

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**KARASİNEK ALGORİTMASI: KÜRESEL OPTİMİZASYON AMAÇLI YENİ BİR
METASEZGİSEL YAKLAŞIMI**

Semih YÜKSEL

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

HAZİRAN 2023

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**KARASİNEK ALGORİTMASI: KÜRESEL OPTİMİZASYON AMAÇLI YENİ BİR
METASEZGİSEL YAKLAŞIMI**

Semih YÜKSEL

DANIŞMAN
Doç. Dr. Abdullah ELEN

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

HAZİRAN 2023

ONAY

Semih YÜKSEL tarafından hazırlanan “**Karasinek Algoritması: Küresel Optimizasyon Amaçlı Yeni Bir Metasezgisel Yaklaşımı**” adlı tez çalışması 23/06/2023 tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bandırma Onyedli Eylül Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Doç. Dr. Abdullah ELEN

(BANÜ)

(Danışman)

Doç. Dr. Rafet DURGUT

(BANÜ)

(Üye)

Dr. Öğr. Üyesi Emrullah SONUÇ

(KBÜ)

(Üye)

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../...gün ve .../... sayılı kararı ile onaylanmıştır.

Doç. Dr. Abdullah YEŞİL

Enstitü Müdürü

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS ÇALIŞMASI
ETİK BEYANI

Bandırma Onyedi Eylül Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre hazırlamış olduğum “**Karasinek Algoritması: Küresel Optimizasyon Amaçlı Yeni Bir Metasezgisel Yaklaşımı**” adlı tezimin özgün bir çalışma olduğunu, tez hazırlanırken tüm aşamalarda bilimsel etik ilkelerine uygun davrandığımı, tez kapsamında sunulan tüm verileri bilimsel etik ilkelerine uygun elde ettiğimi, tezde faydalandığım tüm eserlere atıf yaptığımı ve kaynaklar kısmında bu eserleri gösterdiğimi beyan ederim. 23/06/2023

Semih YÜKSEL
İmza

ÖNSÖZ

Günümüzde artan nüfus, buna paralel olarak artan enerji ihtiyacı, doğal kaynakların tükenmesi ve artan maliyetler nedeniyle her konuda verimli çalışmak önemli hale gelmiştir. Veriminin arttırılması konusunda yapılan çalışmalarda optimizasyon önemli bir çalışma alanıdır. Optimizasyon problemlerinde kısıtlı kaynaklarla istenilen bir işin optimum şekilde yapılması amaçlanmaktadır. Son yıllarda optimizasyon problemlerinin çözümünde metasezgisel yaklaşımlar geliştirilmiştir. Özellikle doğadan esinlenerek geliştirilen metasezgisel yaklaşımlar, gerçek dünya problemlerini ele alarak en ideal sonuçlar bulmayı amaçlamaktadır.

Bu tez çalışmasında optimizasyon problemlerini çözmek amacıyla yeni bir metasezgisel algoritma önerilmektedir. Önerdiğimiz “Karasinek Algoritması” klasik deneysel kıyaslama ve gerçek dünya mühendislik problemleri üzerinde test edilmiştir. Literatürde bilinen başarılı metasezgisel algoritmalar ile kıyaslandığında oldukça rekabetçi sonuçlar elde edilmiştir.

Bu tez çalışması sırasında, tez konusunun belirlenmesinden başlayarak son aşamaya kadar her konuda benden yardımlarını esirgemeyen danışman hocam Sayın Doç. Dr. Abdullah ELEN’e şükranlarımı sunarım.

Bu günlere gelmemde büyük emekleri olan annem Kadriye, babam Recep’e, varlıkları ile birer motivasyon kaynağı olan ablam Selin, yeğenim Yağız ve Eylül’e teşekkür ederim.

ÖZET

YÜKSEK LİSANS TEZİ

KARASİNEK ALGORİTMASI: KÜRESEL OPTİMİZASYON AMAÇLI YENİ BİR METASEZGİSEL YAKLAŞIMI

Semih YÜKSEL

**Bandırma Onyedli Eylül Üniversitesi
Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı**

**Danışman: Doç. Dr. Abdullah ELEN
Haziran 2023, 94 sayfa**

Optimizasyon problemlerini çözmek, mühendislik sistem tasarımının önemli bir parçasıdır. Günümüze kadar, mühendislik optimizasyon problemlerinin optimal çözümlerini aramak için çeşitli türde optimizasyon algoritmaları geliştirilmiştir. Algoritmaların rastgele özellikleri açısından optimizasyon yöntemleri, deterministik yöntemler ve modern metasezgisel algoritmalar olmak üzere iki kategoriye ayrılabilir. Genel olarak, metasezgisel algoritmaların çoğu doğadan ilham almaktadır.

Bu tez çalışmasında, optimizasyon problemlerinin çözülmesi amacıyla doğadan esinlenilerek yeni bir metasezgisel algoritma önerilmiştir. Karasinek Algoritması (KSA) olarak isimlendirdiğimiz yöntem, karasineklerin doğal davranışları göz önüne alınarak geliştirilmiştir. Önerilen algoritma, sentetik kıyaslama fonksiyonları ve gerçek dünya problemleri kullanılarak test edilmiştir. Sonuçlar, literatürde iyi bilinen metasezgisel yöntemler ile karşılaştırılmıştır. Deneysel çalışmalarda elde edilen sonuçlara göre önerilen yöntem hem sentetik kıyaslama fonksiyonlarında hem de gerçek dünya problemlerinde oldukça başarılı sonuçlar vermiştir.

Anahtar kelimeler: Optimizasyon, Metasezgiseller, Sürü Zekâsı, Karasinek Algoritması.

ABSTRACT

M.Sc. THESIS

HOUSEFLY ALGORITHM: A NOVEL METAHEURISTIC APPROACH FOR GLOBAL OPTIMIZATION

Semih YÜKSEL

**Bandırma Onyedi Eylül University
Graduate School of Natural and Applied Sciences
Department of Mechatronics Engineering**

Supervisor: Assoc. Prof. Dr. Abdullah ELEN

June 2023, 94 pages

Solving optimization problems is an important part of engineering system design. To date, various types of optimization algorithms have been developed to search for optimal solutions of engineering optimization problems. In terms of the random properties of algorithms, optimization methods can be divided into two categories, deterministic methods and modern metaheuristic algorithms. In general, most metaheuristic algorithms are inspired by nature.

In this thesis, a new metaheuristic algorithm, inspired by nature, is proposed to solve optimization problems. The method we call the Housefly Algorithm (KSA) was developed by considering the natural behavior of black flies. The proposed algorithm has been tested using synthetic comparison functions and real-world problems. The results were compared with metaheuristic methods well known in the literature. According to the results obtained in the experimental studies, the proposed method gave very successful results both in synthetic comparison functions and in real world problems.

Keywords: Optimization, Metaheuristics, Swarm Intelligence, Housefly Algorithm.

İÇİNDEKİLER

	Sayfa
ÖZET	I
ABSTRACT	II
İÇİNDEKİLER.....	III
ŞEKİL LİSTESİ	VII
TABLO LİSTESİ.....	IX
SİMGE VE KISALTMA LİSTESİ.....	X
1. GİRİŞ.....	1
2. LİTERATÜR İNCELEMESİ	3
3. METASEZGİSEL YÖNTEMLER	11
3.1 METASEZGİSELLERİN ÖZELLİKLERİ.....	11
3.1.1 Keşif ve Sömürü.....	12
3.1.2 Stokastik Arama	12
3.1.3 İteratif İyileştirme.....	12
3.1.4 Sağlamlık	13
3.1.5 Esneklik	13
3.1.6 Paralellik	13
3.2 METASEZGİSELLERİN SINIFLANDIRILMASI	14
3.2.1 Doğal Fenomenlerden Esinlenme.....	14
3.2.2 Popülasyon Tabanlı / Tek Çözüm	14
3.2.3 Deterministik veya Stokastik.....	14
3.2.4 Yerel veya Küresel Arama.....	15
3.3 METASEZGİSEL ALGORİTMA TASARIMI.....	15
3.3.1 Kodlama	15
3.3.2 Popülasyon	16

3.3.3	Operatörler.....	16
3.3.4	Algoritmik Tasarım.....	17
3.4	METASEZGİSELLERİN DEĞERLENDİRİLMESİ.....	18
3.4.1	Performans Ölçümü	18
3.4.2	DeneySEL Kıyaslama.....	18
3.4.3	İstatistiksel Testler.....	19
4.	MATERYAL VE METOT.....	21
4.1	POPÜLASYON TABANLI METASEZGİSELLER.....	21
4.1.1	Parçacık Sürü Optimizasyonu (PSO)	21
4.1.2	Ateşböceği Algoritması (AA)	22
4.1.3	Guguk Kuşu Arama Algoritması (GKAA)	24
4.1.4	İstilacı Yabani Ot Optimizasyon Algoritması (İYOOA)	25
4.2	KARASİNEK ALGORİTMASI.....	28
4.2.1	Yaşam Döngüsü ve Davranışları.....	28
4.2.1.1	Yumurta	29
4.2.1.2	Larva	29
4.2.1.3	Pupa.....	30
4.2.1.4	Yetişkinlik Dönemi.....	31
4.2.2	Önerilen Yöntem	32
4.3	DENEYLERDE KULLANILAN PROBLEMLER	36
4.3.1	Gerçek Dünya Problemleri.....	36
4.3.1.1	Dört Kademeli Dişli Kutusu Problemi	36
4.3.1.2	Hidrostatik Baskı Yatağı Tasarımı Problemi.....	40
4.3.1.3	Kaynaklı Kiriş Tasarımı.....	41
4.3.1.4	Endüstriyel Soğutma Sisteminin Optimal Tasarımı	43
4.3.2	Sentetik Kıyaslama Fonksiyonları	45
4.3.2.1	Complex Fonksiyonu	46

4.3.2.2	Matyas Fonksiyonu	46
4.3.2.3	BentCigar Fonksiyonu	47
4.3.2.4	Brown Fonksiyonu	48
4.3.2.5	ChungReynolds Fonksiyonu	49
4.3.2.6	Weierstrass Fonksiyonu.....	50
4.3.2.7	Quartic Fonksiyonu	51
4.3.2.8	Sphere Fonksiyonu.....	52
4.3.2.9	XinSheYang No.1 Fonksiyonu	53
4.3.2.10	XinSheYang No.2Fonksiyonu	54
4.3.2.11	XinSheYang No.3Fonksiyonu	55
4.3.2.12	Bukin No.4Fonksiyonu	56
4.3.2.13	CamelThreeHump Fonksiyonu.....	57
4.3.2.14	DropWave Fonksiyonu	58
4.3.2.15	Alpine No.1 Fonksiyonu.....	59
4.3.2.16	Csendes Fonksiyonu.....	60
4.3.2.17	Griewank Fonksiyonu.....	61
4.3.2.18	Salomon Fonksiyonu	62
4.3.2.19	WWavy Fonksiyonu.....	63
4.3.2.20	XinSheYang No.4Fonksiyonu	64
5.	DENEYSEL ÇALIŞMALAR	66
5.1	PERFORMANS ÖLÇÜM KRİTERLERİ	66
5.2	SENTETİK KIYASLAMA FONKSİYON TESTLERİ.....	66
5.2.1	Deney 1: Tek Modlu Kıyaslama Fonksiyonları	66
5.2.2	Deney 2: Çok Modlu Kıyaslama Fonksiyonları.....	71
5.3	GERÇEK DÜNYA PROBLEMLERİNİN TESTLERİ	76
5.3.1	Deney No. 1.....	76
5.3.2	Deney No. 2.....	78

5.3.3 Deney No. 3.....	80
6. TARTIřMA VE SONUÇ	83
7. KAYNAKLAR	84
8. EKLER	91
ÖZGEÇMİř	94



ŞEKİL LİSTESİ

<u>Sekil</u>	<u>Sayfa</u>
Şekil 1. Parçacık Sürü Optimizasyon Algoritmasının Akış Diyagramı [68].	22
Şekil 2. Ateş Böceği Algoritmasının Akış Diyagramı [70].	23
Şekil 3. Guguk Kuşu Arama Algoritmasının Akış Diyagramı [72].	25
Şekil 4. İstilacı Yabani Ot Optimizasyon Algoritmasının Akış Diyagramı [74].	27
Şekil 5. Yetişkin bir karasinek (<i>Musca domestica</i>) ve yumurtaları.	29
Şekil 6. Karasinek larvaları [77].	30
Şekil 7. Karasineklerin pupa dönemleri.	30
Şekil 8. Yetişkin Bir Karasinek.	31
Şekil 9. Karasinek Algoritmasının Akış Diyagramı.	35
Şekil 10. Dört Kademeli Dişli Kutusu Problemi [82].	37
Şekil 11. Hidrostatik Baskı Yatağı Tasarım Problemi [84].	40
Şekil 12. Kaynaklı Kiriş Tasarımı [86].	42
Şekil 13. Endüstriyel Soğutma Sisteminin Optimal Tasarımı [88].	43
Şekil 14. Complex Fonksiyonunun Karakteristik Grafiği.	46
Şekil 15. Matyas Fonksiyonunun Karakteristik Grafiği.	47
Şekil 16. BentCigar Fonksiyonunun Karakteristik Grafiği.	48
Şekil 17. Brown Fonksiyonunun Karakteristik Grafiği.	49
Şekil 18. ChungReynolds Fonksiyonunun Karakteristik Grafiği.	50
Şekil 19. Weierstrass Fonksiyonunun Karakteristik Grafiği.	51
Şekil 20. Quartic Fonksiyonunun Karakteristik Grafiği.	52
Şekil 21. Sphere Fonksiyonunun Karakteristik Grafiği.	53
Şekil 22. XinSheYang No. 1 Fonksiyonunun Karakteristik Grafiği.	54
Şekil 23. XinSheYang No. 2 Fonksiyonunun Karakteristik Grafiği.	55
Şekil 24. XinSheYang No. 3 Fonksiyonunun Karakteristik Grafiği.	56
Şekil 25. Bukin No. 4 Fonksiyonunun Karakteristik Grafiği.	57
Şekil 26. CamelThreeHump Fonksiyonunun Karakteristik Grafiği.	58
Şekil 27. DropWave Fonksiyonunun Karakteristik Grafiği.	59
Şekil 28. Alpine No. 1 Fonksiyonunun Karakteristik Grafiği.	60
Şekil 29. Csendes Fonksiyonunun Karakteristik Grafiği.	61
Şekil 30. Griewank Fonksiyonunun Karakteristik Grafiği.	62

Şekil 31. Salomon Fonksiyonunun Karakteristik Grafiği.	63
Şekil 32. WWay Fonksiyonunun Karakteristik Grafiği.	64
Şekil 33. XinSheYang No. 4 Fonksiyonunun Karakteristik Grafiği.	65
Şekil 34. Deney 1'e göre tek modlu fonksiyonlardaki yakınsama grafikleri.	68
Şekil 35. KSA'nın tek modlu fonksiyonlardaki arama geçmişi.	70
Şekil 36. Deney 2'ye göre çok modlu fonksiyonlardaki yakınsama grafikleri.	72
Şekil 37. KSA'nın çok modlu fonksiyonlardaki arama geçmişi.	75
Şekil 38. Deney No. 1'e Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.	78
Şekil 39. Deney No. 2'ye Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.	80
Şekil 40. Deney No. 3'e Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.	82



TABLO LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 1. Önerilen Yöntemin Sabitleri	35
Tablo 2. Sentetik Kıyaslama Fonksiyonları.	45
Tablo 3. Tek modlu sentetik kıyaslama fonksiyonları.	67
Tablo 4. Global-best skorları.	67
Tablo 5. Deney 1'e Göre Algoritmaların Çalışma Süreleri (sn).	69
Tablo 6. Wilcoxon işaretli sıralar testi (p -değerleri).	69
Tablo 7. Çok modlu sentetik kıyaslama fonksiyonları.	71
Tablo 8. Global-best skorları.	71
Tablo 9. Deney 2'ye Göre Algoritmaların Çalışma Süreleri (sn).	73
Tablo 10. Wilcoxon işaretli sıralar testi (p -değerleri).	73
Tablo 11. Gerçek Dünya Problemlerinin fonksiyonları.	76
Tablo 12. Deney No. 1'e Göre Global En İyi Skorlar.	76
Tablo 13. Deney No. 1'e Göre Algoritmaların Çalışma Süreleri (sn).	77
Tablo 14. Deney No. 1'e Göre Wilcoxon işaretli sıralar testi (p -değerleri).	77
Tablo 15. Deney No. 2'ye Göre Global En İyi Skorlar.	79
Tablo 16. Deney No. 2'ye Göre Algoritmaların Çalışma Süreleri (sn).	79
Tablo 17. Deney No. 2'ye Göre Wilcoxon işaretli sıralar testi (p -değerleri).	79
Tablo 18. Deney No. 3'e Göre Global En İyi Skorlar.	81
Tablo 19. Deney No. 3'e Göre Algoritmaların Çalışma Süreleri (sn).	81
Tablo 20. Deney No. 3'e Göre Wilcoxon işaretli sıralar testi (p -değerleri).	81

SİMGE VE KISALTMA LİSTESİ

Simgeler

Açıklama

σ	: Standart Sapma
$H(u)$	: Heaviside fonksiyonu
v	: Hız
x	: Konum

Kisaltmalar

Açıklama

AA	: Ateşböceği Algoritması (Firefly Algorithm)
ABC	: Yapay Arı Kolonisi Optimizasyonu (Artificial Bee Colony Optimization)
ACO	: Karınca Koloni Optimizasyonu (Ant Colony Optimization)
AISA	: Ergen Kimlik Arama Algoritması (Adolescent Identity Search Algorithm)
AO	: Aquila Optimizasyon Algoritması (Aquila Optimizer)
AOS	: Atomik Yörünge Arama (Atomic Orbital Search)
BBO	: Biyocoğrafya Tabanlı Optimizasyon (Biogeography-based Optimization)
BWO	: Karadul Optimizasyonu (Black Widow Optimization Algorithm)
CapSA	: Kapuçin Arama Algoritması (The Capuchin Search Algorithm)
CEC	: Evrimsel Hesaplama Kongresi (Congress on Evolutionary Computation)
CGO	: Kaos Oyunu Optimizasyonu (Chaos Game Optimization)
CNN	: Evrişimsel Sinir Ağı (Convolutional Neural Network)
CPU	: Merkezi İşlem Birimi (Central Process Unit)
CSA	: İş birliği Arama Algoritması (Cooperation Search Algorithm)
DA	: Yusufçuk Algoritması (Dragonfly Algorithm)
DE	: Diferansiyel Evrim (Differential Evolution)
DSA	: Yunus Sürüsü Algoritması (Dolphin Swarm Algorithm)
FHO	: Ateş Şahini Optimizasyon Algoritması (Fire Hawk Optimizer)
FPA	: Çiçek Tozlaşma Algoritması (Flower-Pollinating Algorithm)
GA	: Genetik Algoritma (Genetic Algorithm)
GBO	: Gradyan Tabanlı Optimizasyon (Gradient-based Optimizer)
GKAA	: Guguk Kuşu Arama Algoritması (Cuckoo Search)
GOA	: Çekirge Optimizasyon Algoritması (Grasshopper Optimisation Algorithm)

HS	: Armoni Arama Algoritması (Harmony Search)
İYOOA	: İstilacı Yabani Ot Optimizasyonu (Invasive Weed Optimization)
JS	: Denizanası Arama Algoritması (Jellyfish Search)
KSA	: Karasinek Algoritması (Housefly Algorithm)
Npdf	: Normal Olasılık Yoğunluk Fonksiyonu
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
TGA	: Ağaç Büyüme Algoritması (Tree Growth Algorithm)
SA	: Benzetimli Tavlama (Simulated Annealing)
WCA	: Su Döngüsü Algoritması (Water Cycle Algorithm)
WOA	: Balina Optimizasyon Algoritması (Whale Optimization Algorithm)



1. GİRİŞ

Optimizasyon bir eniyileme işlemidir. Optimizasyon, matematiksel bir fonksiyonun belirli kısıtlar altında alabileceği en düşük ya da en yüksek değerleri bulmak için değişkenlerin alabileceği en uygun değerleri bulmayı amaçlamaktadır. Optimizasyon istenilen hedefe ulaşmak için daha verimli sonuçlar elde ettiğinden teoride ve pratikte çalışma alanı gün geçtikçe artmaktadır. Gerçek hayatta ekonomi, mühendislik, ulaşım ve benzer alanlarda optimizasyon problemleri karşımıza çıkmaktadır. Genel olarak optimizasyon problemleri, kısıtlı veya kısıtsız, kesikli veya sürekli, statik veya dinamik, tek veya çok amaçlı olmak üzere çeşitli kategorilere ayrılmaktadır [1].

Çoğu gerçek dünya optimizasyon problemi, yüksek karmaşıklığa, doğrusal olmayan kısıtlamalara, değişkenler arasında karşılıklı bağımlılığa ve geniş bir çözüm yelpazesine sahiptir [2]. Bu nedenden dolayı böyle karmaşık optimizasyon problemlerini çözüme ulaştırmak zor ve maliyetlidir. Bu tür problemleri çözmek için yapılan çalışmalarda, matematiksel modeller oluşturularak programlama yaklaşımları kullanılmış ve metasezgisel algoritma olarak isimlendirilen algoritmalar geliştirilmiştir.

Metasezgisel yaklaşımlarda, algoritma çözüm uzayında en uygun sonuca ulaşmak için kullandığı iki yöntemden yararlanır. Bu yöntemler keşif ve sömürüdür. Keşif aşamasında algoritma tüm çözüm uzayında arama yaparak sonucu çeşitlendirmeyi amaçlar. Sömürü aşamasında, istenilen sonuca yakın olduğu düşünülen bölgelerde daha ayrıntı bir şekilde arama yapılarak o bölgedeki en uygun sonucun bulunması amaçlanır. Algoritmanın verimli çalışabilmesi için bu iki yöntemin dengelenmesi önemlidir [3]. Bu nedenden dolayı algoritma geliştirilirken bu dengeyi kurmaya dikkat edilmelidir. Denge sağlanamadığında çeşitli sorunlar oluşmakta ve optimum sonuç elde edilememektedir.

Son yirmi yılda metasezgisel yöntemlere artan bir ilgi gözlemlenmiştir. Bu alanda yapılan çalışmalar arttıkça birçok yeni algoritma geliştirilmiştir. Metasezgisel algoritmaların büyük bir kısmı doğadan esinlenerek tasarlanmış ve geliştirilmiştir [4]. Metasezgisel algoritmalar tasarlanırken yararlanılan ilham kaynaklarına göre kategorilersek bu şekilde bir sınıflandırma yapabiliriz;

- a) Genetik evrim tabanlı algoritmalar. Bu tür algoritmalar evrimden esinlenerek üreme, çaprazlama, mutasyon ve benzer operatörler kullanarak oluşturulmuştur. Genetic Algorithm (GA) [5, 6] ve Differential Evolution (DE) [7] bu türün en popüler iki algoritmasıdır.

- b) Hayvan tabanlı algoritmalar. Bu algoritmalar, hayvanların beslenme, avlanma, hayatta kalma, göç etme, birlikte yaşama gibi davranışlarından ilham alınarak oluşturulmuştur. Bazı önemli örnekler olarak Particle Swarm Optimization (PSO) [8], Artificial Bee Colony (ABC) [9], Biogeography-Based Optimization (BBO) [10] ve Ant Colony Optimization (ACO) [11] verilebilir.
- c) Bitki tabanlı algoritmalar. Bu algoritmalar, bitkilerin davranışını taklit ederek oluşturulmuştur. Örneğin, Invasive Weed Optimization (İYOOA) algoritması [12] yabancı ot istilası sürecini örnek alarak oluşturulmuştur. Flower-Pollinating Algorithm (FPA) [13] çiçek tozlaşma özelliklerinden ilham alarak oluşturulmuştur.
- d) İnsan faaliyetine dayalı algoritmalar. Bu algoritmalar insan faaliyetleri örnek alınarak oluşturulmuştur. Örneğin, Harmony Search (HS) algoritması [14], mükemmel bir uyum hali aramanın müzikal sürecini örnek almıştır. Teaching-Learning Based Optimization (TLBO) algoritması [15], öğretmen öğretme ve öğrenci öğrenme sürecinden ilham alınarak oluşturulmuştur. Passing Vehicle Search (PVS) [16], iki şeritli otoyollarda sürüş temel alınarak geliştirilmiştir.
- e) Diğer kaynaklardan ilham alan algoritmalar. Fizik tabanlı algoritmalar (örneğin, Water Cycle Algorithm (WCA) [17]) ve matematik tabanlı algoritmalar (örneğin, Stochastic Fractal Search (SFS) algoritması [18]) oluşan algoritmalar.

2. LİTERATÜR İNCELEMESİ

IEEE CEC, dünyanın dört bir yanından evrimsel hesaplama ve hesaplama zekâ alanındaki araştırmacıları ve uygulayıcıları bir araya getiren birinci sınıf bir konferanstır. Konferans programında genel kurul konuşmaları, dersler, düzenli ve özel oturumlar, eğitimler, yarışmalar, atölye çalışmaları ve panel tartışmaları yer alır. IEEE Congress on Evolutionary Computation (CEC) test fonksiyonları, literatürde algoritmaların kıyaslanmasında sıklıkla kullanılan test takımı olarak bilinen bazı sayısal optimizasyon problemlerinin çözümünde kullanılmıştır. Uluslararası bir konferans olan bu etkinlikte yapılan çalışmalar literatürde test için sıklıkla kullanılmaktadır.

Chou ve Truong [19] yaptıkları çalışmada, denizanasının okyanustaki davranışından esinlenerek Jellyfish Search (JS) olarak isimlendirilen yeni bir metasezgisel algoritma geliştirmişlerdir. Önerilen JS algoritmasını, kapsamlı bir dizi matematiksel kıyaslama fonksiyonu ve bir dizi yapısal mühendislik problemi üzerinde test etmişlerdir. Deneysel çalışmalarda, iyi bilinen on metasezgisel algoritma ile karşılaştıklarını ve elde edilen sonuçlara göre önerilen yöntemin matematiksel kıyaslama fonksiyonlarını çözmeye daha iyi performans gösterdiğini bildirmişlerdir. Bogar ve Beyhan [20] yaptıkları çalışmada, ergenlerin kimlik geliştirme/arama sürecini ilham alan, Adolescent Identity Search Algorithm (AISA) isimli yeni bir popülasyon tabanlı metasezgisel optimizasyon algoritması önermişlerdir. Önerdikleri algoritmayı, keşif, kullanım, yerel optimum kaçınma ve yakınsama özelliklerini test etmek için otuz dokuz iyi bilinen tek modlu, çok modlu, sabit boyutlu çok modlu, bileşik ve CEC 2019 kıyaslama işlevi üzerinde değerlendirmişler ve sonuçları, on üç son teknoloji meta-sezgisel algoritma ile kapsamlı bir karşılaştırmalı çalışma ile doğrulamışlardır. Ayrıca AISA, gerçek hayat mühendislik uygulamaları olarak kabul edilen yedi Serbestlik Dereceli (7-DOF) bir robot manipülatörün sistem tanımlama ve ters kinematik problemini çözmek için kullanmışlardır. Genel optimizasyon sonuçları, AISA'nın çeşitli karmaşık sayısal optimizasyon problemlerini çözmeye diğer rakip metasezgisel algoritmalara göre üstün performans üretmek için güçlü ve sağlam bir yeteneğe sahip olduğunu bildirmişlerdir. Ali ve ark. [21] yaptıkları çalışmada, algılayıcı düğümlerin iletişim enerjisi tüketimini azaltmak için Novel Ranked-Based Clustering (NRC) adı verilen bir sezgisel yöntemle entegre edilmiş yeni bir ARSH-FATI-based Cluster Head Selection (ARSH-FATI-CHS) algoritması geliştirmişlerdir. Önerilen ARSH-FATI-CHS algoritmasını simüle etmişler ve kablolu algılayıcı düğümlerin, enerji tüketimi açısından performansını belirlemek için çeşitli sonuçlar üretmişlerdir. Sonuçları en gelişmiş Particle Swarm Optimization (PSO) ile karşılaştırmışlar ve ARSH-FATI-CHS algoritmasının, algılayıcı düğümlerin enerji tüketimini

