



**FİZİK TABANLI YENİ HİBRİT OPTİMİZASYON
ALGORİTMALARININ GELİŞTİRİLMESİ VE VERİ
MADENCİLİĞİNDE UYGULAMALARI**

Soner KIZILOLUK

**Doktora Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Ahmet Bedri ÖZER
KASIM-2017**

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

**FİZİK TABANLI YENİ HİBRİT OPTİMİZASYON ALGORİTMALARININ
GELİŞTİRİLMESİ VE VERİ MADENCİLİĞİNDE UYGULAMALARI**

DOKTORA TEZİ

Soner KIZILOLUK

(122129201)

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Yazılım

Danışman: Doç. Dr. Ahmet Bedri ÖZER

Tezin Enstitüye Verildiği Tarih: 25 Ekim 2017

KASIM-2017

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**FİZİK TABANLI YENİ HİBRİT OPTİMİZASYON ALGORİTMALARININ
GELİŞTİRİLMESİ VE VERİ MADENCİLİĞİNDE UYGULAMALARI**

DOKTORA TEZİ

Soner KIZILOLUK

122129201

Tezin Enstitüye Verildiği Tarih : 25 Ekim 2017

Tezin Savunulduğu Tarih : 24 Kasım 2017

Tez Danışmanı : Doç. Dr. Ahmet Bedri ÖZER (F.Ü)

Diğer Jüri Üyeleri : Doç. Dr. Bilal ALATAŞ (F.Ü)

Doç. Dr. İlhan AYDIN (F.Ü)

Yrd. Doç. Dr. İ. Berkan AYDİLEK (H.Ü)

Yrd.Doç. Dr. Fatih TÜYSÜZ (H.Ü)

KASIM-2017

ÖNSÖZ

Sezgisel yöntemler birçok problemde oldukça başarılı sonuçlar verebilmesine rağmen her problemin çözümünde yüksek başarı gösterememektedir. Bundan dolayı, araştırmacılar yeni yöntemler geliştirmekte veya mevcut yöntemlerin performansını artırmak için öneriler sunmaktadır. Algoritmaların performansını artırmak amacıyla, iki veya daha fazla algoritmanın özelliklerini tek bir algorithmada birleştiren hibritleme yöntemi kullanılabilir. Bu tez çalışmasında, iki farklı hibrit algoritma önerilmiş ve önerilen yöntemlerin küresel optimizasyon problemlerindeki ve sınıflandırma kural keşfindeki performansları test edilmiştir.

Doktora çalışmalarım boyunca, değerli görüş ve katkılarıyla beni yönlendiren, her konuda desteklerini hiçbir zaman esirgemeyen, çalışma disiplinini bize kazandıran ve kıymetli tecrübelerinden faydalandığım tez danışmanım Sayın Doç. Dr. Ahmet Bedri ÖZER'e teşekkürü bir borç bilirim.

Soner KIZILOLUK
ELAZIĞ-2017

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	X
KISALTMALAR.....	XII
SEMBOLLER LİSTESİ	XIII
1. GİRİŞ.....	1
1.1. Genel Bilgiler	1
1.2. Literatür Taraması	3
1.3. Tezin Amacı	7
1.4. Tezin Yapısı.....	8
2. OPTİMİZASYON HAKKINDA GENEL BİLGİLER	9
2.1. Optimizasyon Yöntemleri.....	11
2.1.1. Klasik Yöntemler.....	11
2.1.2. Sezgisel Algoritmalar	12
2.2. Fizik Tabanlı Sezgisel Optimizasyon Algoritmaları	15
2.2.1. Yerçekimsel Arama Algoritması	15
2.2.2. İyon Hareketi Algoritması	16
2.2.3. Su Döngüsü Optimizasyon Algoritması	17
2.2.4. Karadelik Arama Algoritması	17
2.2.5. Yüklü Sistem Arama Algoritması	18
2.2.6. Yapay Fizik Optimizasyon Algoritması	19
2.2.7. Parçacık Çarpışma Algoritması	19
2.2.8. Büyük Çöküş Algoritması	20
2.3. Hibrit Algoritmalar	21
3. VERİ MADENCİLİĞİ HAKKINDA GENEL BİLGİLER	24
3.1. Veri Madenciliği Süreci	24
3.1.1. Veri Temizleme	25
3.1.2. Veri Bütünleştirme	26
3.1.3. Veri İndirgeme.....	26
3.1.4. Veri Dönüştürme	26

3.1.5. Veri Madenciliği Algoritmasını Uygulama	26
3.1.6. Sonuçlar Sunum ve Değerlendirme	27
3.2. Veri Madenciliği Modelleri	27
3.2.1. Sınıflandırma ve Regresyon	28
3.2.2. Kümeleme.....	29
3.2.3. Birliktelik Kuralları	29
4. HİBRİT ELEKTROMANYETİZMA BENZERİ – PARÇACIK SÜRÜ OPTİMİZASYONU – DİFERANSİYEL GELİŞİM ALGORİTMASI.....	30
4.1. Elektromanyetizma Benzeri Algoritma	30
4.1.1. Başlangıç Popülasyonunu Oluşturma	31
4.1.2. Yerel Arama	32
4.1.3. Toplam Kuvvet Hesaplama	32
4.1.4. Parçacıkların Hareketi	33
4.2. Parçacık Sürü Optimizasyon Algoritması	34
4.3. Diferansiyel Gelişim Algoritması.....	35
4.3.1. Başlangıç Popülasyonunun Oluşturulması ve Kullanılan Parametreler	36
4.3.2. Mutasyon	36
4.3.3. Çaprazlama	36
4.3.4. Değerlendirme ve Seçim	37
4.4. Hibrit EM-PSO-DE Algoritması	37
5. HİBRİT PARLAMENTER OPTİMİZASYON – BÜYÜK PATLAMA BÜYÜK ÇÖKÜŞ ALGORİTMASI.....	40
5.1. Parlamenter Optimizasyon Algoritması	40
5.1.1. Başlangıç Popülasyonunu Oluşturma ve Popülasyonun Bölümlendirilmesi	41
5.1.2. Grup İçi Rekabet.....	41
5.1.3. Gruplar Arası İşbirliği	42
5.1.4. Algoritmanın Sonlandırılması	43
5.2. Büyük Patlama – Büyük Çöküş Algoritması.....	43
5.3. HPO-BBBC algoritması	44
6. DENEYSEL ÇALIŞMADA KULLANILAN TEST FONKSİYONLARI VE VERİ SETLERİ	47
6.1. Kalite Testi Fonksiyonları	47
6.1.1. Rastrigin Fonksiyonu.....	47
6.1.2. Rosenbrock Fonksiyonu	48
6.1.3. Sphere Fonksiyonu	49
6.1.4. Griewank Fonksiyonu	49

6.1.5. Ackley Fonksiyonu.....	50
6.1.6. Levy Fonksiyonu	51
6.1.7. Alpine Fonksiyonu	51
6.1.8. Quintic Fonksiyonu	52
6.1.9. Trigonometric Fonksiyonu	53
6.2. Nümerik Sınıflandırma Veri Setleri	53
6.2.1. Kural Temsili.....	54
6.2.2. Uygunluk Fonksiyonu	55
7. DENEYSEL SONUÇLAR	56
7.1. Hibrit EM-PSO-DE Sonuçları	56
7.2. HPO-BBBC Sonuçları.....	70
8. SONUÇLAR VE ÖNERİLER.....	83
KAYNAKLAR.....	85
ÖZGEÇMİŞ	92

ÖZET

Optimizasyon, bir problemin alternatif çözümlerinin arasından en iyi olanını seçme işlemidir. Doğrusal olmayan ve birçok değişkene sahip problemleri matematiksel olarak çözmek zordur. Bu zorluğun üstesinden gelebilmek için genellikle sezgisel yöntemlere başvurulmaktadır. Sezgisel yöntemler birçok problemde istenilen çözüme ulaşmada gayet başarılı olsa da bazı problemlerde istenilen çözümü elde edememektedir. Bu nedenle yeni algoritmalar geliştirilmekte veya var olan algoritmaların daha etkili çalışması için öneriler sunulmaktadır. Hibritleme, bir algoritmanın çözüm kalitesini veya performansını artırmak için kullanılan yöntemlerden biridir.

Bu tez çalışmasında iki farklı hibrit algoritma önerilmiştir. Bunlardan birincisi; Elektromanyetizma Benzeri (EM - Electromagnetism like) algoritmasının çözüm kalitesini ve performansını artırmak amacıyla EM, Parçacık Sürü Optimizasyon (PSO – Particle Swarm Optimization) algoritması ve Diferansiyel Gelişim (DE - Differential Evolution) algoritması kullanılarak önerilen hibrit EM-PSO-DE algoritmasıdır. İkincisi ise; Parlamenter Optimizasyon Algoritması' nın (POA - Parliamentary Optimization Algorithm) çözüm kalitesi ve performansını artırmak amacıyla POA ve Büyük Patlama Büyük Çöküş (BB-BC - Big Bang Big Crunch) algoritması kullanılarak önerilen Hibrit Parlamenter Optimizasyon – Büyük Patlama Büyük Çöküş (HPO-BBBC) algoritmasıdır. Önerilen hibrit algoritmaların küresel optimizasyon problemlerindeki performansı 9 farklı kalite testi fonksiyonu kullanılarak test edilmiştir. Sınıflandırma kural keşfindeki performansları ise UCI ve KEEL veri ambarlarından alınan veri setleri üzerinde analiz edilmiştir. Deneysel sonuçlara göre, hibrit EM-PSO-DE algoritması küresel optimizasyon problemlerinde EM, PSO ve DE' den daha iyi sonuçlar vermektedir. Sınıflandırma kural keşfinde, hibrit EM-PSO-DE, EM' nin biraz gerisinde kalsa da, EM' ye göre çalışma süresini önemli ölçüde azaltmıştır. HPO-BBBC algoritmasının ise çalışma süresi POA' ya göre daha fazla olsa da küresel optimizasyon ve sınıflandırma kural keşfinde POA ve BB-BC' ye göre daha başarılı sonuçlar vermiştir.

Anahtar Kelimeler: Hibrit Optimizasyon Algoritmaları, Küresel Optimizasyon, Veri Madenciliği, Sınıflandırma

SUMMARY

Developing New Physical Based Hybrid Optimization Algorithms and Applications in Data Mining

Optimization is the process of choosing the best solution for a problem from other alternative solutions. When the problem is not linear and problem has many variables, it is difficult to solve the problem mathematically. Usually heuristic methods has been using for overcome this difficulty. Heuristic methods are very successful to find the desired solution in many problems, but they could fail obtain the desired solution in some problems. Therefore, new algorithms have been developed, or suggestions have been proposed for improve the existing algorithms. Hybridization is one of the techniques to improve the quality of the solutions of an algorithm or improve the performance of an algorithm.

Two different hybrid algorithms have been proposed in this thesis. The first of these; Hybrid Electromagnetism Like - Particle Swarm Optimization - Differential Evolution (EM-PSO-DE) algorithm which is the combination of the EM, the PSO and the DE in order to improve the performance of the EM; the second; the Hybrid Parliamentary Optimization-Big Bang Big Crunch (HPO-BBBC) algorithm which is the combination of the POA and the BB-BC to improve the performance of the POA. The performances of the proposed hybrid algorithms on global optimization problems has been tested using mathematical test functions. Moreover, the performances of the proposed methods on classification have been tested with data sets from UCI and KEEL data warehouses. Performed experimental study has shown that, the hybrid EM-PSO-DE has better results than the EM, PSO and DE in global optimization problems. In the classification problems, although the hybrid EM-PSO-DE has shown slightly worse results than the EM, processing time of the hybrid EM-PSO-DE is faster than the EM. The HPO-BBBC has shown better results than the POA and the BBBC in global optimization and classification problems, however, processing time of the HPO-BBBC is slightly slower than POA.

Key Words: Hybrid Optimization Algorithms, Global Optimization, Data Mining, Classification

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Bir fonksiyonun yerel minimum, yerel maksimum, küresel minimum ve küresel maksimum noktaları.....	10
Şekil 2.2. Sezgisel Yöntemler.....	15
Şekil 2.3. İşbirlikçi hibritler, a) çok kademeli b) sıralı c) paralel.....	22
Şekil 2.4. Bütünleyici hibritler, a) tam manipülasyon b) kısmi manipülasyon	23
Şekil 3.1. Veri madenciliği sürecindeki adımlar	25
Şekil 3.2. Veri madenciliği modelleri.....	28
Şekil 4.1. Parçacıkların hareketi	30
Şekil 5.1. POA akış şeması.....	40
Şekil 5.2. POA' nın yönelme mekanizması.....	41
Şekil 5.3. Grupların birleşmesi	42
Şekil 5.4. BB-BC akış şeması.....	44
Şekil 5.5. HPO-BBBC akış şeması.....	45
Şekil 5.6. Genel üyeler etrafında daha iyi birey arama mekanizması	46
Şekil 6.1. Rastrigin fonksiyonu	48
Şekil 6.2. Rosenbrock fonksiyonu.....	48
Şekil 6.3. Sphere fonksiyonu.....	49
Şekil 6.4. Griewank fonksiyonu	50
Şekil 6.5. Ackley fonksiyonu	50
Şekil 6.6. Levy fonksiyonu.....	51
Şekil 6.7. Alpine fonksiyonu	52
Şekil 6.8. Quintic fonksiyonu	52
Şekil 6.9. Trigonometric fonksiyonu.....	53
Şekil 6.10. Bir kuralın temel yapısı	54
Şekil 6.11. Örnek kural.....	54
Şekil 7.1. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonları için yakınsama grafikleri	61
Şekil 7.2. EM-PSO-DE ve bileşenlerinin 20 boyutlu kalite testi fonksiyonları için yakınsama grafikleri	62

Şekil 7.3. EM-PSO-DE ve bileşenlerinin 30 boyutlu kalite testi fonksiyonları için	
yakınsama grafikleri	63
Şekil 7.4. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonları için	
yakınsama grafikleri	74
Şekil 7.5. HPO-BBBC ve bileşenlerinin 20 boyutlu kalite testi fonksiyonları için	
yakınsama grafikleri	75
Şekil 7.6. HPO-BBBC ve bileşenlerinin 30 boyutlu kalite testi fonksiyonları için	
yakınsama grafikleri	76



TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. EM' nin genel yapısı	31
Tablo 4.2. Başlangıç popülasyonu oluşturma.....	31
Tablo 4.3. Yerel arama	32
Tablo 4.4. Toplam kuvvet hesaplama.....	33
Tablo 4.5. Parçacıkların hareketi	34
Tablo 4.6. Hibrit EM-PSO-DE' nin temel yapısı	38
Tablo 4.7. EM-PSO-DE hareket evresi	38
Tablo 6.1. Deneysel çalışmada kullanılan veri setleri ve özellikleri	54
Tablo 7.1. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki sonuçları	58
Tablo 7.2. EM-PSO-DE ve bileşenlerinin 20 boyutlu kalite testi fonksiyonlarındaki sonuçları	59
Tablo 7.3. EM-PSO-DE ve bileşenlerinin 30 boyutlu kalite testi fonksiyonlarındaki sonuçları	60
Tablo 7.4. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki ortalama çalışma süreleri(sn).....	64
Tablo 7.5. EM-PSO-DE ve bileşenlerinin, 10 boyutlu test fonksiyonları için uygunluk fonksiyonunun 10000 defa çağırılmasındaki sonuçları	65
Tablo 7.6. EM-PSO-DE ve bileşenlerinden Banana veri setinde elde edilen sonuçlar.....	67
Tablo 7.7. EM-PSO-DE ve bileşenlerinden Banknote veri setinde elde edilen sonuçlar ...	68
Tablo 7.8. EM-PSO-DE ve bileşenlerinden Phoneme veri setinde elde edilen sonuçlar....	69
Tablo 7.9. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki sonuçları	71
Tablo 7.10. HPO-BBBC ve bileşenlerinin 20 boyutlu kalite testi fonksiyonlarındaki sonuçları	72
Tablo 7.11. HPO-BBBC ve bileşenlerinin 30 boyutlu kalite testi fonksiyonlarındaki sonuçları	73
Tablo 7.12. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki ortalama çalışma süreleri(sn).....	77

Tablo 7.13. HPO-BBBC ve bileşenlerinin, 10 boyutlu test fonksiyonları için uygunluk fonksiyonunun 10000 defa çağırılmasındaki sonuçları	78
Tablo 7.14. HPO-BBBC ve bileşenlerinden Banana veri setinde elde edilen sonuçlar	80
Tablo 7.15. HPO-BBBC ve bileşenlerinden Banknote veri setinde elde edilen sonuçlar ...	81
Tablo 7.16. HPO-BBBC ve bileşenlerinden Phonome veri setinde elde edilen sonuçlar ...	82



KISALTMALAR

SSO	: Swallow Swarm Optimization (Kırlangıç Sürü Optimizasyonu)
PSO	: Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
ACO	: Ant Colony Optimization (Karıncı Koloni Optimizasyonu)
HS	: Harmony Search (Armoni Arama)
ABC	: Artificial Bee Colony (Yapay Arı Kolonisi)
CEA	: Co-evolutionary Algorithm (Eş-Evrimsel Algoritma)
CA	: Cultural Algorithm (Kültürel Algoritma)
GA	: Genetic Algorithm (Genetik Algoritma)
FPA	: Flower Pollination Algorithm (Çiçek Tozlaşma Algoritması)
CSA	: Clonal Selection Algorithm (Klonal Seçim Algoritması)
MVO	: Multi-Verse Optimizer (Çoklu-Evren Optimizasyon)
BB-BC	: Big Bang – Big Crunch (Büyük Patlama – Büyük Çöküş)
CG	: Conjugate Gradient (Eşlenik Eğim)
EM	: Electromagnetism-like (Elektromanyetizma-Benzeri)
COA	: Chaos Optimization Algorithm (Kaos Optimizasyon Algoritması)
GSA	: Gravitational Search Algorithm (Yerçekimsel Arama Algoritması)
DE	: Differential Evolution (Diferansiyel Gelişim)
POA	: Parliamentary Optimization Algorithm (Parlamentar Optimizasyon Algoritması)

SEMBOLLER LİSTESİ

A	: Alt sınır
B	: Bayrak
c₁, c₂	: Sınıflandırma uygunluk fonksiyonunda bulunan kullanıcı tanımlı ağırlıklar
c₁, c₂	: PSO' daki Öğrenme katsayıları
CR	: Çaprazlama oranı
DN	: Doğru Negatif
DP	: Doğru Pozitif
F	: Ölçekleme faktörü
f(x)	: Uygunluk fonksiyonu
f(x^{best})	: En iyi parçacığın uygunluk değeri
Fⁱ	: i parçacığına diğer parçacıklar tarafından etki eden toplam kuvvet
gbest	: Tüm popülasyonda o ona kadar bulunmuş en iyi konum
k	: BB-BC' de o anki iterasyon sayısı
l_k, l	: Arama uzayının alt sınırı
LSITER	: Yerel aramadaki maksimum iterasyon sayısı
M	: POA' da grup sayısı
m, n	: Kullanıcı tanımlı ağırlık sabitleri
MAXITER	: Maksimum iterasyon sayısı
N	: POA' da birey sayısı
ort(Qⁱ)	: Aday üyelerin uygunluk değerlerinin ortalaması
ort(rⁱ)	: Genel üyelerin uygunluk değerlerinin ortalaması
p'	: Genel üyenin yeni konumu
p₀	: Genel üyenin mevcut konumu
pbest	: Bir parçacığın kişisel en iyi konumu
P_d	: Grupların silinme olasılığı
p_i	: i. aday üye
P_m	: Grupların birleşme olasılığı
powerⁱ	: i. grubun gücü
r	: BB-BC' de 0 ile 1 arasında üretilen rastgele bir sayı
r₁, r₂, r₃	: DE' de rastgele seçilen 3 farklı kromozom
u_k	: Arama uzayının üst sınırı
Ü	: Üst sınır
v	: PSO' da hız
w	: Atalet katsayısı
x	: PSO' da konum
x^c	: Kütle merkezi
x^{new}	: Yeni parçacık
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
δ	: Yerel arama parametresi
λ, λ₁, λ₂, JI	: 0 ile 1 arasında üretilen rastgele bir sayı

1. GİRİŞ

1.1. Genel Bilgiler

“En iyileme” anlamına gelen optimizasyon genel bir ifade ile, bir problemin alternatif çözümleri arasından en iyi olanını seçme işlemidir şeklinde tanımlanabilir [1]. Optimizasyon matematiksel olarak ise, bir $f(x)$ fonksiyonunun sonucunu maksimum veya minimum yapacak x değerlerini bulmak şeklinde tanımlanabilir [2, 3]. Problemin doğrusal olmadığı ve birçok değişkene sahip olduğu durumlarda o problemi matematiksel olarak çözmek zorlaşır. Bu zorluğun üstesinden gelebilmek için genellikle sezgisel yöntemlere başvurulmaktadır [3].

Tarih boyunca insanlar problemleri çözmek için sezgisel, diğer bir ifade ile deneme yanılma yaklaşımını kullanmışlardır. Birçok önemli keşif de deneme yanılma ile gerçekleştirilmiştir. Canlıların günlük yaşamdaki birçok deneyimi de sezgiseldir. Çünkü sezgisellik doğadaki kısıtlı kaynakları oldukça verimli bir şekilde kullanmayı sağlar. Sezgisellik araştırmacıların da ilgisini çekmiş ve bunu kullanarak daha esnek ve verimli yöntemler geliştirmeyi amaçlamışlardır [4, 5]. Sezgisel yöntemler, karmaşık bir problem için makul sürede deneme yanılma yolu ile kabul edilebilir çözümler sunmayı amaçlamaktadır. Çünkü ilgilenilen problemin karmaşıklığı mümkün olan her çözümü ve kombinasyonu aramayı imkânsız hale getirebilir, bu yüzden kabul edilebilir bir zaman ölçeğinde iyi uygulanabilir çözüm bulmak amaçlanır. Sezgisel yöntemler en iyi çözümü bulacaklarını garanti etmez fakat kısa zamanda en makul çözümü ortaya koyar [5].

Sezgisel yöntemlerin birçoğu doğadan esinlenilerek geliştirilmiştir. Sezgisel yöntemler doğadaki esinlendikleri alana göre; sosyal tabanlı, fizik tabanlı, biyoloji tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı, matematik tabanlı ve hibrit algoritmalar olmak üzere farklı kategorilere ayrılabilir [6-8]. Sosyal tabanlı algoritmalar, insanların sosyal hayattaki davranışlarından esinlenilerek geliştirilmiştir. Emperyalist Yarışmacı Algoritma [9] ve Öğretme-Öğrenme Tabanlı Optimizasyon Algoritması [10] sosyal tabanlı algoritmalara örnek olarak verilebilir. Doğadaki fiziksel olaylardan esinlenilerek geliştirilen algoritmalar ise fizik tabanlı algoritmalarlardır. Yerçekimsel Arama Algoritması [11] ile Su Döngüsü Algoritması [12] fizik tabanlı algoritmalara örnek olarak verilebilir. Biyoloji tabanlı algoritmalar ise canlıların davranışlarının ve biyolojik yapılarının taklidi ile ortaya

ıkııı algoritmalardır ve en bilineni Genetik Algoritmadır [13]. Kimyasal olaylardan esinlenilerek geliştirilen algoritmalar ise kimya tabanlı algoritmalar ve Yapay Kimyasal Tepkime Optimizasyon Algoritması [14] bunlara örnek olarak verilebilir. Müzik tabanlı algoritmaların en bilineni ise bir orkestradaki en iyi armoniyi sağlayabilmek amacıyla alınacak notaları yakalayabilme prensibine dayanan Armoni Arama Algoritmasıdır [15]. Spor tabanlı algoritmaların en bilineni ise bir ligdeki spor takımlarının birbiri ile karşılaşmasını taklit eden Lig Şampiyonası Algoritmasıdır [16]. Sezgisel ve matematik programlama tekniklerinin birleştirilmesi ile oluşturulmuş algoritmalar ise matematik tabanlı algoritmalar. Hibrit algoritmalar ise birden fazla algoritmanın bir arada kullanılması ile elde edilmektedir [8].

Günümüzde literatürde başarılı sezgisel algoritmalar bulunsa da; her problemin çözümünde yüksek başarı gösterememektedir. Bu durum, yeni sezgisel yöntemlerin geliştirilmesine veya var olan sezgisel yöntemlerin diğer sezgisel yöntemler, makine öğrenme teknikleri vb. ile hibritlenmesine olan ihtiyacı açıka ortaya koymaktadır [17, 18]. Hibritleme, iki veya daha fazla algoritmanın özelliklerini tek bir algortmada birleştirmek olarak tanımlanabilir [19]. Bir algoritmanın çözüm kalitesini veya performansını (örneğin: yakınsama hızı, alışma süresi vb.) geliştirmek hibritlemenin temel amaçlarından [20].

Sezgisel algoritmalar arama ve optimizasyon problemlerinin dışında; mühendislik problemleri, yönetim bilimi, web analizi, telekomünikasyon, kriptoloji, lojistik, akıllı şehirler ve finans gibi eşitli alanlarda sıklıkla kullanılmaktadır [7, 21]. Veri madenciliği tekniklerinde de sezgisel algoritmalar sıklıkla başvurulmaktadır.

Günümüzde bilişim teknolojilerinin gelişmesiyle birlikte artık tüm veriler sayısal ortamda saklanmaktadır. Fakat bu veriler tek başlarına bir anlam ifade etmezler. Eldeki bu veri yığınının anlamlı, ilgin ve işe yarar bilgiler ıkarmak için veri madenciliği teknikleri kullanılmaktadır. Veri madenciliğinde kullanılan teknikler tahmin edici ve tanımlayıcı olmak üzere ikiye ayrılır. Tahmin edici tekniklerde; sonuçları bilinen verilerden bir model geliştirilir ve bu modelden yararlanılarak bilinmeyen veri kümeleri için sonuç tahmini amaçlanır. Tanımlayıcı tekniklerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. Tahmin edici tekniklerin en önemlilerinden birisi sınıflandırmadır. Sınıflandırmada ilk olarak sınıfı belli olan eğitim verileri üzerinde analiz yapılarak belirli kurallar ıkarılır. Daha sonra ıkarılan bu kurallara göre sınıfı belli olmayan veriler bir sınıfa dâhil edilebilmektedir [2,6].

1.2. Literatür Taraması

Bir algoritmanın performansını arttırmak için hibritleme yöntemi sıkça kullanılmaktadır. Literatürde araştırmacıların sezgisel yöntemleri kullanarak geliştirdiği birçok hibrit algoritma bulunmaktadır. Bu bölümde, literatürde bulunan bazı hibrit algoritma çalışmalarına değinilmiştir. Tez konusu gereği ağırlıklı olarak fizik tabanlı sezgisel yöntemler kullanılarak geliştirilen hibrit algoritmalar incelenmiştir.

Kaveh ve arkadaşları [22], SSO (Swallow Swarm Optimization) algoritması ve PSO (Particle Swarm Optimization) algoritmasını hibritleyerek küresel optimizasyon için HPSSO algoritmasını geliştirmişlerdir. Çalışmada algoritmanın performansını 11 farklı kalite testi fonksiyonu ile test etmişlerdir. Deneysel sonuçlar HPSSO' nun bileşenlerine göre daha iyi sonuçlar verdiğini göstermiştir.

Kıran ve arkadaşları [23], küresel optimizasyon problemlerinin çözümü için ACO (Ant Colony Optimization) ve PSO' yu kullanarak HAP (Hybrid Ant Particle) optimizasyon algoritmasını önermişlerdir. Önerilen yöntemde ACO ve PSO her iterasyonda bağımsız olarak çalışır ve kendi sonuçlarını üretir. Üretilen en iyi sonuç küresel en iyi olarak seçilir ve bir sonraki iterasyonda parçacıkların ve karıncaların yeni konumlarını belirlemede kullanılır. Çalışmada önerilen yöntemin performansı kalite testi fonksiyonlarında test edilmiş ve elde edilen sonuçlar HAP, PSO ve ACO' ya göre daha başarılı olduğunu göstermiştir.

Wu ve arkadaşları [24] HS (Harmony Search) algoritmasının doğruluk ve yakınsama oranını arttırmak için ABC (Artificial Bee Colony) algoritmasını kullanarak hibrit armoni arama algoritmasını önermişlerdir. Çalışmada ABC algoritması ve türevleri HS algoritmasının armoni hafıza bileşenini geliştirmek için kullanılmıştır. Önerilen yöntemin performansı 12 iyi bilinen kalite testi fonksiyonu kullanılarak test edilmiştir.

Li ve arkadaşları [25], PSO ve ABC algoritmalarının avantajlarını birleştirerek yeni bir hibrit PS-ABC algoritması önermişlerdir. Önerilen yöntem, PSO algoritmasındaki lokal arama evresi ve ABC algoritmasındaki iki farklı küresel arama evresi birleştirilerek oluşturulmuştur. 13 farklı kalite testi fonksiyonu kullanılarak PS-ABC algoritmasının performansı ölçülmüş ve elde edilen sonuçlar önerilen yöntemin daha başarılı olduğunu göstermiştir.

Sun ve arkadaşları [26] CEA (Co-evolutionary Algorithm), CA (Cultural Algorithm) ve PSO algoritmalarını kullanarak küresel optimizasyon problemlerinin çözümü için yeni bir

hibrit algoritma önermişlerdir. Deneysel çalışma 13 farklı kalite testi fonksiyonu kullanılarak gerçekleştirilmiş ve elde edilen sonuçlar önerilen yöntemin PSO' nun farklı varyasyonlarına karşı daha başarılı olduğunu göstermiştir.

Ali ve Tawhid [27] tarafından PSO ve GA (Genetic Algorithm) kullanılarak HPSOGA şeklinde isimlendirilen hibrit algoritma önerilmiştir. Önerilen yöntem temel olarak üç adımdan oluşmaktadır. İlk adımda PSO algoritması uygulanır. İkinci adımda popülasyon alt popülasyonlara bölünür ve her alt popülasyon üzerinde çaprazlama operatörü uygulanır. Son adımda ise tüm popülasyona mutasyon operatörü uygulanır. Önerilen yöntemin başarısı çeşitli kalite testi ve potansiyel enerji fonksiyonları kullanılarak ölçülmüştür. Elde edilen sonuçlar önerilen yöntemin daha başarılı sonuçlar verdiğini göstermiştir.

Nabil [28] FPA (Flower Pollination Algorithm) ve CSA (Clonal Selection Algorithm) kullanarak FPA algoritmasının modifiye edilmiş versiyonu olan MFPA' yı önermiştir. Önerilen yöntem 23 kalite testi fonksiyonu kullanılarak test edilmiştir ve sonuçlar beş farklı algoritmanın sonuçları ile karşılaştırılmıştır.

Jangir ve arkadaşları [29] küresel optimizasyon ve optimum reaktif güç dağılımı problemlerinin çözümü için PSO ve MVO (Multi Verse Optimizer) kullanılarak geliştirilen HPSO-MVO algoritmasını önermiştir. Deneysel sonuçlar önerilen yöntemin, bileşenlerine göre daha başarılı sonuçlar verdiğini göstermiştir.

Zandi ve arkadaşları [30] optimum reaktif güç dağılımı problemlerinin çözümü için BB-BC (Big Bang - Big Crunch) ve PSO algoritmalarının karışımından hibrit BB-BC algoritmasını geliştirmişlerdir. Önerilen yöntemde yeni parçacıklar, BB-BC algoritmasının kütle merkezini kullanmanın yanı sıra PSO' da mevcut olan parçacıkların kişisel en iyi konumu ve tüm parçacıkların en iyi konumu kullanılarak oluşturulmuştur. Deneysel çalışma, önerilen yöntemin PSO ve BB-BC algoritmalarına göre daha başarılı sonuçlar verdiğini göstermiştir.

