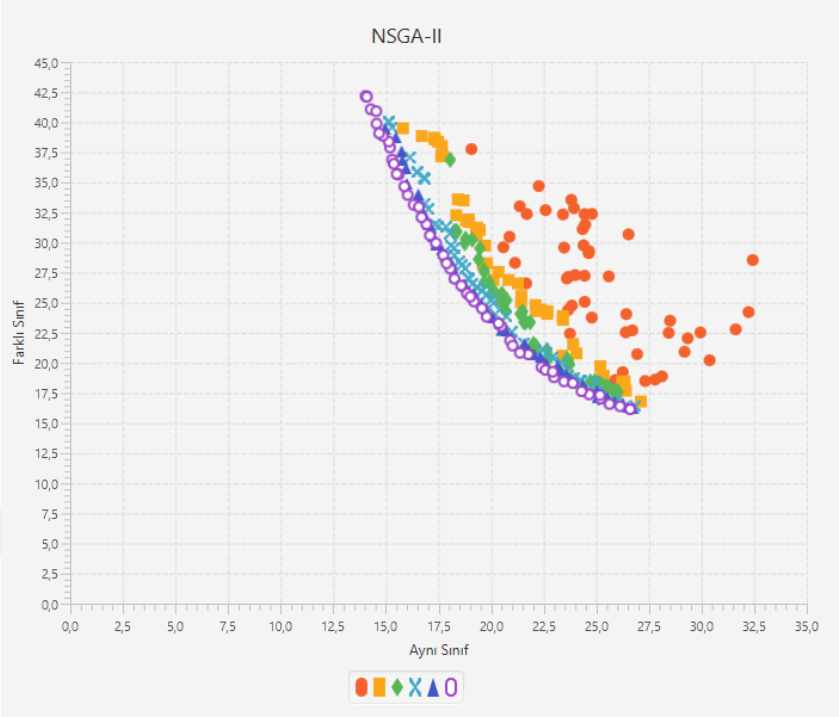
Şekil 4.3. Set1 için benzerlik optimizasyonu.



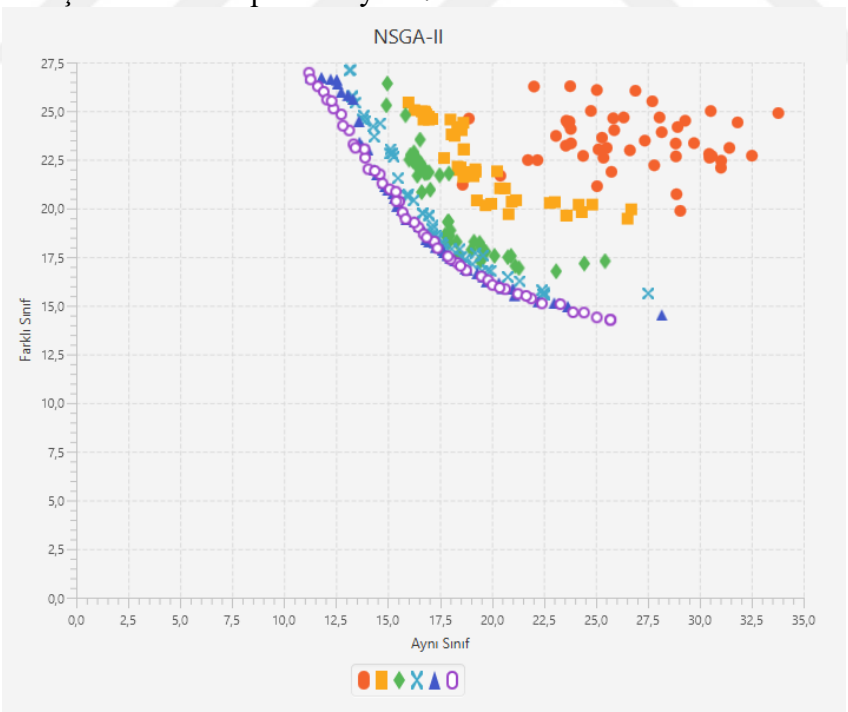
Şekil 4.4. Set2 için benzerlik optimizasyonu.



Şekil 4.5. Set3 için benzerlik optimizasyonu.



Şekil 4.6. Set4 için benzerlik optimizasyonu.



Şekil 4.7. Set5 için benzerlik optimizasyonu.

Yukarıdaki şekiller incelendiğinde son üç pareto cephesinin birbirlerine çok yakın olduğu görünmektedir. Bu durum optimizasyonun son halinin belirlenmesinde etkili olmuştur.

Maksimum iterasyon sayısına ulaşıldığında son pareto cephesinden yoğunluk mesafesi en yüksek olan birey seçilir. Bu bireye ait kromozomlar gerçek değerlerine çevrilir. Bu değerler 0 ile 1 arasında virgülden sonra 4 hassasiyetli katsayı değerleridir. Veri setindeki her bir sütun ait olduğu katsayı ile çarpılarak nesnelerin yeni değerleri bulunur.

Her bir veri seti diğer veri setleri için eğitim seti olarak kullanılmış ve sınıflandırma başarılarının ortalama değerleri bulunmuştur. Bu değerler optimizasyon sonrası sınıflandırma başarıları ile karşılaştırılmış ve sonuçlar Çizelge 4.5'te verilmiştir.