~%25 oranında iyileştirdiğini kanıtladığını bildirmişlerdir. Zhang ve ark. [22] yaptıkları çalışmada, kuyruktaki insan faaliyetlerinden ilham alan, Queuing Search (QS) adı verilen yeni bir metasezgisel algoritma geliştirmişlerdir. Mevcut metasezgisel algoritmaların çoğundan farklı olarak QS, tüm metasezgisel algoritmalarda iki ortak parametre olan popülasyon boyutu ve durdurma kriteri dışında diğer parametreleri önceden ayarlamasına gerek olmadığından, yeni optimizasyon problemlerine kolayca uygulanabildiği bildirilmiştir. QS algoritmasının performansı, CEC 2014'ten 30 ve 100 boyutlu, 5 standart ve 4 zorlayıcı kısıtlı mühendislik optimizasyon problemi ile 30 sınır kısıtlanmalı kıyaslama işlevi optimizasyon probleminde çözerek kontrol etmişlerdir. Bazı son teknoloji veya iyi bilinen meta-sezgisel algoritmalarla karşılaştırdıktan sonra, deney sonuçları, QS algoritmasının performansının başarılı olduğunu bildirmişlerdir. Ghasemian ve ark. [23] yaptıkları çalışmada, kentleşme ve yaşam durumlarını iyileştirmek için insan davranışından ilham alan Human Urbanization Search Algorithm (HUS) adı verilen yeni bir metasezgisel algoritma önerilmişlerdir. HUS algoritmasının, yasakları kaldırma, iyileştirme ve yok etme konusunda insan davranışlarından esinlendiği belirtilmiştir. Önerilen HUS algoritmasını, tip bakımından zengin (tek modlu, serbest boyutlu çok modlu, sabit boyutlu çok modlu) 20 işlevlik bir set üzerinde test etmişlerdir. Sonuçları Imperialist Competitive Algorithm (ICA), Particle Swarm Optimization (PSO) ve Gravitational Search Algorithm (GSA) ile karşılaştırmışlardır. Sonuçların, HUS'un diğer algoritmalarla karşılaştırıldığında oldukça rekabetçi olduğunu gösterdiğini belirtmişlerdir. Tek modlu fonksiyonlar için, HUS çoğu durumda diğer algoritmalarından daha iyi performans gösterdiğini ve çoğu multimodal fonksiyon için daha iyi veya eşit performans sergilediğini bildirmişlerdir. Zhang ve Jin [24] yaptıkları çalışmada, ekstra kontrol parametrelerinin nasıl ayarlanacağı zorluğunun üstesinden gelmek için Group Teaching Optimization Algorithm (GTOA) adı verilen yeni bir metasezgisel algoritma sunmuşlardır. Sunulan GTOA'da, grup öğretim mekanizmasından esinlenmişlerdir. Grup öğretimini genelliği kaybetmeden bir optimizasyon tekniği olarak kullanıma uygun hale getirmek için önce dört basit kural tanımlamışlardır. Daha sonra öğretmen atama aşaması, yetenek gruplama aşaması, öğretmen aşaması ve öğrenci aşamasından oluşan dört kuralın rehberliğinde bir grup öğretim modeli oluşturmuşlardır. GTOA algoritmasını, ilk olarak 28 adet iyi bilinen kısıtlanmamış kıyaslama problemi üzerinde incelemişler ve optimizasyon sonuçlarını son teknoloji dokuz algoritma ile karşılaştırmışlardır. Deneysel sonuçların, önerilen algoritmayı bu problemler için çözüm kalitesi, yakınsama hızı ve kararlılık açısından üstün performans göstermekte olduğunu belirtmişlerdir. Ayrıca algoritmayı, gerçek dünyada dört kısıtlı mühendislik tasarım optimizasyon problemini çözmek için kullanmışlardır. Simülasyon sonuçlarında, önerilen GTOA'nın, bildirilen optimize

edicilere kıyasla daha yüksek hızda daha iyi çözümler bulabileceğini göstermişlerdir. Azizi ve ark. [25] yaptıkları çalışmada, ısılkçı çaylakların, kara çaylakların ve kahverengi şahinlerin yiyecek arama davranışına dayanan ve Fire Hawk Optimizer'ı (FHO) olarak adlandırılan yeni bir metasezgisel algoritma önermişlerdir. Önerdikleri algoritmayı kullanarak, boyutları 2–100 arasında değişen 233 matematiksel test fonksiyonu üzerinde sayısal bir inceleme yapmışlar ve optimizasyon amacıyla 150.000 fonksiyon değerlendirmesi gerçekleştirmişlerdir. Karşılaştırma için, alternatif yaklaşımlar olarak toplam on farklı klasik ve yeni metasezgisel algoritma kullanmışlardır. Elde ettikleri sonuçların, FHO algoritmasının literatürdeki karşılaştırdıkları algoritmalarından daha iyi performans gösterdiğini belirtmişlerdir. Talatahari ve Azizi [26] yaptıkları çalışmada, optimizasyon problemlerini çözmek için kaos teorisinin bazı ilkelerine dayanan ve Chaos Game Optimization (CGO) olarak adlandırılan yeni bir metasezgisel algoritma geliştirmişlerdir. CGO algoritmasının sonuçlarını değerlendirmek için farklı özelliklere sahip üç karşılaştırmalı analiz yapmışlardır. İlk adımda literatürden altı farklı metasezgisel algoritma seçilerek CGO için fonksiyon değerlendirme sayısı ile birlikte minimum, ortalama ve standart sapma değerleri hesaplamış ve bu algoritmalar karşılaştırmışlardır. İkincisinde, CGO algoritmasının sonuçları, yakın zamanda geliştirilen bazı fraktal ve kaos tabanlı algoritmalarla karşılaştırmışlardır. Son olarak, CGO algoritmasının performansı, en son teknolojiye sahip bazı algoritmalarla ve “CEC-2017” adlı tek amaçlı gerçek parametrelili sayısal optimizasyon yöntemlerinden biriyle karşılaştırmışlardır. Ek olarak, sunulan algoritma için hesaplamalı bir maliyet analizi de yapmışlardır. Elde edilen sonuçların, çoğu durumda CGO'nun diğer metasezgisel yöntemlere göre üstün olduğunu kanıtladığını bildirmişlerdir. Azizi [27] çalışmasında, kuantum mekaniğinin bazı ilkelerine ve çekirdek etrafındaki elektronların genel konfigürasyonunun perspektifte olduğu kuantum tabanlı atom modeline dayanan, Atomic Orbital Search (AOS) olarak adlandırılan yeni bir metasezgisel algoritma önermiştir. Bu algoritmanın performansını değerlendirmek için, 2–100 farklı boyutlarda toplam 20 kısıtsız matematiksel test fonksiyonu kullanmış, istatistiksel amaçlar için 100 bağımsız optimizasyon çalıştırması ile maksimum 150.000 fonksiyon değerlendirmesi dikkate almıştır. Kolmogorov Smirnov, Wilcoxon ve Kruskal Wallis testlerinden yararlanılarak bir istatistiksel analiz yapmış, karşılaştırma amacıyla alternatif olarak 8 metasezgisel yöntem kullanmıştır. Ek olarak, en son Evrimsel Hesaplama Yarışmalarının “CEC-2020” kısıtlı optimizasyon problemlerinden bazıları da dahil olmak üzere toplam 5 kısıtlı mühendislik tasarım problemi tasarım örnekleri olarak kullanmıştır. AOS algoritmasının kısıtlama problemleriyle başa çıkmada elde ettiği sonuçları, literatürdeki farklı standart, geliştirilmiş ve hibrit metasezgisel algoritmaların sonuçlarıyla karşılaştırmıştır. Elde

edilen sonuçların, önerilen AOS algoritmasının matematiksel ve mühendislik tasarım problemlerinin çözümünde çok başarılı sonuçlar verdiğini bildirmiştir. Benyamin ve ark. [28] yaptıkları çalışmada, vahşi dağ ceylanlarının sosyal yaşamı ve hiyerarşisinden esinlenen yeni bir meta-sezgisel algoritma olan Mountain Gazelle Optimizer (MGO) önermişlerdir. Bu algorithmada ceylanların hiyerarşik ve sosyal yaşamları matematiksel olarak formüle etmişler ve bir optimizasyon algoritması geliştirmek için kullanmışlardır. MGO algoritmasını, elli iki standart kıyaslama işlevi ve yedi farklı mühendislik problemi kullanılarak değerlendirmiş ve test etmişlerdir. Sonucu doğrulamak için diğer dokuz güçlü meta-sezgisel algoritma ile karşılaştırmışlardır. Karşılaştırmalı algoritmalar arasındaki önemli farkları, Wilcoxon's rank-sum ve Friedman's testlerini kullanarak göstermişlerdir. Çok sayıda deneyde, MGO'nun en yüksek kıyaslama işlevlerinde karşılaştırılabilir algoritmalarından daha iyi performans gösterdiğini belirtmişlerdir. Ayrıca yaptıkları testlere göre MGO'nun arama yeteneklerini korumuş olduğunu ve optimizasyon problemlerinin boyutlarını büyütürken bile iyi performans göstermiş olduğunu bildirmişlerdir. Cheraghalipour ve ark. [29] yaptıkları çalışmada, ağaçların ışık ve yiyecek elde etme rekabetinden esinlenerek optimizasyon amaçlı Tree Growth Algorithm (TGA), yaklaşımını önermişlerdir. Bu yeni yaklaşımını, iyi bilinen optimizasyon algoritmaları ile karşılaştırıp, TGA'nın bu problemlerde üstün olduğunu belirtmişlerdir. Ayrıca, TGA'nın etkinliğini ve sağlamlığını doğrulamak için bazı parametrik olmayan tekniklerle yakınsama analizi ve anlamlılık testleri kullanmışlardır. Yapılan testlerin sonuçlarına göre TGA'nın başarılı bir metasezgisel ve optimizasyon problemleri için uygun bir yaklaşım olduğunu bildirmişlerdir. Xie ve ark. [30] yaptıkları çalışmada, Tuna Swarm Optimization (TSO) olarak adlandırılan yeni bir sürü tabanlı metasezgisel algoritma önerilmişlerdir. TSO'nun performansını, bir dizi kıyaslama işlevi ve birkaç gerçek mühendislik problemi üzerinde diğer metasezgisel yöntemlerle karşılaştırarak değerlendirmişlerdir. Simülasyon sonuçlarına göre TSO'nun diğer karşılaştırmalı algoritmalara kıyasla daha iyi performans gösterdiğini bildirmişlerdir. Abualigah ve ark. [31] yaptıkları çalışmada, Aquila'nın avını yakalama sürecinde doğadaki davranışlarından ilham alan Aquila Optimizer (AO) adlı yeni bir popülasyon tabanlı optimizasyon yöntemi önermişlerdir. AO yönteminin farklı optimizasyon sorunları için en uygun çözümü bulma yeteneğini doğrulamak için bir dizi deneysel seri yürütmüşlerdir. İlk deney sırasında, iyi bilinen 23 fonksiyonun çözümünü bulmak için AO algoritmasını uygulamışlardır. İkinci ve üçüncü deney serisinde, AO'nun sırasıyla 30 CEC-2017 test fonksiyonu ve 10 CEC-2019 test fonksiyonu gibi daha karmaşık problemlere çözüm bulma performansını değerlendirmişlerdir. Son olarak, yedi gerçek dünya mühendislik problemi seti kullanmışlardır. AO'nun bilinen meta-sezgisel yöntemlerle karşılaştırıldığında elde

edilen deneysel sonuçlarından, geliştirilen AO algoritmasının üstünlüğünü bildirmişlerdir. Feng ve ark. [32] yaptıkları çalışmada, karmaşık küresel optimizasyon problemini ele almak için Called Cooperation Search Algorithm (CSA) adı verilen yeni bir popülasyona dayalı evrimsel yöntem önermişlerdir. CSA yönteminin yakınsama hızını ve arama doğruluğunu test etmek için üç tür matematiksel optimizasyon problemi (24 ünlü test fonksiyonu, 25 CEC-2005 test problemi ve 30 CEC-2014 test problemi dahil) kullanmışlardır. Ayrıca, CSA yönteminin mühendislik uygulanabilirliğini kanıtlamak için birkaç ünlü mühendislik optimizasyon problemi (Dişli tren tasarımı, Kaynaklı giriş tasarımı ve Hız düşürücü tasarımı gibi) kullanarak testler gerçekleştirmişlerdir. Farklı senaryolardaki sonuçları, mevcut birkaç evrimsel algoritma ile karşılaştırdıklarında, CSA yönteminin karar alanını etkili bir şekilde keşfedebildiğini ve çeşitli performans değerlendirme göstergeleri açısından rekabetçi sonuçlar üretebildiğini bildirmişlerdir. Rashid [33] yaptığı çalışmada, Tiki-Taka Algorithm (TTA) adı verilen bir futbol oynama stiline dayanan, spordan ilham alan yeni bir algoritma önermiştir. Algoritmayı, 19 kıyaslama işlevi ve beş mühendislik tasarım problemini kullanarak test etmiştir. Önerilen algoritmanın performansı, spor tabanlı, yüksek oranda alıntı yapılan ve yeni algoritmalarından alınan diğer 11 meta-sezgisel ile karşılaştırmıştır. Sonuçlara göre TTA'nın son derece rekabetçi olduğunu, kıyaslama problemlerinin %84'ünde birinci ve ikinci sırada yer aldığını belirtmiştir. Önerilen algoritmanın iki mühendislik tasarım probleminde en iyi performansı gösterdiğini ve kalan üç problemde ikinci sırada yer aldığını bildirmiştir. Hashim ve Hussien [34] yaptıkları çalışmada, yılanların özel çiftleşme davranışını taklit eden çeşitli optimizasyon görevlerinin üstesinden gelmek için Snake Optimizer (SO) adlı yeni bir doğadan ilham alan metasezgisel algoritma önermişlerdir. Önerilen yöntemin geçerliliğini ve üstünlüğünü doğrulamak için SO algoritmasını, 29 kısıtlanmamış Evrimsel Hesaplama Kongresi kıyaslama işlevi ve dört kısıtlı gerçek dünya mühendislik problemi üzerinde test etmişlerdir. Deneysel sonuçlar ve istatistiksel karşılaştırmaların, SO'nun farklı arazilerde keşif-sömürü dengesi ve yakınsama eğrisi hızı açısından etkin ve verimli olduğunu bildirmişlerdir. Hayyolalam ve Pourhaji [35] yaptıkları çalışmada, sürekli doğrusal olmayan optimizasyon problemleri için uygun yeni bir meta-sezgisel algoritma önermişlerdir. Önerdikleri yöntem, Black Widow Optimization Algorithm (BWO), kara dul örümceklerinin benzersiz çiftleşme davranışından esinlenmişlerdir. BWO algoritmasının problemler için en uygun çözümleri elde etmedeki etkinliğini doğrulamak için, 51 farklı kıyaslama işlevi üzerinde değerlendirmişlerdir. Elde edilen sonuçlarda, önerilen algoritmanın diğer algoritmalara göre erken yakınsama ve optimize edilmiş uygunluk değeri elde etme gibi farklı açılardan birçok avantajı olduğunu göstermişlerdir. Gerçek vaka çalışması problemlerinin sonuçlarında, önerilen

algoritmanın bilinmeyen ve zorlu uzaylarla ilgili gerçek dünya sorunlarını çözmede etkin olduğunu bildirmişlerdir. Nematollahi ve ark. [36] yaptıkları çalışmada, The Golden Ratio Optimization Method (GROM) olarak bilinen yeni bir parametresiz meta-sezgisel optimizasyon algoritması önermişlerdir. Önerilen algorithmada, ünlü matematikçi Fibonacci tarafından formüle edilen bitki ve hayvan büyümesinin altın oranından esinlenmişlerdir. Önerilen yöntemi değerlendirmek için 29 iyi bilinen kıyaslama testi fonksiyonu ve ayrıca 4 makine mühendisliği problemi ve 1 elektrik mühendisliği problemi olmak üzere 5 klasik mühendislik optimizasyon problemi kullanılmışlardır. Çeşitli test fonksiyonları kullanarak, önerilen yöntemin ayrık, sürekli, yüksek boyutlu ve yüksek kısıtlı problemler dahil olmak üzere farklı problemlerin çözümündeki performansını test etmişlerdir. Önerilen yöntemin sonuçlarını, 11 adet son teknoloji optimizasyon algoritmasının sonuçlarıyla karşılaştırmışlardır. Karşılaştırmaları, elde edilen nihai yanıt, yakınsama hızı ve davranışı ve CPU zaman tüketimi gibi farklı yönlerden yapmışlardır. Amaçlanan yöntemin farklı açılardan üstünlüğünü, karşılaştırmalarla elde ettiklerini bildirmişlerdir. Rahmani ve Aliabdi [37] yaptıkları çalışmada, bitki rekabet süreçlerinden ilham alan yeni bir evrimsel algoritma, Plant Competition Optimization (PCO) algoritmasını önermişlerdir. Algoritmanın verimliliğini, iyi bilinen optimizasyon problemleri üzerinde sayısal testler yaparak ve sonuçları diğer evrimsel algoritmalarla karşılaştırarak göstermişlerdir. Sonuçların, PCO'nun optimum altı çözümler bulmada etkili ve verimli olduğunu ve genetik algoritma (GA), parçacık sürü optimizasyonu (PSO), simulated annealing (SA), grasshopper optimisation algorithm (GOA) ve dragonfly algorithm (DA) yöntemlerinden daha iyi sonuçlar verdiğini bildirmişlerdir. Braik ve ark. [38] yaptıkları çalışmada, kısıtlı ve küresel optimizasyon problemlerini çözmek için The Capuchin Search Algorithm (CapSA) adı verilen doğadan ilham alan yeni bir arama optimizasyonu algoritması önermişlerdir. CapSA'nın ana ilham kaynağını, kapuçin maymunlarının dinamik davranışları olarak oluşturmuşlardır. Önerilen algoritmayı, 23 iyi bilinen karşılaştırmalı değerlendirme işlevinde ve aynı zamanda çeşitli zorlu ve hesaplama açısından maliyetli mühendislik problemlerini çözmede karşılaştırmışlardır. Optimizasyon hassasiyeti ve istatistiksel test analizi açısından CapSA'nın birkaç önde gelen meta-sezgisel algoritma üzerindeki etkinliğini göstermek için geniş bir karşılaştırmalı çalışma yapmışlardır. Genel sonuçların, CapSA'nın rekabetçi meta-sezgisel yöntemlere kıyasla yüksek bir yakınsama oranıyla daha kesin çözümler sunduğunu belirtmişlerdir. Oyelade ve ark. [39] yaptıkları çalışmada, Ebola virüsü hastalığının yayılma mekanizmasına dayanan Ebola Optimization Search Algorithm (EOSA) adlı yeni bir metasezgisel algoritma sunmuşlardır. Önerilen yöntemin performansını ve kabiliyetini diğer optimizasyon yöntemleriyle karşılaştırmalı olarak değerlendirmek için, kırk yedi

klasik ve otuz kısıtlı IEEE-CEC kıyaslama işlevinden oluşan iki kıyaslama işlevi seti incelemiştir. Sonuçların, önerilen algoritmanın performansının ölçeklenebilirlik, yakınsama ve duyarlılık analizlerine dayalı diğer en gelişmiş optimizasyon yöntemleriyle rekabet edebildiğini belirtmiştir. Ayrıca algoritmayı, dijital mamografinin görüntü sınıflandırmasında convolutional neural network (CNN) hiper parametrelerinin en iyi kombinasyonunu seçme karmaşık problemini çözmek için uygulamıştır. Elde edilen sonuçların, optimize edilmiş CNN mimarisinin meme kanserini dijital görüntülerden %96,0 doğrulukla başarılı saptadığını bildirmiştir. Shadravan ve ark. [40] yaptıkları çalışmada, bir grup avlanan yelken balığından ilham alan, SailFish Optimizer (SFO) adlı, doğadan ilham alan yeni bir metasezgisel optimizasyon algoritması sunmuşlardır. SFO algoritmasını, algoritmanın farklı özelliklerini test etmek için iyi bilinen 20 tek modlu ve çok modlu matematiksel fonksiyon üzerinde değerlendirmiştir. Ayrıca SFO algoritmasını, düşük ve yüksek boyutlarda altı adet son teknoloji metasezgisel algoritma ile karşılaştırmıştır. SFO algoritmasının, karşılaştırmaların çoğunda literatürdeki en iyi algoritmalarından daha iyi performans gösterdiğini bildirmiştir. Ahmadianfar ve ark. [41] yaptıkları çalışmada, yeni bir metasezgisel optimizasyon algoritması olan gradient-based optimizer (GBO) önermişlerdir. Yeni algoritmanın performansını iki aşamada değerlendirmiştir. GBO'nun çeşitli özelliklerini değerlendirmek için önce 28 matematiksel test fonksiyonu kullanmışlar ve daha sonra GBO tarafından altı mühendislik problemini optimize etmişlerdir. İlk aşamada, GBO'nun gelişmiş keşif, kullanım, yakınsama ve yerel optimumlardan etkili bir şekilde kaçınma yetenekleri nedeniyle çok umut verici sonuçlar verdiğini belirtmiştir. İkinci aşamada, karmaşık gerçek dünya mühendislik problemlerini çözmede GBO'nun üstün performansı gösterdiğini bildirmiştir. Agushaka ve ark. [42] yaptıkları çalışmada, ceylanların yırtıcıların egemen olduğu ortamlarda hayatta kalma yeteneklerinden esinlenen Gazelle Optimization Algorithm (GOA) adlı yeni bir popülasyon tabanlı metasezgisel algoritma önermişlerdir. Geliştirilen algoritmanın bir optimizasyon aracı olarak sağlamlığı ve etkinliği, kıyaslama optimizasyon test fonksiyonları ve seçilmiş mühendislik tasarım problemlerini (onbeş klasik, on bileşik fonksiyon ve dört makine mühendisliği tasarım problemi) kullanarak test etmişlerdir. GOA'nın sonuçlarını, diğer dokuz son teknoloji algoritmayla karşılaştırmıştır. Elde edilen simülasyon sonuçlarının, GOA algoritmasının literatürde bulunan dokuz son teknoloji algoritmadan üstün olduğunu bildirmiştir. Qiao ve Yang [43] yaptıkları çalışmada, Quantum Dolphin Swarm Algorithm (QDSA) önermişlerdir. Son zamanlarda önerilen Dolphin Swarm Algorithm (DSA) gibi, meta-sezgisel algoritmaların yerel minimum sorununu çözmek amacıyla bu çalışmada, kuantum arama algoritmasını DSA'ya tanıtmıştır. Önerilen algoritmanın performansını test etmek için, yaygın olarak kullanılan altı büyük ölçekli fonksiyonu

(örn. Döndürölmüş hiper-elipsoid fonksiyon) örneğ olarak almışlardır. Ayrıca, karşılaştırma için bazı gelişmiş algoritmalar (örn. Whale Optimization Algorithm (WOA)) kullanmışlardır. Sonuçlar, QDSA'nın çözüm elde etme yeteneğinin, DSA ile karşılaştırıldığında açıkça iyileştirildiğini ve QDSA'nın performansının, karşılaştırma için dikkate alınan diğer algoritmalarından üstün olduğunu bildirmişlerdir.



3. METASEZGİSEL YÖNTEMLER

Son yirmi yılda, farklı problemlere çok sayıda doğal fenomenlerden esinlenen yaklaşımlar uygulandı ve bunlardan bazıları olağanüstü performans gösterdi. Bu artan ilgi, genellikle daha önce literatürde bildirilen çeşitli yöntemlerin uyarlanması, geliştirilmesi ve analizine odaklanan ilgili yayınlanmış materyalin miktarında olağanüstü bir artışa yol açmıştır [44].

Metasezgiseller, genellikle doğadan ilham alan, karmaşık optimizasyon problemlerini çözmek için tasarlanmış genel algoritmik çerçevelerdir ve birkaç on yıldan beri büyüyen bir araştırma alanıdır. Son yıllarda meta-sezgisel yöntemler, matematiksel formülasyonlarında belirsiz, stokastik ve dinamik bilgileri içeren optimizasyon problemlerini çözmek için daha klasik yaklaşımlara başarılı alternatifler olarak ortaya çıkmaktadır [45].

Metasezgisellik, uzun hesaplama süreleri veya eksik problem tanımları nedeniyle matematiksel optimizasyonun mümkün olmadığı karmaşık optimizasyon problemlerinin çözümünde önemli bir rol oynar [46]. Daha verimli arama alanı keşfetme yetenekleri sağlamak için daha yüksek düzeyde bir çerçevede düzenlenmiş sezgisel yöntemlerin bir kombinasyonunu kullanırlar ve bir algoritma yerine bir tasarım modeli olarak tanımlanabilirler [47].

Doğadan ilham alan meta-sezgisel algoritmaların uygulanmasına yönelik araştırmalar, son kırk yılda giderek daha fazla popülerlik kazanmıştır. Bununla birlikte, bu algoritmaların elde ettiği başarılarla güçlü bir şekilde atfedilen bu popüleritenin birçok nedeni vardır. Genel olarak, önde gelen nedenlerden birinin, bu algoritmaların doğadaki en ilginç ve başarılı fenomenlerden bazılarını taklit ederek ilham alınmış ve tasarlanmış olması olduğunu söylemek delildir. Metasezgisel algoritmaların neredeyse %99'unun geliştirilmesine ilham vermiş olan bu doğal oluşumlu süreçler basitçe ikiye ayrılabilir: biyolojik oluşum süreci ve kimya ve fizik yasalarını veya sistemlerini içeren fiziksel yasalar veya sistemler [48].

3.1 METASEZGİSELLERİN ÖZELLİKLERİ

Matematiksel modellere ve varsayımlara dayanan geleneksel optimizasyon tekniklerinden farklı olarak, metasezgisel yöntemler problemden bağımsızdır ve farklı türde problemlere en az değişiklikle uyum sağlayabilir [49]. Metasezgisel algoritmaların temel özelliklerinden bazıları aşağıda maddeler halinde verilmiştir.

3.1.1 Keşif ve Sömürü

Genel olarak metasezgisel algoritmalar ve özel olarak sürü zekâsında kullanılan iki temel arama davranışı, keşif ve sömürüdür. Keşif, uygulanabilir bölgenin keşfedilmemiş alanının aranması anlamına gelirken, sömürü, gelecek vaat eden bir bölgedeki alanın aranması anlamına gelir. Bu algoritmaların başarısı, büyük ölçüde, bu iki arama davranışının nasıl dengelendiğine bağlıdır [50]. Algoritma ıraksadığında, sürü bireyleri içindeki D boyutunun değerleri arasındaki fark büyür, bu da sürü bireylerinin arama ortamında dağıldığı anlamına gelir. Buna metasezgisel araştırmalarda keşif veya çeşitlendirme denir. Öte yandan, sürü yakınsadığında, konumsal fark en aza iner ve sürü bireyleri yoğun bir alanda toplanır. Buna sömürü veya yoğunlaştırma denir. Yinelemeler sırasında, farklı metasezgisel algoritmalar, sürü bireyleri içinde keşif ve sömürüyü zorlamak için farklı stratejiler kullanır. Bu iki kavram, herhangi bir metasezgisel algoritmada bulunur [51].

3.1.2 Stokastik Arama

Metasezgiseller, yeni çözümler üretmek ve yerel optimuma takılmaktan kaçınmak için rastgeleleştirmeyi kullanır. Çözümlerin çeşitliliği genellikle rastgelelik yoluyla gerçekleştirilir. Rastgeleleştirme, yerel aramadan küresel ölçekte aramaya geçmek için iyi bir yol sağlar. Böylece daha geniş bir arama uzayında arama yapılarak keşif operatörü kullanılmış olur [52].

3.1.3 İteratif İyileştirme

Metasezgisel yöntemler, mevcut çözümlere tekrar tekrar değişiklikler uygulayarak çözümlerin kalitesini yinelemeli olarak geliştirir. Metasezgisel algoritmalar, başlangıç koşullarıyla rastgele olarak oluşturulmuş ilk çözümden başlar ve mevcut çözümün yineleme süresince iyileştirmeler yaparak daha iyi bir çözüm bulmaya çalışır. Yinelemeli iyileştirme olarak bilinen özellik ile algoritma, iyileştirici bir çözüm için keşif yapar. Böyle bir çözüm bulunursa, mevcut çözümün yerine geçer ve sömürü ile daha iyi bir çözüm olup olmadığını aramaya devam eder. Bu adımlar, mevcut çözümün komşuluğunda iyileşen komşu çözüm bulunamayana ve algoritma yerel bir optimumda sona erene kadar tekrarlanır [53].

3.1.4 Sağlamlık

Metasezgisel yöntemler, belirli problem özelliklerine ve varsayımlara bağımlı olmaktan kaçınarak gürültülü ve belirsiz problem durumlarını ele almak için tasarlanmıştır. Algoritmik sağlamlık, bir algoritmanın, değişken uzaydan ve ilk başlangıç noktasından bağımsız olarak sürekli olarak düşük maliyetli tasarımlara yakınsama yeteneği olarak tanımlanır. Büyük değişken uzayları ve yönetici metrik eksikliği, belirli bir metasezgisel algoritmanın iki ayrı zamanda çalıştırıldığında çok farklı iki tasarıma yakınsayabileceği anlamına gelir. Buna ek olarak, daha küçük değişken uzaylarında kullanıldığında sağlam görünebilen algoritmalar, değişken uzayı arttıkça sağlamlığını hızla kaybedebilir. Bu nedenle, küçük problemler üzerinde algoritmalar çalışmak, özellikle gerçekçi boyutlara sahip büyük değişken uzayları göz önüne alındığında, algoritmik performansı tam olarak keşfetmek için yeterli olmayabilir. Bu nedenle, tasarım optimizasyonunu uygulamacı mühendisler için daha faydalı bir yöne taşımak için metasezgisel algoritmaların algoritmik sağlamlığı araştırılmalıdır [54].

3.1.5 Esneklik

Metasezgisel yöntemler, seçim kriterleri, komşuluk yapıları ve arama stratejileri değiştirilerek özelleştirilebilir ve farklı problem alanlarına uyarlanabilir. Daha güçlü araçlar oluşturmak için diğer buluşsal veya kesin algoritmalarla melezleştirilebilirler [55]. Metasezgiselliğin çok çeşitli problemlere uyarlanabilir olması sebebiyle, farklı disiplinlerdeki bilimsel çalışma alanlarında da popüler bir araçtır. Bu yönüyle, algoritmanın problemden bağımsız olarak çözüm üretebilmesi en önemli özelliğidir [56].

3.1.6 Paralellik

Arama sürecini hızlandırmak ve büyük ölçekli sorunları verimli bir şekilde çözmek için metasezgisel yöntemler kolayca paralelleştirilebilir. Literatürde, yörünge algoritmalarına dayalı üç model vardır: komşuluğun paralel keşfi ve değerlendirmesi, paralel çoklu başlangıç ve tek çözümün paralel değerlendirilmesi. Öte yandan, popülasyon tabanlı algoritmalar da iki yaklaşımla paralel hesaplama uygulanmıştır: hesaplamalı paralelleştirme ve popülasyon paralelleştirme. Önceki yaklaşımda, bireyler üzerindeki işlemler paralel olarak yapılırken, sonraki yaklaşım, popülasyonu paralel olarak yürütülecek alt popülasyonlara ayırır [57].

3.2 METASEZGİSELLERİN SINIFLANDIRILMASI

Metasezgisel algoritmalar, farklı kriterlere göre çeşitli şekillerde sınıflandırılabilir. Bu bölüm başlığı altında, esin kaynaklarına, popülasyon durumuna, deterministik/stokastik, global ve yerel arama durumlarına göre sınıflandırılmıştır.

3.2.1 Doğal Fenomenlerden Esinlenme

Metasezgisel yöntemler, doğal süreçlerden ilham alıp almadıklarına göre sınıflandırılabilir. Doğadan ilham alan metasezgisel yöntemler, genetik algoritmalar [58, 59], parçacık sürüsü optimizasyonu ve karınca kolonisi optimizasyonu gibi algoritmaları içerir. Doğadan ilham almayan meta-sezgisel yöntemler, benzetilmiş tavlama ve tabu arama gibi algoritmaları içerir [49].

3.2.2 Popülasyon Tabanlı / Tek Çözüm

Metasezgisel yöntemler, tek bir çözüm üzerinde mi yoksa bir çözüm popülasyonu üzerinde mi çalıştıklarına göre de sınıflandırılabilir. Tek çözümlü meta-sezgisel yöntemler, benzetilmiş tavlama ve tepeye tırmanma gibi algoritmaları içerir. Popülasyon tabanlı meta-sezgisel yöntemler, genetik algoritmalar, parçacık sürüsü optimizasyonu ve karınca kolonisi optimizasyonu gibi algoritmaları içerir.

Metasezgisel yöntemler, tek bir çözüm mü yoksa çoklu çözümler mi bulmaya odaklandıklarına göre de sınıflandırılabilir. Yörünge tabanlı meta-sezgisel yöntemler, tek bir çözüm bulmaya odaklanır ve bu çözümü iyileştirmek için yinelemeli iyileştirme kullanır. Örnekler arasında tepe tırmanışı ve simüle edilmiş tavlama yer alır. Popülasyona dayalı meta-sezgisel yöntemler, çoklu çözümler bulmaya odaklanır ve arama uzayını keşfetmek için bir çözüm popülasyonu kullanır. Örnekler arasında genetik algoritmalar ve parçacık sürüsü optimizasyonu yer alır [60].

3.2.3 Deterministik veya Stokastik

Metasezgisel yöntemler, deterministik veya stokastik süreçleri kullanmalarına göre sınıflandırılabilir. Deterministik metasezgisel yeni çözümler üretmek için deterministik süreçleri

kullanır. Örnekler arasında tepe tırmanışı ve deterministik tavlama yer alır. Stokastik metasezgisel yeni çözümler üretmek için stokastik süreçleri kullanır. Örnekler arasında genetik algoritmalar ve benzetimli tavlama yer alır [49].

3.2.4 Yerel veya Küresel Arama

Yerel arama tabanlı ve küresel arama tabanlı: Metasezgisel yöntemler, yerel arama alanını mı yoksa küresel arama alanını mı keşfetmeye odaklandıklarına göre sınıflandırılabilir. Yerel arama tabanlı meta-sezgisel yöntemler, mevcut çözümün hemen yakınında çözümler bulmaya odaklanır. Örnekler tepeye tırmanma ve tabu aramayı içerir. Küresel arama tabanlı meta-sezgisel yöntemler, tüm arama alanını keşfetmeye odaklanır. Örnekler arasında genetik algoritmalar ve karınca kolonisi optimizasyonu yer alır [61].

3.3 METASEZGİSEL ALGORİTMA TASARIMI

3.3.1 Kodlama

Kodlama, algoritmik tasarımda alınması gereken ilk önemli karardır. Aday çözümleri temsil edecek kodlama türüne karar verilmelidir (gerçek veya ayrık; ikili, permütasyon, rasgele anahtarlar vb.). Uzunluğu da (çözümü oluşturan parametrelerin sayısı olarak anlaşılır) da önemli bir seçimdir. Bu uzunluk, problemin boyutuna veya optimize edilecek parametre sayısına bağlı olabilir. Dolayısıyla bu seçimlere bağlı olarak kodlanan çözümler farklı anlamlar kazanabilmektedir. Örneğin aday, TSP için permütasyon kodlamasında olduğu gibi problem çözümünün kendisini (genotip = fenotip olduğunda) veya Karınca Kolonisi Optimizasyonu kullanılırken normalde olduğu gibi kısmi bir çözümü temsil edebilir. Bununla birlikte, aday, belirli bir sistem için girdi olarak işlev gören bir dizi değeri veya daha sonra tam problem çözümünde bir rol oynayacak olan tanımlanmış bir tercihler dizisi konfigürasyonunu temsil edebilir. Bu özelliği hesaba katarak, yalnızca kodlamayı problemle eşleştirmek (genotipe karşı fenotip) değil, aynı zamanda açıkça detaylandırmak da önemlidir.

3.3.2 Popölasyon

Bir yandan, Benzetimli Tavlama ve Tabu Arama'da olduđu gibi optimize edilecek aday çözüm sayısı yalnızca bir ise, metasezgisel yöntemi yörünge tabanlı bir algoritma olarak düşünebiliriz. Öte yandan, bir grup çözümle uğraşırsak, teknik popölasyon tabanlı olarak sınıflandırılır. Bu çözücülerin örnekleri GA ve PSO'dur. Ek bir husus, birden fazla olabilen popölasyon sayısıdır. Bu yöntemler, yapılarına göre çok nüfuslu, çok memeli veya ada tabanlı yöntemler olarak adlandırılabilir. Bu yaklaşımların örnekleri, Emperyalist Rekabetçi Algoritma veya dağıtılmış ve paralel GA'lardır. Bu özel durumlarda, bireylerin her bir alt popölasyona dahil edilme şekli açıkça belirtilmeli ve çözümlerin bir bölgeden diğerine göç etme şekli de formüle edilmelidir. Son olarak, Yapay Arı Kolonisi ve Guguk Kuşu Arama gibi iyi bilinen yöntemler, çok etmenli olmaları, yani topluluktaki her bireyin farklı davranabileceđi anlamına gelir [61].