Ahmadi ve Abdi [31] tarafından fotovoltaik panel, rüzgar türbini ve batarya bankı gibi bağımsız bir hibrit güç sisteminin optimal boyutlandırılması için hibrit BB-BC algoritması önerilmiştir. Çalışmada BB-BC algoritmasının keşif yeteneğini geliştirmek için PSO' nun özelliklerinden yararlanılmıştır. Ayrıca yeni arama alanları keşfetmek için, parçacıkların konumlarının güncellenmesinden sonra bir mutasyon operatörü de kullanılmıştır. Önerilen yöntem İran'da, coğrafi ve ekonomik sebeplerden dolayı elektrik şebekesi bulunmayan bir köyde uygulanmış ve umut verici sonuçlar elde edilmiştir.

Hora ve arkadaşları [32] bir hidroelektrik santrali hidro jeneratör gruplarının 10 yıllık arıza ve tamir süreleri ile ilgili veriler üzerinde en iyi dağılım fonksiyonlarını tanımlamak amacı ile BB-BC ve CG (Conjugate Gradient) algoritmalarından oluşan hibrit BB-BC-CG algoritmasını önermişlerdir. Deneysel çalışma önerilen yöntemin umut vadeden sonuçlar verdiğini göstermiştir.

Sedighzadeh ve arkadaşları [33] proton değişimli membran yakıt hücresinin parametre tanımlaması için BB-BC ve PSO algoritmalarını kullanarak hibrit BB-BC algoritmasını önermişlerdir. Deneysel sonuçlar önerilen yöntemin proton değişimli membran yakıt hücresinin parametrelerini tanımlamak için etkin ve güvenilir bir teknik olduğunu ortaya koymuştur.

Sedighzadeh ve Bakhtary [34] dengeli ve dengesiz radyal dağıtım sistemlerinin bir ağ konfigürasyonu ve kondansatör yerleşimi optimizasyonu için BB-BC ve PSO algoritmaları ile Hibrit BB-BC algoritması geliştirmişlerdir. Önerilen yöntemde yeni parçacıklar kütle merkezi, en iyi yerel ve küresel pozisyonlar kullanılarak hesaplanmıştır. Simülasyon sonuçları önerilen yöntemin performansının karşılaştırılan algoritmalara göre biraz daha yüksek veya benzer olduğunu göstermiştir. Sedighzadeh ve arkadaşları [35] başka bir çalışmada ise aynı yöntemi otonom mikrogridlerde gerilim ve frekans regülasyonu amacı ile kullanmışlardır. Simülasyon sonuçları önerilen yöntemin bileşenlerine göre daha iyi sonuçlar verdiğini göstermiştir.

Wang ve arkadaşları [36] EM (Electromagnetism-like algorithm) ve COA (Chaos Optimization Algorithm) tabanlı yeni bir hibrit EM algoritmasını geliştirmişlerdir. Çalışmada EM algoritması için COA tabanlı yeni bir yerel arama prosedürü önermişlerdir. Önerilen yöntem dört farklı kalite testi fonksiyonu üzerinde test edilmiştir. Test sonuçları önerilen yöntemin bileşenlerine göre daha başarılı olduğunu göstermiştir.

Javadi ve arkadaşları [37] hücre içi ve hücre dışı yerleşim problemleri için yeni bir matematiksel model önermişlerdir. Çalışmada önerdikleri bu yeni modelde hücre içi ve hücre dışı malzeme akışlarının yeniden düzenleme maliyetlerini en aza indirmek için, EM ve GA tabanlı yeni bir hibrit algoritma geliştirmişlerdir. Önerilen algoritmada, ilk başta rastgele bir şekilde başlangıç popülasyonu oluşturulur. Daha sonra önceden belirlenmiş sayıda iterasyon ile bu popülasyon GA ile geliştirilir. Ardından GA tarafından modifiye edilmiş popülasyon EM algoritmasına başlangıç popülasyonu olarak verilir. Deneysel çalışma önerilen yöntemin gayet başarılı olduğunu göstermiştir.

Chang ve Lee [38] kademeli PID kontrol tasarımı için EM ve GA tabanlı geliştirilmiş EM-GA algoritmasını önermişlerdir. Önerilen yöntemde GA, EM'nin yerel arama evresinde kullanılmıştır. Simülasyon sonuçları önerilen yöntemin EM algoritmasına göre hem daha başarılı sonuçlar verdiğini hem de daha az çalışma süresine sahip olduğunu göstermiştir.

Jamili ve arkadaşları [39] EM ile tavlama benzetimi algoritmasını kullanarak hibrit bir algoritma önermişlerdir. Önerilen yöntemde ilk başta tavlama benzetimi algoritması ile iyi bir başlangıç çözümü bulunur. Daha sonra rastgele bir biçimde geri kalan aday çözümler üretilerek başlangıç popülasyonu oluşturulur. Daha sonra EM ve tavlama benzetimi algoritmaları döngüsel olarak çalıştırılır. Önerilen yöntem iş çizelgeleme amacıyla kullanılmış ve başarılı sonuçlar elde edilmiştir.

Muhsen ve arkadaşları [40] Fotovoltaik modül modelinin parametrelerinin çıkarılması için DE ve EM algoritmalarından oluşan bir hibrit yöntem önermişlerdir. Önerilen yöntemde, DE algoritmasının mutasyon operatörünü geliştirmek için EM algoritmasından yararlanılmıştır.

Teimouri ve arkadaşları [41] hafıza tabanlı EM algoritmasını önermişlerdir. Önerilen yöntemde, iyi çözümlerin özelliklerini kaydetmek ve algoritmayı geri beslemek için yerel arama prosedürüne hafıza matrisi eklenmiştir. Önerilen yöntem iyi bilinen kalite testi fonksiyonlarında test edilmiş ve başarılı sonuçlar elde edilmiştir.

Lee ve Lee [42] EM algoritmasının yerel arama prosedüründe PSO algoritmasını kullanarak EM algoritmasının etkinliğini arttırmışlardır.

Sun ve arkadaşları [43] GSA (Gravitational Search Algorithm) ve GA kullanarak hibrit GSA-GA algoritmasını önermişlerdir. Önerilen yöntemde GSA'nın işleyişi sırasında yerel bir noktada algoritmanın takılması halinde, GA'nın rulet seçimi ve ayırık mutasyon operatörleri çalıştırılmaktadır. Önerilen yöntem 6 farklı fotoğraf üzerinde test edilmiş ve başarılı sonuçlar vermiştir.

Sarafrazi ve arkadaşları [44] sayısal optimizasyon problemlerinin çözümü için GSA ve Kepler algoritmasını kullanarak yeni bir hibrit algoritma önermişlerdir. Önerilen yöntemde, GSA ve Kepler algoritmaları birbiri ardına uygulanmıştır. Bir algoritmanın çıkışı, diğer algoritmanın girişini oluşturmaktadır. Deneysel çalışmalardan elde edilen sonuçlar, önerilen hibrit algoritmanın, kalite testi fonksiyonlarını ve pratik optimizasyon problemlerini optimize edecek kadar sağlam olduğunu ortaya koymaktadır.

1.3. Tezin Amacı

Bu tez çalışmasında küresel optimizasyon ve sınıflandırma kural keşfi için fizik tabanlı algoritmalar kullanılarak yeni hibrit algoritmaların geliştirilmesi amaçlanmıştır. Çalışmada iki farklı hibrit algoritma önerilmiştir. Bunlardan birincisi; Elektromanyetizma Benzeri Algoritma (EM - Electromagnetism-like algorithm), Parçacık Sürü Optimizasyon (PSO - Particle Swarm Optimization) Algoritması ve Diferansiyel Gelişim (DE - Differential Evolution) Algoritması kullanılarak geliştirilen Hibrit Elektromanyetizma Benzeri - Parçacık Sürü Optimizasyon - Diferansiyel Gelişim (EM-PSO-DE) algoritmasıdır. İkincisi; Parlamenter Optimizasyon Algoritması (POA - Parliamentary Optimization Algorithm) ve Büyük Patlama Büyük Çöküş (BB-BC - Big Bang Big Crunch) algoritması kullanılarak geliştirilen Hibrit Parlamenter Optimizasyon - Büyük Patlama Büyük Çöküş (HPO-BBBC) algoritmasıdır. Önerilen yöntemlerin küresel optimizasyondaki başarıları 9 farklı kalite testi fonksiyonu kullanılarak test edilmiştir. Sınıflandırma performansları ise üç farklı sınıflandırma veri seti kullanılarak test edilmiştir. Sonuçlar incelendiğinde, EM-PSO-DE kalite testi fonksiyonlarında genel olarak bileşenlerinden daha iyi sonuçlar elde etmiştir. Sınıflandırmada ise bir veri setinde tüm bileşenlerinden daha iyi, diğer iki veri setinde ise EM hariç diğer bileşenlerinden daha iyi sonuçlar vermiştir. HPO-BBBC ise hem kalite testi fonksiyonlarında hem de sınıflandırmada bileşenlerine göre daha başarılı sonuçlar vermiştir.

Bu tezin literatüre katkıları genel olarak şu şekilde özetlenebilir:

- Optimizasyon, veri madenciliği ve sezgisel yöntemler hakkında temel bilgiler sunmaktadır.
- Optimizasyon ve veri madenciliği problemlerinin çözümüne yönelik fizik tabanlı iki yeni hibrit algoritma geliştirilmiştir.
- Geliştirilen algoritmalar için hem küresel optimizasyon problemlerinde hem de veri madenciliği tekniklerinden olan sınıflandırma problemleri üzerinde testleri yapılmıştır.
- Tez kapsamında geliştirilen hibrit algoritmalar, çözüm kalitesi, çalışma süresi ve yakınsama hızları bakımından mevcut algoritmaların performanslarını arttırmıştır.

1.4. Tezin Yapısı

Bu tez çalışması sekiz temel bölümden oluşmaktadır. Birinci bölümde, optimizasyon, sezgisel optimizasyon algoritmaları, hibritleme ve veri madenciliği hakkında temel bilgiler verilmiş, literatürde bulunan bazı hibrit yöntemlere değinilmiş ve tezin amacı açıklanmıştır.

İkinci bölümde, optimizasyon konusu detaylı bir biçimde anlatılmış, literatürdeki bazı fizik tabanlı sezgisel algoritmalar incelenmiş ve hibrit algoritmalar hakkında genel bilgiler verilmiştir.

Üçüncü bölümde, veri madenciliği ve veri madenciliği teknikleri hakkında ayrıntılı bilgi verilmiştir.

Dördüncü ve Beşinci bölümde, bu tez çalışmasında önerilen EM-PSO-DE ve HPO-BBBC algoritmaları ve bileşenleri hakkında ayrıntılı bilgi verilmiştir.

Altıncı bölümde, deneysel çalışmalarda kullanılan kalite testi fonksiyonları ve sınıflandırma veri setleri hakkında ayrıntılı bilgi verilmiştir.

Yedinci bölümde, önerilen hibrit yöntemlerin kalite testi fonksiyonlarından ve sınıflandırma veri setlerinden elde edilen karşılaştırmalı test sonuçları verilmiştir.

Sekizinci bölümde, tezde yapılan çalışmalar ve elde edilen sonuçlar değerlendirilmiş, ileriye dönük çalışmalar için önerilerde bulunulmuştur.

2. OPTİMİZASYON HAKKINDA GENEL BİLGİLER

“En iyileme” anlamına gelen optimizasyon genel bir ifade ile, bir problemin alternatif çözümleri arasından en iyi olanını seçme işlemi şeklinde tanımlanabilir. Daha teknik bir ifade ile bir problemin karar değişkenlerine bağlı olarak en uygun çözümünü bulmaya optimizasyon denir. Böyle bir problemde en uygun çözümler, aslında kastedilen herhangi bir kıstasın maksimize veya minimize edilmesidir. Söz konusu kıstas bir fonksiyon ile temsil edilir ve bu fonksiyona amaç fonksiyonu adı verilir. Karar değişkenlerini x vektörüyle temsil edecek olursak amaç fonksiyonumuz ise $f(x)$ olur [45]. Buna göre:

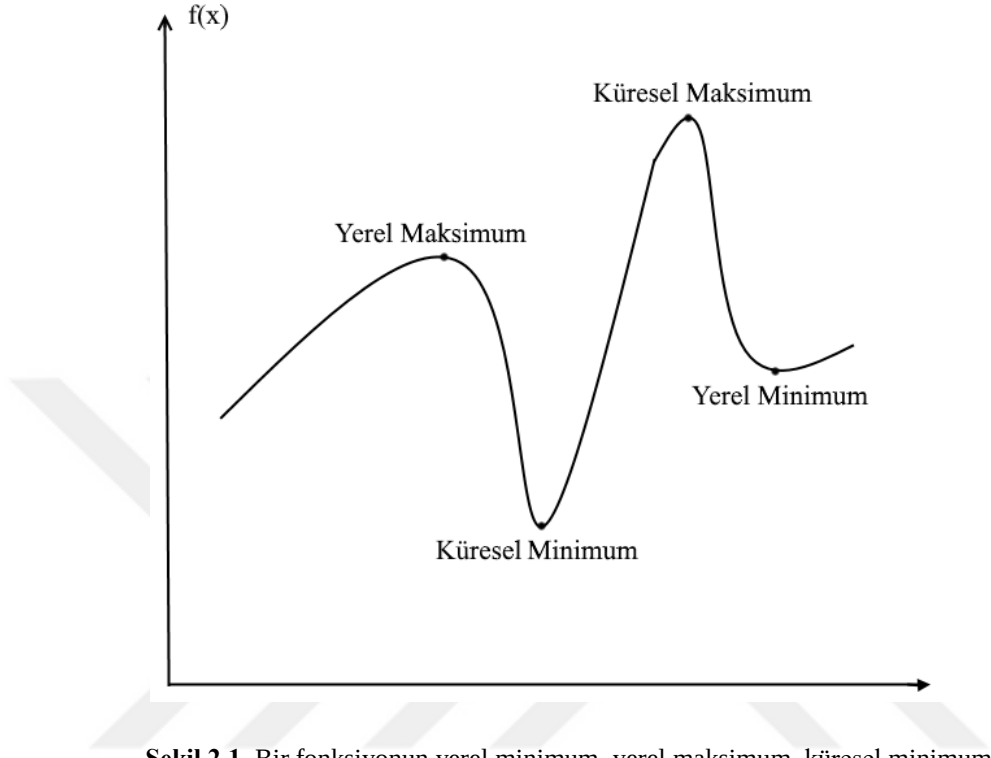
n değişkenli $x=(x_1, x_2, \dots, x_n)$ vektörünü tanımlayalım. Buna göre amaç fonksiyonu $f(x)$ denklem (2.1)’de ifade edildiği gibidir [46]:

$$f(x) = f(x_1, x_2, \dots, x_n) \quad (2.1)$$

Amaç fonksiyonu herhangi bir optimizasyon probleminde karar değişkenlerinin durumunun değerlendirilmesi için kullanılan fonksiyondur. Girişleri karar değişkenleri olan fonksiyonun çıkış değeri, çözümün uygunluğunu değerlendirmek için kullanılır. Yani bu fonksiyon karar değişkenlerinin amaç üzerindeki etkilerini gösterir [47]. Fonksiyonu minimum yapan karar değişkenlerinin bulunması işlemine minimizasyon, fonksiyonu maksimum yapan karar değişkenlerinin bulunması işlemine ise maksimizasyon adı verilir. Bazı problemlerde birden fazla amaç fonksiyonu bulunabilir. Bu tür problemlere çok amaçlı (multi-objective) optimizasyon problemleri denir.

Problem için sınırlamaları sağlayan olası tüm çözümlerin oluşturduğu bölge, araştırma yapılabilecek uygun çözüm bölgesi veya arama uzayı olarak adlandırılır. Örneğin bir $f(x)$ amaç fonksiyonuna sahip minimizasyon problemini ele alalım. Bir çözüm arama uzayı içindeki tüm muhtemel çözümler arasında en uygun çözümü sağlıyorsa $f(x)$ fonksiyonu küresel minimuma sahiptir. Eğer bu durum, arama uzayı içinde dar bir komşuluğa sahip muhtemel çözümler için sağlanıyorsa $f(x)$ fonksiyonu yerel minimuma sahiptir [46]. Benzer durum maksimizasyon problemleri için de geçerlidir. Bu durumda isimlendirme küresel maksimum ve yerel maksimum şeklinde olacaktır. Şekil 2.1’ de $f(x)$ fonksiyonu için yerel

minimum, yerel maksimum, küresel minimum ve küresel maksimum noktaları görülmektedir.



Şekil 2.1. Bir fonksiyonun yerel minimum, yerel maksimum, küresel minimum ve küresel maksimum noktaları.

Amaç fonksiyonunda x ile ilgili herhangi bir sınırlamanın olmadığı problemler *sınırlamasız*, x ile ilgili sınırlamaların olduğu problemler ise *sınırlamalı* optimizasyon problemleridir. Bir optimizasyon problemi lineer amaç ve sınırlama fonksiyonlarına sahip ise bu problem *doğrusal* programlama problemi, fonksiyon nonlineer ise *doğrusal olmayan* programlama modeli olarak adlandırılır. Bir fonksiyonun değişkenleri sürekli değerler alıyor ise bu tür problemlere *sürekli* optimizasyon problemleri denir [46]. Değişkenler sürekli olabileceği gibi ayrık değerler de alabilir. Değişkenler; verilen bir kümeden belirli değerler, sadece tam sayı değerleri veya ikili değerler alabilir. Bu durumda problem *ayrık* optimizasyon problemi olarak adlandırılabilir [47].

2.1. Optimizasyon Yöntemleri

Optimizasyon yöntemleri, optimizasyon problemlerini çözebilen teknikler veya algoritmalarıdır. Optimizasyon yöntemleri temel olarak iki ana sınıfa ayrılabilir. Bu sınıflar; kesin çözümü bulan *klasik yöntemler* ve yaklaşık optimum çözümü bulan *sezgisel algoritmalarıdır* [48].

2.1.1. Klasik Yöntemler

Klasik yöntemlerin hızlılık ve tam sonuç verme gibi avantajları bulunmaktadır. Klasik yöntemler, bir grubu detaylı arama fikrini temel almaktadır. Basit detaylı aramada, tüm çözümler kontrol edilerek en iyi çözüm bulunmaya çalışılır. Detaylı arama kullanan yöntemler her zaman optimum çözümü bulabilir. Klasik yöntemlerin birçok dezavantajı da vardır. Bu dezavantajlar genel olarak şu şekildedir [48]:

- Klasik yöntemlerde amaç fonksiyonunun tam olarak matematiksel bir şekilde ifade edilmesi gerekmektedir. Fakat gerçek dünyada, problemlerin çoğu için matematiksel bir model bulunmamaktadır. Bu nedenle, klasik yöntemlerin uygulanabileceği problem türü az sayıdadır.
- Genellikle ilgilenilen probleme özgüdürler. Yani bir yöntem bir problem üzerinde çok başarılı performans gösterirken, başka bir problem üzerinde aynı performansı sergilemeyebilir.
- Çoğunlukla küçük boyutlu problemler üzerinde uygulanmaktadır. Problem boyutunun artmasıyla birlikte klasik yöntemlerin hesaplama maliyetleri de fazlasıyla artmaktadır.
- Bu yöntemlerdeki bir diğer eksiklik ise farklı tipte karar değişkenlerinin, amaç ve sınırlayıcıların olması durumunda, genel çözüm stratejileri sunamamalarıdır. Yani çoğu klasik yöntem belirli tipteki amaç fonksiyonu ya da sınırlayıcıların olduğu modelleri çözmektedir. Ancak yönetim bilimi, bilgisayar, mühendislik gibi birçok farklı alandaki optimizasyon problemleri, farklı tipteki karar değişkenleri, amaç fonksiyonu ve sınırlayıcıları gerektirir [2, 49].

2.1.2. Sezgisel Algoritmalar

Gerçek yaşamda problemlerin çoğunda çözüm uzayı tüm çözümlerin değerlendirilemeyeceği kadar büyüktür ve matematiksel bir model çıkarmak zordur. Klasik yöntemlerin bir önceki alt başlıkta belirtilen dezavantajlarından dolayı, bu tür problemlere uygulanması oldukça zaman almakta veya hiç uygulanamamaktadır. Bunun için kabul edilebilir bir sürede çözümlerin değerlendirilerek iyi bir çözümün bulunması gerekmektedir. Böyle problemler için kabul edilebilir bir sürede çözümlerin değerlendirilmesiyle aslında tüm çözüm uzayında “bazı çözümlerin” değerlendirilmesi aynı anlama gelmektedir. Böyle bir durumda, sezgisel algoritmalarla ihtiyaç duyulmaktadır. Sezgisel algoritmalar, karmaşık bir problem için makul sürede deneme yanılma yolu ile kabul edilebilir çözümler sunmayı amaçlamaktadır. Çünkü ilgili problemin karmaşıklığı mümkün olan her çözümü ve kombinasyonu aramayı imkânsız hale getirir. Bu yüzden kabul edilebilir bir zaman ölçeğinde iyi uygulanabilir çözüm bulmak amaçlanır. Sezgisel algoritmalar, en iyi çözümü bulacaklarını garanti etmez fakat en kısa zamanda en makul çözümü ortaya koyarlar [5, 45]. Sezgisel algoritmalarla ihtiyaç duyulmasının sebepleri aşağıda belirtildiği gibidir [46, 47]:

- Sezgisel algoritmalar, yapılarında fazla bir değişikliğe ihtiyaç duymaksızın uygulanacak probleme göre uyarlanabilir.
- Basittirler ve karmaşık matematiksel hesaplamalara ihtiyaç duymazlar.
- Sezgisel algoritmalar, problemin nasıl çözüleceğini bilmeden problemi çözebilirler. Bu nedenle uzman deneyimi olmayan problemlerin çözümleri için uygulanabilirler.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar vericiler için daha basit olabilirler.
- Sezgisel algoritmalar, kesin çözümü bulmanın bir parçası olarak kullanılabilirler.
- Genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) matematik formülleriyle yapılan tanımlamalarda ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

Sezgisel algoritmaların optimizasyon süreci temel olarak şu şekilde işler: Sezgisel algoritmalar ilk olarak bir başlangıç popülasyonu ile başlarlar. Popülasyondaki üyeler genellikle arama uzayı sınırları içerisinde rastgele bir biçimde ilk değerlerini alırlar. Daha sonra mevcut popülasyonun tamamı veya bir kısmı, yeni nesil üyeler elde etmek için seçilir.

Yeni nesil üyelerin üretilme işlemi algoritmalar arasında farklılık gösterebilir. Örneğin; ebeveyn üyeler arasında çaprazlama yapılarak yeni nesil üyeler elde edilebilir. Diğer taraftan, yeni üyeler diğer üyelerden bağımsız olarak üzerinde belirli değişiklikler yapılarak da oluşturulabilir. Ayrıca yeni üyeler daha önceden elde edilen bilgiler kullanılarak da oluşturulabilir. Yeni üyeler belirli kurallar çerçevesinde popülasyondaki ebeveyn üyelerin bir kısmının veya tamamının yerini alabilirler. Bahsedilen bu süreçler belirlenen maksimum iterasyon sayısına veya istenilen bir kritere sahip oluncaya kadar döngüsel olarak devam eder. Optimizasyon süreci sonunda elde edilen sonuç, optimum veya istenilen kriteri sağlayan optimuma yakın bir değer olabilir [50].

Sezgisel algoritmalarda keşif (exploration) ve sömürü (exploitation) olmak üzere iki temel bileşen bulunmaktadır:

- Keşif; arama uzayının tamamının araştırılması ile farklı aday çözümler üretilmesidir.
- Sömürü ise o anki aramada en iyi aday çözümün bulunduğu arama bölgesine odaklanması ve oradaki bilginin kullanılması anlamına gelmektedir.

Sezgisel algoritmaların çoğu, keşif ve sömürü arasındaki dengeyi oluşturarak en iyi çözüme ulaşmayı hedefler [51].

Algoritma tek bir çözümünden başlayıp bunu operatörlerle ilerletiyorsa bunlara *tek noktalı* yöntemler denir. Tabu arama, ısıtma işlemi gibi bütün yerel arama tabanlı algoritmalar bu gruba girer. Çok noktadan yani bir popülasyon üzerinden çözüme başlanıp farklı noktalarla optimizasyon yapılıyorsa bunlar, *çok noktalı* ya da popülasyon tabanlı yöntemlerdir. Bunlara GA ve PSO örnek olarak verilebilir [2, 49].

Bazı sezgisel algoritmalar problemin gösterimini gerçekleştirirken amaç fonksiyonunu sabit tutar ve *sabit amaç fonksiyonlu* olarak adlandırılırlar. Bazıları da amaç fonksiyonlarını değiştirir ve *değişken amaç fonksiyonlu* olarak adlandırılırlar. Değiştirmekteki amaç, yerel minimumdan kurtulmaktır [2, 49].

Sezgisel algoritmaların çoğu tek bir komşuluk yapısında çalışır ve *tek komşuluk yapılı* olarak sınıflandırılabilir. Bazı sezgisel algoritmalar da değişken arama işlemini düzenli bir biçimde değiştirerek birden çok yerel arama metodu ile diğer çözüm alanlarına ulaşmaya çalışır ve *değişken komşuluk yapılı* şeklinde sınıflandırılabilir [2, 49].

Sezgisel algoritmalar çalışma sırasında elde edilen durumların bir kısmını veya tamamını hafızaya kaydedebilir. Kaydedilen bu bilgiler, algoritmanın bir sonraki adımlarında vereceği kararlarda etkili olacak şekilde kullanılabilir. Eğer bu şekilde sezgisel algoritmalar çalışırken

daha önceki durumlar ya da en iyi durumlar hatırlanıyorsa *hafızalı*, hatırlanmıyorsa *hafızasız* şeklinde sınıflandırılabilir. Örneğin PSO hafızalı iken GA hafızasızdır [2, 49].

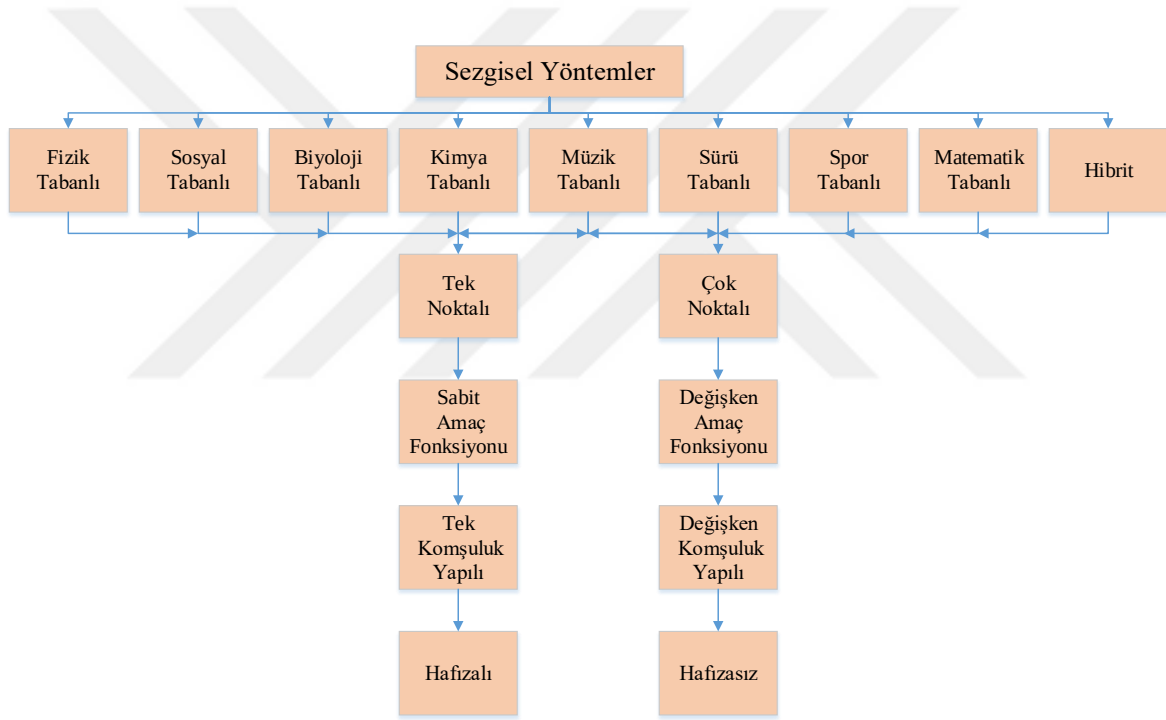
Sezgisel algoritmalar aşağıdaki kriterler göz önüne alınarak değerlendirilebilmektedir [45]:

- *Çözüm Kalitesi ve Hesaplama Zamanı*: Bir algoritmanın etkinliğinin değerlendirilmesi için önemli bir kriterdir. Bir algoritma, ayarlanabilir parametreler setine sahip olmalı ve bu parametreler kullanıcıya önemlilik açısından hesaplama maliyeti ile çözüm kalitesi arasında bir vurgulamanın yapılabilmesine imkân vermelidir. Diğer bir deyişle, çözüm kalitesi ile hesap zamanı arasındaki ilişki kontrol edilebilmelidir.
- *Kod Basitliği ve Gerçeklenebilirlik*: Algoritmanın prensipleri basit ve genel olarak uygulanabilir olmalıdır. Bu durum, algoritmaların uzman deneyimi olmadan ve problemin nasıl çözüleceğini bilmeden yeni problemlere kolaylıkla uygulanabilmesini sağlar.
- *Esneklik*: Algoritmalar modelde, sınırlamalarda ve amaç fonksiyonlarında yapılacak değişiklikleri kolayca karşılayabilmelidir.
- *Dinçlik*: Algoritma, başlangıç çözümüne bağlı olmaksızın her zaman kaliteli ve kabul edilebilir çözümleri sunabilme kabiliyetine sahip olmalıdır.
- *Basitlik ve Analiz Edilebilirlik*: Karmaşık algoritmalar, esneklik ve çözüm kalitesi açısından basit algoritmalarından daha zor analiz edilebilmektedir. Algoritma kolayca analiz edilebilir olmalıdır.
- *Etkileşimli Hesaplama ve Teknoloji Değişimleri*: Algoritma içinde insan-makine etkileşimini kullanma fikri çoğu sistemde yaygın olarak gerçekleştirilmektedir. Herkesçe bilindiği gibi iyi bir kullanıcı ara yüzü herhangi bir bilgisayar sistemini veya algoritmayı daha çekici yapmaktadır. Bunun en önemli avantajı çözümlerin grafiksel olarak sergilenebilmesidir.

Sezgisel algoritmaların birçoğu doğadan esinlenilerek geliştirilmiştir. Sezgisel algoritmalar doğadaki esinlendikleri alana göre fizik tabanlı, sosyal tabanlı, biyoloji tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı, matematik tabanlı olmak üzere farklı kategorilere ayrılabilir. Bunların dışında hibrit algoritmalar da mevcuttur. Hibrit algoritmalar birden fazla algoritmanın bir arada kullanılması ile elde edilebilmektedir [2, 8]. Tüm bu yöntemler Şekil 2.2' de görülmektedir.

2.2. Fizik Tabanlı Sezgisel Optimizasyon Algoritmaları

Karmaşık problemleri çözmek için doğal fenomenleri modellemek ve taklit etmek, son yıllarda aktif bir araştırma alanı olmuştur. Literatüre bakıldığında araştırmacılar tarafından çeşitli fizik tabanlı sezgisel optimizasyon algoritmaları önerilmiştir. Bu algoritmaların bir kısmı yer çekimi, kütle çekimi, manyetizma, karadelikler vb. fiziksel yasa ve teorilerden ya da dünyadaki suyun döngüsü, suyun akışı, erozyon vb. fiziksel olaylardan esinlenilerek geliştirilmiştir [52-54]. Bu bölümde bazı fizik tabanlı sezgisel optimizasyon algoritmalarının nereden esinlenilerek geliştirildikleri ve algoritmaların temel adımları hakkında bilgi verilecektir.