Çizelge 4.5. Benzerlik optimizasyonu öncesi ve sonrası ortalama sınıflandırma başarıları (%)

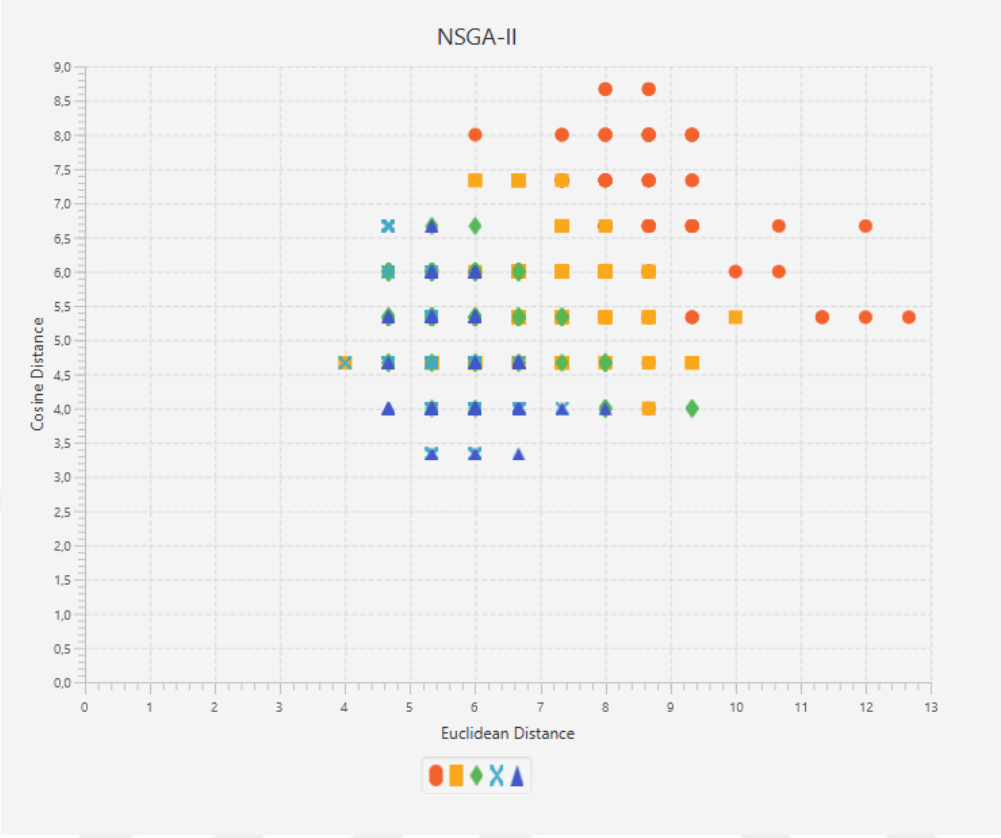
Veri Setleri	***Optimizasyon Öncesi***	***Optimizasyon Sonrası***
SET 1	91.38	91.9
SET 2	91.32	93.54
SET 3	88.8	91.1
SET 4	93.4	93.98
SET 5	90.86	92.22

4.4. Sınıflandırma Optimizasyonu

Sınıflandrıma bir veri kümesi(data-set) üzerinde tanımlı kategorilere veri kümesi içerisindeki nesneleri dağıtmak olarak tanımlanabilir. Birçok sınıflandırma algoritması vardır ve bu algoritmaların temel işlevi, belirlenen eğitim kümesinden bir dağıtım şekli öğrenmek ve daha sonra sınıfının belirli olmadığı test verilerinin ait oldukları sınıfı doğru şekilde bulmaktır. Farklı kategorilere ayrılmış her veri kümesinde sınıfların belirtildiği etiket (label) isimli alanlar vardır. Hem eğitim hem de test işlemlerinde verinin sınıfının belirlenmesi etiketler kullanılarak yapılır.

Çalışmanın bu aşamasında K en yakın komşu algoritması kullanılarak sınıflandırma optimizasyonu yapılmıştır. Bu adım gerçekleştirilirken veri setindeki her birey için bütün sınıf eğitim kümesi olarak belirlenmiştir. Her eğitim kümesi önce ilk haliyle KNN algoritmasına tabi tutulmuş, sınıflandırma başarısının en düşük çıktığı k parametresi belirlenmiş ve optimizasyon işlemi bu parametreye göre gerçekleştirilmiştir. Optimizasyonun her bir iterasyonunda bütün veri seti için sınıflandırma işlemi yapılmıştır. Amaç fonksiyonlarının ilkinde Öklid uzaklık mesafesi, diğerinde ise Kosinüs benzerlik yöntemi kullanılmıştır. Şekil 4.8'de farklı sonlandırma kriterlerine göre uygulanmış sınıflandırma optimizasyonlarının sonucu görülmektedir.

İki amaçta da hatalı sınıflandırma sonuçlarının sıfıra indirilmesi hedeflenmektedir. Şekil 4.8'e göre program toplamda 5 sefer çalıştırılmış ve son 3 seferde en az iki sonuç aynı değerleri vermiştir. Bu durum bize programın kaçıncı iterasyonda sonlandırılması gerektiği hakkında bilgi sunmuştur.

