



**BİLECİK
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı**

**SINIFLANDIRMA PROBLEMLERİNDE ÖZELLİK
SEÇİMİ İÇİN KARŞITLIK TABANLI GRİ KURT
OPTİMİZASYON ALGORİTMASI**

**Melis KARAKAŞ
Yüksek Lisans Tezi**

**Tez Danışmanı
Doç. Dr. Uğur YÜZGEÇ**

**BİLECİK, 2020
Ref. No: 10342839**



BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ

**BİLECİK
ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı**

**SINIFLANDIRMA PROBLEMLERİNDE ÖZELLİK
SEÇİMİ İÇİN KARŞITLIK TABANLI GRİ KURT
OPTİMİZASYON ALGORİTMASI**

**Melis KARAKAŞ
Yüksek Lisans**

**Tez Danışmanı
Doç. Dr. Uğur YÜZGEÇ**

BİLECİK, 2020



BİLECİK
SEYH EDEBALI UNIVERSITY

Graduate School of Sciences
Department of Computer Engineering

**OPPOSITION BASED GRAY WOLF ALGORITHM FOR
FEATURE SELECTION IN CLASSIFICATION PROBLEMS**

Melis KARAKAŞ
Master's Thesis

Thesis Advisor
Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2020

TEŐEKKÖR

Yüksek lisans eğitimin ve tez çalışmam süresince desteęi, anlayışı, özverisi ve yol gösterici katkılarından dolayı çok değerli hocam sayın Doç. Dr. Uęur YÜZGEÇ'e

Desteęini hiç esirgemedен her zaman yanımda olan sonsuz sevgi ve anlayış gösteren annem ve babama,

En içten saygı, sevgi ve teşekkürlerimi sunarım.



BEYANNAME

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kılavuzu'na uygun olarak hazırladığım bu tez çalışmada, tez içindeki tüm verileri akademik kurallar çerçevesinde elde ettiğimi, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun olarak sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu Üniversite veya başka bir üniversitede herhangi bir tez çalışmada kullanılmadığını beyan ederim.

17/04/2020

(İmza)

Melis Karakaş

ÖZET

Teknolojinin hızlı bir şekilde ilerlemesiyle çoğalan verilere kolay ve hızlı bir şekilde erişmek için veri kümesinde tanımlanan veriler çeşitli sınıflar arasında dağıtılarak sınıflandırma yapılır. Sınıflandırma problemlerini çözmek için geliştirilen sınıflandırma algoritmalarından yararlanarak veriler benzer özelliklere göre sınıflandırılırlar. Bu sınıflandırma algoritmaları, verilen eğitim veri kümesiyle eğitilerek, öğrenme sağlanır ardından sınıfları belirli olmayan test verileri ile işlem yapıldığında bu verileri doğru bir şekilde sınıflandırmak için çalışırlar. Sezgisel algoritmalar optimizasyon problemlerinde son yıllarda gittikçe popüler bir algoritma haline gelmektedir. Gri Kurt Optimizasyon (GWO) algoritması, gri kurtların toplumsal ve avcılık davranışlarını taklit edilerek geliştirilmiş bir meta sezgisel optimizasyon algoritmasıdır. Bu çalışma kapsamında belirlenen sınıflandırıcıların (K Nearest Neighbour, Support Vector Machine vb.) özellik seçimi için Gri Kurt Optimizasyon algoritması ve karşıtlık tabanlı öğrenme yöntemi kullanılarak geliştirilmiştir.

Karşıtlık tabanlı öğrenme, olasılık kuramına göre rastgele bir noktanın karşıt durumu çözüme, rastgele noktadan daha yakın olabilir. Karşıtlık tabanlı öğrenmede birinci aşama karşıtlık tabanlı başlangıç popülasyonunun belirlenmesi, bir sonraki aşama ise karşıtlık temelli jenerasyon atlama işlemidir. Önerilen algoritma için karşıtlık tabanlı öğrenme dışında mutasyon ve sınır değeri kontrollü gibi yenilikler eklenmiştir. Bu çalışma kapsamında geliştirilen karşıtlık tabanlı GWO algoritması, mevcut kaynaklardan elde edilen sınıflandırma veri setleri için orijinal GWO algoritması ile eş zamanlı olarak test edilip sonuçlar karşılaştırılmıştır. Bu karşılaştırmalar algoritmaların işleyiş zamanları, özellik sayıları, doğruluk değerleri gibi değerler için karşılaştırılmıştır. Karşılaştırmaların sonucunda önerilen geliştirilmiş GWO algoritması orijinal GWO 'ya kıyasla daha başarılı sonuçlar vermiştir. Karşılaştırmalar zaman, doğruluk değeri, sınıflandırma hatası gibi etmenlerle yapılmıştır.

Anahtar Kelimeler: Özellik seçimi; Sınıflandırma; Optimizasyon; Gri Kurt Algoritması; Karşıtlık tabanlı öğrenme.

ABSTRACT

With the rapid advancement of technology, the data defined in the dataset is distributed and distributed among various classes in order to easily and quickly access the reproduced data. Using the classification algorithms developed to solve classification problems, the data are classified according to similar features. These classification algorithms are trained with the given training dataset, learning is provided, and then they work to classify these data correctly when processing with undetermined test data. Intuitive algorithms have become an increasingly popular algorithm in optimization problems in recent years. Gray Wolf Optimization (GWO) algorithm is a meta heuristic optimization algorithm developed by imitating gray wolves' social and hunting behavior. It was developed by using Gray Wolf optimization algorithm and opposition-based learning method for feature selection of classifiers (K Nearest Neighbour, Support Vector etc.) determined within the scope of this study.

Opposite based learning, according to probability theory, the opposite situation of a random point may be closer to the solution than the random point. In the opposition-based learning, the first stage is to determine the opposite-based initial population, and the next stage is the opposition-based generation jump. For the proposed algorithm, innovations such as mutation and boundary value credits have been added apart from opposition-based learning. The opposition-based GWO algorithm developed within the scope of this study was tested simultaneously with the original GWO algorithm for the classification datasets obtained from existing sources and the results were compared. These encounters have been compared for the algorithms' operating times, feature numbers, and accuracy values. The improved GWO algorithm proposed as a result of the comparisons yielded more successful results than the original GWO. Comparisons include time, accuracy, cost value, etc. It was made with factors such as.

Keywords: Feature selection; Classification; Optimization; Gray Wolf Algorithm; Opposition based learning.

İÇİNDEKİLER

Sayfa No

TEŞEKKÜR.....	
BEYANNAME.....	
ÖZET.....	i
ABSTRACT.....	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	viii
SİMGE ve KISALTMALAR DİZİNİ	ix
1.GİRİŞ	1
2.BÜYÜK VERİ (BIG DATA) ve VERİ MADENCLİĞİ (DATA MINING)	3
2.1.Büyük Veri (Big Data)	3
2.2. Veri Madenciliği (Data Mining).....	5
2.2.1. Veri Madenciliğinde Bilgi Keşfinin Süreçleri.....	5
2.2.1.1.Veri Temizleme	6
2.2.1.2.Veri Bütünleştirme.....	6
2.2.1.3.Veri İndirgeme	7
2.2.1.4.Veri Dönüştürme.....	8
2.2.1.5.Seçilen Veri Madenciliği Algoritmasını Uygulama	8
2.2.1.6.Uygulanan Algoritma Sonuçlarını Değerlendirme	8
2.2.2. Veri Madenciliği Yöntemleri	9
2.2.2.1.Sınıflandırma (Classification).....	9
2.2.2.2.Kümeleme (Clustering).....	10
2.2.2.3. Birliktelik Kuralları (Association Rules).....	10
2.2.3.K-En Yakın Komşu Algoritması (K-Nearest Neighbour/K-NN).....	10
3. ÖZELLİK SEÇİMİ (FEATURE SELECTION)	12
3.1.Özellik Seçim Yöntemleri	13
3.1.1.Filtreleme Yöntemleri (Filter Methods)	13
3.1.2.Sarmal Yöntemler (Wrapper Methods)	14
3.1.3.Gömülü Yöntemler (Embedded Methods)	14
4. SEZGİSEL ALGORİTMALAR (HEURISTIC ALGORITHMS).....	15

4.1.Sürü Temelli Algoritmalar	16
4.1.1.Yapay Arı Koloni Algoritması	16
4.1.2.Karınca Koloni Optimizasyon Algoritması	17
4.1.3.Gri Kurt Optimizasyon Algoritması	18
4.1.4.Balina Optimizasyonu Algoritması	18
4.2.Fizik Temelli Algoritmalar	19
4.2.1.Çiçek Tozlaşma Algoritması	20
4.2.2.Yerçekimsel Arama Algoritması	20
4.3.Evrimsel Algoritmalar	21
4.3.1.Genetik Algoritma	21
4.3.2.Differansiyel Evrim Algoritması	22
5. GRİ KURT OPTİMİZASYON ALGORİTMASI (GRAY WOLF OPTIMIZATION ALGORITHM).....	25
5.1.Sosyal Hiyerarşi	26
5.2. Avı Çevreleme.....	26
5.3.Avlanma	27
5.4. Ava Saldırma	28
5.5.Arama	29
6. KARŞITLIK TABANLI GRİ KURT OPTİMİZASYON ALGORİTMASI.....	31
6.1.Karşıtlık Tabanlı Öğrenme	33
6.2.Mutasyon	34
6.3.Sınır Değeri Kontrolü	35
7.DENEYSEL SONUÇLAR.....	37
7.1.Önerilen OppGWO Algoritmasının Analiz Sonuçları	37
7.1.1.Yakınsama Analizleri	38
7.1.2. Arama Geçmişi Analizleri	39
7.1.3. Yörünge Analizleri	40
7.1.4. Ortalama Mesafe Analizleri	42
7.2.Optimizasyon Test Fonksiyonu Sonuçları	43
7.3. Özellik Seçim Sonuçları	47
8. SONUÇ VE TARTIŞMA.....	55
KAYNAKLAR	57

EKLER.....	63
ÖZGEÇMİŞ.....	



ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2.1. Büyük Veri	3
Şekil 2.2. Veri Madenciliği ile Diğer Disiplinler Arası İlişki.....	5
Şekil 2.3. Veri Madenciliği Süreçleri	5
Şekil 2.4. Veri İndirgeme Yöntemleri.....	6
Şekil 2.5. Örnek Bir Dendogram	8
Şekil 2.6. İki Boyutlu Koordinat Sisteminde K-NN Sınıflandırma Aşamaları.....	10
Şekil 3.1. Özellik Seçimi (Feature Selection).....	11
Şekil 3.2. Özellik Seçim Yöntemleri	12
Şekil 4.1. Sezgisel Algoritmaların Sınıflandırılması	14
Şekil 4.2. Yapay Arı Koloni Algoritmasının Akış Diyagramı	15
Şekil 4.3. Gerçek Karıncaların En Kısa Yolu Bulması.....	16
Şekil 4.4. KKO Algoritmasının İşleyişi	16
Şekil 4.5. Kambur Balinaların Su Kabarcığı Yöntemiyle Avlanması	17
Şekil 4.6. Yerçekimsel Arama Algoritmasının Akış Şeması.....	19
Şekil 4.7. Genetik Algoritmanın Akış Şeması.....	20
Şekil 4.8. Diferansiyel Gelişim Algoritmasının Akış Diyagramı	21
Şekil 4.9. Sezgisel Algoritmaların Akış Diyagramı	22
Şekil 5.1. Gri Kurtların Hiyerarşik Yapısı	23
Şekil 5.2. Gri Kurtların Avlanma Stratejisi	25
Şekil 5.3. Avını Ararken ve Avına Saldırırken Gri Kurtlar	26
Şekil 5.4. GWO Algoritmasının Sözde Kodu.....	27
Şekil 6.1. OppGWO Algoritmasının Sözde Kodu	32
Şekil 7.1. OppGWO'nun Analizinde Kullanılan Test Fonksiyonlarının Üç Boyutlu İz Düşüm Görüntüleri	33
Şekil 7.2. OppGWO Algoritmasının Yakınsama Analizleri.....	34
Şekil 7.3. OppGWO Algoritmasının Arama Geçmişi Analizleri	35
Şekil 7.4. FN3 Optimizasyon Fonksiyonu için Seçkin Yörünge Analizi	36
Şekil 7.5. FN6 Optimizasyon Fonksiyonu için Seçkin Yörünge Analizi	37
Şekil 7.6. FN10 Optimizasyon Fonksiyonu için Seçkin Yörünge Analizi	37

Şekil 7.7. FN26 Optimizasyon Fonksiyonu için Seçkin Yörünge Analizi	38
Şekil 7.8. Arama Ajanları Arasındaki Ortalama Mesafe Analizi	39
Şekil 7.9. OppGWO ve GWO için Uygunluk Eğrileri	49



ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 7.1. GWO-OppGWO için 10- Boyutlu CEC'14 Optimizasyon Sonuçları	40
Çizelge 7.2.: GWO-OppGWO için 30- Boyutlu CEC'14 Optimizasyon Sonuçları	41
Çizelge 7.3. GWO-OppGWO için 50- Boyutlu CEC'14 Optimizasyon Sonuçları	42
Çizelge 7.4. 10,30,50 Boyut İçin GWO ve OppGWO Algoritmalarının Başarı Yüzdeleri	42
Çizelge 7.5. Sınıflandırma Veri Setleri.....	43
Çizelge 7.6. GWO ve OppGWO Algoritmalarının Özellik Seçimi İçin Doğruluk Değeri Sonuçları	45
Çizelge 7.7. GWO ve OppGWO Algoritmalarının Özellik Seçimi İçin Sınıflandırma Hatası Sonuçları	45
Çizelge 7.8. GWO ve OppGWO Algoritmalarının Özellik Seçimi İçin İşleyiş Süreleri (Sn).....	46
Çizelge 7.9. GWO ve OppGWO Algoritmalarının Özellik Seçimi İçin Özellik Sayıları Değeri Sonuçları.....	47

SİMGE VE KISALTMALAR DİZİNİ

Simgeler

X : Herhangi Bir Gri Kurdun Konumu

J_r : Atlama Hızı

U_b : Üst Sınır (Upper Boundary)

L_b : Alt Sınır (Lower Boundary)

X_p : Avın Konum Vektörü

Kısaltmalar

GWO: Gray Wolf Optimization

ÇTA: Çiçek Tozlaşma Algoritması

OppGWO: Opposition Based Gray Wolf Optimization

PSA: Pattern Search Algorithm

DEA: Differential Evolution Algorithm

GA: Genetic Algorithm

PGWO: Powell Gray Wolf Optimization

OBL: Oppositon Based Learning

KNN: K- Nearest Neighbour

SVM: Support Vector Machine

KKO: Karınca Koloni Optimizasyonu

Rnd: Rastgele (Random)

GB:Gigabyte

TB: Terabyte

PB: Petabyte

1.GİRİŞ

Teknolojide yaşanan gelişmelerle birlikte hızla çoğalan verilere hızlı ve kolay bir şekilde erişmek için birçok veri madenciliği metodu geliştirilmiştir. Veri madenciliği, mevcut kaynaklardaki tanımlamalara dayanarak, farklı araç ve teknolojilerden yararlanarak büyük veri kümeleri içerisinde saklı kalmış ilişki, örüntü ve bilgilerin açığa çıkarılmasını hedefleyen çok aşamalı bir süreç denilebilir. Bu sürecin önemli adımlarından biri de özellik seçimidir (Budak,2018).

Özellik seçiminin (feature selection) temel amacı, performansı etkilemeden orijinal veri kümesini temsil edebilecek en iyi alt kümeyi seçme işlemi olarak tanımlanmaktadır. Özellik seçimi (nitelik seçimi veya değişken seçimi), kullanılacak algoritmaya uygun özellikleri değerlendirerek veri kümesindeki n adet özellik içerisinde en iyi k adedi seçme işlemi olarak tanımlanmaktadır. (Forman,2013).

Özellik seçimi, çözüme ulaştırılmak istenen problem için en belirleyici ve en önemli özellikleri seçerek veri setinde bulunan özellik sayısını en aza indirmeyi amaçlamaktadır. Veri boyutunu azaltarak, özellik sayısını indirmek çözümleme sürecinde uygulayıcıya çok yarar sağlayacaktır.

Özellik seçimi ile ilgili mevcut kaynaklarda birçok metot bulunur. Bu yöntemlere örnek verilecek olursa, karar ağaçlarından bahsedilebilir. Sınıflandırma problemlerinde yaygın olarak kullanılan karar ağaçları özellik seçiminde de kullanılmaktadır. Özellik seçimi için karar ağaçlarının yapısında farklı metotlar bulunmaktadır. Bunlardan en yaygın olarak kullanılan algoritma ID3 algoritmasıdır. ID3 algoritması, bir özelliği seçme ve bu özelliğin değerlerine göre verilen örnek kümesini ayırma işlemini tekrarlanan bir süreç ve bir dizi eğitim kümesi aracılığıyla öğrenmektedir. Bu örneğe ek olarak, Destek vektör makineleri-tekrarlı özellik elemesi bir çeşit geriye doğru özellik seçim yöntemidir. Bu yöntem sınıflandırma başarısını optimize eden özellik alt kümesini bulmak için, öncelikle tüm özellikleri bir amaç fonksiyonuna bağlı olarak derecelendirmekte ve daha sonra en düşük skora sahip özelliği özellik kümesinden çıkartmaktadır. Bu yöntemlere ek olarak son yıllarda popüler olan sezgisel algoritmalar da bu yöntemler arasındadır (Budak,2018).

Sezgisel Algoritmalar, matematiksel modelinin tam olarak bilinmediği, problemin çözümüne sezgisel bir yaklaşımla ulaşıldığı yöntemlerdir. Burada modelin

doğruluğunu ispatı istenmez. Algoritmadan beklenen problem çözümünü daha basit hale getirmesi veya tatmin edici bir sonuç üretmesidir (Ateş,2018).

Son zamanlarda gittikçe popüler olan sezgisel algoritmalarından biri olan popülasyon temelli Gri Kurt Optimizasyon algoritması gri kurtların liderlik hiyerarşisine ve doğada avlanmak için sergiledikleri içgüdüsel davranışlara dayanmaktadır. Önerilen algoritma yani Karşıtlık Tabanlı Gri Kurt Optimizasyon algoritmasının (OppGWO) dayandığı fikir karşıtlık tabanlı öğrenmedir. Bu yöntem olasılık tabanlıdır. Eğer bir nokta çözüme yakın değilse onun karşıt noktasında yer alan nokta çözüme daha yakın olabilir. Önerilen algoritmayı iyileştirmek için karşıtlık tabanlı öğrenme dışında da bazı eklemeler yapılmıştır.

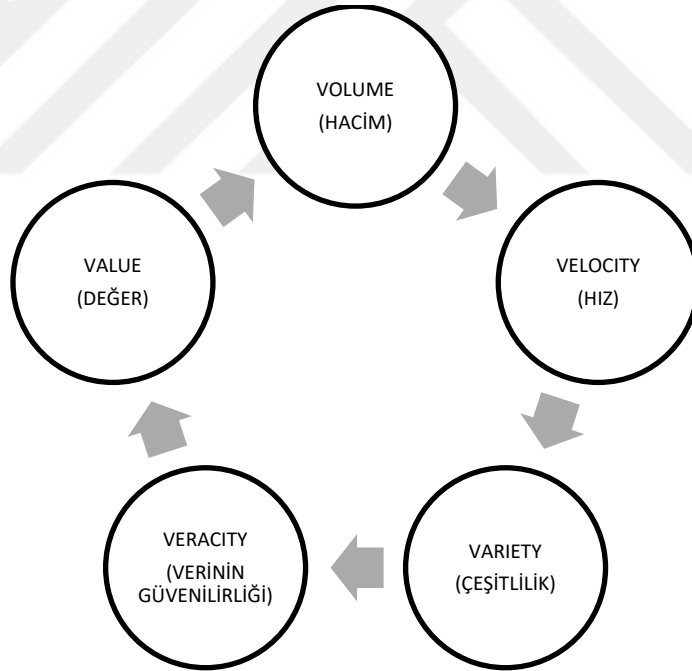
Bu eklemelerle birlikte önerilen algoritma (OppGWO) ve orijinal algoritma (GWO) eş zamanlı olarak testlere (benchmark,sınıflandırma) tabi tutulmuşlardır. Bu testlerden ilki CEC'14 benchmark testidir. Bu testlerde belirlenen boyutlarda ve fonksiyonlarda iki algoritmada (OppGWO ve GWO) test edilmiştir ve sonuçlar elde edilmiştir. Ardından iki algoritmaya da (OppGWO ve GWO) sınıflandırma testleri uygulanmıştır. Bu testler literatürden alınan, birbirinden farklı birçok veri seti üzerinde uygulanmıştır. Sonuçlar karşılaştırılarak analizler yapılmıştır.

Bu çalışma kapsamında, özellik seçimi problemleri için karşıtlık tabanlı öğrenmeyi temel alan Gri Kurt Optimizasyon algoritması geliştirilmiştir. OppGWO algoritması mevcut kaynaklardan elde edilen sınıflandırma test verileri için orijinal GWO algoritması ile karşılaştırılmıştır.

2.BÜYÜK VERİ (BIG DATA) ve VERİ MADENCLİĞİ (DATA MINING)

2.1.Büyük Veri (Big Data)

Veri, bilgisayarların algıladığı, işlediği veya daha sonra kullanabilmek için depoladığı her şeye denir (Bery ve Linoff,2000). Büyük veri kavramı, ilk kez Michael Cox ve David Ellsworth tarafından 1997 yılında düzenlenen 8. IEEE Görüntüleme Konferansı'nda (Proceedings of the 8th Conference on Visualization), "ApplicationControlled Demand Paging for Out-of-core Visualization" adlı makalede kullanılmıştır (Cox ve Ellsworth, 1997). Büyük veri, veri tabanı yönetim araçlarını kullanarak işlenmesi zor olan büyük ve karmaşık veri kümelerinin toplanması için kullanılan terimdir (Mahalakshmi, vd.,2001). Büyük veri kavramı, bu kavramı oluşturan beş ögeyle açıklanmaktadır (Şekil 2.1). Buna 5V kavramı denir.



Şekil 2.1.Büyük veri

Hacim (Volume): Büyük veri hacmi günden güne katlanarak artmaktadır. Toplanan bilgiler oldukça büyüktür. Bu sebeple modern veri tabanı yönetim araçları bu bilgileri işleyememektedir. Bu yüzden de bu bilgiler kullanılamamaktadır. En büyük

zorluk ise, büyük veri hacimleri arasındaki ilişkinin nasıl belirleneceği ve ilgili verilerden nasıl değer yaratılacağıdır.

Hız (Velocity): Veriler katlanarak artan bir hızla üretilmektedir. Terabayt (1024GB) ve petabaytlarda (1024TB) sürekli büyümektedir. Hız, hem verilerin ne kadar hızlı üretildiğini hem de verilerin talebi karşılamak için ne kadar hızlı işlenmesi gerektiğini içerir.

Çeşitlilik (Variety): Veriler, yapılandırılmamış, yarı yapılandırılmış ve yapılandırılmış veriler olmak üzere farklı şekillerde oluşturulmaktadır. Bu veriler heterojen ve doğaldır. En büyük zorluk ise, insanlar tarafından veya bir uygulamaya girdi olarak, yapılandırılmamış verileri almak ve bu verilerden anlamı çıkarmak için büyük verilerin işlenmesidir.

Veri Güvenliği (Veracity): Veri bankası içinde tutulan özetlerde kullanılan varyans miktarını tanımlamaktadır ve veri kümesinde nasıl yayıldıklarını veya nasıl kümeleştiklerini ifade etmektedir. Üretilen veriler doğada belirsizdir. Hangi bilgilerin doğru ve hangilerinin güvenilir olmadığını bilmek zordur. Veri güvenliği belirsiz veya kesin olmayan verilerle ilgilenmektedir.

Değer (Value): Büyük Veri yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış bilgilerden oluşur. Yapılandırılmış veriler, tamsayılar, karakterler ve tamsayıların veya karakterlerin dizileri gibi temel veri türleridir. İlişkisel veri tabanlarında kullanılırlar. Yapılandırılmış verilerin veya veri kayıtlarının en yaygın biçimi, sütun ve satırlardan oluşan belirli bilgilerin depolandığı bir veri tabanıdır. Yapılandırılmış veriler, içindeki veri türüne göre de aranabilir. Yapılandırılmış veriler bilgisayarlar ve insanlar tarafından anlaşılır bir şekilde tasarlanır.