Şekil 2.2. Sezgisel Yöntemler

2.2.1. Yerçekimsel Arama Algoritması

Rashedi ve arkadaşları [11] tarafından önerilen yerçekimsel arama algoritması, Newton' un yerçekimi yasasından esinlenilerek geliştirilmiştir. Algoritmada popülasyondaki bireyler kütle olarak adlandırılmaktadır. Yerçekimi yasasına göre kütleler yerçekimi kuvveti tarafından birbirlerini çekerler. Bu kuvvet, kütlelerin büyüklüğü ile doğru orantılı ve kütleler arasındaki mesafe ile ters orantılıdır. Algoritmada bir aday çözüm ne kadar iyi ise kütlesi o

kadar büyüktür. Algoritma sonunda en büyük kütle problemin çözümü olarak kabul edilir. Algoritmanın temel adımları şu şekilde özetlenebilir:

ADIM 1: Rastgele bir biçimde başlangıç popülasyonunu oluştur.

ADIM 2: Her kütle için uygunluk değerini hesapla.

ADIM 3: Popülasyondaki en iyi (best), en kötü (worst), yerçekimi sabiti (G) ve kütlelerin büyüklüklerini (M) güncelle.

ADIM 4: Farklı yönlerden gelen toplam kuvveti hesapla.

ADIM 5: Hız ve ivme değerlerini hesapla.

ADIM 6: Popülasyondaki kütleleri, hesaplanan hız ve ivme değerlerine göre hareket ettir.

ADIM 7: Eğer bitim şartı sağlanmıyorsa ADIM 2' ye git.

ADIM 8: En iyi çözümü getir.

2.2.2. İyon Hareketi Algoritması

İyon hareketi algoritması Javidy ve arkadaşları [55] tarafından iyonların hareketlerinden esinlenilerek geliştirilmiştir. Yüklü parçacıklara iyon denmektedir. Negatif yüklü iyonlar anyon, pozitif yüklü iyonlar ise katyondur. Aynı yüklü iyonlar birbirlerini iterken, zıt yüke sahip iyonlar birbirlerini çeker. Algoritmanın temel mantığı, bu itme ve çekme kuvvetleri ile iyonların arama uzayında hareket ettirilmesidir. Algoritmada popülasyondaki bireyler anyonlardan ve katyonlardan oluşmaktadır. Algoritmanın temel adımları şu şekildedir [55, 56]:

ADIM 1: Eşit sayıda anyon ve katyonları gelişigüzel oluştur.

ADIM 2: Anyon ve katyonların uygunluk değerlerini hesapla.

ADIM 2.1: En iyi ve en kötü anyon ve katyonu belirle.

ADIM 2.2: Anyonların katyonlar ve katyonların anyonlar tarafından çekilme kuvvetlerini hesapla.

ADIM 2.3: Anyon ve katyonların pozisyonlarını hesapla.

ADIM 3: En kötü iyonların ortalama uygunluğu, en iyi iyonların uygunluğundan eşit veya küçük ise, anyon ve katyonları en iyi iyonların etrafına dağıt. Ayrıca çok küçük bir olasılık ile (örneğin 0.05) anyon ve katyonları yeniden oluştur.

ADIM 4: Eğer bitim şartı sağlanmıyorsa ADIM 2' ye git.

ADIM 5: En iyi çözümü getir.

2.2.3. Su Döngüsü Optimizasyon Algoritması

Eskandar ve arkadaşları [12] tarafından önerilen su döngüsü optimizasyon algoritması, suyun atmosferden yağmurlar ile yeryüzüne, dere ve nehirlerle deniz veya okyanuslara sonrasında tekrardan buharlaşarak atmosfere ulaşması şeklinde devam eden su döngüsünden esinlenilerek geliştirilmiştir. Algoritmada popülasyondaki bireyler yağmur damlası olarak adlandırılır. Algoritmanın temel adımları ise [12, 57]:

ADIM 1: Gelişigüzel değerlere sahip yağmur damlaları ile N_{pop} sayıda N_{var} boyutlu başlangıç popülasyonunu oluştur.

ADIM 2: Popülasyondaki bütün yağmur damlalarının uygunluk değerlerini hesapla.

ADIM 3: Uygunluk değeri iyi olan N_{sr} adet birkaç yağmur damlası “nehir” olarak seç. Uygunluk değeri en iyi olan nehri ise “deniz” olarak belirle. Geri kalan N_d sayıdaki bireyi ise “dere” olarak adlandır.

ADIM 4: Nehirleri denize bağla. Nehirlere ise hangi derelerin bağlanacağını belirle.

ADIM 5: Dereleri bağlı olduğu nehirlerle doğru, nehirleri ise denize doğru hareket ettir.

ADIM 6: Hareket sonrası eğer bir dere bağlı olduğu nehirden daha iyi bir uygunluk değerine ulaşmışsa, o dere ile bağlı olduğu nehri yer değiştir. Benzer şekilde eğer bir nehir, denizden daha iyi bir uygunluk değerine ulaşmışsa o nehir ile denizi yer değiştir.

ADIM 7: Nehirler ile deniz arasındaki uzaklıkları kontrol et. Eğer bir nehir denize d_{max} değerinden daha küçük bir mesafede ise, nehir ve denizin birleştiği kabul edilir ve buharlaşma için gerekli olan şart sağlanmış olur. Buharlaşma işleminde, deniz dışındaki diğer tüm bireyler gelişigüzel bir şekilde tekrar oluştur ve uygunluk değerlerine göre nehirleri ve dereleri tekrar belirle.

ADIM 8: Eğer bitim şartı sağlanmamış ise ADIM 5’ e dön.

ADIM 9: Denizi çözüm olarak getir.

2.2.4. Karadelik Arama Algoritması

Hatamlou [58] tarafından önerilen karadelik arama algoritması, ilk 18. yüzyılda ortaya atılan karadelik kavramından esinlenilerek geliştirilmiştir. Karadelikler devasa büyüklükteki yıldızların içlerine çökmeleri sonucunda oluşmaktadır. İçine çöken bu büyük kütleli yıldız, çok küçük bir alanda devasa bir yoğunluğa sahiptir. Bu nedenle çevresindeki her şeyi

kendisine çeken ve yutan güçlü bir yerçekimi alanına sahip olur. Işık bile bu güçlü yerçekiminden kurtulamaz ve bu yüzden karadelikler karanlıktır ve dış dünya tarafından görülemez. Karadeliklerin küre şeklinde belli bir yarıçapa sahip sınırları vardır. Bu sınır içine giren her şey karadelik tarafından yutulur. Algoritmada popülasyondaki bireyler, yıldız olarak adlandırılır. En iyi uygunluk değerine sahip yıldız ise karadelik olarak seçilir. Algoritmanın temel adımları aşağıda belirtildiği gibidir:

ADIM 1: Gelişigüzel bir biçimde başlangıç popülasyonundaki yıldızları oluşturun.

ADIM 2: Her yıldızın uygunluk değerini hesapla.

ADIM 3: En iyi uygunluk değerine sahip olan yıldızı karadelik olarak belirle.

ADIM 4: Yıldızların konumlarını güncelle.

ADIM 5: Eğer bir yıldız karadelikten daha iyi bir konuma ulaşmış ise, karadelik ile o yıldızın yer değiştir.

ADIM 6: Karadelik'in sınır yarıçapını hesapla. Eğer bir yıldız ile karadelik arasındaki uzaklık yarıçaptan daha küçük bir değere sahip olmuşsa, o yıldız gelişigüzel bir biçimde yeniden oluşturun.

ADIM 7: Eğer bitim şartı sağlanmıyorsa ADIM 2'ye git.

ADIM 8: En iyi çözümü getir.

2.2.5. Yüklü Sistem Arama Algoritması

Kaveh ve Talatahahari [59] tarafından önerilen yüklü sistem arama algoritması, Coulomb kanunu ve Newton'un hareket kanunundan esinlenilerek geliştirilmiştir. Algoritmada popülasyondaki bireyler yüklü parçacık olarak adlandırılır. Yüklü parçacıklar yükleri oranında ve aralarındaki mesafenin tersi bir oranda birbirlerine bir kuvvet uygularlar. Daha sonra yüklü parçacıklar üzerindeki bu kuvvet aracılığı ile yeni konumuna hareket ettirilir. Algoritmanın temel adımları aşağıdaki gibidir:

ADIM 1: Başlangıç parametrelerini belirle ve başlangıç popülasyonunu gelişigüzel bir biçimde oluşturun.

ADIM 2: Parçacıkların uygunluk değerlerini, en iyi ve en kötü parçacığı ve parçacıkların yüklerini hesapla.

ADIM 3: En iyi birkaç yüklü parçacığı hafızada sakla.

ADIM 4: Yüklü parçacıklar arasındaki çekim kuvvetlerini hesapla.

ADIM 5: Yüklü parçacıkların hızını hesapla ve o hıza göre hareket ettir.

ADIM 6: Yüklü parçacıkların yeni konumlarının arama uzayı sınırları içerisinde olup olmadığını kontrol et.

ADIM 7: Yüklü parçacıkların uygunluk değerlerini yeniden hesapla ve uygunluk değerlerine göre sırala.

ADIM 8: Eğer hafızadaki yüklü parçacıklardan daha iyi uygunluk değerine sahip parçacıklar varsa hafızayı güncelle.

ADIM 9: Eğer bitim şartı sağlanmıyorsa ADIM 4' e git.

ADIM 10: En iyi çözümü getir.

2.2.6. Yapay Fizik Optimizasyon Algoritması

Xie ve arkadaşları [60, 61] tarafından önerilen yapay fizik algoritması, yerçekimsel arama algoritmasına benzer şekilde yerçekimi yasasından esinlenilerek geliştirilmiştir. Algoritmada parçacıklar kendinden küçük kütleli parçacıkları çekerken, kendinden büyük kütleli parçacıkları ise iter. Algoritmanın temel adımları şu şekildedir.

ADIM 1: Başlangıç parametrelerini ayarla ve başlangıç popülasyonunu gelişigüzel bir biçimde oluştur.

ADIM 2: Parçacıkların uygunluk değerlerini hesapla, en iyi ve en kötü parçacığı belirle. Parçacıkların kütlelerini uygunluk değerlerine göre hesapla.

ADIM 3: Toplam kuvveti hesapla.

ADIM 4: Parçacıkların hızlarını güncelle.

ADIM 4: Parçacıkların konumunu güncelle.

ADIM 5: Eğer bitim şartı sağlanmıyorsa ADIM 2' ye git.

ADIM 6: En iyi çözümü getir.

2.2.7. Parçacık Çarpışma Algoritması

Sacco ve arkadaşları [62, 63] tarafından önerilen parçacık çarpışma algoritması, nükleer fizikteki parçacıkların çarpışmasından, özellikle de parçacıkların çarpışması sırasında meydana gelen emilim ve saçılma olaylarından esinlenilerek geliştirilmiştir. Algoritmada, bir parçacık düşük uygunluk değerlerine sahip bir bölgede ise o parçacık diğer bölgelere

saçılırken, yüksek uygunluk değerlerine sahip bölgelerdeki parçacık ise emilime uğrar. Algoritmanın temel adımları aşağıda belirtildiği gibidir:

ADIM 1: Rastgele bir biçimde başlangıç parçacığını (x) oluştur.

ADIM 2: Rastgele bir ax değeri ile x^i parçacığını oluştur: $x^i = x + ax$

ADIM 2.1: Eğer x^i parçacığının uygunluk değeri x parçacığından daha iyi ise $x = x^i$ şeklinde parçacıkları yer değiştir ve ADIM 4' e (KEŞİF) git, yoksa ADIM 6' ya (SAÇILMA) ya git.

ADIM 2.2: Eğer maksimum iterasyon sayısına ulaşılmamış ise ADIM 2.1' e dön.

ADIM 3: Çözümü döndür.

ADIM 4: KEŞİF

ADIM 4.1: $x^i = x + ax$ şeklinde yeni bir parçacık oluştur.

ADIM 4.2: Eğer x^i parçacığının uygunluk değeri x parçacığından daha iyi ise $x = x^i$ şeklinde parçacıkları yer değiştir.

ADIM 4.3: Eğer maksimum iterasyon sayısına ulaşılmamış ise ADIM 4.1' e dön.

ADIM 5: Çözümü döndür.

ADIM 6: SAÇILMA

ADIM 6.1: Parçacığın saçılma olasılığını hesapla.

ADIM 6.2: 0 ile 1 arasında rastgele bir biçimde k değerini oluştur. Eğer saçılma olasılığı k değerinden büyük ise x parçacığını rastgele bir biçimde oluştur, değilse ADIM' 4 e dön.

ADIM 7: Çözümü Döndür.

2.2.8. Büyük Çöküş Algoritması

Kripka ve Kripka [64] tarafından önerilen büyük çöküş algoritması, evrendeki yerçekimi enerjisinin büyük patlama ile evrenin oluşumundan açığa çıkan enerjiden daha büyük bir seviyeye ulaşması sonucunda, evrenin büyümesi duracak ve ardından evren sonsuz yoğunlukta tek bir noktaya çökecektir fikrinden esinlenilerek geliştirilmiştir. Algoritma, diğer algoritmalara benzer şekilde rastgele üretilen başlangıç popülasyonu ile başlar. Fakat algoritma adımları devam ettikçe parçacıklar evrenin merkezine doğru çöker ve

popülasyonda tek bir parçacık kalana kadar bu işlem devam eder. Algoritmanın temel adımları aşağıda belirtildiği gibidir:

ADIM 1: N adet sayıda rastgele bir biçimde oluşturulmuş parçacıklar ile başlangıç popülasyonunu oluştur.

ADIM 2: Her bir parçacığın kütlesini uygunluk fonksiyonuna göre hesapla ve bunları küçükten büyüğe sırala.

ADIM 3: Parçacıklar tarafından birbirlerine etki eden yerçekimi kuvveti F_g ' yi hesapla.

ADIM 4: F_g ' nin merkezini hesapla.

ADIM 5: Parçacıkların konumunu, parçacık ile F_g ' nin merkezi arasındaki uzaklığın küçük bir kesiti olacak şekilde değiştir.

ADIM 6: Eğer bir parçacık başka bir parçacığa belirlenen çok küçük bir değerden daha fazla yaklaşmışsa, popülasyon sayısını 1 azalt.

ADIM 7: Yerel bir arama gerçekleştirmek için her parçacığı rastgele bir biçimde hareket ettir.

ADIM 8: Parçacıkların kütlelerini yeniden hesapla ve bunları büyükten küçüğe sırala.

ADIM 9: Eğer popülasyon sayısı 1' den büyükse yada maksimum iterasyon sayısına ulaşılmamış ise ADIM 3' e dön.

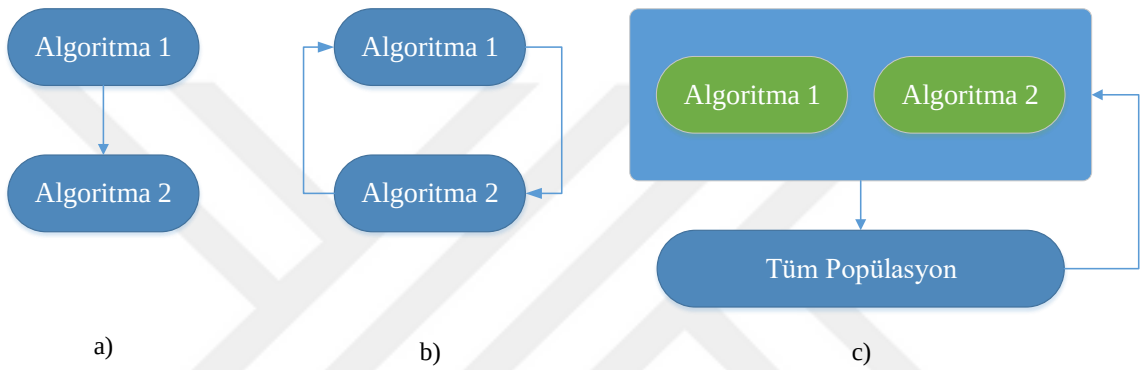
ADIM 10: Çözümü döndür.

2.3. Hibrit Algoritmalar

Hibritleme, bir algoritmanın performansını arttırmak için sıkça kullanılan bir yöntemdir. Hibritleme, iki veya daha fazla algoritmanın özelliklerini tek bir algortmada birleştirmek olarak tanımlanabilir. Mevcut algoritmanın çözüm kalitesini veya performansını geliştirmek ise hibritlemenin temel amaçlarındandır [19, 20]. Hibrit algoritmalar, işbirlikçi (collaborative) ve bütünleyici (integrative) olmak üzere temel olarak iki sınıfa ayrılabilir [65].

İşbirlikçi hibritler, sıralı veya paralel olarak çalışan iki veya daha fazla algoritmanın kombinasyonunu içerir. Katkıda bulunan algoritmaların etkisi %50'ler civarında olup hemen hemen birbirine eşittir. Şekil 2.3' te işbirlikçi hibritlerdeki üç temel yaklaşım görülmektedir.

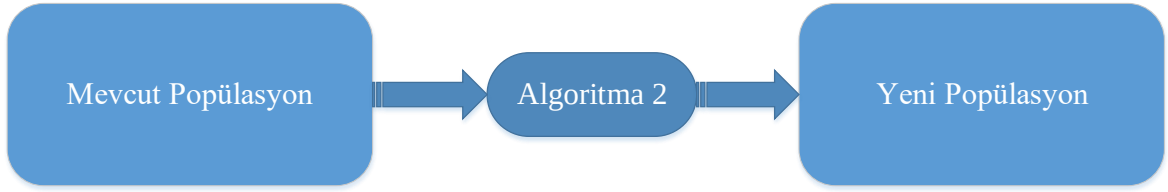
- Çok kademeli (Multi-stage): Bu yaklaşımda, algoritmalarından biri küresel arama gerçekleştirirken diğer algoritma yerel arama gerçekleştirmektedir.
- Sıralı (Sequential): Bu yapıda, her iki algoritma yakınsama kriterlerinden biri yerine getirilinceye kadar alternatif olarak çalıştırılır. Bir algoritma diğerinden sonra sıkı sıkıya uygulanır ve bilgi sadece bir yönde geçirilir. Kolaylık sağlamak için, bir sonraki algoritmaya geçmeden önce her iki algoritma benzer sayıda yineleme için çalıştırılır.
- Paralel: İki algoritma, aynı popülasyon üzerinde eş zamanlı olarak çalıştırılır.



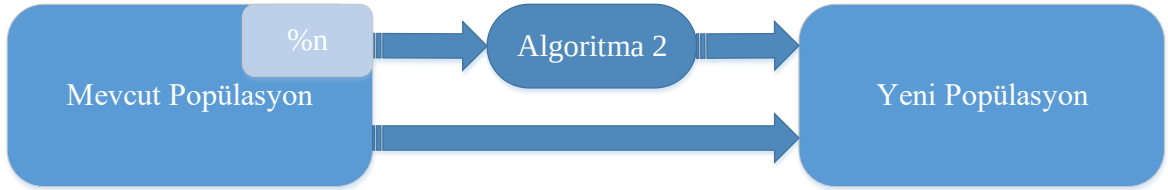
Şekil 2.3. İşbirlikçi hibritler, a) çok kademeli b) sıralı c) paralel

Bütünleyici hibritler, bir ana algoritmaya ikincil bir algoritmanın bütünleyici olarak eklenmesi ile oluşturulur. İkincil algoritmanın katkısı %10-%20' ler civarındadır ve ikincil bir algoritmadan bir manipülasyon operatörünün birincil bir algoritmaya dahil edilmesini içerir. GA' daki mutasyon operatörünün PSO içinde kullanılması örnek olarak verilebilir. Bu kategoride ise Şekil 2.4' te görüldüğü gibi iki olası yaklaşım bulunmaktadır.

- Tam Manipülasyon (Full manipulation): Tüm popülasyon her iterasyonda manipüle edilir. Bu tür bir işlem, mevcut kaynak kodu ile aynı doğrultuda, genellikle bir alt program / alt fonksiyon olarak entegre edilebilir.
- Kısmi Manipülasyon (Partial manipulation): Tüm popülasyonun sadece bir kısmı manipüle edilir.



a)



b)

Şekil 2.4. Bütünleyici hibritler, a) tam manipülasyon b) kısmi manipülasyon

3. VERİ MADENCİLİĞİ HAKKINDA GENEL BİLGİLER

Bilişim alanındaki gelişmeler ve bilişim sistemlerinin yaşamımızın birçok alanına girmesiyle birlikte yaptığımız her işlem sayısal ortamda kaydedilmektedir. Örneğin; internette, bankada, hastanede veya markette yaptığımız her işlem artık veri tabanlarında tutulmaya başlanmıştır. Bunun sonucunda elde edilen veriler sayılamayacak kadar büyüktür. Toprak örtüsü altındaki değerli madenleri gün yüzüne çıkarmak madencilik ile sağlanır. Benzer şekilde büyük veriler arasındaki yararlı bilgilere ulaşmak ise “veri madenciliği” kavramının ortaya çıkmasına sebep olmuştur.

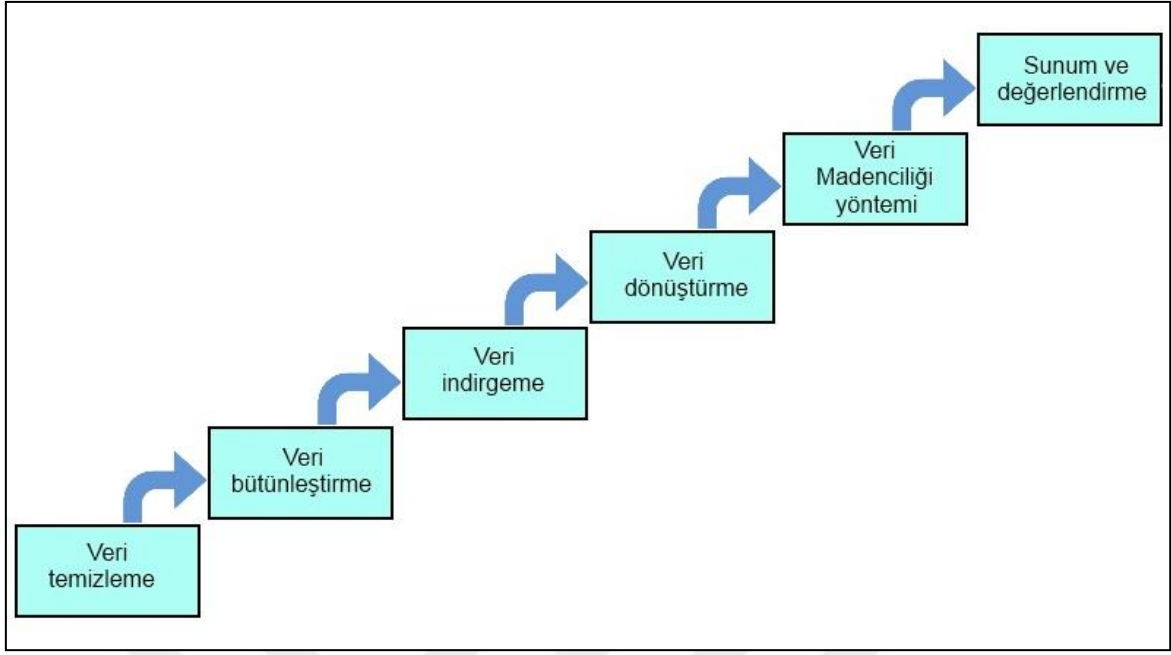
Veri madenciliği konusunda çeşitli tanımlamalar yapılabilir. Veri madenciliği, büyük ölçekli veriler arasından “değeri olan” bir bilgiyi elde etme işlemidir. Başka bir tanıma göre ise veri madenciliği; bir kurumda üretilen tüm verilerin belirli yöntemler kullanılarak, var olan bilgiye veya gelecekte ortaya çıkabilecek gizli bilgilere ulaşma işlemidir [66].

3.1. Veri Madenciliği Süreci

Veri madenciliği bir süreç olarak değerlendirilmelidir. Bu süreç altı adımdan oluşur:

- Veri temizleme
- Veri bütünleştirme
- Veri indirgeme
- Veri dönüştürme
- Veri madenciliği algoritmasını uygulama
- Sonuçları sunum ve değerlendirme

Bu adımlar Şekil 3.1’ de gösterilmiştir [66].



Şekil 3.1. Veri madenciliği sürecindeki adımlar

3.1.1. Veri Temizleme

Bazı uygulamalarda, üzerinde işlem yapılacak veriler eksik veya uygun olmayan değerlere sahip olabilir. Veri tabanındaki bu tutarsız veriler, gürültü olarak adlandırılır. Bu gibi durumlarda bu verilerin temizlenmesi gerekir. Bunun için aşağıdaki yöntemler kullanılabilir [66]:

- Eksik değer içeren kayıt, veri tabanından silinebilir.
- Kayıp değerler yerine genel bir sabit kullanılabilir.
- Diğer tüm verilerdeki değerlerin ortalaması, eksik değer yerine yazılabilir.
- Diğer tüm veriler yerine sadece bir sınıfa ait verilerdeki değerlerin ortalaması, eksik değer yerine yazılabilir.
- Verilere uygun bir tahmin yapılarak elde edilen değer, eksik değer yerine kullanılabilir.

3.1.2. Veri Bütünleştirme

Farklı veri tabanlarından alınan verilerin birlikte değerlendirilmeye alınabilmesi için farklı türdeki verilerin tek bir türe dönüştürülmesi yani bütünleştirilmesi gerekecektir [66].

3.1.3. Veri İndirgeme

Veri madenciliği uygulamalarında eğer çözümlemekten elde edilecek sonucun değişmeyeceğine inanılıyorsa, veri sayısı ya da değişkenlerin sayısı azaltılabilir. Veri indirgeme çeşitli biçimlerde yapılabilir [66]:

- Veri birleştirme veya veri küpü: Çözümlemeler sadece belirli boyutlara göre yapılabilir.
- Boyut indirgeme: Veriler arasında bir seçme işlemi yapılarak, gereksiz veriler veri tabanından silinebilir böylece boyut azaltma sağlanabilir.
- Veri sıkıştırma: Büyük veri kümeleri sıkıştırılarak daha az yer kaplamaları sağlanabilir.
- Örneklem: Büyük veri toplulukları yerine onları temsil edecek daha küçük veri toplulukları kullanılabilir.
- Genelleme: Verilerin tek tek değil genel kavramlarla ifade edilmesidir.

3.1.4. Veri Dönüştürme

Bazı durumlarda verileri, veri madenciliği çözümlemelerine aynen katmak uygun olmayabilir. Bu yüzden verinin, kullandığımız veri madenciliği tekniğinde kullanılabilecek hale dönüştürülmesi gerekir [66].

3.1.5. Veri Madenciliği Algoritmasını Uygulama

Veriyi hazır hale getirmek için yukarıdaki adımlardan uygun görülenleri yapılır. Veri hazır hale getirildikten sonra veri madenciliği algoritmaları kullanılabilir [66].

3.1.6. Sonuçlar Sunum ve Değerlendirme

Veri madenciliği algoritması, veriler üzerinde uygulandıktan sonra sonuçlar elde edilir ve bu sonuçlar değerlendirilir.

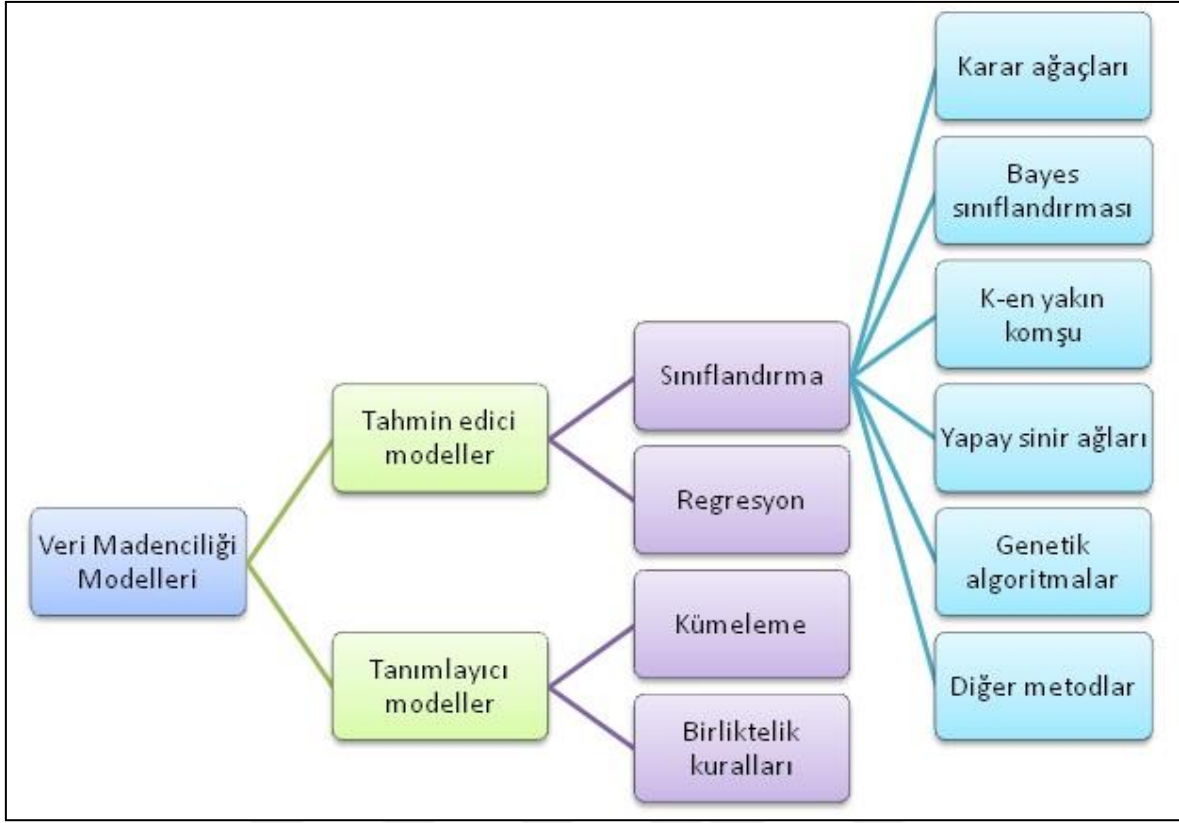
3.2. Veri Madenciliği Modelleri

Veri madenciliğinde kullanılan modeller, tahmin edici modeller ve tanımlayıcı modeller olmak üzere iki ana başlık altında toplanabilir.

Tahmin edici modellerde; sonuçları bilinen verilerden yola çıkarak bir model geliştirilir ve bu modelden yararlanarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerinin tahmin edilmesi amaçlanır. Örneğin; bir bankanın kredilerle ilgili veri tabanında müşterinin özellikleri ve krediyi geri ödeyip ödemediği bilgisini tutan veriler bulunur. Bu verilere uygun olarak kurulan modelle, daha sonra kredi talep eden müşterinin özelliklerine göre müşterinin krediyi geri ödeyip ödemeyeceğinin tahmini yapılabilir. Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. Veri madenciliği modellerini gördükleri işlevlere göre üçe ayırmak mümkündür [2, 67]:

- Sınıflandırma ve Regresyon
- Kümeleme
- Birlikte Kuralları

Veri madenciliğindeki modeller ve modellere göre gördükleri işlevleri Şekil 3.2’ de gösterilmiştir.



Şekil 3.2. Veri madenciliği modelleri

3.2.1. Sınıflandırma ve Regresyon

Sınıflandırma ve regresyon tahmin edici modellerdir. Sınıflandırma kategorik değerlerin, regresyon ise süreklilik gösteren değerlerin tahmininde kullanılır.

Sınıflandırma ve regresyon geleceğe ait veri tahmini yapmak ve önemli veri sınıflarını açıklayan modeller çıkarmak için kullanılır. Sınıflandırma modeli iki adımdan oluşur. İlk adımda sınıflandırma algoritması kullanılarak o sınıfa ait kurallar bulunur. İkinci adımda ise, sınıfı bilinmeyen farklı ve yeni veriler geldiğinde, bu verilerin ilk adımda elde edilen kurallara göre uygun sınıf ataması yapılır. Veri madenciliğinde genel olarak, sınıf etiketlerinin belirlenmesinde kullanılan tahmin işlemi sınıflandırmadır. Regresyon ise sürekli değerlerin belirlenmesinde kullanılan tahmin işlemidir [2,68].