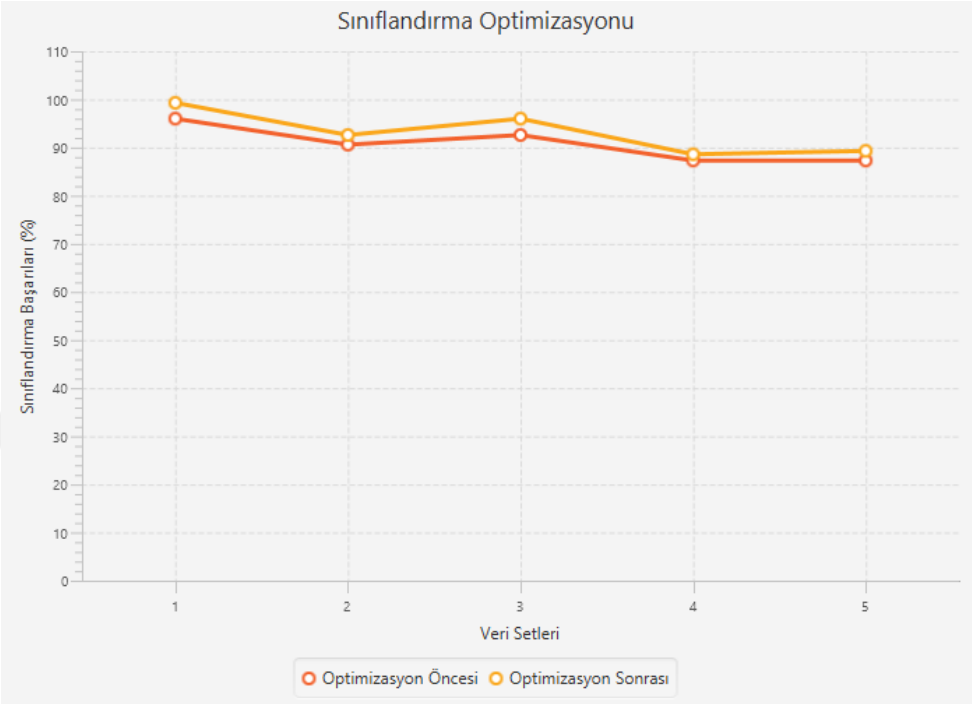


Şekil 4.8. Artan maksimum iterasyonlarda sınıflandırma optimizasyonu.

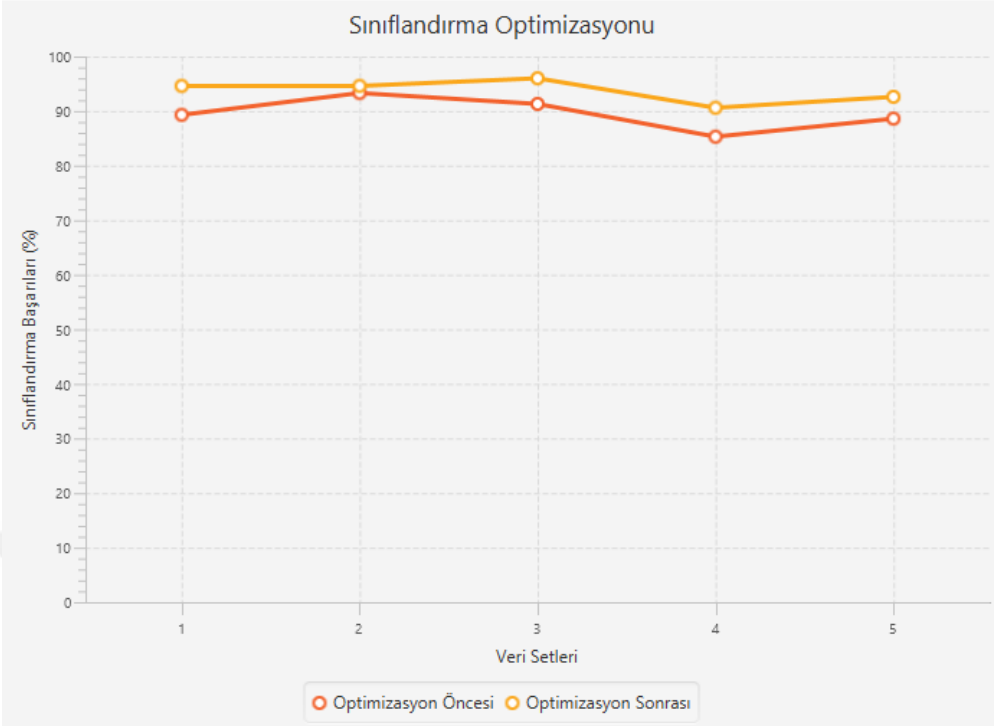
4.4.1 Rasgele seçilen veri setlerinde sınıflandırma optimizasyonu

Rasgele seçilen veri setleri üzerinde ayrı ayrı sınıflandırma optimizasyonu uygulanmıştır.Veri setlerindeki bütün nesneler her bir nesne için eğitim verisi olmuş ve KNN algoritmasıyla her jenerasyonda sınıflandırma başarısı ayrı ayrı ölçülmüştür. Bir sonraki jenerasyon için yanlış sınıflandırma sayısı sıfıra indirilmeye çalışılmıştır. Genetik operatörler; çaprazlama oranı 0.5, mutasyon oranı 0.05, popülasyon sayısı 50 ve sonlandırma kriteri olarak maksimum jenerasyon sayısı da 50 olarak belirlenmiştir. Her veri seti hem kendi içlerinde hem de diğer veri setleri için eğitim seti olarak kullanılmıştır. Optimizasyon öncesi ve sonrası sınıflandırma başarılarındaki değişim