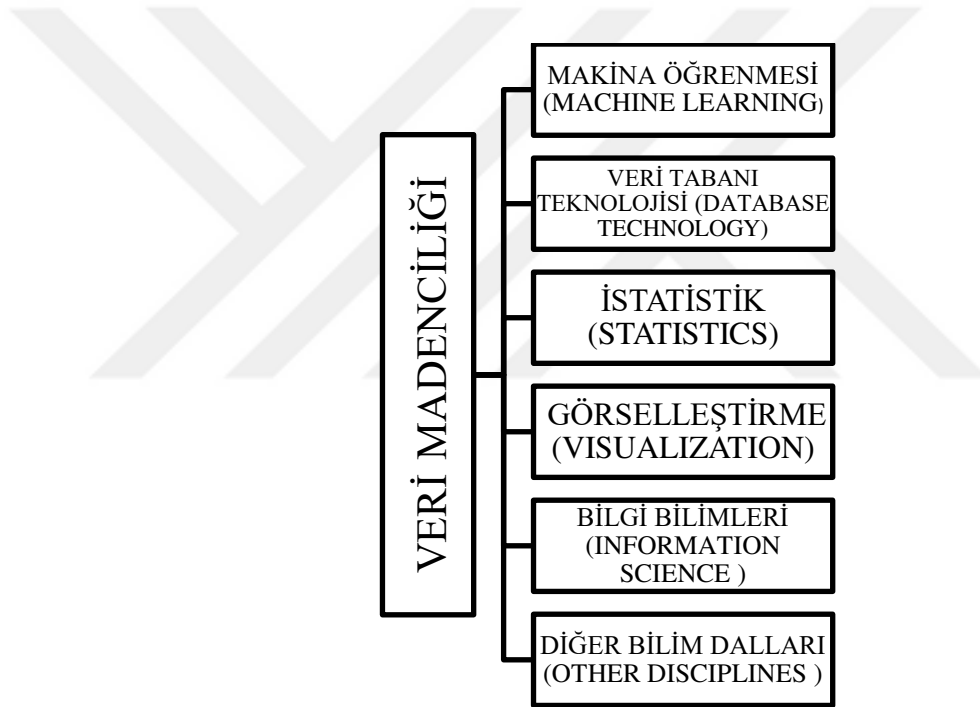
Yapılandırılmamış bilgi, e-postalar, videolar, tweet'ler, facebook gönderileri, çağrı merkezi sohbetleri, kapalı devre TV görüntüleri, web sitesi tıklamaları gibi büyük verinin %90'nını oluşturur. Bu tür verilen önceden tanımlanmış bir biçimi yoktur. Yarı yapılandırılmış veriler, önceki iki veri türünün birleşimidir, genellikle XML kullanılarak temsil edilir. Bahsedilen 4V bir araya gelince başka bir V kavramı ortaya çıkar. Bu da value (değer)'dir. Verinin değeri çok önemlidir. İşte bahsedilen bu 5V büyük veri kavramını oluşturmaktadır.

Verilerin artan taleplerini karşılamak için araç ve yöntemlerin kapasitesini ve performansını artırmalıyız. Büyük verileri düzenli bir şekilde işleyebilirsek verilerimizi

değerli hale getiririz. Veri madenciliği ile büyük veri yığınları arasında gizli kalmış verileri elde ederiz (Mahalakshmi, vd.,2001).

2.2. Veri Madenciliği (Data Mining)

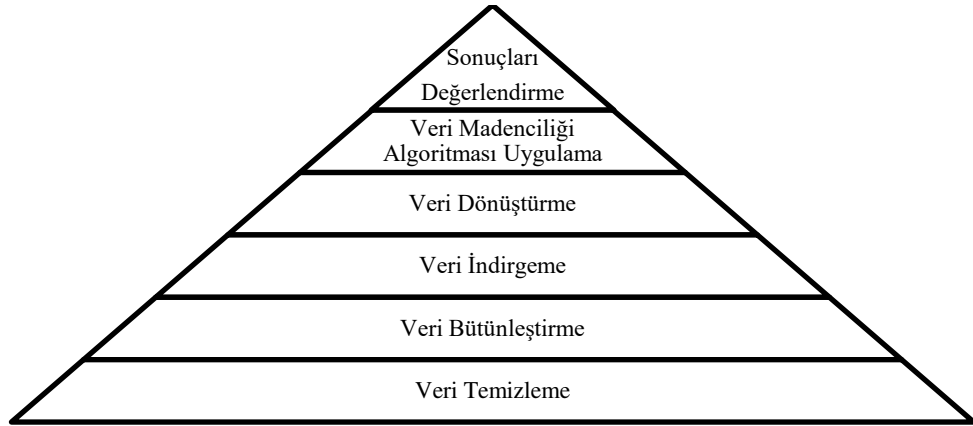
Veri madenciliği, diğer bir adla veri tabanında bilgi keşfi; çok büyük veri yığınları arasında tutulan, anlamı daha önce keşfedilmemiş potansiyel olarak faydalı ve anlaşılır bilgilerin çıkarıldığı ve arka planda Şekil 2.2’de gösterilen veri tabanı yönetim sistemleri, istatistik, yapay zekâ, makine öğrenme, paralel ve dağıtık işlemlerin bulunduğu veri analiz tekniklerine denir (Berry ve Linoof, 2000; Han, vd.,2011; Ayre,2006).



Şekil 2.2. Veri madenciliği ile diğer disiplinler arası ilişki

2.2.1. Veri Madenciliğinde Bilgi Keşfinin Süreçleri

Büyük veri tabanlarında değerli veya gizli kalmış olan bilgileri algılamak ve erişmek oldukça zordur. Veri tabanında bilgi keşfinin süreçleriyle (Şekil 2.3), bu değerli veya gizli kalmış bilgilere belli metotlar uygulanarak verilerin kullanılabilir hale getirilmesinde çok büyük rol oynar. Veri tabanında bilgi keşfinin süreçleri şu şekilde ilerler:



Şekil 2.3. Veri madenciliği süreçleri

2.2.1.1. Veri Temizleme

Bazı uygulamalarda elde edilen veriler, üzerinde işlem yapılmasına uygun özelliklere sahip olmayabilir. Veriler eksik, hatalı veya tutarsız olabilir. Bu tür verilere gürültülü veriler denir. Buna benzer durumlarda verilerin bahsedilen gürültüden temizlenmesi gerekir. Bunun için aşağıdaki metotlar kullanılabilir.

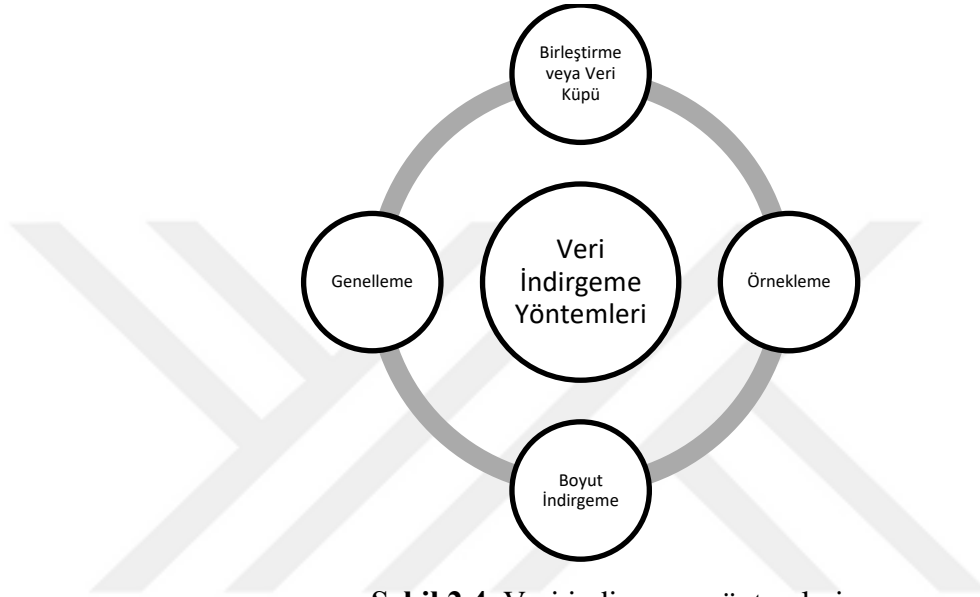
- a) Eksik değer içeren veri setinden çıkarılabilir.
- b) Kayıp değerlerin yerine genel bir sabit kullanılabilir. Bu sabit bütün kayıp değerler için kullanılabilir.
- c) Parametrenin içerdiği sadece bir sınıfa ait veya tüm sayısal değerlerin ortalaması alınıp eksik değer yerine kullanılabilir.
- d) Eksik değere uygun bir tahmin yöntemi (Bayes formülü veya karar ağacı) uygulanabilir. Eksik değer tahmin edilip yerine yazılabilir (Özkan,2016).

2.2.1.2. Veri Bütünleştirme

Farklı veri tabanlarından veya veri kaynaklarından elde edilen verilerin beraber değerlendirmeye alınabilmesi için farklı türdeki verilerin tek bir türe dönüştürülmesi veya bütünleştirilmesi gerekir. Veri madenciliği uygulaması için veri ambarı hazırlanmış ise veri, veri ambarı oluşturma sırasında bütünleştirilir. Fakat böyle bir yapı yer almıyorsa veri bütünleştirme işlemi doğrudan veri kümesine uygulanır (Özkan,2016).

2.2.1.3. Veri İndirgeme

Veri madenciliği uygulamalarında analiz süreçleri uzun zaman alabilir. Analizden önce, elde edilecek sonucun değişmeyeceğini öngörülüyorsa veri sayısı veya öznitelik (özellik) sayısı azaltılabilir. Veri indirgeme Şekil 2.4'te gösterildiği gibi çeşitli biçimlerde yapılabilir.



Şekil 2.4. Veri indirgeme yöntemleri

Veriyi birleştirme veya veri küpü aşamasında veri çok boyutlu veri küpleri haline dönüştürebilir. Bu sayede yapılacak olan analizler sadece belirlenen boyutlara göre yapılır.

Veri madenciliğinde kullanılan veriler, veri tabanındaki tutulan tablolar gibi düşünülebilir. Bu tablolar gözlemleri veya örnekleri içerebilir. Bunlara veri madenciliğinde öznitelik veya özellik adı verilir. Veri madenciliği uygulamasındaki çözümlemelerde bütün özelliklerin katılması uygun olmayabilir. Bu durumda ‘özellik seçimi’ yöntemleri kullanılarak boyut azaltılabilir (Özkan ve Erol,2015).

Örneklemede ise, büyük bir veri kümesi yerine onu temsil edecek daha küçük bir veri kümesi tercih edilir. Genellemede, veri tek tek değil de genel olarak ifade edilir.

2.2.1.4. Veri Dönüştürme

Verileri belirli durumlarda veri madenciliği analizlerine doğrudan katmak uygun olmayabilir. Özniteliklerin ortalama ve varyans değerleri birbirinden farklı olduğu zaman değeri daha büyük değere sahip olanlar, diğerleri üzerinde daha fazla baskısı olur. Çok büyük ve çok küçük değerler değerlendirmenin sağlıklı bir biçimde sonuçlanmasını engeller. Bu sebeple dönüşüm yöntemi kullanılarak söz konusu özniteliklerin veya değişkenlerin standartlara uygun olması sağlanacaktır.

Ve bu amaçla kullanılan yöntemlerden bazıları;

a) Min-Max Normalleştirilmesi

Bir gruptaki verilerin en büyük ve en küçük değerleri ile işlem yapılır. Diğer veriler, bu değerlere göre normalleştirilir. Buradaki amaç 0-1 aralığında olacak şekilde normalleştirmektir.

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.1)$$

X^* dönüştürülmüş değeri, X gözlem değerini, X_{min} en küçük gözlem değerini, X_{max} ise en büyük gözlem değerini ifade eder.

b) Z-Skor Standartlaştırma

Değişkenin ortalaması ve standart sapmasına bağlı olarak yeni bir değere dönüştürülmesi işlemidir.

$$X^* = \frac{X - \bar{X}}{\sigma_x} \quad (2.2)$$

X^* dönüştürülmüş değeri, X gözlem değerini, $\bar{X}$ verinin aritmetik ortalaması, σ_x ise gözlem değerlerinin standart sapmasını ifade eder.

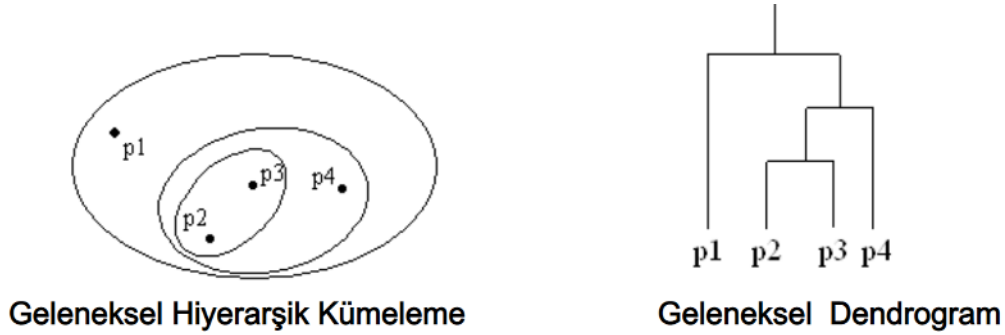
2.2.1.5. Seçilen Veri Madenciliği Algoritmasını Uygulama

Veri madenciliği yöntemlerinde işlemlerden gerekli olanlar yapıldıktan sonra veri hazır hale getirilir. Sonra konuyla ilgili veri madenciliği algoritmaları uygulanır. Bu algoritmalar sınıflandırma, kümeleme, birliktelik kuralları konusunda olacaktır.

2.2.1.6. Uygulanan Algoritma Sonuçlarını Değerlendirme

Veri madenciliği algoritması uygulandıktan sonra, elde edilen sonuçlar düzenlenerek ilgili yerlere iletilir. Sonuçlar çoğunlukla grafiklerle desteklenir. Örneğin

hiyerarşik kümeleme modeli uygulanmışsa sonuçlar Şekil 2.5'te gösterilen dendrogram (öbek ağacı) denilen özel bir grafikte sunulur (Özkan,2016).



Şekil 2.5. Örnek bir dendrogram (Öbek Ağacı) (Steinbach, vd.,2005).

2.2.2. Veri Madenciliği Yöntemleri

Veri Madenciliğinde yöntemin uygulanacağı veri seti ve uygulama yapılacak alana göre farklı yöntemler bulunmaktadır. Veri analiz aşamasında kullanılacak yöntem ya da yöntemlerin neler olacağı bilgi keşfi sürecinde tanımlanan probleme göre belirlenir.

2.2.2.1.Sınıflandırma (Classification)

Sınıflandırma veri madenciliğinde, veri tabanındaki gizli örüntüleri ortaya çıkarmak için sıkça kullanılan bir yöntemdir.

Veri madenciliğinde bağımsız değişkenler bilinen niteliklerdir, istenen ise bağımlı değişkenleri tahmin etmektir. Sınıflama teknikleri ile veri seti içinde sınıfı belli olmayan gözlemlerin çeşitli özelliklerine göre, nitelikleri önceden bilinen, eğitim seti ile tanımlanan bir veri sınıfına ataması yapılır. Bu yöntem ile veri kümesi içerisindeki kategorisi bilinmeyen bir verinin kategorik değerleri tahmin edilebilir (Bharadwaj, vd.,2018; Vadim,2018; Bilen,2014; Alagöz,2014).

Sınıflandırma problemleri için geliştirilmiş çeşitli veri madenciliği algoritmaları bulunmaktadır. Bu algoritmalarından en çok kullanılanları karar ağaçları, yapay sinir ağları, bayes sınıflandırıcılar, k-en yakın komşu, destek vektör makineleri şeklindedir.

2.2.2.2.Kümeleme (Clustering)

Kümeleme, verilerin belirlenen özniteliklerine göre ve öznitelik değerlerine göre kümelere ya da alt gruplara ayrılabilmesi için istatistik ve makine öğrenmesi alanlarında geliştirilen bir yöntemdir.

Makine öğrenmesinin temel adımlarından biri olan denetimsiz öğrenme küme analizine dayanır. Verilerin benzerlik durumlarına göre gruplandırması ile insan beynini taklit eder. Buna göre kümeleme gizli örüntülerin denetimsiz öğrenmeyle ortaya çıkarılmasıdır.

2.2.2.3. Birliktelik Kuralları (Association Rules)

Veri tabanı içinde yer alan verilerin birbirleri ile olan bağlantılarını araştırarak hangi olayların senkronik bir şekilde birlikte gerçekleşebileceklerini ortaya çıkarmaya çalışan bir veri madenciliği yöntemidir (Özkan,2016).

Birliktelik kuralları daha çok pazarlama alanında uygulanmaktadır. Söz gelimi, müşteri sepetine bir ürün aldığı anda başka hangi ürünleri aldığı belirli bir olasılıkla hesaplanır. Birlikte alınan ürünler belirlendikten sonra mağazalar rafları ona göre düzenlediğinde müşteriler istedikleri ürünlere daha kolay ulaşabilirler.

2.2.3.K-En Yakın Komşu Algoritması (K-Nearest Neighbour/K-NN)

K En Yakın Komşu yöntemi, sınıflandırma problemini çözen denetimli öğrenme yöntemleri arasında yer alır. Bu yöntemde; sınıflandırma yapılacak verilerin öğrenme kümesindeki normal davranış verilerine benzerlikleri hesaplanarak; en yakın olduğu düşünülen k verinin ortalamasıyla, belirlenen eşik değere göre sınıflara atamaları yapılır. Bu süreç aşağıdaki adımlara göre gerçekleşir (Çalışkan ve Soğukpınar,2008). Şekil 2.6 ‘da bu sürecin grafiksel anlatımı yer almaktadır.

1.K’nın belirlenmesi: İki boyutlu koordinat düzlemindeki problem çerçevesinde aranacak komşu sayıları belirlenir (Şekil 2.6.1).

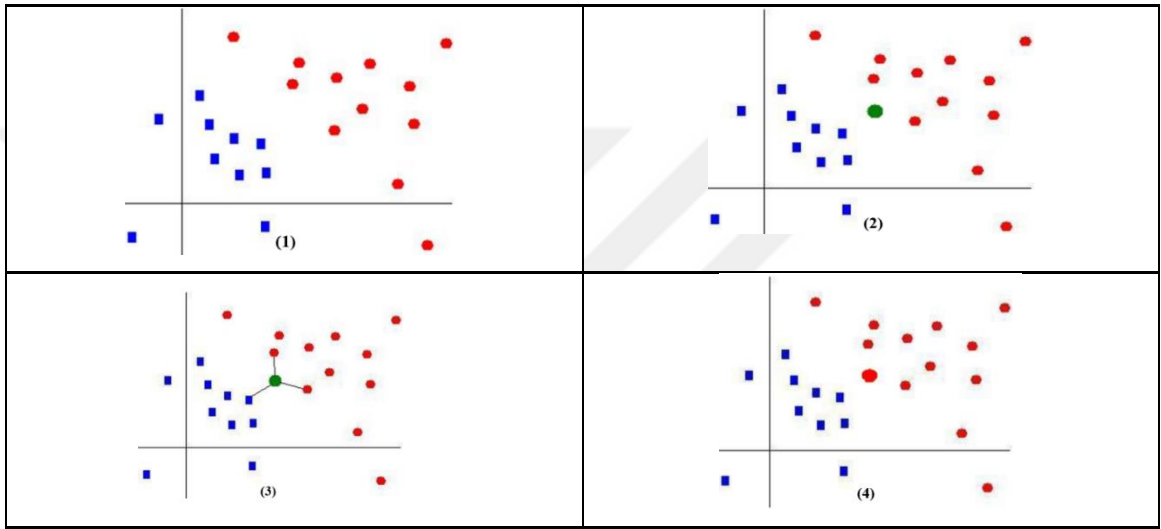
2.Uzaklıkları Hesaplanması: Problem çerçevesinde belirlenen gözlem noktaları arasında mesafe Öklid uzaklık formülüne göre hesaplanır (Şekil 2.6.2).

$$d(i, j) = \sqrt{\sum_{k=1}^p (X_{ik} - X_{jk})^2} \quad (2.3)$$

3. En Küçük Uzaklıkların Belirlenmesi: Satırlar sıralanarak en küçük k adet satır belirlenir. Bu k adet nokta gözlem noktasına en yakın noktalardır (Şekil 2.6.3).

4. Seçilen Satırlara İlişkin Sınıfların Belirlenmesi: Gözlem noktasına en yakın olan gözlem değerlerinin sınıflarına bakılır ve hangi sınıfın çoğunlukta olduğuna bakılır (Şekil 2.6.4).

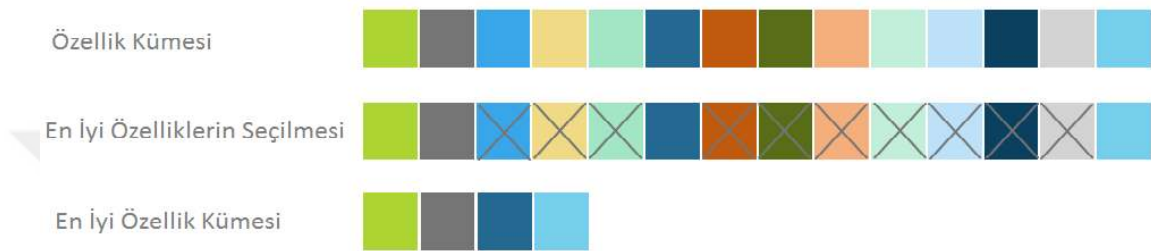
5. Yeni Gözlem Sınıfı: Gözlem noktasının yeni sınıfı çoğunlukta olan sınıf olarak belirlenmiş olur.



Şekil 2.6. İki boyutlu koordinat sisteminde K-NN sınıflandırma aşamaları. (Şeker,2015)

3. ÖZELLİK SEÇİMİ (FEATURE SELECTION)

Özellik seçimi sınıflandırma problemlerinde ve birçok öğrenme algoritmalarında önemli bir yere sahiptir. Özellik seçiminde amaç, veri setini temsil edebilecek en iyi alt kümeyi seçme işlemidir. Yani n adet özellik içerisinde en iyi k adedi seçme işlemidir (Şekil 3.1).



Şekil 3.1.Özellik Seçimi (Medium,2020)

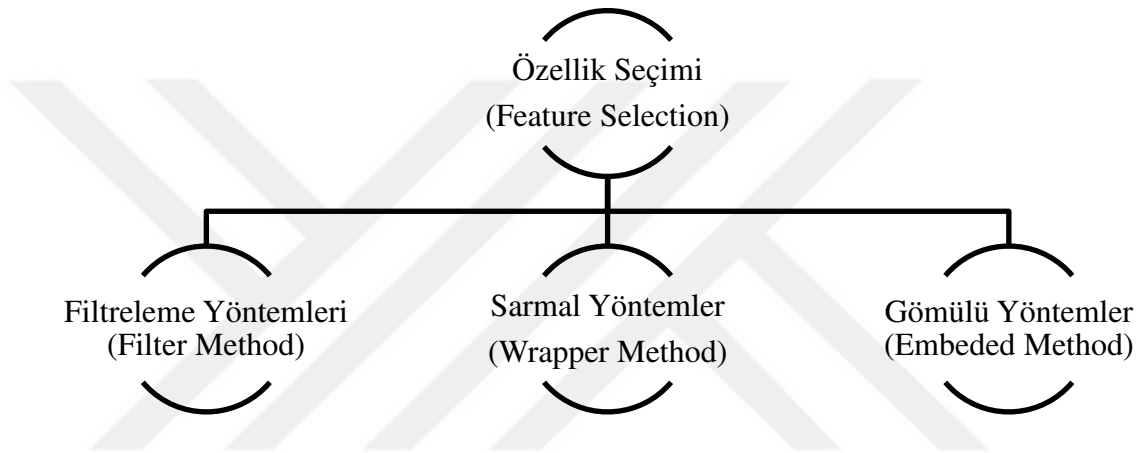
Özellik seçiminde ilgilenilen problem için en yararlı ve en önemli özellikleri seçerek veri setindeki özellik sayısını azaltmak amaçlanmaktadır. Özellik seçme işleminin avantajları aşağıdaki gibidir:

- Özellik kümesinin boyutunu düşürerek algoritmanın hızını artırır.
- Gürültülü veriyi ortadan kaldırır.
- Veri kalitesini geliştirir.
- Veri kümesinin daha basit ve anlaşılır olmasını sağlar.
- Veri kümesinin oluşturmak için yapılan veri toplama işleminde kaynak tasarrufu sağlar.
- Veri depolamak için ihtiyaç duyulan hafıza miktarını azaltır.
- En iyi özelliklerle çalışmak elde edilen modelin başarısını artırır.

Özellik seçimi birçok alanda kullanılır. Söz gelimi, metin madenciliği, kanser teşhisi, sahtecilik tespiti, müşteri kayıp analizi gibi birçok örnek gösterilebilir. Özellik seçimi için kullanılan bazı yöntemler vardır. Bunlar, istatistiksel bilgiye dayanan filtreleme yöntemi, özellikler üzerinde arama yapmayı sağlayan sarmal yöntemler ve en iyi bölen ölçütünü bulmaya dayalı gömülü yöntemler olmak üzere üç gruba ayrılır (Budak,2018).

3.1.Özellik Seçim Yöntemleri

Özellik seçim yöntemleri Şekil 3.2’de görüldüğü gibi üçe ayrılır. Bunlardan ilki olan filtreleme yöntemlerinde önce özellik seçimi yapılır ardından da veri madenciliği algoritması çalışır, ikinci olarak sarmal yöntemlerde veri madenciliği algoritması en iyi özelliklerin seçimi için bir araç olarak kullanılmaktadır. Son olarak gömülü yöntemlerde ise, veri madenciliği algoritması ve özellik seçimi algoritması senkronik bir şekilde çalışmaktadır.