3.2.2. Kümeleme

Kümeleme verilerin kendi aralarındaki benzerliklerin göz önüne alınarak gruplandırılmasıdır. Bu nedenle pek çok alanda kullanılabilir. Pazar arařtırmalarında, resim işleme, desen tanımlama ve uzaysal harita verilerinin analizlerinde kullanılmaktadır [66]. Kümeleme analizi, sınıflandırma ve regresyondan farklı olarak, bilinen sınıf etiketini dikkate almaksızın veri nesnelerini analiz etmektedir. Kümeler çeşitli etiketler oluşturmak için kullanılır. Aynı küme içindeki nesnelerin özellikleri birbirine benzerdir. Farklı kümelerdeki nesnelerin benzerlikleri ise çok düşüktür [2, 68].

3.2.3. Birliktelik Kuralları

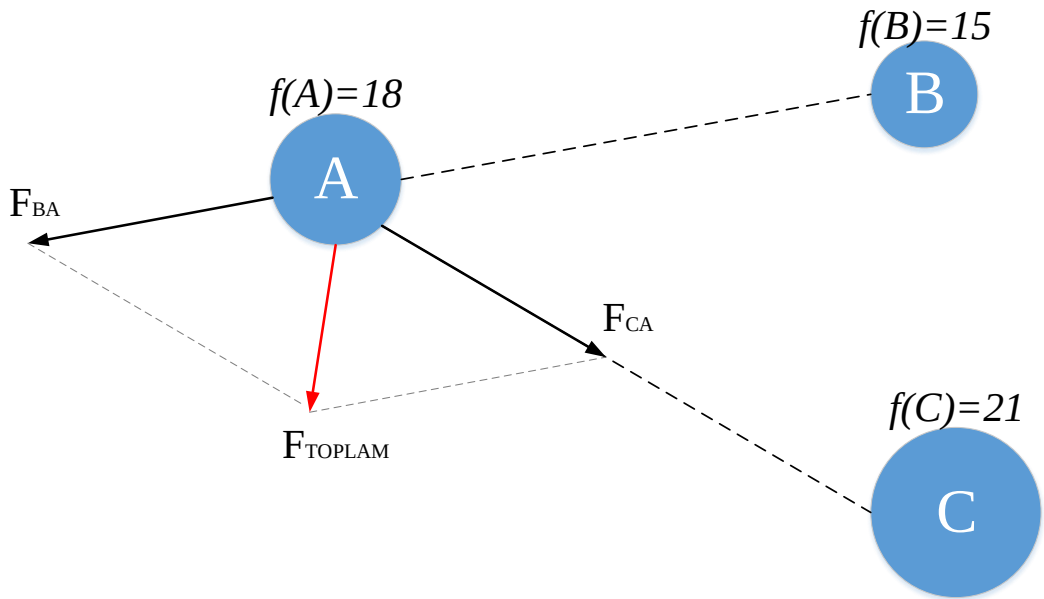
Veri tabanı içinde yer alan kayıtların birbirleriyle ilişkileri incelenerek, hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan veri madenciliği yöntemleri bulunmaktadır. Bu ilişkilerin belirlenmesi sonucunda birliktelik kuralları elde edilir.

Birliktelik kuralları özellikle pazar arařtırmalarında kullanılır. Pazar sepet analizi adı verilen uygulamalarda bu yöntemler kullanılmaktadır. Bu tür çözümlemelerin amacı, müşterinin alışveriş alışkanlıklarını ortaya koymaktır. Bir müşteri herhangi bir ürün aldığında, sepetine başka hangi ürünü koyduğu belirli bir olasılığa göre hesaplanır. Birlikte satılan ürünler belirlendiğinde, mağazalarda raflar ona göre düzenlenir ve müşterinin o ürünlere daha kolay ulaşması sağlanmış olur [66].

4. HİBRİT ELEKTROMANYETİZMA BENZERİ – PARÇACIK SÜRÜ OPTİMİZASYONU – DİFERANSİYEL GELİŞİM ALGORİTMASI

4.1. Elektromanyetizma Benzeri Algoritma

Elektromanyetizma (EM) benzeri algoritma Birbil ve Fang [69] tarafından, Coulomb kanunundaki yüklü parçacıklar arasındaki itme-çekme mekanizmasından esinlenilerek geliştirilmiştir. Bu algoritmaya göre, popülasyondaki her bir eleman yüklü birer parçacık olarak temsil edilir. Her bir parçacığın yükü, o parçacığın uygunluk değeri ile ilişkilendirilmiştir. Parçacıklar yüklerine göre birbirlerini bir kuvvet ile iterler veya çekerler. Bir parçacığa diğer parçacıklar tarafından etki eden toplam kuvvet vektörel olarak hesaplanır ve parçacık bu kuvvet doğrultusunda hareket eder. Bu hareket ile optimum çözüme ulaşmaya çalışılır. Bir parçacığın üzerine diğer parçacıklar tarafından etki eden toplam kuvvet doğrultusunda hareket etmesi Şekil 4.1’ de gösterilmiştir. Şekilde A parçacığı kendinden daha küçük uygunluk değerine sahip B parçacığı tarafından itilirken, kendinden daha büyük uygunluk değerine sahip C parçacığı tarafından çekilir.



Şekil 4.1. Parçacıkların hareketi

EM algoritması, başlangıç popülasyonu oluşturma, yerel arama, toplam kuvvet hesaplama ve parçacıkların hareketi olmak üzere dört temel adımdan oluşur. Algoritmanın temel adımları Tablo 4.1’ de verilmiştir [18, 69]:

Tablo 4.1. EM’ nin genel yapısı

```

1: başlangıçpopulasyonu()
2: iterasyon  $\leftarrow 1$ 
3: while iterasyon < MAXITER do
4:   yerelarama(LSITER,  $\delta$ )
5:   toplamkuvvet()
6:   hareket()
7:   iterasyon  $\leftarrow$  iterasyon + 1
8: end while

```

Burada *MAXITER* maksimum iterasyon sayısını, *LSITER* yerel aramadaki maksimum iterasyon sayısını ve δ yerel arama parametresini temsil etmektedir.

4.1.1. Başlangıç Popülasyonunu Oluşturma

n boyutlu arama uzayı sınırları dâhilinde, m adet yüklü parçacığa rastgele değerler verilerek başlangıç popülasyonu oluşturulur. Her parçacığın uygunluk değeri $f(x)$ uygunluk fonksiyonu ile hesaplanır. Daha sonra en iyi uygunluk değerine sahip olan parçacık, x^{best} olarak saklanır. Tablo 4.2’de bu evrenin sözde kodu verilmiştir [18, 69].

Tablo 4.2. Başlangıç popülasyonu oluşturma

```

1: for  $i = 1$  to  $m$  do
2:   for  $k = 1$  to  $n$  do
3:      $\lambda \leftarrow \text{random}(0, 1)$ 
4:      $x_k^i \leftarrow l_k + \lambda(u_k - l_k)$ 
5:   end for
6:   hesapla  $f(x^i)$ 
7: end for
8:  $x^{best} \leftarrow \min\{f(x^i), \forall i\}$ 

```

Kodda λ , 0 ile 1 arasında gelişigüzel bir sayıyı, u_k ve l_k ise arama uzayının üst ve alt sınırını temsil etmektedir.

4.1.2. Yerel Arama

Yerel arama, parçacıkların çevresinde yerel bilgi toplamak amacıyla kullanılır. Bu evrenin sözde kodu Tablo 4.3’ te verilmiştir.

Tablo 4.3. Yerel arama

```
1: sayaç←1
2: Length←  $\delta(\max_k \{u_k-l_k\})$ 
3: for i = 1 to m do
4:   for k = 1 to n do
5:      $\lambda_1 \leftarrow \text{random}(0,1)$ 
6:     while sayaç < LSITER do
7:        $y \leftarrow x^i$ 
8:        $\lambda_2 \leftarrow \text{random}(0,1)$ 
9:       if  $\lambda_1 > 0.5$  then
10:         $y_k \leftarrow y_k + \lambda_2(\text{Length})$ 
11:       else
12:         $y_k \leftarrow y_k - \lambda_2(\text{Length})$ 
13:       end if
14:       if  $f(y) < f(x^i)$  then
15:         $x^i \leftarrow y$ 
16:        sayaç← LSITER – 1
17:       end if
18:       sayaç ← sayaç + 1
19:     end while
20:   end for
21: end for
22:  $x^{best} \leftarrow \min\{f(x^i), \forall i\}$ 
```

İlk olarak, maksimum uygun adım uzunluğu (*Length*), δ değeri ile hesaplanır. Daha sonra x^i parçacığı geçici y değişkenine atanır. Daha sonra 0 ile 1 arasında rastgele oluşturulan λ_2 sayısı ile *Length* değeri kullanılarak, y değişkenine ekleme veya çıkarma yapılarak koordinat koordinat arama gerçekleştirilir. Bu arama, belirlenen yerel arama iterasyon sayısı (*LSITER*) kadar devam eder. Bu arama sırasında y noktası x^i ’ den daha iyi bir konuma ulaşırsa x^i ’ ye y ’ nin değeri atanır ve o parçacık için arama sona erer. En son, mevcut en iyi parçacık olan x^{best} güncellenir [18, 69].

4.1.3. Toplam Kuvvet Hesaplama

İlk olarak parçacıkların yükü, denklem (4.1)’ de gösterildiği gibi hesaplanır. $f(x^i)$ i . parçacığın uygunluk değerini ve $f(x^{best})$ ise en iyi parçacığın uygunluk değerini temsil eder.

$$q^i = \exp\left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^m (f(x^k) - f(x^{best}))}\right), \forall i \quad (4.1)$$

Daha sonra i parçacığına, diğer parçacıklar tarafından etki eden toplam F^i kuvveti denklem (4.2)' de gösterildiği gibi hesaplanır. Denklemden $\|x^j - x^i\|^2$ iki parçacık arasındaki Öklid uzaklığıdır.

$$F^i = \sum_{j \neq i}^m \begin{cases} (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2} & \text{eğer } f(x^j) < f(x^i) \\ (x^i - x^j) \frac{q^i q^j}{\|x^j - x^i\|^2} & \text{eğer } f(x^j) \geq f(x^i) \end{cases} \quad (4.2)$$

Toplam kuvvet hesaplama evresinin sözde kodu Tablo 4.4' te verilmiştir [18, 69].

Tablo 4.4. Toplam kuvvet hesaplama

```

1: for i=1 to m do
2:   hesapla  $q^i$ ,  $F^i \leftarrow 0$ 
3: end for
4: for i=1 to m do
5:   for j=1 to m do
6:     if  $f(x^j) < f(x^i)$  then
7:        $F^i \leftarrow F^i + (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2}$  {Çekim}
8:     else
9:        $F^i \leftarrow F^i - (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2}$  {İtim}
10:    end if
11:  end for
12: end for

```

4.1.4. Parçacıkların Hareketi

Algoritmanın hareket evresinde ise i parçacığı denklem (4.3)' te verildiği gibi, F^i kuvveti doğrultusunda arama uzayının alt ve üst sınırlarını aşmayacak şekilde λ sayısı ile gelişigüzel hareket ettirilir.

$$x^i = x^i + \lambda \frac{F^i}{\|F^i\|} (u_k - l_k) \quad (4.3)$$

Hareket evresinin sözde kodu ise Tablo 4.5’ te gösterilmiştir [18, 69].

Tablo 4.5. Parçacıkların hareketi

```

1: for  $i = 1$  to  $m$  do
2:   if  $i \neq x^{best}$  then
3:      $\lambda \leftarrow U(0, 1)$ 
4:      $F_k^i \leftarrow \frac{F_k^i}{\|F_k^i\|}$ 
5:     for  $k = 1$  to  $n$  do
6:       if  $F_k^i > 0$  then
7:          $x_k^i \leftarrow x_k^i + \lambda F_k^i (u_k - x_k^i)$ 
8:       else
9:          $x_k^i \leftarrow x_k^i + \lambda F_k^i (x_k^i - l_k)$ 
10:      end if
11:    end for
12:  end if
13: end for

```

Algoritmanın yerel arama, toplam kuvvet hesaplama ve hareket evreleri, bitim şartı sağlanana kadar tekrar eder. Algoritma sonunda en iyi uygunluk değerine sahip olan parçacık, çözüm olarak seçilir.

4.2. Parçacık Sürü Optimizasyon Algoritması

Kennedy ve Eberhart [70] tarafından önerilen PSO; kuş, balık veya böcek sürülerinin iki boyutlu hareketlerinden esinlenilerek geliştirilmiştir. PSO sürüdeki bireylerin aralarındaki sosyal bilgi paylaşımını esas almaktadır. Algoritma, sürü halindeki hayvanların yiyecek ararken sergilemiş oldukları rastgele davranışlar ile amaçlarına daha kolay ulaşabildikleri fikrinden esinlenilerek geliştirilmiştir. Algoritmada sürüdeki her birey, parçacık olarak adlandırılır. Başlangıç popülasyonu rastgele oluşturulmuş parçacıklar ile başlatılır. Her iterasyonda tüm parçacıklar $pbest$ ve $gbest$ değerlerine göre güncellenmektedir. $pbest$ bir parçacığın o ana kadarki en iyi konumunu, $gbest$ ise tüm popülasyonda o ana kadar bulunmuş en iyi konumu temsil eder. Bu iki değer dikkate alınarak denklem (4.4) ve denklem (4.5)’ teki gibi parçacıkların hızları ve konumları güncellenir.

$$V_{i(t+1)} = w \cdot V_{i(t)} + c_1 \cdot rand1 \cdot (pbest_i - X_i) + c_2 \cdot rand2 \cdot (gbest - X_i) \quad (4.4)$$

$$X_{i(t+1)} = X_{i(t)} + V_{i(t+1)} \quad (4.5)$$

$X_{i(t)}$ ve $X_{i(t+1)}$ i. parçacığın mevcut konumunu ve yeni konumunu, $V_{i(t)}$ ve $V_{i(t+1)}$ ise i. parçacığın mevcut hızını ve yeni hızını temsil etmektedir. $rand1$ ve $rand2$, 0 ile 1 arasında rastgele üretilmiş değerlerdir. w atalet katsayısı, c_1 ve c_2 ise öğrenme katsayılarıdır ve hızlanma katsayıları olarak da adlandırılır. c_1 ve c_2 katsayıları, bir iterasyonda bir parçacığın kendi en iyi ve küresel en iyi pozisyonlarına doğru olan hızlanmanın alabileceği adımın maksimum boyutunu etkiler. Düşük değerlerin seçilmesi parçacıkların hedef bölgeye ulaşmadan önce, uzak yerlerde dolaşmalarına imkân verir. Yüksek değerlerin seçilmesi ise hedefe ulaşmayı hızlandırır fakat beklenmedik hareketlerin oluşmasına ve hedef bölgeye ulaşılmamasına neden olabilir. Algoritma temel olarak aşağıdaki basamaklardan oluşur [70,71]:

ADIM 1: Parçacıkların başlangıç pozisyonlarını arama uzayı sınırları içerisinde rastgele değerler vererek oluştur ayrıca parçacıkların başlangıç hızlarını ayarla.

ADIM 2: Tüm parçacıkların uygunluk değerlerini hesapla.

ADIM 3: Her bir parçacığın mevcut nesil için yerel en iyi ($pbest$) değerlerini bul.

ADIM 4: Mevcut nesildeki yerel en iyiler içerisinde küresel en iyiyi ($gbest$) seç.

ADIM 5: Pozisyonları ve hızları denklem (4.4) ve denklem (4.5)' i kullanarak güncelle.

ADIM 6: Bitim şartı sağlamıyorsa ADIM 2' ye git.

4.3. Diferansiyel Gelişim Algoritması

DE, Price ve Storn [72, 73] tarafından optimizasyon problemlerinin çözümü amacıyla geliştirilen popülasyon tabanlı sezgisel bir algoritmadır. Algoritmada kullanılan operatörler GA temellidir. Popülasyondaki bireyler kromozom olarak adlandırılır. Algoritmanın temel adımları aşağıda verildiği gibidir:

ADIM 1: Kontrol parametrelerini ayarla ve başlangıç popülasyonunu rastgele bir biçimde oluştur.

ADIM 2: Mutasyon

ADIM 3: Çaprazlama

ADIM 4: Değerlendirme

ADIM 5: Seçim

ADIM 6: Bitim şartı sağlanmıyorsa ADIM 2' ye git

4.3.1. Başlangıç Popülasyonunun Oluşturulması ve Kullanılan Parametreler

Algoritma, en az dört adet rastgele değere sahip kromozomdan oluşan başlangıç popülasyonu ile başlar. Çünkü yeni kromozomların üretilmesi için mevcut kromozom dışında üç adet daha kromozom gerekmektedir. Ayrıca çaprazlama oranı (CR), ölçekleme faktörü (F) ve maksimum iterasyon sayısı gibi parametre değerleri de ayarlanmalıdır [72-74].

4.3.2. Mutasyon

Mutasyon, bir kromozomun genleri üzerinde rastgele miktarlarda değişiklik yapmak amacıyla kullanılır. Bu değişiklikler sayesinde kromozom, çözüm uzayında hareket ettirilerek değişik noktalarda arama gerçekleştirilir. Mutasyon işlemi için mevcut kromozom r dışında r_1 , r_2 ve r_3 olmak üzere rastgele üç farklı kromozom seçilir. Daha sonra denklem (4.6)'da görüldüğü gibi; seçilen kromozomlardan ilk ikisinin farkı alınır ve F parametresi ile çarpılarak fark kromozomu elde edilir. Ardından fark kromozomu ile seçilen üçüncü kromozom toplanır. Böylece çaprazlamada kullanılacak olan r_ϵ kromozomu elde edilmiş olur [72, 73].

$$r_\epsilon^i = (r_1^i - r_2^i) \cdot F + r_3^i \quad (4.6)$$

4.3.3. Çaprazlama

Çaprazlamada, mutasyon sonucu oluşturulan r_ϵ kromozomu ile mevcut kromozom r kullanılarak yeni bir aday kromozom r_a üretilir. Yeni aday kromozomdaki her bir gen CR olasılık ile r_ϵ kromozomundan, $1-CR$ olasılık ile de mevcut kromozomdan seçilir. En az bir genin r_ϵ kromozomundan gelmesini sağlamak için, CR değerine bakılmaksızın rastgele seçilen bir gen r_ϵ den alınır. Denklem (4.7) de çaprazlama prosedürü görülmektedir [72, 73].

$$r_a^i = \begin{cases} r_\epsilon^i & \text{rand}[0,1] < CR \\ r^i & \text{Aksi durumda} \end{cases} \quad (4.7)$$

4.3.4. Değerlendirme ve Seçim

Üzerinde işlem yapılan problemin, amaç fonksiyonuna göre mevcut kromozomun uygunluk değeri ile mutasyon ve çaprazlama sonucu oluşturulan aday kromozomun uygunluk değeri hesaplanır. Daha sonra mevcut kromozomun uygunluk değeri ile aday kromozomun uygunluk değeri karşılaştırılır ve yeni nesil için uygunluğu yüksek olan kromozom seçilir.

4.4. Hibrit EM-PSO-DE Algoritması

EM algoritmasında bir parçacığın hareketi, kendi en iyi konumunu ve küresel en iyi konumu bilmeden, tamamen o parçacık üzerinde diğer parçacıklar tarafında etki eden toplam kuvvet oranında ve yönünde olmaktadır. Önerilen hibrit algoritmada, EM algoritmasının hareket evresi, PSO algoritmasının hız ve konum güncelleme denklemleri kullanılarak güncellenmiştir. PSO' nun hız denklemindeki hızlanma katsayıları olan c_1 , c_2 ile $rand_1$ ve $rand_2$ değerleri yerine EM algoritmasındaki q ve F değerleri kullanılmıştır. Hibrit EM-PSO-DE' nin hız ve konum güncelleme denklemleri, denklem (4.8) ve denklem (4.9)' verilmiştir:

$$V_k^i = w.V_k^i + q^i.(pbest_k^i - X_k^i) + |F_k^i|. (gbest_k - X_k^i) \quad (4.8)$$

$$X_k^i = X_k^i + V_k^i \quad (4.9)$$

Bu sayede, bir parçacık kendi yük değeri olan q^i büyüklüğünde bir adımla kendi kişisel en iyisine ve üzerine etki eden toplam kuvvetin mutlak değeri olan F_k^i büyüklüğünde küresel en iyiye doğru adım atmış olacaktır.

EM' nin yerel arama evresinde koordinat koordinat arama gerçekleştirilmektedir. Bu aramada, önce işlem yapılan parçacığın ilk boyutunda değişiklik yapılır. Daha sonra bu parçacığın uygunluk değeri hesaplanır ve uygunluk değerinde artış yok ise ikinci boyuta geçilir. İkinci boyutta da değişiklik yapılır ve tekrardan uygunluk değeri hesaplanır ve artış kontrol edilir. Uygunluk değerinde bir artış yoksa bu işlemler o parçacık için *LSITER* kadar devam eder. Bu durum, problemin boyut sayısı veya popülasyon sayısı arttıkça algoritmanın çalışma zamanını aşırı bir biçimde arttırmaktadır. Önerilen yöntemde ise bu durumun önüne

geçmek için koordinat koordinat yerel arama yerine, *LSITER* iterasyon sayısı kadar DE algoritması kullanılmıştır. Hibrit EM-PSO-DE algoritmasının genel yapısının sözde kodu Tablo 4.6’ daki gibidir:

Tablo 4.6. Hibrit EM-PSO-DE’ nin temel yapısı

```

1: başlangıçpopulasyonu()
2: iterasyon ← 1
3: while iterasyon < MAXITER do
4:   DE ile yerelarama(LSITER)
5:   pbest, gbest ve w güncelle
6:   toplamkuvvet()
7:   hareket()
8:   iterasyon ← iterasyon + 1
9: end while

```

Tablo 4.7. EM-PSO-DE hareket evresi

```

1: for i = 1 to m do
2:   if  $i \neq x^{best}$  then
3:      $\lambda \leftarrow U(0, 1)$ 
4:      $F_k^i \leftarrow \frac{F_k^i}{||F_k^i||}$ 
5:     for k = 1 to n do
6:        $V_k^i = w \cdot V_k^i + q^i \cdot (pbest_k^i - X_k^i) + |F_k^i| \cdot (gbest_k - X_k^i)$ 
7:       if  $V_k^i > Vmax$ 
8:          $V_k^i = Vmax$ 
9:       else if  $V_k^i < Vmin$ 
10:         $V_k^i = Vmin$ 
11:       end if
12:        $X_k^i = X_k^i + V_k^i$ 
13:     end for
14:   end if
15: end for

```

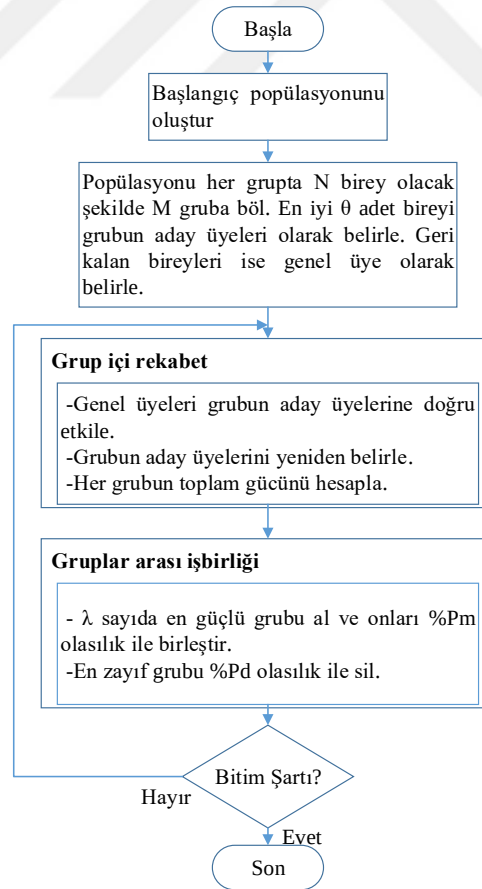
Önerilen yöntemde toplam kuvvet hesaplama EM algoritması ile aynıdır. EM' nin başlangıç popülasyonunu oluşturma evresine ek olarak EM-PSO-DE' de başlangıç hızları, maksimum hız (V_{max}), minimum hız (V_{min}) ve atalet katsayısının (w) başlangıç değerleri de oluşturulur. Durdurma kriteri sağlanıncaya kadar EM-PSO-DE' nin 4, 5, 6 ve 7 numaralı adımları tekrar edilir. Algoritma sonunda en iyi uygunluk değerine sahip olan parçacık çözüm olarak seçilir. EM-PSO-DE algoritmasının hareket evresinin sözde kodu ise Tablo 4.7' deki gibidir.



5. HİBRİT PARLAMENTER OPTİMİZASYON – BÜYÜK PATLAMA BÜYÜK ÇÖKÜŞ ALGORİTMASI

5.1. Parlamenter Optimizasyon Algoritması

İlk olarak Borji [75] tarafından önerilen POA, parlamentodaki partilerin rekabetçi ve iş birliğine dayalı davranışlarından esinlenilerek geliştirilmiş, sosyal tabanlı bir algoritmadır. POA, genetik algoritmalar ve parçacık sürü algoritmaları gibi olasılıksal, iteratif ve popülasyon tabanlı küresel optimizasyon tekniğidir. İnsan sosyal hayatının birçok durumunda rekabetçi davranışlar gözlemlenebilmektedir. Bilhassa bu metot, parlamentonun kontrolünü ele geçirme çalışmaları sırasındaki grup içi ve gruplar arası çekişmeleri ve işbirliklerini simüle etmeye çalışmıştır. Algoritmanın temel adımları Şekil 5.1’ deki akış şemasında verilmiştir [75,76].



Şekil 5.1. POA akış şeması

5.1.1. Başlangıç Popülasyonunu Oluşturma ve Popülasyonun Bölümlendirilmesi

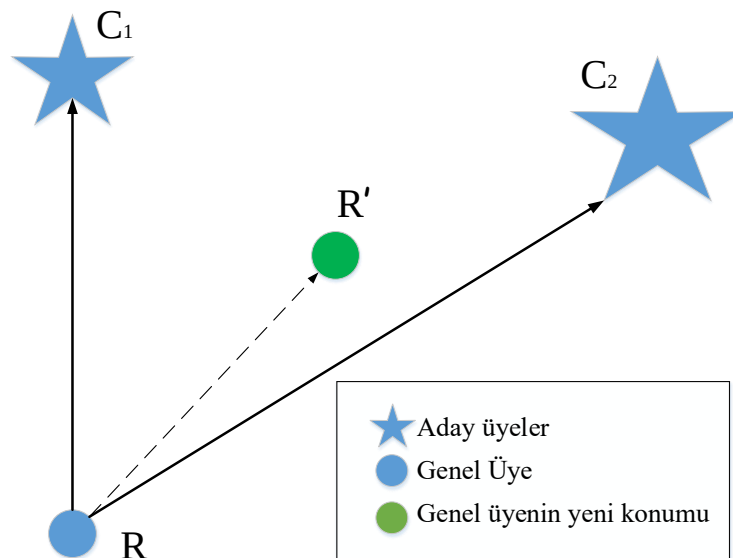
POA ilk olarak, rastgele bir biçimde başlangıç popülasyonunun oluşturulması ile başlar. Daha sonra başlangıç popülasyonundaki bireyler M adet gruba, her grupta N adet birey olacak şekilde rastgele dağıtılır. Her grupta en iyi uygunluk değerine sahip birkaç birey, o grubun aday üyeleri olarak seçilir. Geri kalan bireyler ise o grubun genel üyeleri olarak adlandırılır [2, 75-77].

5.1.2. Grup İçi Rekabet

Bir grubun aday ve genel üyeleri belirlendikten sonra genel üyeler, aday üyelere doğru yönelirler. Bu yönelim aday üyelerin uygunluk değerlerine bağlı olarak gerçekleşmektedir. Genel üyenin yeni pozisyonu denklem (5.1)' deki gibi hesaplanır [2, 75-77].

$$p' = p_0 + \lambda \left(\frac{\sum_{i=1}^{\theta} (p_i - p_0) \cdot f(p_i)}{\sum_{i=1}^{\theta} f(p_i)} \right) \quad (5.1)$$

Denklemden λ , 0.5 ile 2 arasında rastgele bir değerdir. p' genel üyenin yeni konumu, p_0 genel üyenin mevcut konumunu ve p_i ise i . aday üyenin mevcut konumunu temsil etmektedir. f ise uygunluk fonksiyonudur. Şekil 5.2' de genel üyelerin aday üyelere doğru yönelme mekanizması görülmektedir.



Şekil 5.2. POA' nın yönelme mekanizması

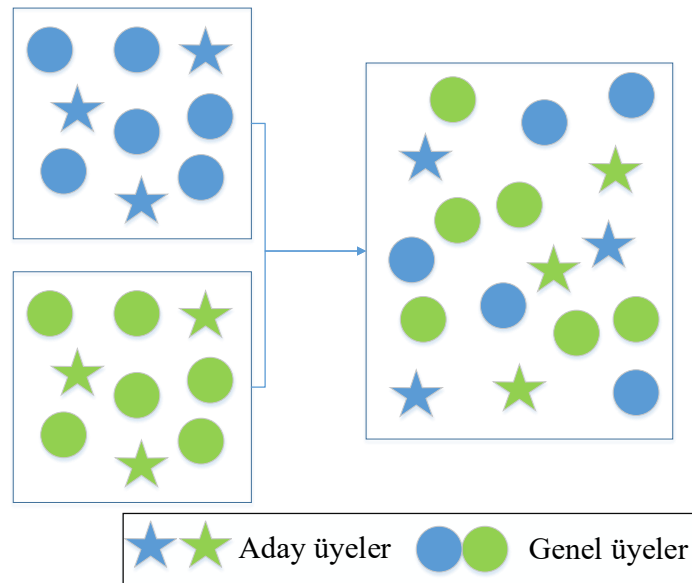
Yönelme sonrasında genel üyenin yeni konumu mevcut konumundan daha kötü ise genel üyenin mevcut durumu korunur. Ayrıca yönelme sonrasında genel üyeler, aday üyelerden daha iyi bir konuma ulaşabilir. Bu nedenle grubun aday üyelerini yeniden belirlemek gerekir. Daha sonra her grubun gücü, gruptaki aday ve genel üyelerin uygunluk değerleri ile orantılı olacak bir biçimde denklem (5.2)' deki gibi hesaplanır [2, 75-77].

$$power^i = \frac{m \cdot ort(Q^i) + n \cdot ort(R^i)}{m+n} ; m > n. \quad (5.2)$$

Denklemden $power^i$, i . grubun gücünü temsil etmektedir. $ort(Q^i)$ aday üyelerin, $ort(R^i)$ ise genel üyelerin uygunluk değerlerinin ortalamasıdır. m ve n ise kullanıcı tanımlı ağırlık sabitleridir.

5.1.3. Gruplar Arası İşbirliği

Güçlü gruplar bazen kendi güçlerini arttırmak için bir gruba katılır ve birleşirler. Birleşmeyi gerçekleştirmek için rastgele bir sayı üretilir ve bu sayı p_m ' den küçük ise, λ sayıda en güçlü grup seçilir ve bir grupta birleştirilir. Şekil 5.3, bu birleşmeyi göstermektedir. Devam eden algoritma boyunca; zayıf gruplar, hesaplama gücünü korumak için silinir. Silme işlemi için rastgele bir sayı üretilir ve üretilen sayı p_d 'den küçükse, y sayıda minimum güce sahip grup silinir [2, 75-77].



Şekil 5.3. Grupların birleşmesi

5.1.4. Algoritmanın Sonlandırılması

Maksimum sayıda iterasyona ulaşıldığında algoritma sonlandırılır. Algoritma sonunda, en iyi uygunluk değerine sahip olan birey, çözüm olarak kabul edilir.