Özetlemek gerekirse, göz önünde bulundurulacak çözümlerin sayısı, popölasyonun yapısı ve bireylerin davranışları, derinlemesine incelenmesi gereken üç husustur. Bir önceki durumda olduđu gibi, işlevsel olmayan gereksinimlerin doğru karar verebilmek için dikkatlice analiz edilmesi gerekir. Örneğın, çözücü dağıtılmış bir ortamda çalıştırılıyorsa, çok nüfuslu bir yöntem veya dağıtılmış bir usta-çırak yaklaşımı (hem senkronize hem de asenkron) umut verici seçenekler olabilir. Ayrıca, çalışma süresi kritik bir husus ise ve problemin değeriendirilmesi pahalı değilse, tek nokta arama algoritması düşünölebilir. Bu bağlamda, uygun alternatifi seçmek için fonksiyonel gereksinimler de analiz edilmelidir. Örneğın, çözüm uzayı dışbükey değilse ve yerel optimumların sayısı yüksekse, arama uzayının keşfini geliştirdiđi için popölasyon tabanlı bir metasezgisel seçilmelidir. Bu özellik özellikle çok modlu optimizasyonda gözlemlenebilir [44].

3.3.3 Operatörler

Operatörlerin tasarımı ve uygulanması da dikkatle yürütölmesi gereken önemli bir adımdır. A priori, işlevlerin işlevsel terimlerle geliştirilmesi için kesin bir kılavuz değildir. Ayrıca, seçim, ardıl veya değıştirme işlevleri gibi farklı türde operatörler vardır. Her halükârda ve kullanılan terminolojiyle ilgili herhangi bir belirsizliđi önlemek için, bireylerin uygulama boyunca nasıl geliştirdiđi standart bir matematik dili kullanılarak detaylandırılmalıdır. Bunu yapmak için, her operatörün girdileri ve çıktıları hem algoritma açıklamaları hem de standart matematiksel gösterim kullanılarak açıklanmalıdır. Operatörlerin doğasını ve nasıl çalıştıkları da açıklanmalıdır. Ayrıca, belirsizlikten kaçınmayı göz önünde bulundurarak, literatürdeki diğer algoritmalarla olası

benzerlikleri tahmin etmek ve (varsa) farklılıkları matematiksel olarak tanımlanmış kavramları bir kez daha kullanarak vurgulamak tavsiye edilir.

Yine, tüm algoritmanın bir parçası olacak tüm operatörleri doğru bir şekilde seçmek veya tasarlamak için işlevsel olmayan gereksinimler dikkatlice incelenmelidir. Örneğin, bazı operatörler, yöntemin keşif kapasitesini artırmak için nihai olarak eksik ve/veya uygulanabilir olmayan çözümlerin (yani tüm kısıtlamaları karşılamayan çözümler) üretilmesine izin verir. Her durumda, gerçek dünya senaryosunun beklenmedik algoritma kesintilerini dikkate alması durumunda bu alternatiflerden kaçınılmalıdır. Ek olarak, çalışma süresinin kritik bir sorun olması durumunda, algoritmanın yakınsamasını destekleyen operatörlere öncelik verilmelidir [62].

3.3.4 Algoritmik Tasarım

Algoritmik tasarım operatörlerin çözüme veya çözüm gruplarına nasıl uygulanacağını belirler. Geliştirilen metasezgisel yöntemin türünü bu tasarımın belirlediği söylenebilir. Bu noktada, algoritmanın genel detaylarının verilmesi zorunlu olmalıdır. Bunu yapmak için, akış diyagramı, matematiksel açıklama veya yöntemin sözde kodu gibi çeşitli alternatifler yararlıdır. Ayrıca, modellenen teknik herhangi bir yeni bileşen içeriyorsa, benzer amaçlar için yapılan diğer algoritmik şemalara yapılan referansları kullanarak yöntemin bu genel açıklamasının yürütülmesi oldukça arzu edilir. Ayrıca, bir çözüm meta-sezgisel oluşturmak için olası alternatiflerin sayısı gerçekten çok fazladır ve burada vurgulanması gereken tüm yönleri belirtmek imkansızdır. Her durumda, açıklanması gereken yönlerden bazıları seçim kriteri, çözümler arasındaki etkileşim kriteri (GA'larda rekombinasyon veya çok nüfuslu metasezgiselde göç açısından), kabul kriteri (yerine koyma) ve sonlandırma kriteridir [63].

Araştırmacılar, metasezgisel tasarımın uygun şekilde seçilmesi için hem işlevsel hem de işlevsel olmayan gereksinimleri iyice düşünmelidir. Örneğin, hesaplama açısından zorlu tasarımlar, yalnızca çalışma süresinin kritik olmadığı durumlarda kabul edilebilir. Aksine, çözümde bazı kaliteden ödün vererek uygulama süresini azaltmak istiyorsak, sonlandırma kriteri uygun ve arzu edilen bir yakınsamaya ulaşmak için bir köşe taşı olacaktır. Uygulanan algoritmalar, gelişmiş ve dikkatle tasarlanmış iletişim mekanizmaları gerektiren dağıtılmış bir ortamda konuşlandırılacaksa, aday çözümler arasındaki etkileşim de büyük önem kazanacaktır [44].

3.4 METASEZGİSELLERİN DEĞERLENDİRİLMESİ

3.4.1 Performans Ölçümü

Bir optimizasyon algoritması, farklı bakış açılarından ve birçok özelliğe dayalı olarak değerlendirilebilir. Geleneksel olarak, ana ölçüler performansla ilgilidir (uygunluk işlevi veya hata ölçüsü). Bununla birlikte, gerçek dünya bağlamında birçok başka olası ilgi ölçüsü vardır:

- Hesaplama karmaşıklığına bağlı olan işlem süresi. Birçok gerçek dünya bağlamında bu yanıt süresi çok önemlidir. Bu bağlamda, ele alınan tüm metasezgisel yöntemler tarafından sunulan yürütme sürelerini içeren bir kayıt oluşturulması önerilir. Bu kayıt, tekniklerin çalıştırıldığı hesaplama ortamı ile ilişkilendirilmelidir. Bu nedenle, bu günlük defteri sonraki dağıtım aşamasında faydalı olacaktır ve özellikle sistem geçişinin algoritmanın çalışma süresi üzerindeki etkisini doğru bir şekilde ölçmek için çok önemlidir.
- Bellek gereksinimleri: Bu, özellikle algoritmanın sınırlı kaynaklara sahip bir donanımda çalışması beklendiğinde önemlidir.
- Dağıtılmış algoritmalar için, iletişim gecikmesi veya elde edilen hızlanma (düğüm sayısına göre) gibi özel ölçüler vardır. Ek olarak, bir düğüm arızalandığında sağlamlık, fazlalık vb. gibi diğer işlevsel olmayan gereksinimler de dikkate alınabilir ve ölçülebilir.
- Özellikle her değerlendirmenin önemli hesaplama kaynakları gerektirdiği problemlerde, oldukça iyi bir çözüm elde etmek için gereken süre. Bu senaryolarda, algoritmalar genellikle yürütme sürelerini azaltmak için vekil modeller uygular.

Genel bir kural olarak, bir optimizasyon algoritmasının performansının değerlendirilmesi, yalnızca tek bir çalıştırmanın ölçüsü rapor edilirse garanti edilemez. Bir değerlendirme metriğinin sağlam tahmin edicileri, yalnızca yeterli bilgi mevcutsa hesaplanabilir. Bu anlamda, aşağıda açıklanan istatistiksel yöntemlerin önemli sonuçlara varabilmesi için birden çok çalışma dikkate alınmalıdır. Birden fazla çalıştırmanın bağımsız olması gerektiğine, yani bir çalıştırmadan diğerine hiçbir bilgi beslenmemesine de özel dikkat gösterilmelidir [64].

3.4.2 DeneySEL Kıyaslama

Bu genellikle araştırmacıların önerilerinin deneySEL değerlendirmesini tasarlarırken yüzleşmeleri gereken ilk karardır. Bu bağlamda, katkının türüne bağlı olarak (daha teorik veya daha uygulamalı), deneyde kullanılan problemlerin doğası farklı olabilir. İlk durumda,

araştırmacılar, farklı tekliflerin veya dikkate alınan algoritmaların performansını veya avantajlarını değerlendirmek için muhtemelen sentetik kıyaslama işlevlerini kullanırdı. İkinci durumda, yazarlar normal olarak çözmeye çalıştıkları problemle ilgili bir dizi örnek önerirler. Bazen, özellikle bir romanla ya da güçlü gereksinimleri olan çok özel bir sorunla uğraşırken, kişi kendi ölçütünü tasarlamak zorundadır. Her durumda, bu problem örnekleri, deneylerin bir veya başka bir teknik seçme kararının önemini değerlendirmede başarılı olmasını sağlamak için birkaç koşula uymalıdır. Bunu yapmak için, kıyaslama ölçütleri, yalnızca işlevsel gereksinimlerin değil, aynı zamanda işlevsel olmayan gereksinimlerin de karşılanma derecesini analiz etmek için tasarlanmalıdır. Ek olarak, karşılaştırmalı analizde, çevrim sıralaması veya döngü sıralaması nedeniyle potansiyel sorunları azaltmak için çok sayıda problem ve/veya tek sayıda problem gibi istatistiksel testlerin nihai uygulamasına izin veren veya teşvik eden özel koşullar da dikkate alınmalıdır [44].

Her durumda, değerli sentetik problem örneklerinin üretilmesi için, güvenilir test senaryoları oluşturmak amacıyla gerçek durumu oluşturan tüm değişkenler kapsamlı bir şekilde incelenmelidir. Bu nedenle, yeni oluşturulan veri kümeleri bu değişkenlere uyarlanmalıdır. Son olarak, herhangi bir gerçek örnek mevcutsa, ilham kaynağı olarak gerçek örnek kullanılarak ek sentetik test durumlarının oluşturulması önerilir [64].

Deneyisel kıyaslama sentetik fonksiyonlardan oluştuğunda, bu tür fonksiyonlar optimizasyon algoritmaları için bir meydan okuma olmalıdır. Bu nedenle, kıyaslama, ele alınması gereken sorunun beklenen özelliklerine bağlı olarak, farklı sayıda yerel optimum, genel optimumun kayması, koordinat sisteminin dönüşü, fonksiyonun alt bileşenlerine ayrılamaz olması, gürültü ve çeşitli problem boyutları gibi farklı nitelikteki ve zorlayıcı özelliklere sahip işlevleri içermelidir. Böyle bir kriter tasarlamak zor bir görev olabilir, bu nedenle literatürdeki bazı mevcut kıyaslamaların kullanılması iyi bir tavsiyedir. İyi bilinen karşılaştırma ölçütlerinin kullanılması, karşılaştırmaya dahil edilecek referans algoritmaların seçimini de kolaylaştırır. Son olarak belirtmek gerekir ki, gerçek ortamda gerçek dünya problemini çözen bir teknik uygulanacak olsa da bu tür veri kümeleri ile testlerin yapılması, geliştirilen önerinin kalitesinin ölçülmesi açısından büyük önem taşımaktadır [65].

3.4.3 İstatistiksel Testler

Özellikle deneyde kullanılan algoritmalar stokastik olduğunda, sonuçların istatistiksel karşılaştırmalarının zorunlu olduğu düşünülmelidir. Ancak istatistiksel karşılaştırmalar yapılsa

bile her zaman doğru yapılmamaktadır. Çıkarımsal hipotez testinde t -testi veya ANOVA ailesi gibi bazı popüler yöntemler vardır. Bununla birlikte, parametrik testler olarak adlandırılan bu testler, uygulandıkları veriler (normallik, homokedastiklik, vb.) Bu varsayımlar tutmazsa, güvenilirlikleri garanti edilmez ve alternatif yaklaşımlar düşünülmelidir. Bu, Wilcoxon testi gibi, altta yatan verilerin dağılımının herhangi bir özel özelliğini varsaymayan parametrik olmayan testlerin durumudur. Sonuç olarak, bu testler daha genel olarak uygulanabilir. Ancak, farklı tekliflerin gerçek hata değerleri yerine göreceli sıralamayı dikkate aldıkları için parametrik testlerden daha az güçlüdürler [66].

Ek olarak, birkaç karşılaştırma yapıldığında kümülatif hata dikkatle değerlendirilmelidir. Örneğin, popüler Wilcoxon testi, iki veri örneğini karşılaştırmak için tasarlanmış bir testtir (genellikle karşılaştırmaya konu olan algoritmaların hatalarından gelir). Aralarında ikiden fazla örnek karşılaştırıldığında kümülatif hata artabilir [44].

Bununla birlikte, istatistiksel testlerin kullanılmasının, sonuçların yorumlanmasında hataların olmayacağını garanti etmediğine dikkat edilmelidir. Aslında, p -değeri kavramı birkaç yanlış yorumlamaya yol açabilir. Aynı sorun, güven aralıkları yöntemleri kullanılırken de ortaya çıkabilir, ancak bunun daha küçük bir ölçekte gerçekleştiği kanıtlanmıştır. Bilhassa, iyi bilinen t -testi veya Wilcoxon testi kullanılırken yapıldığı gibi algoritma çiftleri ile karşılaştırma, karşılaştırılan algoritmaların hiçbirinin en iyi algoritma olarak tanımlanamayacağı sonucuna vararak döngü sıralaması paradoksunu üretebilir. Ayrıca, çoklu algoritmaları karşılaştıran ANOVA gibi yöntemler, belirlenen kazananın algoritma çiftlerinin istatistiksel karşılaştırmaları yoluyla elde edilenden farklı olabileceği en uygun olmayan paradoksun hayatta kalmasına yol açabilir [66].

4. MATERYAL VE METOT

Tez çalışmasında yararlanmış olduğumuz metasezgisel algoritmalar ve bunların performans testleri için kullanmış olduğumuz sentetik kıyaslama fonksiyonları ve gerçek-dünya problemlerinin detaylı açıklamaları bu bölüm altında ele alınmıştır. Ayrıca, bu bölümde Kara Sinek Algoritması olarak isimlendirmiş olduğumuz optimizasyon algoritmasının matematiksel modeli ve sözde kodu yer almaktadır.

4.1 POPÜLASYON TABANLI METASEZGİSELLER

4.1.1 Parçacık Sürü Optimizasyonu (PSO)

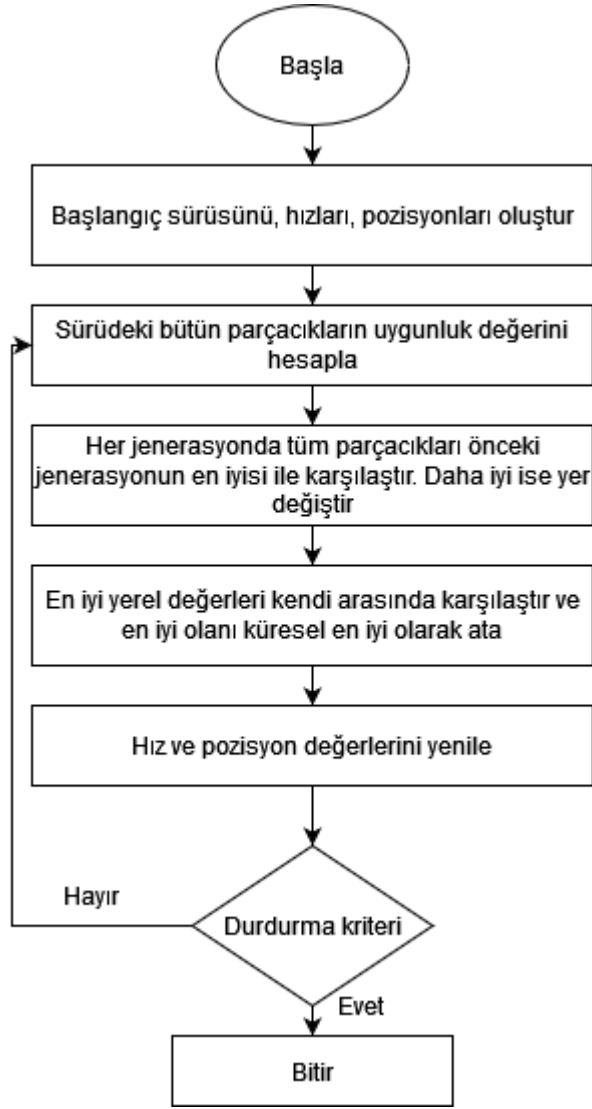
Parçacık Sürü Optimizasyonu (*Particle Swarm Optimization*), 1995 yılında Kennedy ve Eberhart tarafından [8] doğadaki balık ve kuş sürüsü gibi sürü davranışları temel alınarak geliştirilmiştir [67]. PSO'da her parçacığın bir v_i hızı ve x_i konumu vardır ve güncellemeleri Eşitlik 4.1'e göre belirlenebilir:

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^* - x_i^t] \quad (4.1)$$

Burada g^* mevcut en iyi çözümdür ve x_i^* , i -inci parçacığı için bireysel en iyi çözümdür. Burada ε_1 ve ε_2 , 0 ile 1 arasındaki tekdüze dağılımdan alınan iki rasgele değişkendir. Ayrıca α ve β öğrenme parametreleridir. Parçacıkların konumları Eşitlik 4.2'deki gibi güncellenir:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (4.2)$$

Standart PSO algoritmasını genişleten birçok varyant vardır ve en göze çarpan gelişme, muhtemelen bir atalet fonksiyonu kullanmaktır.



Şekil 1. Parçacık Sürü Optimizasyon Algoritmasının Akış Diyagramı [68].

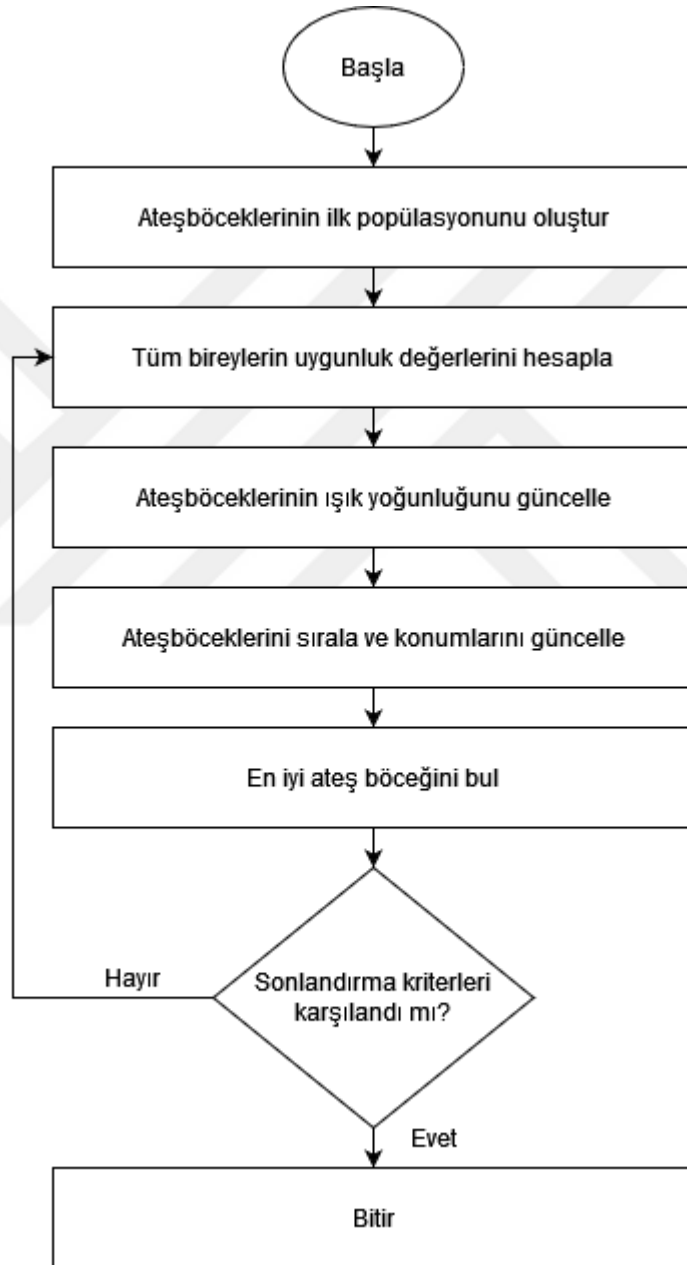
4.1.2 Ateşböceği Algoritması (AA)

Ateşböceği Algoritması (*Firefly Algorithm*), ilk olarak Xin-She Yang tarafından 2007'de [69] ateşböceklerinin yanıp sönme modellerine ve davranışlarına dayalı olarak geliştirilmiştir [67]. AA'da, bir başka daha çekici (daha parlak) ateşböceği j 'ye çekilen ateşböceği i 'nin hareketi, Eşitlik 4.3'teki doğrusal olmayan güncelleme denklemi ile belirlenir:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha \varepsilon_i^t \quad (4.3)$$

Burada β_0 , $r = 0$ mesafesindeki çekicilik değeridir. Bir ateşböceğinin çekiciliği, diğer ateşböceklerinden olan uzaklığa göre değişir. Çekicilik, β 'nın r uzaklığı ile değişimi Eşitlik 4.4'teki gibi tanımlanabilir:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (4.4)$$



Şekil 2. Ateş Böceği Algoritmasının Akış Diyagramı [70].

4.1.3 Guguk Kuşu Arama Algoritması (GKAA)

Guguk Kuşu Arama Algoritması (*Cuckoo Search*), 2009 yılında Yang ve Deb [71] tarafından geliştirilmiştir. GKAA, bazı guguk kuşu türlerinin kuluçka asalaklığına dayanmaktadır. Ek olarak, bu algoritma basit izotropik rasgele yürüyüşlerden daha iyi olan Levy uçuşları ile geliştirilmiştir [67].

Bu algoritma, bir anahtarlama parametresi p_a tarafından kontrol edilen, yerel rasgele yürüyüş ile küresel keşif amaçlı rasgele yürüyüşün dengeli bir kombinasyonunu kullanır. Yerel rastgele yürüyüş Eşitlik 4.5'teki gibi yazılabilir:

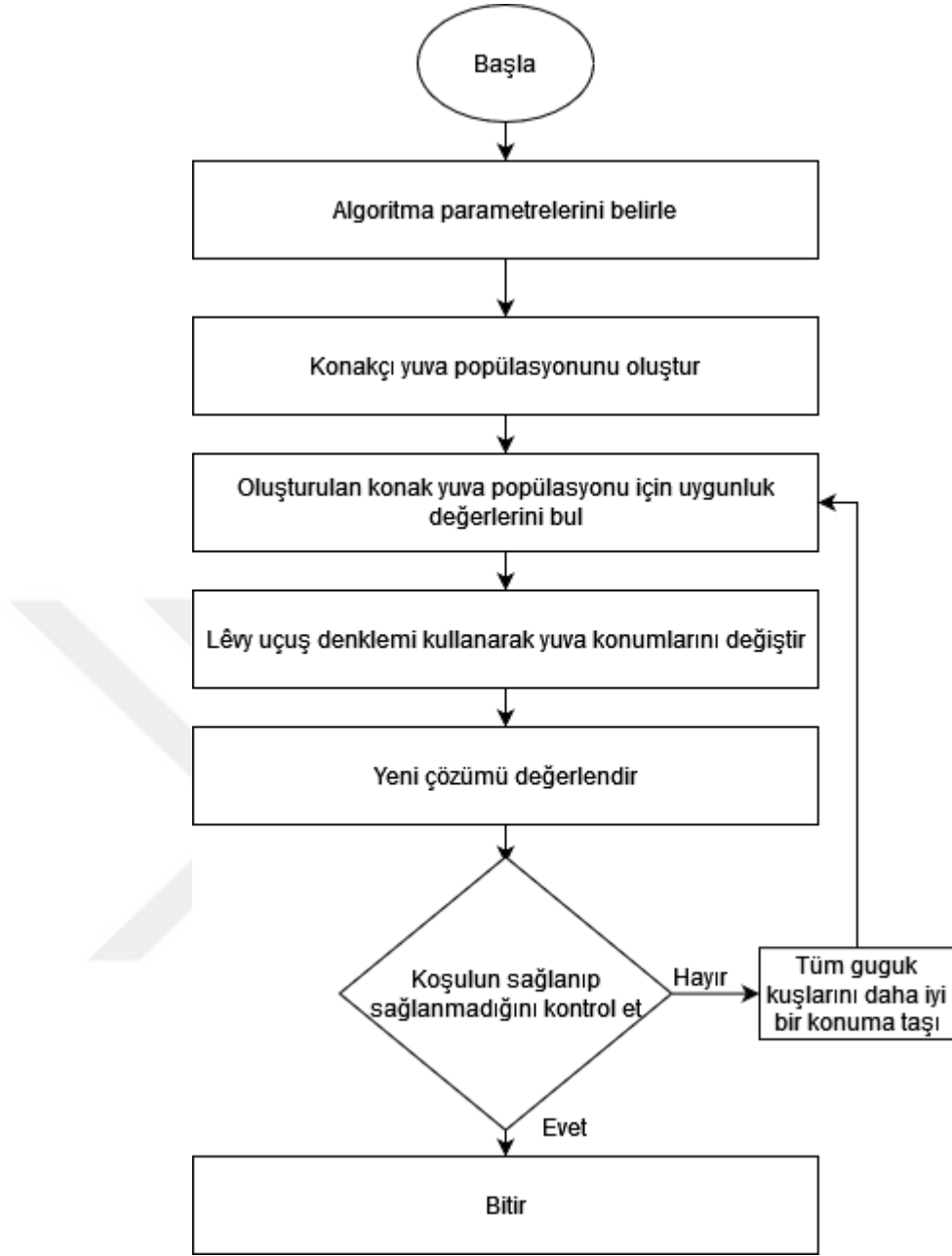
$$x_i^{t+1} = x_i^t + \alpha s H(p_a - \varepsilon)(x_j^t - x_k^t) \quad (4.5)$$

Burada x_j^t ve x_k^t permütasyon tabanlı olarak rastgele seçilen iki farklı çözümü temsil etmektedir. $H(u)$ Heaviside fonksiyonudur ve ε tekdüze bir dağılımdan alınan rastgele bir sayıdır, s Levy dağılımından elde edilen adım boyutudur. Öte yandan, Eşitlik 4.6'da verilen küresel rastgele yürüyüş, Eşitlik 4.7'deki Levy uçuşları ile gerçekleştirilir.

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (4.6)$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \square s_0 > 0) \quad (4.7)$$

Burada $\alpha > 0$, adım boyutlarının ölçüsünü kontrol eden bir ölçekleme faktörüdür ve s_0 , küçük bir sabit adım boyutudur. Bir algoritmanın verimli olması için, yeni çözümlerin önemli bir kısmının, mevcut en iyi çözümden yeterince uzakta konumlarla, uzak alan rasgeleleştirmesiyle üretilmesi gerekir. Bu, sistemin yerel bir optimuma takılmamasını sağlayacaktır.



Şekil 3. Guguk Kuşu Arama Algoritmasının Akış Diyagramı [72].

4.1.4 İstilacı Yabani Ot Optimizasyon Algoritması (İYOOA)

İstilacı Yabani Ot Optimizasyon Algoritması (*Invasive Weed Optimization*), Mehrabian ve Lucas [73] tarafından 2006 yılında geliştirilmiştir. İYOOA, tarımda istilacı yabancı ot kolonileri adı verilen bir olgudan esinlenmiştir. Yaygın bir tanıma göre ot, istem dışı büyüyen bir bitkidir. Yabancı otların bazı yörelerde pek çok kullanım alanı ve faydası bulunsa da aynı bitki insanın ihtiyaç ve faaliyetlerine engel olacak bir bölgede yetişiyorsa buna yabancı ot denir. Bu algoritma,

yabancı ot kolonisinde tohumlama, büyüme ve rekabet gibi temel özellikleri kullanarak optimal çözümlere yakınsamada etkilidir [74].

Yabani otların habitat davranışını simüle etmek için, sürecin bazı temel özellikleri aşağıdaki gibi kabul edilir:

- *Başlangıç popülasyonu*: Arama uzayında sınırlı sayıda tohum dağıtılır.
- *Üreme*: Her tohum çiçekli bir bitkiye dönüşür ve uygunluk değerlerine bağlı olarak tohumlar üretilir. Çim tanelerinin sayısı, S_{max} 'tan S_{min} 'e doğru doğrusal olarak Eşitlik 4.8'deki gibi azalır:

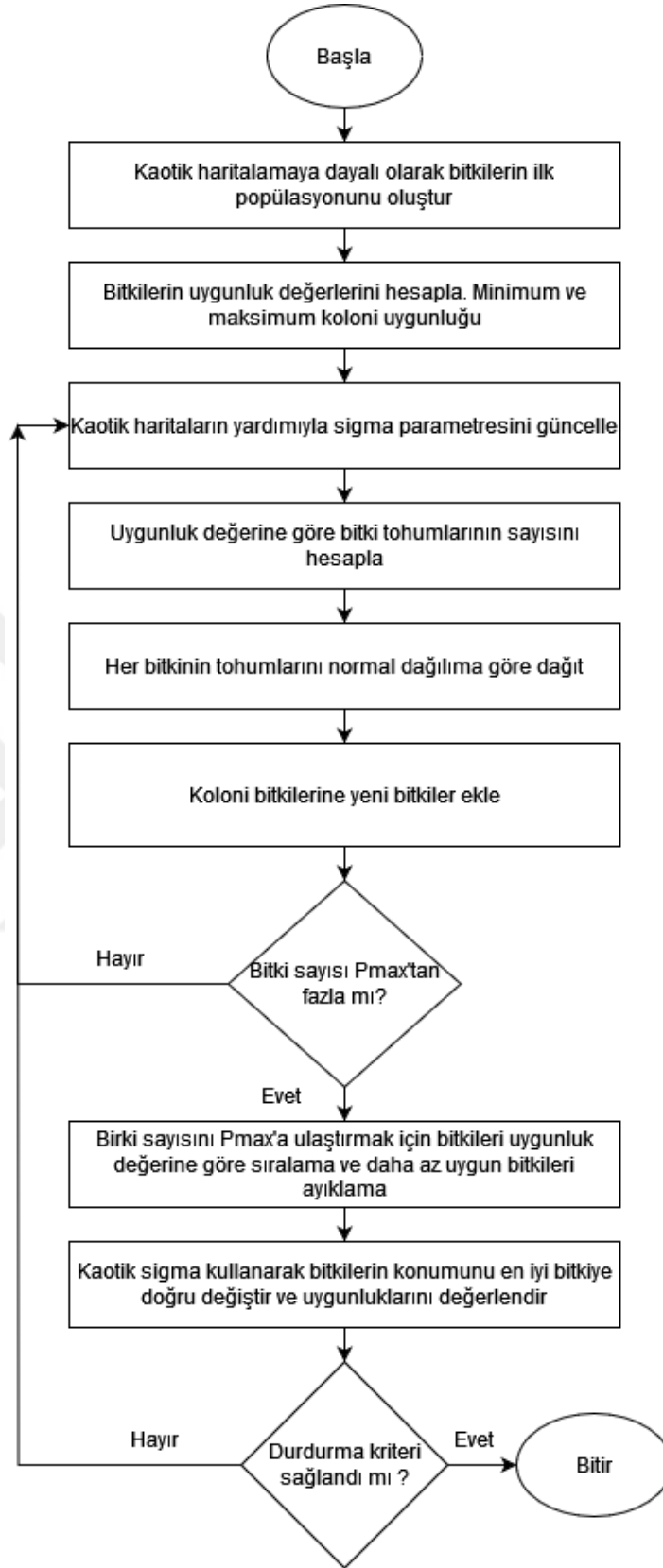
$$n(w_i) = \frac{s_{max} (\max fit - fit(w_i)) s_{min} (fit(w_i) - \min fit)}{\max fit - \min fit} \quad (4.8)$$

- *Spektral Yayılma*: Bir grup tarafından üretilen tohumlar, ortalama ekim konumu ve standart sapma (σ) ile normal dağılımda Eşiklik 4.9'daki denklemle üretilir:

$$\sigma_t = \left(\frac{T-t}{T} \right)^n (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (4.9)$$

Burada T maksimum yineleme sayısı, σ_t mevcut standart sapma, n doğrusal olmayan modülasyon indeksidir. Bu dönüşüm, aralıktaki bir tohum tanesinin düşüşünün her adımda doğrusal olmayan bir şekilde azalmasını sağlayarak daha uygun değerdeki bitkilere yol açar ve uygun olmayan bitkileri ortadan kaldırır ve r 'den K seçimine kadar olan transfer modunu gösterir.