Şekil 3.2. Özellik Seçim Yöntemleri

3.1.1.Filtreleme Yöntemleri (Filter Methods)

Filtreleme yöntemleri veri madenciliğinde kullanılan en eski özellik seçim yöntemleridir. Bu yöntemlerde herhangi bir sınıflandırıcı kullanılmadan belirli (uzaklık, bilgi vb.) ölçümleri istatistiksel kriterlere dayalı fonksiyonlar aracılığıyla özellik seçimi yapılmaktadır (Guyon ve Elisseeff,2003). Bu yöntemlerde, veri setinde bulunan her bir özellik için bir değerlendirme fonksiyonu ile değer elde edilir ve elde edilen bu değerler içerisinde en yüksek değere sahip olan özellikler en iyi özellik alt kümesine aktarılır (Budak,2018). Yaygın olarak kullanılan filtreleme yöntemleri:

- Fisher Skor
- T-Skor
- Welch T-İstatistiği

- Ki-Kare Testi
- Bilgi Kazancı
- Kazanç Oranı
- Korelasyon Tabanlı
- Relielf
- One-R

3.1.2.Sarmal Yöntemler (Wrapper Methods)

Sarmal yöntemler, aramayı bilgilendirmek için bir öğrenme algoritması kullanarak özellik alt kümesinde arama yapar. Özellik alt kümesine eklenebilen veya özellik alt kümesinden kaldırılabilen her bir özellik için, öğrenme algoritmasının tahmini doğruluğunu hesaplar. En iyi tahmin performansını gösteren özellikler seçilir (Das,2001). Sarmal yöntemler, filtreleme yöntemlerine göre daha başarılıdır fakat hesaplama maliyeti yüksektir. Yaygın olarak kullanılan sarmal yöntemler:

- Ardışık İleri Yönde Seçim
- Ardışık Geri Yönde Seçim
- L Ekle R Çıkar
- Ardışık İleri Yönde Kayan Seçim
- Ardışık İleri Yönde Kayan Seçim

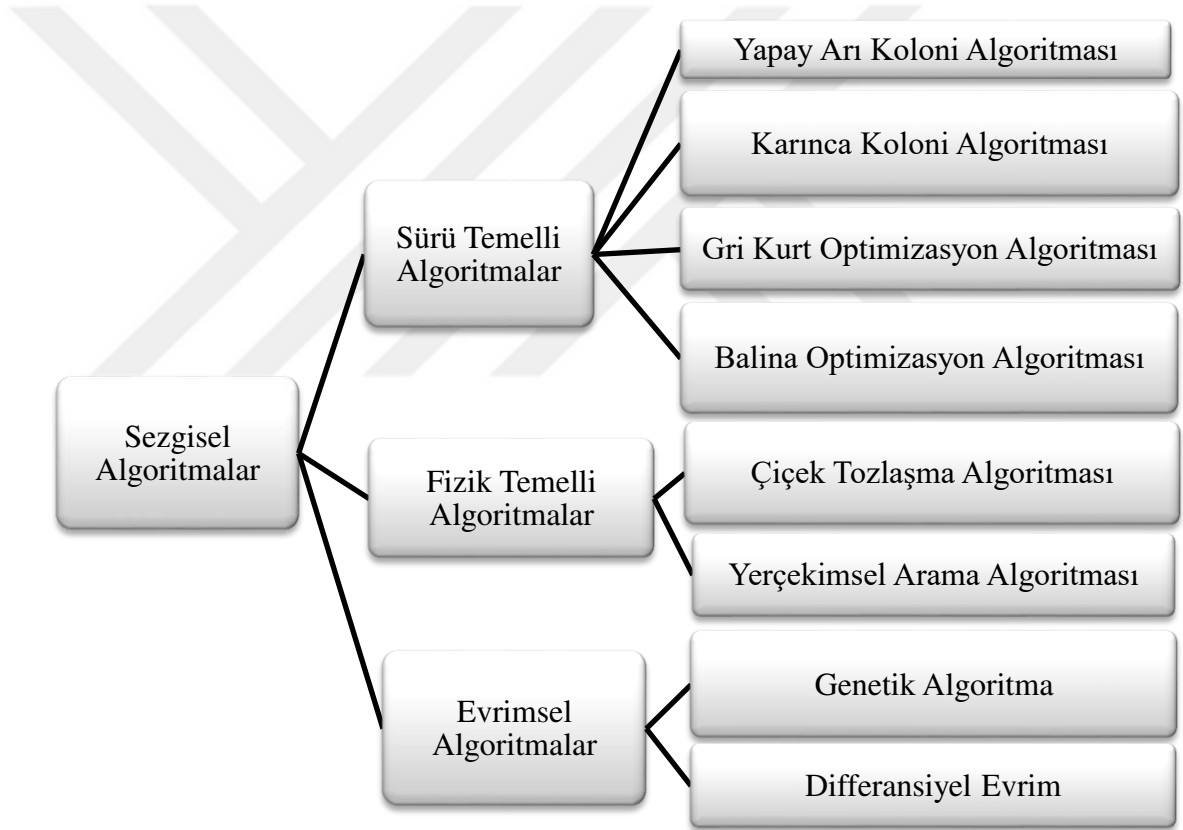
3.1.3.Gömülü Yöntemler (Embedded Methods)

Gömülü yöntemler, sınıflandırma ve özellik seçimi algoritmasını yapısında barındırırlar. Bu sebeple sınıflandırma ve özellik seçimi süreçlerini eşzamanlı olarak gerçekleştirirler. (Budak,2018). Gömülü yöntemler, filtreleme yöntemlerine göre daha başarılıdır fakat hesaplama maliyeti yüksektir. Yaygın olarak kullanılan gömülü yöntemler:

- Karar Ağaçları
- Destek Vektör Makineleri

4. SEZGİSEL ALGORİTMALAR (HEURISTIC ALGORITHMS)

Sezgisel algoritmalar, sistemlerin matematiksel modelinin tam olarak bilinmediği durumlarda yalnızca verilen amaç fonksiyonuna göre ilgili değişkenlerin türetilmesini sağlamaktadır (Ateş,2018). Yani problemin çözümüne sezgisel bir yaklaşımın uygulanmasıdır. Burada uygulanan yöntemin doğruluğunun kanıtı istenmez. Algoritmadan beklenen problemin karmaşıklığı daha basite indirilmesi ya da algoritmanın tatmin edici sonuç bulabilmesidir. Sezgisel algoritmalarda sürü temelli, fizik temelli vb. olmak üzere birçok yöntem vardır (Şeker, 2008). Bunlardan bazıları Şekil 4.1’de listelenmiştir.



Şekil 4.1. Sezgisel Algoritmaların Sınıflandırılması

Sezgisel algoritmalar, arama uzayında o algoritmaların temel stratejisine göre tarama yapar. Örneğin Genetik algorithmada genlerin değişim durumlarına göre, yapay arı koloni algoritmasında arıların hareket yapısına göre, karınca kolonisi algoritmasında karıncaların en kısa yolu bulma yöntemine göre, gri kurt optimizasyonunda gri kurtların

doğadaki avlanma stratejilerine göre, balina optimizasyonu kambur balinaların beslenme davranışına göre, diferansiyel gelişim algoritması da genetik algoritma bireylerin arasındaki farkın bir başka bireye ilave edilmesine göre, çiçek tozlaşma algoritması çiçekli bitkilerin üreme sürecine göre, yerçekimsel arama algoritması da Newton'un hareket kanunlarından ikincisi olan ivme kanunu ve evrensel çekim kanuna göre tasarlanmışlardır (Akyol ve Alataş, 2012).

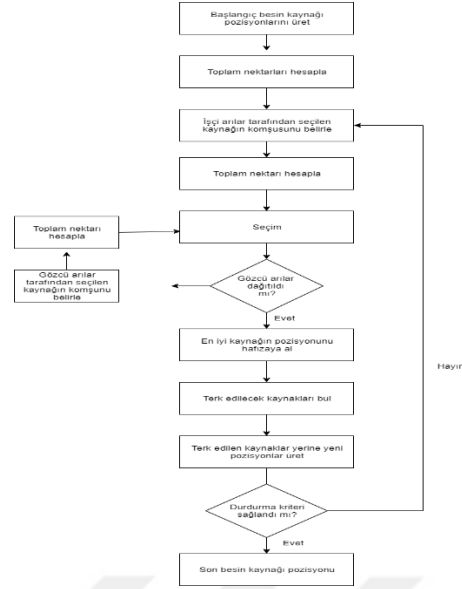
4.1.Sürü Temelli Algoritmalar

Sürü, birbiri ile etkileşim içinde olan düzensiz yapılı bireylerin oluşturduğu topluluğa denir. Sürülerde n adet temsilci belirli bir amaca yönelik davranışları gerçekleştirmek ve hedefe ulaşmak için birlikte çalışmaktadır. Temsilciler hareketlerini yönetmek için basit bireysel kurallar kullanmakta ve grubun kalanıyla kendilerine (sürüye özgü) özgü bir etkileşim yolu ile amaçlarına ulaşmaktadır. (Akyol ve Alataş, 2012).

4.1.1.Yapay Arı Koloni Algoritması

Yapay Arı Koloni algoritması, bal arılarının besin ararken sergiledikleri davranışlardan esinlenerek geliştirilmiş sürü zekasına dayalı sezgisel bir algoritmadır. Besin aramada üç ana öge vardır. Bunlar; besin kaynakları, görevi belli olan işçi arılar ve görevi belli olmayan işçi arılardır (Tereshko ve Loengarov, 2005). Görevi belirli olan işçi arılar önceden belirlen besinleri kovana getirmekle ve besin kaynağının konumu diğer arılarla paylaşmakla görevlidirler. Görevi belli olmayan işçi arılar ise, rastgele besin arayan kâşif arılar ve kovana bekleyen gözcü arılar olmak üzere ikiye ayrılırlar (Karaboğa ve Öztürk,2010). YAKA'da ilk adım başlangıç değerlerinin atanması ve görevli arı için yeni kaynak üretilmesidir. Ardından gözcü arılar için seçim işleminin gerçekleştirilmesidir. Seçim işleminden sonra gözcü arılar besin kaynağı tüm bölgeleri dağıldıysa en iyi çözüm saklanır ve sonlandırma şartları kontrol edilir. Eğer besin kaynağı tüm bölgeleri dağılmadıysa, bu adım dağılım sağlanıncaya dek devam eder (Akyol ve Alataş, 2012).

Şekil 4.2.'de Yapay Arı Koloni Algoritmasının akış diyagramı verilmiştir.

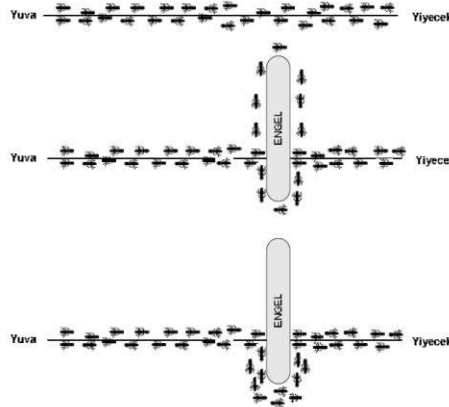


Şekil 4.2.Yapay Arı Koloni Algoritması Akış Diyagramı

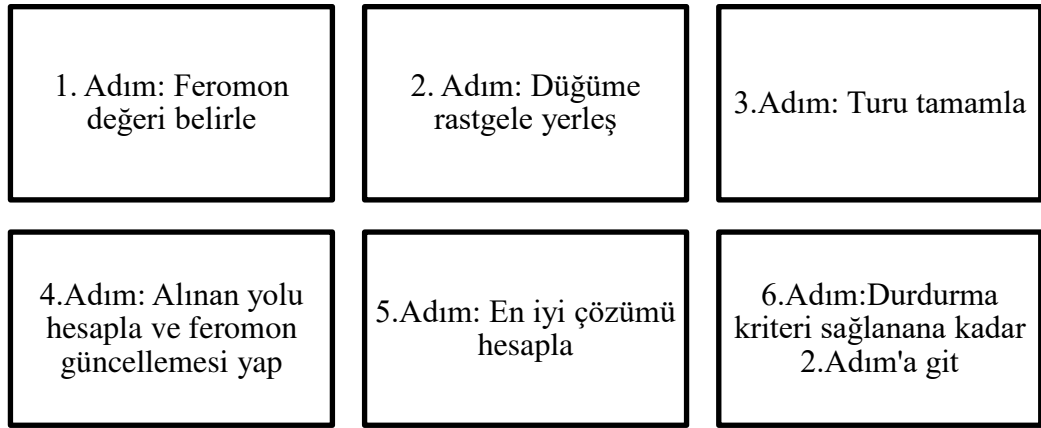
4.1.2.Karınca Koloni Optimizasyon Algoritması

Karınca koloni algoritması, tümleşik optimizasyon problemlerinin çözülmesi için karınca kolonilerinin işbirlikçi davranışlarından esinlenerek geliştirilen meta-sezgisel bir algoritmadır. Karıncalar yiyecek arayışlarında izlediği yola feromon adında bir koku bırakırlar. Eğer yol kısa ise koku daha yoğundur. Diğer karıncalar seçeceği yola koku yoğunluğuna göre karar verirler (Şekil 4.3). Algoritma da bu davranıştan esinlenerek yapay feromonların güncelleştirilmesi ile tekrarlanan bir algoritmadır.

Şekil 4.4’de KKO algoritmasının işleyişi verilmiştir (Kesintürk ve Söyler, 2006).



Şekil 4.3. Gerçek karıncaların en kısa yolu bulması



Şekil 4.4. KKO algoritmasının işleyişi

4.1.3. Gri Kurt Optimizasyon Algoritması

Bu algoritma, gri kurtların doğadaki avlanma ve toplumsal davranışlarından esinlenerek geliştirilmiş sezgisel bir algoritmadır. Çalışmanın ileriki bölümlerinde bu algoritma kapsamlı olarak anlatılacaktır.

4.1.4. Balina Optimizasyonu Algoritması

Balinalar hava kabarcığı denilen bir yöntemle beslenirler. Suyun altında nefes vererek hava kabarcıkları oluştururlar. Bu hava kabarcıkları avları toplar. Balinalar su kabarcıklarının içerisinde yüzeye doğru yükselirler. Yükselirken de nefes verdikleri için hava kabarcığı oluşturmaya devam ederler. Avına yaklaştıkça kabarcık çemberini daraltarak hedef küçülür. Bu sayede avını hareketsiz hale getirerek Şekil 4.5'teki gibi avlanır. Algoritmada bu davranıştan esinlenerek üç bölüme (avın etrafını sarma, ava doğru hareket etme, av arama) ayrılarak algoritmayı çalıştırır.



Şekil 4.5. Kambur balinaların su kabarcığı yöntemi ile avlanması (Tanyıldızı ve Cigal,2017).

Aşağıda balina optimizasyonu algoritmasının sözde (kaba) kodu verilmiştir (Tanyıldızı ve Cıgal,2017).

A ve C: Yakınsama Vektörleri

p: [0,1] arası rastgele bir sayı

l: [-1,1] arası rastgele bir sayı

α : iterasyon boyunca 2'den 0'da lineer azalan vektör

Başlangıç popülasyonu ayarla X_i ($i = 1, 2, \dots, n$)

Her bir arama ajanının uygunluk değerini hesapla

X^* = Bilinen en iyi arama ajanı

while (t (mevcut iterasyon) < maksimum iterasyon sayısı)

for (her bir arama ajanı için)

 Güncelle α, A, C, l ve p

if (p < 0.5)

if ($|A| < 1$)

 Arama ajanı konumu güncelle

else if ($|A| < 1$)

 Rastgele bir arama ajanı seç

 Arama ajanını güncelle

end if

else if (p ≥ 0.5)

 Arama ajanının konumu güncelle

end if

end for

Kısıt dışına çıkan bireylere sınır değerini ver amaç fonksiyon değerlerini hesapla

Daha iyi çözüm bulunmuşsa en iyi ajanı güncelle

t=t+1

end while

Sonuç X^*

4.2.Fizik Temelli Algoritmalar

Doğada gerçekleşen fiziksel olaylardan esinlenerek geliştirilen algoritmalar.

4.2.1.Çiçek Tozlaşma Algoritması

Çiçek Tozlaşma Algoritması (ÇTA) doğadaki işleyiş taklit edilerek ortaya çıkmış meta-sezgisel algoritmalarından birisi olup, 2012 yılında Xie-She Yang tarafından geliştirilmiştir (Yang,2014). ÇTA, çiçekli bitkilerin üreme sürecinden esinlenerek geliştirilmiştir. Aşağıda ÇTA'nın sözde kodu verilmiştir (Korkmaz ve Akgüngör,2018).

```

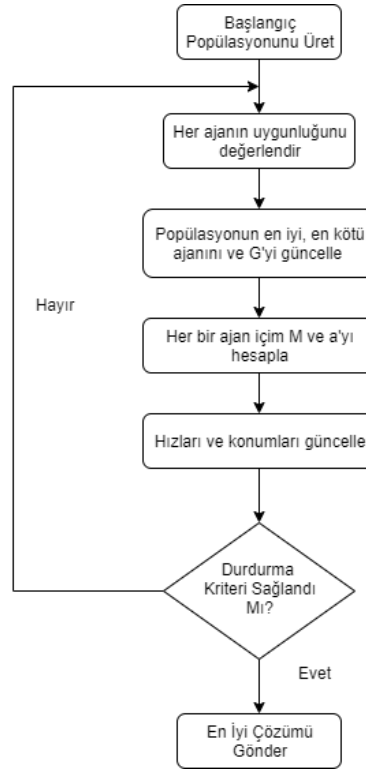
n: Polen Sayısı
p: [0,1] arası rastgele sayı
t: mevcut iterasyon
while (t<maksimum iterasyon)
    for i=1: n
        if rand<p (üniform dağılım)
            Levy dağılımı:L (parametre sayısı kadar)
            Biyotik Üreme
        else
            Draw  $\epsilon$  (üniform dağılıma göre [0,1])
            Rastgele j ve k çözümlerini seç
            Abiyotik üreme
        end if
        Yeni çözümü al ve kontrol et
        Yeni çözüm daha iyiye popülasyonu güncelle
    end for
    En iyi çözümü seç  $g_*$ 
end while

```

4.2.2.Yerçekimsel Arama Algoritması

Bu algoritma, Newton'un hareket ve yerçekimi kanununa dayanarak geliştirilmiş bir algoritmadır. Nesnelerin belirli bir kütlesi vardır ve yerçekimi kuvvetiyle birbirlerini çekerler. Nesneler, kütlesi en ağır olana doğru hareket ederler. Her nesne bir çözümü temsil etmektedir. Bu çözümlerin ağırlıkları başarılarını göstermektedir. Algoritma sonuçlanana dek çözümlerin ağırlığı en yüksek olana doğru yakınsaması ile devam etmektedir. En sonda ağırlığı en yüksek olan çözüm küresel çözüm olarak belirlenmektedir. Algoritmanın işlediği süre boyunca en ağır kütle diğerlerine göre daha

yavaş hareket eder ve diğerlerini kendine çekecek bir kuvvet uygular. Sonlandırma koşulu sağlandığında kütlesi en fazla olan nesne, problemin optimum çözümünü oluşturmuş olur. (Eskandar, vd.,2012; Demiro1, vd.,2018). Şekil 4.6'te GSA'nın akış diyagramı yer almaktadır.



Şekil 4.6. Yerçekimsel Arama Algoritmasının Akış Diyagramı (Koç, vd.,2018)

4.3.Evrimsel Algoritmalar

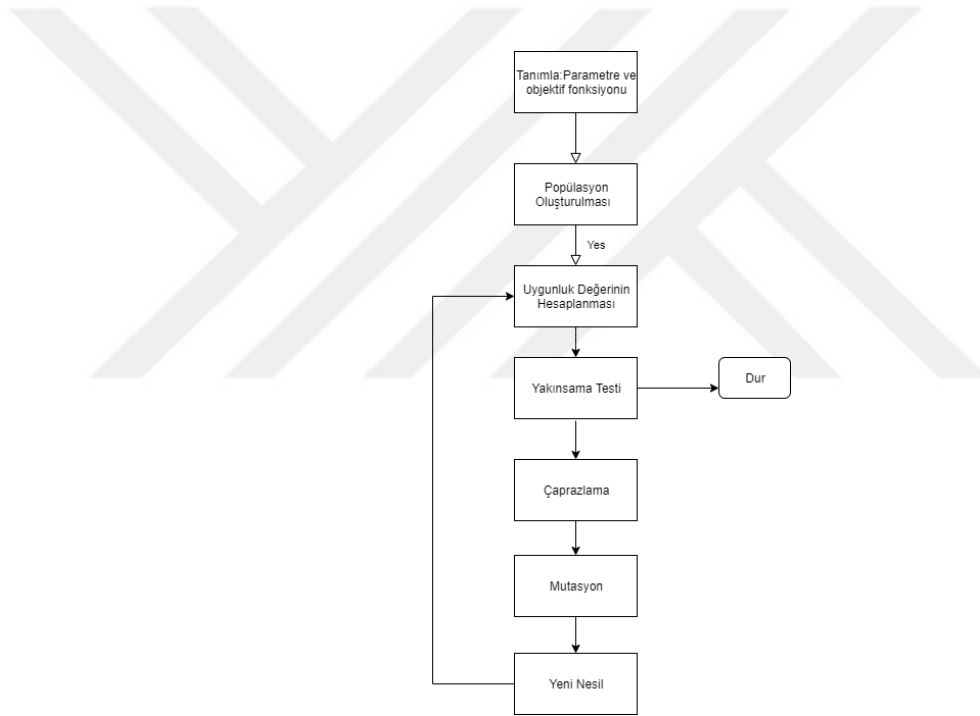
Bu algoritmalar, güçlü olan yaşar ilkesine dayanarak, Darwin'in evrim teorisi temel olarak geliştirilen algoritmalarlardır. Evrimsel algoritma, bir veri kümesinden özel bir veriyi bulmak için kullanılır. Bu özelliğiyle uygun bir optimizasyon yöntemi sayılır. Evrim Teorisi'nin güçlü olan yaşar ilkesine bağlı olarak, algoritma sürekli iyileşen çözümler üretir. Kötü olan çözümler ise elenir (Akduman ve Türkay,2010).

4.3.1.Genetik Algoritma

Doğadaki evrimsel süreçlerden esinlenerek, benzer bir mantıkla çalışan optimizasyon yöntemidir. GA değişkenleri, genleri temsil ederken, değişken kümesi de kromozomları oluşturur. Genetik algoritmada aday çözüm kromozom biçiminde temsil

edilir. Bu aday çözümlerin oluşturduğu kümelere de popülasyon adı verilir. Genetik algoritmalarda Çaprazlama ve Mutasyon denilen iki adet temel işlemci vardır. Çaprazlamada popülasyondan iki adet birey seçilir. Bu bireylerde çaprazlama yapılacak nokta belirlenir ve bu noktadan itibaren bireylerin elemanları karşılıklı olarak yer değiştirilir. Mutasyonda ise bireylerin genleri değiştirilir (Özsağlam ve Çunkaş,2008). Genetik Algoritmanın ilk adımı başlangıç popülasyonunun belirlenmesidir. Arından her bireyin uygunluk değeri hesaplanır. Sonlandırma koşulu testi gerçekleştirilir. Eğer koşul sağlanıyorsa algoritma sonlandırılır. Koşul sağlanmıyorsa bir sonraki adıma geçilerek yeni çözümler aranır. Sonlandırma koşulu sağlanana dek bu işlem devam eder.

Şekil 4.7’de GA’nın akış diyagramı verilmiştir.

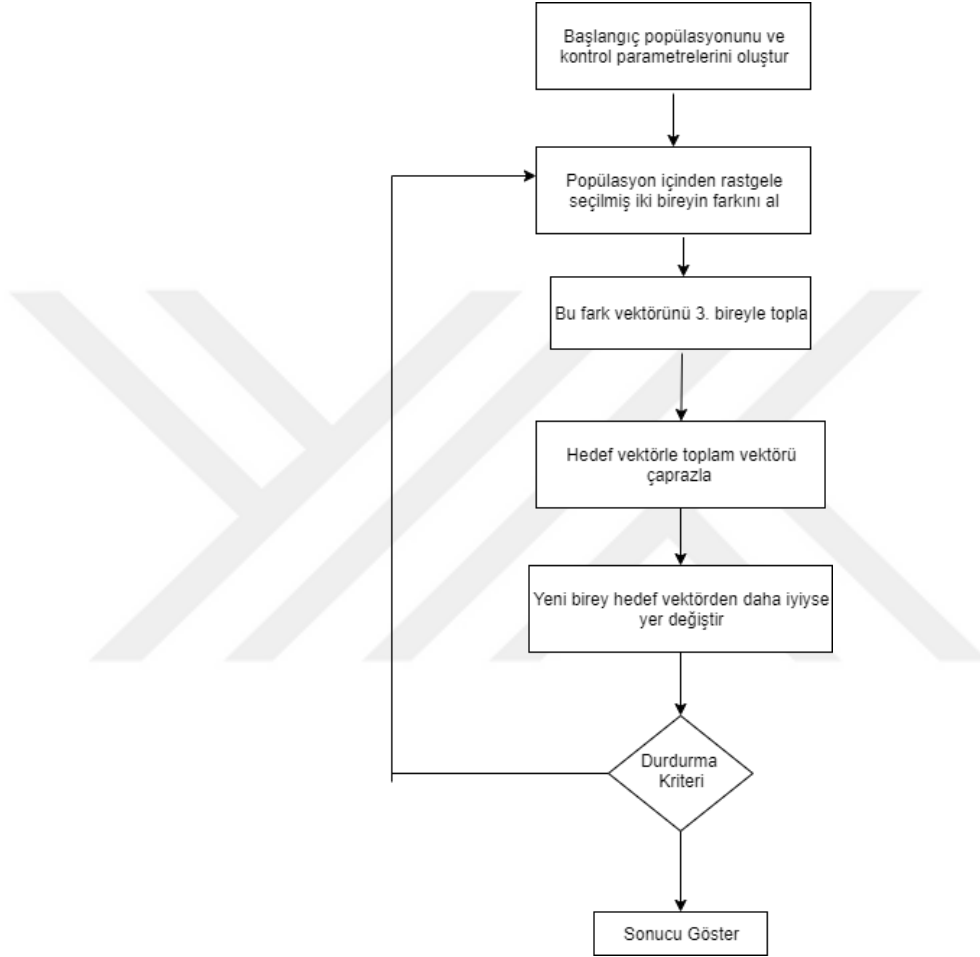


Şekil 4.7.Genetik Algoritma Akış Şeması (Özsağlam ve Çunkaş,2008)

4.3.2.Differansiyel Evrim Algoritması

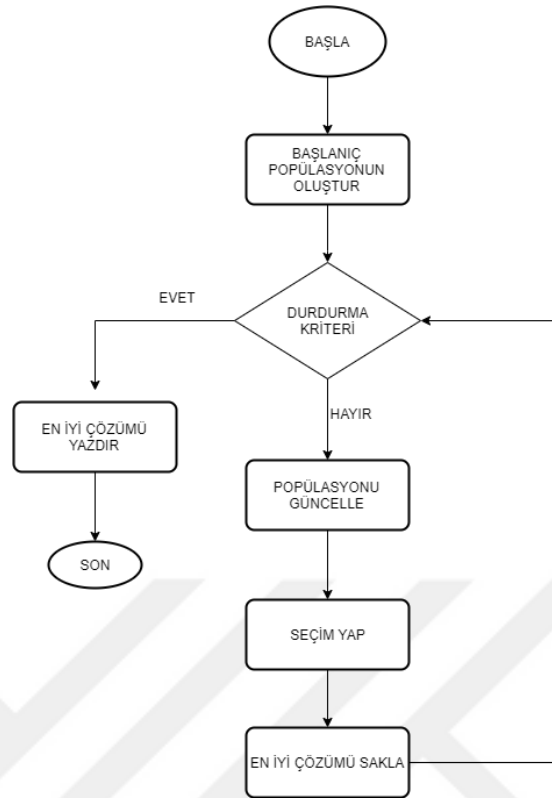
Optimizasyon problemlerinde yaygın olarak kullanılan bu algoritma, gelişim algoritmasıdır (Storn,1997; Kenneth,1999). DEA, GA’dan farkı gelişmiş mutasyon işlemi kullanılmaktadır. Vektör çiftlerinin farkına dayalı olan mutasyon işlemi, vektörlerin kendi dağılımları tarafından belirlenir. Buna ek olarak, bir ebeveyn vektörden, bir deneme vektörü üretmek için mutasyon ve çaprazlama birlikte kullanılır.