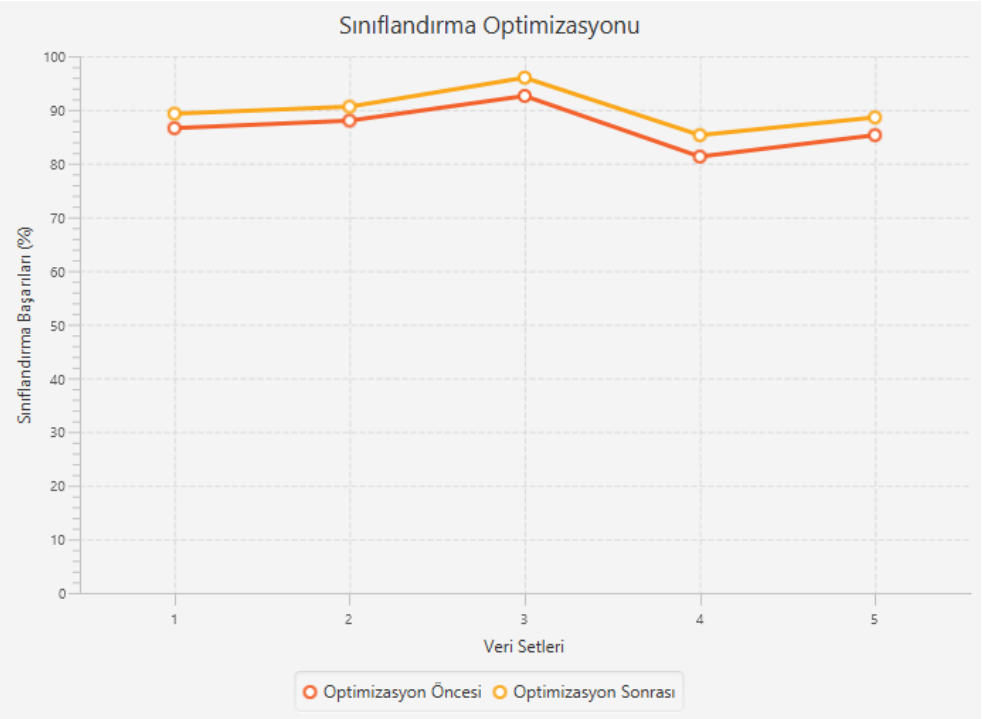
Şekil 4.9'da Set1 için, Şekil 4.10'da Set2 için, Şekil 4.11'de Set3 için, Şekil 4.12'de Set13 için ve Şekil 4.13'te Set5 için ayrı ayrı verilmiştir.



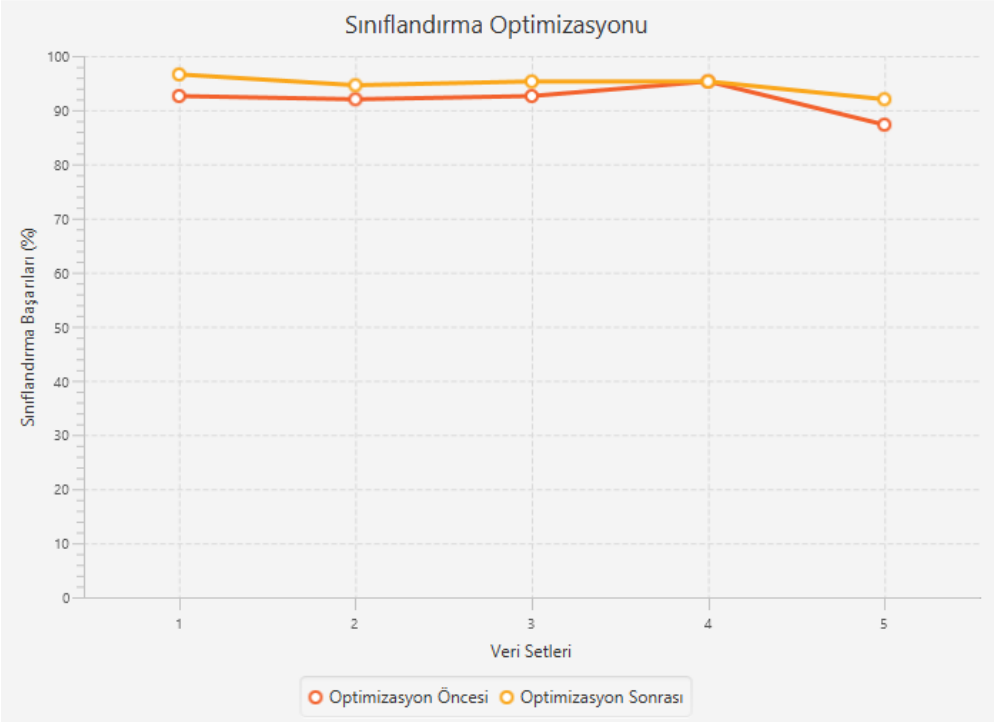
Şekil 4.9. SET1 için sınıflandırma optimizasyonu.



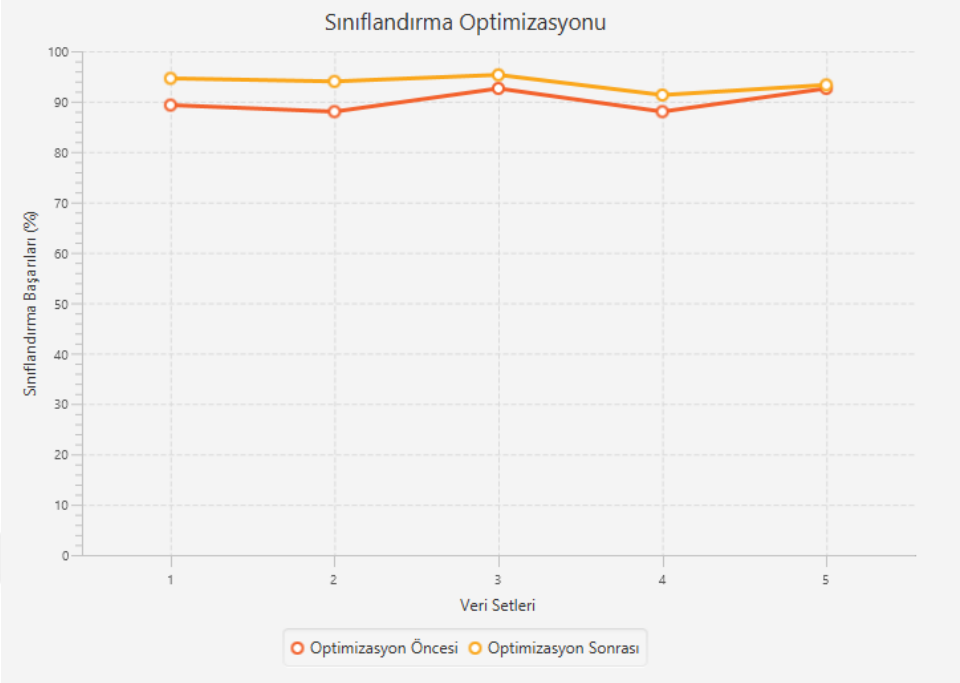
Şekil 4.10. SET2 için sınıflandırma optimizasyonu.