5.2. Büyük Patlama – Büyük Çöküş Algoritması

BB-BC algoritması Erol ve Eksin [78] tarafından, evren büyük patlama (big bang) ile oluşmuştur ve büyük çöküş (big crunch) ile son bulacaktır teorisinden esinlenilerek geliştirilmiştir. BB-BC algoritması büyük-patlama ve büyük-çöküş olmak üzere temel iki evreden oluşur.

İlk olarak büyük patlama ile arama uzayı sınırları içerisinde rastgele değerlere sahip parçacıklar ile başlangıç popülasyonu oluşturulur. Daha sonra büyük çöküş evresine geçilir. Bu evrede, her parçacığın konumları ve uygunluk değerleri dikkate alınarak kütle merkezi hesaplanır. Kütle merkezini temsil eden x^c noktası denklem (5.3) ile hesaplanır.

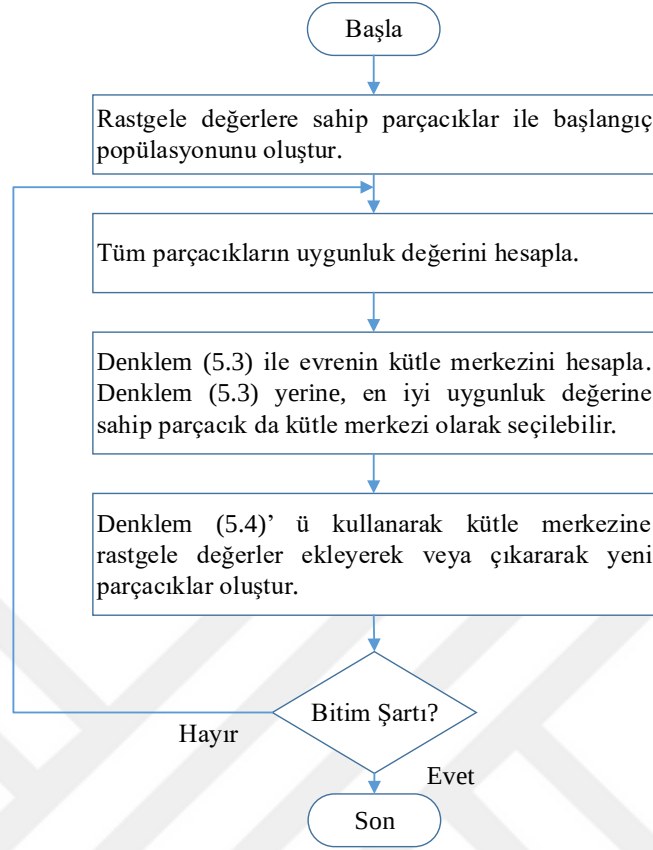
$$x^c = \frac{\sum_{i=1}^N \frac{x_i^i}{f_i}}{\sum_{i=1}^N \frac{1}{f_i}} \quad (5.3)$$

Denklemde x_i^i , i . parçacığın konumunu f_i , i . parçacığın uygunluk değerini ve N ise popülasyon sayısını temsil etmektedir. Yukarıdaki denklemin dışında alternatif olarak, en iyi uygunluk değerine sahip parçacık da kütle merkezi x^c olarak seçilebilir.

Büyük çöküş evresinden sonra, kütle merkezinin çevresinde yeni popülasyonun oluşturulduğu büyük patlama evresine geçilir. Bu evrede, denklem (5.4)'te gösterildiği gibi; kütle merkezine, her iterasyonda azalacak şekilde rastgele değerler eklenerek veya çıkarılarak yeni parçacıklar oluşturulur.

$$x^{new} = x^c + \frac{l.r}{k} \quad (5.4)$$

Denklemde x^c kütle merkezini, l arama uzayının üst limitini, r ise 0 ile 1 arasında rastgele bir sayıyı, k ise o anki iterasyon adımını ve x^{new} yeni oluşan parçacığın konumunu temsil etmektedir. Algoritmanın temel adımları Şekil 5.4' teki akış şemasında verilmiştir [78].



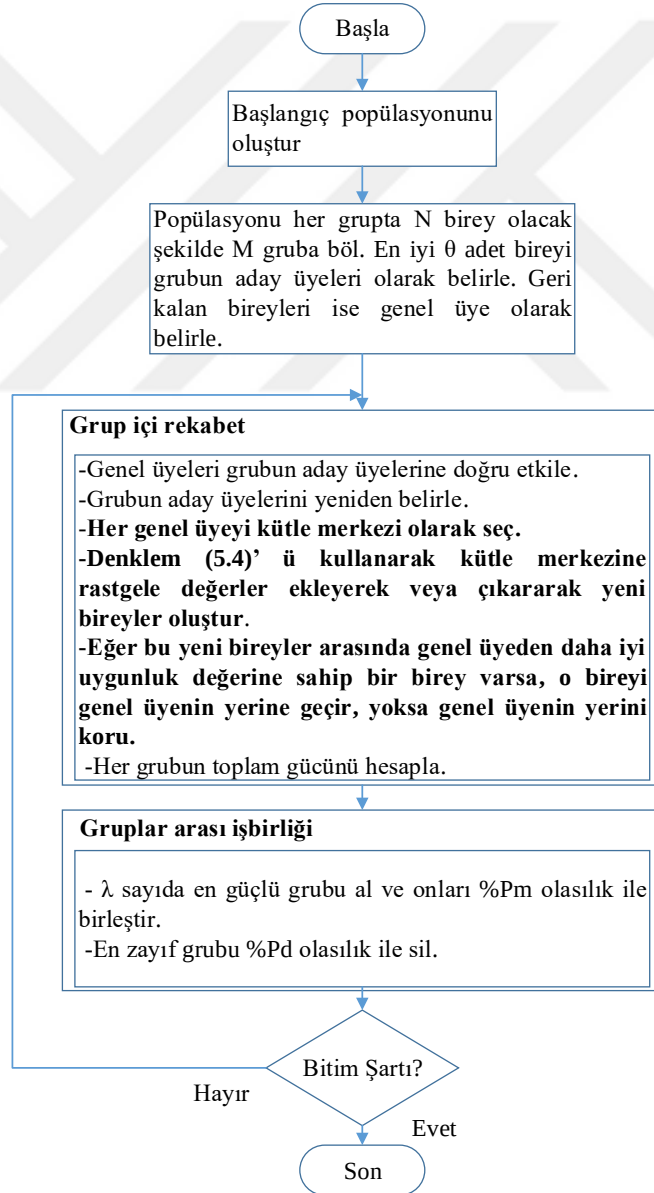
Şekil 5.4. BB-BC akış şeması

5.3. HPO-BBBC algoritması

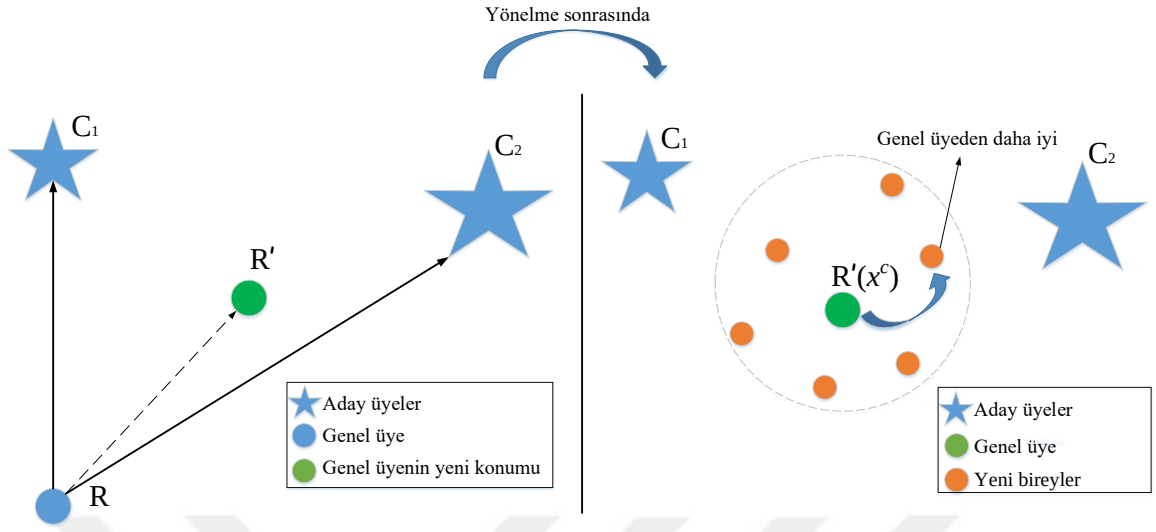
POA grup içi rekabet ve gruplar arası işbirliği olmak üzere iki temel evreden oluşmaktadır. Grup içi rekabet evresinde genel üyeler, aday üyelere doğru yönelirler. Yönelme sonrasında genel üyeler, aday üyelere daha iyi bir uygunluk değerine sahip olabilir. Bu durumda grubun aday üyeleri tekrar güncellenir. Bu işlemler algoritmaya, çözüm uzayında arama yapmasına izin verir ve bir keşif mekanizması sağlar [75, 76]. Diğer yandan, BB-BC algoritmasında sömürü mekanizmasının başarılı olduğu bilinmektedir [33, 78].

Önerilen HPO-BBBC algoritmasında, grup içi rekabet evresi sonunda genel üyeler, grubun aday üyesi olmadıkları için kendilerini başarısız kabul ederler. Bu yüzden kendileri yerine çevrelerinde, bir sonraki grup içi yarışma için kendilerinden daha iyi ve aday üye olma ihtimali daha yüksek olabilecek bireyler ararlar. Genel üye, çevresinde daha iyi bir birey bulursa gruptan ayrılır ve yerine daha iyi olan birey geçer, bulamaz ise grupta kalmaya devam eder.

Önerilen yöntemde, POA' nın genel üyelerinin aday üyelere doğru yönelmesi ve aday üyelerin yeniden belirlenmesi adımından sonra, her genel üye kütle merkezi x^c olarak seçilir. Daha sonra x^c çevresinde daha iyi bireyler aramak için büyük patlama işlemi uygulanır. Denklem (5.4) kullanılarak x^c etrafında p adet yeni birey oluşturulur. Bu yeni bireyler arasında genel üyeden daha iyi uygunluk değerine sahip bir birey varsa, o birey genel üyenin yerine geçer, yoksa genel üye yerini korur. Bu sayede genel üyeler etrafında yerel arama gerçekleştirilmiş olur. HPO-BBBC' nin temel adımları ise Şekil 5.5' teki akış şemasında verilmiştir. Genel üyeler çevresinde daha iyi birey arama mekanizması Şekil 5.6' da temsili olarak gösterilmektedir.



Şekil 5.5. HPO-BBBC akış şeması



Şekil 5.6. Genel üyeler etrafında daha iyi birey arama mekanizması

6. DENEYSEL ÇALIŞMADA KULLANILAN TEST FONKSİYONLARI VE VERİ SETLERİ

Bu bölümde deneysel çalışmalarda kullanılan kalite testi fonksiyonları ve nümerik sınıflandırma veri setleri hakkında ayrıntılı bilgi verilmiştir.

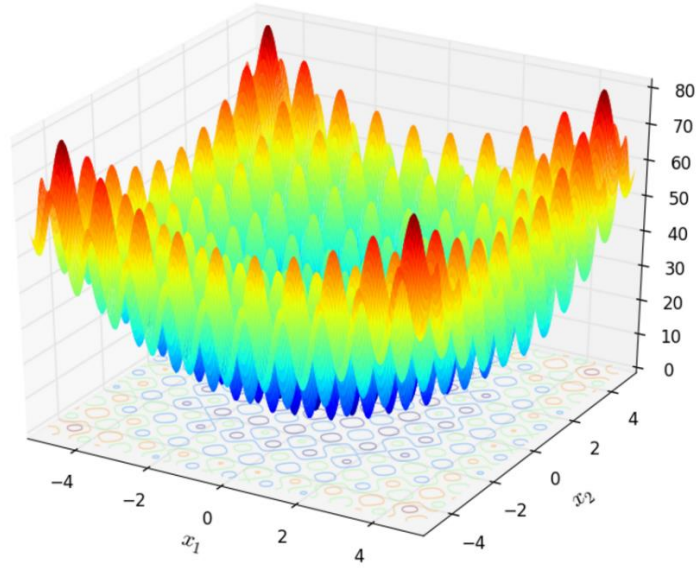
6.1. Kalite Testi Fonksiyonları

Optimizasyon algoritmalarının performansını ölçmek amacıyla kalite testi fonksiyonları sıklıkla kullanılmaktadır. Kalite testi fonksiyonlarının çoğu, mühendislik problemlerinin karmaşıklığına sahiptir. Ayrıca fonksiyonların zorluk dereceleri parametrelerin değiştirilmesi ile ayarlanabilir. Bu çalışmada 9 farklı kalite test fonksiyonu seçilmiştir. Fonksiyonlardan Sphere ve Rosenbrock tek modlu (sadece tek bir küresel optimuma sahip olan), diğer fonksiyonlar ise çok modludur (birden çok yerel optimuma ve tek bir küresel optimuma sahip olan) [79-81].

6.1.1. Rastrigin Fonksiyonu

Denklem (6.1)' de görülen Rastrigin fonksiyonu çok modludur. Sınırları $-5.12 \leq x \leq 5.12$ aralığında seçilmiştir. Küresel minimum noktası $x^* = (0,0,...,0)$ ve $f(x^*) = 0$. Fonksiyonun grafiği Şekil 6.1' de verilmiştir.

$$10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \quad (6.1)$$

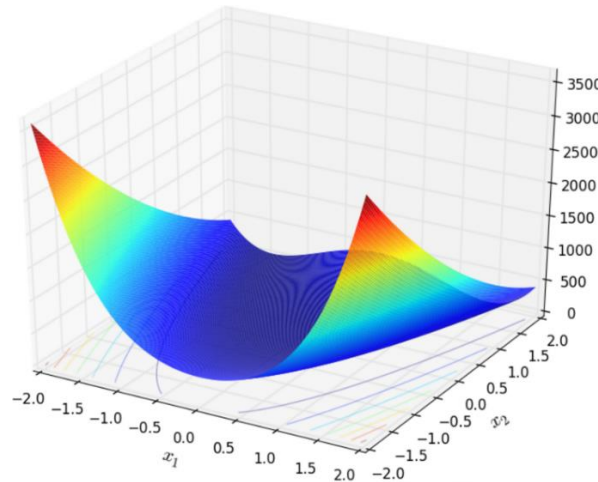


Şekil 6.1. Rastrigin fonksiyonu

6.1.2. Rosenbrock Fonksiyonu

Küresel minimumu dar bir vadide yer alan Rosenbrock fonksiyonu tek modlu bir fonksiyondur. Vadiye ulaşmak kolaydır fakat küresel minimuma yakınsamak zordur. Sınırları $-2.048 \leq x \leq 2.048$ aralığında seçilmiştir. Küresel optimum noktası $x^* = (1,1,...,1)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.2)' de, grafiği ise Şekil 6.2' de verilmiştir.

$$\sum_{i=1}^{d-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i^2)] \quad (6.2)$$

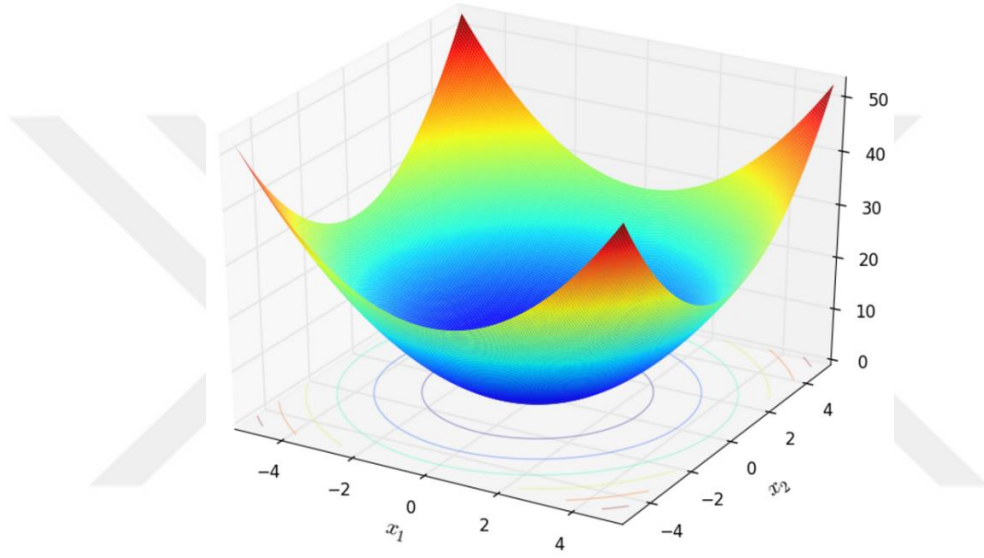


Şekil 6.2. Rosenbrock fonksiyonu

6.1.3. Sphere Fonksiyonu

Tek modlu bir fonsiyondur. Küresel minimumu $x^* = (0,0,...,0)$ ve $f(x^*) = 0$. Sınırlar $-5.12 \leq x \leq 5.12$ aralığında seçilmiştir. Fonksiyonun matematiksel ifadesi Denklem (6.3)' te, grafiği ise Şekil 6.3' te verilmiştir.

$$\sum_{i=1}^d x_i^2 \quad (6.3)$$

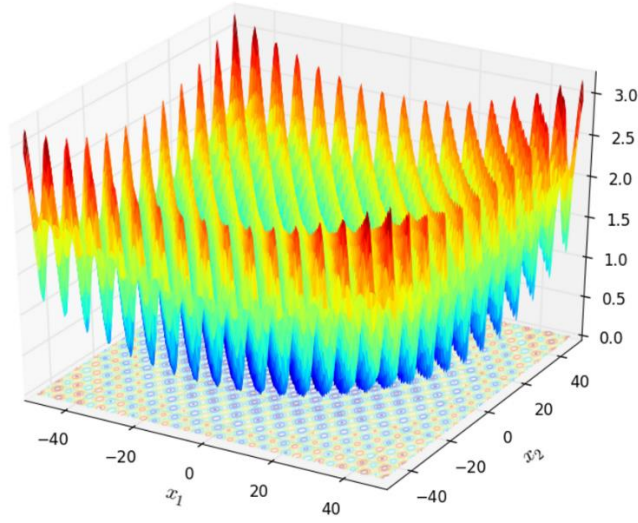


Şekil 6.3. Sphere fonksiyonu

6.1.4. Griewank Fonksiyonu

Düzenli dağıtılmış birçok yerel optimum noktasına sahip olan Griewank fonksiyonu çok modludur. Sınırları $-10 \leq x \leq 10$ aralığında seçilmiştir. Küresel optimum noktası $x^* = (0,0,...,0)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.4)' te, grafiği ise Şekil 6.4' te verilmiştir.

$$\sum_{i=1}^d \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (6.4)$$

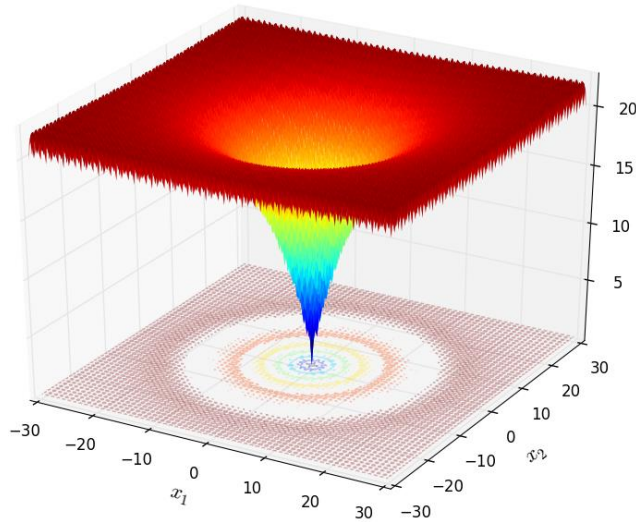


Şekil 6.4. Griewank fonksiyonu

6.1.5. Ackley Fonksiyonu

Birçok yerel derin optimum noktaya sahip olan Ackley fonksiyonu çok-modlu bir fonksiyondur. Sınırları $-32 \leq x \leq 32$ aralığında seçilmiştir. Küresel optimum noktası $x^* = (0,0,...,0)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.5)' te, grafiği ise Şekil 6.5' te verilmiştir.

$$20 + e - 20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) \quad (6.5)$$

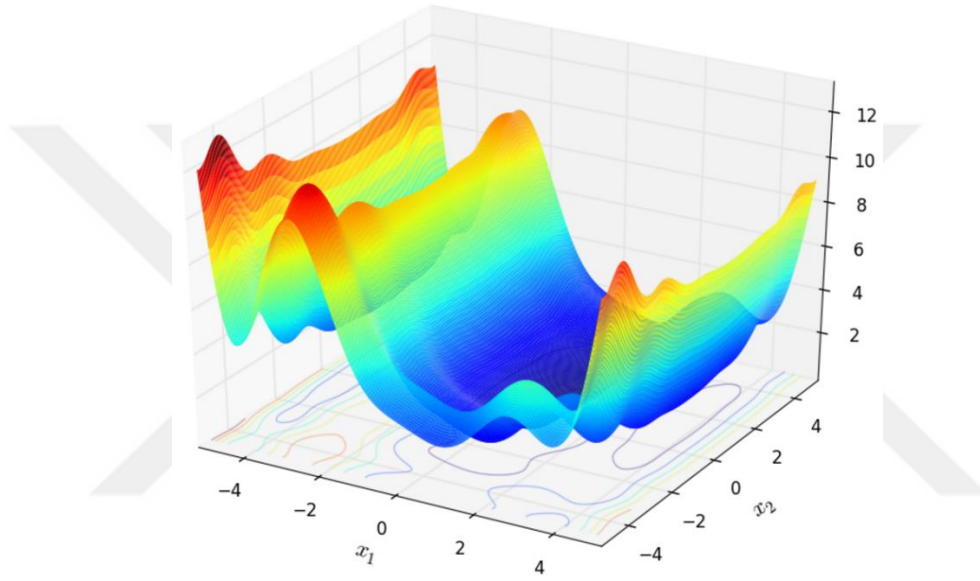


Şekil 6.5. Ackley fonksiyonu

6.1.6. Levy Fonksiyonu

Çok modlu bir fonksiyondur. Sınırları $-10 \leq x \leq 10$ aralığında seçilmiştir. Küresel minimum noktası $x^* = (1,1,...,1)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.6)' da, grafiği ise Şekil 6.6' da verilmiştir.

$$\sin^2(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2, y_i = 1 + \frac{x_i - 1}{4} \quad (6.6)$$

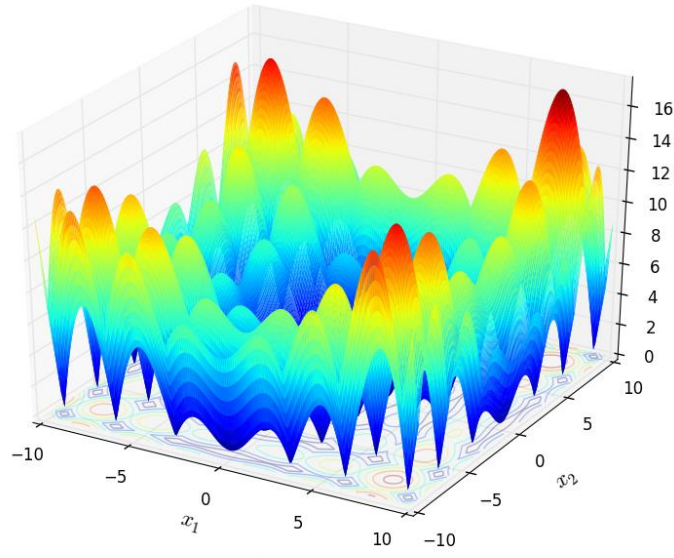


Şekil 6.6. Levy fonksiyonu

6.1.7. Alpine Fonksiyonu

Birçok derin yerel optimum noktaya sahip olan Alpine fonksiyonu çok-modlu bir fonksiyondur. Sınırları $-10 \leq x \leq 10$ aralığında seçilmiştir. Küresel optimum noktası $x^* = (0,0,...,0)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.7)' de, grafiği ise Şekil 6.7' de verilmiştir.

$$\sum_{i=1}^d |x_i \sin(x_i) + 0.1 x_i| \quad (6.7)$$

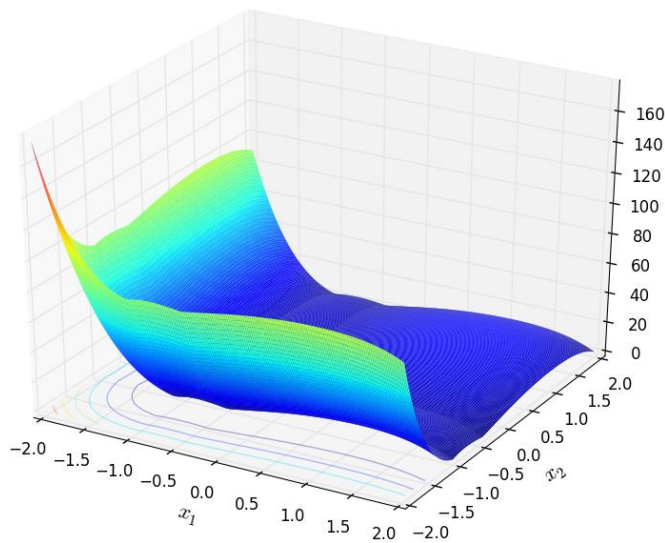


Şekil 6.7. Alpine fonksiyonu

6.1.8. Quintic Fonksiyonu

Çok modlu bir fonksiyondur. Sınırları $-10 \leq x \leq 10$ aralığında seçilmiştir. Küresel minimum noktası $x^* = (-1 \text{ veya } 2)$ ve $f(x^*) = 0$. Fonksiyonun matematiksel ifadesi Denklem (6.8)' de, grafiği ise Şekil 6.8' de verilmiştir.

$$\sum_{i=1}^d |x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4| \quad (6.8)$$

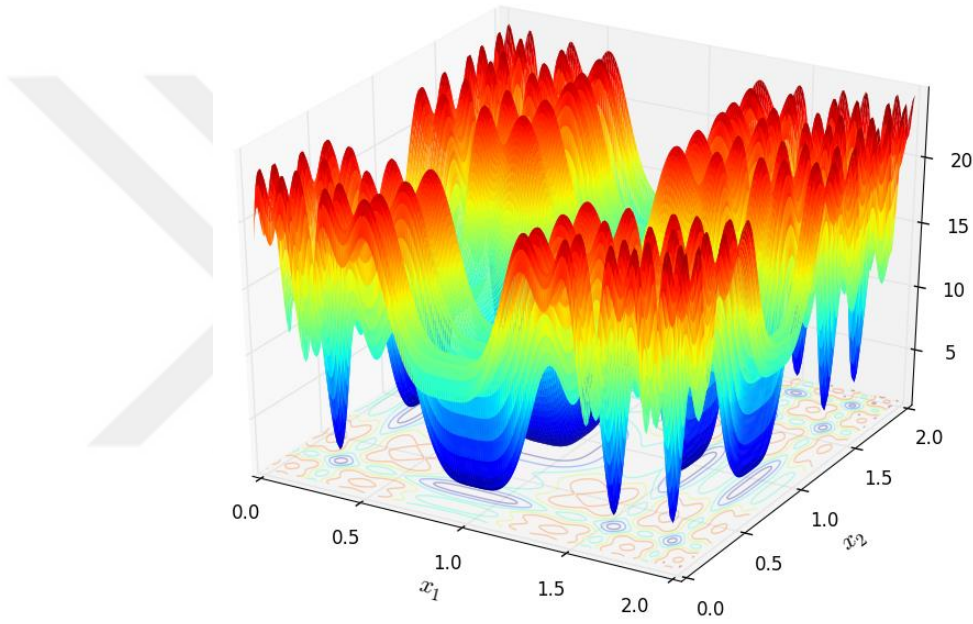


Şekil 6.8. Quintic fonksiyonu

6.1.9. Trigonometric Fonksiyonu

Birçok derin ve yerel optimum noktaya sahip olan Trigonometric fonksiyonu, çok modlu ve küresel minimum noktaya ulaşması zor bir fonksiyondur. Sınırları $-500 \leq x \leq 500$ aralığında seçilmiştir. Küresel optimum noktası $x^* = (0.9, 0.9, \dots, 0.9)$ ve $f(x^*) = 1$. Fonksiyonun matematiksel ifadesi Denklem (6.9)' da grafiği ise Şekil 6.9' da verilmiştir.

$$1 + \sum_{i=1}^d 8 \sin^2[7(x_i - 0.9)^2] + 6 \sin^2[14(x_i - 0.9)^2] + (x_i - 0.9)^2 \quad (6.9)$$



Şekil 6.9. Trigonometric fonksiyonu

6.2. Nümerik Sınıflandırma Veri Setleri

Bu tez çalışmasında, KEEL [82] ve UCI [83] veri ambarlarından alınan Banknote, Phoneme ve Banana isimli üç farklı veri seti kullanılarak önerilen hibrit algoritmaların nümerik sınıflandırma kural keşfindeki performansları test edilmiştir. Tüm veri setleri, iki sınıftan oluşmaktadır. Banknote veri seti dört, Phoneme veri seti beş ve Banana veri seti ise iki özellikten oluşur. Özelliklerin tamamı reel değerler almaktadır. Deneysel çalışmada kullanılan veri setlerinin özellikleri Tablo 6.1' de verilmiştir.

Tablo 6.1. Deneyisel çalışmada kullanılan veri setleri ve özellikleri

VERİ SETİ	ÖZELLİK SAYISI	SINIF ETİKETİ	EĞİTİM	TEST	TOPLAM
Banknote	4	0	534	228	1372
		1	427	183	
Phoneme	5	0	2673	1145	5404
		1	1111	475	
Banana	2	1	4276	238	10069
		-1	5262	293	

6.2.1. Kural Temsili

Deneyisel çalışmada kullanılan tüm algoritmaların başlangıç popülasyonu, arama uzayı sınırları içinde olacak şekilde rastgele değerler verilerek oluşturulmuştur. Popülasyondaki her birey aslında birer aday kuraldır. Oluşturulan aday kurallar, üzerinde sınıflandırma kural keşfi işlemi yapılacak veri setinin özellik sayısı kadar boyuttan oluşur. Her boyut ise 3 alt kısımdan oluşur. Şekil 6.10’ da bir aday kuralın nasıl temsil edildiği görülmektedir [2].

Özellik ₁			Özellik ₂			...	Özellik _n		
B ₁	A ₁	Ü ₁	B ₂	A ₂	Ü ₂	...	B _n	A _n	Ü _n

Şekil 6.10. Bir kuralın temel yapısı

Şekilde *B* bayraktır. Bayrak 1 ise o özellik kuralda bulunur, 0 ise kuralda bulunmaz. *A* alt sınır, *Ü* ise üst sınırdır. Kural temsili için, Şekil 6.11’ de 3 özellikten oluşan bir veri setinde *Y* sınıfına ait örnek bir kural temsili gösterilmiştir.

Özellik ₁			Özellik ₂			Özellik ₃		
1	5	10	0	4	16	1	2	7

Şekil 6.11. Örnek kural

Şekilde *Özellik₁* ve *Özellik₃*’ ün bayrakları 1 olduğu için kuralda bulunacak, *Özellik₂*’ nin bayrağı 0 olduğu için kuralda bulunmayacaktır. Örnek kural aşağıdaki gibi ifade edilir:

Eğer $5 \leq \text{Özellik}_1 \leq 10$ ve $2 \leq \text{Özellik}_3 \leq 7$ ise Sınıf = *Y*’ dir.