Algoritmanın ilk adımı olarak başlangıç popülasyonu belirlenir. Ardından mutasyon ve çaprazlama işlemleri gerçekleştirilir. Sonlandırma koşulu sağlanıyor ise sonuç gösterilir. Koşul sağlanmıyorsa Şekil 4.8'deki başa dönülür. Sonlandırma koşulu sağlanana dek bu adımlar gerçekleştirilir (Özsağlam ve Cunkaş,2008).



Şekil 4.8.Differansiyel Gelişim Algoritması Akış Şeması

Bu tip algoritmaları test etmek için farklı optimizasyon problemlerine ait matematiksel fonksiyonlar kullanılmaktadır. Bu fonksiyonlarda amaç verilen arama uzayında en iyi (minimum veya maksimum uygunluk değeri) maliyet değerine en yakın çözümü bulmaktır. Sezgisel algoritmalarda, başlangıçta arama uzayı belirli bir yöntemle göre veya rastgele oluşturulabilir. Şekil 4.9'da sezgisel algoritmaların genel akış şeması gösterilmektedir.

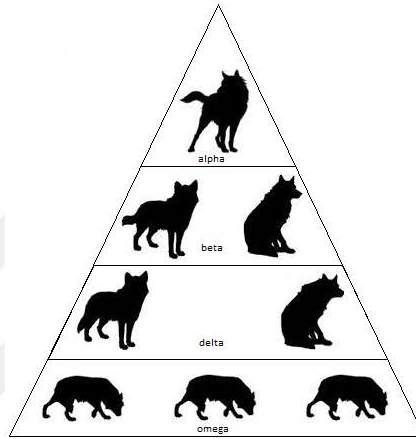


Şekil 4.9. Sezgisel Algoritmaların Akış Şeması

Birçok algoritmada başlangıç popülasyonu rastgele olarak oluşturulur. Bazı algoritmalarda ise başlangıç popülasyonu belli bir yöntemle oluşturulur. Sezgisel algoritmaların genel olarak popülasyondaki yeni değerlerini üretirken, arama uzayındaki herhangi değerden yola çıkarak bir sonraki adım değerini üretir. Ama bazen de arama uzayındaki değerden farklı olarak bir sonraki adım değeri üretebilmektedir. Bu yapı algoritmaların çeşitlenmesindeki en önemli faktörler arasında gösterilir.

5. GRİ KURT OPTİMİZASYON ALGORİTMASI (GRAY WOLF OPTIMIZATION ALGORITHM)

Gri kurtlar (*Canis Lupus*) Canidae familyasına aittir. Gri kurtlar süper avcı olarak kabul edilir. Bu yüzden besin zincirinin en tepesinde yer alırlar. Gri kurtlar doğada grup halinde yaşarlar. Gruplar 5-12 kurttan oluşur. Gri kurtların hiyerarşik düzeni Şekil 5.1’de gösterildiği gibidir.



Şekil 5.1. Gri Kurtların Hiyerarşik Yapısı (Doğan ve Yüzgeç,2018)

En tepede yer alan grup alfa kurdudur. Bir erkek ve bir dişiden oluşur. Alfa kurdu uyuma yeri, uyuma zamanı ve avlanma gibi birçok konuda karar veren gruptur. Alfa'nın kararları gruba kabul ettirilir. Bununla birlikte alfa kurdunun diğer kurtları takip etmesi gibi bazı demokratik davranışları da gözlemlenmiştir. Gri kurtlar alfanın kararlarını, kuyruklarını aşağı doğru tutarak kabul ederler.

Piramitte ikinci kurt grubu beta kurdudur. Beta kurdu alfaya karar vermesinde yardımcı olan kurt grubudur. Beta kurdu erkek veya dişi olabilir. Beta kurdu, alfa kurdu öldüğünde veya yaşlandığında alfa olabilmek için muhtemel en iyi adaydır. Beta kurdu alfaya saygı göstermelidir fakat ek olarak da alt seviyedeki kurtlara komut verir.

En düşük seviyeli gri kurt omegadır. Omega günah keçisi rolünü oynar. Omega kurtları daima diğer kurtlara itaat etmek zorundadırlar. Yemek yemesi için izin verilen son kurtlardır. Omega grupta önemli bir birey gibi görünmeyebilir fakat omegayı kaybettikleri takdirde kavgaların ve problemlerin olduğu gözlemlenmiştir. Bunun

nedeni şiddetin omega kurdu tarafından önlenmesidir. Bu bütün grubun tatmin edilmesine ve baskın yapının korunmasına yardım eder. Bazı durumlarda omega grubun bebek bakıcısıdır.

Eğer bir kurt alfa, beta ve omega değilse ona ast (bazı kaynaklarda delta) denir. Delta kurtları alfa ve beta kurduna itaat etmek zorundadır fakat omega kurtlarına hükmedebilirler.

İzciler, gözcüler, avcılar, yaşlılar ve bakıcılar bu kategoriye aittir. İzciler bölge sınırlarını izlemekten ve herhangi bir tehlike durumunda grubu uyarmaktan sorumludur. Nöbetçiler grubun güvenliğini korur ve garanti eder. Alfa ve beta olarak geçen yaşlılar deneyimli kurtlardır. Avcılar avlanma ve grup için yemek sağlanacağı zaman alfa ve betaya yardım ederler. Son olarak bakıcılar gruptaki hasta, halsiz ve yaralı kurtları taşımakla sorumludurlar. Grup avlama işlemi, kurtların toplumsal etkileşimlerinin yanı sıra gri kurtların ilginç bir sosyal davranışdır

Gri Kurt Optimizasyon algoritması, gri kurtların yukarıda anlatıldığı gibi avcılık davranışını ve toplumsal davranışlarını esinlenerek önerilmiştir. GWO'nun ana bölümleri avı çevreleme, avlama ve avına saldırma aşamaları oluşturmaktadır (Mirjalili ve Lewis,2014).

5.1.Sosyal Hiyerarşi

GWO algoritmasında aday çözümler gri kurtların sürü içerisindeki ilişkileri göz önünde tutularak geliştirilir. Kurtların sosyal hiyerarşisini matematiksel olarak modellemek amacıyla GWO tasarlanırken en uygun çözümü alfa olarak değerlendirilir. Sonuç olarak, ikinci ve üçüncü en iyi çözümler sırasıyla beta ve deltadır. Aday çözümlerin geri kalanının omega (x) olduğu varsayılır. GWO algoritmasında avlanma (optimizasyon) alfa, beta ve delta tarafından yönlendirilir. Omega ise bu üç kurdu takip eder.

5.2. Avı Çevreleme

Gri kurtlar Eşitlik 5.1 – 5.2'ye göre avını rastgele çevreleyebilir.

$$D = |C \cdot X_p(t) - X(t)| \quad (5.1)$$

$$X(t + 1) = |X_p(t) - A \cdot D| \quad (5.2)$$

Burada t mevcut iterasyonu, A ve C katsayı vektörlerini, X_p avın konum vektörünü, X ise herhangi bir gri kurdun konumunu göstermektedir. A ve C değerleriyle Eşitlik 5.3- 5.4.'ye göre hesaplanır:

$$A = |2 \cdot \alpha \cdot r_1 - \alpha| \quad (5.3)$$

$$C = |2 \cdot \alpha \cdot r_2| \quad (5.4)$$

α (alfa kurt) 'nın bileşenleri tekrarlamalar sırasında doğrusal olarak 2'den 0'a düşürülür. r_1 ve r_2 $[0,1]$ arasında rastsal bir vektördür. r_1 ve r_2 rastsal vektörleri, kurtların arama uzayındaki noktalardan herhangi bir yere ulaşmasına izin verir. Böylelikle, gri kurt, rastgele konumda yer alan avın etrafında bulunan boşluk içerisindeki konumunu Eşitlik (5.1) ve (5.2)'ye göre düzenleyebilir. Aynı şekilde, iki ve üç boyutlu uzayı, n boyutlu bir arama uzayına genişletilebilir ve böylelikle gri kurtların, şimdiye kadar elde ettikleri en iyi çözüm etrafında hareket etmesi sağlanacaktır (Faris, vd.,2017).

5.3.Avlanma

Gri kurtlar avını tanıma ve kuşatma yeteneğine sahiptir. Av genellikle alfa tarafından yönlendirilir. Beta ve delta ara sıra avlanmaya katılabilir. Soyut bir arama uzayında optimum konum hakkında hiçbir bilgi yoktur. Gri kurtların avlanma davranışını matematiksel olarak modellemek için alfa (en iyi aday çözüm), beta ve deltanın avın potansiyel konumu hakkında daha iyi bilgiye sahiptirler. Bundan dolayı, elde edilen çözümlerden en iyi ilk üç çözüm kaydedilir. Diğer kurtların en iyi arama ajanlarının pozisyonları da bu kaydedilen çözüme göre güncellenir. Bu bağlamda Eşitlik 5.5 – 5.11 kullanılabilir (Jayakumar, vd.,2016).

$$D_\alpha = |C_1 \cdot X_\alpha - X| \quad (5.5)$$

$$D_\beta = |C_2 \cdot X_\beta - X| \quad (5.6)$$

$$D_\delta = |C_3 \cdot X_\delta - X| \quad (5.7)$$

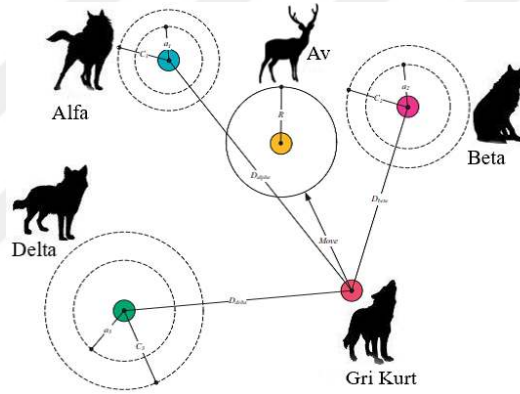
$$X_1 = |X_\alpha - A_1 \cdot D_\alpha| \quad (5.8)$$

$$X_2 = |X_\beta - A_2 \cdot D_\beta| \quad (5.9)$$

$$X_3 = |X_\delta - A_3 \cdot D_\delta| \quad (5.10)$$

$$X(t+1) = \frac{(X_1 + X_2 + X_3)}{3} \quad (5.11)$$

Eşitlik 5.5 – 5.11 ‘e göre D_α , D_β , D_δ sırasıyla alfa, beta, delta kurtlarının av ile arasında olan mesafeleri, X_α , X_β , X_δ alfa, beta ve delta kurtları için avın konumunu, X gri kurdun t. iterasyondaki konumunu, C_1, C_2, C_3 , A_1, A_2, A_3 alfa, beta ve delta kurtlarının katsayı vektörlerini, X_1, X_2, X_3 alfa, beta ve delta kurtları için deneme vektörlerini göstermektedir. Şekil 5.2’de gri kurtların avlanma stratejisi gösterilmiştir.

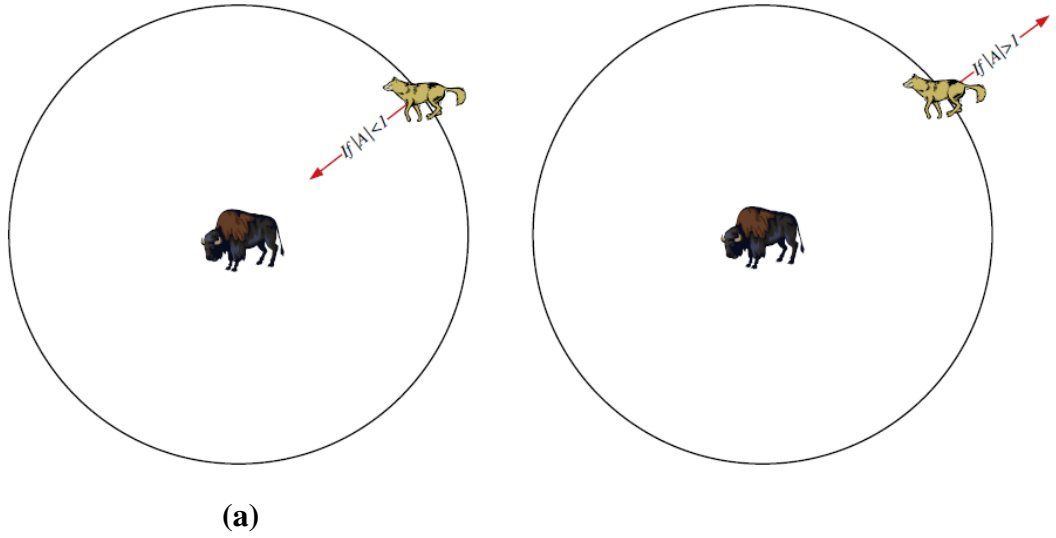


Şekil 5.2. Gri Kurtların Avlanma Stratejisi (Doğan ve Yüzgeç,2018)

5.4. Ava Saldırma

Bölüm 5.3’te bahsedildiği gibi gri kurtlar av hareket etmeyi bıraktığında ava saldırarak avlanmayı bitirir.

Bu adımda, a değeri azaltılır ve bundan dolayı α ’nın değişim aralığı azaltılır. A , $[-1,1]$ değer aralığında rastgele değerlere aldığında, arama ajanının bir sonraki konumu, şimdiki konumu ile avın konumu arasında rastgele bir yerde olacaktır (Koç, vd.,2018). Şekil 5.3 (a)’da gösterildiği gibi $|A| < 1$ olduğunda kurtları avına saldırmaya zorlar.



Şekil 5.3. Avını Ararken ve Avına Saldırırken Gri Kurtlar (Mirjalili ve Lewis,2014)

5.5.Arama

Gri kurtlar genellikle alfa, beta ve delta kurtlarının pozisyonlarına göre arama yaparlar. Avını aramak ve avlarına saldırmak için derhal bir araya gelmek koşuluyla birbirlerinden ayrılırlar. Dağılımın matematiksel modellenmesi için, $A > 1$ veya $A < 1$ rastgele değerlere sahip bir A değişkeni kullanılır. Kullanılan yöntem, aramayı önemli hale getirir ve GWO algoritmasının evrensel aramasına yardımcı olur (Koç, vd.,2018). Şekil 15(b)'de gösterildiği gibi $|A| > 1$ olduğunda gri kurtlar daha iyi avlar bulma umuduyla avdan sapmaya zorlanırlar. GWO algoritmasının sözde kodu Şekil 5.4'te verilmiştir.

```

Gri kurtların pozisyonunu başlat
Gri kurtların maliyet değerini hesapla
En iyi kurdu alfa kurdu olarak kaydet
İkinci en iyi kurdu beta kurdu olarak kaydet
Üçüncü en iyi kurdu delta kurdu olarak kaydet
while (iterasyon sayısı<maksimum iterasyon sayısı)
 $\alpha$ 'yı azalt
    for (her bir gri kurt için)
        Alfa,beta,delta kayşayı vektörlerini oluştur
        Mesafe vektörlerini hesapla
        Deneme vektörlerini hesapla
        Gri kurdun konumunu güncelle
    end for
    Güncellenen her gri kurtların maliyet değerini hesapla
    for (her gri kurt için)
        if (gri kurt<alfa kurt)
            Alfa kurdunu güncelle
        else if (gri kurt<beta kurt)
            Beta kurdunu güncelle
        else if (gri kurt<delta kurt)
            Delta kurdunu güncelle
        end if
    end for
    Seçilenlerin durumunu güncelle
    Yinelemeyi bir arttır
end while
Alfa kurdunu döndür

```

Şekil 5.4.GWO Algoritmasının Sözde Kodu

6. KARŞITLIK TABANLI GRİ KURT OPTİMİZASYON ALGORİTMASI

GWO algoritması popülasyon temelli sezgisel algoritmalar arasında en popüler olan algoritmalarından bir tanesidir. Bu yüzden de GWO algoritmasının performansını arttırmak için literatürde birçok çalışma mevcuttur. Bu çalışmalar aşağıdaki gibi kategorize edilmiştir.

Güncelleme Mekanizması: Bu alanda yapılan çalışmalardan bazıları şunlardır:

- Mittal ve arkadaşları, GWO'daki arama sürecini, doğrusal olarak değiştirmek yerine, üstel bir azalma fonksiyonunu kullanıp a 'nın değerini azaltarak geliştirme olasılığını incelediler (Mittal, vd.,2016).
- Koza, aralıksız doğrusal olmayan modülasyon indeksi kullanarak parametrenin doğrusal olmayan bir şekilde uyarlanmasını araştırmıştır (Koza,1992).
- Rodriguez ve arkadaşları, bazı parametrelerin dinamik uyarlamasını denemişlerdir (Rodriguez, vd.,2016).
- Dudani ve arkadaşları, kurtların mevcut nesildeki arama uzayında bireyin uygunluğuyla orantılı olan bir adım büyüklüğünün dahil edilmesine dayanan pozisyon güncelleme stratejisini benimsemiştir (Dudani, vd.,2016).
- Malik ve arkadaşları, bireylerin pozisyonlarını güncellemek için farklı yaklaşımlar benimsemişlerdir. En iyilerin basit ortalamasını kullanmak yerine seçilen üç tanesinin ağırlıklı ortalamasını kullanmayı tercih ettiler (Malik, vd.,2015).
- Rodriguez ve arkadaşları, Algoritmadaki omega kurtlarının pozisyonlarını güncellemek için üç farklı metot önerdiler. Bu üç yöntem ağırlıklı ortalama, uygunluğa dayalı ve bulanık mantığa dayalıdır (Rodriguez, vd., 2017).

Yeni Operatörler: Bu alanda yapılan çalışmalardan bazıları şunlardır:

- Kishor ve arkadaşları, iki farklı birey arasında basit bir çaprazlama operatörü içeren bir GWO versiyonu önermişlerdir. Çaprazlamanın amacı, gruptaki bireyler arasında bilgi paylaşımını kolaylaştırmaktır (Kishor, vd.,2016).
- Chandra ve arkadaşları, GWO'nun yukarıda bahsedilen sürümünü web servislerini ve uygun hizmet niteliği gereksinimlerini seçmek için geliştirdiler (Chandra, vd., 1994).
- Saremi ve arkadaşları, evrimsel popülasyon dinamiği (Evolutionary Population Dynamics-EPD) adı verilen bir operatör kullanarak GWO'yu geliştirdiler. EPD, popülasyondaki en kötü bireyleri elemek ve onları önde gelen (alfa,beta,delta)

bireylerin etrafında yeniden konumlandırmak için GWO'ya uygulandı (Saremi, vd., 2015).

➤ Zhang ve arkadaşları, Powell optimizasyon algoritmasını (Powell, 1997) yerel bir arama operatörü olarak GWO'ya entegre ettiler ve adına PGWO olarak adlandırdılar (Zhang, vd., 2015).

➤ Mahdad ve arkadaşları, kritik durumlarda güvenlik akıllı şebeke güç sistemi yönetimini çözmek için yerel bir arama olarak GWO'yu PSA (Pattern Search Algorithm) ile birleştirdi (Mahdad, vd., 2015).

➤ Zhou ve arkadaşları, GWO'yu, küçük su jeneratörü kümelenmesinin eşdeğer modelinin parametrelerini ayarlayarak kaotik yerel arama ile birleştirmeyi önerdiler (Zhou, vd., 2015).

➤ Rodrı'guez, Castillo ve arkadaşları, bulanık bir hiyerarşik operatör sunmuşlardır (Rodriguez, vd., 2017).

Farklı Kodlama Şemaları: Bu alanda yapılan çalışmalardan bazıları şunlardır:

➤ Lou ve arkadaşları, tipik gerçek değerli olanların yerine karmaşık değerli bir kodlama yöntemi kullandılar. Bu kodlamada, bireydeki genlerinin iki ana bölümü vardır: hayali bölüm ve gerçek bölüm. Yazarlar, bu sunumun aday çözümlerin bilgi kapasitesini genişletebileceğini ve popülasyonun çeşitliliğini artırabileceğini savunuyorlar (Lou, vd., 2015).

Popülasyon Yapısı ve Hiyerarşisi: Bu alanda yapılan çalışmalardan bazıları şunlardır:

➤ Yang ve arkadaşları, burada farklı liderlik hiyerarşisi olan bir farklı bir GWO önerildi. Popülasyondaki değişkenler iki alt popülasyona bölünmüştür. Bunlardan ilki avlanma grubu ikincisi izci grubudur. İzci grubunun görevi geniş bir arama yapmaktır. Avlanma grubunun göreviyse derin bir sömürü gerçekleştirmektir (Yang, vd., 2016).

Yukarıda GWO algoritması için literatürde yapılan çalışmaları incelendi. Bu çalışmada, GWO algoritmasının başarısının arttırmak için karşıtlık tabanlı öğrenme yöntemi temel alınmıştır. Bununla birlikte genetik algoritmanın temel adımlarından biri olan mutasyon operatörü GWO algoritmasının arama uzayındaki başarısını (en iyi çözüm) arttırmak için eklenmiştir. Yapılan tüm geliştirmeler ve eklenen yenilikler alt başlıklar halinde aşağıda verilmektedir.

6.1.Karşıtlık Tabanlı Öğrenme

Karşıtlık tabanlı öğrenme (Opposition Based Learning- OBL) fikri Tizhoosh tarafından ortaya atılmıştır. Meta-sezgisel ve optimizasyona dayanan karşıt bir çift aday çözümlerin arasındaki ilişki olarak tanımlanır. (Tizhoosh,2005; Mahdavi, vd.,2018).

Hem kombinasyonel hem de sürekli optimizasyonlarda her aday çözüm noktası için tanımlanmış bir karşıt çözüm noktası vardır.

Birçok durumda öğrenme rastgele bir noktada başlar. Sıfırdan başlanır ve mevcut bir çözüme doğru ilerlenir. Rastgele nokta, optimal çözümden uzak değilse, hızlı bir yakınsamaya neden olabilir. Bununla birlikte, mevcut çözümden çok uzakta olan rastgele bir nokta ile başlanırsa, o zaman arama veya optimizasyon süreci zaman alacaktır. Olasılık kuramına göre, mevcut çözüm aranırken x noktası çözüme uzak ise, x'in karşıtı olan noktanın çözüme daha yakın olabileceği göz önüne alınarak, x'in karşıt noktası hesaplanarak çözüm araması yapılabilir (Rojas-Moreles, vd.,2017; Mahdavi, vd.,2018).

Karşıtlık tabanlı öğrenmede birinci aşama karşıtlık tabanlı başlangıç popülasyonunun belirlenmesi, bir sonraki aşama ise karşıtlık temelli jenerasyon atlama işlemidir. Optimizasyon süreci başlatılırken aday çözüm noktaları rastgele olarak belirlenir. Karşıtlık tabanlı öğrenme işlemiyle aday çözüm noktalarının rastgele bir şekilde belirlenmesine ek olarak, rastgele belirlenen aday çözüm noktalarının karşıtlarının bulunması ile optimizasyon başlatılır. Böylelikle optimizasyon sürecinin başlangıcında probleminin çözümünde avantajlı aday çözüm noktalarıyla algoritmanın çözümüne başlanır. Karşıt noktalar ($\tilde{x}_{i,1}$) Eşitlik 6.1'e göre bulunur.

$$\tilde{x}_{i,1} = a + b - x_{i,1} \rightarrow i = 1,2, \dots PS \quad (6.1)$$

Burada, a alt sınır değerini, b üst sınır değerini, PS popülasyonun boyutunu, $x_{i,1}$ başlangıçta rastgele olarak belirlenen aday çözümleri göstermektedir (Mahdavi, vd.,2018). Optimizasyon işlemi esnasında, karşıt popülasyon daha önceden belirlenmiş bir atlama olasılığı değerine göre hesaplanır. Daha iyi olan adaylar mevcut ve karşıt popülasyonun içinden seçilir. Karşıtlık tabanlı jenerasyon atlama, atlama hızına (J_r) ve Eşitlik 6.2'de verilen $[0,1]$ aralığında değişen rastgele bir sayıya bağlıdır:

$x_{i,g+1}$: Karşıt Çözüm Noktası

J_r : Atlama Değeri

$rnd_1 \in [0,1]$

$$x_{i,g+1} = \begin{cases} \check{x}_{i,g} = a + b - x_{i,g}, & \text{if}(rnd_1 < J_r) \\ x_{i,g} , & \text{diğer} \end{cases} \quad (6.2)$$

6.2.Mutasyon

Mutasyon operatörü GA'nın temel aşamalarındandır. Yapay genetik sistemlerde mutasyon operatörü, bir daha elde edilemeyebilir iyi bir çözümün kaybına karşı koruma sağlamaktadır (Goldberg, 1989; Taşkın, vd.,2002). Mutasyon GA'lardaki operasyonda karar verici olarak ikinci seviyede rol oynar (İşçi ve Korukoğlu,2003). Mutasyonun genel amacı genetik çeşitliliği sağlamak ve korumaktır (Braysy,2001). Bu operatör tüm genlere uygulanmayıp, üretilen genlerin belirli bir yüzdesine uygulanmaktadır. Bu yüzdeye mutasyon oranı veya mutasyon olasılığı adı verilir. Mutasyon olasılığı genellikle (0.01 gibi) düşük tutulmaktadır. Bu nedenle mutasyon etkileri kromozomlarda az görülmektedir (Çolak,2010).