Şekil 4.11. SET3 için sınıflandırma optimizasyonu.



Şekil 4.12. SET4 için sınıflandırma optimizasyonu.



Şekil 4.13. SET5 için sınıflandırma optimizasyonu.

Sınıflandırma optimizasyonu sonrası elde edilen pareto cephedeki en yüksek yoğunluk mesafesine sahip birey seçilmiştir. Seçilen bireye ait kromozom gerçek değerlere dönüştürülmüştür. Bunun sonucunda virgülden sonra dört hassasiyete sahip 10 katsayı elde edilmiştir. Veri setlerindeki her sütun ilgili katsayı ile çarpılmış ve yeni değerler oluşturulmuştur. Elde edilen yeni değerler diğer veri setleri için eğitim seti olarak kullanılmıştır. Optimizasyon öncesi ve sonrası sınıflandırma başarıları izlenmiştir. Bu işlem sırasında KNN algoritması,*k* değeri 2, 3, 5, 7, 9 için ayrı ayrı çalıştırılmıştır.Yukarıdaki grafiklerde değerler, optimizasyonun uygulandığı *k* parametresine göre belirlenmiştir.Çizelge 4.6'da sınıflandırma başarılarındaki değişim verilmiştir.

Çizelge 4.6. Sınıflandırma optimizasyonu öncesi ve sonrası sınıflandırma başarıları değişimi (%)

Veri Setleri	SET 1	SET 2	SET 3	SET 4	SET 5
SET 1	+3.3	+2.0	+3.4	+1.3	+2
SET 2	+5.3	+1.3	+4.7	+4.7	+4.0
SET 3	+2.7	+2.6	+3.4	+4.0	+3.3

SET 4	+4.0	+2.6	+2.7	+0.3	+4.7
SET 5	+4.7	+6.0	+2.7	+3.3	+0.7

Çizelge 4.7. Sınıflandırma optimizasyonu öncesi ve sonrası ortalama sınıflandırma başarıları (%)

Veri Setleri	***Optimizasyon Öncesi***	***Optimizasyon Sonrası***
SET 1	91,38	92.54
SET 2	91,32	93.38
SET 3	88,8	90.18
SET 4	93.4	94.75
SET 5	90.86	92.94

Her k parametresi için elde edilen sınıflandırma başarılarının ortalamaları alınmıştır. Bu ortalamalar hem optimizasyon öncesi hem de optimizasyon sonrası için ayrı ayrı kaydedilmiştir. Her sınıf için diğer sınıflardaki ortalama başarıların ortalaması da alınarak genel bir sınıflandırma başarısı elde edilmiştir. Çizelge 4.7'de bu başarıların optimizasyon öncesi ve sonrası değişimleri verilmiştir.

5. TARTIŞMA ve SONUÇ

Bu tez çalışmasında veri madenciliği uygulamalarından olan sınıflandırma başarımının,sezgisel yöntemlerle artırılması incelenmiştir. Bu işlem için çok amaçlı genetik algoritma yöntemlerinden olan Sıralı Seçkin Bastırılamayan Genetik Algoritma (NSGA II) kullanılmıştır.Çok amaçlı optimizasyonun uygulanmasıbenzerlik optimizasyonu ve sınıflandırma optimizasyonu olmak üzere iki farklı yönden ele alınmıştır.NSGA II için Java ve JavaFX kullanılarak tamamen özgün bir yazılım geliştirilmiştir. Bu algoritmanın çalışması literatürdeki farklı test fonksiyonları ile denenmiş ve sonuçlar karşılaştırılarak doğrulanmıştır.

Uygulama sırasında sınıflandırma algoritmalarından olan K En Yakın Komşu algoritması ile başarı ölçümleri yapılmıştır. Sınıflandırma başarılarındaki doğrulama için K Katmanlı Çapraz Doğrulama yöntemi uygulanmıştır. Çok amaçlı optimizasyon için amaç fonksiyonları, Öklid Mesafesi ve Kosinüs Benzerliğinden yola çıkılarak belirlenmiştir. Java'nın nesne tabanlı bir dil olmasının verdiği avantaj kullanılarak özgün bir KNN sınıflandırma uygulaması geliştirilmiştir. Geliştirilen yazılımın sınıflandırma sonuçları WEKA yazılımı ile karşılaştırılmış ve uygulamanın WEKA ile benzer sonuçlar verdiği görülmüştür.

Uygulamanın hangi veriler üzerinde test edileceği belirlenirken; verinin ham olarak ele alınmasına ve tüm veri madenciliği aşamalarının sırayla uygulanmasına dikkat edilmiştir. Bu aşamada metin madenciliğinde iyi bilinen bir kıyas veri koleksiyonu olan Classic veri koleksiyonu kullanılmıştır. Belgelerdeki kelimelerin gövdeleme işlemiiçin Porter Stemmer algoritması kullanılmıştır. Bunun sonucunda belgeler içerdikleri gövde kelimelere yani terimlere göre modellenmiştir.Elde edilen kelime frekanslarına Temel Bileşen Analizi (PCA) ile boyut azaltma işlemi uygulanmıştır. Verinin ilk haline göre daha az sütuna sahip yeni hali üzerinde farklı çapraz doğrulama değerleri ile KNN sınıflandırma algoritması uygulanmış ve farklı KFold değerleri ile elde edilmiş sınıflandırma başarılarının ortalaması %94,36 olarak ölçülmüştür. PCA'nın uygulanması MATLAB üzerinden gerçekleştirilmiştir.