- *Rekabetçi yoksunluk*: Otların sayısı kolonideki maksimum ot sayısını (P_{max}) aşarsa, en kötü uygunluktaki yabancı ot koloniden çıkarılır, böylece kolonide sabit sayıda bitki kalır.
- Bu işlem, maksimum iterasyon sayısına ulaşılan kadar devam eder ve ardından otların minimum koloni maliyeti fonksiyonu saklanır.



Şekil 4. İstilacı Yabani Ot Optimizasyon Algoritmasının Akış Diyagramı [74].

4.2 KARASİNEK ALGORİTMASI

4.2.1 Yaşam Döngüsü ve Davranışları

Ev sineği (*Musca domestica* Linnaeus), hem çiftliklerde hem de evlerde yaşayan, iyi bilinen kozmopolit bir haşeredir. Bu tür, her zaman insanlarla veya insanların faaliyetleriyle birlikte bulunur. Domuz ve kümes hayvanı çiftliklerinde, at ahırlarında ve çiftliklerde bulunan en yaygın türdür. Ev sinekleri baş belası olmanın yanı sıra hastalığa neden olan organizmaları da taşıyabilirler. Aşırı sinek popülasyonları, yalnızca çiftlik çalışanları için rahatsız edici olmakla kalmaz, aynı zamanda yakınarda insan yerleşimleri olduğunda, bir halk sağlığı sorunu ortaya çıkabilir [75].

Gün boyunca, sinekler çoğunlukla beslenme ve yetiştirme çevresinde veya etrafında toplanır. Işık, sıcaklık, nem ve yüzey üzerindeki reaksiyonlarından büyük ölçüde etkilenirler. Dinlenmek için tercih edilen sıcaklık 35 °C ile 40 °C. Sinekler düşük hava nemliliğinde en aktiftir. Yüksek sıcaklıklarda (20 °C'nin üstünde), Çoğu ev sinekleri vakitlerini açık hava veya açık havada kapalı alanlarda geçirirler. Sinekler geceleri aktif değildir. Kanatlı çiftliklerinde, gece vakti, sinek kümeleri çoğunlukla dallarda ve çalılarda bulunurken, kapalı alandaki popülasyonların neredeyse tamamı genellikle kümeslerin tavan alanında toplanır [76].

Ev sineği, farklı yumurta, larva veya kurtçuk, pupa ve yetişkin aşamaları ile tam bir metamorfoza sahiptir. Ev sineği, kışı ya larva ya da pupa aşamasında gübre yığınlarının altında ya da diğer korunan yerlerde geçirir. Ev sineğinin gelişimi için genellikle ılık yaz koşulları optimumdur ve yaşam döngüsünü yedi ila on gün gibi kısa bir sürede tamamlayabilir. Bununla birlikte, yetersiz koşullar altında, yaşam döngüsü iki aya kadar sürebilir. Ilıman bölgelerde yılda 10 ila 12 nesil meydana gelebilirken, subtropikal ve tropikal bölgelerde 20'den fazla nesil meydana gelebilir.

Dişi sinekler yumurtalarını çürük, fermente edici veya çürüten hayvansal veya bitkisel menşeli organik materyal üzerine bırakır. Birikmiş hayvan dışkılarının yığınları ev sinekleri için en önemli üreme yerleri arasındadır. Çöp, çoğalma için ana ortamı sağlar. Evde ve halkın genelinde gıda hazırlama, yemek pişirme ve sunma ile bağlantılı yerler ve meyveler de dâhil olmak üzere yiyeceklerin taşınması, depolanması ve satışı yapılan yerler üreme alanlarıdır.

4.2.1.1 Yumurta

Yaklaşık 1,2 mm uzunluğundaki beyaz yumurta tek tek yumurtlanır, ancak yumurtalar küçük gruplar halinde istiflenir. Her dişi sinek, üç ila dört günlük bir süre boyunca 75 ila 150 yumurtalık birkaç partide 500'e kadar yumurta bırakabilir. Maksimum yumurta üretimi 25 ila 30°C'de gerçekleşir. Çoğu zaman, birkaç sinek yumurtalarını birbirine yakın bir yere bırakarak büyük larva ve pupa kütlelerine yol açar. Yumurtalar nemli kalmalıdır, aksi takdirde yumurtadan çıkmazlar.



Şekil 5. Yetişkin bir karasinek (*Musca domestica*) ve yumurtaları.

4.2.1.2 Larva

Erken dönem larvaları 3 ila 9 mm uzunluğunda, tipik kremsi beyazımsı renkte, silindirik fakat başa doğru incelmektedir. Kafa bir çift koyu kanca içerir. Bacaksız kurtçuk, sıcak havalarda sekiz ila 20 saat içinde yumurtadan çıkar. Kurtçuklar, yumurtanın bırakıldığı malzemede hemen beslenmeye ve gelişmeye başlar.

Larva üç evreden geçer ve 7 ila 12 mm uzunluğunda tam yetişkin bir kurtçuk yağlı, krem renkli bir görünüme sahiptir. Yüksek nemli gübre, ev sineği larvasının hayatta kalmasını kolaylaştırır. Larva gelişimi için en uygun sıcaklık 35 ila 38°C'dir, ancak larva sağkalımının en yüksek olduğu sıcaklık 17 ila 32 °C'dir. Larvalar optimum sıcaklıklarda gelişimlerini 4 ila 13 günde tamamlarlar, ancak 12 ila 17 °C sıcaklıklarda 14 ila 30 gün gerektirirler.

Hayvan gübresi gibi besin açısından zengin substratlar mükemmel bir gelişimsel substrat sağlar. Larva gelişimi için çok az gübreye ihtiyaç vardır ve az miktarda bozulmuş gübre içeren kum veya toprak, başarılı bir yer altı gelişimi sağlar. Kurtçuk tam olarak büyüdüğünde, üreme

materyalinin yakınında kuru ve serin bir yere 50 fit kadar sürünebilir ve pupa aşamasına dönüşebilir.



Şekil 6. Karasinek larvaları [77].

4.2.1.3 Pupa

Yaklaşık 8 mm uzunluğundaki pupa aşaması, pupa yaşlandıkça rengi sarı, kırmızı, kahverengi ve siyah arasında değişen son larva derisinden oluşan bir pupa durumunda geçer. Pupanın şekli larvadan oldukça farklıdır ve her iki ucu keskin bir şekilde yuvarlaktır. Pupalar gelişimlerini 32 ila 37 °C'de iki ila altı günde tamamlarlar, ancak yaklaşık 14 °C'de 17 ila 27 gün gerekir. Ortaya çıkan sinek, pupa durumundan, kafasının ön tarafında pnömatik bir çekiç gibi kasayı kırmak için kullandığı ptilinum adı verilen dönüşümlü olarak şişen ve küçülen bir kese kullanılarak kaçar.



Şekil 7. Karasineklerin pupa dönemleri.

4.2.1.4 Yetiřkinlik Dönemi

Kara sinek, 6 ila 7 mm uzunluğundadır ve diři genellikle erkekten daha büyüktür. Diři, gözleri arasındaki nispeten geniş boşlukla erkekten ayırt edilebilir (erkeklerde gözler neredeyse birbirine değecektir). Yetiřkin sineğin baři kırmızımsı gözlerle ve süngerimsi ağız parçalarına sahiptir. Göğüs dört dar siyah řerit taşır ve dördüncü uzunlamasına kanat damarında yukarı doğru keskin bir kıvrım vardır. Erkeğin alt tarafı sarımsıdır.



řekil 8. Yetiřkin Bir Karasinek.

Yetiřkinler genellikle 15 ila 25 gün yaşar, ancak iki aya kadar yaşayabilirler. Yiyecek olmadan sadece iki ila üç gün hayatta kalırlar. Uzun ömür, uygun gıdanın, özellikle řekerin mevcudiyeti ile artar. Hayvan gübresine erişim yetiřkin ömrünü uzatmaz ve daha düşük sıcaklıklarda daha uzun yaşarlar. Çiftleşmeden önce yemek yemeleri gerekir ve çiftleşme iki dakika gibi kısa bir sürede veya 15 dakika kadar uzun bir sürede tamamlanır. Yumurtlama, çiftleşmeden dört ila 20 gün sonra başlar. Diři sineklerin yumurta üretebilmeleri için uygun besine (protein) erişmeleri gerekir ve gübre tek başına yeterli değildir. Sineklerin potansiyel üreme kapasitesi muazzamdır, ama neyse ki asla gerçekleştirilemez. Bilim adamları, Nisan ayında üremeye başlayan bir çift sineğin, optimal koşullar altında ve hepsinin yaşaması halinde ağustos ayına kadar yaklaşık 191 seksilyon sinek ataları olabileceğini hesaplamışlardır.

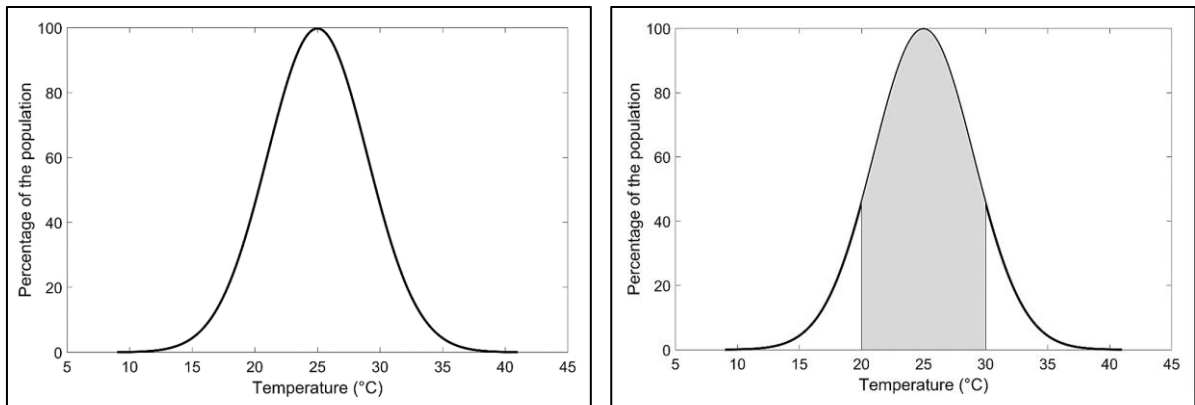
ABD, Teksas'ta yapılan bir araştırmaya göre üreme alanı uygunluğu (azalan sırayla), at gübresi, insan dışkısı, inek gübresi, mayalanan bitkisel maddeler ve mutfak atıklarıdır. Bununla

birlikte, başka bir araştırma, domuz, at, koyun, sığır ve kümes hayvanı içeren yapıların sinek bolluğunda farklılık gösterdiğini, en çok domuz tesisleri ve en az kümes hayvanı içerdiğini bulmuştur.

Sadece sınırlı sayıda çalışma hem sıcaklığın hem de yoğunluğun böceklerin aktivitesi üzerindeki etkisini ele almıştır. Ancak her iki faktörün de beslenme aktivitesini etkileyebildiği gösterilmiştir. Mevcut çalışma [78], sıcaklığın gerçekten de gündüz saatlerinde karasineğin lokomotor aktivitesi üzerinde büyük bir etkiye sahip olduğunu göstermektedir. Locomotor aktivite, 10 ila 30–35 °C aralığında sıcaklıkla artmış ve sıcaklık 35 °C'nin üzerindeyken azalmıştır. Belirli bir bölgedeki uçuş sayısı, üreme yerlerinin mevcudiyetine, güneşli, sıcaklık ve neme bağlıdır. Uçma yoğunlukları ortalama olarak en yüksek 20-25 °C sıcaklık; bunun üstünde ve altında sıcaklıklar ve 45 °C'nin üzerindeki sıcaklıklarda ve 10 °C'nin altındaki sıcaklıklarda saptanamaz hale gelir. Çok düşük sıcaklıklarda, tür yetişkinlikte uykulu bir halde hayatta kalabilir.

4.2.2 Önerilen Yöntem

Önerilen yöntem, karasineklerin doğal ortamlarındaki sosyal davranışlarından esinlenerek geliştirilmiş ve Karasinek Algoritması (KSA) olarak isimlendirilmiştir. İlkbaharla başlayan ve yaz mevsimi sona erinceye dek hemen hemen her yerde karşılaşılabileceğimiz bu sinek türleri, özellikle yaşam alanlarımızda beslenme ve üreme faaliyetlerini gerçekleştirmektedir. Karasineklerin motor aktiviteleri belirli bir sıcaklık ve nem oranına bağlı olarak değişiklik göstermektedir. Buradan hareketle, ortam sıcaklığına bağlı olarak karasineklerin ideal uçma yoğunluğu grafiği Şekil 4-1'deki gibidir.



Şekil 4-1: Karasineklerin ortam sıcaklığına göre ideal uçma yoğunluğu

Popülasyondaki karasineklerin ortam sıcaklığına göre uçuş oranları hesaplanırken, Eşitlik 4.10’de gösterilen normal Olasılık Yoğunluk Fonksiyonundan (nPDF) faydalanılmıştır.

$$nPDF = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(T_E - \mu)^2}{2\sigma^2}\right) \quad (4.10)$$

Burada μ , ortalama sıcaklık değerini, σ sıcaklığın standart sapmasını ve T_E ise ortamın gözlemlenen sıcaklık değerini temsil etmektedir. Önerilen yöntemde $\mu = 25$ ve $\sigma = 4$ olarak kullanılmıştır. Buna göre, popülasyondaki uçan karasinek sayısını hesaplamak için Eşitlik 4.11’de verilen formül kullanılmıştır.

$$H_M = 10 \cdot nPDF \cdot N \quad (4.11)$$

Burada N , popülasyon sayısını temsil etmektedir. Ortamın gözlemlenen sıcaklık değeri (T_E) Eşitlik 4.12’te gösterildiği gibi her bir iterasyonda rastgele olarak yeniden üretilir.

$$T_E = \text{rand}(T_{\min}, T_{\max}) \quad (4.12)$$

Burada $T_{\min}$ ve $T_{\max}$ sırasıyla minimum ve maksimum sıcaklık değerlerini temsil etmektedir. Önerilen yöntemde $T_{\min} = 9$ ve $T_{\max} = 41$ olarak kullanılmıştır. Popülasyon içerisinde uçuş eyleminde bulunacak karasineklerin seçilmesi ve yönlerinin tayin edilmesi için Eşitlik 4.13’ten yararlanılmaktadır.

$$X^B = \text{Brownian}(\text{rand}_p(N, H_M), D) \quad (4.13)$$

Burada $\text{rand}_p()$ tam sayıların rastgele permütasyonunun üreten bir fonksiyon, $\text{Brownian}()$, Brown hareketi olarak bilinen ve bir sıvı içerisinde yüzen veya asılı durumdaki parçacıkların rastlantısal hareketi için kullanılan fonksiyonu ve D ise problemin boyutunu temsil etmektedir. Buna göre, uçuş eyleminde bulunan karasineklerin her bir iterasyondaki yeni konumları Eşitlik 4.14’te gösterildiği gibi hesaplanmaktadır.

$$x_{i,j}^{t+1} = x_{i,j}^t + X_{i,j}^B \quad (4.14)$$

Burada t iterasyon sayısını, i uçan bir karasineğin popülasyon indisini, j ise i -inci karasineğin konum vektör indisini temsil etmektedir. Popülasyonun geri kalan kısmındaki (uçmayan) karasinekler (H_W) Eşitlik 4.15’de gösterildiği gibi hesaplanır. Bu durumdaki karasinekler, ışık ve bağıl nem faktörlerine bağlı olarak uçuş eylemi göstermeden farklı yönlerde hareket etme eğilimindedir.

$$H_W = N - H_M \quad (4.15)$$

Bir diğer parametre olan λ ise ortamdaki ışığın şiddetine ve neme bağlı olarak karasineğin hızını ve hareket yönünü ifade etmektedir. Bu parametre Eşitlik 4.16’de gösterildiği gibi her bir iterasyonda yeniden üretilmektedir.

$$\lambda = (-L + 2L \cdot \text{rand}_N(H_W)) \cdot \varphi \quad (4.16)$$

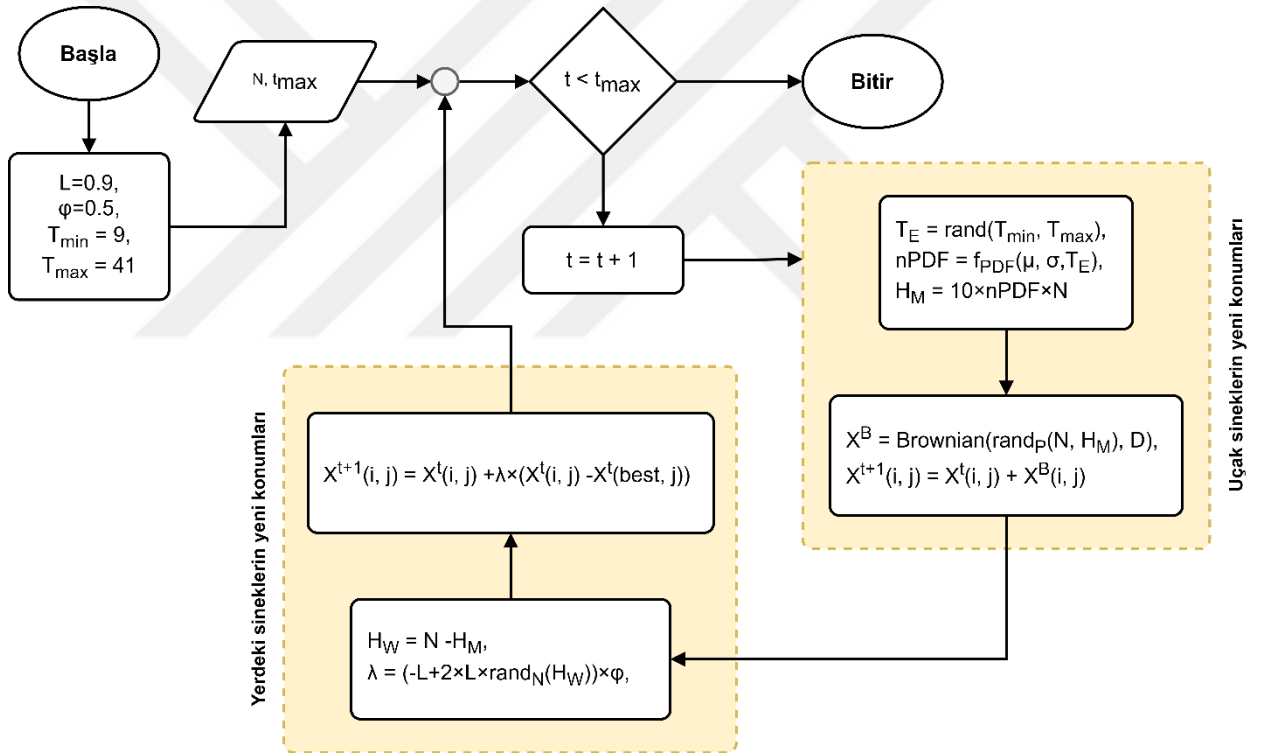
Burada L ortamın 0 ila 1 arasındaki ışıklık değerini temsil etmektedir. L ’nin değeri 1’e doğru yaklaştıkça karasinekler daha hızlı hareket etmektedir. Bunun tersi durumunda ise karasineklerin konum değiştirme hızı ve aralığı azalmaktadır. L ’nin sıfır olduğu durumda ise herhangi bir konum değişikliği söz konusu değildir. Deneysel çalışmalarımızda $L = 0,9$ olarak belirlenmiştir. Eşitlikteki diğer ifadeler; φ , bağıl nem oranını (varsayılan değeri 0,5 olarak belirlenmiştir) ve $\text{rand}_N(\)$ ise normal dağılım gösteren rasgele sayı üreticisini temsil etmektedir. Buna göre, yerde gezinen karasineklerin her bir iterasyondaki yeni konumları Eşitlik 4.17’deki gibi hesaplanmaktadır.

$$x_{i,j}^{t+1} = x_{i,j}^t + \lambda (x_{i,j}^t - x_{best,j}^t) \quad (4.17)$$

Tüm bu işlemler, belirlen yineleme sayısı boyunca tekrar edilerek optimal çözümün bulunması amaçlanır. KSA, hem keşif hem de sömürü kriterleri göz önünde bulundurularak tasarlanmış ve çeşitli türdeki optimizasyon problemleri ile test edilmiştir. Şekil 9’da KSA’nın akış diyagramı verilmiştir. Tablo 1’de önerilen yöntemde kullanılan sabitler verilmiştir.

Tablo 1. Önerilen Yöntemin Sabitleri

Sabit	Açıklama
$\mu = 25$	Ortalama sıcaklık değeri
$\sigma = 4$	Sıcaklığın standart sapması
$T_E = 25$	Ortamın gözlemlenen sıcaklık değeri
$T_{\min} = 9$	Minimum sıcaklık değerleri
$T_{\max} = 41$	Maksimum sıcaklık değerleri
$L = 0,9$	Işıklık değeri
$\varphi = 0,5$	Bağıl nem oranı



Şekil 9. Karasinek Algoritmasının Akış Diyagramı.

4.3 DENEYLERDE KULLANILAN PROBLEMLER

4.3.1 Gerçek Dünya Problemleri

Gerçek dünya problemleri, günlük hayatımızda karşımıza çıkan karmaşık tasarımların daha verimli olmasını sağlamayı amaçlayan bir çalışma alanıdır [79, 80]. Elle çözmenin mümkün olmadığı çok değişken ve kısıtın bulunduğu bu problemlerin çözümü için matematiksel modeller oluşturulmuştur. Matematiksel denklemler, bilgisayar kullanılarak geliştirilen algoritmalar yardımıyla çözülerek optimum sonuca ulaşmak amaçlanmaktadır. Sayısal optimizasyon problemlerinde iyi performans gösteren algoritmalar, gerçek dünya problemlerini çözmek için etkili yöntemler olarak kabul edilir.

Bu çalışmada, önerdiğimiz algoritmayı test etmek amacıyla gerçek dünya problemlerinden yararlanılmıştır. Bu problemler; dört kademeli dişli kutusu problemi, hidrostatik baskı yatağı tasarımı problemi, kaynaklı kiriş tasarımı problemi ve endüstriyel soğutma sisteminin optimum tasarım problemi olmak üzere dört tanedir.

4.3.1.1 Dört Kademeli Dişli Kutusu Problemi

Bu tasarım örneğinde, dişli ve pinyon konumlarını, diş sayısını ve boşluk kalınlığını belirlemek için 86 tasarım kısıtlaması ile 22 tasarım değişkenine sahip dört kademeli bir dişli kutusunun ağırlık optimizasyonu ele alınmıştır. Bu problemin uygulanabilir arama uzayı, birçok yerel çözümle 0:0001'den daha düşük bir orana sahiptir. Sorun şu şekilde tanımlanır: [81]

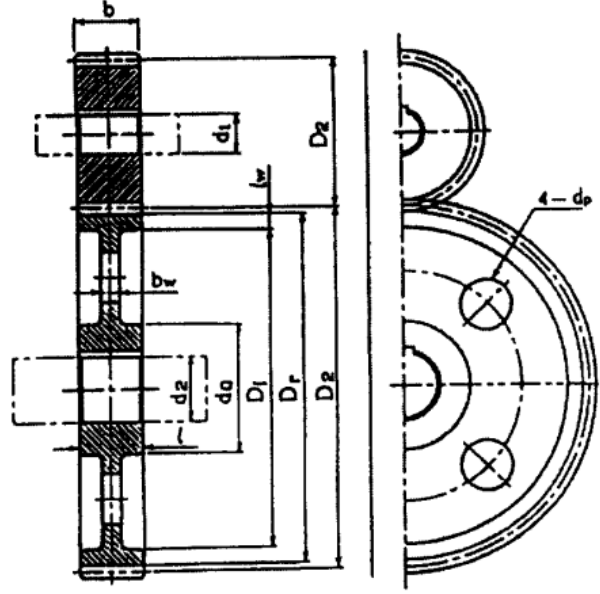


Fig. 1: An OWD problem of the gear

Şekil 10. Dört Kademeli Dişli Kutusu Problemi [82].

Eşitlik 4.18’da problemin amaç fonksiyonu verilmiştir.

$$f(\vec{x}) = \frac{\pi}{1000} \sum_{i=1}^4 \frac{b_i c_i^2 (N_{pi}^2 + N_{gi}^2)}{(N_{pi} + N_{gi})^2}, \text{ where } i = \{1, \dots, 4\} \quad (4.18)$$

Problemdeki kısıtlar Eşitlik 4.19 ile 4.51 arasında verilmiştir.

$$g_1(\vec{x}) = \left(\frac{366000}{\Pi w_1} + \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \right) \left(\frac{(N_{p1} + N_{g1})^2}{4b_1 c_1^2 N_{p1}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (4.19)$$

$$g_2(\vec{x}) = \left(\frac{366000 N_{g1}}{\Pi w_1 N_{p1}} + \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \right) \left(\frac{(N_{p2} + N_{g2})^2}{4b_2 c_2^2 N_{p2}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (4.20)$$

$$g_3(\vec{x}) = \left(\frac{366000 N_{g1} N_{g2}}{\Pi w_1 N_{p1} N_{p2}} + \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \right) \left(\frac{(N_{p3} + N_{g3})^2}{4b_3 c_3^2 N_{p3}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (4.21)$$

$$g_4(\vec{x}) = \left(\frac{366000 N_{g1} N_{g2} N_{g3}}{\Pi w_1 N_{p1} N_{p2} N_{p3}} + \frac{2c_4 N_{p4}}{N_{p4} + N_{g4}} \right) \left(\frac{(N_{p4} + N_{g4})^2}{4b_4 c_4^2 N_{p4}} \right) - \frac{\sigma_N J_R}{0.0167 W K_0 K_m} \leq 0 \quad (4.22)$$

$$g_5(\bar{x}) = \left(\frac{366000}{\Pi w_1} + \frac{2c_1 N_{p1}}{N_{p1} + N_{g1}} \right) \left(\frac{(N_{p1} + N_{g1})^3}{4b_1 c_1^2 N_{g1} N_{p1}^2} \right) - \left(\frac{\sigma_H}{c_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334WK_0 K_m} \right) \leq 0 \quad (4.23)$$

$$g_6(\bar{x}) = \left(\frac{366000 N_{g1}}{\Pi w_1 N_{p1}} + \frac{2c_2 N_{p2}}{N_{p2} + N_{g2}} \right) \left(\frac{(N_{p2} + N_{g2})^3}{4b_2 c_2^2 N_{g2} N_{p2}^2} \right) - \left(\frac{\sigma_H}{c_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334WK_0 K_m} \right) \leq 0 \quad (4.24)$$

$$g_7(\bar{x}) = \left(\frac{366000 N_{g1} N_{g2}}{\Pi w_1 N_{p1} N_{p2}} + \frac{2c_3 N_{p3}}{N_{p3} + N_{g3}} \right) \left(\frac{(N_{p3} + N_{g3})^3}{4b_3 c_3^2 N_{g3} N_{p3}^2} \right) - \left(\frac{\sigma_H}{c_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334WK_0 K_m} \right) \leq 0 \quad (4.25)$$

$$g_8(\bar{x}) = \left(\frac{366000 N_{g1} N_{g2} N_{g3}}{\Pi w_1 N_{p1} N_{p2} N_{p3}} + \frac{2c_4 N_{p4}}{N_{p4} + N_{g4}} \right) \left(\frac{(N_{p4} + N_{g4})^3}{4b_4 c_4^2 N_{g4} N_{p4}^2} \right) - \left(\frac{\sigma_H}{c_p} \right)^2 \left(\frac{\sin(\phi) \cos(\phi)}{0.0334WK_0 K_m} \right) \leq 0 \quad (4.26)$$

$$g_{9-12}(\bar{x}) = -N_{pi} \sqrt{\frac{\sin^2(\phi)}{4} - \frac{1}{N_{pi}} + \left(\frac{1}{N_{pi}} \right)^2} + N_{gi} \sqrt{\frac{\sin^2(\phi)}{4} - \frac{1}{N_{gi}} + \left(\frac{1}{N_{gi}} \right)^2} + \frac{\sin(\phi)(N_{pi} + N_{gi})}{2} + CR_{\min} \pi \cos(\phi) \leq 0 \quad (4.27)$$

$$g_{13-16}(\bar{x}) = d_{\min} - \frac{2c_i N_{pi}}{N_{pi} + N_{gi}} \leq 0 \quad (4.28)$$

$$g_{17-20}(\bar{x}) = d_{\min} - \frac{2c_i N_{gi}}{N_{pi} + N_{gi}} \leq 0 \quad (4.29)$$

$$g_{21}(\bar{x}) = x_{p1} + \left(\frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \right) - L_{\max} \leq 0 \quad (4.30)$$

$$g_{22-24}(\bar{x}) = L_{\max} + \left(\frac{(N_{pi} + 2)c_i}{N_{gi} + N_{pi}} \right)_{i=2,3,4} + x_{g(i-1)} \leq 0 \quad (4.31)$$

$$g_{25}(\bar{x}) = -x_{p1} + \frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} \leq 0 \quad (4.32)$$

$$g_{26-28}(\bar{x}) = \left(\frac{(N_{pi} + 2)c_i}{N_{pi} + N_{gi}} - x_{g(i-1)} \right)_{i=2,3,4} \leq 0 \quad (4.33)$$

$$g_{29}(\bar{x}) = y_{p1} + \frac{(N_{p1} + 2)c_1}{N_{p1} + N_{g1}} - L_{\max} \leq 0 \quad (4.34)$$

$$g_{30-32}(\bar{x}) = -L_{\max} + \left(\frac{c_i(2 + N_{pi})}{N_{pi} + N_{gi}} + y_{g(i-1)} \right)_{i=2,3,4} \leq 0 \quad (4.35)$$

$$g_{33}(\bar{x}) = \frac{(2 + N_{p1})c_1}{N_{p1} + N_{g1}} - y_{p1} \leq 0 \quad (4.36)$$

$$g_{34-36}(\bar{x}) = \left(\frac{c_i(2 + N_{pi})}{N_{pi} + N_{gi}} - y_{g(i-1)} \right)_{i=2,3,4} \leq 0 \quad (4.37)$$

$$g_{37-40}(\bar{x}) = -L_{\max} + \frac{c_i(2 + N_{gi})}{N_{pi} + N_{gi}} + x_{gi} \leq 0 \quad (4.38)$$

$$g_{41-44}(\bar{x}) = -x_{gi} + \frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} \leq 0 \quad (4.39)$$

$$g_{45-48}(\bar{x}) = y_{gi} + \frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} - L_{\max} \leq 0 \quad (4.40)$$

$$g_{49-52}(\bar{x}) = -y_{gi} + \frac{(N_{gi} + 2)c_i}{N_{pi} + N_{gi}} \leq 0 \quad (4.41)$$

$$g_{53-56}(\bar{x}) = (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)(-N_{pi} + 0.945c_i - N_{gi})(-1) \leq 0 \quad (4.42)$$

$$g_{57-60}(\bar{x}) = (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)(-N_{pi} + 0.646c_i - N_{gi}) \leq 0 \quad (4.43)$$

$$g_{61-64}(\bar{x}) = (b_i - 5.175)(b_i - 3.175)(b_i - 12.70)(-N_{pi} + 0.504c_i - N_{gi}) \leq 0 \quad (4.44)$$

$$g_{65-68}(\bar{x}) = (b_i - 5.175)(b_i - 3.175)(b_i - 8.255)(0c_i - N_{gi} - N_{pi}) \leq 0 \quad (4.45)$$

$$g_{69-72}(\bar{x}) = (b_i - 8.255)(b_i - 5.715)(b_i - 12.70)(N_{gi} + N_{pi} - 1.812c_i) \leq 0 \quad (4.46)$$

$$g_{73-76}(\bar{x}) = (b_i - 8.255)(b_i - 3.175)(b_i - 12.70)(-0.945c_i + N_{pi} + N_{gi}) \leq 0 \quad (4.47)$$

$$g_{77-80}(\bar{x}) = (b_i - 5.175)(b_i - 3.175)(b_i - 12.70)(-0.646c_i + N_{pi} + N_{gi})(-1) \leq 0 \quad (4.48)$$

$$g_{81-84}(\bar{x}) = (b_i - 5.175)(b_i - 3.175)(b_i - 8.255)(N_{pi} + N_{gi} - 0.504c_i) \leq 0 \quad (4.49)$$

$$g_{85} = w_{\min} - \frac{w_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} \leq 0 \quad (4.50)$$

$$g_{86} = \frac{w_1(N_{p1}N_{p2}N_{p3}N_{p4})}{(N_{g1}N_{g2}N_{g3}N_{g4})} - w_{\max} \leq 0 \quad (4.51)$$

Eşitlik 4.52’te bazı değişkenlerin alabileceği değerler ve sabit ifadeleri verilmiştir.