Mutasyon operatörüyle aday çözümler değiştirilir. Bu değişim genellikle popülasyonun %1-5'lik bir kısmına denk gelir. Mutasyon, popülasyonun içinde çeşitlilik sağlar (Özsağlam ve Cunkaş,2008). Problem sonucunun yerel optimum noktalara takıldığı durumlarda bir başka optimum çözüme sıçrayabilmesini sağlar. Mutasyon değerinin çok küçük seçilmesi çözümün optimum sonuca ulaşılmasını engelleyebilir, mutasyon değerlerinin çok büyük seçilmesi de salınma (osilasyon) neden olur (Kahraman ve Özdağlar,2004). Mutasyon operatörünün kaba kod yapısı aşağıdaki gibidir.

mutasyon değeri belirle

rastgele değer seç

if (rastgele değer < mutasyon değeri)

adaya çözüme mutasyon işlemi uygula

pozisyonun güncelle

end

6.3.Sınır Değeri Kontrolü

Aday çözüm noktaları belirli bir arama uzayı içerisinde problem çözümünü gerçekleştirirler. Aday çözümler arama uzayından ne kadar uzaklaşırlarsa problemin çözümüne ulaşmak da aynı derecede güçleşir. İşte bu yüzden sınır değeri kontrolünün yapılması problemin çözümünden uzaklaşılmasını engeller.

Aday çözüm noktaları ya da arama ajanları (search agents), optimizasyon problemindeki arama uzayının sınırları dışına çıkarsa, OppGWO algoritmasında Eşitlik 6.3 ve 6.4'teki işlemler uygulanır.

$$\text{if } (X > U_b) \rightarrow X = U_b - 0.25(U_b - L_b) * \text{rnd}_1 \quad (6.3)$$

$$\text{if } (X < L_b) \rightarrow X = L_b + 0.25(U_b - L_b) * \text{rnd}_2 \quad (6.4)$$

Burada X aday çözüm noktasını (arama ajanı), U_b (*Upper Boundary*) ve L_b (*Lower Boundary*) de alt ve üst sınır değerlerini temsil etmektedir. Aday çözüm noktaları arama uzayının sınırlarından dışarı çıktığı takdirde bu aday çözüm noktaları sınırın 0.25 oranla arama uzayının içerisine gönderilir.

Şekil 6.1'de OppGWO algoritmasının sözde kodu verilmiştir. Renkli olarak belirtilen yerler orijinal algoritmaya ek olarak yapılan yeniliklerdir.

```

Sıçrama değerini (jump_rate) belirle
Gri kurtların pozisyonlarını başlat
Gri kurtların pozisyonlarının karşıt noktalarını bul
Gri kurtların maliyet değerlerini hesapla
Karşıt noktaları bulunan gri kurtların maliyet değerlerini hesapla

En iyi gri kurtu alfa kurt olarak kaydet
İkinci en iyi gri kurtu beta kurt olarak kaydet
Üçüncü en iyi gri kurtu delta kurt olarak kaydet

while (iterasyon sayısı < maksimum iterasyon sayısı)
    α 'yı azalt
    for (her gri kurt için)
        Alfa, beta, delta katsayı vektörlerini oluştur
        Mesafe vektörlerini hesapla
        Deneme vektörlerini hesapla
        Gri kurdun konumunu güncelle
    end for

    Güncellenen gri kurtların maliyet değerlerini hesapla
    for (her gri kurt için)
        if (gri kurt < alfa kurt)
            alfa kurdunu güncelle
        else if (gri kurt < beta kurt)
            beta kurdunu güncelle
        else if (gri kurt < delta kurt)
            delta kurdunu güncelle
        end if

        if (rastgele değer < mutasyon değeri)
            aday çözüme mutasyon işlemi uygula
            pozisyonun güncelle
        end

        Gri kurdun pozisyonu alt sınıra veya üst sınıra yakınsa
        belirlenen oranda arama uzayının içine yönlendir
    end for

    if (rastgele değer < sıçrama değeri)
        gri kurtların karşıt noktalarını bul
        karşıt pozisyonları belirle ve maliyet değerlerini hesapla
    end

    Seçilenlerin durumunu güncelle
    Yinelemeyi bir artır
end while
alfa kurdunu döndür

```

Şekil 6.1. OppGWO Algoritmasının Sözde Kodu (Karakaş ve Yüzgeç,2019)

7.DENEYSEL SONUÇLAR

Çalışmanın bu bölümde geliştirilen OppGWO algoritmasının analizleri yapılmıştır. Analizlerde GWO ve OppGWO algoritmaları eş zamanlı olarak CEC'14 (Liang, vd.,2013) fonksiyonları test edilmiştir. Ardından UCI'den (UCI,2019) elde edilen birbirinden farklı özellikteki on sekiz adet veri seti (Çizelge 7.5) üzerinden özellik seçim sonuçları test edilmiştir. Bu sonuçlar aşağıda çizelgeler ve grafikler halinde sunulmuştur.

7.1.Önerilen OppGWO Algoritmasının Analiz Sonuçları

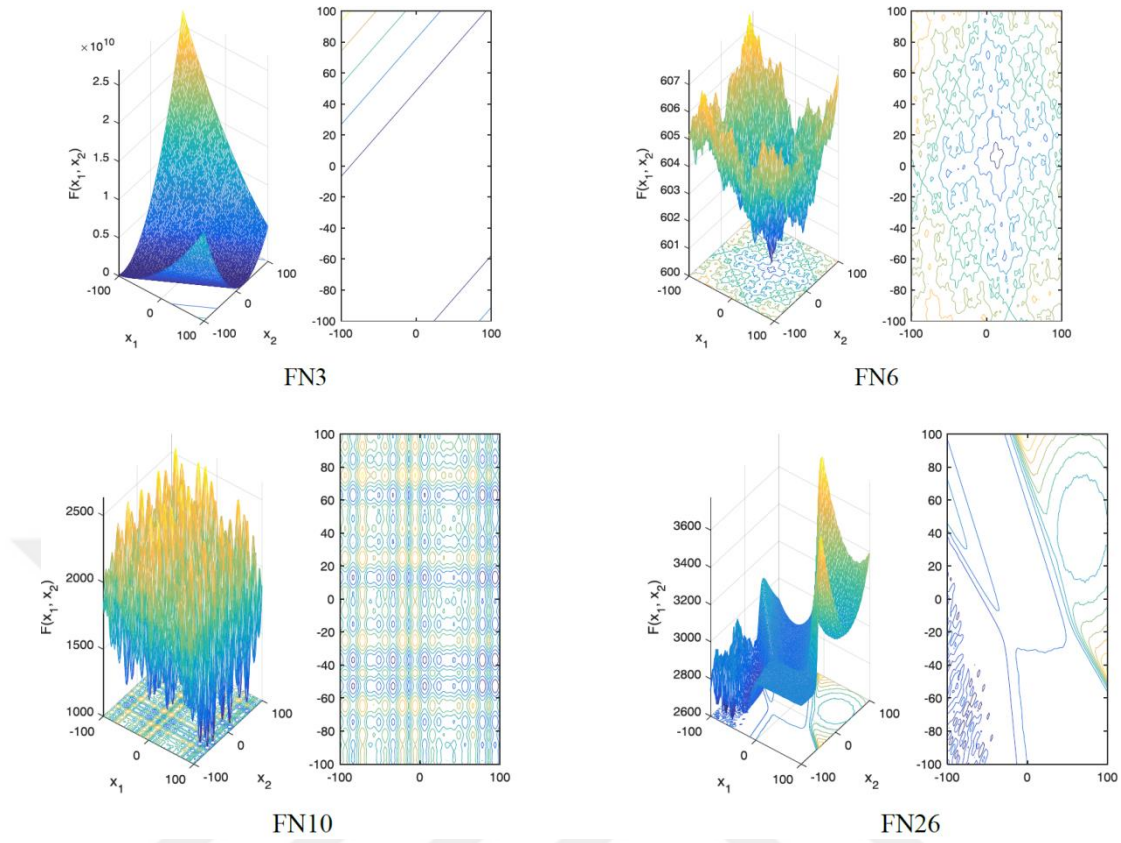
CEC 2014'te 30 farklı optimizasyon problemi yer almaktadır. Bu optimizasyon problemleri 4 farklı grupta toplanır. Her fonksiyonun çözümü, Eşitlik 7.1'de olduğu gibi fonksiyon numarası (FN) ile 100 çarpım sonucu elde edilir. Hata değeri ise Eşitlik 7.2'de olduğu gibi istenen çözüm ile bulunan çözümün farkının boyut sayısı ile çarpımı ile bulunur (Liang, vd.,2013).

$$F_i(x^*) = FN * 100 \quad (7.1)$$

$$E = F_i(x) - F_i(x^*) * \text{Max_Fes} \quad (7.2)$$

Eşitlik 7.1'de FN fonksiyon numarasını, Eşitlik 7.2'de Max_Fes ise boyutu temsil etmektedir. Hata değeri hangi boyut için hesaplanacaksa Max_Fes o değeri alır (10D=100000,30D=300000,50D=500000).

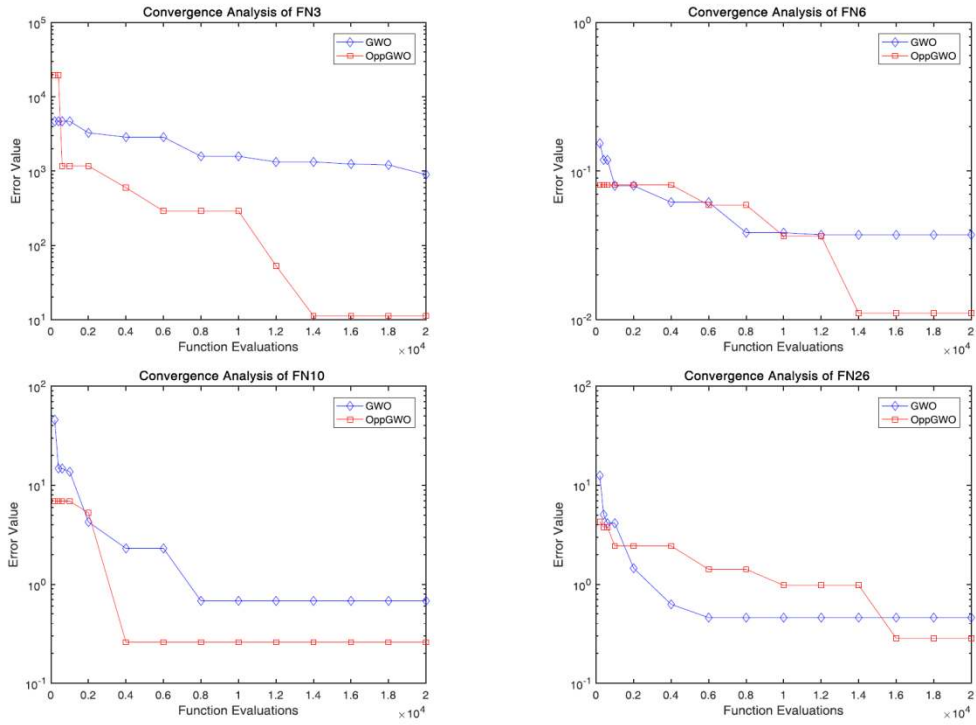
Bu çalışmada, CEC 2014'ün farklı özelliklere sahip dört fonksiyonu kullanılmıştır (Liang, vd.,2013). İlk fonksiyon FN3, tek modlu fonksiyonlardan biridir. İkinci fonksiyon FN6, Weierstrass'in basit çok modlu fonksiyonudur. Üçüncü fonksiyon FN10, Schwefel'in basit çok modlu fonksiyonudur. Son olarak, dördüncü fonksiyon FN26, kompozisyon fonksiyonların dördüncüsüdür (N = 5). Analizde kullanılan dört fonksiyon, Şekil 7.1'de gösterilmektedir.



Şekil 7.1. OppGWO'nun Analizinde Kullanılan Test Fonksiyonlarının Üç Boyutlu İz Düşüm Görüntüleri

7.1.1.Yakınsama Analizleri

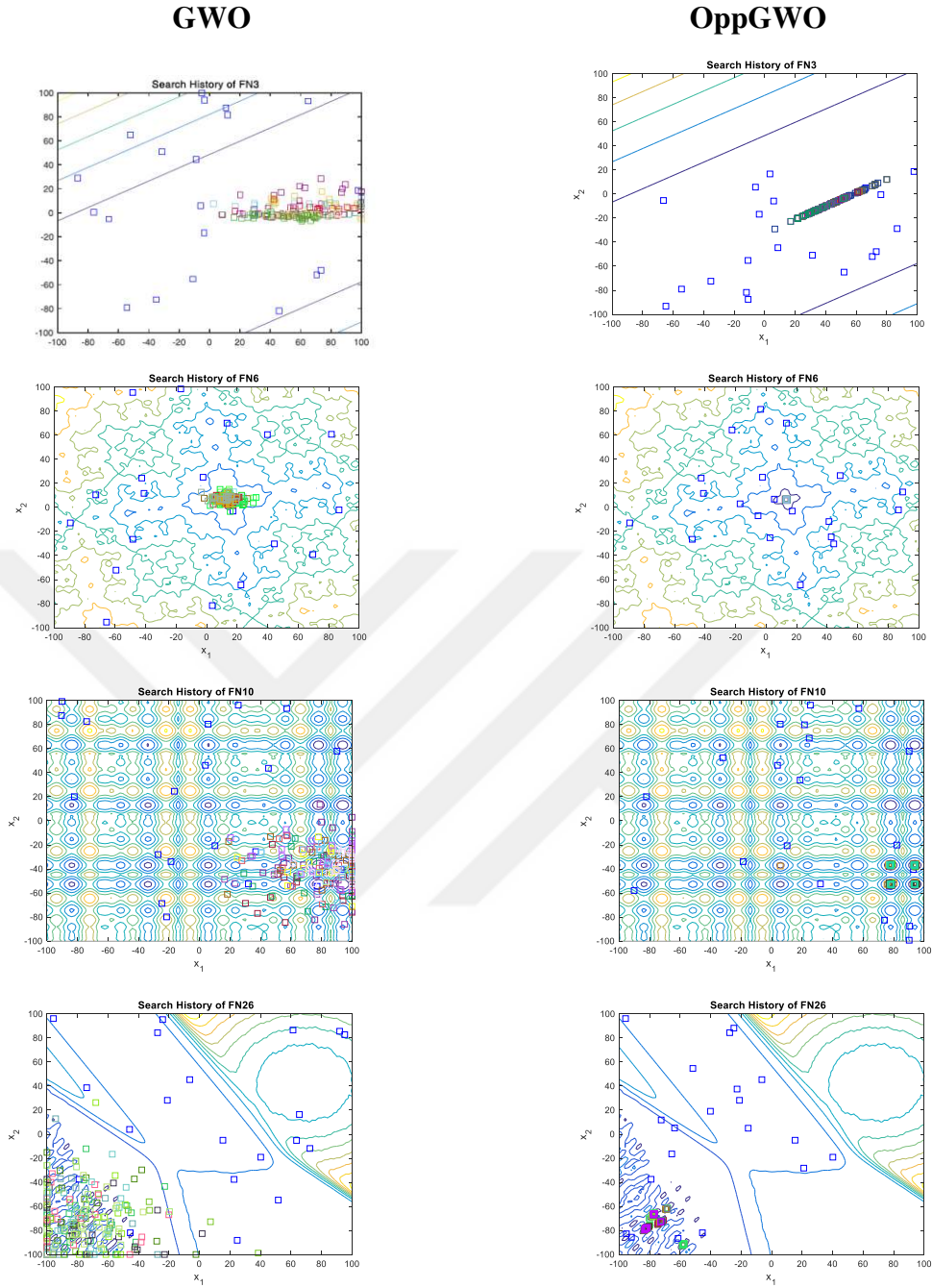
Bu analizde, GWO ve OppGWO'nun hata değerleri karşılaştırılmıştır. Dört fonksiyonun her biri için aynı başlangıç parametreleri kullanılmıştır. Bu fonksiyonlar için yakınsama analizleri Şekil 7.2'de gösterilmiştir. Grafıklara bakıldığı zaman OppGWO algoritması dört fonksiyon içinde GWO algoritmasından daha başarılı sonuçlar vermiştir.



Şekil 7.2. OppGWO Algoritmasının Yakınsama Sonuçları

7.1.2. Arama Geçmiş Analizleri

Bu analizde OppGWO algoritmasının belirlenen dört fonksiyon üzerinde çözüm konumlarının yineleme adımları verilmiştir. Şekil 7.3.'de ilk sütun GWO algoritmasına ait analizleri, ikincisi ise OppGWO algoritmasına ait analizleri göstermektedir. Grafiklerde renkli verilen kareler aday çözümlerinin hangi bölgelerde olduğunu göstermektedir. Grafiklere bakıldığı zaman aday çözümlerinin genellikle tek bir noktada yoğunlaştıkları görülmektedir. Şekil 7.3'te bulunan üçüncü grafiğe bakıldığı zaman 10 numaralı fonksiyonun arama geçmiş analizi verilmektedir. Grafikte ortada bulunan çukur alan avların yoğunlukta bulunduğu bölgeyi temsil etmektedir. Aday çözümlerin bulunduğu alan bakıldığı zaman ise çukur alanda yoğunlaştıkları gözlemlenmektedir.

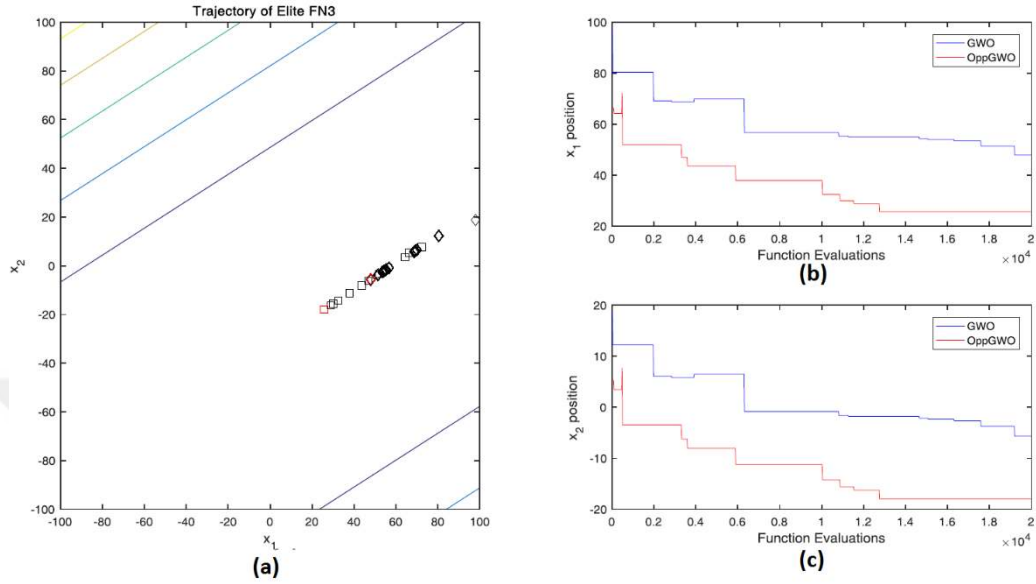


Şekil 7.3.OppGWO Algoritmasının Arama Geçmişi Analizleri

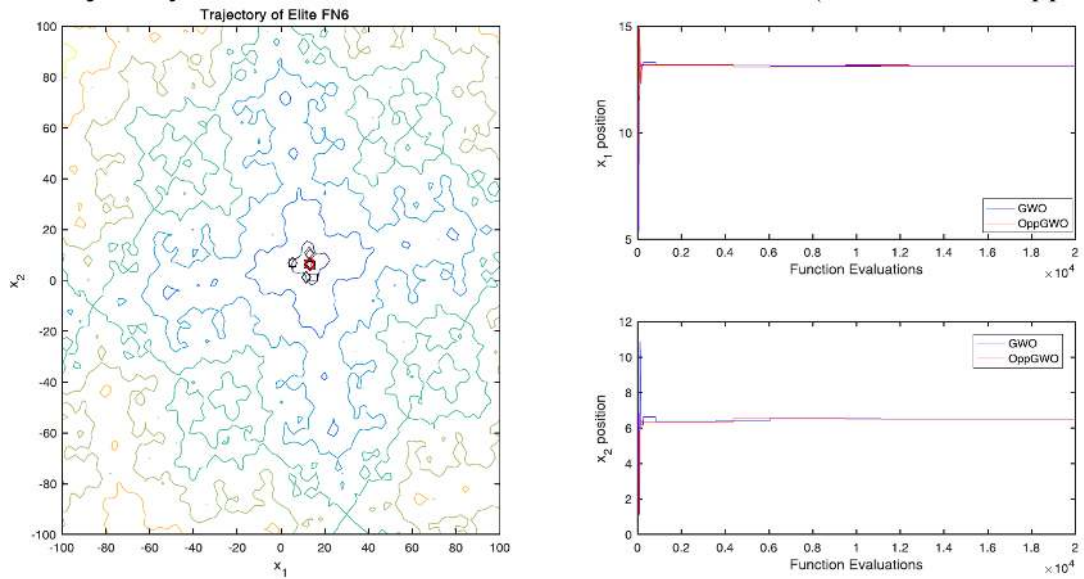
7.1.3. Yörünge Analizleri

Yörünge analizi, elit (seçkin) bireylerin arama uzayındaki geçmişidir. Bu analizde seçilen kıyaslama fonksiyonları için, Şekil 7.4-7.7, belirli yinelemelerde GWO ve OppGWO'nun elit çözümlerinin pozisyonlarını göstermektedir. Şekil 7.4.b ve

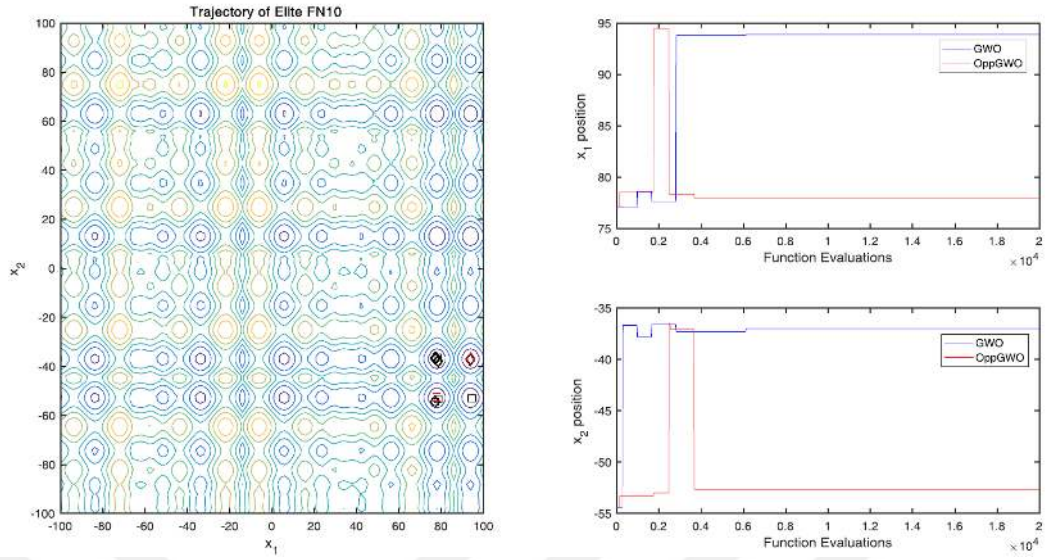
7.4.c'de yer alan grafikler ise iki boyutlu problemin çözümünde en iyi çözümün pozisyon değişimi göstermektedir.