6.2.2. Uygunluk Fonksiyonu

Bu çalışmada aday kuralların uygunluk değerleri denklem (6.10)'daki uygunluk fonksiyonu kullanılarak hesaplanmıştır [84].

$$F = c_1 \cdot \frac{DP}{DP+YN} \cdot \frac{DN}{DN+YP} \cdot \frac{N}{N+YP} + c_2 \cdot \frac{DP}{DP+YP} \quad (6.10)$$

Denklemdaki değişkenlerin tanımları ise aşağıdaki gibidir [84]:

- DP , doğru pozitiflerdir ve kuralla aynı sınıf etiketine sahip olan, kural tarafından kapsanan örneklerin sayısıdır.
- YP , yanlış pozitiflerdir ve kuraldan farklı sınıf etiketine sahip olan, kural tarafından kapsanan örneklerin sayısıdır.
- YN , yanlış negatiflerdir ve kural tarafından kapsanmayan fakat kuralla aynı sınıf etiketine sahip örneklerin sayısıdır.
- DN ise doğru negatiflerdir ve kural tarafından kapsanmayan ve kuralla da aynı etikete sahip olmayan örneklerin sayısıdır.
- N ise veri setindeki toplam veri sayısıdır.
- c_1 ve c_2 ($c_1+c_2=1$) değerleri ise ağırlıklardır ve en iyi sonucu verecek şekilde kullanıcı tarafından belirlenir.

7. DENEYSEL SONUÇLAR

Bu tez çalışmasında önerilen hibrit EM-PSO-DE ve HPO-BBBC algoritmalarının küresel optimizasyon problemlerini çözmedeki başarıları, literatürde yaygın olarak kullanılan 9 farklı kalite testi fonksiyonu ile test edilmiştir. KEEL [82] ve UCI [83] veri ambarlarından alınan üç farklı veri seti kullanılarak ise önerilen hibrit algoritmaların nümerik sınıflandırma kural keşfindeki performansları test edilmiştir. Bu bölümde, algoritmalar tarafından elde edilen sonuçlar, karşılaştırmalı olarak sunulmuştur.

7.1. Hibrit EM-PSO-DE Sonuçları

Hibrit EM-PSO-DE ile algoritmayı oluşturan bileşenleri, ilk önce ayrıntıları Bölüm 6.1' de verilen kalite testi fonksiyonları üzerinde test edilmiştir. Algoritmalar arasında adil bir karşılaştırma yapmak amacıyla, tüm algoritmaların popülasyon sayıları 30 ve maksimum iterasyon sayıları ise 1000 olarak ayarlanmıştır. Tüm kalite testi fonksiyonları 10, 20 ve 30 boyut ile test edilmiştir. Her algoritma her bir test fonksiyonu için 30 defa çalıştırılmıştır ve karşılaştırmalar; ortalama, ortanca, en iyi, en kötü ve standart sapma değerlerine göre yapılmıştır. Ortalama, ortanca, en iyi ve en kötü sonuçlar algoritmaların küresel yakınsama başarılarını, standart sapma ise algoritmanın ne kadar kararlı çalıştığını gösterir [24].

Bazı algoritmaların parametrelerine, kalite testi fonksiyonlarında en iyi sonucu verecek şekilde farklı değerler verilmiştir. EM için tüm fonksiyonlarda $LSITER=5$ olarak alınmıştır. Yerel aramada kullanılan δ ise Rastrigin, Rosenbrock, Griewank, Ackley ve Levy fonksiyonları için 0.1, Alpine, Quentic ve Trigonometric fonksiyonları için 0.001 ve Sphere fonksiyonu için ise 0.0001 olarak ayarlanmıştır. PSO' da tüm test fonksiyonları için; w değişkeni 0.9' dan 0.2' ye her iterasyonda doğrusal olarak azalacak şekilde, $c_1=2$ ve $c_2=2$ olarak ayarlanmıştır. EM-PSO' da ise $LSITER$ ve δ değerleri EM ile aynı, $c_1=2$, $c_2=2$ ve w değişkeni 0.9' dan 0.6' ya her iterasyonda doğrusal olarak azalacak şekilde seçilmiştir. DE' de ise tüm test fonksiyonları için; $F=0.2$ ve $CR=0.1$ olarak ayarlanmıştır. Önerilen hibrit EM-PSO-DE algoritmasında ise, Sphere, Alpine, Quintic ve Trigonometrik fonksiyonları için $CR=0.1$, $F=0.0001$, w ise 0.9'dan 0.8' e doğrusal olarak azalacak şekilde, Griewank, Ackley ve Levy fonksiyonları için $CR=0.15$, $F=0.2$, w ise 0.9'dan 0.7' e doğrusal olarak azalacak şekilde, Rastrigin fonksiyonu için $CR=0.8$, $F=0.0001$, w ise 0.9'dan 0.6' ya

doğrusal olarak azalacak şekilde ve Rosenbrock fonksiyonu için ise $CR=0.9$, $F=0.8$, w ise 0.9'dan 0.2'ye doğrusal olarak azalacak şekilde ayarlanmıştır.

10, 20 ve 30 boyutlu fonksiyonlardan 30 bağımsız çalıştırmada elde edilen karşılaştırmalı sonuçlar Tablo 7.1, Tablo 7.2 ve Tablo 7.3'te görülmektedir. Sonuçlar incelendiğinde 10 ve 20 boyutlu test fonksiyonlarının neredeyse tamamında önerilen hibrit yöntemin başarılı sonuçlar verdiği görülmüştür. 30 boyutlu test fonksiyonlarında ise önerilen hibrit algoritma Rastrigin, Griewank, Levy ve Trigonometric fonksiyonları dışında daha iyi sonuçlar verdiği görülmektedir. Şekil 7.1, Şekil 7.2 ve Şekil 7.3'te ise algoritmaların yakınsama grafikleri görülmektedir. Grafikler incelendiğinde, yakınsama hızı konusunda öne çıkan bir algoritmanın bulunmadığı görülmektedir. Örneğin Sphere fonksiyonunda EM-PSO-DE algoritmasının yakınsama hızı daha iyi iken, Rastrigin fonksiyonu için EM'nin daha ön plana çıktığı görülmektedir. Ayrıca Tablo 7.4'teki 10 boyutlu test fonksiyonlarındaki ortalama çalışma sürelerine bakıldığında önerilen hibrit yöntem, EM'nin çalışma süresini önemli ölçüde azaltmıştır.

Tüm bu sonuçlar ışığında, EM algoritmasının hareket evresinde PSO'nun hız ve konum güncelleme operatörleri kullanılması ile EM'nin küresel optimizasyondaki başarısının bir miktar geliştirildiği görülmüştür. Yerel Arama evresinde ise DE algoritması kullanılması ile birlikte EM'nin küresel optimizasyondaki başarısı büyük bir biçimde artmış ayrıca çalışma süresinde ise önemli bir düşüş görülmüştür.

Tablo 7.1. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki sonuçları

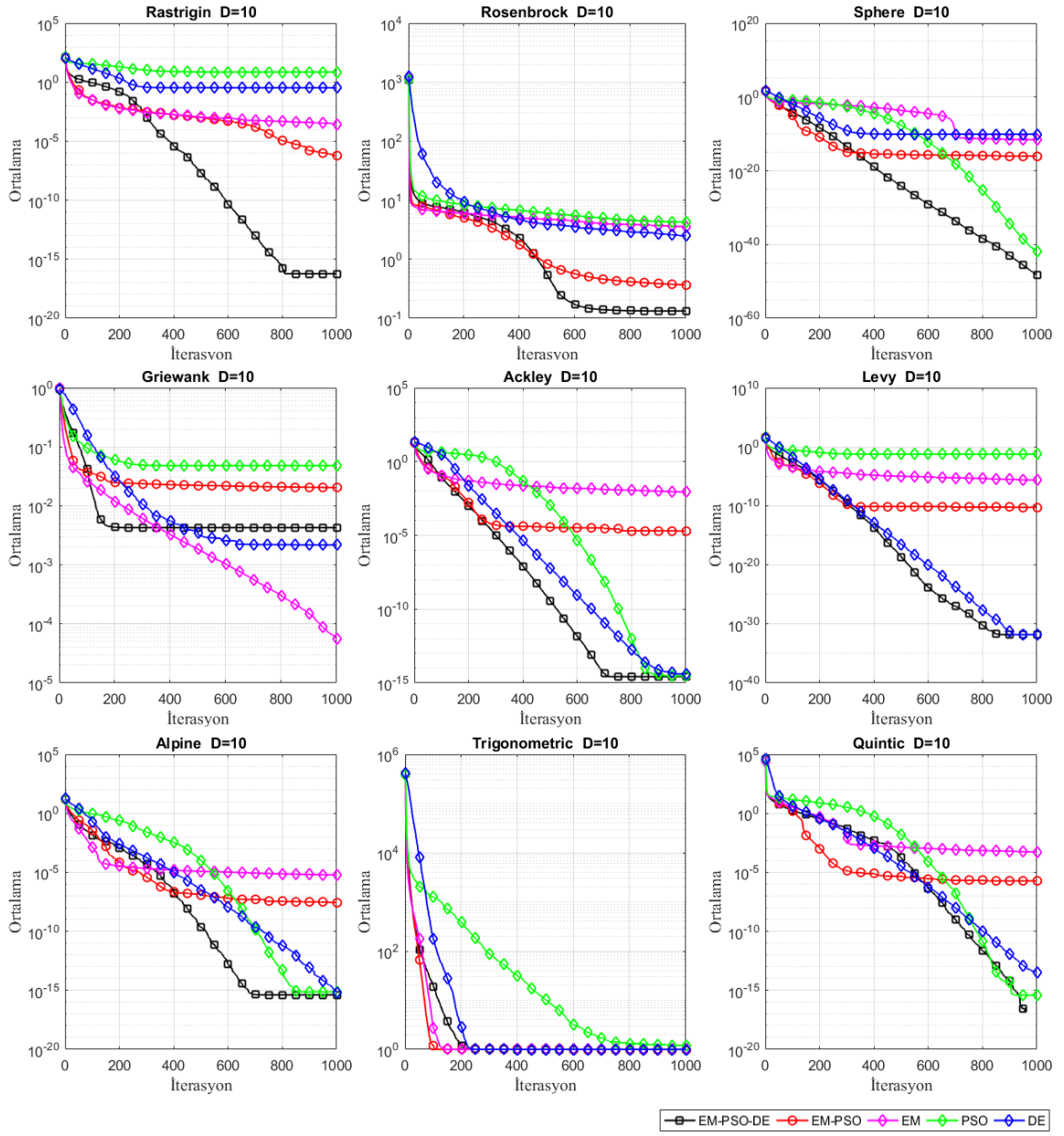
Fonksiyon (10-D)		EM-PSO- DE	EM-PSO	EM	PSO	DE
Rastrigin	Ortalama	5.92E-17	5.66E-07	0.000276	7.130538	0.341406
	Ortanca	0	9.58E-08	0.000236	5.969754	0
	En iyi	0	6.94E-10	3.10E-05	2.984877	0
	En Kötü	1.78E-15	9.07E-06	0.000849	13.92942	1.989918
	Std. Spm.	3.19E-16	1.66E-06	0.000185	2.985795	0.545790
Rosenbrock	Ortalama	0.132892	0.368356	3.480668	4.245480	2.491194
	Ortanca	2.37E-06	0.095898	3.578692	3.956980	2.202780
	En iyi	2.16E-07	0.055352	0.932733	3.329944	0.020194
	En Kötü	3.986580	4.204134	5.975678	7.963440	6.517844
	Std. Spm.	0.715612	1.018440	1.099378	1.058443	1.788919
Sphere	Ortalama	5.67E-49	8.76E-17	2.59E-12	1.34E-42	7.87E-11
	Ortanca	8.88E-50	3.19E-18	1.82E-12	7.39E-46	2.40E-36
	En iyi	9.62E-53	9.95E-23	7.11E-13	6.31E-49	2.90E-37
	En Kötü	8.00E-48	8.75E-16	1.29E-11	2.90E-41	2.36E-09
	Std. Spm.	1.48E-48	2.23E-16	2.35E-12	5.39E-42	4.24E-10
Griewank	Ortalama	0.004271	0.020401	5.62E-05	0.048039	0.002180
	Ortanca	0	0.012316	2.12E-12	0.034462	1.06E-05
	En iyi	0	0	3.39E-13	0	0
	En Kötü	0.017236	0.122347	0.000987	0.220396	0.012316
	Std. Spm.	0.005281	0.027852	0.000198	0.046633	0.003553
Ackley	Ortalama	2.66E-15	1.98E-05	0.008780	3.02E-15	3.73E-15
	Ortanca	2.66E-15	1.03E-05	0.008432	2.66E-15	2.66E-15
	En iyi	2.66E-15	9.02E-08	0.004213	2.66E-15	2.66E-15
	En Kötü	2.66E-15	0.000104	0.015843	6.22E-15	6.22E-15
	Std. Spm.	0	2.61E-05	0.003005	1.07E-15	1.63E-15
Levy	Ortalama	1.50E-32	5.54E-11	2.38E-06	0.057592	1.50E-32
	Ortanca	1.50E-32	1.04E-13	2.34E-06	1.50E-32	1.50E-32
	En iyi	1.50E-32	3.87E-20	5.30E-07	1.50E-32	1.50E-32
	En Kötü	1.50E-32	1.56E-09	5.12E-06	1.727747	1.50E-32
	Std. Spm.	1.09E-47	2.79E-10	1.06E-06	0.310140	1.09E-47
Alpine	Ortalama	4.11E-16	2.79E-08	6.08E-06	7.42E-16	6.95E-16
	Ortanca	4.16E-16	1.04E-08	5.69E-06	6.11E-16	6.41E-17
	En iyi	3.90E-43	6.97E-11	2.83E-06	3.33E-38	1.19E-19
	En Kötü	1.22E-15	3.16E-07	1.07E-05	2.72E-15	1.58E-14
	Std. Spm.	3.70E-16	5.92E-08	1.70E-06	6.95E-16	2.81E-15
Trigonometric	Ortalama	1+4.11E-34	1+1.08E-11	1+1.24E-06	1.209413	1+2.03E-20
	Ortanca	1	1+1.39E-12	1+1.19E-06	1+1.21E-30	1+1.36E-31
	En iyi	1	1+2.61E-14	1+2.69E-07	1	1
	En Kötü	1+1.23E-32	1+1.73E-10	1+3.73E-06	5.038872	1+6.08E-19
	Std. Spm.	2.21E-33	3.20E-11	7.34E-07	0.757447	1.09E-19
Quentic	Ortalama	0	1.87E-06	0.000536	3.85E-16	3.09E-14
	Ortanca	0	8.54E-07	0.000536	0	1.55E-15
	En iyi	0	1.13E-08	0.000190	0	0
	En Kötü	0	1.46E-05	0.000984	3.55E-15	6.45E-13
	Std. Spm.	0	3.02E-06	0.000214	1.06E-15	1.16E-13

Tablo 7.2. EM-PSO-DE ve bileşenlerinin 20 boyutlu kalite testi fonksiyonlarındaki sonuçları

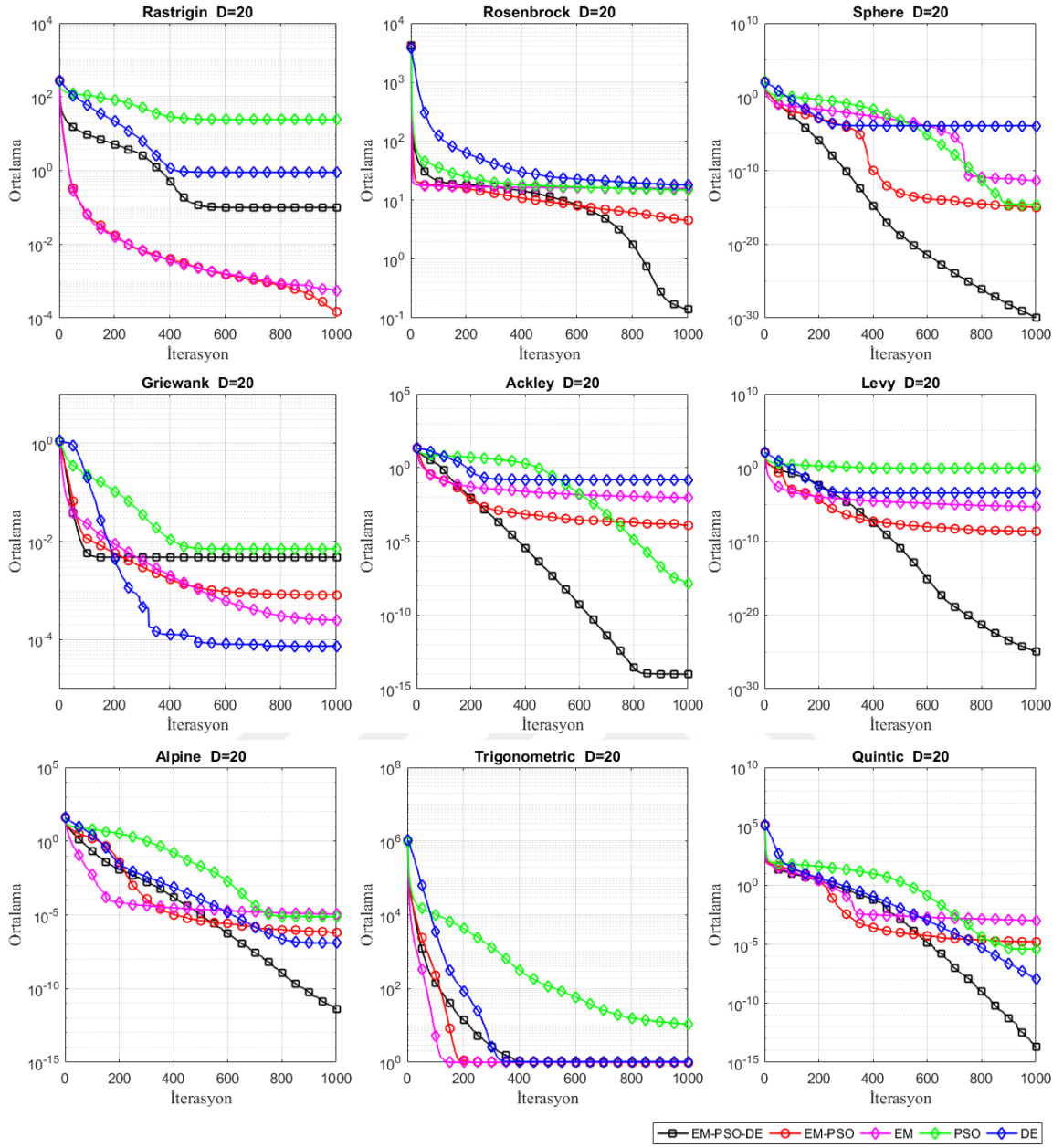
Fonksiyon (20-D)		EM-PSO- DE	EM-PSO	EM	PSO	DE
Rastrigin	Ortalama	0.099496	0.000146	0.000561	24.27697	0.897943
	Ortanca	8.45E-12	0.000125	0.000526	24.37646	0.994959
	En iyi	1.53E-13	3.36E-05	6.08E-05	11.93949	0
	En Kötü	0.994959	0.000389	0.001408	36.81343	2.984877
	Std. Spm.	0.298488	9.75E-05	0.000307	6.804608	0.859241
Rosenbrock	Ortalama	0.141056	4.506214	15.74324	14.87585	17.90114
	Ortanca	0.002987	4.491537	15.90783	14.61108	16.15566
	En iyi	1.46E-05	2.166165	12.47818	14.23841	7.373397
	En Kötü	3.987743	9.767454	16.29167	19.06464	65.66489
	Std. Spm.	0.714657	1.429763	0.654812	1.087013	10.71494
Sphere	Ortalama	1.09E-30	1.05E-15	4.46E-12	2.33E-15	0.000124
	Ortanca	7.21E-31	8.27E-16	3.62E-12	1.75E-19	4.97E-24
	En iyi	1.20E-31	1.12E-16	1.50E-12	1.55E-21	2.16E-25
	En Kötü	6.98E-30	3.59E-15	1.50E-11	6.84E-14	0.003364
	Std. Spm.	1.32E-30	9.06E-16	2.71E-12	1.23E-14	0.000605
Griewank	Ortalama	0.004762	0.000821	0.000251	0.007063	7.41E-05
	Ortanca	0	5.44E-15	2.03E-12	0.007396	0
	En iyi	0	4.44E-16	3.23E-13	0	0
	En Kötü	0.024590	0.014772	0.007523	0.019678	0.001493
	Std. Spm.	0.006573	0.003137	0.001350	0.005606	0.000284
Ackley	Ortalama	1.00E-14	0.000126	0.009304	1.50E-08	0.151905
	Ortanca	1.15E-14	0.000131	0.009207	3.88E-09	9.07E-12
	En iyi	6.22E-15	4.62E-05	0.005335	7.82E-11	3.52E-12
	En Kötü	1.33E-14	0.000203	0.018918	1.64E-07	2.960720
	Std. Spm.	3.42E-15	3.65E-05	0.002614	3.17E-08	0.552842
Levy	Ortalama	1.07E-25	2.44E-09	5.11E-06	0.879017	0.000380
	Ortanca	8.27E-26	1.61E-09	4.58E-06	1.95E-17	7.42E-24
	En iyi	1.68E-26	8.46E-11	2.22E-06	1.68E-21	6.55E-25
	En Kötü	4.45E-25	1.55E-08	1.86E-05	5.183240	0.011399
	Std. Spm.	8.07E-26	3.43E-09	3.20E-06	1.385932	0.002046
Alpine	Ortalama	3.96E-12	6.04E-07	1.11E-05	7.68E-06	1.25E-07
	Ortanca	2.05E-14	3.88E-07	1.10E-05	5.48E-10	7.46E-11
	En iyi	2.27E-16	5.39E-08	6.89E-06	6.16E-12	9.68E-13
	En Kötü	4.02E-11	2.19E-06	1.58E-05	0.000230	3.74E-06
	Std. Spm.	9.73E-12	5.97E-07	2.40E-06	4.13E-05	6.71E-07
Trigonometric	Ortalama	1+2.36E-22	1+5.40E-09	1+2.77E-06	11.00811	1.044870
	Ortanca	1+8.81E-23	1+4.99E-09	1+2.57E-06	9.975227	1+1.34E-19
	En iyi	1+4.68E-24	1+1.52E-09	1+5.86E-07	1.897438	1+6.60E-21
	En Kötü	1+1.69E-21	1+1.26E-08	1+4.96E-06	23.43908	1+0.897438
	Std. Spm.	3.98E-22	2.93E-09	1.05E-06	5.923083	0.177602
Quentic	Ortalama	2.14E-14	1.67E-05	0.001081	3.96E-06	1.18E-08
	Ortanca	0	1.50E-05	0.001052	3.57E-08	1.48E-09
	En iyi	0	3.13E-06	0.000592	5.95E-10	9.13E-11
	En Kötü	6.13E-13	5.93E-05	0.001981	5.61E-05	1.27E-07
	Std. Spm.	1.10E-13	1.05E-05	0.000321	1.18E-05	3.03E-08

Tablo 7.3. EM-PSO-DE ve bileşenlerinin 30 boyutlu kalite testi fonksiyonlarındaki sonuçları

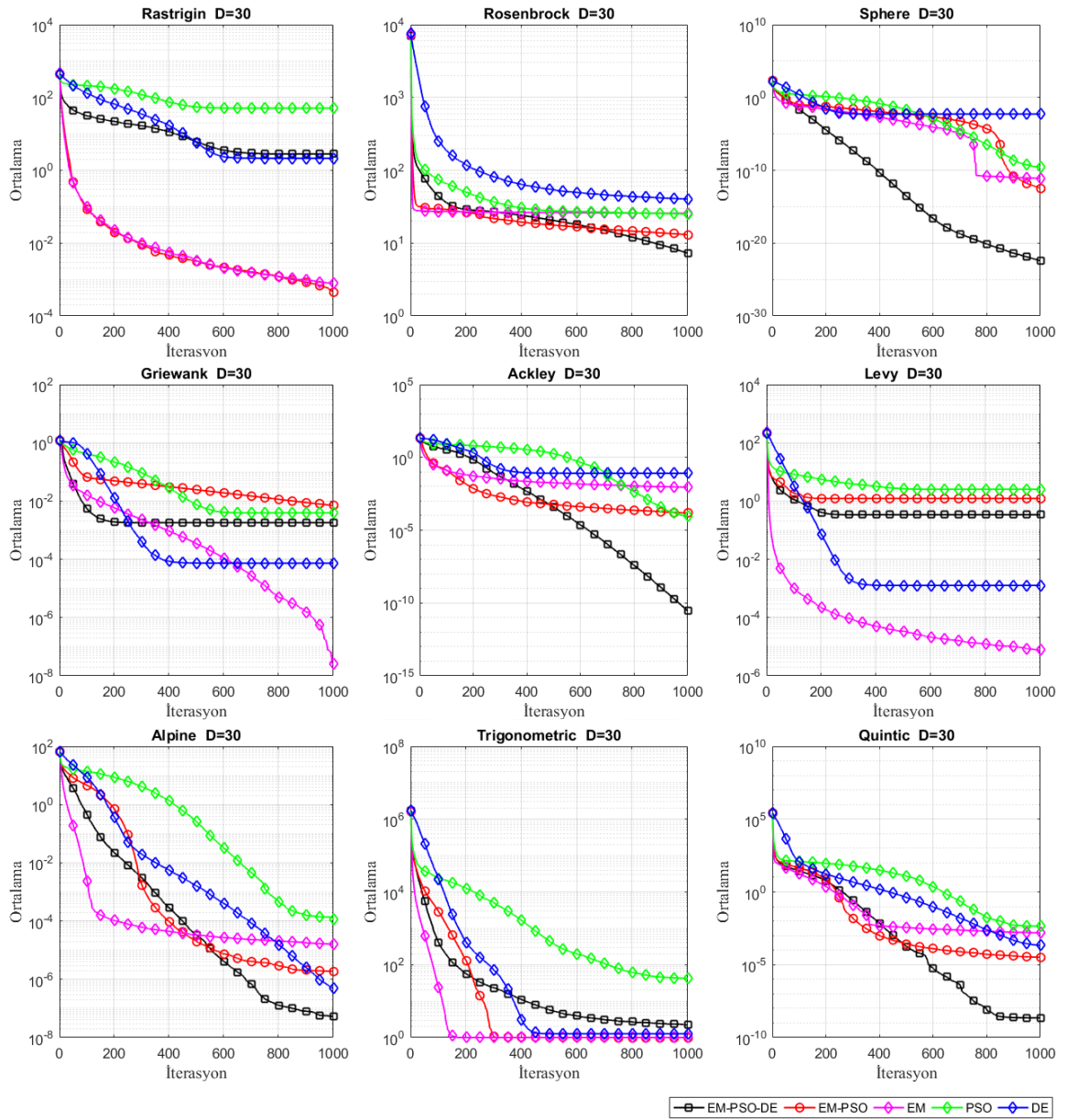
Fonksiyon (30-D)		EM-PSO- DE	EM-PSO	EM	PSO	DE
Rastrigin	Ortalama	2.802671	0.000450	0.000792	51.01051	2.135369
	Ortanca	2.985715	0.000375	0.000766	46.76298	1.993887
	En iyi	5.56E-06	0.000119	0.000325	27.85884	1.34E-08
	En Kötü	6.974884	0.001479	0.001971	103.4755	4.988211
	Std. Spm.	1.774301	0.000271	0.000327	19.54991	1.266481
Rosenbrock	Ortalama	7.306313	13.04869	25.62783	25.14491	40.17670
	Ortanca	7.045739	12.93517	25.62508	25.10353	26.93613
	En iyi	4.099483	10.50175	23.93553	23.83426	7.608328
	En Kötü	10.73795	16.27164	26.35233	27.54836	109.5771
	Std. Spm.	1.512751	1.343416	0.444581	0.554921	25.03242
Sphere	Ortalama	3.77E-23	3.44E-13	7.94E-12	2.64E-10	0.005383
	Ortanca	3.37E-23	6.14E-15	7.20E-12	4.03E-11	1.24E-11
	En iyi	2.86E-24	1.62E-15	3.16E-12	1.94E-13	4.26E-19
	En Kötü	1.14E-22	8.79E-12	1.67E-11	3.80E-09	0.157990
	Std. Spm.	2.40E-23	1.58E-12	3.69E-12	7.07E-10	0.028340
Griewank	Ortalama	0.001806	0.007014	2.64E-08	0.003943	7.35E-05
	Ortanca	5.55E-17	0.006464	1.46E-12	3.41E-10	5.71E-10
	En iyi	0	1.54E-14	7.11E-13	1.78E-13	0
	En Kötü	0.017236	0.026872	7.92E-07	0.014772	0.000933
	Std. Spm.	0.004670	0.007844	1.42E-07	0.005113	0.000217
Ackley	Ortalama	2.82E-11	0.000155	0.008868	9.30E-05	0.081121
	Ortanca	2.09E-11	0.000145	0.008762	2.01E-05	6.39E-06
	En iyi	5.82E-12	4.74E-05	0.005721	2.43E-06	2.62E-09
	En Kötü	1.15E-10	0.000380	0.011736	0.001132	0.622818
	Std. Spm.	2.34E-11	7.06E-05	0.001545	0.000207	0.193060
Levy	Ortalama	0.339360	1.227540	7.70E-06	2.530766	0.001265
	Ortanca	6.70E-21	4.06E-09	7.23E-06	2.182071	2.61E-10
	En iyi	4.97E-22	1.00E-09	3.29E-06	1.29E-12	9.62E-19
	En Kötü	1.727747	10.54765	1.46E-05	6.910987	0.028919
	Std. Spm.	0.535304	2.616939	2.77E-06	1.858858	0.005367
Alpine	Ortalama	5.32E-08	1.88E-06	1.61E-05	0.000114	4.85E-07
	Ortanca	1.84E-09	1.18E-06	1.56E-05	1.22E-05	8.50E-08
	En iyi	8.65E-14	1.79E-07	1.04E-05	5.49E-07	2.15E-09
	En Kötü	4.29E-07	7.39E-06	2.59E-05	0.002141	3.69E-06
	Std. Spm.	1.11E-07	2.07E-06	3.36E-06	0.000386	9.03E-07
Trigonometric	Ortalama	2.227868	1+1.58E-08	1+4.31E-06	41.92968	1.269253
	Ortanca	1.897292	1+1.31E-08	1+4.47E-06	43.00083	1+9.37E-07
	En iyi	1+4.32E-16	1+4.62E-09	1+2.05E-06	12.66896	1+4.76E-14
	En Kötü	8.179988	1+5.47E-08	1+1.04E-05	94.80335	3.244157
	Std. Spm.	1.705714	1.08E-08	1.76E-06	20.41060	0.586274
Quentic	Ortalama	2.22E-09	3.26E-05	0.001541	0.004534	0.000215
	Ortanca	2.50E-12	3.37E-05	0.001457	0.000875	1.26E-05
	En iyi	4.62E-14	6.94E-06	0.000942	1.22E-05	4.86E-07
	En Kötü	6.60E-08	5.74E-05	0.002876	0.073466	0.005522
	Std. Spm.	1.18E-08	1.33E-05	0.000371	0.013172	0.000987



Şekil 7.1. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonları için yakınsama grafikleri



Şekil 7.2. EM-PSO-DE ve bileşenlerinin 20 boyutlu kalite testi fonksiyonları için yakınsama grafikleri



Şekil 7.3. EM-PSO-DE ve bileşenlerinin 30 boyutlu kalite testi fonksiyonları için yakınsama grafikleri

Tablo 7.4. EM-PSO-DE ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki ortalama çalışma süreleri(sn)

Fonksiyon (10-D)		EM-PSO-DE	EM-PSO	EM	PSO	DE
Rastrigin	Ortalama	2.679	9.478	8.068	0.502	0.750
	Std. Spm.	0.029	0.088	0.046	0.004	0.011
Rosenbrock	Ortalama	1.732	6.792	5.856	0.357	0.471
	Std. Spm.	0.021	0.062	0.025	0.003	0.005
Sphere	Ortalama	1.709	6.714	5.019	0.351	0.482
	Std. Spm.	0.018	0.038	0.026	0.004	0.003
Griewank	Ortalama	1.816	7.428	5.782	0.383	0.540
	Std. Spm.	0.036	0.349	0.023	0.004	0.004
Ackley	Ortalama	1.802	8.091	6.602	0.378	0.523
	Std. Spm.	0.029	0.131	0.047	0.004	0.006
Levy	Ortalama	3.165	15.098	12.457	0.595	0.960
	Std. Spm.	0.020	0.055	0.047	0.006	0.004
Alpine	Ortalama	1.801	7.443	5.646	0.370	0.501
	Std. Spm.	0.024	0.067	0.038	0.004	0.003
Trigonometric	Ortalama	1.844	7.497	5.778	0.376	0.511
	Std. Spm.	0.019	0.085	0.027	0.003	0.005
Quentic	Ortalama	2.645	12.405	9.573	0.524	0.833
	Std. Spm.	0.026	0.240	0.038	0.005	0.009

Hibrit EM-PSO-DE ve bileşenleri, 10 boyutlu test fonksiyonlarında uygunluk fonksiyonunun çağırılma sayısı sabit tutularak da test edilmiştir. Her bir algoritmanın bitim şartı; maksimum uygunluk fonksiyonu çağırılma sayısı 10000 olarak ayarlanmıştır ve her algoritma tüm test fonksiyonları için 30 defa çalıştırılmıştır. Sonuçlar Tablo 7.5’ te verilmiştir.