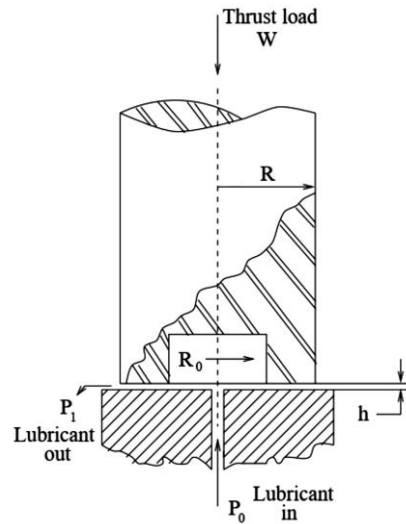
$$\begin{aligned}
\bar{x} &= \{N_{p1}N_{g1}N_{p2}N_{g2}...b_1b_2...x_{p1}x_{g1}x_{g2}...y_{p1}y_{g1}y_{g2}...y_{g4}\}, \\
c_i &= \sqrt{(y_{gi} - y_{pi})^2 + (x_{gi} - x_{pi})^2}, \quad K_0 = 1.5, \quad d_{\min} = 25, \\
J_R &= 0.2, \quad \phi = 120^\circ, \quad W = 55.9, \quad K_M = 1.6, \quad CR_{\min} = 1.4, \\
L_{\max} &= 127, \quad c_p = 464, \quad \sigma_H = 3290, \quad w_{\max} = 255, \quad w_1 = 5000, \\
\sigma_N &= 2090, \quad w_{\min} = 245.
\end{aligned} \tag{4.52}$$

Eşitlik 4.53'te bazı değişkenlerin alabileceği değerler ve eşitsizlik ifadeleri verilmiştir.

$$\begin{aligned}
b_i &\in \{3.175, 12.7, 8.255, 5.715\} \\
y_{p1}, x_{p1}, y_{gi}, x_{gi} &\in \{12.7, 38.1, 25.4, 50.8, 76.2, 63.5, 88.9, 114.3, 101.6\} \\
7 \leq N_{gi}, N_{pi} &\leq 76 \in \mathbb{N}
\end{aligned} \tag{4.53}$$

4.3.1.2 Hidrostatik Baskı Yatağı Tasarımı Problemi

Bu tasarım probleminin temel amacı, dört tasarım değişkeni kullanarak yatak gücü kaybını optimize etmektir. Bu tasarım değişkenleri yağ viskozitesi (μ), yatak yarıçapı (R), akış hızı (Q) ve girinti yarıçapı (R_0)'dır. Bu problem, giriş yağ basıncı, yük taşıma kapasitesi, yağ filmi kalınlığı ve giriş yağ basıncı ile ilişkili yedi doğrusal olmayan kısıtlama içerir. Problem şu şekilde tanımlanır. [83]



Şekil 11. Hidrostatik Baskı Yatağı Tasarım Problemi [84].

Eşitlik 4.54’da problemin amaç fonksiyonu verilmiştir.

$$f(\vec{x}) = (QP_0/0.7) + E_f \quad (4.54)$$

Problemdeki kısıtlar Eşitlik 4.55’de verilmiştir.

$$\begin{aligned} g_1(\vec{x}) &= 1000 - P_0 \leq 0, \\ g_2(\vec{x}) &= W - 101000 \leq 0, \\ g_3(\vec{x}) &= 5000 - \frac{W}{\pi(R^2 - R_0^2)} \leq 0, \\ g_4(\vec{x}) &= 50 - P_0 \leq 0, \\ g_5(\vec{x}) &= 0.001 - \frac{0.0307}{386.4P_0} \left(\frac{Q}{2\pi Rh} \right) \leq 0, \\ g_6(\vec{x}) &= R - R_0 \leq 0, \\ g_7(\vec{x}) &= h - 0.001 \leq 0. \end{aligned} \quad (4.55)$$

Eşitlik 4.56’de bazı değişkenlere ait eşitliklere yer verilmiştir.

$$\begin{aligned} W &= \frac{\pi P_0}{2} \frac{R^2 - R_0^2}{\ln(R/R_0)}, \quad P_0 = \frac{6\mu Q}{\pi h^3} \ln(R/R_0), \quad E_f = 9336Q \times 0.0307 \times 0.5\Delta T, \\ \Delta T &= 2(10^P - 559.7), \quad P = \frac{\log 10 \log 10 (8.122 \times 10^6 \mu + 0.8) + 3.55}{10.04}, \\ h &= \left(\frac{2\pi \times 750}{60} \right)^2 \frac{2\pi\mu}{E_f} \left(\frac{R^4 - R_0^4}{4} \right) \end{aligned} \quad (4.56)$$

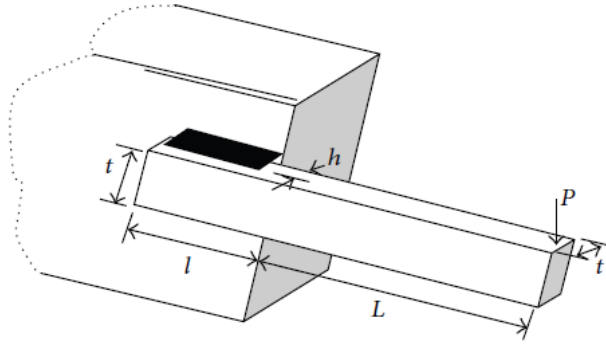
Eşitlik 4.57’de bazı değişkenlerin sınır değerleri verilmiştir.

$$1 \leq R \leq 16, \quad 1 \leq R_0 \leq 16, \quad 1 \leq Q \leq 16, \quad 1 \times 10^{-6} \leq \mu \leq 16 \times 10^{-6} \quad (4.57)$$

4.3.1.3 Kaynaklı Kiriş Tasarımı

Bu problemin temel amacı minimum maliyetle kaynaklı bir kiriş tasarlamaktır. Bu problem beş kısıtlama içerir ve kaynaklı bir kiriş geliştirmek için dört değişken kullanılır. Kirişin

kısıtlamaları, kayma gerilmesi (τ), kirişteki eğilme gerilmesi (θ), çubuk üzerindeki burkulma yükü (P_c), kirişin uç defleksiyonu (δ) ve yan kısıtlamalardır. Bu problemin sırasıyla kaynak kalınlığı (h), çubuğun ekli kısmının uzunluğu (l), çubuğun yüksekliği (t) ve çubuğun kalınlığı (b) olmak üzere dört değişkeni vardır. Bu problem şu şekilde formüle edilir: [85]



Şekil 12. Kaynaklı Kiriş Tasarımı [86].

Eşitlik 4.58’de problemin amaç fonksiyonu verilmiştir.

$$f(\bar{x}) = 0.04811x_3x_4(x_2 + 14) + 1.1047x_1^2x_2 \quad (4.58)$$

Problemdeki kısıtlar Eşitlik 4.59’de verilmiştir.

$$\begin{aligned} g_1(\bar{x}) &= x_1 - x_4 \leq 0, \\ g_2(\bar{x}) &= \delta(\bar{x}) - \delta_{\max} \leq 0, \\ g_3(\bar{x}) &= P \leq P_c(\bar{x}) \\ g_4(\bar{x}) &= \tau_{\max} \geq \tau(\bar{x}), \\ g_5(\bar{x}) &= \sigma(\bar{x}) - \sigma_{\max} \leq 0 \end{aligned} \quad (4.59)$$

Eşitlik 4.60’de bazı değişkenlerin alabileceği değerler ve sabit ifadeleri verilmiştir.

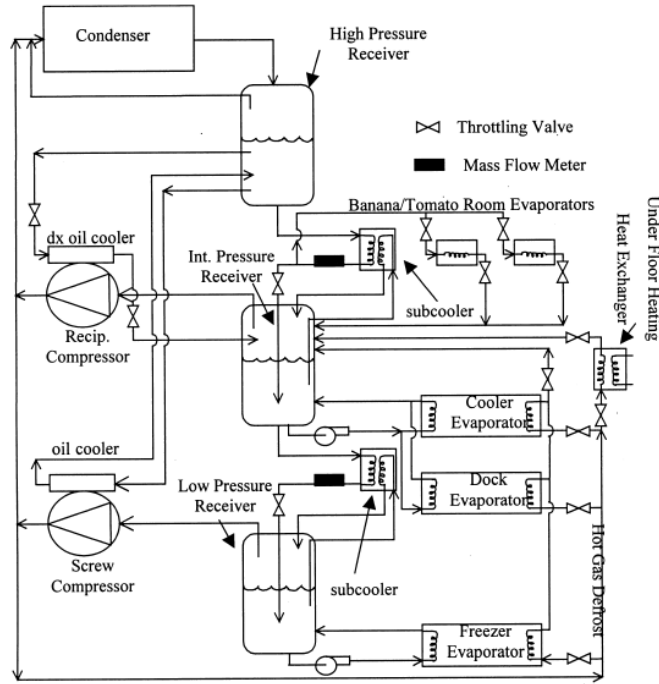
$$\begin{aligned}
\tau &= \sqrt{\tau'^2 + \tau''^2 + 2\tau'\tau'' \frac{x_2}{2R'}}, \quad \tau'' = \frac{RM}{j}, \quad \tau' = \frac{P}{\sqrt{2}x_2x_1}, \quad M = p\left(\frac{x_2}{2} + L\right), \\
R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J = \left(\left(\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\sqrt{2}x_1x_2\right), \quad \sigma(\bar{x}) = \frac{6PL}{x_4x_3^2}, \\
\delta(\bar{x}) &= \frac{6PL^3}{Ex_3^2x_4}, \quad P_c(\bar{x}) = \frac{4.013Ex_3x_4^3}{6L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \\
L &= 14in, \quad P = 6000Ib, \quad E = 30 \times 10^6 psi, \quad \sigma_{\max} = 30000 psi, \\
\tau_{\max} &= 13600 psi, \quad G = 12 \times 10^6 psi, \quad \delta_{\max} = 0.25in.
\end{aligned} \tag{4.60}$$

Eşitlik 4.61'te bazı değişkenlerin sınır değerleri verilmiştir.

$$0.1 \leq x_3, x_2 \leq 10, \quad 0.1 \leq x_4 \leq 2, \quad 0.125 \leq x_1 \leq 2 \tag{4.61}$$

4.3.1.4 Endüstriyel Soğutma Sisteminin Optimal Tasarımı

Bu mühendislik tasarım problemi, 14 tasarım değişkeni ($x_1, x_2, \dots, x_{14}$) ve 15 eşitsizlik tasarım kısıtlaması ($g_1, g_2, \dots, g_{15}$) olan bir endüstriyel soğutma sisteminin optimum tasarımını ele almaktadır. Bu problemin tam matematiksel formülasyonu Andrei [87] tarafından sunulmuştur.



Şekil 13. Endüstriyel Soğutma Sisteminin Optimal Tasarımı [88].

Eşitlik 4.62’te problemin amaç fonksiyonu verilmiştir.

$$\begin{aligned}
f(\bar{x}) = & 63098.88x_2x_4x_{12} + 5441.5x_2^2x_{12} + 115055.5x_2^{1.664} + \\
& 6172.27x_2^2x_6 + 63098.88x_1x_3x_{11} + 5441.5x_1^2x_{11} + 115055.5x_1^{1.664}x_5 + \\
& 6172.27x_1^2x_5 + 140.53x_1x_{11} + 281.29x_3x_{11} + 70.26x_1^2 + 281.29x_1x_3 + \\
& 281.29x_3^2 + 11437x_8^{1.8812}x_{12}^{0.3424}x_{10}^{-1}x_{14}^2x_1^2x_7x_9^{-1} + 20470.2x_7^{2.893}x_{11}^{0.316}x_1^2
\end{aligned} \tag{4.62}$$

Problemdeki kısıtlar Eşitlik 4.63’te verilmiştir.

$$\begin{aligned}
g_1(\bar{x}) &= 1.524x_7^{-1} \leq 1, \\
g_2(\bar{x}) &= 1.524x_8^{-1} \leq 1, \\
g_3(\bar{x}) &= 0.07789x_1 - 2x_7^{-1}x_9 - 1 \leq 0, \\
g_4(\bar{x}) &= 7.05305x_9^{-1}x_1^2x_{10}x_8^{-1}x_2^{-1}x_{14}^{-1} - 1 \leq 0, \\
g_5(\bar{x}) &= 0.0833x_{13}^{-1}x_{14} - 1 \leq 0, \\
g_6(\bar{x}) &= 47.136x_2^{0.333}x_{10}^{-1}x_{12} - 1.333x_8x_{13}^{2.1195} + 62.08x_{13}^{2.1195}x_{12}^{-1}x_8^{0.2}x_{10}^{-1} - 1 \leq 0, \\
g_7(\bar{x}) &= 0.04771x_{10}x_8^{1.8812}x_{12}^{0.3424} - 1 \leq 0, \\
g_8(\bar{x}) &= 0.0488x_9x_7^{1.893}x_{11}^{0.316} - 1 \leq 0, \\
g_9(\bar{x}) &= 0.0099x_1x_3^{-1} - 1 \leq 0, \\
g_{10}(\bar{x}) &= 0.0193x_2x_4^{-1} - 1 \leq 0, \\
g_{11}(\bar{x}) &= 0.0298x_1x_5^{-1} - 1 \leq 0, \\
g_{12}(\bar{x}) &= 0.056x_2x_6^{-1} - 1 \leq 0, \\
g_{13}(\bar{x}) &= 2x_9^{-1} - 1 \leq 0, \\
g_{14}(\bar{x}) &= 2x_{10}^{-1} - 1 \leq 0, \\
g_{15}(\bar{x}) &= x_{12}x_{11}^{-1} - 1 \leq 0
\end{aligned} \tag{4.63}$$

Eşitlik 4.64’da bazı değişkenlerin sınır değerleri verilmiştir.

$$0.001 \leq x_i \leq 5, \quad i = 1, \dots, 14 \tag{4.64}$$

4.3.2 Sentetik Kıyaslama Fonksiyonları

Test fonksiyonları, optimizasyon algoritmalarının performansını doğrulamak ve karşılaştırmak için önemlidir. Literatürde bildirilen birçok test veya kıyaslama işlevi vardır; ancak standart bir liste veya kıyaslama fonksiyon seti yoktur. Bu nedenle, araştırmacılar çalışmalarında farklı konfigürasyonlara sahip fonksiyon kümelerini seçerler. İdeal olarak, test fonksiyonlarının yeni algoritmaları tarafsız bir şekilde test etmek için gerçekten yararlı olması için çeşitli özelliklere sahip olması gerekir.

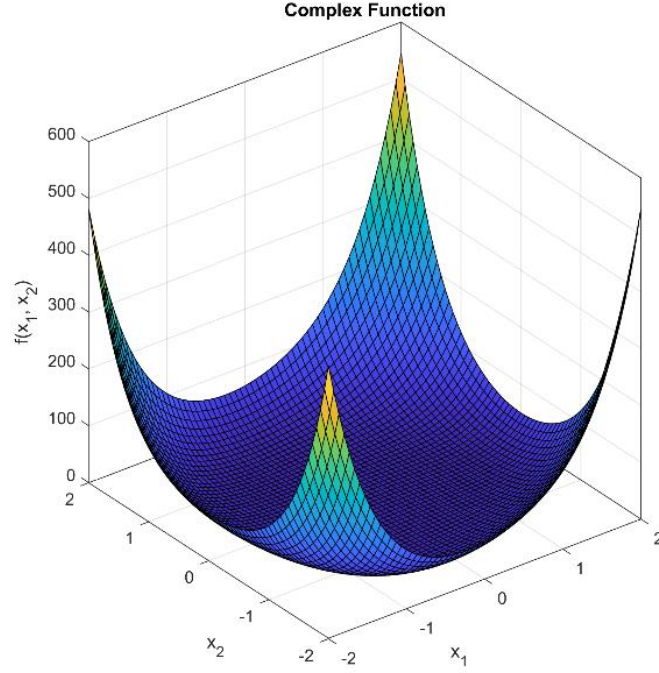
Bu çalışmada, önerdiğimiz algoritmayı test etmek amacıyla Tablo 2'deki gibi tek modlu ve çok modlu olmak üzere 20 fonksiyondan oluşan bir veri seti oluşturulmuştur.

Tablo 2. Sentetik Kıyaslama Fonksiyonları.

#	Sentetik Fonksiyonlar	Boyut	Modalite	Alt Sınır	Üst Sınır	f_{min}
1	Complex	2	Tek Modlu	-2	2	0
2	Matyas	2	Tek Modlu	-10	10	0
3	BentCigar	D	Tek Modlu	-100	100	0
4	Brown	D	Tek Modlu	-1	4	0
5	ChungReynolds	D	Tek Modlu	-100	100	0
6	Weierstrass	D	Çok Modlu	-0,5	0,5	0
7	Quartic	D	Tek Modlu	-500	500	0
8	Sphere	D	Tek Modlu	0	10	0
9	XinSheYang No.1	D	Tek Modlu	-5	5	0
10	XinSheYang No.2	D	Tek Modlu	-2π	2π	0
11	XinSheYang No.3	D	Tek Modlu	-2π	2π	-1
12	Bukin No.4	2	Çok Modlu	-15	5	0
13	CamelThreeHump	2	Çok Modlu	-5	5	0
14	DropWave	2	Çok Modlu	-5.12	5.12	-1
15	Alpine No.1	D	Çok Modlu	-10	10	0
16	Csendes	D	Çok Modlu	-1	1	0
17	Griewank	D	Çok Modlu	-100	100	0
18	Salomon	D	Çok Modlu	-100	100	0
19	WWavy	D	Çok Modlu	$-\pi$	π	0
20	XinSheYang No.4	D	Çok Modlu	-10	10	-1

4.3.2.1 Complex Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-2 \leq x_i \leq 2$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 14'te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



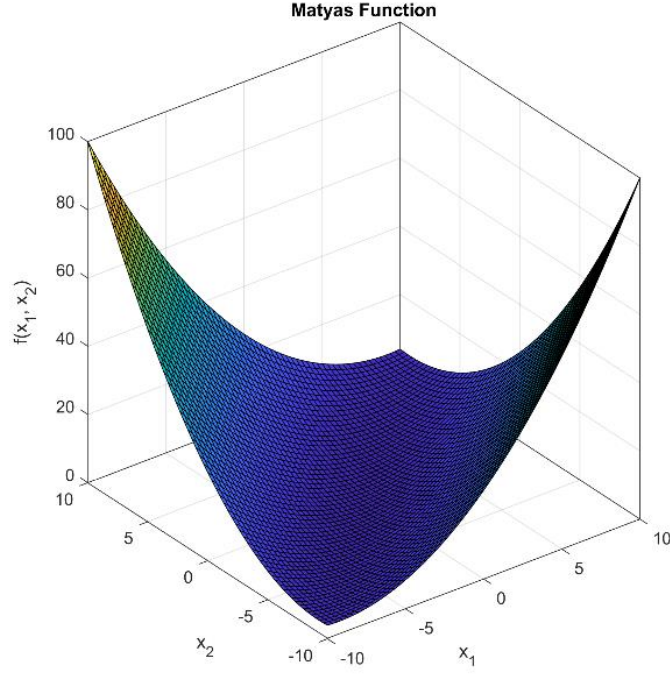
Şekil 14. Complex Fonksiyonunun Karakteristik Grafiği.

Complex fonksiyonun 2-boyutlu formülasyonu Eşitlik 4.65'de verilmiştir.

$$f_1(x) = (x_1^3 - 3x_1x_2^2 - 1)^2 + (3x_2x_1^2 - x_2^3)^2 \quad (4.65)$$

4.3.2.2 Matyas Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-10 \leq x_i \leq 10$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 15'te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



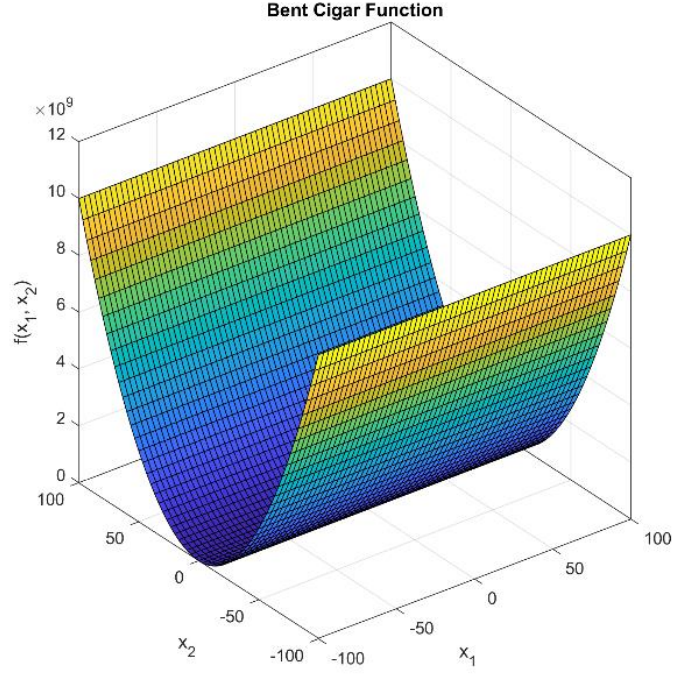
Şekil 15. Matyas Fonksiyonunun Karakteristik Grafiği.

Matyas fonksiyonun 2-boyutlu formülasyonu Eşitlik 4.66’de verilmiştir.

$$f_2(x) = \frac{26}{100}(x_1^2 + x_2^2) - \frac{48}{100}x_1x_2 \quad (4.66)$$

4.3.2.3 BentCigar Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-100 \leq x_i \leq 100$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 16’da fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



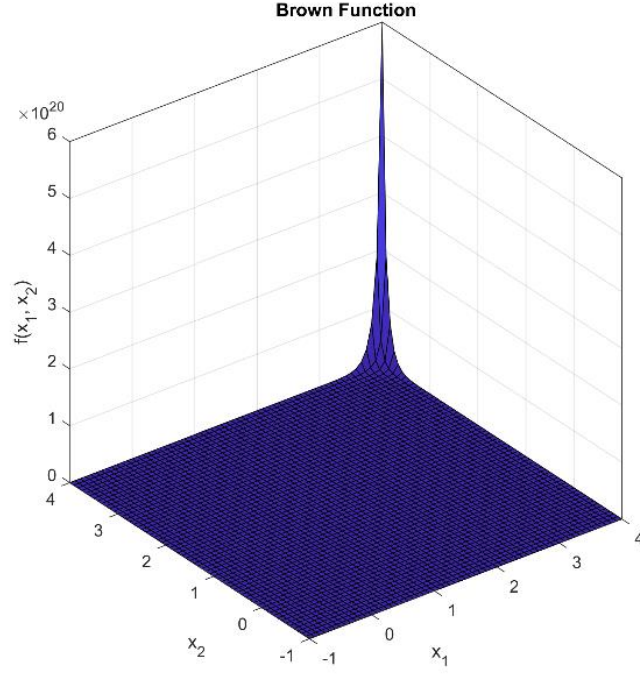
Şekil 16. BentCigar Fonksiyonunun Karakteristik Grafiği.

BentCigar fonksiyonun D -boyutlu formülasyonu Eşitlik 4.67’de verilmiştir.

$$f_3(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2 \quad (4.67)$$

4.3.2.4 Brown Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-1 \leq x_i \leq 4$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 17’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



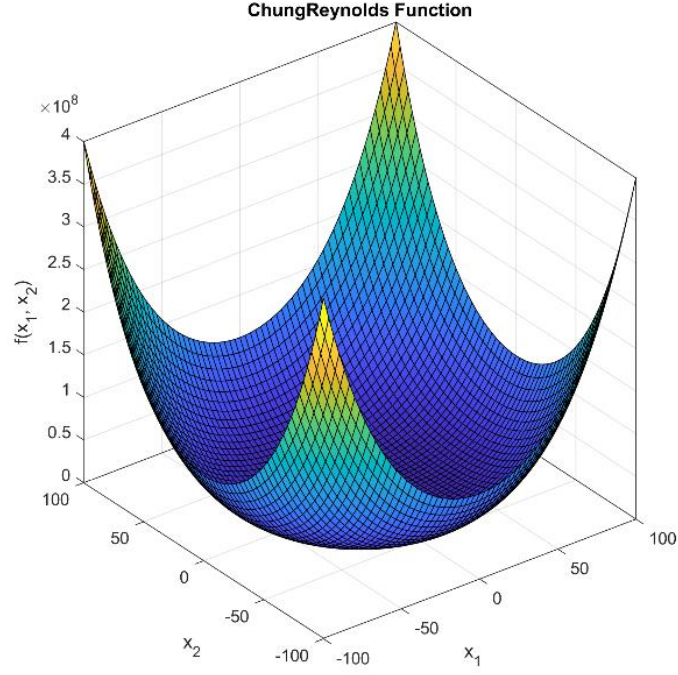
Şekil 17. Brown Fonksiyonunun Karakteristik Grafiği.

Brown fonksiyonun D -boyutlu formülasyonu Eşitlik 4.68’de verilmiştir.

$$f_4(x) = \sum_{i=1}^{D-1} \left(x_i^2 \right)^{x_{i+1}^2 + 1} + \left(x_{i+1}^2 + 1 \right)^{x_i^2 + 1} \quad (4.68)$$

4.3.2.5 ChungReynolds Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-100 \leq x_i \leq 100$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 18’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



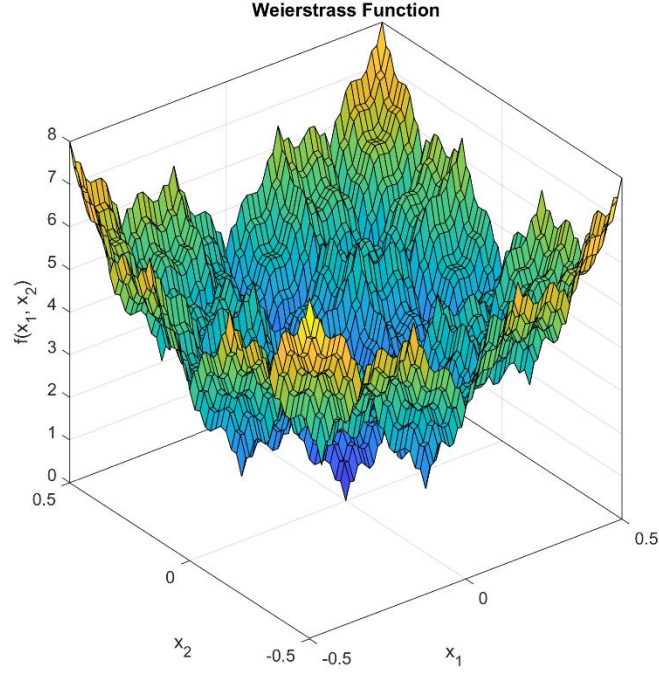
Şekil 18. ChungReynolds Fonksiyonunun Karakteristik Grafiği.

ChungReynolds fonksiyonun D -boyutlu formülasyonu Eşitlik 4.69’de verilmiştir.

$$f_5(x) = \sum_{i=1}^D (x_i^2)^2 \quad (4.69)$$

4.3.2.6 Weierstrass Fonksiyonu

Çok modlu ve D boyutlu bir fonksiyondur. Arama uzayı $-0.5 \leq x_i \leq 0.5$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 19’da fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



Şekil 19. Weierstrass Fonksiyonunun Karakteristik Grafiği.

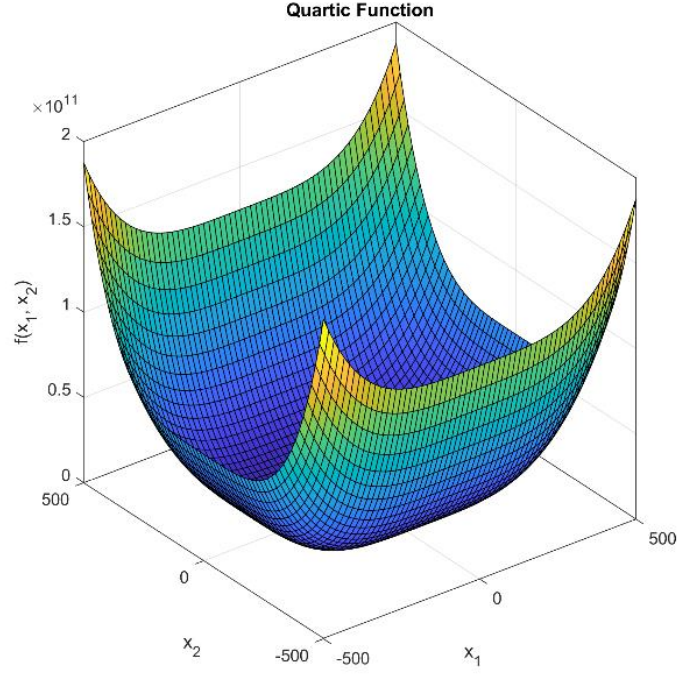
Weierstrass fonksiyonun D -boyutlu formülasyonu Eşitlik 4.70’de verilmiştir.

$$f_6(x) = \sum_{i=2}^D \sum_{k=0}^{k_{\max}} a^k \cos(2\pi b^k (x_i + 0.5)) - D \sum_{k=0}^{k_{\max}} a^k \cos(\pi b^k) \quad (4.70)$$

Burada $a = 0.5$, $b = 3$ ve $k_{\max} = 20$ sabit değer olarak belirlenmiştir.

4.3.2.7 Quartic Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-500 \leq x_i \leq 500$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 20’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



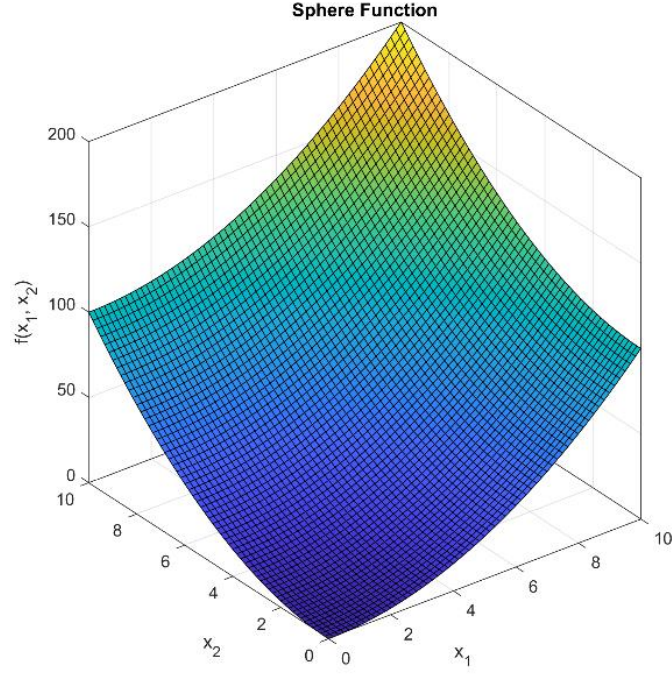
Şekil 20. Quartic Fonksiyonunun Karakteristik Grafiği.

Quartic fonksiyonun D -boyutlu formülasyonu Eşitlik 4.71’de verilmiştir.

$$f_7(x) = \sum_{i=1}^D (i \cdot x_i^4) + \text{random}[0, 1) \quad (4.71)$$

4.3.2.8 Sphere Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $0 \leq x_i \leq 10$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 21’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



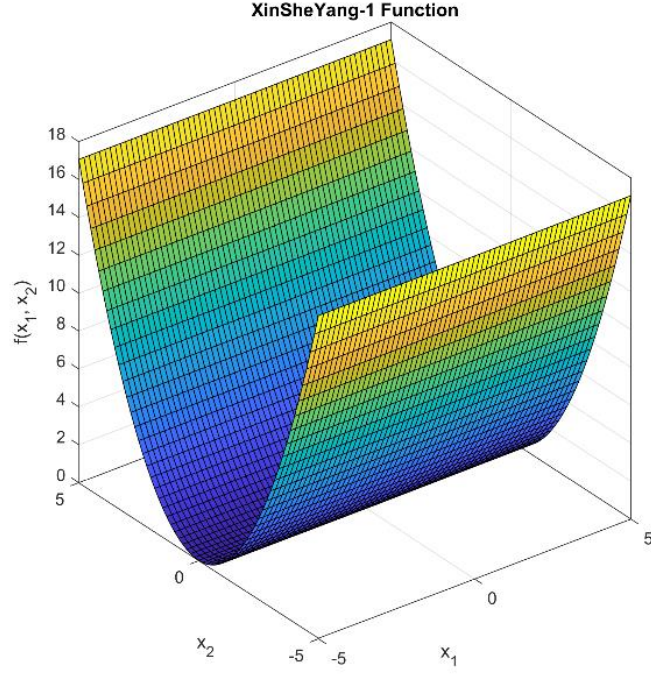
Şekil 21. Sphere Fonksiyonunun Karakteristik Grafiği.