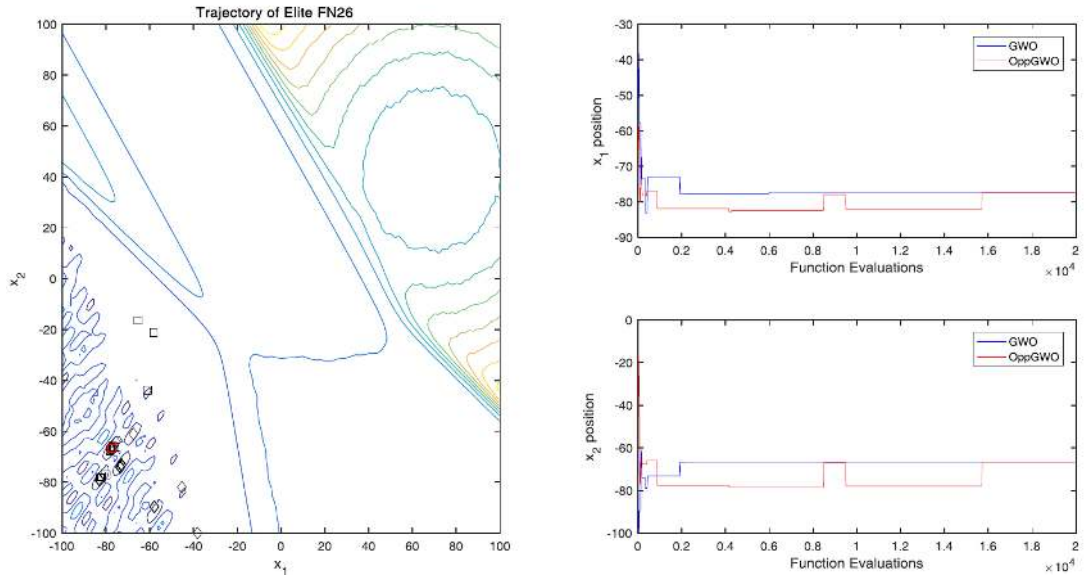
Şekil 7.4. FN3 Optimizasyon Fonsiyonları için Şeçkin Yörünge Analizleri ($\diamond$: GWO, $\square$: OppGWO)



Şekil 7.5. FN6 Optimizasyon Fonsiyonları için Şeçkin Yörünge Analizleri ($\diamond$: GWO, $\square$: OppGWO)



Şekil 7.6. FN10 Optimizasyon Fonsiyonları için Şeçkin Yörünge Analizleri
($\diamond$: GWO, $\square$: OppGWO)

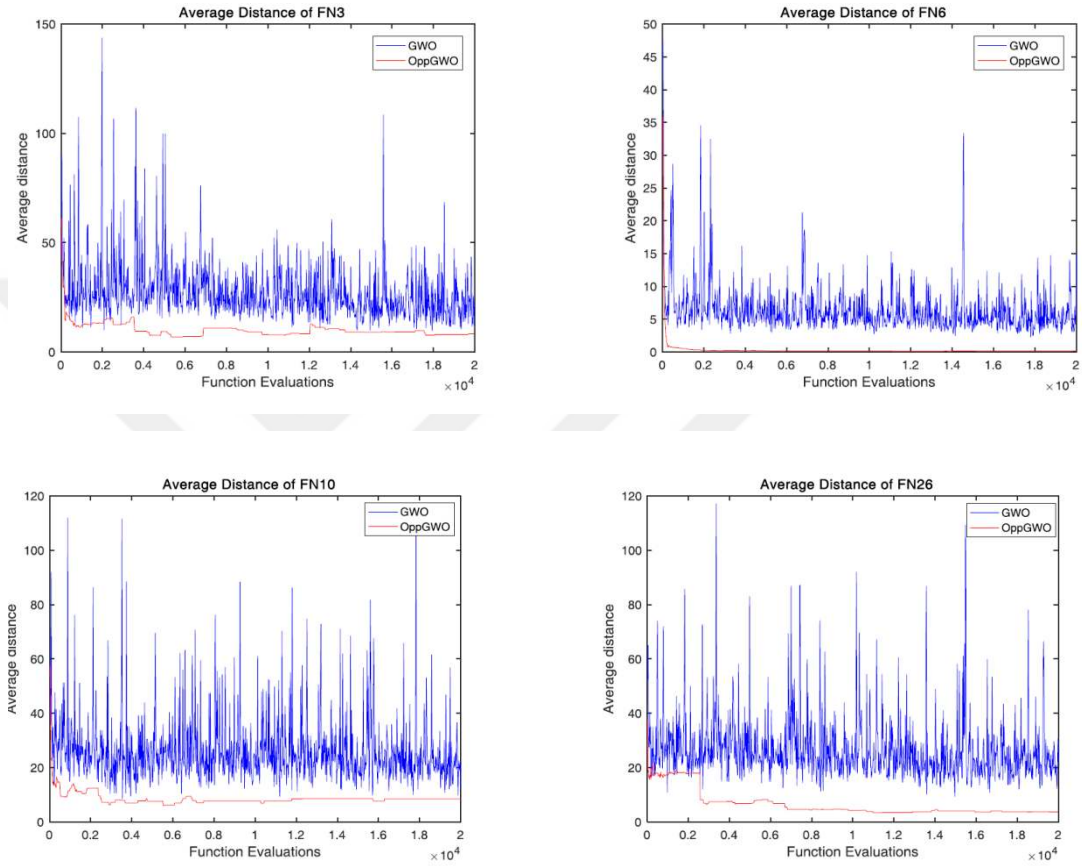


Şekil 7.7. FN26 Optimizasyon Fonsiyonları için Şeçkin Yörünge Analizleri
($\diamond$: GWO, $\square$: OppGWO)

7.1.4. Ortalama Mesafe Analizleri

Ortalama mesafe analizi, ilk arama ajanının birinci pozisyon değerinin (x_1) diğer arama ajanlarına olan mesafelerinin ortalaması olarak tanımlanır. Bu analizde GWO ve OppGWO algoritmaları belirlenen dört fonksiyon için eş zamanlı olarak test edilmiştir.

Her iki algoritma için de arama uzayında aday çözümlerin birbirleri arasındaki ortalama mesafeleri grafikler halinde gösterilmiştir. Şekil 7.8’de görüldüğü gibi OppGWO algoritması sıfıra daha hızlı ve salınımlı olmadan yaklaşırken GWO algoritmasının içerisinde seçim mekanizması bulunduğundan salınımlarda bulunur.



Şekil 7.8.Aday Çözümleri Arasındaki Ortalama Mesafe Analizi

7.2.Optimizasyon Test Fonksiyonu Sonuçları

GWO ve önerilen OppGWO algoritmalarının performanslarını değerlendirmek için 10,30,50-boyutlu 30 adet CEC-2014 test (Ek-1) fonksiyonunda eş zamanlı birbirinden bağımsız 30 deneme ile çalıştırılarak test edilmiştir. Yapılan analizler, 2.4 GHz 4 çekirdek 8. Nesil intel core i5 işlemciye sahip, MacOS Mojave işletim sisteminde Matlab üzerinde geliştirilmiştir. Sonuçları aşağıda tablolar halinde sunulmuştur.

OppGWO ve GWO algoritmaları aynı fonksiyonlar üstünde eş zamanlı olarak test edilerek elde edilen sonuçlar karşılaştırılmalı olarak çizelgeler halinde verilmiştir.

Çizelge 7.1'de GWO ve OppGWO algoritmalarının 10-boyutlu 30 adet CEC'14 benchmark fonksiyonunda birbirinden bağımsız 30 (otuz) tekrarlı çalıştırma sonucunda elde edilen sonuçlar çizelgede verilmiştir. Çizelge 7.1'de verilen sonuçlara göre, OppGWO algoritmasının GWO algoritmasına kıyasla daha başarılı sonuçlar (GWO: 51/150, OppGWO: 99/150) vermiştir. Ayrıca 11 fonksiyonun bütün değerleri için OppGWO algoritması daha başarılı sonuçlar vermiştir.

Çizelge 7.1. GWO ve OppGWO için 10- Boyutlu CEC'14 optmizasyon sonuçları

FN	10D									
	GWO					OppGWO				
	En İyi	En Kötü	Medyan	Ortalama	S.Sapma	En İyi	En Kötü	Medyan	Ortalama	S.Sapma
1	4.190e+05	1.604e+07	3.365e+06	3.871e+06	2.953e+06	3.788e+05	6.999e+06	1.202e+06	1.706e+06	1.325e+06
2	2.025e+07	1.286e+08	7.063e+08	7.347e+08	2.192e+08	2.490e+07	1.178e+08	5.944e+07	5.899e+07	1.974e+07
3	5.511e+02	9.272e+03	1.843e+03	3.062e+03	2.425e+03	1.479e+02	1.439e+03	6.893e+02	7.079e+02	2.965e+02
4	1.066e+01	4.015e+01	3.633e+01	3.370e+01	7.855e+00	9.560e+00	4.615e+01	4.239e+01	3.657e+01	1.189e+01
5	2.018e+01	2.050e+01	2.035e+01	2.035e+01	7.461e-02	9.261e+00	2.052e+01	2.036e+01	1.977e+01	2.423e+00
6	2.288e+00	6.234e+00	3.404e+00	3.547e+00	8.307e-01	2.151e+00	5.288e+00	3.117e+00	3.205e+00	6.313e-01
7	1.743e+00	4.741e+00	3.064e+00	3.087e+00	5.999e-01	1.783e+00	3.278e+00	2.339e+00	2.372e+00	3.075e-01
8	1.193e+01	2.718e+01	1.785e+01	1.825e+01	3.472e+00	1.285e+01	3.174e+01	1.894e+01	1.892e+01	3.597e+00
9	1.531e+01	3.243e+01	2.345e+01	2.317e+01	4.031e+00	1.243e+01	3.301e+01	2.416e+01	2.471e+01	4.260e+00
10	1.189e+02	7.243e+02	3.598e+02	4.007e+02	1.446e+02	2.043e+02	9.643e+02	4.890e+02	5.361e+02	2.198e+02
11	1.495e+02	1.143e+03	6.271e+02	6.377e+02	2.449e+02	2.567e+02	1.379e+03	7.653e+02	7.815e+02	2.667e+02
12	5.016e-01	1.565e+00	1.087e+00	1.066e+00	2.500e-01	7.688e-01	1.501e+00	1.186e+00	1.177e+00	1.996e-01
13	1.878e-01	4.696e-01	3.267e-01	3.163e-01	6.775e-02	1.380e-01	3.610e-01	2.707e-01	2.624e-01	5.214e-02
14	1.438e-01	9.499e-01	2.634e-01	3.194e-01	1.779e-01	1.252e-01	7.779e-01	1.993e-01	2.296e-01	1.218e-01
15	2.098e+00	5.651e+00	3.355e+00	3.425e+00	6.817e-01	1.815e+00	4.079e+00	3.127e+00	3.139e+00	5.379e-01
16	1.761e+00	3.245e+00	2.633e+00	2.663e+00	3.319e-01	1.095e+00	3.228e+00	2.387e+00	2.388e+00	4.217e-01
17	2.025e+03	3.467e+05	5.016e+03	1.406e+04	4.795e+04	1.275e+03	1.820e+04	4.591e+03	5.404e+03	3.429e+03
18	4.929e+02	2.737e+04	6.904e+03	8.576e+03	6.213e+03	5.285e+02	2.245e+04	4.617e+03	6.619e+03	5.431e+03
19	2.243e+00	5.179e+00	3.173e+00	3.290e+00	7.043e-01	1.967e+00	4.915e+00	2.849e+00	3.038e+00	6.319e-01
20	5.760e+01	1.189e+04	2.547e+02	2.322e+03	3.215e+03	6.673e+01	3.903e+02	1.575e+02	1.636e+02	5.835e+01
21	9.317e+02	1.280e+04	5.241e+03	6.113e+03	4.185e+03	7.073e+02	9.631e+03	2.100e+03	3.046e+03	2.189e+03
22	2.717e+01	1.690e+02	4.278e+01	5.414e+01	3.696e+01	2.820e+01	1.649e+02	3.651e+01	4.582e+01	2.913e+01
23	3.300e+02	3.398e+02	3.312e+02	3.321e+02	2.194e+00	3.298e+02	3.312e+02	3.305e+02	3.305e+02	3.054e-01
24	1.204e+02	1.411e+02	1.298e+02	1.294e+02	4.821e+00	1.215e+02	1.531e+02	1.315e+02	1.311e+02	5.890e+00
25	1.367e+02	2.033e+02	2.010e+02	1.915e+02	1.921e+01	1.320e+02	2.022e+02	1.838e+02	1.731e+02	2.837e+01
26	1.002e+02	1.004e+02	1.003e+02	1.003e+02	4.825e-02	1.001e+02	1.003e+02	1.002e+02	1.002e+02	4.340e-02
27	3.821e+00	4.117e+02	3.938e+02	3.412e+02	1.354e+02	3.697e+00	4.017e+02	3.807e+02	2.976e+02	1.643e+02
28	3.937e+02	5.627e+02	4.381e+02	4.486e+02	5.118e+01	3.091e+02	6.603e+02	4.977e+02	4.948e+02	5.966e+01
29	3.412e+02	2.131e+06	8.403e+02	2.302e+05	6.364e+05	3.376e+02	4.875e+03	8.804e+02	1.184e+03	8.119e+02
30	5.048e+02	2.448e+03	9.635e+02	1.064e+03	4.502e+02	7.296e+02	2.326e+03	1.277e+03	1.377e+03	3.805e+02

Çizelge 7.2'de GWO ve OppGWO algoritmalarının aynı tekrar sayısı ile 30-boyut için elde edilen sonuçlar çizelgede verilmiştir. Bu çizelgede verilen sonuçlara göre, OppGWO algoritmasının GWO algoritmasına kıyasla daha başarılı sonuçlar (GWO: 45/150, OppGWO: 105/150) vermiştir. Ayrıca 12 fonksiyonun bütün değerleri için OppGWO algoritması daha başarılı sonuçlar vermiştir.

Çizelge 7.2. GWO ve OppGWO için 30- Boyutlu CEC'14 optmizasyon sonuçları

FN	30D									
	GWO					OppGWO				
	En İyi	En Kötü	Medyan	Ortalama	S.Sapma	En İyi	En Kötü	Medyan	Ortalama	S.Sapma
1	2.140e+07	1.076e+08	5.100e+07	5.616e+07	2.085e+07	2.881e+07	8.917e+07	4.923e+07	5.089e+07	1.332e+07
2	1.824e+09	6.097e+09	3.094e+09	3.163e+09	6.408e+08	1.466e+09	2.867e+09	2.245e+09	2.233e+09	3.326e+08
3	8.630e+03	3.156e+04	2.087e+04	2.091e+04	5.699e+03	3.879e+03	1.020e+04	6.620e+03	6.922e+03	1.407e+03
4	2.071e+02	3.631e+02	2.634e+02	2.646e+02	3.121e+01	2.038e+02	3.805e+02	2.813e+02	2.805e+02	3.120e+01
5	2.085e+01	2.102e+01	2.096e+01	2.095e+01	4.166e-02	2.078e+01	2.102e+01	2.094e+01	2.094e+01	4.967e-02
6	1.793e+01	2.521e+01	2.198e+01	2.174e+01	1.826e+00	1.607e+01	2.282e+01	1.933e+01	1.929e+01	1.765e+00
7	2.111e+01	5.719e+01	2.918e+01	3.010e+01	6.540e+00	1.543e+01	2.575e+01	2.010e+01	2.022e+01	2.185e+00
8	1.123e+02	1.699e+02	1.429e+02	1.419e+02	1.434e+01	1.214e+02	1.862e+02	1.589e+02	1.574e+02	1.414e+01
9	1.643e+02	2.346e+02	1.951e+02	1.970e+02	1.517e+01	1.559e+02	2.191e+02	1.915e+02	1.902e+02	1.363e+01
10	2.767e+03	5.234e+03	3.622e+03	3.768e+03	5.570e+02	3.326e+03	6.133e+03	4.885e+03	4.742e+03	7.242e+02
11	3.922e+03	6.769e+03	5.646e+03	5.614e+03	6.490e+02	5.330e+03	7.093e+03	6.511e+03	6.452e+03	4.034e+02
12	1.864e+00	2.992e+00	2.422e+00	2.402e+00	2.688e-01	1.970e+00	2.947e+00	2.523e+00	2.495e+00	2.347e-01
13	5.034e-01	9.662e-01	7.011e-01	6.997e-01	9.421e-02	4.149e-01	7.092e-01	5.885e-01	5.860e-01	6.951e-02
14	1.865e+00	1.639e+01	6.867e+00	7.015e+00	3.136e+00	4.246e-01	5.499e+00	1.557e+00	1.848e+00	1.154e+00
15	3.018e+01	1.440e+02	3.935e+01	4.361e+01	1.664e+01	2.432e+01	4.049e+01	3.048e+01	3.063e+01	3.379e+00
16	1.093e+01	1.259e+01	1.176e+01	1.172e+01	4.030e-01	1.097e+01	1.236e+01	1.174e+01	1.173e+01	2.929e-01
17	2.969e+05	6.869e+06	1.184e+06	1.694e+06	1.402e+06	4.511e+05	2.964e+06	1.260e+06	1.306e+06	5.074e+05
18	6.306e+06	5.495e+07	2.371e+07	2.633e+07	1.216e+07	5.672e+06	4.440e+07	1.574e+07	1.871e+07	9.998e+06
19	2.048e+01	8.519e+01	3.046e+01	3.241e+01	1.075e+01	2.126e+01	9.012e+01	2.799e+01	2.975e+01	9.352e+00
20	1.384e+03	2.332e+04	8.273e+03	8.691e+03	4.620e+03	5.661e+02	1.963e+03	1.246e+03	1.267e+03	3.633e+02
21	1.432e+05	1.333e+06	3.831e+05	5.355e+05	3.556e+05	1.125e+05	1.102e+06	3.321e+05	3.830e+05	2.087e+05
22	2.034e+02	6.910e+02	3.743e+02	3.855e+02	1.153e+02	2.191e+02	6.614e+02	3.423e+02	3.865e+02	1.109e+02
23	3.244e+02	3.414e+02	3.296e+02	3.300e+02	3.951e+00	3.215e+02	3.324e+02	3.251e+02	3.255e+02	2.264e+00
24	2.000e+02	2.000e+02	2.000e+02	2.000e+02	5.483e-04	2.000e+02	2.000e+02	2.000e+02	2.000e+02	2.937e-03
25	2.065e+02	2.177e+02	2.115e+02	2.114e+02	2.618e+00	2.080e+02	2.166e+02	2.119e+02	2.116e+02	1.884e+00
26	1.004e+02	2.008e+02	1.006e+02	1.065e+02	2.373e+01	1.004e+02	2.016e+02	1.005e+02	1.045e+02	1.969e+01
27	4.235e+02	1.001e+03	8.511e+02	7.601e+02	2.076e+02	4.177e+02	9.302e+02	4.340e+02	5.305e+02	1.708e+02
28	1.094e+03	1.989e+03	1.155e+03	1.212e+03	1.646e+02	1.284e+03	1.715e+03	1.405e+03	1.424e+03	8.792e+01
29	1.626e+05	1.321e+07	4.538e+05	8.084e+05	1.792e+06	4.563e+05	3.267e+06	1.181e+06	1.237e+06	5.101e+05
30	1.713e+04	7.710e+04	3.822e+04	4.003e+04	1.464e+04	1.660e+04	7.071e+04	4.273e+04	4.349e+04	1.326e+04

Çizelge 7.3'te GWO ve OppGWO algoritmalarının aynı tekrar sayısıyl 50-boyut için elde edilen sonuçlar çizelgede verilmiştir. Bu çizelgede verilen sonuçlara göre, OppGWO algoritmasının GWO algoritmasına kıyasla daha başarılı sonuçlar (GWO: 58/150, OppGWO: 92/150) vermiştir. Ayrıca 10 fonksiyonun bütün değerleri için OppGWO algoritması daha başarılı sonuçlar vermiştir.

Çizelge 7.3. GWO ve OppGWO için 50- Boyutlu CEC'14 optimizasyon sonuçları

FN	50D									
	GWO					OppGWO				
	En İyi	En Kötü	Medyan	Ortalama	S.Sapma	En İyi	En Kötü	Medyan	Ortalama	S.Sapma
1	6.525e+07	2.418e+08	1.118e+08	1.201e+08	3.463e+07	6.793e+07	1.697e+08	1.144e+08	1.158e+08	2.174e+07
2	7.119e+09	1.548e+10	1.046e+10	1.063e+10	1.935e+09	4.841e+09	9.143e+09	7.562e+09	7.456e+09	8.659e+08
3	2.464e+04	6.130e+04	3.648e+04	3.819e+04	7.555e+03	1.197e+04	2.343e+04	1.748e+04	1.768e+04	2.436e+03
4	5.701e+02	1.882e+03	7.951e+02	8.219e+02	2.023e+02	5.349e+02	1.117e+03	7.927e+02	8.008e+02	9.369e+01
5	2.100e+01	2.120e+01	2.114e+01	2.113e+01	3.797e-02	2.099e+01	2.118e+01	2.112e+01	2.111e+01	4.596e-02
6	3.892e+01	4.832e+01	4.349e+01	4.370e+01	2.276e+00	3.572e+01	4.464e+01	4.020e+01	4.008e+01	2.388e+00
7	7.151e+01	1.377e+02	8.831e+01	9.014e+01	1.191e+01	4.932e+01	7.672e+01	6.164e+01	6.260e+01	6.490e+00
8	2.886e+02	3.703e+02	3.288e+02	3.266e+02	1.814e+01	3.067e+02	4.317e+02	3.565e+02	3.550e+02	2.192e+01
9	3.585e+02	4.396e+02	3.991e+02	4.001e+02	1.792e+01	3.576e+02	4.688e+02	3.964e+02	3.984e+02	1.808e+01
10	7.314e+03	1.028e+04	8.575e+03	8.652e+03	6.928e+02	8.803e+03	1.236e+04	1.104e+04	1.089e+04	8.568e+02
11	1.069e+04	1.369e+04	1.209e+04	1.215e+04	6.636e+02	1.070e+04	1.390e+04	1.275e+04	1.274e+04	5.350e+02
12	2.404e+00	3.759e+00	3.422e+00	3.374e+00	2.770e-01	2.699e+00	3.746e+00	3.440e+00	3.389e+00	2.548e-01
13	7.301e-01	1.177e+00	9.357e-01	9.562e-01	1.025e-01	6.525e-01	1.018e+00	8.187e-01	8.173e-01	9.240e-02
14	1.807e+01	4.110e+01	2.398e+01	2.435e+01	4.201e+00	6.951e+00	1.987e+01	1.549e+01	1.524e+01	2.306e+00
15	2.161e+02	5.205e+03	4.002e+02	5.990e+02	7.369e+02	8.257e+01	3.109e+02	1.528e+02	1.576e+02	4.645e+01
16	2.052e+01	2.195e+01	2.116e+01	2.118e+01	3.610e-01	2.049e+01	2.197e+01	2.148e+01	2.142e+01	3.343e-01
17	3.105e+06	2.034e+07	5.990e+06	6.892e+06	3.270e+06	2.622e+06	1.096e+07	7.013e+06	7.146e+06	2.075e+06
18	9.931e+07	5.429e+08	2.213e+08	2.307e+08	7.261e+07	1.185e+08	3.066e+08	2.018e+08	2.035e+08	4.112e+07
19	6.705e+01	1.197e+02	9.567e+01	9.483e+01	1.123e+01	6.123e+01	1.232e+02	9.655e+01	9.322e+01	1.756e+01
20	4.611e+03	2.237e+04	1.084e+04	1.117e+04	4.541e+03	1.309e+03	5.463e+03	2.430e+03	2.662e+03	8.874e+02
21	7.407e+05	7.016e+06	2.307e+06	2.756e+06	1.509e+06	8.921e+05	4.797e+06	2.193e+06	2.398e+06	9.528e+05
22	7.012e+02	1.932e+03	1.346e+03	1.318e+03	2.811e+02	9.220e+02	2.006e+03	1.533e+03	1.505e+03	1.880e+02
23	4.157e+02	4.914e+02	4.473e+02	4.454e+02	1.788e+01	3.891e+02	4.436e+02	4.121e+02	4.123e+02	1.248e+01
24	2.000e+02	2.000e+02	2.000e+02	2.000e+02	5.288e-04	2.000e+02	2.731e+02	2.000e+02	2.029e+02	1.415e+01
25	2.197e+02	2.450e+02	2.299e+02	2.297e+02	5.596e+00	2.240e+02	2.426e+02	2.302e+02	2.305e+02	4.052e+00
26	1.006e+02	2.074e+02	1.013e+02	1.498e+02	5.040e+01	1.006e+02	2.153e+02	1.009e+02	1.521e+02	5.303e+01
27	1.298e+03	1.607e+03	1.446e+03	1.453e+03	6.972e+01	1.221e+03	1.527e+03	1.362e+03	1.358e+03	7.016e+01
28	1.804e+03	4.906e+03	2.217e+03	2.482e+03	7.114e+02	2.718e+03	4.013e+03	3.004e+03	3.076e+03	2.770e+02
29	2.990e+06	2.465e+07	7.666e+06	8.481e+06	3.839e+06	5.486e+06	2.750e+07	1.167e+07	1.231e+07	3.925e+06
30	9.169e+04	3.956e+05	2.084e+05	2.189e+05	7.045e+04	1.360e+05	5.223e+05	2.295e+05	2.471e+05	7.799e+04

Çizelge 7.4'de ise GWO ve OppGWO algoritmalarının 30 adet CEC'14 benchmark fonksiyonları üzerinde test edildikten sonraki başarı yüzdeleri karşılaştırmalı olarak verilmiştir. Bu çizelgeye bakıldığı zaman OppGWO algoritması GWO algoritmasına göre daha başarılı olmuştur. Ayrıca OppGWO algoritmasının 30-boyut da en iyi sonucu verdiği gözlemlenmiştir.

Çizelge 7.4. 10,30,50 boyut için GWO ve OppGWO algoritmalarının başarı yüzdeleri

	GWO			OppGWO		
	10D	30D	50D	10D	30D	50D
En İyi	%33.3	%40	%40	%66.7	%60	%60
En Kötü	%30	%20	%33.3	%70	%80	%66.7
Medyan	%36.6	%33.3	%43.3	%63.4	%66.7	%56.7
Ortalama	%30	%40	%40	%70	%60	%60
S.Sapma	%36.6	%11.1	%33.3	%63.4	%88.9	%66.7
Toplam	%33.3	%28.6	%38	%66.7	%71.4	%62

7.3. Özellik Seçim Sonuçları

Bu çalışmada mevcut kaynaklardan elde edilen farklı özelliklerdeki on sekiz adet sınıflandırma veri seti için GWO ve OppGWO algoritmaları eş zamanlı olarak çalıştırılarak her bir veri seti için sınıflandırma hatası, zaman, doğruluk ve özellik sayısı ölçümleri istatistiksel olarak elde edilmiştir. Bu çalışmada algoritmalar birbirinden bağımsız ve eş zamanlı olarak otuz defa çalıştırılmıştır. Sonuçlar çizelgeler halinde sunulmuştur. Çizelge 7.5'te kullanılan sınıflandırma veri setleri ile ilgili bilgiler verilmektedir.