Tablo 7.5. EM-PSO-DE ve bileşenlerinin, 10 boyutlu test fonksiyonları için uygunluk fonksiyonunun 10000 defa çağırılmasındaki sonuçları

Fonksiyon (10-D)		EM-PSO- DE	EM-PSO	EM	PSO	DE
Rastrigin	Ortalama	1,819319	3,623607	3,223963	10,09423	3,868681
	Ortanca	1,805713	3,440635	3,261298	10,04817	3,678576
	En iyi	0,643688	0,830051	0,296123	3,232906	0,974482
	En Kötü	3,697432	9,190114	8,529203	17,395	7,349464
	Std. Spm.	0,694272	1,999115	1,879877	3,710629	1,721389
Rosenbrock	Ortalama	7,319865	9,238638	8,379273	7,37898	13,90812
	Ortanca	7,361939	9,040715	8,390278	7,262421	10,87312
	En iyi	4,661296	7,02082	6,990965	5,668563	4,79371
	En Kötü	9,549865	11,91945	9,750194	9,623947	37,26225
	Std. Spm.	0,937825	1,109052	0,526827	0,95864	8,188738
Sphere	Ortalama	3,6E-05	0,134837	0,01625	0,001591	5,19E-05
	Ortanca	3,1E-05	0,057769	0,013808	0,001576	4,38E-05
	En iyi	7,54E-06	0,005391	0,006102	0,000142	1,17E-05
	En Kötü	8,88E-05	0,969583	0,03913	0,004976	0,000155
	Std. Spm.	2,23E-05	0,196846	0,007475	0,001159	3,09E-05
Griewank	Ortalama	0,017142	0,099018	0,013954	0,045551	0,045288
	Ortanca	0,009904	0,100464	0,010524	0,037587	0,042671
	En iyi	6,26E-06	0,015041	0,001761	0,000235	0,005938
	En Kötü	0,155331	0,269202	0,084681	0,223011	0,0933
	Std. Spm.	0,02959	0,047158	0,014719	0,043709	0,023915
Ackley	Ortalama	0,054995	2,999901	2,144665	0,582398	0,167585
	Ortanca	0,049024	2,935267	2,199547	0,483277	0,132559
	En iyi	0,016586	1,513795	0,721268	0,092881	0,048203
	En Kötü	0,110931	5,072041	3,089736	2,070325	0,603932
	Std. Spm.	0,024974	0,895877	0,573978	0,384275	0,115535
Levy	Ortalama	0,018324	0,457219	0,033845	0,081931	0,000128
	Ortanca	0,000164	0,168921	0,032995	0,001912	0,000108
	En iyi	1,68E-05	0,015534	0,01058	0,000133	4,86E-05
	En Kötü	0,45438	2,614085	0,068361	2,404413	0,00037
	Std. Spm.	0,082545	0,633918	0,017099	0,431276	7,7E-05
Alpine	Ortalama	0,00917	0,9085	0,126173	0,015942	0,005636
	Ortanca	0,00683	0,882342	0,097211	0,013812	0,005623
	En iyi	0,001494	0,226555	0,031962	0,005288	0,002799
	En Kötü	0,035215	1,733296	0,379905	0,055243	0,010246
	Std. Spm.	0,007693	0,404085	0,084611	0,010324	0,001584
Trigonometric	Ortalama	18,33204	650,4879	193,6671	68,1306	16,94336
	Ortanca	19,26165	438,2547	176,3697	64,01023	14,88493
	En iyi	5,588382	127,6723	77,50458	37,48149	3,519106
	En Kötü	35,60299	2391,366	394,8374	125,4421	40,83561
	Std. Spm.	7,25798	561,1932	71,03683	18,07993	7,994112
Quentic	Ortalama	0,189672	6,084589	4,371161	2,291464	0,921311
	Ortanca	0,128644	6,341625	4,268191	2,244452	0,90252
	En iyi	0,04352	2,121047	1,86695	0,828645	0,440263
	En Kötü	1,018518	9,786046	7,431537	4,360317	1,552894
	Std. Spm.	0,192244	2,003285	1,338498	0,933509	0,291417

Hibrit EM-PSO-DE ve bileşenlerinin sınıflandırma kural keşfindeki performansları Tablo 6.1’ de özellikleri verilen veri setleri kullanılarak test edilmiştir. Algoritmalar arasında adil bir karşılaştırma yapmak amacıyla, tüm algoritmaların başlangıç popülasyon sayısı 50 ve her bir kural için maksimum iterasyon sayısı 100 olarak ayarlanmıştır. Her algoritma her bir veri setinde 30 defa çalıştırılmıştır ve karşılaştırmalar, ortalama doğru sınıflandırma değerlerine göre yapılmıştır. Deneysel çalışmada algoritmaların parametreleri ise; EM için $LSITER=5$, $\delta=0.1$ olarak ayarlanmıştır. EM-PSO algoritmasının parametreleri $LSITER=5$, $\delta=0.1$, $c_1=2$, $c_2=2$ ve w değişkeni 0.9’ dan 0.6’ ya her iterasyonda doğrusal olarak azalacak şekilde seçilmiştir. PSO’ da w değişkeni 0.9’ dan 2’ ye her iterasyonda doğrusal olarak azalacak şekilde, $c_1=2$ ve $c_2=2$ olarak ayarlanmıştır. DE’ de ise $F=0.2$ ve $CR=0.1$ olarak ayarlanmıştır. Önerilen hibrit EM-PSO-DE algoritmasında ise $LSITER=5$, $CR=0.2$, $F=0.1$, w ise 0.9’ dan 0.6’ ya doğrusal olarak azalacak şekilde ayarlanmıştır. Algoritmaların sonlandırma şartı Banknote veri seti için; kural sayısının 10’ a erişmesi veya keşfedilen kurallar tarafından kapsanmayan veri sayısının 430’ un altına düşmesi, Phoneme veri seti için; kural sayısının 10’ a erişmesi veya kurallar tarafından kapsanmayan veri sayısının 1150’ nin altına düşmesi, Banana veri seti için; kural sayısının 10’ a erişmesi veya kurallar tarafından kapsanmayan veri sayısının 4300’ ün altına düşmesi olarak belirlenmiştir.

30 bağımsız çalıştırmada elde edilen karşılaştırmalı sonuçlar Tablo 7.6, Tablo 7.7 ve Tablo 7.8’ de görülmektedir. Sonuçlar incelendiğinde Phoneme veri setinde önerilen hibrit algoritmanın en yüksek doğrulukta test verilerini sınıflandırdığı görülmüştür. Banana ve Banknote veri setlerinde ise EM algoritması test verilerini en yüksek doğrulukta sınıflandırdığı, EM-PSO-DE’ nin ise ikinci sırada kaldığı görülmüştür. Buna karşın, çalışma sürelerine bakıldığında önerilen hibrit yöntem EM’ nin çalışma süresini önemli bir ölçüde azaltmıştır.

Tablo 7.6. EM-PSO-DE ve bileşenlerinden Banana veri setinde elde edilen sonuçlar

BANANA					
	EM-PSO-DE	EM-PSO	EM	PSO	DE
1	463	448	470	446	449
2	454	451	466	418	429
3	466	467	469	457	451
4	454	458	471	412	461
5	469	449	467	466	463
6	456	460	464	457	464
7	453	455	466	450	459
8	452	459	464	444	462
9	470	440	467	419	454
10	450	466	467	412	457
11	457	450	466	461	431
12	464	412	464	412	457
13	465	455	467	445	453
14	455	454	468	449	454
15	457	450	465	451	456
16	455	425	466	448	462
17	466	468	464	452	467
18	470	408	464	451	463
19	465	424	464	393	460
20	454	453	464	453	460
21	456	454	464	449	439
22	458	407	467	451	454
23	452	451	468	447	449
24	455	391	469	446	452
25	459	443	467	412	450
26	467	448	466	451	463
27	466	450	467	424	459
28	454	452	470	456	468
29	455	412	469	412	461
30	468	450	468	401	469
Ort.	459.5	443.666	466.6	438.166	455.866
%	86.534	83.553	87.871	82.5172	85.850
Süre (sn)	27.49	52.29	84.12	6.01	8.85

Tablo 7.7. EM-PSO-DE ve bileşenlerinden Banknote veri setinde elde edilen sonuçlar

	BANKNOTE				
	EM-PSO-DE	EM-PSO	EM	PSO	DE
1	392	359	393	387	388
2	383	393	392	375	387
3	391	375	393	379	388
4	389	384	392	375	388
5	391	394	394	375	374
6	393	374	394	375	389
7	391	375	392	384	389
8	393	383	393	375	388
9	393	375	394	359	388
10	383	389	392	359	394
11	392	374	393	375	387
12	393	375	393	375	388
13	393	375	393	375	375
14	393	375	393	359	376
15	385	366	393	379	387
16	387	358	386	375	392
17	389	375	393	375	385
18	393	390	393	359	386
19	385	375	390	379	375
20	389	375	378	366	375
21	392	380	393	359	387
22	382	375	392	375	387
23	393	377	392	359	392
24	389	366	393	359	391
25	388	382	392	382	392
26	393	378	393	375	388
27	389	375	393	375	392
28	391	386	392	375	387
29	386	375	394	366	387
30	392	393	391	359	386
Ort.	389.766	377.533	391.966	371.466	386.266
%	94.833	91.857	95.369	90.381	93.982
Süre (sn)	3.10	17.61	25.67	0.39	0.79

Tablo 7.8. EM-PSO-DE ve bileşenlerinden Phonome veri setinde elde edilen sonuçlar

	PHONOME				
	EM-PSO-DE	EM-PSO	EM	PSO	DE
1	1286	1319	1321	1288	1266
2	1299	1296	1311	1300	1331
3	1307	1303	1325	1300	1346
4	1332	1310	1321	1298	1268
5	1291	1298	1267	1312	1348
6	1315	1286	1316	1304	1302
7	1310	1301	1324	1284	1333
8	1312	1320	1318	1295	1294
9	1308	1284	1298	1273	1287
10	1325	1306	1306	1284	1318
11	1297	1252	1296	1265	1294
12	1307	1314	1311	1290	1297
13	1315	1237	1323	1281	1287
14	1308	1288	1298	1296	1321
15	1311	1302	1292	1279	1319
16	1336	1289	1314	1246	1301
17	1336	1270	1347	1307	1289
18	1340	1278	1260	1303	1294
19	1315	1310	1294	1289	1276
20	1305	1301	1314	1289	1288
21	1279	1268	1243	1278	1296
22	1322	1284	1280	1293	1317
23	1305	1276	1305	1282	1366
24	1297	1308	1284	1314	1288
25	1318	1280	1296	1311	1273
26	1314	1291	1302	1285	1304
27	1300	1276	1313	1273	1319
28	1365	1311	1300	1309	1271
29	1346	1284	1273	1283	1319
30	1308	1294	1309	1301	1306
Ort.	1313.633	1291.2	1302.033	1290.4	1303.933
%	81.088	79.703	80.372	79.654	80.489
Süre (sn)	11.99	85.62	119.42	1.24	3.60

7.2. HPO-BBBC Sonuçları

HPO-BBBC ile POA ve BB-BC Bölüm 6.1’ de ayrıntılı bahsedilen kalite testi fonksiyonları kullanılarak test edilmiştir. Tüm algoritmaların başlangıç popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak ayarlanmıştır. Tüm kalite testi fonksiyonları 10, 20 ve 30 boyut ile test edilmiştir. Her algoritma her bir test fonksiyonu için 30 defa çalıştırılmış ve karşılaştırmalar; ortalama, ortanca, en iyi, en kötü ve standart sapma değerlerine göre yapılmıştır.

Deneyisel çalışmada, POA’ da grup sayısı $M=6$, gruptaki üye sayısı $N=5$ olarak ayarlanmıştır. Tüm kalite testi fonksiyonları için; grupların birleşme olasılığı $P_m=0.007$ ve grupların silinme olasılığı $P_d=0.001$ olarak seçilmiştir. BB-BC’ de ise kütle merkezi x^c en iyi uygunluk değerine sahip olan parçacık olarak seçilmiştir. Önerilen HPO-BBBC algoritmasında ise POA’ ya benzer şekilde $M=6$, $N=5$ olarak ayarlanmıştır. Tüm kalite testi fonksiyonları için; $P_m=0.007$ ve $P_d=0.001$, genel üyeler x^c olarak seçilmiş ve x^c etrafında büyük patlama ile oluşturulacak birey sayısı $p=5$ olarak ayarlanmıştır.

10, 20 ve 30 boyutlu fonksiyonlardan 30 bağımsız çalıştırmada elde edilen karşılaştırmalı sonuçlar Tablo 7.9, Tablo 7.10 ve Tablo 7.11’ de görülmektedir. Sonuçlar incelendiğinde 10 ve 20 boyutlu test fonksiyonlarının tamamında ve 30 boyutlu test fonksiyonlarından Quintic dışında geri kalan fonksiyonların tamamında önerilen HPO-BBBC algoritmasının daha iyi sonuçlar verdiği görülmüştür. Şekil 7.4, Şekil 7.5 ve Şekil 7.6’ daki grafiklerde ise algoritmaların yakınsama eğrileri görülmektedir. Grafikler incelendiğinde HPO-BBBC algoritmasının POA ve BB-BC’ ye göre daha hızlı yakınsadığı görülmektedir. Ayrıca Tablo 7.12’ deki 10 boyutlu test fonksiyonlarındaki ortalama çalışma sürelerine bakıldığında önerilen yöntem biraz daha uzun sürelerle sahiptir.

Tablo 7.9. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki sonuçları

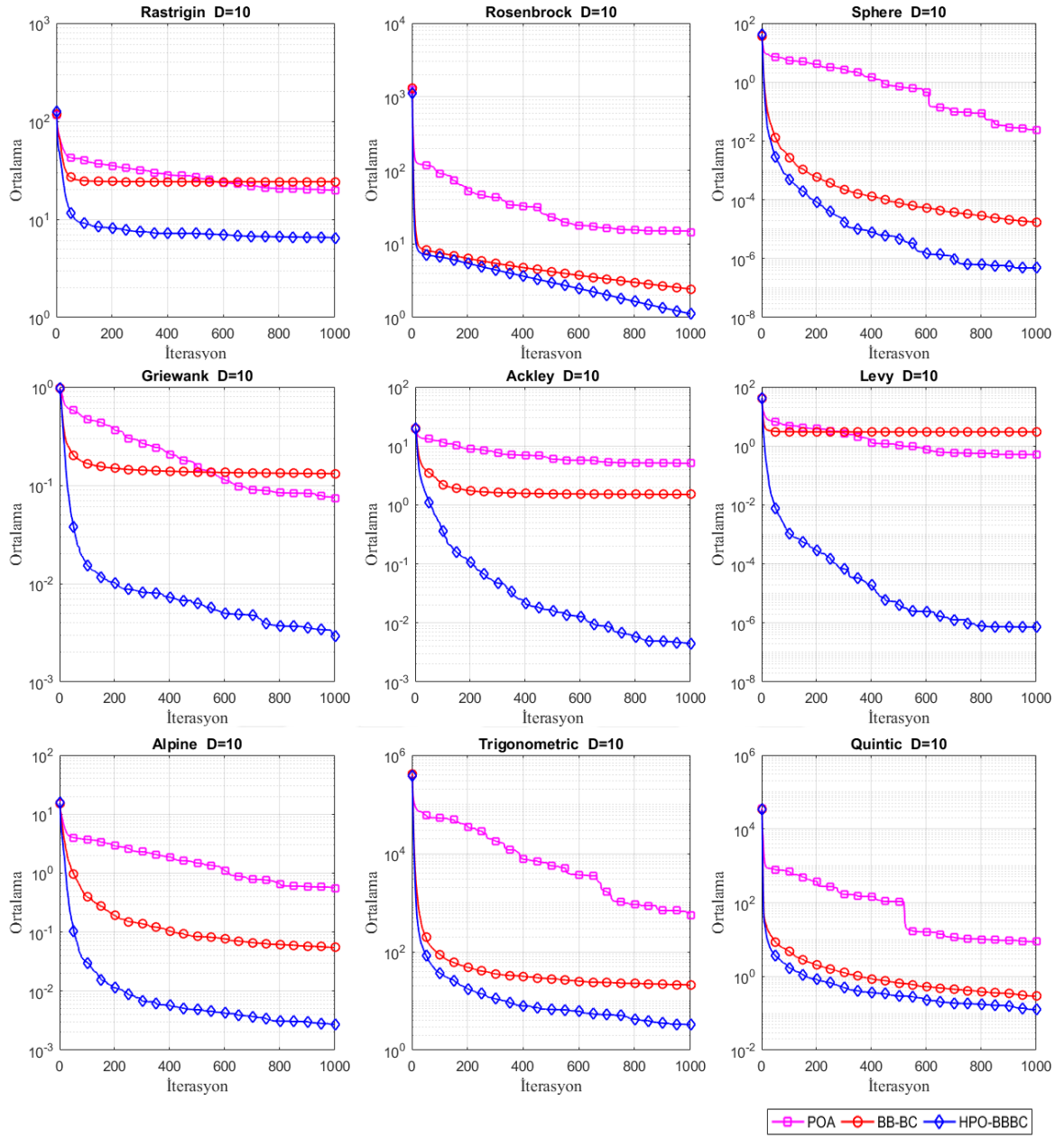
Fonksiyon (10-D)		POA	BBBC	HPO-BBBC
Rastrigin	Ortalama	19.84537	24.15733	6.482099
	Ortanca	18.74975	21.89261	6.179029
	En iyi	8.954675	5.970904	2.986103
	En Kötü	35.88499	45.77054	10.78899
	Std. Spm.	7.713776	10.43034	2.176435
Rosenbrock	Ortalama	14.48284	2.435909	1.120592
	Ortanca	9.186074	1.974159	0.990532
	En iyi	4.596860	0.500355	0.381389
	En Kötü	56.76800	7.354197	2.794516
	Std. Spm.	13.24130	1.652818	0.560288
Sphere	Ortalama	0.024066	1.70E-05	4.82E-07
	Ortanca	0.002249	1.74E-05	9.40E-08
	En iyi	2.03E-06	9.89E-06	8.76E-11
	En Kötü	0.234055	2.20E-05	3.31E-06
	Std. Spm.	0.055949	3.01E-06	8.00E-07
Griewank	Ortalama	0.073605	0.131234	0.002910
	Ortanca	0.062980	0.111268	2.77E-06
	En iyi	0.012365	0.019685	1.93E-08
	En Kötü	0.545879	0.337412	0.016725
	Std. Spm.	0.091953	0.081125	0.004723
Ackley	Ortalama	5.085986	1.504334	0.004516
	Ortanca	4.847677	0.037911	0.001608
	En iyi	2.322365	0.026534	0.000222
	En Kötü	9.732233	19.95147	0.018050
	Std. Spm.	1.758916	4.949552	0.005254
Levy	Ortalama	0.511288	2.975436	7.25E-07
	Ortanca	0.454398	2.441821	1.47E-07
	En iyi	0.000368	2.27E-05	1.42E-09
	En Kötü	2.277379	9.217470	5.19E-06
	Std. Spm.	0.570515	2.508798	1.15E-06
Alpine	Ortalama	0.554411	0.054695	0.002742
	Ortanca	0.064482	0.030420	0.001741
	En iyi	0.000658	0.001480	0.000429
	En Kötü	4.133778	0.238027	0.011246
	Std. Spm.	1.068707	0.063656	0.002585
Quintic	Ortalama	8.827116	0.289553	0.125991
	Ortanca	5.719443	0.281173	0.137459
	En iyi	0.091151	0.204449	0.013545
	En Kötü	35.95792	0.513012	0.214793
	Std. Spm.	9.129235	0.066400	0.053082
Trigonometric	Ortalama	565.8183	22.09043	4.308290
	Ortanca	226.6719	23.34842	3.278674
	En iyi	20.66961	8.170881	1.000182
	En Kötü	4026.968	36.77341	16.16963
	Std. Spm.	928.2101	7.562761	3.805618

Tablo 7.10. HPO-BBBC ve bileşenlerinin 20 boyutlu kalite testi fonksiyonlarındaki sonuçları

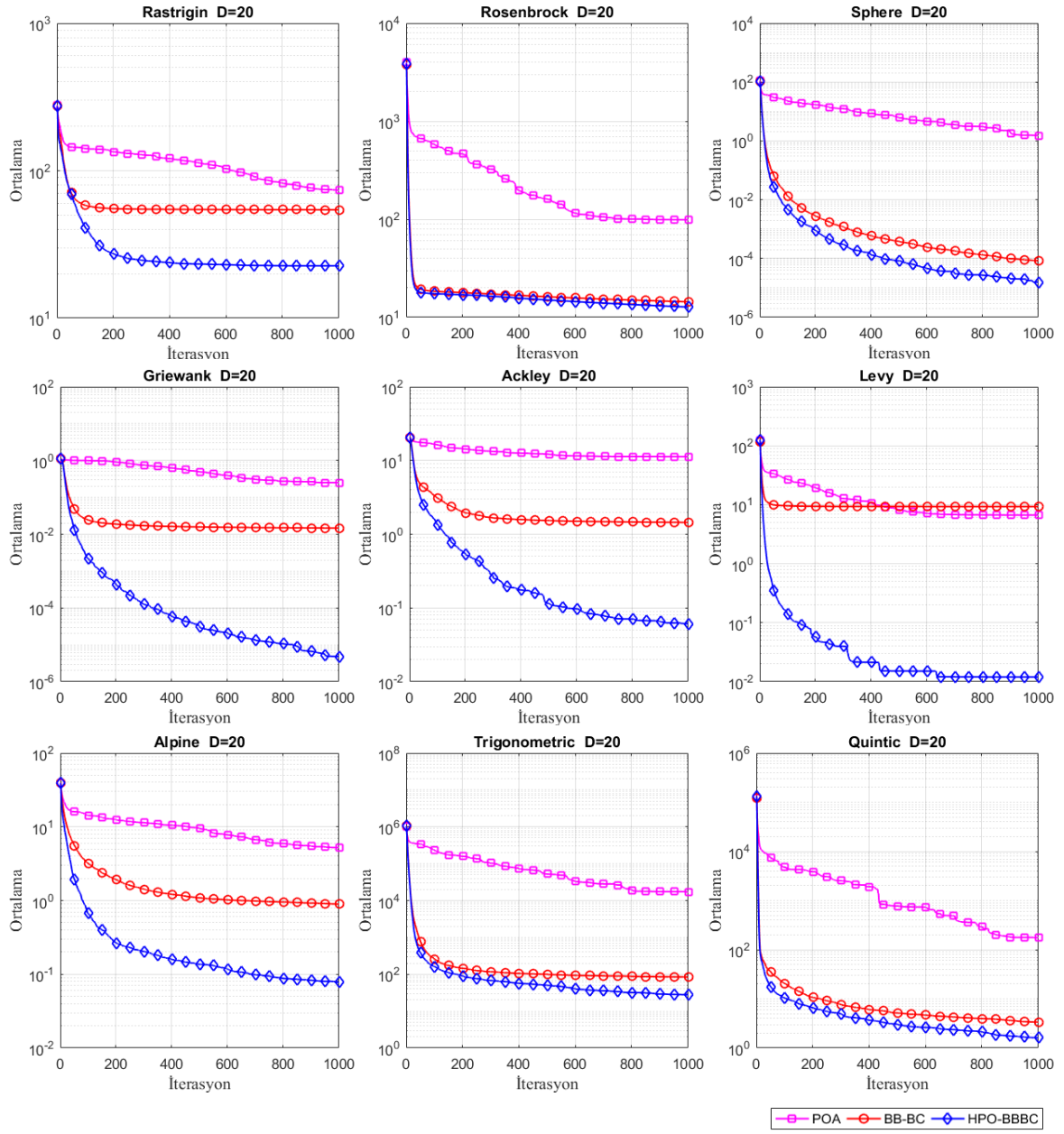
Fonksiyon (20-D)		POA	BBBC	HPO-BBBC
Rastrigin	Ortalama	73.51111	54.30417	22.62500
	Ortanca	70.26956	44.29017	23.40299
	En iyi	19.84310	25.88343	11.95693
	En Kötü	163.6371	161.8240	28.73265
	Std. Spm.	31.48446	29.04609	3.619987
Rosenbrock	Ortalama	98.97801	14.39960	12.73931
	Ortanca	100.2752	14.21637	12.66247
	En iyi	17.58133	12.74596	11.31419
	En Kötü	247.9963	17.49790	13.81702
	Std. Spm.	47.81917	0.982291	0.666578
Sphere	Ortalama	1.482802	8.50E-05	1.57E-05
	Ortanca	0.961368	8.84E-05	9.79E-06
	En iyi	0.089181	5.38E-05	9.28E-07
	En Kötü	9.454089	0.000108	5.16E-05
	Std. Spm.	1.868320	1.34E-05	1.48E-05
Griewank	Ortalama	0.250467	0.014656	4.82E-06
	Ortanca	0.194057	0.008649	3.52E-06
	En iyi	0.053801	1.89E-05	2.53E-07
	En Kötü	0.729642	0.097344	1.82E-05
	Std. Spm.	0.174217	0.023636	3.90E-06
Ackley	Ortalama	11.28958	1.446589	0.060471
	Ortanca	10.99259	0.060133	0.023549
	En iyi	6.855468	0.039344	0.009651
	En Kötü	15.87299	19.89825	1.156560
	Std. Spm.	2.412904	4.926161	0.203703
Levy	Ortalama	6.684531	9.329240	0.011990
	Ortanca	6.203046	10.13218	4.24E-05
	En iyi	1.369089	0.544002	3.00E-06
	En Kötü	19.34892	16.54491	0.089604
	Std. Spm.	3.634004	4.317301	0.030425
Alpine	Ortalama	5.222608	0.896657	0.078098
	Ortanca	4.178834	0.280156	0.049643
	En iyi	0.237761	0.070188	0.002192
	En Kötü	18.59180	10.10525	0.309427
	Std. Spm.	4.163716	1.821285	0.080302
Quintic	Ortalama	176.2390	3.340705	1.611967
	Ortanca	56.82555	1.535426	1.208547
	En iyi	27.43619	0.770679	0.414190
	En Kötü	3119.972	13.63612	9.231467
	Std. Spm.	549.0583	3.812428	1.553685
Trigonometric	Ortalama	16660.55	83.71284	27.88196
	Ortanca	10234.50	84.12003	21.41455
	En iyi	2634.085	45.25908	1.841621
	En Kötü	59308.94	112.6638	65.80363
	Std. Spm.	15180.22	15.42299	18.86866

Tablo 7.11. HPO-BBBC ve bileşenlerinin 30 boyutlu kalite testi fonksiyonlarındaki sonuçları

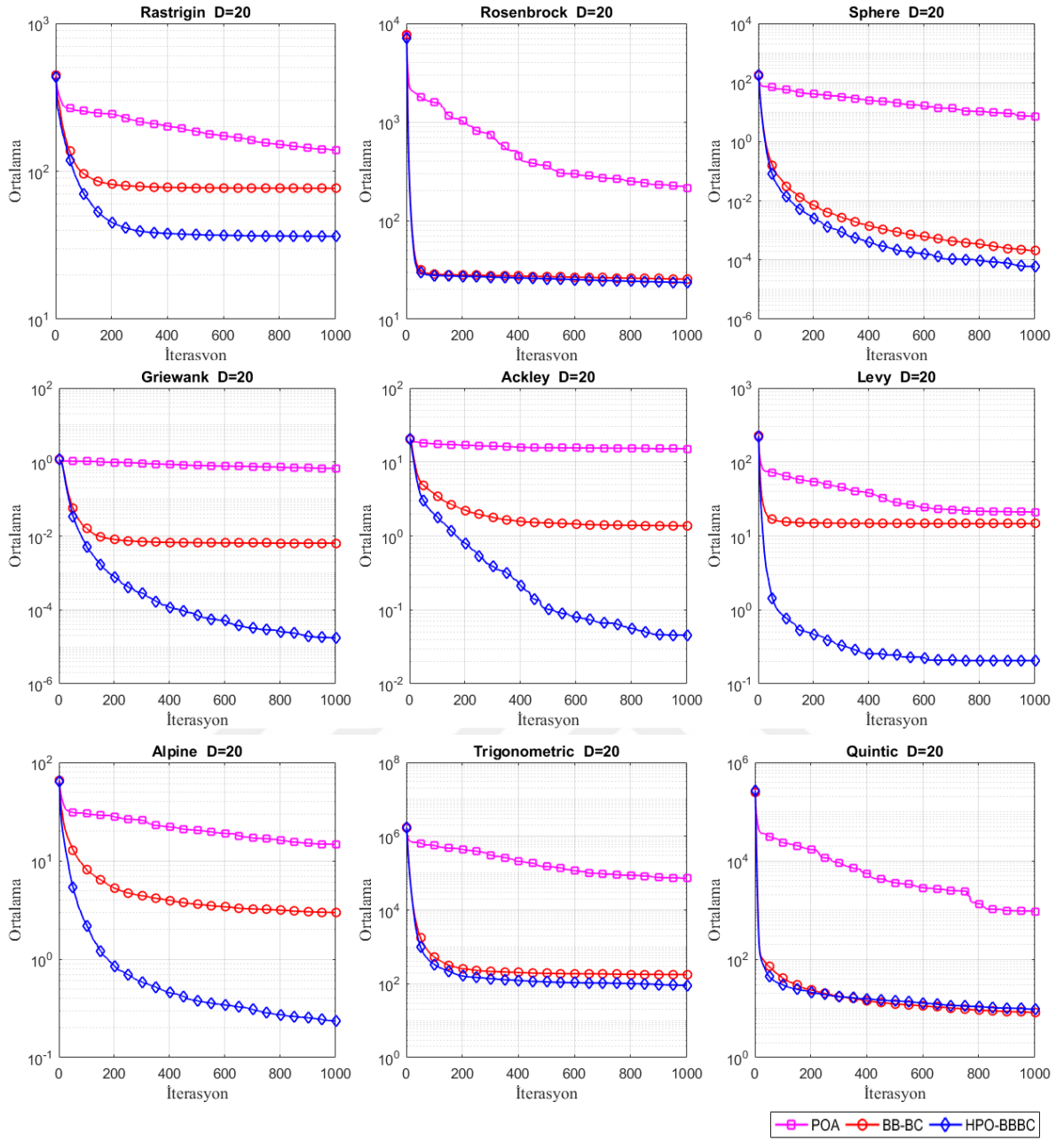
Fonksiyon (30-D)		POA	BBBC	HPO-BBBC
Rastrigin	Ortalama	139.3867	76.72499	36.31531
	Ortanca	129.1035	64.71984	35.27711
	En iyi	42.85194	38.83687	27.84697
	En Kötü	240.3742	146.0525	47.60840
	Std. Spm.	43.87944	27.89124	5.294134
Rosenbrock	Ortalama	215.0203	25.44456	23.51946
	Ortanca	205.4104	25.51786	23.52951
	En iyi	104.6751	21.39373	21.57485
	En Kötü	392.5203	28.25669	25.21088
	Std. Spm.	69.04831	1.160922	0.904639
Sphere	Ortalama	7.156554	0.000204	5.99E-05
	Ortanca	5.648231	0.000208	5.26E-05
	En iyi	1.973961	0.000135	1.25E-05
	En Kötü	23.89804	0.000253	0.000145
	Std. Spm.	5.201467	3.32E-05	3.28E-05
Griewank	Ortalama	0.667964	0.006368	1.75E-05
	Ortanca	0.665403	5.34E-05	1.59E-05
	En iyi	0.286765	3.41E-05	4.71E-06
	En Kötü	1.018586	0.027115	4.03E-05
	Std. Spm.	0.205145	0.008355	8.47E-06
Ackley	Ortalama	15.11142	1.370263	0.045324
	Ortanca	15.34156	0.078528	0.046053
	En iyi	10.91007	0.064685	0.018909
	En Kötü	17.50826	20.12731	0.073746
	Std. Spm.	1.765264	3.586247	0.013535
Levy	Ortalama	21.04214	14.69447	0.205904
	Ortanca	20.06519	14.18580	0.179129
	En iyi	10.77167	3.270748	4.68E-05
	En Kötü	56.30012	24.85243	0.716286
	Std. Spm.	8.402381	5.454371	0.189352
Alpine	Ortalama	14.54210	2.971544	0.234448
	Ortanca	13.50276	1.444623	0.175977
	En iyi	3.805212	0.163949	0.011440
	En Kötü	38.28231	23.39385	0.663158
	Std. Spm.	8.799228	4.612461	0.173764
Quintic	Ortalama	918.5836	8.236364	9.580975
	Ortanca	449.5083	5.463708	9.142110
	En iyi	72.93530	1.466310	2.781350
	En Kötü	6358.368	25.66050	20.74130
	Std. Spm.	1473.500	6.742632	4.186720
Trigonometric	Ortalama	72518.95	176.5215	90.48701
	Ortanca	61458.03	178.2274	87.78863
	En iyi	11158.34	141.5033	22.52228
	En Kötü	253017.3	259.1077	233.6894
	Std. Spm.	51795.64	24.44663	42.40636



Şekil 7.4. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonları için yakınsama grafikleri



Şekil 7.5. HPO-BBBC ve bileşenlerinin 20 boyutlu kalite testi fonksiyonları için yakınsama grafikleri



Şekil 7.6. HPO-BBBC ve bileşenlerinin 30 boyutlu kalite testi fonksiyonları için yakınsama grafikleri

Tablo 7.12. HPO-BBBC ve bileşenlerinin 10 boyutlu kalite testi fonksiyonlarındaki ortalama çalışma süreleri(sn)

Fonksiyon (10-D)		POA	BB-BC	HPO-BBBC
Rastrigin	Ortalama	2,950	0,464	5,808
	Std. Spm.	0,102	0,003	0,209
Rosenbrock	Ortalama	2,658	0,398	5,018
	Std. Spm.	0,136	0,004	0,252
Sphere	Ortalama	2,498	0,388	5,031
	Std. Spm.	0,269	0,004	0,228
Griewank	Ortalama	2,538	0,370	5,261
	Std. Spm.	0,192	0,003	0,281
Ackley	Ortalama	2,610	0,337	5,167
	Std. Spm.	0,165	0,007	0,320
Levy	Ortalama	2,975	0,541	5,964
	Std. Spm.	0,225	0,008	0,268
Alpine	Ortalama	2,564	0,321	5,006
	Std. Spm.	0,165	0,005	0,272
Trigonometric	Ortalama	2,639	0,372	4,797
	Std. Spm.	0,153	0,005	0,291
Quentic	Ortalama	2,814	0,492	6,013
	Std. Spm.	0,188	0,004	0,407

HPO-BBBC ve bileşenleri, 10 boyutlu test fonksiyonlarında uygunluk fonksiyonunun çağrılma sayısı sabit tutularak da test edilmiştir. Her bir algoritmanın bitim şartı için maksimum uygunluk fonksiyonu çağrılma sayısı 10000 olarak ayarlanmıştır ve her algoritma tüm test fonksiyonları için 30 defa çalıştırılmıştır. Elde edilen sonuçlar Tablo 7.13’te verilmiştir.