Sphere fonksiyonun D -boyutlu formülasyonu Eşitlik 4.72’de verilmiştir.

$$f_8(x) = \sum_{i=1}^D x_i^2 \quad (4.72)$$

4.3.2.9 XinSheYang No.1 Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-5 \leq x_i \leq 5$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 22’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



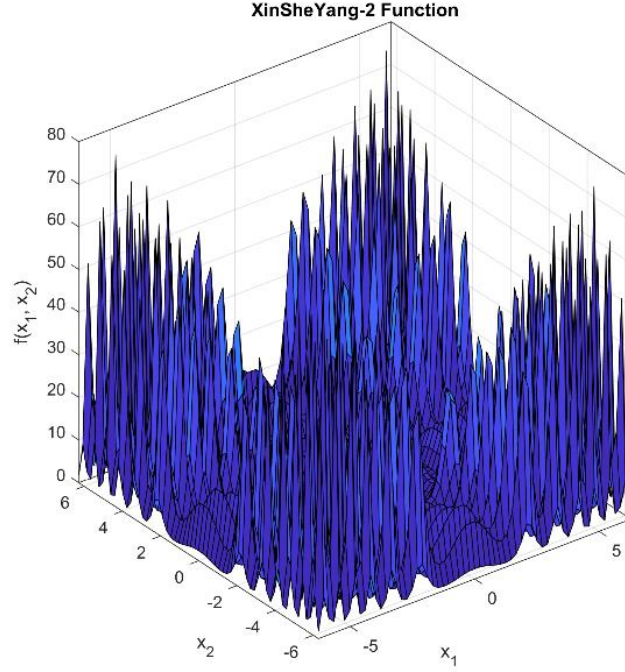
Şekil 22. XinSheYang No. 1 Fonksiyonunun Karakteristik Grafiği.

XinSheYang No.1 fonksiyonun D -boyutlu formülasyonu Eşitlik 4.73’de verilmiştir.

$$f_9(x) = \sum_{i=1}^D |x_i|^i \text{ random}[0, 1) \quad (4.73)$$

4.3.2.10 XinSheYang No.2Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-2\pi \leq x_i \leq 2\pi$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 23’te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



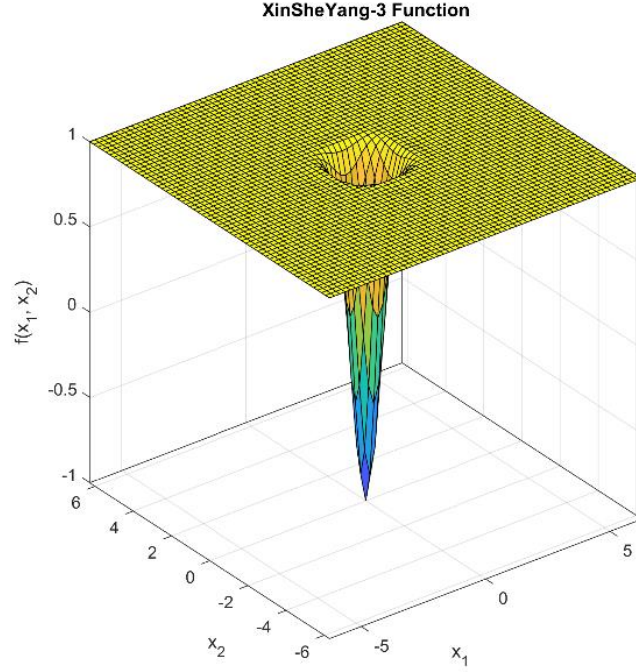
Şekil 23. XinSheYang No. 2 Fonksiyonunun Karakteristik Grafiği.

XinSheYang No.2 fonksiyonun D -boyutlu formülasyonu Eşitlik 4.74’de verilmiştir.

$$f_{10}(x) = \left(\sum_{i=1}^D |x_i| \right) \exp \left(- \sum_{i=1}^D \sin(x_i^2) \right) \quad (4.74)$$

4.3.2.11 XinSheYang No.3Fonksiyonu

Tek modlu bir fonksiyondur. Arama uzayı $-2\pi \leq x_i \leq 2\pi$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = -1$ olarak bulunur. Şekil 24’te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



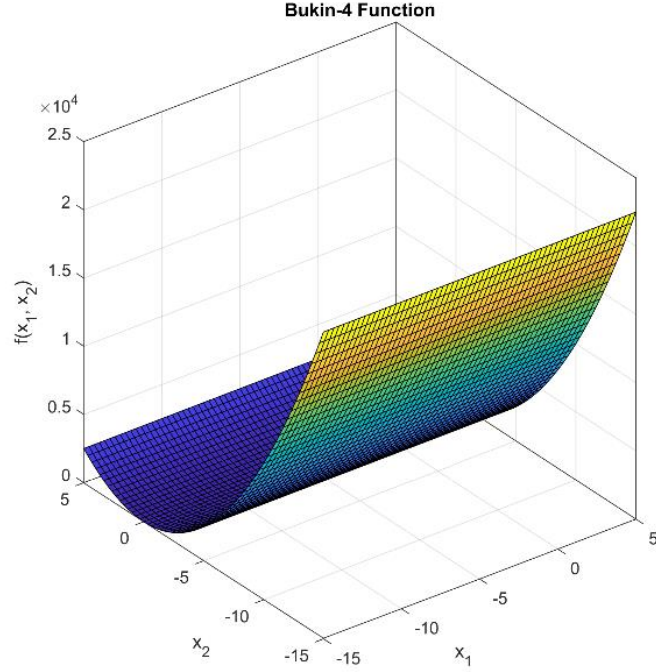
Şekil 24. XinSheYang No. 3 Fonksiyonunun Karakteristik Grafiği.

XinSheYang No.3 fonksiyonun D -boyutlu formülasyonu Eşitlik 4.75’de verilmiştir.

$$f_{11}(x) = e^{-\sum_{i=1}^D (x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^D x_i^2} \cdot \prod_{i=1}^D \cos^2 x_i \quad (4.75)$$

4.3.2.12 *Bukin No.4Fonksiyonu*

Çok modlu bir fonksiyondur. Arama uzayı $-15 \leq x_i \leq 5$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 25’te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



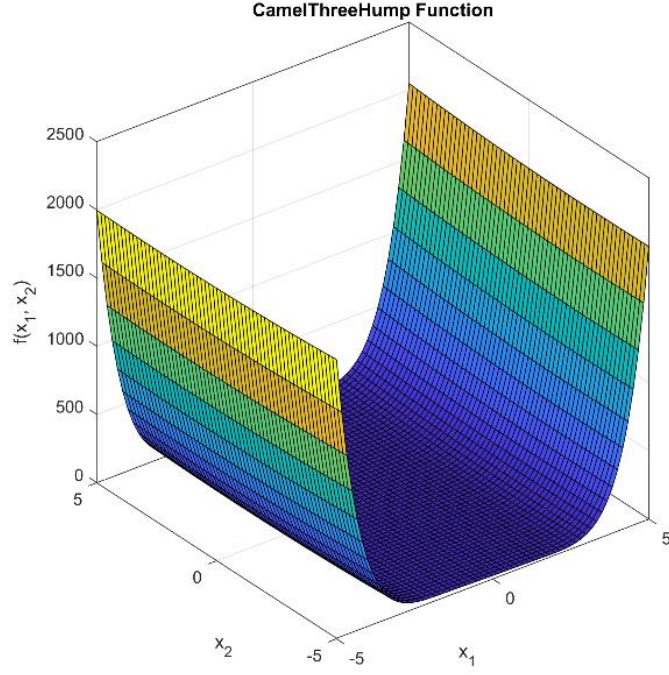
Şekil 25. Bukin No. 4 Fonksiyonunun Karakteristik Grafiği.

Bukin No.4 fonksiyonun 2-boyutlu formülasyonu Eşitlik 4.76’de verilmiştir.

$$f_{12}(x) = 100x_2^2 + 0.01\|x_1 + 10\| \quad (4.76)$$

4.3.2.13 CamelThreeHump Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-5 \leq x_i \leq 5$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 26’da fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



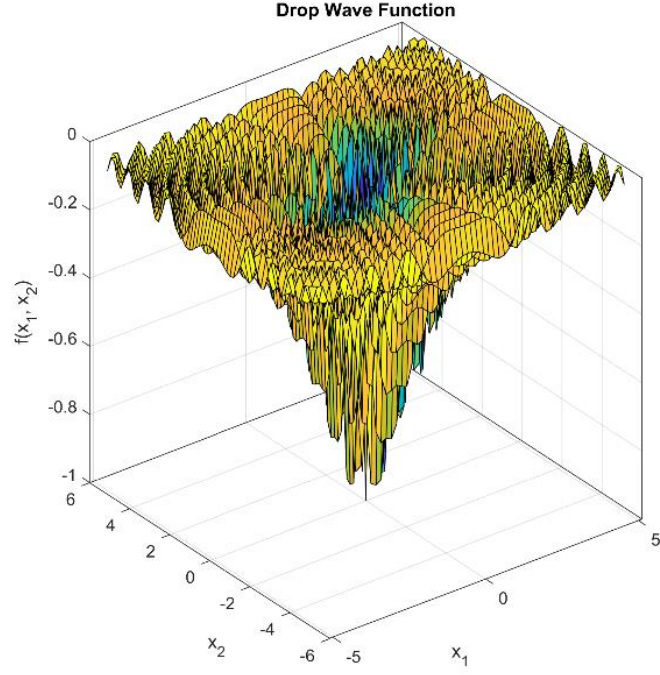
Şekil 26. CamelThreeHump Fonksiyonunun Karakteristik Grafiği.

CamelThreeHump fonksiyonun 2-boyutlu formülasyonu Eşitlik 4.77’de verilmiştir.

$$f_{13}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2 \quad (4.77)$$

4.3.2.14 DropWave Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-5.12 \leq x_i \leq 5.12$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = -1$ olarak bulunur. Şekil 27’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



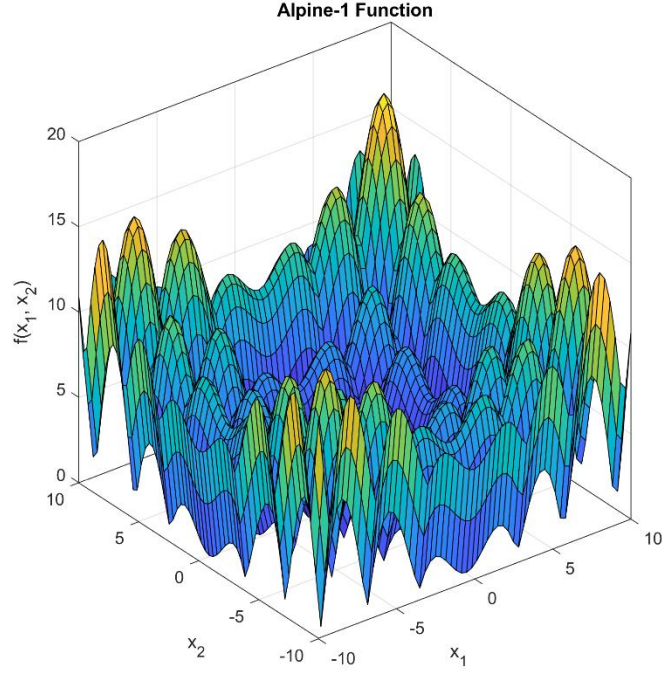
Şekil 27. DropWave Fonksiyonunun Karakteristik Grafiği.

DropWave fonksiyonun 2-boyutlu formülasyonu Eşitlik 4.78’de verilmiştir.

$$f_{14}(x) = -\left(1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)\right) / \left(0.5(x_1^2 + x_2^2) + 2\right) \quad (4.78)$$

4.3.2.15 Alpine No.1 Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-10 \leq x_i \leq 10$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 28’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



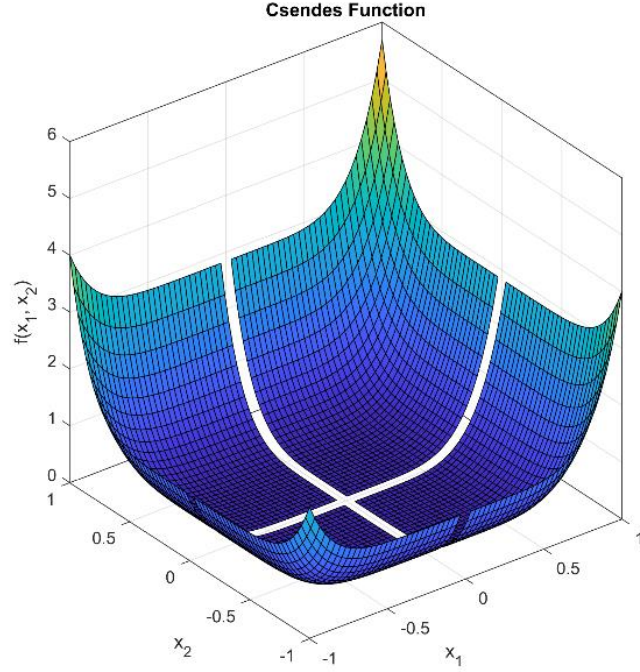
Şekil 28. Alpine No. 1 Fonksiyonunun Karakteristik Grafiği.

Alpine No.1 fonksiyonun D -boyutlu formülasyonu Eşitlik 4.79’de verilmiştir.

$$f_{15}(x) = \sum_{i=1}^D \|x_i \sin(x_i) + 0.1x_i\| \quad (4.79)$$

4.3.2.16 Csendes Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-1 \leq x_i \leq 1$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 29’da fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



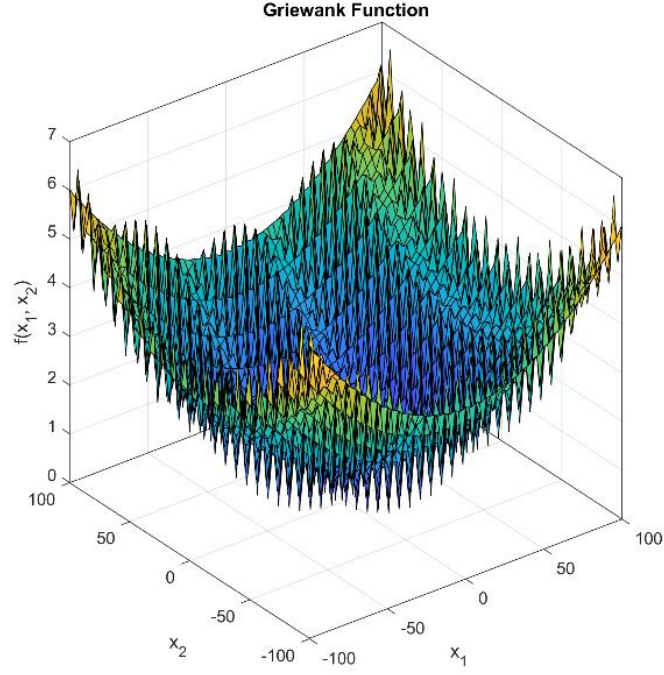
Şekil 29. Csendes Fonksiyonunun Karakteristik Grafiği.

Csendes fonksiyonun D -boyutlu formülasyonu Eşitlik 4.80’de verilmiştir.

$$f_{16}(x) = \sum_{i=1}^D x_i^2 \left(2 + \sin \frac{1}{x_i} \right) \quad (4.80)$$

4.3.2.17 Griewank Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-100 \leq x_i \leq 100$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 30’da fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



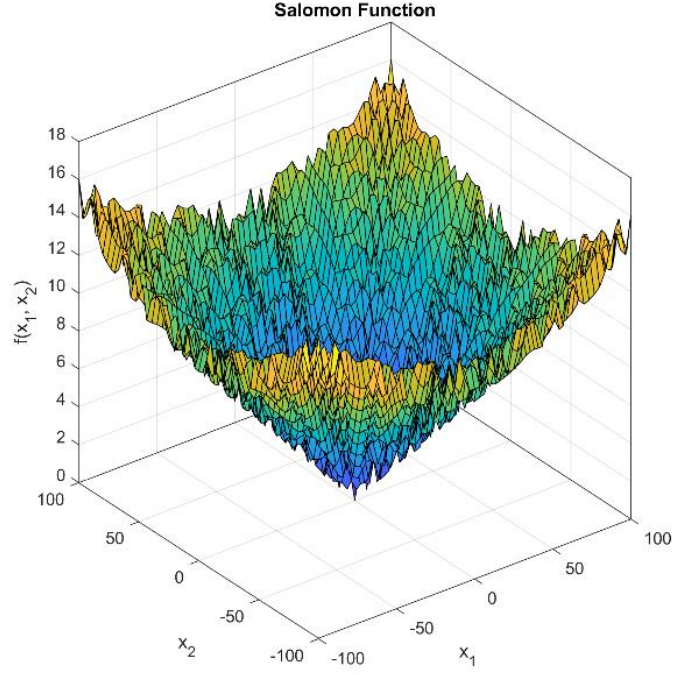
Şekil 30. Griewank Fonksiyonunun Karakteristik Grafiği.

Griewank fonksiyonun D -boyutlu formülasyonu Eşitlik 4.81’de verilmiştir.

$$f_{17}(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (4.81)$$

4.3.2.18 Salomon Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-100 \leq x_i \leq 100$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 31’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



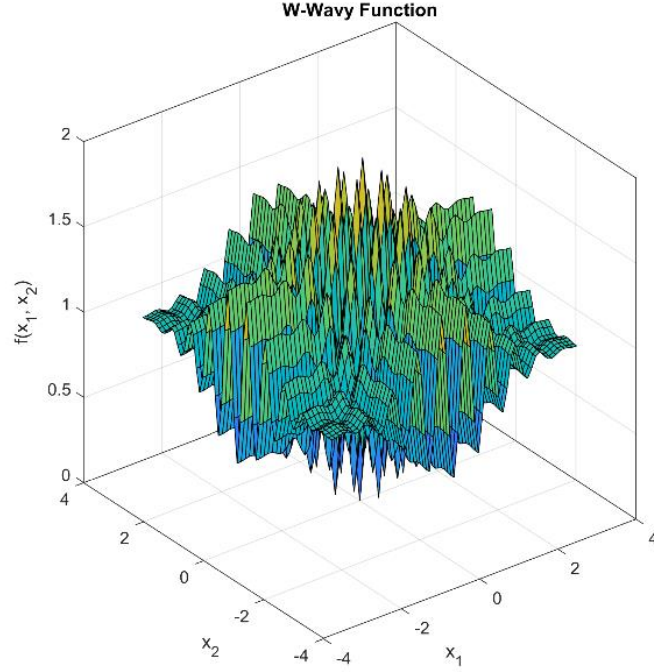
Şekil 31. Salomon Fonksiyonunun Karakteristik Grafiği.

Salomon fonksiyonun D -boyutlu formülasyonu Eşitlik 4.82’de verilmiştir.

$$f_{18}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^D x_i^2} \quad (4.82)$$

4.3.2.19 WWavy Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-\pi \leq x_i \leq \pi$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = 0$ olarak bulunur. Şekil 32’de fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



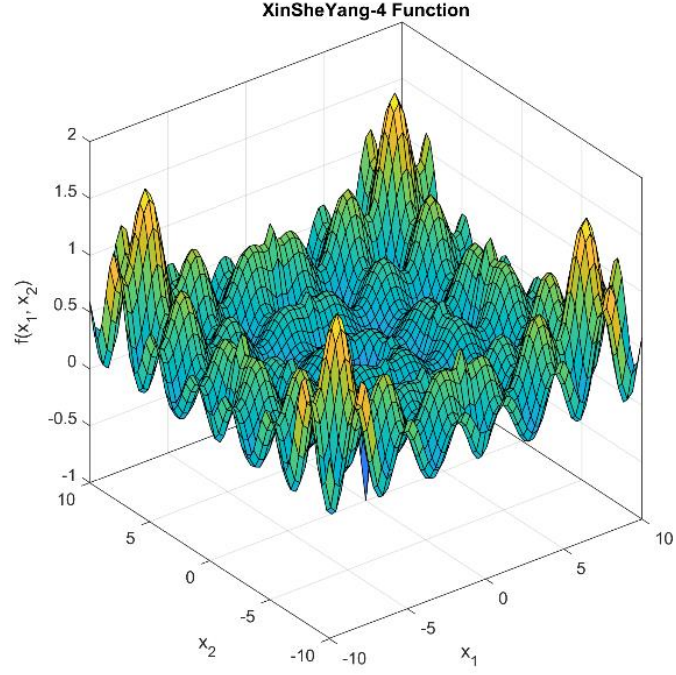
Şekil 32. WWavy Fonksiyonunun Karakteristik Grafiği.

WWavy fonksiyonun D -boyutlu formülasyonu Eşitlik 4.83’de verilmiştir.

$$f_{19}(x) = 1 - D^{-1} \sum_{i=1}^D \cos(10x_i) \exp\left(-\frac{x_i^2}{2}\right) \quad (4.83)$$

4.3.2.20 XinSheYang No.4Fonksiyonu

Çok modlu bir fonksiyondur. Arama uzayı $-10 \leq x_i \leq 10$ aralığında olup, global minimum tüm karar değişkenlerinin sıfır olduğu durumda $f(0, \dots, 0) = -1$ olarak bulunur. Şekil 33’te fonksiyonun iki boyutlu formundaki karakteristik grafiği gösterilmektedir.



Şekil 33. XinSheYang No. 4 Fonksiyonunun Karakteristik Grafiği.

XinSheYang No.4 fonksiyonun D -boyutlu formülasyonu Eşitlik 4.84’de verilmiştir.

$$f_{20}(x) = e^{-\sum_{i=1}^D \left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^D (x_i - \pi)^2} \cdot \prod_{i=1}^D \cos^2 x_i \quad (4.84)$$

5. DENEYSEL ÇALIŞMALAR

Bazı optimizasyon algoritmaları, en iyi performansı elde etmek için önceden ayarlanması gereken parametreler içerir. Optimizasyon algoritmalarının arama uzayında doğru yönün belirlenmesinde ve global optimalin elde edilmesinde parametre belirleme önemli bir konudur. Bu nedenle deneysel çalışmalarda kullanılan optimizasyon algoritmalarının bu parametreleri ayarlanırken varsayılan değerler (geliştiriciler tarafından önerilen veya yaygın olarak tercih edilen) kullanılmıştır: PSO için atalet ağırlığı aralığı ve öğrenme faktörleri ($c1$ ve $c2$) sırasıyla [0.2, 0.9], 0.2 ve 0.2 olarak ayarlanmıştır. AA için ışık soğurma katsayısı (γ), cazibe katsayısı (β), alfa ve teta parametreleri sırasıyla 1.0, 2.0, 0.2 ve 0.97 olarak belirlenmiştir. GKAA için yabancı yumurtaların keşif oranı 0.25 olarak belirlenmiştir. İYOOA için minimum ve maksimum tohum sayıları sırasıyla 0 ve 5'tir. Standart sapmanın başlangıç değeri 0.5 olup, nihai değeri ise 0.001 olarak belirlenmiştir. KSA'da ise ışık oranı ve bağıl nem oranı sırasıyla 0.9 ve 0.5 olarak belirlenmiştir.

5.1 PERFORMANS ÖLÇÜM KRİTERLERİ

Deneysel çalışmalarda optimizasyon algoritmalarının performans ölçüm kriterleri olarak minimum, maksimum, standart sapma ve ortalama uygunluk değerleri kullanılmıştır. Diğer bir ölçüm kriteri olarak da algoritmaların çalışma anında harcadıkları süre esas alınır. Çalışma zamanı analiz bilgisayar sisteminin özellikleri şu şekildedir: 64 bit işletim sistemi (Windows 10), Intel i7 3.6 GHz CPU, 8-GB sistem belleği. Deneysel çalışmalarla elde edilen optimizasyon algoritmalarının sonuçları arasında anlamlı bir ilişki olup olmadığını ölçmek için hipotez testi olarak Wilcoxon işaretli sıralar testi kullanılmıştır.

5.2 SENTETİK KIYASLAMA FONKSİYON TESTLERİ

Sentetik kıyaslama deneyleri, farklı popülasyon büyüklüğü ve problem boyutlarına bağlı olarak birkaç aşamadan meydana gelmektedir.

5.2.1 Deney 1: Tek Modlu Kıyaslama Fonksiyonları

Bu deneyde, tek modlu fonksiyonlar test edilmiştir. Yineleme sayısı 100, popülasyon sayısı 30 ve D -boyutlu fonksiyonlar için boyut değeri 10 olarak belirlenmiştir. Yapılan bu deneyde, her bir problem için algoritmalar 30 kez tekrarlı olarak çalıştırılmış ve elde edilen skorların ortalama değerleri alınmıştır. Tablo 3'te tek modlu sentetik kıyaslama fonksiyonları ve bunların özellikleri verilmiştir.

Tablo 3. Tek modlu sentetik kıyaslama fonksiyonları.

#	Problem	Boyut	Modalite	Alt Sınır	Üst Sınır	Global Skor
1	BentCigar	10	Unimodal	-100	100	0
2	Brown	10	Unimodal	-1	4	0
3	ChungReynolds	10	Unimodal	-100	100	0
4	Complex	2	Unimodal	-2	2	0
5	Matyas	2	Unimodal	-10	10	0
6	Sphere	10	Unimodal	0	10	0
7	Quartic	10	Unimodal	-500	500	0
8	XinSheYang1	10	Unimodal	-5	5	0
9	XinSheYang2	10	Unimodal	-2π	2π	0
10	XinSheYang3	10	Unimodal	-2π	2π	-1

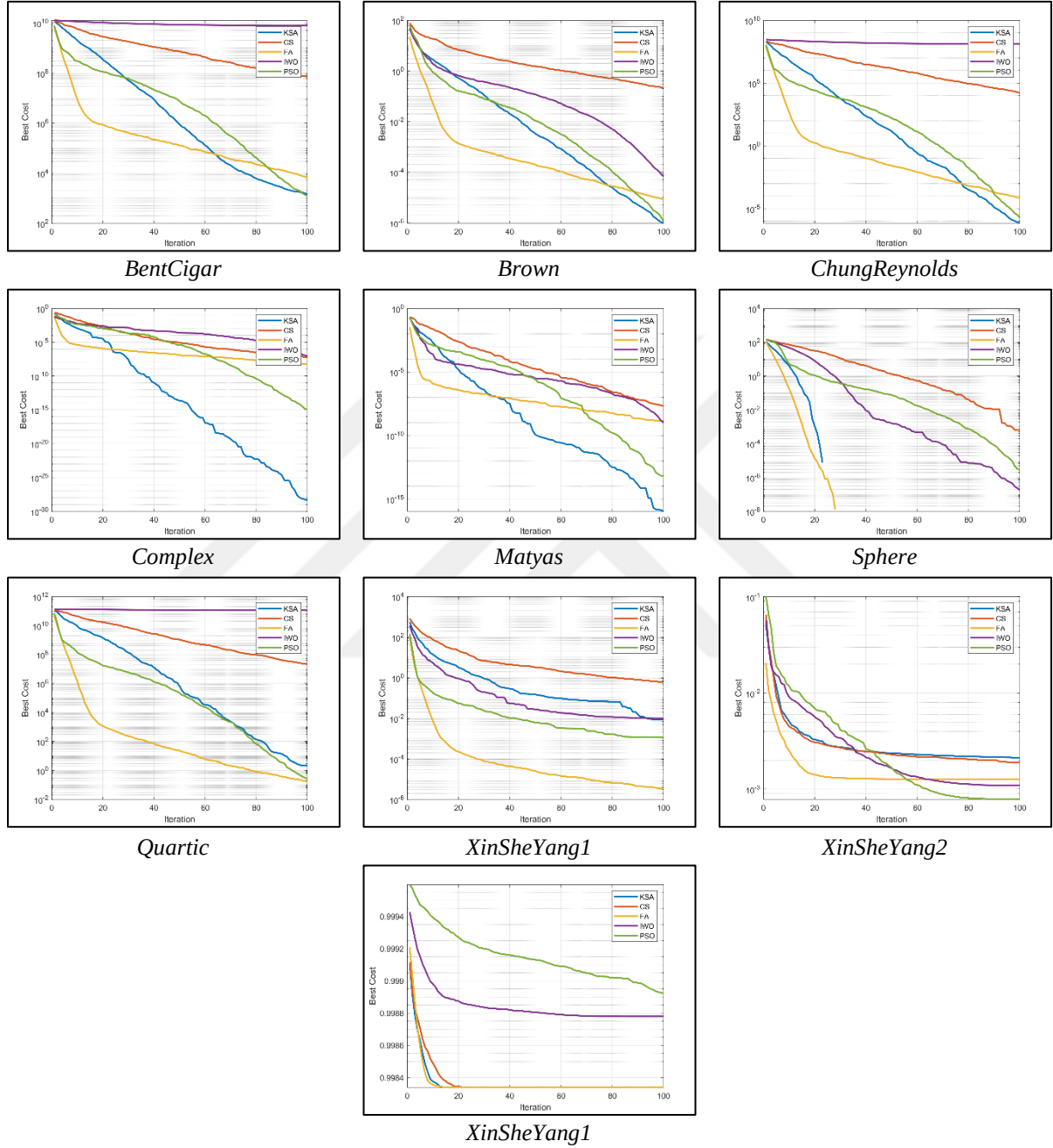
Tablo 4'te, tek modlu sentetik kıyaslama fonksiyonlarının global en iyi skorları verilmiştir. On adet fonksiyonun beş farklı algoritma ile test edilmesi sonucunda önerilen KSA, oldukça başarılı sonuçlar elde etmiştir. KSA'nın, fonksiyonların çoğunda en iyi skoru aldığı gözlemlenmiştir.

Tablo 4. Global-best skorları.

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	BentCigar	1498,869736	72721101,11	6850,769424	8042108742	1237,721971
2	Brown	9,64279E-07	0,225007623	8,56519E-06	6,78504E-05	1,11439E-06
3	ChungReynolds	6,78447E-07	16960,82951	6,95986E-05	132268832,9	1,8763E-06
4	Complex	4,33608E-29	5,74578E-08	5,58823E-09	8,96566E-08	1,15236E-15
5	Matyas	1,06368E-16	2,07879E-08	1,33129E-09	1,14976E-09	6,37652E-14
6	Sphere	0	0,000618145	0	1,80503E-07	2,66007E-06
7	Quartic	2,090911603	21142137,94	0,172492492	1,14952E+11	0,261543931
8	XinSheYang1	0,008540307	0,615242837	3,41371E-06	0,009821378	0,001181908
9	XinSheYang2	0,002123041	0,001890823	0,001262908	0,001092301	0,000781964
10	XinSheYang3	0,998338406	0,998338406	0,998338406	0,998779927	0,998925782

Şekil 34'te, Deney 1'e göre sentetik tek modlu fonksiyonlardaki yakınsama grafikleri gösterilmiştir. Grafiklerde görüldüğü gibi KSA, test edilen çoğu fonksiyonda en iyi sonuca

ulaşmıştır. AA, genel olarak daha düşük iterasyon sayısında sonuca yaklaşmaktadır. Deneyde KSA mavi renk ile GKAA turuncu renk ile AA sarı renk ile İYOOA mor renk ile PSO yeşil renk ile gösterilmiştir.



Şekil 34. Deney 1'e göre tek modlu fonksiyonlardaki yakınsama grafikleri.

Tablo 5'te kullandığımız beş metasezgisel algoritmanın çalışma süreleri saniye cinsinden verilmiştir. KSA'nın diğer dört algortmadan genelde daha hızlı çalıştığı Deney 1'de tespit edilmiştir. Sadece iki fonksiyonda, PSO'nun daha iyi sonuç aldığı gözlemlenmiştir.

Tablo 5. Deney 1'e Göre Algoritmaların Çalışma Süreleri (sn).