Çizelge 7.5. Sınıflandırma veri setleri (UCI,2019)

	Veri Seti Adı	Özellik Sayısı	Örnek Sayısı
1	Breastcancer	9	695
2	BreastEW	30	569
3	Cnae	857	1080
4	CongressEW	16	435
5	Glass	10	214
6	HeartEW	13	270
7	IonosphereEW	34	351
8	ChessEW	36	3196
9	Lymphography	18	148
10	PageBlocking	10	5473
11	ObsNetwork	22	1075
12	Segment	19	2310
13	SonarEW	60	208
14	SpectEW	22	267
15	Tic-tac-toe	9	958
16	WaveformEW	40	5000
17	WineEW	13	178
18	Zoo	16	101

OppGWO ve GWO algoritmalarının sınıflandırma problemlerinde özellik seçimi için kullanılan sınıflandırma hatası (minimum), her bir çözümde seçilen özelliklerin sayısı (minimum) ile bu seçilen özellikler kullanılarak elde edilen sınıflandırmanın doğruluk oranı (maksimum) arasında bir denge sağlamak amacıyla Eşitlik 7.1'deki gibi hesaplanır (Mafarja ve Mirjalili,2017):

$$f_i = \alpha(1 - R_c) + (1 - \alpha) \frac{|R|}{|C|} \quad (7.1)$$

Burada f_i i. aday çözüme ait sınıflandırma hatasını, R_c sınıflandırma doğruluk oranını, $|R|$ seçilen özellik sayılarının toplamını, $|C|$ veri setinde bulunan toplam özellik sayısını, α ise $[0,1]$ aralığında değişen sınıflandırma değişkenini göstermektedir.

Bu çalışma kapsamında OppGWO ve GWO algoritmaları özdeş veri setleri (Çizelge 7.5) üzerinde eş zamanlı test edilerek sonuçlar karşılaştırılmalı olarak sunulmuştur. Çizelge 7.6'de GWO ve OppGWO algoritmalarının, on sekiz veri seti için birbirinden bağımsız 30 tekrarlı çalıştırma sonucunda elde ettiği sınıflandırma doğruluk sonuç değerleri verilmiştir. Çizelge gösterilen sonuçlara göre, OppGWO algoritmasına ait doğruluk değerlerinin (0 ile 1 arasında) GWO algoritmasına kıyasla daha başarılı sonuçlar (40/32) elde ettiği gözlemlenmiştir. Çizelgede verilen OppGWO algoritmasına ait doğruluk değeri sonuçları incelendiği zaman doğruya en yakın (1'e en yakın) sınıflandırmayı yaptığı gözlemlenmiştir.

Çizelge 7.7'de GWO ve OppGWO algoritmalarının tekrarlı çalıştırılması sonucunda elde edilen sınıflandırma hatası değerlerinin en iyi, en kötü, ortalama ve standart sapma ölçütleri verilmiştir. Çizelgedeki sonuçlar incelendiği zaman sınıflandırma hatası (minimum) değerlerinin daha iyi sonuçlar (47/25) vermiştir.

Çizelge 7.8'de iki algoritmanın da (GWO ve OppGWO) işleyiş süreleri karşılaştırılmıştır. OppGWO algoritması on sekiz veri seti içinde dört ölçüm değerine göre algoritmanın işleyiş süresi bakımından (37/35) GWO algoritmasına göre daha iyi sonuçlar vermiştir.

GWO ve OppGWO algoritmalarının otuz bağımsız tekrar sonucunda elde ettiği sınıflandırma veri setlerinin özellik sayıları Çizelge 7.9'de verilmiştir. Önerilen OppGWO algoritması daha az özellik kullanarak sınıflandırma (39/33) yaptığı gözlemlenmiştir. Elde edilen sonuçlara göre bu çalışma kapsamında önerilen OppGWO algoritması özellik sayısı, işleyiş süresi, doğruluk değeri ve sınıflandırma hatası ölçümleri bakımından orijinal GWO algoritmasına göre daha başarılı olduğu görülmüştür.

Elde edilen sonuçlara bakıldığında zaman OppGWO algoritmasının kullanılması veri setleri için özellik sayılarının azaltmakla beraber veri boyutunun da azalmasına sebep olur. Daha az sayıda özellik, daha kısa sürede, en az hata ile en doğru sınıflandırmayı yapmak sınıflandırma problemleri için önemli kazançlar sağlamaktadır.

Şekil 7.10'de on sekiz veri seti için tek tekrarlı çalıştırma sonucunda elde edilen uygunluk grafikleri gösterilmiştir.

Çizelge 7.6. GWO ve OppGWO algoritmalarının özellik seçimi için doğruluk değeri sonuçları

Veri Setleri	Doğruluk							
	GWO				OppGWO			
	En İyi	En Kötü	Ortalama	S.Sapma	En İyi	En Kötü	Ortalama	S.Sapma
Breastcancer	0.971223	0.956835	0.963549	0.003631	0.968345	0.949640	0.963453	0.003865
BreastEW	0.943761	0.929701	0.936028	0.003593	0.943761	0.926186	0.935852	0.003937
Cnae	0.999074	0.997222	0.998148	0.000942	0.999074	0.997222	0.998519	0.000828
CongressEW	0.967816	0.921839	0.952720	0.010811	0.970115	0.935632	0.955402	0.009986
Glass	1.000000	0.981308	0.990498	0.005688	1.000000	0.981308	0.991589	0.005264
HeartEW	0.814815	0.740741	0.792716	0.018797	0.814815	0.777778	0.798025	0.009566
IonosphereEW	0.951567	0.920228	0.937702	0.008289	0.954416	0.914530	0.939791	0.008084
ChessEW(krvskp)	0.979349	0.956195	0.970484	0.007861	0.982478	0.952128	0.974739	0.006862
Lymphography	0.885135	0.824324	0.852928	0.016329	0.878378	0.824324	0.849324	0.012566
PageBlocking	0.961813	0.954138	0.958481	0.001522	0.961995	0.957062	0.959108	0.001191
Magic	0.796057	0.784911	0.793623	0.001911	0.796793	0.785857	0.793386	0.001820
Segment	0.974892	0.969697	0.972583	0.001380	0.974459	0.969697	0.972525	0.001265
SonarEW	0.947115	0.879808	0.919712	0.013489	0.951923	0.899038	0.926923	0.012524
SpectEW	0.823970	0.794007	0.809238	0.008166	0.823970	0.794007	0.807990	0.007803
Tic-tac-toe	0.823591	0.773486	0.807411	0.012769	0.822547	0.783925	0.811517	0.009439
WaveformEW	0.794400	0.777800	0.786873	0.004126	0.794200	0.781600	0.787827	0.003191
WineEW	0.966292	0.915730	0.947191	0.010801	0.960674	0.932584	0.949813	0.008072
Zoo	0.990099	0.960396	0.981848	0.006413	1.000000	0.980198	0.984488	0.005627

Çizelge 7.7. GWO ve OppGWO algoritmalarının özellik seçimi için sınıflandırma hatası sonuçları

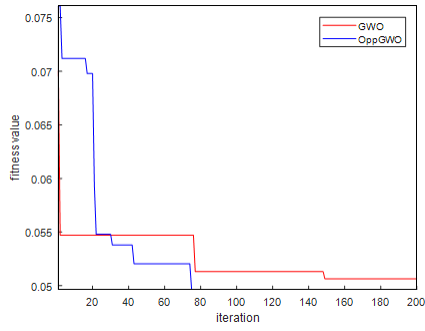
Veri Setleri	Sınıflandırma Hatası							
	GWO				OppGWO			
	En İyi	En Kötü	Ortalama	S. Sapma	En İyi	En Kötü	Ortalama	S.Sapma
Breastcancer	2.84e-02	3.49e-02	3.25e-02	1.43e-03	2.84e-02	3.49e-02	3.22e-02	1.73e-03
BreastEW	5.04e-02	5.66e-02	5.42e-02	1.54e-03	4.82e-02	6.14e-02	5.40e-02	2.57e-03
Cnae	9.87e-04	2.81e-03	1.90e-03	8.93e-04	1.33e-03	3.24e-03	2.01e-03	6.99e-04
CongressEW	2.82e-02	4.22e-02	3.57e-02	4.58e-03	2.59e-02	4.22e-02	3.45e-02	4.90e-03
Glass	2.00e-03	3.00e-03	2.17e-03	3.79e-04	2.00e-03	3.003e-03	2.33e-03	4.79e-04
HeartEW	1.62e-01	2.16e-01	1.78e-01	1.31e-02	1.63e-01	1.96e-01	1.73e-01	6.31e-03
IonosphereEW	4.15e-02	5.82e-02	4.96e-02	4.69e-03	3.87e-02	5.94e-02	4.83e-02	5.00e-03
ChessEW(krvskp)	2.26e-02	4.54e-02	3.15e-02	7.65e-03	2.10e-02	4.84e-02	2.71e-02	6.25e-03
Lymphography	9.81e-02	1.49e-01	1.18e-01	1.03e-02	1.04e-01	1.44e-01	1.20e-01	8.79e-03
PageBlocking	4.22e-02	4.58e-02	4.30e-02	6.91e-04	4.24e-02	4.37e-02	4.29e-02	3.25e-04
Magic	2.05e-01	2.12e-01	2.06e-01	1.30e-03	2.04e-01	2.12e-01	2.06e-01	1.29e-03
Segment	2.45e-02	2.66e-02	2.57e-02	5.56e-04	2.49e-02	2.71e-02	2.56e-02	5.83e-04
SonarEW	4.09e-02	7.51e-02	5.94e-02	9.13e-03	3.63e-02	7.42e-02	5.48e-02	9.42e-03
SpectEW	1.71e-01	2.04e-01	1.85e-01	8.80e-03	1.75e-01	2.04e-01	1.85e-01	9.73e-03
Tic-tac-toe	1.72e-01	2.20e-01	1.83e-01	1.26e-02	1.72e-01	2.02e-01	1.80e-01	8.39e-03
WaveformEW	2.03e-01	2.18e-01	2.09e-01	3.36e-03	2.06e-01	2.17e-01	2.10e-01	2.16e-03
WineEW	2.69e-02	4.35e-02	3.51e-02	6.37e-03	2.69e-02	4.68e-02	3.23e-02	5.77e-03
Zoo	4.38e-03	2.52e-02	1.21e-02	5.16e-03	4.38e-03	2.34e-02	1.08e-02	4.83e-03

Çizelge 7.8. GWO ve OppGWO algoritmalarının özellik seçimi için işleyiş süreleri (sn)

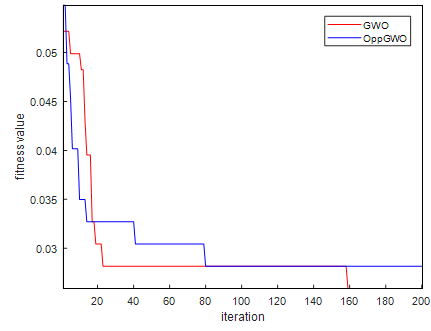
Veri Seti	Zaman							
	GWO				OppGWO			
	En İyi	En Kötü	Ortalama	S.Sapma	En İyi	En Kötü	Ortalama	S.Sapma
Breastcancer	285.80	301.77	293.88	3.86	285.65	309.42	295.36	4.91
BreastEW	273.99	305.16	290.03	6.79	272.96	299.73	287.44	7.38
Cnae	421.43	483.75	446.08	16.76	389.49	709.19	584.51	87.33
CongressEW	270.75	300.99	278.26	6.31	262.67	285.09	277.66	3.73
Glass	266.88	304.77	270.89	7.74	266.65	307.60	271.79	7.38
HeartEW	268.75	2475.50	345.44	402.31	279.22	281.08	273.01	3.03
IonosphereEW	259.37	290.85	278.21	5.68	273.59	285.70	279.37	3.31
ChessEW(krvskp)	590.00	712.07	640.61	27.59	606.84	688.21	653.92	20.58
Lymphography	265.89	293.56	274.47	5.14	262.98	283.34	273.96	4.58
PageBlocking	548.92	590.71	568.41	12.68	549.80	594.01	570.49	13.06
Magic	1214.35	1521.00	1356.84	41.68	1234.27	1396.19	1360.71	27.69
Segment	392.74	402.93	398.51	2.97	395.08	423.00	402.36	5.30
SonarEW	248.29	301.26	254.10	9.30	249.81	282.95	254.99	5.78
SpectEW	264.82	299.85	277.05	5.75	86.08	318.21	261.8	49.86
Tic-tac-toe	308.40	339.38	314.25	5.98	305.28	322.49	314.09	4.63
WaveformEW	890.48	1062.39	981.21	44.59	879.86	1087.13	992.57	45.33
WineEW	260.52	288.59	264.50	5.09	261.04	271.48	264.78	2.81
Zoo	259.66	274.32	262.90	3.36	259.42	270.52	263.81	3.38

Çizelge 7.9. GWO ve OppGWO algoritmalarının özellik seçimi için özellik sayıları değeri sonuçları

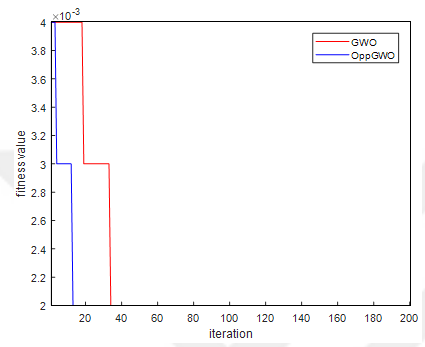
Veri Setleri	Özellik Sayısı							
	GWO				OppGWO			
	En İyi	En Kötü	Ortalama	S.Sapma	En İyi	En Kötü	Ortalama	S.Sapma
Breastcancer	3	7	4.93	1.36	3	7	5.26	1.53
BreastEW	5	14	8.67	2.17	7	14	10.2	2.13
Cnae	1	17	5.63	4.01	6	79	51.37	18.29
CongressEW	2	9	4.17	1.62	2	6	3.90	1.24
Glass	2	3	2.17	0.37	2	3	2.33	0.48
HeartEW	4	7	5.47	0.86	5	7	5.97	0.67
IonosphereEW	5	13	7.33	1.89	5	10	7.47	1.38
ChessEW(krvskp)	13	23	18.23	2.13	16	22	19.37	1.67
Lymphography	4	11	7.20	1.47	5	13	7.63	1.88
PageBlocking	4	6	4.97	0.89	4	6	4.83	0.79
Magic	4	5	4.97	0.18	4	5	4.97	0.18
Segment	4	6	4.70	0.65	4	6	5	0.59
SonarEW	12	26	18.40	3.328	15	27	20.2	3.01
SpectEW	1	10	7.17	2.35	1	11	7.30	2.79
Tic-tac-toe	5	7	6.50	0.73	5	7	6.50	0.82
WaveformEW	11	16	13.53	1.41	11	17	13.87	1.50
WineEW	3	6	5.40	0.86	3	8	5.90	1.09
Zoo	6	10	6.83	1.09	6	9	6.83	1.02



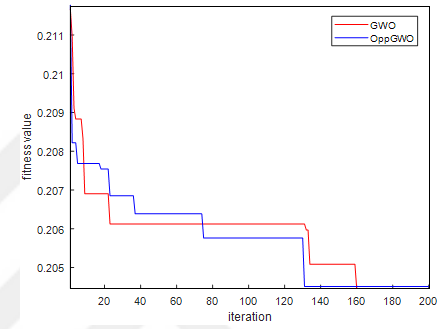
(BreastEW)



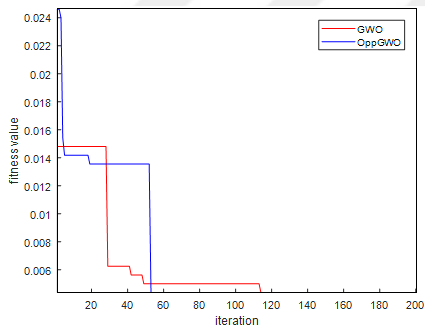
(CongressEW)



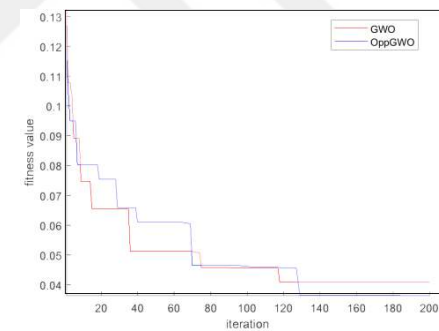
(Glass)



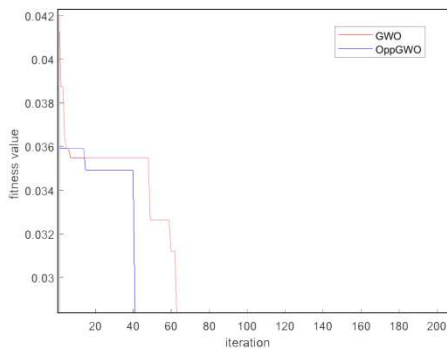
(Magic)



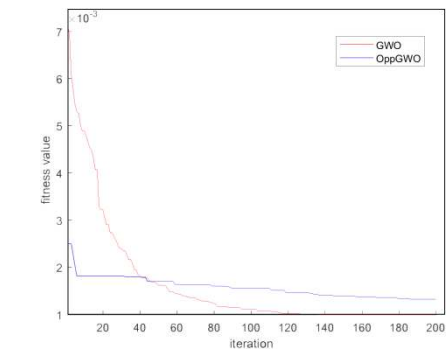
(Zoo)



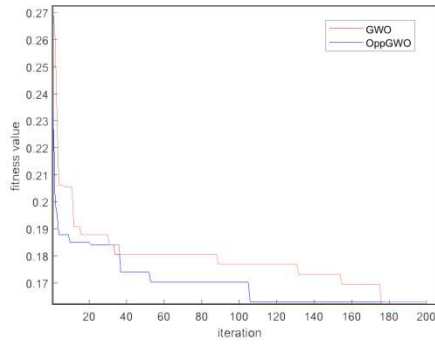
(Sonar)



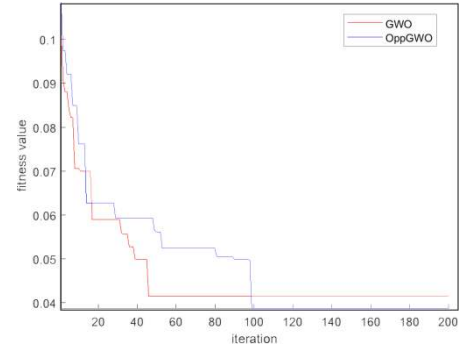
(Breastcancer)



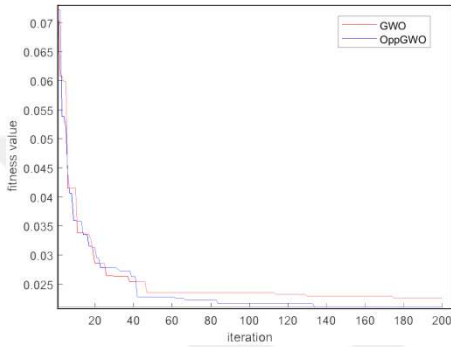
(Cnae)



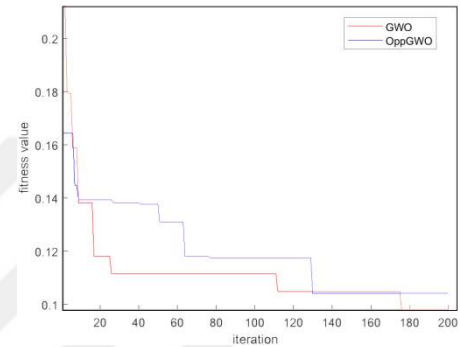
(Heart)



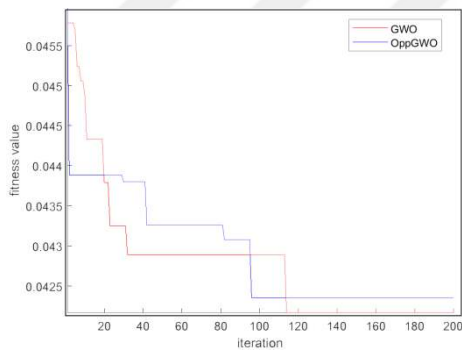
(Ionosphere)



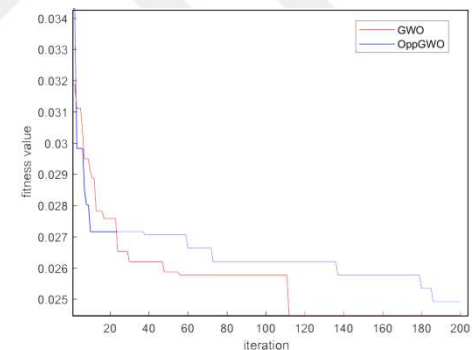
(Kr-vs-Kp)



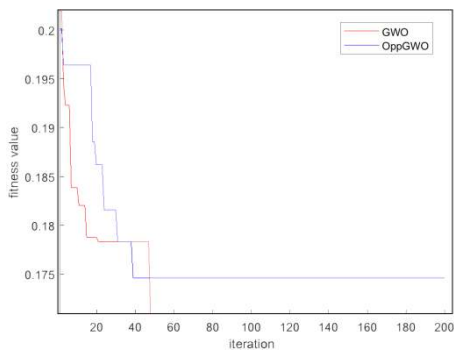
(Lymphography)



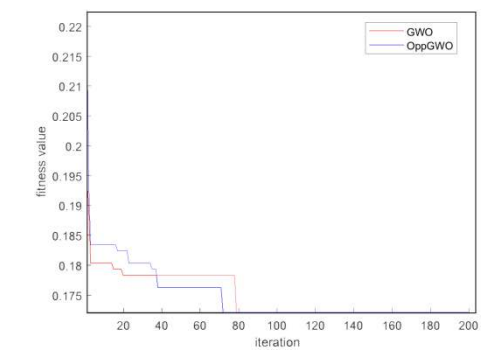
(Page Blocking)



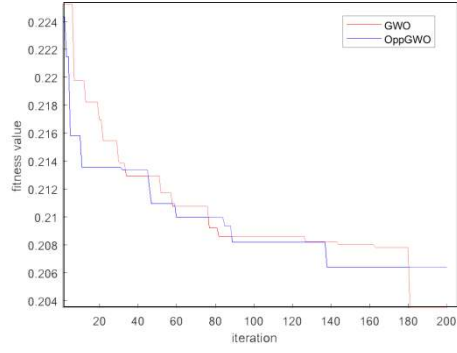
(Segment)



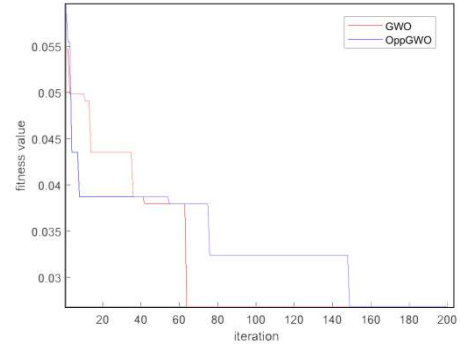
(Spect)



(Tic-Tac-Toe)



(Waveform)



(Wine)

Şekil 7.9. OppGWO ve GWO Algoritmaları İçin Uygunluk Eğrileri

8. SONUÇ VE TARTIŞMA

Günümüzde teknoloji hızla gelişmektedir. Teknolojinin gelişmesi de beraberinde birçok ihtiyacı doğurur. Bunlardan en önemlisi de gittikçe artan verilerdir. Bu verileri hızlı ve kolay şekilde erişebilmek için bir veri kümesinde bulunan çeşitli sınıflara dağıtılarak sınıflandırma yapılır. Bu sınıflandırma işlemleri için geliştirilmiş olan sınıflandırma algoritmaları kullanılarak veriler benzer özelliklere göre sınıflandırılırlar. Bu sınıflandırma algoritmalarını kullanırken ilk önce eğitim veri seti denilen sınıfları belli olan veri seti ile algoritma eğitilir ardından test veri seti denilen sınıfları belirli olmayan veri setiyle bu verileri doğru bir şekilde sınıflandırmaya çalışırlar. Sistemlerin matematiksel modelinin tam olarak bilinmediği durumlarda sadece verilen bir amaç fonksiyonuna göre ilgili parametrelerin türetilmesini sağlayan sezgisel algoritmalar optimizasyon problemlerinde son yıllarda oldukça popüler hale gelmiştir.

GWO algoritması, gri kurtların doğadaki yaşam ve avlanma stratejilerinden esinlenilerek geliştirilmiş bir meta sezgisel optimizasyon algoritmasıdır. Bu çalışma kapsamında sınıflandırma problemlerinde doğruya en yakın sınıflandırma yapabilmemiz için gerekli olan özellik seçimi ele alınarak, bu problem için karşıtlık tabanlı Gri Kurt Optimizasyon algoritması (OppGWO) önerilmiştir.