Tablo 7.13. HPO-BBBC ve bileşenlerinin, 10 boyutlu test fonksiyonları için uygunluk fonksiyonunun 10000 defa çağırılmasındaki sonuçları

Fonksiyon (10-D)		POA	BBBC	HPO-BBBC
Rastrigin	Ortalama	35,8719	24,44965	11,73754
	Ortanca	37,84433	22,91613	11,94884
	En iyi	8,168566	10,98292	6,106497
	En Kötü	67,94721	52,77481	18,88747
	Std. Spm.	14,32019	9,658333	2,494352
Rosenbrock	Ortalama	75,27478	5,108003	7,021882
	Ortanca	61,74713	4,93165	7,251298
	En iyi	8,879769	2,206236	5,126601
	En Kötü	219,1398	9,421416	8,311251
	Std. Spm.	59,5236	1,477121	0,92052
Sphere	Ortalama	3,923551	0,00021	0,001603
	Ortanca	2,484472	0,000208	0,001363
	En iyi	0,089094	8,82E-05	0,000315
	En Kötü	15,18165	0,000293	0,00417
	Std. Spm.	3,995139	5,5E-05	0,000918
Griewank	Ortalama	0,427036	0,111357	0,021053
	Ortanca	0,426181	0,09325	0,014686
	En iyi	0,016643	0,009912	0,000777
	En Kötü	0,91063	0,339065	0,103668
	Std. Spm.	0,225856	0,080217	0,025239
Ackley	Ortalama	10,20866	1,443947	0,772119
	Ortanca	10,56882	0,158888	0,747199
	En iyi	3,493673	0,077037	0,104341
	En Kötü	16,472	19,59413	1,502231
	Std. Spm.	3,353607	4,832463	0,319874
Levy	Ortalama	4,0714	3,086244	0,009482
	Ortanca	2,298429	2,442412	0,00591
	En iyi	0,268716	0,000212	0,001682
	En Kötü	14,27425	12,54765	0,096676
	Std. Spm.	3,894376	3,219723	0,016611
Alpine	Ortalama	3,263359	0,130226	0,064766
	Ortanca	3,454641	0,095543	0,020583
	En iyi	0,011217	0,008397	0,005669
	En Kötü	7,17803	0,880813	0,461161
	Std. Spm.	1,766439	0,164312	0,102053
Trigonometric	Ortalama	37230,68	35,92574	34,439
	Ortanca	19435,6	35,48647	33,44648
	En iyi	432,3724	24,29559	6,784009
	En Kötü	169961,4	50,80415	62,14533
	Std. Spm.	42804,45	6,087548	10,6135
Quintic	Ortalama	296,1242	1,42029	3,207765
	Ortanca	39,17091	0,94141	3,180694
	En iyi	2,355729	0,654164	1,475411
	En Kötü	2334,732	11,19003	6,745943
	Std. Spm.	498,8647	1,917597	1,196986

HPO-BBBC, POA ve BB-BC' nin sınıflandırma kural keşfindeki performansları Tablo 6.1' de özellikleri verilen veri setleri kullanılarak ölçülmüştür. Tüm algoritmaların başlangıç popülasyon sayıları 50 ve her kural keşfi için maksimum iterasyon sayısı 100 olarak ayarlanmıştır. Her algoritma her bir veri setinde 30 defa çalıştırılmıştır ve karşılaştırmalar ortalama doğru sınıflandırma değerlerine göre yapılmıştır. Deneysel çalışmada algoritmaların parametreleri ise; POA' da grup sayısı $M=5$, gruptaki üye sayısı $N=10$, grupların birleşme olasılığı $P_m=0.05$ ve grupların silinme olasılığı $P_d=0.001$ olarak seçilmiştir. BB-BC' de ise kütle merkezi x^c en iyi uygunluk değerine sahip olan parçacık olarak seçilmiştir. Önerilen HPO-BBBC algoritmasında ise POA' ya benzer şekilde $M=5$, $N=10$, $P_m=0.05$ ve $P_d=0.001$ olarak ayarlanmıştır. Genel üyeler x^c olarak seçilmiş ve x^c etrafında büyük patlama ile oluşturulacak birey sayısı $p=5$ olarak ayarlanmıştır. Algoritmaların sonlandırma şartı Banknote veri seti için; kural sayısının 10' a erişmesi veya keşfedilen kurallar tarafından kapsanmayan veri sayısının 430' un altına düşmesi, Phoneme veri seti için; kural sayısının 10' a erişmesi veya kurallar tarafından kapsanmayan veri sayısının 1150' nin altına düşmesi, Banana veri seti için; kural sayısının 10' a erişmesi veya kurallar tarafından kapsanmayan veri sayısının 4300' ün altına düşmesi olarak belirlenmiştir.

30 bağımsız çalıştırmada elde edilen karşılaştırmalı sonuçlar Tablo 7.14, Tablo 7.15 ve Tablo 7.16' da görülmektedir. Sonuçlar incelendiğinde tüm veri setlerinde HPO-BBBC algoritmasının, POA ve BB-BC' ye oranla gayet yüksek doğrulukta test verilerini sınıflandırdığı görülmüştür. POA ise tüm veri setlerinde ikinci en yüksek değerleri vermiştir. BB-BC' nin ise her üç veri setinde de HPO-BBBC ve POA' ya göre zayıf kaldığı görülmüştür.

Tablo 7.14. HPO-BBBC ve bileşenlerinden Banana veri setinde elde edilen sonuçlar

BANANA			
	HPO-BBBC	POA	BB-BC
1	467	384	381
2	465	439	416
3	457	402	425
4	464	410	418
5	466	406	359
6	471	415	421
7	465	440	355
8	457	438	409
9	455	391	379
10	458	402	375
11	455	441	403
12	465	445	363
13	466	452	405
14	458	404	415
15	468	440	393
16	453	443	381
17	467	445	385
18	458	416	415
19	457	404	401
20	466	417	405
21	467	430	386
22	467	428	395
23	458	437	406
24	466	414	381
25	468	465	401
26	467	414	393
27	468	414	357
28	469	409	403
29	465	417	408
30	452	414	418
Ort.	462.833	422.533	395.066
%	87.162	79.573	74.400
Süre (sn)	31.74	12.39	3.23

Tablo 7.15. HPO-BBBC ve bileşenlerinden Banknote veri setinde elde edilen sonuçlar

BANKNOTE			
	HPO-BBBC	POA	BB-BC
1	375	375	366
2	393	375	366
3	392	375	376
4	375	375	365
5	393	375	348
6	393	389	362
7	393	375	390
8	392	388	351
9	375	359	338
10	375	375	358
11	393	375	366
12	393	359	344
13	375	359	360
14	392	375	358
15	392	365	350
16	393	359	345
17	392	366	343
18	393	385	377
19	392	381	356
20	375	359	325
21	375	391	354
22	393	375	373
23	375	358	369
24	376	359	364
25	391	359	394
26	393	359	363
27	393	375	359
28	375	359	341
29	393	375	369
30	375	375	371
Ort.	386.166	370.966	360.033
%	93.957	90.259	87.599
Süre (sn)	3.03	1.45	0.68

Tablo 7.16. HPO-BBBC ve bileşenlerinden Phonome veri setinde elde edilen sonuçlar

PHONOME			
	HPO-BBBC	POA	BB-BC
1	1292	1277	1244
2	1340	1278	1266
3	1355	1286	1290
4	1268	1266	1260
5	1314	1285	1229
6	1295	1271	1234
7	1311	1264	1251
8	1339	1291	1230
9	1326	1251	1220
10	1324	1241	1239
11	1313	1262	1255
12	1322	1257	1251
13	1311	1282	1247
14	1326	1265	1278
15	1293	1279	1267
16	1288	1294	1256
17	1323	1276	1270
18	1311	1271	1255
19	1323	1273	1247
20	1328	1288	1223
21	1334	1279	1256
22	1287	1264	1238
23	1346	1301	1217
24	1309	1290	1213
25	1304	1291	1218
26	1332	1296	1228
27	1338	1251	1244
28	1291	1291	1257
29	1329	1281	1257
30	1350	1281	1227
Ort.	1317.4	1276.066	1245.566
%	81.320	78.769	76.886
Süre (sn)	12.26	5.94	2.60

8. SONUÇLAR VE ÖNERİLER

Optimizasyon, bir problemin alternatif çözümleri arasından en iyi olanı seçme işlemidir. Problemin doğrusal olmadığı ve birçok değişkene sahip olduğu durumlarda o problemi matematiksel olarak çözmek zorlaşır. Sezgisel yöntemler, karmaşık bir problem için makul sürede deneme yanılma yolu ile kabul edilebilir çözümler sunmayı amaçlamaktadır. Sezgisel yöntemler; en iyi çözümü bulacaklarını garanti etmezler fakat en kısa zamanda en makul çözümü ortaya koyarlar. Literatüre bakıldığında doğadan esinlenilerek geliştirilmiş birçok algoritma bulunmaktadır. Bütün bu algoritmaların temel amacı problemler için en iyi çözümü elde etmeye çalışmaktır. Fakat her problem için istenilen çözümleri elde edememektedirler. Bu nedenle, yeni yöntemlerin geliştirilmesine veya var olan yöntemlerin çözüm kalitesini arttırmaya ihtiyaç duyulmaktadır. Hibritleme, iki veya daha fazla algoritmanın özelliklerini tek bir algortmada birleştirerek algoritmanın çözüm kalitesini veya performansını geliştirmek amacıyla kullanılabilir.

Bu tez çalışmasında, birçok sezgisel algoritma incelenmiş ve fizik tabanlı hibrit bir yöntem geliştirilmesi amaçlanmıştır. Bu inceleme sürecinde iki farklı hibrit algoritma önerilmiştir. Bunlardan birincisi; EM' nin çözüm kalitesini ve performansını arttırmak amacıyla EM, PSO ve DE kullanılarak önerilen hibrit EM-PSO-DE algoritması, ikincisi ise; POA' nın çözüm kalitesini ve performansını geliştirmek amacıyla POA ve BB-BC' nin kombinasyonundan oluşan HPO-BBBC algoritmasıdır.

Önerilen hibrit algoritmaların performansı iki farklı uygulama ile analiz edilmiştir. Öncelikle önerilen hibrit yöntemler kalite testi fonksiyonları üzerinde çalıştırılarak küresel optimizasyon problemlerindeki başarıları test edilmiştir. İkinci olarak, önerilen hibrit yöntemler UCI ve KEEL veri ambarlarından alınan bazı sınıflandırma veri setleri üzerinde uygulanmış ve sınıflandırma kural keşfindeki performansları incelenmiştir.

Sonuç olarak, hibrit EM-PSO-DE algoritması, küresel optimizasyon problemlerinde EM' nin performansını hem çalışma süresi hem de küresel optimuma ulaşma açısından geliştirmiştir. Sınıflandırma kural keşfinde ise EM' ye göre biraz daha düşük doğrulukta sınıflandırma gerçekleştirmesine rağmen, EM' nin çalışma süresini önemli bir biçimde azaltmıştır. HPO-BBBC algoritması ise POA' nın çalışma süresini biraz uzatsa da çözüm

kalitesini hem küresel optimizasyon problemlerinde hem de sınıflandırma kural keşfinde önemli bir biçimde arttırmıştır. Tüm bu sonuçların geneline bakıldığında, önerilen hibrit algoritmaların literatürde alternatif birer yöntem olarak kullanılabileceği görülmektedir. İleriki çalışmalarda, önerilen hibrit algoritmaların kümeleme ve birliktelik kuralları gibi veri madenciliği modellerinde veya mühendislik optimizasyon problemleri gibi optimizasyon gerektiren alanlarda uygulamalarının gerçekleştirilebileceği düşünülmektedir.



KAYNAKLAR

- [1] **Mahmoodabadi, M. J. and Nemati, A. R.**, 2016. A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems, *Engineering Science and Technology, an International Journal*, **19(4)**, 2002-2021.
- [2] **Kızıloluk, S.**, 2013. Sosyal tabanlı sezgisel optimizasyon algoritmalarıyla sınıflandırma kurallarının keşfi, *Yüksek Lisans Tezi*, Munzur Üniversitesi Fen Bilimleri Enstitüsü, Tunceli.
- [3] **Lin, J. L., Tsai, Y. H., Yu, C. Y. and Li, M. S.**, 2012. Interaction enhanced imperialist competitive algorithms, *Algorithms*, **5(4)**, 433-448.
- [4] **Uymaz, S. A.**, 2015. Yeni bir biyolojik ilhamlı metasezgisel optimizasyon metodu: Yapay alg algoritması, *Doktora Tezi*, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- [5] **Yang, X. S.**, 2010. Nature-inspired metaheuristic algorithms, Luniver press.
- [6] **Kiziloluk, S. and Alatas, B.**, 2015. Automatic mining of numerical classification rules with parliamentary optimization algorithm, *Advances in Electrical and Computer Engineering*, **15 (4)**, 17-24.
- [7] **Kiziloluk, S. and Ozer, A. B.**, 2017. Web Pages Classification with Parliamentary Optimization Algorithm. *International Journal of Software Engineering and Knowledge Engineering*, **27(03)**, 499-513.
- [8] **Canayaz, M.**, 2015. Cırcır böceği algoritması: Yeni bir meta-sezgisel yaklaşım ve uygulamaları, *Doktora Tezi*, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- [9] **Atashpaz-Gargari, E. and Lucas, C.**, 2007. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, *2007 IEEE Congress on Evolutionary Computation*, Singapore, 4661-4667.
- [10] **Rao, R. V. and Patel, V.**, 2013. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, *Scientia Iranica*, **20(3)**, 710-720.
- [11] **Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S.**, 2009. GSA: a gravitational search algorithm, *Information sciences*, **179(13)**, 2232-2248.
- [12] **Eskandar, H., Sadollah, A., Bahreininejad, A. and Hamdi, M.**, 2012. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, **110**, 151-166.

- [13] **Holland, J.H.**, 1975. Adaptation in natural and artificial systems, Ann Arbor, MI: University of Michigan Press.
- [14] **Alatas, B.**, 2011. ACROA: artificial chemical reaction optimization algorithm for global optimization, *Expert Systems with Applications*, **38(10)**, 13170-13180.
- [15] **Geem, Z. W., Kim, J. H. and Loganathan, G. V.**, 2001. A new heuristic optimization algorithm: harmony search, *simulation*, **76(2)**, 60-68.
- [16] **Kashan, A. H.**, 2009. League championship algorithm: a new algorithm for numerical function optimization, *In Soft Computing and Pattern Recognition, SOCPAR'09, IEEE*, 43-48.
- [17] **Altunbey, F. ve Alataş, B.**, 2015. Sosyal Ağ Analizi İçin Sosyal Tabanlı Yapay Zekâ Optimizasyon Algoritmalarının İncelenmesi, *International Journal of Pure and Applied Sciences*, **1(1)**, 33-52.
- [18] **Kızıloluk, S. ve Ozer, A. B.**, 2016. Melez Elektromanyetizma Benzeri Parçacık Sürü Optimizasyon Algoritması, *Dicle Üniversitesi Mühendislik Dergisi*, **7(3)**, 515-526.
- [19] **Malek, M., Guruswamy, M., Owens, H. and Pandya, M.**, 1989. A hybrid algorithm technique, University of Texas at Austin, Department of Computer Sciences.
- [20] **Grosan, C. and Abraham, A.**, 2007. Hybrid evolutionary algorithms: methodologies, architectures, and reviews, *Hybrid evolutionary algorithms*, pp. 1-17, Springer Berlin Heidelberg.
- [21] **Calvet, L., Armas, J. D., Masip, D. and Juan, A. A.**, 2017. Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs, *Open Mathematics*, **15(1)**, 261-280.
- [22] **Kaveh, A., Bakhshpoori, T. and Afshari, E.**, 2014. An efficient hybrid particle swarm and swallow swarm optimization algorithm, *Computers & Structures*, **143**, 40-59.
- [23] **Kıran, M. S., Gündüz, M. and Baykan, Ö. K.**, 2012. A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum, *Applied Mathematics and Computation*, **219(4)**, 1515-1521.
- [24] **Wu, B., Qian, C., Ni, W. and Fan, S.**, 2012. Hybrid harmony search and artificial bee colony algorithm for global optimization problems, *Computers & Mathematics with Applications*, **64(8)**, 2621-2634.
- [25] **Li, Z., Wang, W., Yan, Y. and Li, Z.**, 2015. PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems, *Expert Systems with Applications*, **42(22)**, 8881-8895.

- [26] **Sun, Y., Zhang, L. and Gu, X.**, 2012. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems, *Neurocomputing*, **98**, 76-89.
- [27] **Ali, A. F. and Tawhid, M. A.**, 2017. A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems, *Ain Shams Engineering Journal*, **8(2)**, 191-206.
- [28] **Nabil, E.**, 2016. A modified flower pollination algorithm for global optimization. *Expert Systems with Applications*, **57**, 192-203.
- [29] **Jangir, P., Parmar, S. A., Trivedi, I. N. and Bhesdadiya, R. H.**, 2017. A novel hybrid Particle Swarm Optimizer with multi verse optimizer for global numerical optimization and optimal reactive power dispatch problem, *Engineering Science and Technology, an International Journal*, **20(2)**, 570-586.
- [30] **Zandi, Z., Afjei, E. and Sedighizadeh, M.**, 2013. Hybrid big bang--big crunch optimization based optimal reactive power dispatch for voltage stability enhancement, *Journal of Theoretical & Applied Information Technology*, **47(2)**, 537-546.
- [31] **Ahmadi, S. and Abdi, S.**, 2016. Application of the Hybrid Big Bang–Big Crunch algorithm for optimal sizing of a stand-alone hybrid PV/wind/battery system, *Solar Energy*, **134**, 366-374.
- [32] **Hora, C., Secui, D. C., Bendea, G. and Dzitac, S.**, 2016. BB-BC-CG Algorithm for Operational Reliability Modeling of Hydro Generator Groups, *Procedia Computer Science*, **91**, 1088-1095.
- [33] **Sedighizadeh, M., Mahmoodi, M. M. and Soltanian, M.**, 2014. Parameter identification of proton exchange membrane fuel cell using a Hybrid Big Bang-Big Crunch optimization, *Thermal Power Plants (CTPP), 2014 5th Conference, IEEE*, 35-39.
- [34] **Sedighizadeh, M. and Bakhtiary, R.**, 2016. Optimal multi-objective reconfiguration and capacitor placement of distribution systems with the Hybrid Big Bang–Big Crunch algorithm in the fuzzy framework, *Ain Shams Engineering Journal*, **7(1)**, 113-129.
- [35] **Sedighizadeh, M., Esmaili, M. and Eisapour-Moarref, A.**, 2017. Voltage and frequency regulation in autonomous microgrids using Hybrid Big Bang-Big Crunch algorithm, *Applied Soft Computing*, **52**, 176-189.
- [36] **Wang, Q., Zeng, J. and Song, W.**, 2010. A new electromagnetism-like algorithm with chaos optimization, *Computational Aspects of Social Networks (CASoN), 2010 International Conference*, 535-538, IEEE.

- [37] **Javadi, B., Jolai, F., Slomp, J., Rabbani, M. and Tavakkoli-Moghaddam, R.**, 2014. A hybrid electromagnetism-like algorithm for dynamic inter/intra-cell layout problem, *International Journal of Computer Integrated Manufacturing*, **27(6)**, 501-518.
- [38] **Chang, F. K. and Lee, C. H.**, 2008. Design of fractional PID control via hybrid of electromagnetism-like and genetic algorithms, *Intelligent Systems Design and Applications, 2008. ISDA'08, Eighth International Conference*, **2**, 525-530, IEEE.
- [39] **Jamili, A., Shafia, M. A. and Tavakkoli-Moghaddam, R.**, 2011. A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem, *Expert Systems with Applications*, **38(5)**, 5895-5901.
- [40] **Muhsen, D. H., Ghazali, A. B., Khatib, T. and Abed, I. A.**, 2015. Extraction of photovoltaic module model's parameters using an improved hybrid differential evolution/electromagnetism-like algorithm, *Solar Energy*, **119**, 286-297.
- [41] **Teimouri, M., Zareitalab, A., Niaki, S. T. A. and Sharifi, M.**, 2016. An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem, *Applied Soft Computing*, **38**, 423-436.
- [42] **Lee, C. H. and Lee, Y. C.**, 2012. Nonlinear systems design by a novel fuzzy neural system via hybridization of electromagnetism-like mechanism and particle swarm optimisation algorithms, *Information Sciences*, **186(1)**, 59-72.
- [43] **Sun, G., Zhang, A., Yao, Y. and Wang, Z.**, 2016. A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding, *Applied Soft Computing*, **46**, 703-730.
- [44] **Sarafrazi, S., Nezamabadi-pour, H. and Seydnejad, S. R.**, 2015. A novel hybrid algorithm of GSA with Kepler algorithm for numerical optimization, *Journal of King Saud University-Computer and Information Sciences*, **27(3)**, 288-296.
- [45] **Cura, T.**, 2008. Modern Sezgisel Teknikler ve Uygulamaları, Papatya Yayıncılık Eğitim.
- [46] **Karaboğa, D.**, 2011. Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayın Dağıtım.
- [47] **Kıran, M.S.**, 2014. Optimizasyon problemlerinin çözümü için yapay arı kolonisi algoritması tabanlı yeni yaklaşımlar, *Doktora Tezi*, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.

- [48] **Akay, B.**, 2009. Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi, *Doktora Tezi*, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- [49] **Alataş, B.**, 2007. Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, *Doktora Tezi*, Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- [50] **Başbuğ, S.**, 2013. Metasezgisel algoritmalar ile anten dizi sentezi, *Doktora Tezi*, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- [51] **Afşar, B.**, 2014. Metasezgisel algoritmalarda çevrim içi ayarlama ile uyarlanabilir parameter kontrolü, *Doktora Tezi*, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.
- [52] **Can, U. and Alatas, B.**, 2015. Physics based metaheuristic algorithms for global optimization, *American Journal of Information Science and Computer Engineering*, **1(3)**, 94-106.
- [53] **Siddique, N. and Adeli, H.**, 2016. Physics-based search and optimization: Inspirations from nature, *Expert Systems*, **33(6)**, 607-623.
- [54] **Xing, B. and Gao, W. J.**, 2014. Innovative computational intelligence: a rough guide to 134 clever algorithms, Switzerland: Springer International Publishing.
- [55] **Javidy, B., Hatamlou, A. and Mirjalili, S.**, 2015. Ions motion algorithm for solving optimization problems, *Applied Soft Computing*, **32**, 72-79.
- [56] **Turna, F.**, 2016. Yeni nesil metasezgisel yaklaşımların kümeleme performanslarının karşılaştırılması: resmi istatistiklerde uygulama, *Doktora Tezi*, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- [57] **Kiziloluk, S. and Ozer, A. B.**, 2016. Mining of classification rules with water cycle optimization algorithm, *International Conference on Computer Sciences and Engineering UBMK'16*, Tekirdağ, Turkey, 20-23 October.
- [58] **Hatamlou, A.**, 2013. Black hole: A new heuristic optimization approach for data clustering, *Information sciences*, **222**, 175-184.
- [59] **Kaveh, A. and Talatahari, S.**, 2010. A novel heuristic optimization method: charged system search, *Acta Mechanica*, **213(3)**, 267-289.
- [60] **Xie, L., Zeng, J. and Cui, Z.**, 2009. General framework of artificial physics optimization algorithm, *Nature & Biologically Inspired Computing, NaBIC 2009, World Congress on* pp. 1321-1326, IEEE.
- [61] **Xie, L. P. and Zeng, J. C.**, 2009. A global optimization based on physicomimetics framework, *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation* pp. 609-616, ACM.

- [62] **Sacco, W. F. and De Oliveira, C. R.**, 2005. A new stochastic optimization algorithm based on a particle collision metaheuristic, *Proceedings of 6th WCSMO*.
- [63] **Sacco, W. F. and Pereira, C. M.**, 2006. Two stochastic optimization algorithms applied to nuclear reactor core design, *Progress in Nuclear Energy*, **48(6)**, 525-539.
- [64] **Kripka, M. and Kripka, R. M. L.**, 2008. Big crunch optimization method, In *International conference on engineering optimization, Brazil* (pp. 1-5).
- [65] **Ting, T. O., Yang, X. S., Cheng, S. and Huang, K.**, 2015. Hybrid metaheuristic algorithms: Past, present, and future, In *Recent Advances in Swarm Intelligence and Evolutionary Computation* (pp. 71-83), Springer International Publishing.
- [66] **Özkan, Y.**, 2008. Veri Madenciliği Yöntemleri, Papatya Yayıncılık Eğitim.
- [67] **Uzun, Y.**, 2017. Sürekli optimizasyon problemleri için değişken komşuluk arama tekniklerinin geliştirilmesi, *Doktora Tezi*, Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- [68] **Han J. and Kamber M.**, 2001. Data mining: Concepts and techniques, Morgan Kaufmann Publishers, San Francisco.
- [69] **Birbil, Ş. İ. and Fang, S. C.**, 2003. An electromagnetism-like mechanism for global optimization, *Journal of global optimization*, **25(3)**, 263-282.
- [70] **Kennedy, J. and Eberhart, R.**, 1995. Particle swarm optimisation, In *Proceedings of the IEEE international conference on neural networks*, pp. 1942–1948, Washington DC: IEEE Computer Society.
- [71] **Özsağlam, M. Y. ve Çunkaş, M.**, 2008. Optimizasyon problemlerinin çözümü için parçacık sürü optimizasyonu algoritması, *Politeknik Dergisi*, **11(4)**, 299-305.
- [72] **Storn, R. and Price, K.**, 1995. Differential Evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, *International Computer Science Institute, Berkeley*.
- [73] **Storn, R. and Price, K.**, 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, **11(4)**, 341-359.
- [74] **Keskintürk, T.**, 2006. Diferansiyel gelişim algoritması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, **5(9)**, 85-99.
- [75] **Borji, A.**, 2007. A new global optimization algorithm inspired by parliamentary political competitions, *MICAI 2007: Advances in Artificial Intelligence*, 61-71.

- [76] **Borji, A. and Hamidi, M.**, 2009. A new approach to global optimization motivated by parliamentary political competitions, *International Journal of Innovative Computing, Information and Control*, **5**(6), 1643-1653.
- [77] **Kızıloluk, S., ve Alataş, B.**, 2012. Sosyal tabanlı güncel sezgisel optimizasyon algoritmaları, *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, **13**(2), 39-56.
- [78] **Erol, O. K. and Eksin, I.**, 2006. A new optimization method: big bang-big crunch, *Advances in Engineering Software*, **37**(2), 106-111.
- [79] **Lux, T.**, 2016. Convergence Rate Evaluation of Derivative-Free Optimization Techniques, In *International Workshop on Machine Learning, Optimization and Big Data* pp. 246-256, Springer, Cham.
- [80] **Jamil, M. and Yang, X. S.**, 2013. A literature survey of benchmark functions for global optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, **4**(2), 150-194.
- [81] http://infinity77.net/global_optimization/test_functions.html#test-functions-index
Benchmark Functions For Global Optimization Problems, 18 Ağustos 2017.
- [82] **Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L. and Herrera, F.**, 2011. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, *Journal of Multiple-Valued Logic & Soft Computing*, **17**.
- [83] **Lichman, M.**, 2013. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
- [84] **Alatas, B. and Akin, E.**, 2005. FCACO: fuzzy classification rules mining algorithm with ant colony optimization, *Advances in Natural Computation*, 787-797.

ÖZGEÇMİŞ

Soner KIZILOLUK

Munzur Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

E-Posta: sonerkiziloluk@gmail.com

- 1987** Sivas'ta doğdu
- 2002-2005** Sivas Lisesini tamamladı
- 2005-2009** Mersin Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu
- 2010-2013** Munzur Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği A.B.D' da Yüksek Lisans eğitimini tamamladı
- 2013-** Fırat Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yazılım A.B.D' da doktora eğitimine başladı ve halen devam etmektedir.
- 2009-** Munzur Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.