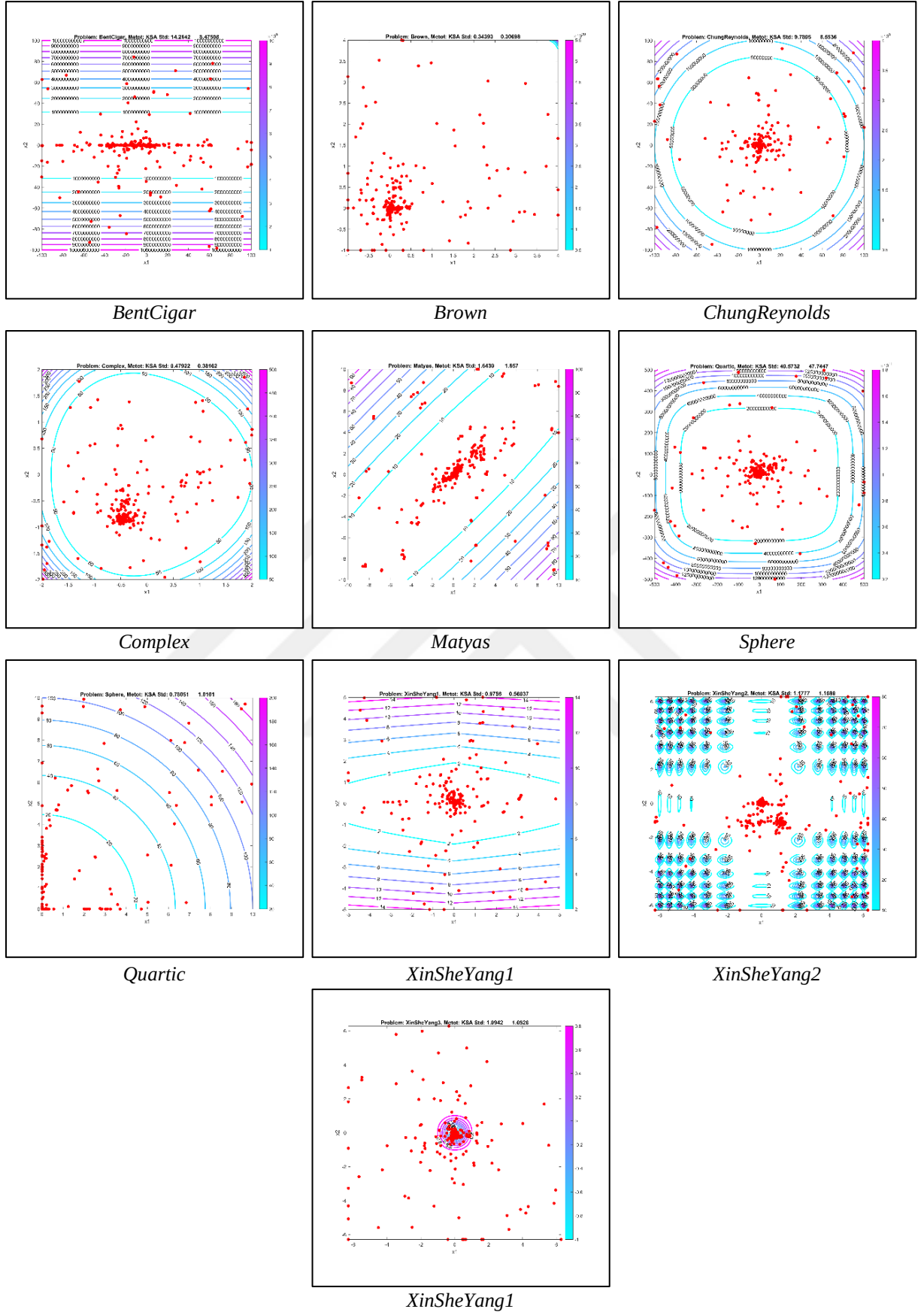
#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	BentCigar	87,1218	120,1354	1705,0844	109,0248	87,4336
2	Brown	95,1833	138,1450	1819,4686	129,3838	94,9854
3	ChungReynolds	87,1227	123,5028	1701,8452	111,3440	83,8779
4	Complex	84,0763	114,8132	1730,5632	167,1915	84,4107
5	Matyas	85,3749	117,6706	1716,0846	158,7059	81,1957
6	Sphere	82,5404	129,2227	1379,4865	138,1272	87,2474
7	Quartic	90,3337	172,6642	1789,8283	118,6180	177,9180
8	XinSheYang1	58,1271	85,5831	949,6643	143,6396	91,6088
9	XinSheYang2	53,7000	78,9675	911,2978	138,3260	90,5610
10	XinSheYang3	66,6912	107,9609	939,6024	168,5951	103,6982

Önerilen yöntemin sonuçlarının, deneysel çalışmalarda kullanılan diğer algoritmalar ile benzerliklerini mukayese etmek amacıyla Tablo 6'daki gibi Wilcoxon işaretli sıralar testi sunulmuştur. Bu testteki güven aralığı %95 nispetinde olup, KSA'nın diğer yöntemlerle arasında genel olarak anlamlı bir ilişki bulunmadığı görülmektedir. Yine bu tabloya göre, KSA'nın davranışı, diğer meta-sezgisellere oranla AA ve PSO algoritmalarına daha yakın seyretmekte olduğu göze çarpmaktadır.

Tablo 6. Wilcoxon işaretli sıralar testi (*p*-değerleri).

#	Problem	KSA-GKAA	KSA-AA	KSA- İYOOA	KSA-PSO
1	BentCigar	3,268E-13	0,251960988	3,66435E-27	0,339768666
2	Brown	3,36753E-14	0,115492078	0,001230284	0,303781836
3	ChungReynolds	3,0621E-14	0,18019876	1,49807E-30	0,264080241
4	Complex	2,94677E-11	2,19784E-08	3,34916E-14	7,05854E-08
5	Matyas	8,2311E-09	0,00485054	1,51275E-07	0,000121828
6	Sphere	0,002893419	0,557905125	0,003366351	1,20275E-07
7	Quartic	5,07174E-08	7,83061E-06	3,07957E-16	0,368520001
8	XinSheYang1	1,89203E-05	8,0176E-07	0,114993352	0,011012462
9	XinSheYang2	0,082385961	0,055155705	0,048452562	0,096906066
10	XinSheYang3	0,362672439	0,461704596	9,33972E-31	4,39211E-35

KSA'nın tek modlu fonksiyonlardaki arama geçmişini (*search history*) gösteren kontur grafikleri Şekil 35'te verilmiştir. Bu aşamada, tek modlu fonksiyonların tümü 2-boyutlu olarak ele alınmıştır. Arama geçmişi, 100 yinelemedeki tüm popülasyonları içermektedir ve her bir grafikte toplamda 300 arama noktası yer almaktadır.



Şekil 35. KSA'nın tek modlu fonksiyonlardaki arama geçmişi.

5.2.2 Deney 2: Çok Modlu Kıyaslama Fonksiyonları

Bu deneyde, çok modlu fonksiyonlar test edilmiştir. Yineleme sayısı 100, popülasyon sayısı 30 ve D -boyutlu fonksiyonlar için boyut değeri 10 olarak belirlenmiştir. Yapılan bu deneyde, her bir problem için algoritmalar 30 kez tekrarlı olarak çalıştırılmış ve elde edilen skorların ortalama değerleri alınmıştır. Tablo 7’de tek modlu sentetik kıyaslama fonksiyonları ve bunların özellikleri verilmiştir.

Tablo 7. Çok modlu sentetik kıyaslama fonksiyonları.

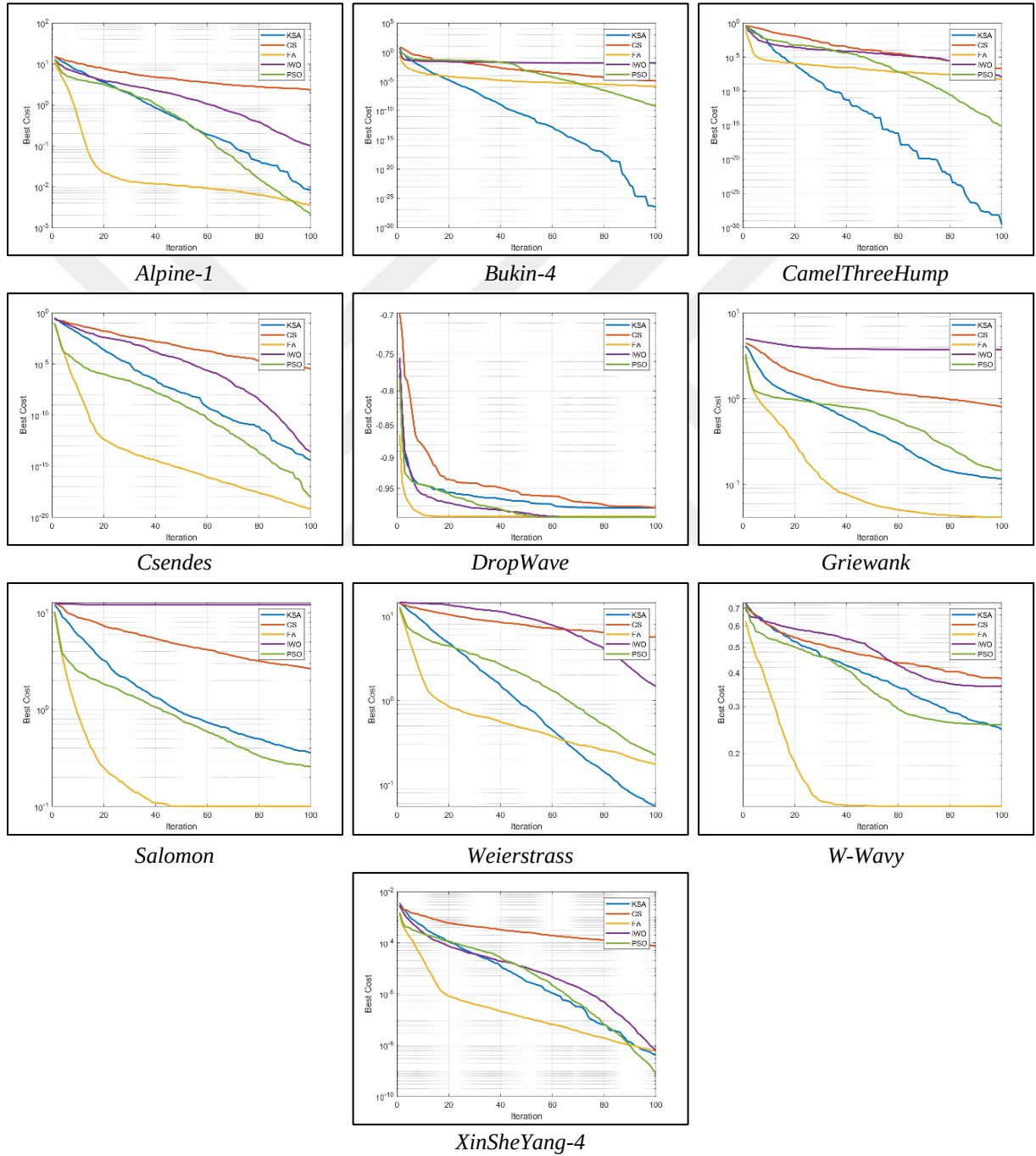
#	Problem	Boyut	Modalite	Alt Sınır	Üst Sınır	Global Skor
1	Alpine1	10	Multimodal	-10	10	0
2	Bukin4	2	Multimodal	-15	5	0
3	CamelThreeHump	2	Multimodal	-5	5	0
4	Csendes	10	Multimodal	-1	1	0
5	Weierstrass	10	Multimodal	-0,5	0,5	0
6	DropWave	2	Multimodal	-5,12	5,12	-1
7	Griewank	10	Multimodal	-100	100	0
8	Salomon	10	Multimodal	-100	100	0
9	W-Wavy	10	Multimodal	$-\pi$	π	0
10	XinSheYang4	10	Multimodal	-10	10	-1

Tablo 8’de, çok modlu sentetik kıyaslama fonksiyonlarının Deney 2’ye göre global en iyi skorları verilmiştir. On adet fonksiyonun beş farklı algoritma ile test edilmesi sonucunda, önerilen KSA oldukça başarılı ve rekabetçi sonuçlar elde etmiştir. KSA’nın, bazı fonksiyonlarda en iyi skoru aldığı gözlemlenmiştir.

Tablo 8. Global-best skorları.

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	Alpine1	0,008230198	2,35176964	0,003468096	0,101349845	0,00211734
2	Bukin4	3,29483E-27	1,18346E-05	1,41927E-06	0,013836031	6,49107E-10
3	CamelThreeHump	2,70258E-30	2,07491E-07	5,86874E-09	1,06133E-08	8,04397E-16
4	Csendes	3,95454E-15	3,06615E-06	6,5926E-20	2,40088E-14	9,33774E-19
5	Weierstrass	0,055253382	5,662615259	0,175848195	1,478048105	0,224893225
6	DropWave	-0,982998575	-0,983036285	-0,997874703	-0,999999698	-1
7	Griewank	0,117172787	0,804393165	0,041290126	3,737071501	0,145211147
8	Salomon	0,355905882	2,662710028	0,099873391	12,04320668	0,257669497
9	W-Wavy	0,245947006	0,382016138	0,126450428	0,356756469	0,25655632
10	XinSheYang4	4,17068E-09	7,68693E-05	5,86387E-09	5,98287E-09	8,5889E-10

Şekil 36’da, Deney 2’ye göre sentetik çok modlu fonksiyonlardaki yakınsama grafikleri gösterilmiştir. Grafiklerde görüldüğü gibi KSA, test edilen fonksiyonların bazılarında en iyi sonuca ulaşmıştır. AA, genel olarak daha düşük iterasyon sayısında sonuca yaklaşmaktadır ve genelde en iyi sonuçlara ulaşmaktadır. Deneyde KSA mavi renk ile GKAA turuncu renk ile AA sarı renk ile İYOOA mor renk ile PSO yeşil renk ile gösterilmiştir.



Şekil 36. Deney 2’ye göre çok modlu fonksiyonlardaki yakınsama grafikleri.

Tablo 9’da kullandığımız beş metasezgisel algoritmanın çalışma süreleri saniye cinsinden verilmiştir. KSA’nın, diğer dört algoritmaya göre fonksiyonların hepsinde daha hızlı çalıştığı Deney 2’de tespit edilmiştir.

Tablo 9. Deney 2’ye Göre Algoritmaların Çalışma Süreleri (sn).

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	Alpine1	14,4862	24,2842	215,3492	16,5457	22,5463
2	Bukin4	12,7543	22,3687	203,5196	21,6628	21,2541
3	CamelThreeHump	12,8358	22,9058	209,3637	25,2221	21,7102
4	Csendes	15,6736	30,7574	253,6213	28,9606	25,2080
5	Weierstrass	158,1626	309,0145	2273,5685	173,6270	166,1933
6	DropWave	11,8293	19,6732	199,9084	39,8239	20,7006
7	Griewank	13,0874	26,2074	214,1398	39,9512	22,1489
8	Salomon	14,4383	22,5735	208,8179	61,3362	21,6001
9	W-Wavy	14,9929	25,0853	222,7638	36,3335	22,4165
10	XinSheYang4	17,4519	28,2060	249,1848	41,9721	24,0238

Önerilen yöntemin sonuçlarının, deneysel çalışmalarda kullanılan diğer algoritmalar ile benzerliklerini mukayese etmek amacıyla Tablo 10’daki gibi Wilcoxon işaretli sıralar testi sunulmuştur. Bu testteki güven aralığı %95 nispetinde olup, KSA’nın diğer yöntemlerle arasında genel olarak anlamlı bir ilişki bulunmadığı görülmektedir. Yine bu tabloya göre, KSA’nın davranışı, diğer meta-sezgisellere oranla PSO algoritmasına daha yakın seyretmekte olduğu göze çarpmaktadır.

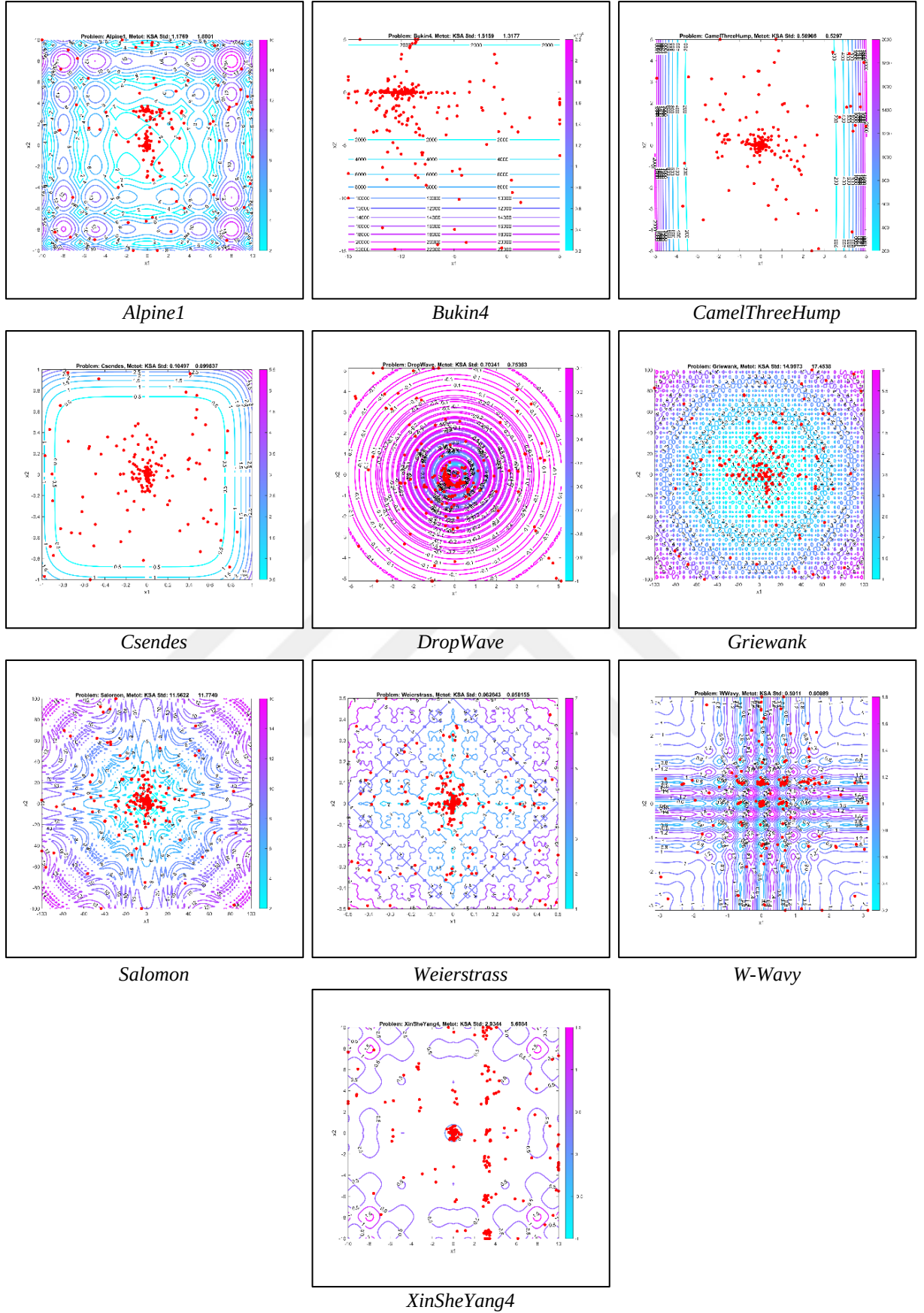
Tablo 10. Wilcoxon işaretli sıralar testi (p -değerleri).

#	Problem	KSA-GKAA	KSA-AA	KSA- İYOOA	KSA-PSO
1	Alpine1	2,85519E-09	0,001732982	0,0170745	0,219082792
2	Bukin4	1,01107E-17	3,56655E-11	3,3098E-17	1,29584E-14
3	CamelThreeHump	4,23377E-19	1,6781E-09	8,23672E-17	1,35778E-09
4	Csendes	8,93361E-12	4,58765E-10	0,000422227	0,212151749
5	Weierstrass	4,46198E-15	0,325692523	1,49152E-13	0,10139873
6	DropWave	0,048977212	0,045012024	0,087912056	0,046047844
7	Griewank	4,90721E-14	0,023096145	1,43551E-29	0,179328759
8	Salomon	6,24915E-15	7,99031E-22	6,82626E-29	0,14810059
9	W-Wavy	0,098675139	0,000558795	0,184355044	0,074894253
10	XinSheYang4	2,56107E-13	0,13205225	0,149805234	0,199856894

KSA’nın çok modlu fonksiyonlardaki arama geçmişini (*search history*) gösteren kontur grafikleri Şekil 37’de verilmiştir. Bu aşamada, çok modlu fonksiyonların tümü 2-boyutlu olarak

ele alınmıştır. Arama geçmişı, 100 yinelemedeki tüm popölasyonları içermektedir ve her bir grafikte toplamda 300 arama noktası yer almaktadır.





Şekil 37. KSA'nın çok modlu fonksiyonlardaki arama geçmişi.

5.3 GERÇEK DÜNYA PROBLEMLERİNİN TESTLERİ

Bu çalışmada, önerdiğimiz algoritmayı test etmek amacıyla gerçek dünya problemlerinden yararlanılmıştır. Bu problemler; dört kademeli dişli kutusu problemi, hidrostatik baskı yatağı tasarımı problemi, kaynaklı kiriş tasarımı problemi ve endüstriyel soğutma sisteminin optimum tasarım problemi olmak üzere dört tanedir. Tablo 11’de deneysel çalışmalarda kullanılan gerçek dünya problemleri fonksiyonları ve bunların özellikleri verilmiştir.

Tablo 11. Gerçek Dünya Problemlerinin fonksiyonları.

#	Problem	Boyut	Kısıtlar	Alt Sınır	Üst Sınır
1	Dört Kademeli Dişli Kutusu Problemi (DKDKP)	22	86	$Lb_1 \dots Lb_{14} = 6,51$ $Lb_{15} \dots Lb_{22} = 0,51$	$Ub_1 \dots Ub_8 = 76,49$ $Ub_9 \dots Ub_{12} = 4,49$ $Ub_{13} \dots Ub_{22} = 9,49$
2	Hidrostatik Baskı Yatağı Tasarımı Problemi (HBYTP)	4	7	$[1, 1, 1E - 6, 1]$	$[16, 16, 16 \times 1E - 6, 16]$
3	Endüstriyel Soğutma Sisteminin Optimal Tasarımı (ESSOT)	4	5	$[0.125, 0.1, 0.1, 0.1]$	$[2, 10, 10, 2]$
4	Kaynaklı Kiriş Tasarımı (KKT)	14	15	$Lb_1 \dots Lb_{14} = 0,001$	$Ub_1 \dots Ub_{14} = 5$

5.3.1 Deney No. 1

Bu deneyde, yineleme sayısı 100, popülasyon sayısı 30 olarak belirlenmiştir. Yapılan bu deneyde, her bir problem için algoritmalar 30 kez tekrarlı olarak çalıştırılmış ve elde edilen skorların ortalama değerleri alınmıştır.

Tablo 12’de, Deney No 1’e göre gerçek dünya problemleri fonksiyonlarının global en iyi skorları verilmiştir. Dört adet fonksiyonun beş farklı algoritma ile test edilmesi sonucunda, önerilen KSA oldukça rekabetçi sonuçlar elde etmiştir. Genelde AA ve PSO’nun en iyi skorları aldığı gözlemlenmiştir.

Tablo 12. Deney No. 1’e Göre Global En İyi Skorlar.

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	1493,971348	2329,598947	66,7678987	1558,952775	234,4855263
2	HBYTP	1031,625205	1781,535236	178,3426963	21100,48389	713,5280486
3	ESSOT	0,367351266	5,716551825	0,605371129	7,955974791	6,464780117
4	KKT	1,718017443	1,737005996	1,633190813	1,737335766	1,62945503

Algoritmaların çalışma süreleri saniye cinsinden ölçülmüştür. Yapılan deneylerde Tablo 13'te görüldüğü üzere önerilen yöntem, tüm problemlerde en düşük sürede tamamlamıştır. DKDKP probleminde KSA'nın, AA'dan yaklaşık 10 kat ve GKAA'den ise 2 kat daha hızlı çalıştığı gözlemlenmiştir.

Tablo 13. Deney No. 1'e Göre Algoritmaların Çalışma Süreleri (sn).

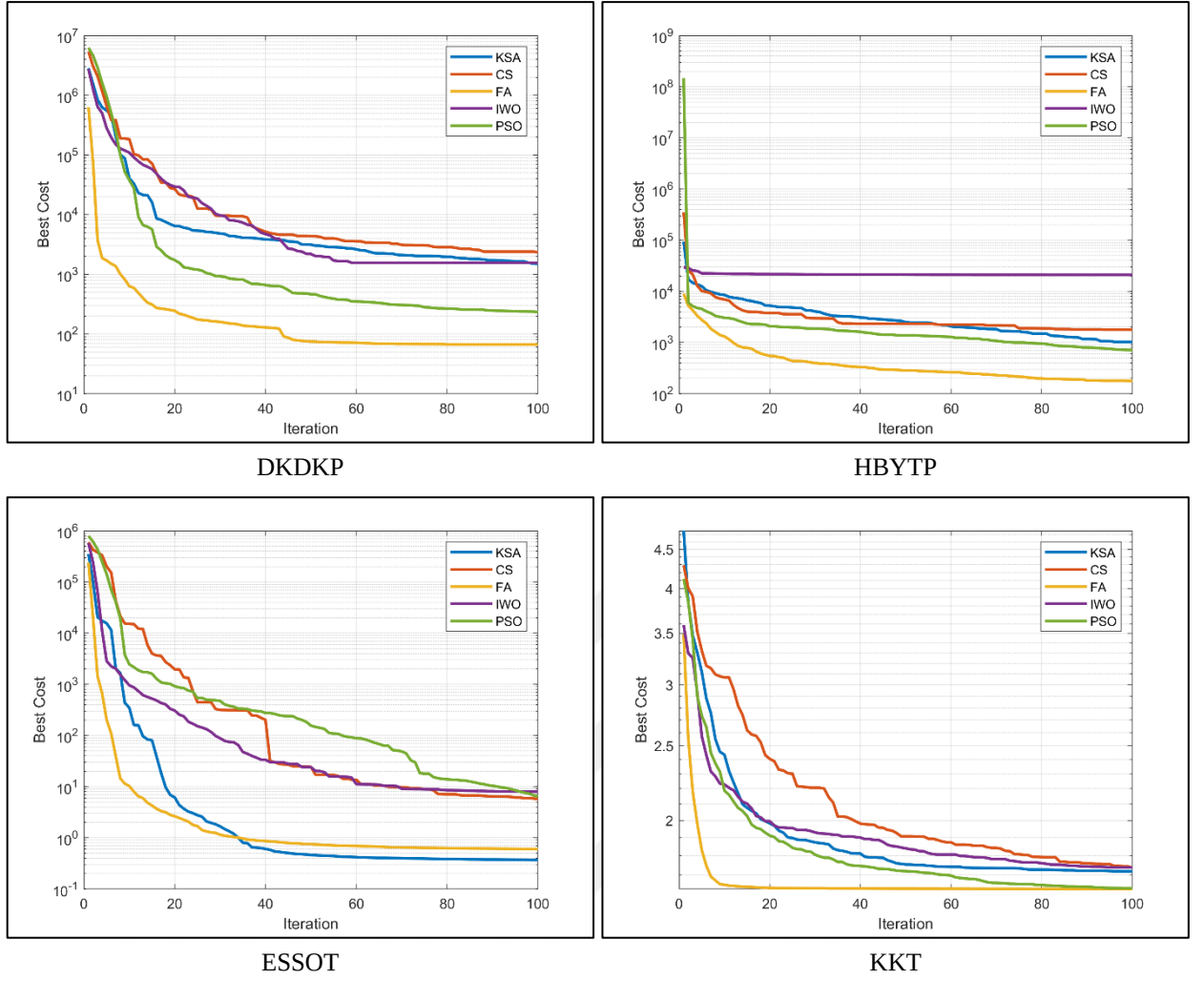
#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	69,42691	136,89876	662,30886	90,79765	80,252
2	HBYP	21,46205	41,16467	348,07972	32,3417	31,66061
3	ESSOT	28,42074	57,6497	433,89348	49,73686	39,83122
4	KKT	20,37797	39,59988	328,35855	32,49773	31,02271

Önerilen yöntemin sonuçlarının, deneysel çalışmalarda kullanılan diğer algoritmalar ile benzerliklerini mukayese etmek amacıyla Tablo 14'teki gibi Wilcoxon işaretli sıralar testi sunulmuştur. Bu testteki güven aralığı %95 nispetinde olup, KSA'nın diğer yöntemlerle arasında genel olarak anlamlı bir ilişki bulunmadığı görülmektedir. Yine bu tabloya göre, KSA'nın davranışı, diğer meta-sezgisellere oranla GKAA algoritmasına daha yakın seyretmekte olduğu göze çarpmaktadır.

Tablo 14. Deney No. 1'e Göre Wilcoxon işaretli sıralar testi (p -değerleri).

#	Problem	KSA-GKAA	KSA-AA	KSA- İYOOA	KSA-PSO
1	DKDKP	0,012114087	2,31121E-27	0,15006331	1,47827E-07
2	HBYP	0,002071572	1,52992E-24	6,05596E-29	0,065454362
3	ESSOT	1,32684E-15	0,25219309	4,40246E-15	2,11467E-14
4	KKT	0,026908158	0,00232227	0,001867342	0,159093796

Şekil 38'de, Deney 1'e göre gerçek dünya problemlerinin yakınsama grafikleri gösterilmiştir. Grafiklerde görüldüğü gibi KSA, test edilen fonksiyonların birinde en iyi sonuca ulaşmıştır. AA, genel olarak daha düşük iterasyon sayısında sonuca yaklaşmaktadır ve genelde en iyi sonuçlara ulaşmaktadır. Deneyde KSA mavi renk ile GKAA turuncu renk ile AA sarı renk ile İYOOA mor renk ile PSO yeşil renk ile gösterilmiştir.



Şekil 38. Deney No. 1'e Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.

5.3.2 Deney No. 2

Bu deneyde, yineleme sayısı 500, popülasyon sayısı 30 olarak belirlenmiştir. Yapılan bu deneyde, her bir problem için algoritmalar 30 kez tekrarlı olarak çalıştırılmış ve elde edilen skorların ortalama değerleri alınmıştır.

Tablo 15'te, Deney No 2'ye göre gerçek dünya problemleri fonksiyonlarının global en iyi skorları verilmiştir. Dört adet fonksiyonun beş farklı algoritma ile test edilmesi sonucunda, önerilen KSA oldukça rekabetçi sonuçlar elde etmiştir. Test edilen problemlerin birinde en iyi sonuca ulaşmıştır.

Tablo 15. Deney No. 2'ye Göre Global En İyi Skorlar.

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	126,5956946	496,365955	91,6611976	104,8157351	274,2535645
2	HBYP	325,820623	744,7544377	125,7616354	18774,46205	225,9872395
3	ESSOT	1,072449837	0,530306696	1,580990073	1,107151764	0,546141067
4	KKT	1,62895982	1,629825643	1,62929959	1,70602321	1,631831603

Algoritmaların çalışma süreleri saniye cinsinden ölçülmüştür. Yapılan deneylerde Tablo 16'da görüldüğü üzere önerilen yöntem, problemlerin çoğunda en düşük sürede tamamlamıştır. DKDKP probleminde KSA'nın, GKAA'dan yaklaşık 2 kat daha hızlı çalıştığı gözlemlenmiştir. Sadece bir problemde İYOOA, KSA'dan daha iyi bir sonuç alabilmiştir.

Tablo 16. Deney No. 2'ye Göre Algoritmaların Çalışma Süreleri (sn).