PCA sonrasıveri setinin son halinde nesnelerin sıraları önce rasgele karıştırılmış, sonra her sınıftan 50 nesne olacak şekilde 150 satıra sahip 5 farklı veri seti rasgele oluşturulmuştur. Optimizasyon işlemleri bu farklı 5 veri seti üzerinde uygulanmıştır. Bu veri setlerinin rasgele oluşturulması için Java ile bir uygulama geliştirilmiştir.

Aynı sınıftaki nesnelerin benzerlikleri veya yakınlıkları artırılmak istenir.Farklı Sınıftaki nesnelerin ise benzerlikleri veya yakınlıkları düşürülmek istenir. Bu iki durum birbirleriyle çelişir. Bu noktada çok amaçlı benzerlik optimizasyonu devreye girer ve aynı sınıftaki nesnelerin benzerliklerini artırırken aynı zamanda farklı sınıftaki nesnelerin benzerliklerini düşürmek için uygulanır. Benzerlik optimizasyonu uygulanırken Öklid mesafesi ve Kosinüs benzerliğinden yararlanılmıştır. Aynı kategorideki nesnelerin birbirlerine olan uzaklıkları minimumdur. Farklı kategorideki nesnelerin ise benzerlikleri minimumdur. Bu durumda iki amaç fonksiyonu belirlenmiş olur. Minimum-Minimum çok amaçlı problemimizde, aynı sınıftaki nesnelerin uzaklıkları Öklid Mesafesiyle, farklı sınıftaki nesnelerin benzerlikleri de kosinüs benzerliği ile hesaplanır. Optimizasyonun her iterasyonunda, veri setindeki her sütun için 0 ile 1 aralığında, virgülden sonra dört hassasiyetli katsayılar karar değişkenleri olarak kullanılır. Optimizasyon tamamlandığında elde edilen katsayılarla her sütun çarpılıp yeni değerler elde edilir. Oluşan veri seti kendisi de dahil olmak üzere her veri seti üzerinde eğitim kümesi olarak uygulanır. Optimizasyon öncesi ve sonrası sınıflandırma başarıları karşılaştırılır.

Benzerlik optimizasyonu sonuçları incelendiğinde optimizasyon sonrası her veri setinde başarı artımı gözlemlenmiştir. Optimizasyon uygulanırken ortaya çıkan Pareto cephelerin oluşumu farklı iterasyon sayılarına göre grafiklerde ayrı ayrı gösterilmiştir. İterasyonlar belirli bir değerin üzerine çıkıldığında, artık her Pareto cephenin üst üste sonuçlandığı gözlemlenmiştir. Bu çalışmada benzerlik optimizasyonunu sonlandırılması için belirlenen maksimum iterasyon sayısı 100 olarak belirlenmiştir. Benzerlik Optimizasyonundaki başarı artırımı ölçülürken KNN algoritması farklı *k* değerlerine göre çalıştırılmıştır. Bu değerler 2, 3, 5, 7 ve 9 olarak belirlenmiştir. Her k değeri için başarı sonucu saklanmış ve her veri seti için ayrı ayrı ortalamalar hesaplanmıştır.Optimizasyonun her çalışmasında değişimin en fazla olduğu *k* değeri

bulunmak istenmiş ancak *k* değerlerin her seferde değişimi bunun mümkün olmadığını göstermiştir. Bu durumda ortalama değişimin ölçülmesi uygun görülmüştür. Genel ortalama sonuçlarındaki %2.5'lukartış göz önüne alındığında, benzerlik optimizasyonu ile genel sınıflandırma başarısında artış sağlamanın mümkün olduğu gözlemlenmiştir.

Sınıflandırma optimizasyonu, uygulandığı verinin sınıflandırma yeteneğini artırmak için tasarlanmıştır. Çok amaçlı optimizasyon için amaç fonksiyonlarından ilki Öklid mesafesine göre, diğeri ise Kosinüs benzerliğine göre oluşturulmuştur. Buna göre KNN algoritması her iterasyonda bütün veriye,verinin kendisinin veri kümesi olarak kullanılmasıyla uygulanmıştır.Elde edilen hatalı sınıflandırma oranlarının sıfıra indirgenmesi amaçlanmıştır. Hatalı sınıflandırma oranlarının, farklı iterasyon sayılarında sıfıra yaklaşması grafiklerle izlenmiştir.

Sınıflandırma Optimizasyonu ciddi oranda zaman ve bellek maliyeti gerektirmektedir. Veri boyutunun her iki yönlü artışı bu oranı katlamaktadır. Oluşturulan 5 farklı veri setinin her birinin 150 satırdan oluşmasında bu durum etkilidir. PCA ile boyut azaltma işlemi bu konuda bizim için bir avantaj olmuştur. Optimizasyonun sonlandırma kriteri olarak maksimum iterasyon sayısının 50 olarak belirlenmesinde son uygunluk oranlarındaki sabitlik etkili olmuştur. Hem benzerlik optimizasyonu hem de sınıflandırma optimizasyonu için genetik operatörlere aynı değerler verilmiştir. Çaprazlama oranı 0.5, mutasyon oranı 0.05 ve popülasyon sayısı 50 olarak belirlenmiştir.