Bu çalışma 2.4 GHz 4 çekirdek 8. Nesil intel core i5 işlemciye sahip, MacOS Mojave işletim sisteminde Matlab’da geliştirilmiştir.Öncelikle önerilen ve GWO algoritması eş zamanlı olarak birbirinden bağımsız 30 denemeyle 30 adet CEC’14 Benchmark fonksiyonlarında test edildi.Fonksiyonların listesi Ek-2’de verildiği gibidir. Sonuçları tablolar ve grafikler halinde sunuldu. (Bknz Syf.36-38) Elde edilen sonuçlara göre OppGWO algoritması benchmark fonsiyonlarında daha iyi sonuçlar vermiştir.

Önerilen OppGWO algoritması içerisinde karşıtlık tabanlı öğrenme ile birlikte mutasyon ve farklı bir sınır kontrol yaklaşımı da kullanılmıştır. Ardından bu algoritmaları (GWO ve OppGWO) sınıflandırma problemlerinde önemli bir yere sahip olan özellik seçimi için eş zamanlı olarak birbirinden bağımsız 30 deneme ile UCI’den elde ettiğimiz 18 adet veri seti (Çizelge 7.5) üzerinde test edilmiştir. Bu çalışmada sınıflandırma algoritması olarak K-En Yakın Komşu (K-NN) algoritması seçilmiştir. Bu test işleminin ardından elde edilen sonuçlar tablolar halinde verilmiştir.

Elde edilen sonuçlara göre önerilen OppGWO algoritmasının özellik seçimi probleminde daha başarılı olduğu gözlemlenmiştir. Bu algoritmanın özellik sayısı (Çizelge 7.9), işleyiş süresi (Çizelge 7.8), doğruluk (Çizelge 7.6) ve sınıflandırma hatası (Çizelge 7.7) ölçütlerine göre orijinal GWO algoritmasına göre daha iyi olduğu söylenebilir. Daha kısa sürede, daha az sayıda özellik, daha az sayıda hata (sınıflandırma hatası) ile doğruya en yakın sınıflandırma işlemi yapılmıştır. İlerleyen adımlarda önerilen algoritmanın (OppGWO) başka sınıflandırıcılar ile (SVM gibi) farklı sınıflandırma metotlarının performanslarının iyileştirilmesinde kullanılması düşünülmektedir.



KAYNAKLAR

- Ateş, A. (2018). Sezgisel Algoritmelerde Çoklu Denetçi Parametrelerin Optimizasyonu. *İnönü Üniversitesi*. 69, 46-61.
- Ayre, L.B. (2006). Data Mining for Information Professionals.
- Akduman, B., & Türkay, B. E. (2010). Evrimsel Algoritmalar ile Elektrik Dağıtım Şebekelerinin Restorasyonu. *İstanbul Teknik Üniversitesi*.
- Akyol, S., & Alataş, B. (2012). Güncel Sürü Zekâsı Optimizasyon Algoritmaları. *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*. 1, 36-50.
- Berry, M. J., & Linoff, G.S. (2000). Mastering Data Mining. *John Wiley & Sons, Inc New York*, 5-407
- Bharadwaj, B.K., & Pal, S. (2011). Data Mining: A Prediction for Performance Improvement Using Classification. *International Journal of Computer Science and Information Security (IJCSIS)*, 9(4), 136-140.
- Bolon-Canedo, V., & Sanchez-Marono, N., & Alonso-Betanzos, A., & Benítez, J.M. & Herrera, F. (2014). A Review of Microarray Datasets and Applied Feature Selection Methods. *Information Sciences*, 282, 111–135.
- Braysy, O. (2001). Local Search and Variable Neighborhood Search Algorithms for The Vehicle Routing Problem with Time Windows, *PHD Thesis*.
- Budak, H. (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22, 21-31.
- Canayaz, M. (2015). Cırcır Böceği Algoritması: Yeni Bir Meta-Sezgisel Yaklaşım ve Uygulamaları. *İnönü Üniversitesi Fen Bilimleri Enstitüsü*, 77.
- Chandra, M., & Agrawal, A., & Kishor, A., & Niyogi, R. (2016). Web Service Selection with Global Constraints Using Modified Gray Wolf Optimizer. *International Conference on Advances in Computing Communications and Informatics (ICACCI)*. *IEEE*, 1989–1994.
- Çalışkan, K. S., & Soğukpınar, İ. (2008) KxKNN: K-Means ve K-En Yakın Komşu Yöntemleri ile Ağlarda Nüfuz Tespiti. *EMO Yayınları*, 120-124.
- Çolak, S. (2010). Genetik Algoritmalar Yardımı ile Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 19(3), 423-438.

- Das, S. (2001). Filters, Wrappers And A Boosting-Based Hybrid For Feature Selection. *In Proceedings of the Eighteenth International Conference on Machine Learning*, 74-81.
- Demirel, D. & Öztemiz, F., & Karcı, A. (2018). Fizik Tabanlı Meta-Sezgisel Algoritmaların Performans Karşılaştırılması. *IEEE*, 978-1-5386-6878-8/18.
- Doğan, L., & Yüzgeç, U. (2018). Robot Path Planning using Gray Wolf Optimizer in Proceedings- International Conference on Advanced Technologies. *Computer Engineering and Science (ICATCES'18)*.
- Dudani, AR, & Chudasama, K. (2016). Partial Discharge Detection in Transformer Using Adaptive Grey Wolf Optimizer Based Acoustic Emission Technique. *Cogent Eng* 3(1):1256083.
- Eskandar, H., & Sadollah, A., & Bahreininejad, A., & Hamdi, M. (2012). Water Cycle Algorithm-A Novel Metaheuristic Optimization Method for Solving Constrained Engineering Optimization Problems. *Computer and Structures Elsevier*.
- Faris, H., & Aljarah, I., & Al-Betar, M.A., & Mirjalili, S. (2018). Grey Wolf Optimizer: A Review of Recent Variants and Applications. *Neural Computing and Applications*. 30, 413-435.
- Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3, 1289–1305.
- Goldberg, D.E. (1999). Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley*.
- Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, 1157-1182.
- Han, J., & Pei, J., & Kamber, M. (2011). *3Th Edition Data Mining Concepts and Technques*. Morgan Kaufmann Publishers, Massachusetts, 78-100.
- İşçi, Ö., & Korukoğlu, S. (2003). Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama. *Yönetim ve Ekonomi*, 10(2).
- Jayakumar, N., & Subramanian, S., & Ganesan, S. & Elanchezhian, E.B. (2016). Grey Wolf Optimization for Combined Heat and Power Dispatch with Cogeneration Systems. *Int. J. Electr. Power Energy Syst.*, 74, 252-264.

- Kahraman, A.M., & Özdağlar, D. (2004). Su Dağıtım Sistemlerinin Genetik Algoritma ile Optimizasyonu. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Derg.*, 6(3), 1-18.
- Karaboğa, D., & Öztürk C. (2010). Fuzzy Clustering with Artificial Bee Colony Algorithm. *Scientific Research and Essays* , 5(14), 1899-1902.
- Karakas, M., & Yüzgeç U. (2019). Opposition based Gray Wolf Algorithm for Feature Selection in Classification Problems. *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*.
- Kenneth, V.P. (1999). *An introduction to Differential Evolution, in New Ideas in Optimization*. McGraw-Hill Publishing Company, New York, 79-108.
- Keskintürk, T., & Söyler, H. (2006). Global Karınca Kolonisi Optimizasyonu. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 21(4), 689-698.
- Kishor, A., & Singh, P.K. (2016). Empirical Study of Grey Wolf Optimizer. *In: Proceedings of fifth international conference on soft computing for problem solving. Springer*, 1037–1049.
- Koc, I., & Baykan, O. K., & Babaoglu, I. (2018). Gri Kurt Optimizasyon Algoritmasına Dayanan Çok Seviyeli İmge Eşik Seçimi. *J. Polytech.*, 21(4), 841-847.
- Koç, I., & Refik, N., & Kahramanlı, H. (2018). Türkiye'de Enerji Talebini Tahmin Etmek için Doğrusal Form Kullanarak GSA (Yerçekimi Arama Algoritması) ve IWO (Yabani Ot Optimizasyon Algoritması) Tekniklerinin Uygulanması. *S.Ü. Müh. Bilim ve Tekn. Derg.*, 6(4), 529-543.
- Korkmaz, E., & Akgüngör, A.P. (2018). Türkiye'deki Araç Sahipliğinin Çiçek Tozlaşma Algoritması ile Tahmini. *Gazi Üniversitesi Mühendislik Bilimleri Dergisi*, 4(1), 39-45.
- Koza, J. R. (1992). *Genetic Programming: On The Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, 619-633.
- Liang, J.J., & Qu, B.Y., & Suganthan, P.N. (2013). Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization.
- Liu, T., & Liu, S., & Chen, Z., & Ma, W.Y. (2003). An Evaluation on Feature Selection for Text Clustering. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*.

- Luo, Q., & Zhang, S., & Li, Z., & Zhou, Y. (2015). A Novel Complex – Valued Encoding Grey Wolf Optimization Algorithm. 9(1):4.
- Mahalakshmi, S., & SaiAshwini, C., & Meghana, S. (2001). Research Study of Big Data Clustering Techniques. *IJIRSE*, 4(5), 80-84.
- Mahdad, B., & Srairi, K. (2015). Blackout Risk Prevention in A Smart Grid Based Flexible Optimal Strategy Using Grey Wolf-Pattern Search Algorithms. *Energy Convers Manag*, 98, 411–429.
- Mahdavi, S., & Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. *Swarm Evol. Comput.*, 39, 1-23.
- Malik, M. R. S., & Mohideen, E.R., & Ali, L. (2015). Weighted Distance Grey Wolf Optimizer For Global Optimization Problems. *IEEE*.
- Medium, Kumar, A., Intl: <https://medium.com/@abhishek.km23/methods-of-feature-selection-3b4c88f0e2d5> (Ziyaret Edilme Tarihi: 04.07.2020)
- Mirjalili, S., & Mirjalili, S.M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in English Software*, 69, 46-61.
- Mittal, N., & Singh, U., & Sohi, B. S. (2016). Modified Grey Wolf Optimizer for Global Engineering Optimization. *Appl Comput Intell Soft Comput*, 8.
- Mohamad, M., & Selamat, A. (2015). An Evaluation on the Efficiency of Hybrid Feature Selection in Spam E-mail Classification. *Computer, Communications and Control Technology (I4CT)*.
- Özkan, Y. (2016). *Veri Madenciliği Yöntemleri*, Papatya Yayıncılık, İstanbul, 27-36.
- Özsağlam, Y. M., & Çunkaş M. (2008). Optimizasyon Problemlerinin Çözümü için Parçaçık Sürü Optimizasyonu Algoritması. *Politek. Derg.*, 11(4), 299-305.
- Powell, M. J. D. (1977). Restart Procedures for The Conjugate Gradient Method. *Math Program*, 12(1), 241–254.
- Pouramirarsalani, A., & Khalilian, M., & Nikravanshalmani, A. (2017). Fraud detection in E-banking by using the hybrid feature selection and evolutionary algorithms. *IJCSNS*, 17(8), 271-279.
- Rodríguez, L., & Castillo, O., & Soria, J. (2016). Grey Wolf Optimizer with Dynamic Adaptation of Parameters Using Fuzzy Logic. In: *2016 IEEE congress on evolutionary computation (CEC)*. *IEEE*, 3116–3123.

- Rodríguez, L., & Castillo, O., & Soria, J., & Melin, P., & Valdez, F., & Gonzalez, C. I., & Martinez, G. E., & Soto, J. (2017). A Fuzzy Hierarchical Operator in The Grey Wolf Optimizer Algorithm. *Appl Soft Comput*, 57, 315–328.
- Rojas-Morales, N., & Riff Rojas, M.C., & Montero Ureta ,E. (2017). A Survey and Classification of Opposition-Based Metaheuristics. *Comput. Ind. Eng.*,110, 424-435.
- Saremi, S., & Mirjalili, S.Z., & Mirjalili, S. M. (2015). Evolutionary Population Dynamics and Grey Wolf Optimizer. *Neural Comput Appl* 26(5),1257–1263.
- Subramanya, K.B., & Somani, A. (2017). Enhanced Feature Mining and Classifier Models to Predict Customer Churn for An E-Retailer, Cloud Computing. *Data Science & Engineering Confluence 7th International Conference on*, 12-13 Ocak, Noida.
- Storn, R. (1997). Differential Evolution A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, 341-359.
- Şeker S.E., Weka ve Verim Madencilği, İstanbul, Bilgisayar Kavramları Yayınları, 2015
- Şeker, S. E. (2008). *Sezgisel Algoritmalar*. Bilgisayar Kavramları, <http://bilgisayarkavramlari.sadievrenseker.com/2008/12/22/sezgisel-algoritmalar-bulussal-algoritmalar-heuristic-algorithms/>, (25.10.2019).
- Tan, P., & Steinbach, M., & Karpatne, A., & Kumar, V. (2005). *Cluster Analysis: Basic Concepts and Algorithms. Introduction to Data Mining*. Pearson, New York, 534-549.
- Tanyıldızı, E., & Cigal T. (2017). Kaotik Haritalı Balina Optimizasyon Algoritmaları. *Fırat Üniversitesi Mühendislik ve Bilim Dergisi*, 29(1), 307-317.
- Tereshko, V., & Loengarov, A. (2005). Collective Decision Making in Honey-Bee Foraging Dynamics. *Computing and Information Systems*, 9(3).
- Taşkın, Ç., & Emel, G.G. (2002). Genetik Algoritmalar ve Uygulama Alanları. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152.
- Tizhoosh, H.R. (2005). Opposition-based learning: A New Scheme for Machine Intelligence in Proceedings. *International Conference on Computational*

- Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet.*
- UCI, Machine Learning Repository, Intl.: <https://archive.ics.uci.edu/ml/datasets.php> (Ziyaret Edilme Tarihi, 05.03.2019).
- Van den Bergh, F., & Engelbrecht, A. (2006). A Study of Particle Swarm Optimization Particle Trajectories. *Inf Sci*, 176, 937–971.
- Yang, X.S. (2014). Nature- Inspired Optimization Algorithm. *Elsevier*.
- Yang, B., & Zhang, X., & Yu, T., & Shu, H., & Fang, Z. (2016). Grouped Grey Wolf Optimizer for Maximum Power Point Tracking of Doubly-Fed Induction Generator Based Wind Turbine. *Energy Convers Manag*, 133, 427–443.
- Yıldız, O., & Tez M., & Bilge H.Ş., & Akcayol M.A., & Güler, İ. (2012). Meme Kanseri Sınıflandırması için Gen Seçimi. *IEEE 20. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*.
- Wang, D., & Zhang, Z., & Bai, R., & Mao, Y. (2017). A Hybrid System With Filter Approach and Multiple Population Genetic Algorithm for Feature Selection in Credit Scoring. *Journal of Computational and Applied Mathematics*, 329, 307-321.
- Zhang, S., & Zhou, Y. (2015). Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis. *Discret Dyn Nat Soc*, 17.
- Zhou, J., & Zhu, W., & Zheng, Y., & Li, C. (2016). Precise Equivalent Model of Small Hydro Generator Cluster and Its Parameter Identification Using Improved Grey Wolf Optimiser. *IET Gener Transm Distrib*, 10(9), 2108–2117.

EKLER

Ek 1. CEC'14 Temel Fonksiyonların Listesi

	Fonksiyon Adı	Fonksiyon
1	High Conditioned Elliptic	$f_1(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$
2	Bent Cigar	$f_2(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$
3	Discus	$f_3(x) = 10^6 x_1^2 \sum_{i=2}^D x_i^2$
4	Rosenbrock's	$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
5	Ackley's	$f_5(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$
6	Weierstrass	$f_6(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k \cdot 0.5)]$ $a = 0.5, b = 3, k_{max} = 20$
7	Griewank's	$f_7(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
8	Rastrigin's	$f_8(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
9	Modified Schwefel's	$f_9(x) = 418.9829 \cdot D - \sum_{i=1}^D g(z_i), \quad z_i = x_i + 4.209687462275036e + 002$ $g(z_i) = \begin{cases} z_i \sin(z_i ^{1/2}), & \text{if } z_i \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{ 500 - \text{mod}(z_i, 500) }) - \frac{(z_i - 500)^{23}}{10000D}, & \text{if } z_i > 500 \\ (\text{mod}(z_i , 500) - 500) \sin(\sqrt{ \text{mod}(z_i , 500) - 500 }) - \frac{(z_i + 500)^2}{10000D}, & \text{if } z_i < -500 \end{cases}$
10	Katsuura	$f_{10}(x) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{ 2^j x_i - \text{round}(2^j x_i) }{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$
11	HappyCat	$f_{11}(x) = \left \sum_{i=1}^D x_i^2 - D \right ^{1/4} + (0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i) / D + 0.5$
12	HGBat	$f_{12}(x) = \left \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right ^{1/2} + (0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i) / D + 0.5$
13	Expanded Griewank's Plus Rosenbrock's	$f_{13}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1))$
14	Expanded Scaffer's F6	$g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$ $f_{14}(x) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$

Ek-2: CEC'14 test grubu fonksiyonlarının listesi

	No	Fonksiyon Adı	Fonksiyon
	1	R.High Conditioned Elliptic	$F_1(x) = f_1(M(x - o_1)) + F_1^*$
Unimodal Functions	2	R.Bent Cigar	$F_2(x) = f_2(M(x - o_2)) + F_2^*$
	3	R.Discus	$F_3(x) = f_3(M(x - o_3)) + F_3^*$
	4	S.R. Rosenbrock's	$F_4(x) = f_4\left(M\left(\frac{2.048(x - o_4)}{100}\right) + 1\right) + F_4^*$
	5	S.R.Ackley's	$F_5(x) = f_5(M(x - o_5)) + F_5^*$
Simple	6	S.R.Weierstrass	$F_6(x) = f_6\left(M\left(\frac{0.5(x - o_6)}{100}\right)\right) + F_6^*$
Multimodal Functions	7	S.R.Griewank's	$F_7(x) = f_7\left(M\left(\frac{600(x - o_7)}{100}\right)\right) + F_7^*$
	8	S.Rastrigin's	$F_8(x) = f_8\left(\frac{5.12(x - o_8)}{100}\right) + F_8^*$
	9	S.R. Rastrigin's	$F_9(x) = f_9\left(\frac{5.12(x - o_9)}{100}\right) + F_9^*$
	10	S.Schwefel's	$F_{10}(x) = f_{10}\left(\frac{1000(x - o_{10})}{100}\right) + F_{10}^*$
	11	S.R. Schwefel's	$F_{11}(x) = f_{11}\left(M\left(\frac{1000(x - o_{11})}{100}\right)\right) + F_{11}^*$
	12	S.R.Katsuura	$F_{12}(x) = f_{12}\left(M\left(\frac{5(x - o_{12})}{100}\right)\right) + F_{12}^*$
	13	S.R.HappCat	$F_{13}(x) = f_{13}\left(M\left(\frac{5(x - o_{13})}{100}\right)\right) + F_{13}^*$
	14	S.R.HGBat	$F_{14}(x) = f_{14}\left(M\left(\frac{5(x - o_{14})}{100}\right)\right) + F_{14}^*$
Simple Multimodal Functions	15	S.R.Expanded Griewank's Plus Rosenbrock's	$F_{15}(x) = f_{15}\left(M\left(\frac{5(x - o_{15})}{100}\right) + 1\right) + F_{15}^*$
	16	S.R.Expanded Scaffer's F6	$F_{16}(x) = f_{16}(M(x - o_{16}) + 1) + F_{16}^*$
Hybrid Functions	17	Hybrid Function-1	N:3 ,p=[0.3,0.3,0.4], g ₁ : Modified Schwefel's f ₉ , g ₂ : Rastrigin's f ₈ g ₃ : High Conditioned Elliptic f ₁
$F(X) = g_1(M_1z_1) + g_2(M_2z_2) + \dots + g_N(M_Nz_N) + F^*(x)$	18	Hybrid Function-2	N:3 ,p=[0.3,0.3,0.4], g ₁ : Bent Cigar f ₂ , g ₂ : HGBat f ₁₂ g ₃ : Rastrigin's f ₈
	19	Hybrid Function-3	N:4 ,p=[0.2,0.2,0.3,0.3], g ₁ : Griewank's f ₇ , g ₂ : Weierstrass f ₆ g ₃ : Rosenbrock's f ₄ , g ₄ : Scaffer's F6 f ₁₄
	20	Hybrid Function-4	N:4 ,p=[0.2,0.2,0.3,0.3], g ₁ : HGBat f ₁₂ , g ₂ : Discus f ₃ g ₃ : Expanded Griewank's Plus Rosenbrock's f ₁₃ , g ₄ : Rastrigin's f ₈
	21	Hybrid Function-5	N:5 ,p=[0.1,0.2,0.2,0.2,0.3], g ₁ : Scaffer's F6 f ₁₄ , g ₂ : HGBat f ₁₂ g ₃ : Rosenbrock's f ₄ , g ₄ : Modified Schwefel's f ₉ , g ₅ : High Conditioned Elliptic f ₁
	22	Hybrid Function-6	N:5 ,p=[0.1,0.2,0.2,0.2,0.3], g ₁ : Katsuura f ₁₀ , g ₂ : HappyCat f ₁₁ g ₃ : Expanded Griewank's Plus Rosenbrock's f ₁₃ , g ₄ : Modified Schwefel's f ₉ , g ₅ : Ackley's f ₅
Composition Functions	23	Composition Functions-1	N=5, $\sigma = [10, 20, 30, 40, 50]$, $\lambda = [1, 1e - 6, 1e - 26, 1e - 6, 1e - 6]$, bias = [0, 100, 200, 300, 400] g ₁ : R. Rosenbrock's f ₄ , g ₂ : High Conditioned Elliptic f ₁ , g ₃ : R. Bent Cigar f ₂ , g ₄ : R. Discus f ₃ g ₅ : High Conditioned Elliptic f ₁
	24	Composition Functions-2	N=3, $\sigma = [20, 20, 20]$, $\lambda = [1, 1, 1]$, bias = [0, 100, 200] g ₁ : Schwefel's f ₁₀ , g ₂ : R. Rastrigin's f ₉ , g ₃ : R. HGBat f ₁₄
	25	Composition Functions-3	N=3, $\sigma = [10, 30, 50]$, $\lambda = [0.25, 1, 1e - 7]$, bias = [0, 100, 200] g ₁ : R. Schwefel's f ₁₁ , g ₂ : R. Rastrigin's f ₉ , g ₃ : R. High Conditioned Elliptic f ₁
	26	Composition Functions-4	N=5, $\sigma = [10, 10, 10, 10, 10]$, $\lambda = [0.25, 1, 1e - 7, 2.5, 10]$, bias = [0, 100, 200, 300, 400] g ₁ : R. RSchwefel's f ₁₁ , g ₂ : R. HappyCat f ₁₃ , g ₃ : R. High Conditioned Elliptic f ₁ , g ₄ : R. Weierstrass f ₆ g ₅ : R. Griewank's f ₇

Ek 2. (devam ediyor)

Compositio n	27	Composition Functions-5	$N=5, \sigma = [10, 10, 10, 20, 20], \lambda = [10, 10, 2.5, 2.5, 1e-6], \text{bias} = [0, 100, 200, 300, 400]$ g_1 : R. HGBat F_{14}' , g_2 : R. Rastrigin's F_9' , g_3 : R. Schwefel's F_{11}' , g_4 : R. <i>Weierstrass</i> F_6' , g_5 : R. High Conditioned Elliptic F_1'
Functions	28	Composition Functions-6	$N=5, \sigma = [10, 20, 30, 40, 50], \lambda = [2.5, 10, 2.5, 5e-4, 1e-6], \text{bias} = [0, 100, 200, 300, 400]$ g_1 : R. E. Griewank's Plus Rosenbrock's F_{15}' , g_2 : R. HappyCat F_{13}' , g_3 : R. Schwefel's F_{11}' , g_4 g_5 : R. High Conditioned Elliptic F_1'
$F(x) = \sum_{i=1}^N \{\omega_i * [\lambda_i g_i(x) + \text{bias}_i]\} + F^*$	29	Composition Functions-7	$N=3, \sigma = [10, 30, 50], \lambda = [1, 1, 1], \text{bias} = [0, 100, 200]$ g_1 : Hibrid 1 F_{17}' , g_2 : Hibrid 2 F_{18}' , g_3 : Hibrid 3 F_{19}'
	30	Composition Functions-8	$N=3, \sigma = [10, 30, 50], \lambda = [1, 1, 1], \text{bias} = [0, 100, 200]$ g_1 : Hibrid 4 F_{20}' , g_2 : Hibrid 5 F_{21}' , g_3 : Hibrid 6 F_{22}'

ÖZ GEÇMİŞ

Kişisel Bilgiler

Adı Soyadı: Melis Karakaş

Doğum Yeri ve Tarihi: Eskişehir/1993



Eğitim Durumu

Lisans Öğrenimi: Trakya Üniversitesi

Bildiği Yabancı Diller: İngilizce

Bilimsel Faaliyetleri: Türkçe Metinlerde Makine Öğrenmesi ile Yazar Tanıma (Tez)

İş Deneyimi

Stajlar: Hisarlar Bilgi ve İnovasyon Merkezi-2014

Köklüce Makina -2015

Projeler: -

Çalıştığı Kurumlar: Bewell Teknoloji Sanayi ve Ticaret A.Ş.

İletişim

Adres: Vişnelik Mah.Tuncay Sok.Duygu Apt. 1/6 Odunpazarı/Eskişehir

Tel: 0 (535) 275 47 89

E-Posta Adresi: meliskarkas@icloud.com

Akademik Çalışmaları

-Karakaş M., Yüzgeç U., “Sınıflandırma Problemlerinde Özellik Seçimi İçin Karşıtlık Tabanlı Gri Kurt Optimizasyon Algoritması”, IEEE, ISMSIT2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies ,978-1-7281-3789-6/19,11-12-13 October 2019, Ankara, Turkey

Yabancı Dil Bilgisi

YÖK-DİL 51.25

27/07/2017

Tarih:17/07/2020