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	732,0993	1412,0959	982,2454	813,8103	817,9174
2	HBYP	224,3843	406,6112	3539,1191	66,1621	315,9889
3	ESSOT	297,4629	564,7376	4366,1065	377,6696	382,9788
4	KKT	205,9286	376,3487	3324,6076	275,0613	299,3416

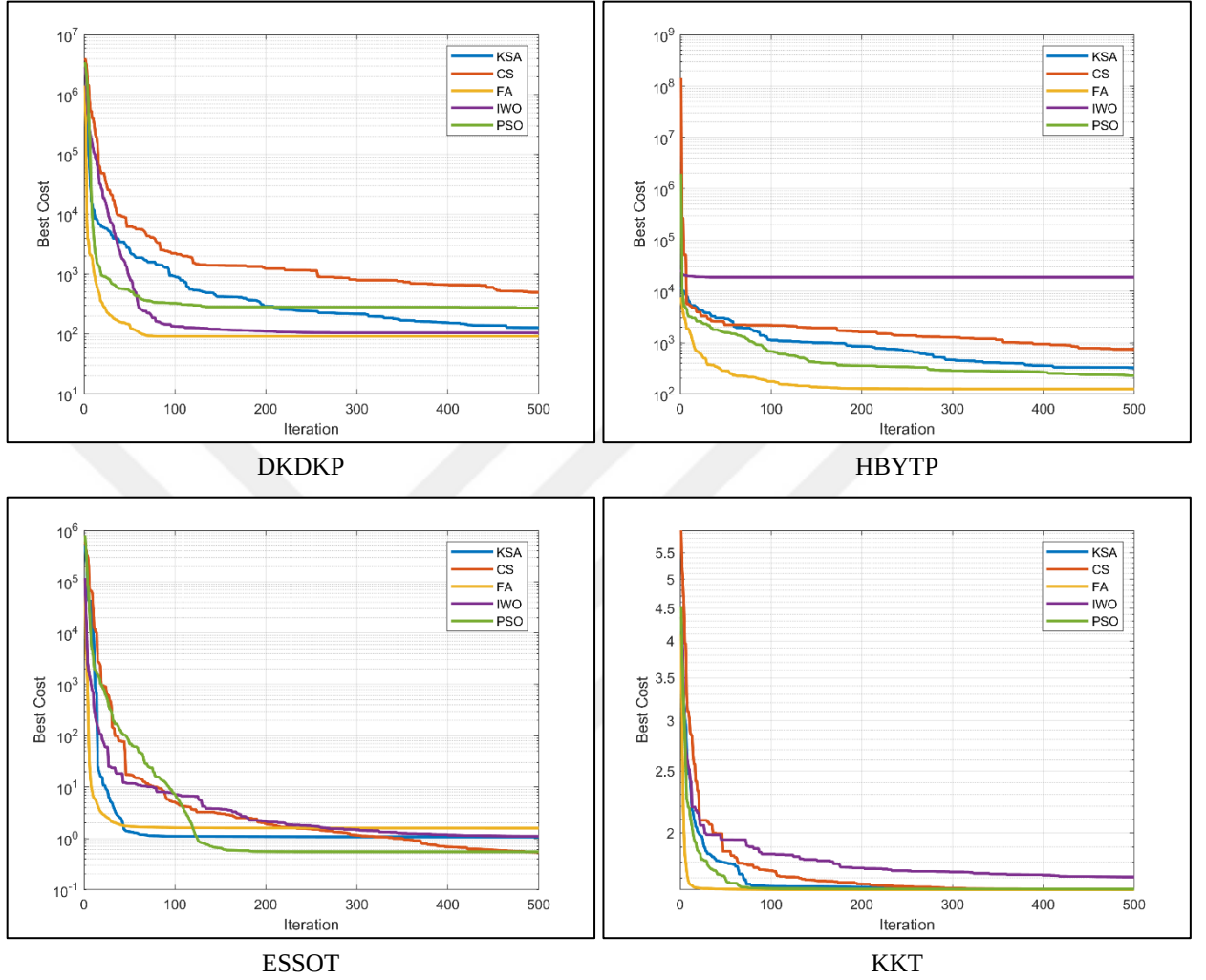
Önerilen yöntemin sonuçlarının, deneysel çalışmalarda kullanılan diğer algoritmalar ile benzerliklerini mukayese etmek amacıyla Tablo 17'deki gibi Wilcoxon işaretli sıralar testi sunulmuştur. Bu testteki güven aralığı %95 nispetinde olup, KSA'nın diğer yöntemlerle arasında genel olarak anlamlı bir ilişki bulunmadığı görülmektedir. Yine bu tabloya göre, KSA'nın davranışı, diğer meta-sezgisellere oranla AA algoritmasına daha yakın seyretmekte olduğu göze çarpmaktadır.

Tablo 17. Deney No. 2'ye Göre Wilcoxon işaretli sıralar testi (*p*-değerleri).

#	Problem	KSA-GKAA	KSA-AA	KSA- İYOOA	KSA-PSO
1	DKDKP	3,99029E-56	6,23366E-64	2,19984E-07	3,12306E-15
2	HBYP	0,038838507	3,8431E-110	2,3812E-162	6,06475E-45
3	ESSOT	9,7073E-08	1,1162E-123	3,16823E-05	1,36243E-17
4	KKT	4,30489E-10	3,86552E-26	6,26888E-44	0,139471118

Şekil 39'da, Deney 2'e göre gerçek dünya problemlerinin yakınsama grafikleri gösterilmiştir. Grafiklerde görüldüğü gibi KSA, test edilen fonksiyonlarda oldukça rekabetçi sonuçlara ulaşmıştır. Genel olarak algoritmaların hepsi, benzer iterasyon sayısında sonuca

yaklaşmaktadır. Deneyde KSA mavi renk ile GKAA turuncu renk ile AA sarı renk ile İYOOA mor renk ile PSO yeşil renk ile gösterilmiştir.



Şekil 39. Deney No. 2'ye Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.

5.3.3 Deney No. 3

Bu deneyde, yineleme sayısı 1000, popülasyon sayısı 30 olarak belirlenmiştir. Yapılan bu deneyde, her bir problem için algoritmalar 30 kez tekrarlı olarak çalıştırılmış ve elde edilen skorların ortalama değerleri alınmıştır.

Tablo 18'de, Deney No 3'e göre gerçek dünya problemleri fonksiyonlarının global en iyi skorları verilmiştir. Dört adet fonksiyonun beş farklı algoritma ile test edilmesi sonucunda, önerilen KSA oldukça rekabetçi sonuçlar elde etmiştir. Problemlerin birinde KSA en iyi skora ulaşmıştır.

Tablo 18. Deney No. 3'e Göre Global En İyi Skorlar.

#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	65,04978784	538,7301613	68,39026191	70,54690895	955,7075354
2	HBYP	124,8583043	246,1802578	125,8011077	6206,190293	124,806261
3	ESSOT	0,049386851	0,030766282	0,039393828	0,142279711	2,601781351
4	KKT	1,629229308	1,628869083	1,628885059	1,632049932	1,628914842

Algoritmaların çalışma süreleri saniye cinsinden ölçülmüştür. Yapılan deneylerde Tablo 19'da görüldüğü üzere önerilen yöntem, problemlerin çoğunu en düşük sürede tamamlamıştır. Problemlerin yalnızca birinde İYOOA'nın, KSA'dan daha iyi sürede çalıştığı gözlemlenmiştir. DKDKP probleminde KSA'nın, PSO'ya yakın ve GKAA'dan ise iki kat daha hızlı çalıştığı gözlemlenmiştir.

Tablo 19. Deney No. 3'e Göre Algoritmaların Çalışma Süreleri (sn).

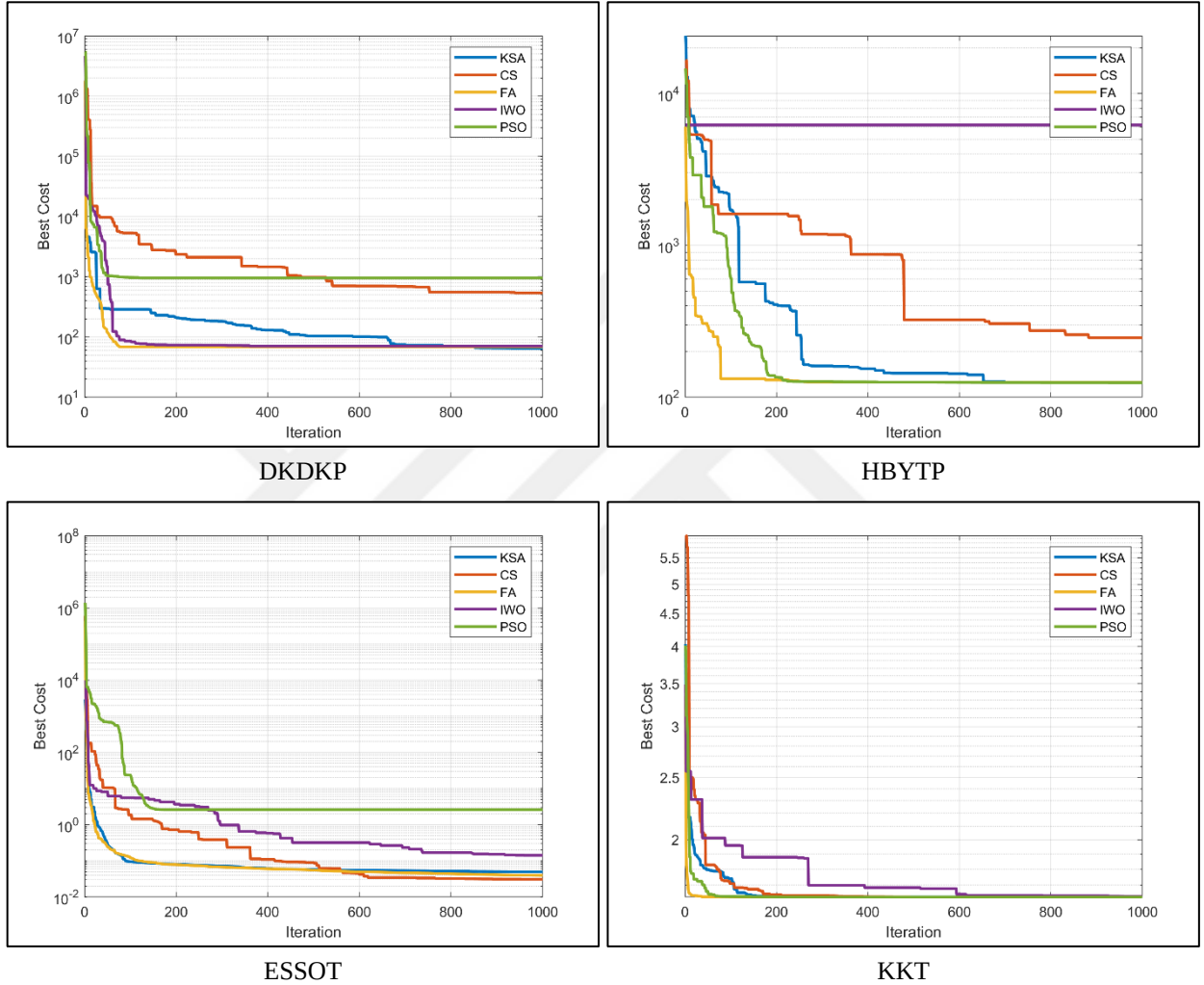
#	Problem	KSA	GKAA	AA	İYOOA	PSO
1	DKDKP	760,06692	1443,2514	1713,90366	906,21584	887,26968
2	HBYP	281,25682	559,74316	4859,61148	261,25294	438,00322
3	ESSOT	358,87426	680,35178	5627,86898	483,78488	538,14412
4	KKT	295,30466	525,5397	4603,01218	334,0449	341,23354

Önerilen yöntemin sonuçlarının, deneysel çalışmalarda kullanılan diğer algoritmalar ile benzerliklerini mukayese etmek amacıyla Tablo 20'deki gibi Wilcoxon işaretli sıralar testi sunulmuştur. Bu testteki güven aralığı %95 nispetinde olup, KSA'nın diğer yöntemlerle arasında genel olarak anlamlı bir ilişki bulunmadığı görülmektedir. Yine bu tabloya göre, KSA'nın davranışı, diğer meta-sezgisellere oranla İYOOA ve PSO algoritmalarına daha yakın seyretmekte olduğu göze çarpmaktadır. KSA, İYOOA ve PSO ile birer problemde benzer olarak gözlemlenmiştir.

Tablo 20. Deney No. 3'e Göre Wilcoxon işaretli sıralar testi (*p*-değerleri).

#	Problem	KSA-GKAA	KSA-AA	KSA- İYOOA	KSA-PSO
1	DKDKP	1,4575E-298	9,6648E-141	1,28369E-61	7,2509E-308
2	HBYP	4,5253E-138	1,29203E-32	0	1,94768E-56
3	ESSOT	0,314864938	2,63309E-17	2,6446E-274	0
4	KKT	1,69368E-14	9,0739E-237	7,1487E-209	8,4509E-124

Şekil 40'ta, Deney 3'e göre gerçek dünya problemlerinin yakınsama grafikleri gösterilmiştir. Grafiklerde görüldüğü gibi KSA, test edilen fonksiyonların çoğunda benzer sonuca ulaşmıştır. AA, genel olarak daha düşük iterasyon sayısında sonuca yaklaşmaktadır. Deneyde KSA mavi renk ile GKAA turuncu renk ile AA sarı renk ile İYOOA mor renk ile PSO yeşil renk ile gösterilmiştir.



Şekil 40. Deney No. 3'e Göre Gerçek Dünya Problemlerinin Yakınsama Grafikleri.

6. TARTIŞMA VE SONUÇ

Bu tez çalışmasında, optimizasyon problemlerinin çözülmesi amacıyla doğadan esinlenilerek yeni bir metasezgisel algoritma önerilmiştir. Karasinek Algoritması (KSA) olarak isimlendirdiğimiz yöntem, karasineklerin doğal davranışları göz önüne alınarak geliştirilmiştir.

Önerilen algoritmayı test etmek için, 10 adet tek modlu sentetik kıyaslama fonksiyonu, 10 adet çok modlu sentetik kıyaslama fonksiyonu ve 4 adet gerçek dünya problemi kullanılmıştır. KSA'nın başarısını ölçmek için PSO, AA, GKAA ve İYOOA gibi literatürde iyi bilinen metasezgisel algoritmalar ile kıyaslanmıştır.

Yapılan deneysel çalışmalarda önerilen yöntem, PSO, AA, GKAA ve İYOOA gibi literatürde iyi bilinen metasezgisel algoritmalarla yarışır düzeyde performans göstermiştir. Tek modlu ve çok modlu sentetik kıyaslama fonksiyonlarında KSA, kıyasladığımız diğer algoritmalarla göre oldukça başarılı sonuçlar elde etmiştir. Gerçek dünya problemlerinin çözümünde ise önerilen algoritmanın, kıyaslama yapılan diğer algoritmalar ile benzer sonuçlar elde ettiği gözlemlenmiştir.

7. KAYNAKLAR

- [1] Dhiman, G., & Kumar, V. (2018). Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge-Based Systems*, 159, 20-50
- [2] Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(4), 503-526.
- [3] Soria-Alcaraz, J. A., Ochoa, G., Espinal, A., Sotelo-Figueroa, M. A., Ornelas-Rodriguez, M., & Rostro-Gonzalez, H. (2020). A Methodology for Classifying Search Operators as Intensification or Diversification Heuristics. *Complexity*, 2020, 1-10.
- [4] Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, 214(1), 108-132.
- [5] Holland, J. (1975). An introduction with application to biology, control and artificial intelligence *Adaptation in Natural and Artificial System*, MIT Press, Cambridge, MA.
- [6] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, Cambridge.
- [7] Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359.
- [8] Kennedy, J., & Eberhart, R. Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, 1942-1948, Perth, WA, Australia.
- [9] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, 200, 1-10.
- [10] Simon, D. (2008). Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, 12(6), 702-713.
- [11] Dorigo, M., & Di Caro, G. (n.d.). Ant colony optimization: a new meta-heuristic. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, 1470-1477.
- [12] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4), 355-366.
- [13] Yang, X.-S. (2012). *Flower Pollination Algorithm for Global Optimization*. *Lecture Notes in Computer Science*, 7445, 240-249.
- [14] Geem, Z.W., Kim, J.H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76(2), 60-68.

- [15] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.
- [16] Savsani, P., & Savsani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling*, 40(5-6), 3951-3978.
- [17] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm– A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110-111, 151-166.
- [18] Salimi, H. (2015). Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75, 1-18.
- [19] Chou, J.-S., & Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535.
- [20] Bogar, E., & Beyhan, S. (2020). Adolescent Identity Search Algorithm (AISA): A novel metaheuristic approach for solving optimization problems. *Applied Soft Computing*, 95, 106503.
- [21] Ali, H., Tariq, U. U., Hussain, M., Lu, L., Panneerselvam, J., & Zhai, X. (2020). ARSH-FATI a Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks. *IEEE Systems Journal*, 15(2), 1-12.
- [22] Zhang, J., Xiao, M., Gao, L., & Pan, Q. (2018). Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Applied Mathematical Modelling*, 63, 464-490.
- [23] Ghasemian, H., Ghasemian, F., & Vahdat-Nejad, H. (2020). Human urbanization algorithm: A novel metaheuristic approach. *Mathematics and Computers in Simulation*, 178, 1-15.
- [24] Zhang, Y., & Jin, Z. (2020). Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Systems with Applications*, 148, 113246.
- [25] Azizi, M., Talatahari, S. & Gandomi, (2023). A.H. Fire Hawk Optimizer: a novel metaheuristic algorithm. *Artif Intell Rev*, 56, 287-363.
- [26] Talatahari, S., & Azizi, M. (2020). Chaos Game Optimization: a novel metaheuristic algorithm. *Artificial Intelligence Review*, 54, 917-1004.
- [27] Azizi, M. (2021). Atomic orbital search: A novel metaheuristic algorithm. *Applied Mathematical Modelling*, 93, 657-683.

- [28] Abdollahzadeh, B., Soleimanian-Gharehchopogh, F., Khodadadi, N. & Mirjalili, S., (2022). Mountain Gazelle Optimizer: A New Nature-inspired Metaheuristic Algorithm for Global Optimization Problems. *Advances in Engineering Software*, 174(103282), 1-34.
- [29] Cheraghalipour, A., Hajiaghaei-Keshteli, M., & Paydar, M. M. (2018). Tree Growth Algorithm (TGA): A novel approach for solving optimization problems. *Engineering Applications of Artificial Intelligence*, 72, 393-414.
- [30] Xie, L., Han, T., Zhou, H., Zhang, Z. R., Han, B., & Tang, A. (2021). Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization. *Computational Intelligence and Neuroscience*, 2021, 9210050.
- [31] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness Mohammed A.A., & Gandomi, A. H. (2021). Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157, 107250.
- [32] Feng, Z., Niu, W., & Liu, S. (2020). Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Applied Soft Computing*, 98, 106734.
- [33] Rashid, M. F. F. (2020). Tiki-taka algorithm: a novel metaheuristic inspired by football playing style. *Engineering Computations*, 38(1), 313-343.
- [34] Hashim, F. A. & Hussien, A.G., (2022). Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*. 242, 108320.
- [35] Hayyolalam, V., & Pourhaji-Kazem, A. A. (2020). Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 87, 103249.
- [36] Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2019). A novel meta-heuristic optimization method based on golden ratio in nature. *Soft Computing*, 24, 1117-1151.
- [37] Rahmani, A. & Aliabdi, I. (2022). Plant competition optimization: A novel metaheuristic algorithm. *Expert Systems*. 39(6), 1-28.
- [38] Braik, M., Sheta, A., & Al-Hiary, H. (2020). A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural Computing and Applications*, 33, 2515-2547.
- [39] Oyelade, O. N., Ezugwu, A. E.-S., Mohamed, T. I. A. & Abualigah, L., (2022). Ebola Optimization Search Algorithm: A New Nature-Inspired Metaheuristic Optimization Algorithm. *IEEE Access*, 10, 1-38.

- [40] Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 80, 20-34.
- [41] Ahmadianfar, I., Bozorg-Haddad, O., & Chu, X. (2020). Gradient-Based Optimizer: A New Metaheuristic Optimization Algorithm. *Information Sciences*, 540, 131-159.
- [42] Agushaka, O., Ezugwu, A., & Abualigah, L. (2022). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 35, 1-33.
- [43] Qiao, W., & Yang, Z. (2019). Solving Large-Scale Function Optimization Problem by Using a New Metaheuristic Algorithm Based on Quantum Dolphin Swarm Algorithm. *IEEE Access*, 7, 138972-138989.
- [44] Osaba, E., Villar-Rodriguez, E., Del-Ser, J., Nebro, A. J., Molina, D., LaTorre, A., Herrera, F. (2021). A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. *Swarm and Evolutionary Computation*, 64, 100888.
- [45] Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2008). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2), 239–287.
- [46] Avcu, E., Elen, A. & Kahramanli-Örnek, H. (2020). Making Inferences About Settlements from Satellite Images Using Glowworm Swarm Optimization, *Journal of Electrical Engineering and Technology*, 15(5), 2345-2360.
- [47] Stegherr, H., Heider, M., & Hähner, J. (2020). Classifying Metaheuristics: Towards a unified multi-level classification system. *Natural Computing*, 21, 155-171.
- [48] Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., & Muhuri, P. K. (2021). Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review*, 54(6), 4237-4316.
- [49] Saman, M. A., Awaz, A.S., Zeravan, A. A., Rasan, I. A. & Jayson, A. D. F. (2023). Overview of Metaheuristic Algorithms. *Polaris Global Journal of Scholarly Research and Trends*, 2(2), 10-32.
- [50] Tilahun, S. L. (2019). Balancing the Degree of Exploration and Exploitation of Swarm Intelligence Using Parallel Computing. *International Journal on Artificial Intelligence Tools*, 28(03), 1950014.

- [51] Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2018). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, 31, 7665-7683.
- [52] Gandomi, A., Yang, X.-S., Talatahari, S. & Alavi, A. (2013). *Metaheuristic Applications in Structures and Infrastructures*, Elsevier.
- [53] Dorigo, M., & Stützle, T. (n.d.). *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*. International Series in Operations Research & Management Science, 57, 250-285.
- [54] Ulusoy, S., Nigdeli, S. M., & Bekdaş, G. (2020). Novel metaheuristic-based tuning of PID controllers for seismic structures and verification of robustness. *Journal of Building Engineering*, 33, 101647.
- [55] Yagiura, M. (1999). *Studies on metaheuristic algorithms for combinatorial optimization problems*. Doktora Tezi, Kyoto University, Kyoto/Japonya.
- [56] El-Nainay, M.Y. (2009). *Island Genetic Algorithm-based Cognitive Networks*. Doktora Tezi, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
- [57] Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2018). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52, 2191-2233.
- [58] Çayiroğlu, İ. & Elen, A. (2012). A Heuristic Optimization Approach for A Real-World University Timetabling Problem. *Advances in Computer Science and Engineering*, 9(2), 103-131.
- [59] Elen, A. & Çayiroğlu, İ. (2010). Solving of Scheduling Problem with Heuristic Optimization Approach. *Engineering Science and Technology an International Journal*, 13(3), 159-172.
- [60] Abd-ElSabour, N. (2016). *Hybrid Metaheuristics for Classification Problems*. Pattern Recognition - Analysis and Applications.
- [61] Kaçabeti, A. (2021). *A New Hybrid Metaheuristic Algorithm Applied On A Classification Problem*. Yüksek Lisans Tezi, İstanbul Sabahattin Zaim Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.
- [62] Yüzügüldü, Ş. Y. (2021). *A Health-Based Metaheuristic Algorithm: Cholesterol Algorithm*. Yüksek Lisans Tezi, Gaziantep Üniversitesi, Fen Bilimleri Ve Uygulamalı Bilimler Enstitüsü, Gaziantep.
- [63] Stützle, T., & López-Ibáñez, M. (2018). Automated Design of Metaheuristic Algorithms. *International Series in Operations Research & Management Science*, 272, 541–579.

- [64] Uymaz, S. A. (2015). Yeni Bir Biyolojik İlhamlı Metasezgisel Optimizasyon Metodu: Yapay Alg Algoritması. Doktora Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya
- [65] Mousavirad, S. J., Schaefer, G., Jalali, S. M. J., & Korovin, I. (2020). A Benchmark Of Recent Population-Based Metaheuristic Algorithms For Multi-Layer Neural Network Training, Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 1402–1408
- [66] Pierezan, J., & Dos Santos Coelho, L. (2018). Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems, 2018 IEEE Congress on Evolutionary Computation (CEC).
- [67] Yang, X.-S., Chien, S. F., & Ting, T. O. (2015). Bio-Inspired Computation and Optimization. Bio-Inspired Computation in Telecommunications, 2015, 1-21.
- [68] Peyvandi, M., Zafarani, M., & Nasr, E. (2011). Comparison of Particle Swarm Optimization and the Genetic Algorithm in the Improvement of Power System Stability by an SSSC-based Controller. Journal of Electrical Engineering and Technology, 6(2), 182-191.
- [69] Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. Lecture Notes in Computer Science, 5792, 169-178.
- [70] Arora, S., & Kaur, R. (2018). An Escalated Convergent Firefly Algorithm. Journal of King Saud University- Computer and Information Sciences, 34(2), 308-315.
- [71] Yang, X.-S., & Suash Deb. (2009). Cuckoo Search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 210-214.
- [72] Abdelaziz, A. Y., & Ali, E. S. (2015). Cuckoo Search algorithm based load frequency controller design for nonlinear interconnected power system. International Journal of Electrical Power & Energy Systems, 73, 632-643.
- [73] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics, 1(4), 355-366.
- [74] Misaghi, M., & Yaghoobi, M. (2019). Improved Invasive weed optimization Algorithm (IWO) Based on Chaos Theory for Optimal design of PID controller. Journal of Computational Design and Engineering, 6(3), 284-295.
- [75] https://entnemdept.ufl.edu/creatures/urban/flies/house_fly.htm (Erişim Tarihi: 14.08.2022)
- [76] <http://www.kznhealth.gov.za/environ/vector/housefly.htm> (Erişim Tarihi: 03.09.2022)
- [77] <https://extension.unh.edu/resource/house-fly-fact-sheet> (Erişim Tarihi: 03.09.2022)

- [78] Schou, T. M., Faurby, S., Kjærsgaard, A., Pertoldi, C., Loeschcke, V., Hald, B., & Bahrndorff, S. (2013). Temperature and Population Density Effects on Locomotor Activity of *Musca domestica* (Diptera: Muscidae). *Environmental Entomology*, 42(6), 1322-1328.
- [79] Elen, A. & Közkurt, C. (2020). Sezgisel algoritmalarla robot kavrayıcı mekanizmasının optimal tasarımı, 2nd International Turkic World Congress on Science and Engineering, 79-88.
- [80] Közkurt, C. & Elen, A. (2020). Planet dişli zinciri tasarım probleminin sezgisel optimizasyon yöntemleriyle çözümü, 2nd International Turkic World Congress on Science and Engineering, 89-100.
- [81] Nadimi-Shahraki, M. H. (2023). An Effective Hybridization of Quantum-based Avian Navigation and Bonobo Optimizers to Solve Numerical and Mechanical Engineering Problems. *Journal of Bionic Engineering*, 20, 1361-1385.
- [82] Yokota, T., Taguchi, T., & Gen, M. (1998). A solution method for optimal weight design problem of the gear using genetic algorithms. *Computers & Industrial Engineering*, 35(3-4), 523-526.
- [83] Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56, 100693.
- [84] Azizi, M., Talatahari, S., & Giaralis, A. (2021). Optimization of Engineering Design Problems Using Atomic Orbital Search Algorithm. *IEEE Access*, 9, 102497-102519.
- [85] Kaveh, A., & Talatahari, S. (2011). Hybrid charged system search and particle swarm optimization for engineering design problems. *Engineering Computations*, 28(4), 423-440.
- [86] Li, Z., Zhou, Y., Zhang, S., & Song, J. (2016). Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems. *Mathematical Problems in Engineering*, 2016, 1-22.
- [87] Andrei, N. (2017). Applications of continuous nonlinear optimization. In *Springer Optimization and Its Applications. Continuous Nonlinear Optimization for Engineering Applications in GAMS Technology*, 121, 47-117.
- [88] Manske, K. A., Reindl, D. T., & Klein, S. A. (2001). Evaporative condenser control in industrial refrigeration systems. *International Journal of Refrigeration*, 24(7), 676-691.

8. EKLER

EK 1. Karasinek Algoritmasının Matlab Programlama Dili ile Hazırlanmış Kaynak Kodu

```
function result = KarasinekAlgoritmasi(options)
    %% Probleme ilişkin parametrelerin belirlenmesi.
    FitnessFunction = options.FitnessFunction; % Uygunluk fonksiyonu
    Dimension = options.Dimension; % Karar değişkenlerinin sayısı
    Lb = options.Lb; % Karar değişkenlerinin alt sınır değerleri
    Ub = options.Ub; % Karar değişkenlerinin üst sınır değerleri
    IterationSize = options.IterationSize; % Maksimum yineleme sayısı
    PopulationSize = options.PopulationSize; % Popülasyon büyüklüğü

    %% Karasinek Algoritmasının parametreleri
    Luminous = 0.90; % Luminous ratio
    Humidity = 0.50; % Relative humidity ratio
    StepCoeff = 0.40; % Step coefficient

    %% Initialization
    init = initHFA(options);
    Swarm = init.Swarm;
    GlobalBest = init.GlobalBest;
    BestCost = zeros(IterationSize, 1); % Array to Hold Best Cost Values

    %% Karasinek Algoritmasının ana döngüsü
    for t = 1 : IterationSize
        % Bazı sineklerin uçuş eylemi.
        temperature = (9)+(41-9) * rand(); % Ortam sıcaklığını belirle.

        K = tempEffectOnFlight(temperature, PopulationSize);
        flies = randperm(PopulationSize, K);
        flyValues = BrownianMotion(K, Dimension);

        % Move house flies.
        for k = 1 : K
            currID = flies(k);
            agent = Swarm(currID);

            % Exploration of house flies.
            agent.Position = agent.Position + flyValues(k, :);

            agent.Position = min(max(agent.Position, Lb), Ub); % Konumu limitlere göre düzelt.
            agent.Cost = FitnessFunction(agent.Position); % Uygunluk değerini hesapla
            % Set result
            if (agent.Cost < Swarm(currID).Cost)
                Swarm(currID) = agent;
            end
        end

        % Yerdeki sineklerin konum değiştirilmesi.
        indices = setdiff((1:PopulationSize), flies);
```

```

% Get direction and velocity.
dirVelo = (-Luminous)+(2.0 * Luminous).*randn(numel(indices), Dimension);

for j = 1 : numel(indices)
    indx = indices(j); % Yerdeki sineklerden j-inci olanı seç.
    agent = Swarm(indx); % Sürüden seçilen karasineği al.
    agent.Position = agent.Position + ... % Yeni konum değerlerini belirle.
        (dirVelo(j, :)).*(agent.Position - GlobalBest.Position) * Humidity;

    % Set result
    if (agent.Cost < Swarm(indx).Cost)
        Swarm(indx) = agent;
    end
end

[cost, index] = min([Swarm.Cost]); % Popülasyondaki en düşük maliyetin tespiti.
if (cost < GlobalBest.Cost) % Bulunan maliyet değeri, en iyi çözümünkinden küçükse,
    GlobalBest = Swarm(index); % Bu karasineği en iyi çözümü olarak belirle.
end

BestCost(t) = GlobalBest.Cost; % t-inci yinelemedeki en iyi maliyeti sakla.
end

result.Name = 'HFA';
result.Costs = BestCost;
disp([result.Name, ': ', num2str(GlobalBest.Cost)])
return;
end

function result = initHFA(options)
    % Empty agent structure
    agent.Position = [];
    agent.Cost = [];
    % Initialize swarm array
    Swarm = repmat(agent, options.PopulationSize, 1);

    % Belirlenen sayıdaki başlangıç popülasyonunu oluştur.
    for i = 1 : options.PopulationSize
        Swarm(i).Position = unifrnd(options.Lb, options.Ub, [1, options.Dimension]);
        Swarm(i).Cost = options.FitnessFunction(Swarm(i).Position);
    end

    [~, index] = min([Swarm.Cost]); % En düşük maliyetli karasineğin indisini bul.
    GlobalBest = Swarm(index); % Bu karasineği en iyi çözümü olarak belirle.

    result.Swarm = Swarm;
    result.GlobalBest = GlobalBest;
    return;
end

```

```

function result = tempEffectOnFlight(temperature, popSize)
    Mu = 25; % Normal dağılımın ortalama değeri.
    Sig = 4; % Normal dağılımın standart sapması.
    % Ortam sıcaklığına göre uçuş eylemini gerçekleştirecek karasinek sayısının belirlenmesi.
    N = normpdf(temperature, Mu, Sig).* 10;
    result = ceil(N * popSize);
end

```

```

function result = BrownianMotion(N, D)
    if (nargin < 2), D = 1; end
    dt = 1.0 / (N * D);
    dW = sqrt(dt) * randn(N, D);
    result = cumsum(dW, 2);
end

```

```

clc;
clear;
close all;

```

```

options = struct( ...
    'Dimension', 30, ...           % Problem boyutu.
    'Lb', -10, ...                % Arama uzayının alt-sınır değeri.
    'Ub', 10, ...                 % Arama uzayının üst-sınır değeri.
    'FitnessFunction', @(x) funcSphere(x), ...
    'IterationSize', 300, ...      % Yineleme sayısı.
    'PopulationSize', 25);        % Popülasyon sayısı.

```

%% Karasinek Algoritması'nın çağırılması.

```
method = KarasinekAlgoritmasi(options);
```

%% Sonuçların gösterilmesi.

```
figure();
semilogy(method.Costs, 'DisplayName', method.Name, 'LineWidth', 2);

```

```

hold off;
xlabel('Iteration');
ylabel('Best Cost');
grid on;

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı
Doğum Yeri
Doğum Tarihi
Uyruğu
Telefon
E-Posta Adresi
Web Adresi

Eğitim Bilgileri Lisans

Üniversite
Fakülte
Bölümü
Mezuniyet Yılı

Yüksek Lisans

Üniversite
Enstitü Adı
Anabilim Dalı
Programı

Makale ve Bildiriler