Sınıflandırma optimizasyonu rasgele oluşturulan beş veri setine de uygulanmış ve uygulandığı veri setihem kendi içinhem de diğer veri setleri için eğitim kümesi olarakkullanılmıştır.Veri setinin yeni hali, optimizasyon öncesi durumuna göre daha yüksek sınıflandırma sonuçları üretmiştir. Bu değişimler her sınıf için grafiklerle gösterilmiş ve artış miktarları incelenmiştir. Bu grafiklerdeki değerler, optimizasyonun uygulandığı *k* değerinin sınıflandırma sonuçlarına göre belirlenmiştir. Sınıflandırma başarısı artışında %6'lık bir değişim görünmüştür.Her veri seti ile test edilen bütün veri setleri için farklı *k*değerlerine göre sınıflandırma başarılarının ortalama değerleri alınmıştır. Bu genel başarı artışıgrafikler ve çizelgelerde gözlemlenmiştir.

Sınıflandırma kavramı, bir veri kümesi (data set) üzerinde tanımlı olan çeşitli kategoriler-sınıflar arasında veriyi dağıtmaktır. Sınıflandırma algoritmaları, belirlenen eğitim kümesinden bu dağıtım şeklini öğrenirler ve daha sonra sınıfının belirli olmadığı test verilerinin ait oldukları sınıfı doğru şekilde bulmaya çalışırlar. Bu tez çalışmasının amacı da sınıflandırma algoritmalarının başarımını artırmaktır. Eğitim verisinin niteliği sınıflandırma başarısındaki en önemli faktörlerdendir. Bu çalışmada eğitim verisinin benzerlik ve sınıflandırma yeteneği göz önüne alınarak düzenli hale getirilmesine çalışılmıştır. Yapılan ölçümler sezgisel optimizasyon yöntemlerinin bu konuda başarılı olduğunu göstermiştir. Sınıflandırma optimizasyonunun bu konuda benzerlik optimizasyonundan daha başarılı olduğu görülmüştür.

Sınıflandırma başarımı artırımı makine öğrenmesinin en önemli konuları arasındadır. Bilgisayarların eldeki verilerle doğru bir çıkarımda bulunması, öğrenme sırasında kullandığı verilerin niteliklerininen iyi düzeyde olmasıyla doğrudan ilgilidir. Bu çalışma bu yönüyle bu konuda yapılacak sonraki çalışmalara referans teşkil edecektir.

KAYNAKLAR

Ali, M., Pant, M., and Abraham, A., 2009. Simplex Differential Evolution***, Network for Innovation and Research Excellence***, **2259:** 6-5.

Amasyalı M. F., Yıldırım T., 2004, Otomatik haber metinleri sınıflandırma***, SIU'04,*** Kusadası.

Bozkurt, İ.N., Bağlıoğlu, Ö., Uyar, E., 2007. Authorship attribution performance of variousfeatures and classification methods***. In Computer and Information Sciences, ISCIS2007 22nd International Symposium***, 1-5.

Coello Coello, C. A., B. Lamont, G. and A., Van Veldhuizen, D., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems, ***Genetic and Evolutionary Computation Series, Springer***.

Dasarathy, B.V., 1991.***Nearest-Neighbor Classification Techniques,***IEEE Computer Society Press,Los Alamitos, California.

Deb, 1999, Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems***, IEEE Transactions on Evolutionary Computation*** 7(3): 205-230.

Deb, K. 2001. ***Multi-Objective Optimization using Evolutionary Algorithms.***Wiley-Interscience Series in Systems and Optimization, 954, Chichester.

Deb, K., Goel, T., 2001. Controlled Elitlist Non-dominated Sorting Genetic Algorithms for Better Convergence, ***In Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization (EMO-2001),*** 61-81.

Deb, K., Goel, T., 2001. A Hybrid Multi-objective Evolutionary Approach to Engineering Shape Design, ***In Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization (EMO-2001),*** 385-399.

Deb, K. & Gupta, H.,2006.Introducing robustness in multi-objective optimization, ***Evolutionary Computation***,**14**(4): 463-494.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II, ***IEEE Transactions on Evolutionary Computation,*****6**(2):182–197.

Dhillon I. S., Fan J., Guan Y., 2001.Efficient Clustering of Very Large Document Collections, ***Data Mining for Scientific and Engineering Applications***.

Durmaz, O. ve Bilge, H., 2011.Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi, ***Signal Processing and Communications Applications (SIU 2011), 2011 IEEE 19th Conference on***, Antalya, 21-24.

Ergül, E.U., 2010.***Çok Amaçlı Genetik Algoritmalar: Temelleri Ve Uygulamaları*** (Doktora Tezi). OMÜ, Fen Bilimleri Enstitüsü, Samsun.

Goel, T., Vaidyanathan, R., Haftka, R.T., Shyy, W., Queipo, N.V., Tucker, K., 2007.Response surface approximation of Pareto optimal front in multi-objective optimization", ***Computer Methods in Applied Mechanics and Engineering***,**196**(4-6): 879-893.

Goldberg, D.E. 1989. ***Genetic Algorithms in Search, Optimization, and Machine Learning***, New York: Addison Wesley.

Goldberg, D.E., 1993. Making genetic algorithms fly: A lesson from the Wright brothers, ***Advanced TechnolojiDevolopment2*** 1-8.

Güven, A., 2007. ***Türkçe Belgelerin Anlam Tabanlı Yöntemlerle Madenciliği***(Doktora Tezi), YTÜ, Fen Bilimleri Enstitüsü, İstanbul.

Han, J. and Kamber, M., 2006.***Data Mining: Concepts and Techniques***, Morgan Kaufmann Publishers, San Francisco, 5-7, 105-106, 348-350.

Hand, D., Mannila, H. and Smyth, P., 2001, ***Principles of Data Mining***, MIT Press, Cambridge, 31-33, 456-464.

Haupt Randly L., Haupt Sue E., 1998.***Practical Genetic Algorithms***, A Willey-Interscience Publication, USA

Holland, J., 1975. ***Adaptation in Natural and Artificial Systems***. University ofMichigan Press, Ann Arbor, ABD.

Ilhan, U., 2001.***Application Of KNN and FPTC Based Text Categorization Algorithms To Turkish News Reports***, Bilkent Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Jackson, P. and Moulinier, 2002.***Natural Language Processing for Online Applications Text Retrieval Extraction and Categorization***, John Benjamins Publishing Company, 32-35.

Jaszkiewicz, A., 1998.***How to Solve It : Modern Heuristics,*** Springer, New York.

Kou H., Gardarin G., 2002.Similarity Model and Term Association for Document Categorization, ***Proceedings of the 13th International Workshop on Database and Expert Systems Applications*** (DEXA'02)

Li, L., Huang, Y.-G. and Liu, Z.-W., 2011.Chinese Text Classification For Small Sample Set, ***The Journal of China Universities of Posts and Telecommunications,*** **18** (1): 83-89

Liao, S.-H., Chu, P.-H. and Hsiao, P.-Y., 2012. Datamining Techniques And Applications – A Decade Review From 2000 To 2011, ***Expert Systems with Applications,*** **39**(12): 11303-11311

Mengle, S.S.R., Goharian, N. and Platt, A., 2007.FACT: Fast Algorithm For Categorizing Text, ***5th IEEE International Conference on Intelligence and Security Informatics,*** New Brunswick, New Jersey, USA, 308-315.

Murata T., 1997. ***Genetic Algorithms for Multi-Objective Optimization***(Doktora Tezi). Osaka Prefecture University

Murata T. ve Ishibuchi H., 1995. MOGA: Multi-Objective Genetic Algorithms, Evolutionary Computation, ***IEEE International Conference***(1): 289

Nanopoulos, A., Theodoridis, Y. and Manolopoulos, Y., 2001, C2P: Clustering based on closest pairs, ***Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001)***, Roma, Italy, 331-340

Osyczka A., 2002. ***Evolutionary Algorithms for Single and Multicriteria Design Optimization***, New York: Physica Verlag.

Pons-Porrata, A., Berlanga-Llavorib, R. and Ruiz-Shulcloper, J. 2007.Topic Discovery Based On Textmining Techniques, ***Information Processing & Management,*** **43** (3): 752-768

Sanwaliya, A., Shanker, K. and Misra, S.C., 2010.Categorization of news articles: A model based on discriminative term extraction method, ***Advances in Databases***

Knowledge and Data Applications (DBKDA 2010), 2010 Second International Conference on, French Alps, France, 145-154.

Salton, G. and Buckley, C., 1988, Term-Weighting Approaches in Automatic Text Retrieval, ***Information Processing and Management,*****24** (5): 513-523.

Schaffer, J.D., 1984. ***Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms***(Doktora Tezi). Vanderbilt University, Nashville, ABD.

Soucy, P. and Mineau, G.W., 2001, A simple Knn Algorithm for text categorization. ***Proceedings IEEE International Conference on Data Mining (ICDM '01),***California, 647-648

Shokri, A., Bozorg Haddad, O. & Mariño, M.A.,2013.Algorithm for Increasing the Speed of Evolutionary Optimization and its Accuracy in Multi-objective Problems, ***Water Resources Management, 27***(7): 2231-2249.

Weng S. S. And Lin Y. J.,2003. A Study on Searching For Similar Documents Based on Multiple Concepts and Distribution of Concepts, ***Expert Systems with Applications***, **25**(3): 355-368.

Yang, Y. and Liu, X.,1999.A Re-Examination Of Text Categorization Methods, ***Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999),*** Berkeley, CA, USA, 42-49.

Zhao, Y., Zobel, J., 2005. Effective And Scalable Authorship Attribution Using Function Words. ***Proceedings of the Second AIRS Asian Information Retrieval Symposium,*** 174- 189.

Zitzler, E., 1999. ***Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications***(Doktora Tezi). Swiss Federal Institute of Technology, Zurich, Switzerland.

ÖZGEÇMİŞ

Hüseyin Ahmetoğlu, 1987 yılında Aksaray'da doğdu. 2005 yılında Mersin Üniversitesi Tarsus Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar EğitimiAnabilim Dalı Bilgisayar Öğretmenliği Bölümü'ne yerleşti. Bu bölümden 2009 yılında mezun oldu. 2010-2015 yıllarında Millî Eğitim Bakanlığı'nda Bilişim Teknolojileri Öğretmenliği yaptı. 2015 yılında Mardin Artuklu Üniversitesi Midyat Meslek Yükseokulu Bilgisayar Programcılığı Bölümü'nde Öğretim Görevlisi olarak akademisyenlik görevine başladı. Bilgisayar Programcılığı Bölüm Başkanlığı ve aynı bölümün uzaktan öğretim Program Koordinatörlüğü görevine devam ettirmektedir. 2015 yılında Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. Evli ve bir çocuk babasıdır.

T.C
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

Tarih: 31/05/2018

Tez Başlığı / Konusu:
BELGE BENZERLİĞİ SONUÇLARININ NSGA-II İLE ÇOK AMAÇLI OPTİMİZASYONU

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 77 sayfalık kısmına ilişkin, 31/05/2018 tarihinde şahsım/tez danışmanım tarafından Turnitin intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 5 (BEŞ) dir.

Uygulanan filtreler aşağıda verilmiştir:
- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit inatch size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

Tarih ve İmza

Adı Soyadı: HÜSEYİN AHMETOĞLU
Öğrenci No: 149101210
Anabilim Dalı: Elektrik Elektronik Mühendisliği
Programı: Tezli Yüksek Lisans
Statüsü: Y. Lisans ☑ Doktora ☐

DANIŞMAN ONAYI
UYGUNDUR
Doç. Dr. Rıdvan SARAÇOĞLU

ENSTİTÜ ONAYI
UYGUNDUR
Prof. Dr. Suat [illegible]
Enstitü Müdürü

(Unvan, Ad Soyad, İmza)