

**T.C.  
PAMUKKALE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**OPTİMİZASYON KURAMINDA YENİ BİR METASEZGİSEL  
YAKLAŞIM: ERGEN KİMLİK ARAMA ALGORİTMASI (AISA)  
VE MÜHENDİSLİK UYGULAMALARI**

**DOKTORA TEZİ**

**EŞREF BOĞAR**

**DENİZLİ, HAZİRAN-2021**

**T.C.**  
**PAMUKKALE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**



**OPTİMİZASYON KURAMINDA YENİ BİR METASEZGİSEL  
YAKLAŞIM: ERGEN KİMLİK ARAMA ALGORİTMASI (AISA) VE  
MÜHENDİSLİK UYGULAMALARI**

**DOKTORA TEZİ**

**EŞREF BOĞAR**

**DENİZLİ, HAZİRAN-2021**

**Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalışmalara atfedildiđine beyan ederim.**

**EŐREF BOĐAR**

## ÖZET

### **OPTİMİZASYON KURAMINDA YENİ BİR METASEZGİSEL YAKLAŞIM: ERGEN KİMLİK ARAMA ALGORİTMASI (AISA) VE MÜHENDİSLİK UYGULAMALARI**

**DOKTORA TEZİ  
EŞREF BOĞAR**

**PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI  
TEZ DANIŞMANI: DOÇ. DR. SELAMİ BEYHAN  
DENİZLİ, HAZİRAN-2021**

Bu tezde, ergenlerin kimlik geliştirme/arama sürecini taklit eden, Ergen Kimlik Arama Algoritması (AISA) adı verilen yeni bir popülasyon temelli metasezgisel optimizasyon algoritması önerilmiştir. AISA, akran grubundaki ergenlerin kimlik oluşturma davranışlarından esinlenerek geliştirilmiş ve bu ergen davranışları optimizasyon problemlerini çözmek için matematiksel olarak modellenmiştir. Önerilen algoritmanın keşif, sömürü, yerel optimumdan kaçınma ve yakınsama özellikleri 39 iyi bilinen tek-modlu, çok-modlu, sabit boyutlu çok-modlu, bileşik ve CEC 2019 kıyaslama fonksiyonları üzerinde test edilmiştir. AISA'nın performansını doğrulamada karşılaştırma için 13 iyi bilinen metasezgisel algoritma kullanılmıştır. Ayrıca, AISA'nın çözüm yeteneğinin performansı gerçek hayat mühendislik uygulamaları olarak kabul edilen IIR sistem tanılama, yedi serbestlik dereceli robot manipülatörün ters kinematiği ve görüntü iyileştirme problemleri üzerinde değerlendirilmiştir. Elde edilen uygulama sonuçları, AISA'nın güçlü ve etkin bir algoritma olduğunu ve hem teorik hem de gerçek hayatta çeşitli karmaşık sayısal optimizasyon problemlerini çözme açısından diğer iyi bilinen metasezgisel algoritmalara göre üstün optimizasyon performansı üretme kabiliyetine sahip olduğunu göstermektedir.

**ANAHTAR KELİMELER:** Ergen kimlik arama algoritması, Metasezgisel, Optimizasyon, Chebyshev fonksiyonel bağlantı ağı, IIR sistem tanılama, Ters kinematik, Görüntü iyileştirme.



## **ABSTRACT**

### **A NOVEL METAHEURISTIC APPROACH IN OPTIMIZATION THEORY: ADOLESCENT IDENTITY SEARCH ALGORITHM (AISA) AND ENGINEERING APPLICATIONS**

**Ph.D THESIS**

**EŞREF BOGAR**

**PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE  
ELECTRICAL AND ELECTRONICS ENGINEERING  
(SUPERVISOR: ASSOC. PROF. DR. SELAMİ BEYHAN)**

**DENİZLİ, JUNE-2021**

In this thesis, a novel population-based metaheuristic optimization algorithm, called Adolescent Identity Search Algorithm (AISA), is proposed which imitates the process of identity development/search of adolescents. AISA is developed inspired by the identity formation behavior of adolescents in the peer group and adolescent behaviors are modelled mathematically to solve optimization problems. The proposed algorithm is evaluated on 39 well-known unimodal, multimodal, fixed-dimensional multimodal, composite and CEC 2019 benchmark functions to test exploration, exploitation, local optima avoidance, and convergence properties. Thirteen state-of-art metaheuristic algorithms are used for comparison to verify the performance of AISA. Additionally, performance of AISA's solution capability is tested IIR system identification, inverse kinematics problem of a seven degrees of freedom robot manipulator and image enhancement problems which are considered as the real-life engineering applications. Obtained application results show that AISA is a strong and effective algorithm and has a high capability to produce superior optimization performance over other well-known metaheuristic algorithms in terms of solving various complex numerical optimization problems on both theoretical and real-life.

**KEYWORDS:** Adolescent identity search algorithm, Metaheuristic, Optimization, Chebyshev functional-link network, IIR system identification, Inverse kinematics, Image enhancement.

# İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> . . . . .	<b>i</b>
<b>ABSTRACT</b> . . . . .	<b>ii</b>
<b>İÇİNDEKİLER</b> . . . . .	<b>iii</b>
<b>KISALTMALAR LİSTESİ</b> . . . . .	<b>vi</b>
<b>ÖNSÖZ</b> . . . . .	<b>vii</b>
<b>ŞEKİL LİSTESİ</b> . . . . .	<b>viii</b>
<b>TABLO LİSTESİ</b> . . . . .	<b>x</b>
<b>1. GİRİŞ</b> . . . . .	<b>1</b>
<b>2. ERGEN KİMLİK ARAMA ALGORİTMASI (AISA)</b> . . . . .	<b>9</b>
2.1 Algoritmanın İlham Kaynağı . . . . .	9
2.2 AISA Algoritması . . . . .	11
2.2.1 Başlangıç Çözüm Üretme . . . . .	11
2.2.2 Uygunluk Hesabı . . . . .	12
2.2.3 Yeni Kimlik (Çözüm) Üretme . . . . .	12
2.2.4 Sınır Kontrol Mekanizması . . . . .	18
2.2.5 Güncelleme Mekanizması . . . . .	18
2.2.6 Durdurma/Sonlandırma Kriteri . . . . .	19
2.3 AISA'nın Diğer Metasezgiseller ile Yapısal Karşılaştırması . . . . .	20
<b>3. STANDART TEST FONKSİYONLARI ÜZERİNDE YAPILAN DENEYSEL ÇALIŞMALAR</b> . . . . .	<b>25</b>
3.1 Standart Kıyaslama Test Fonksiyonları . . . . .	25
3.2 Karşılaştırma için Kullanılan Metasezgisel Algoritmalar . . . . .	32
3.3 Deneysel Plan ve Girdiler . . . . .	35
3.4 Test Fonksiyonları Üzerinde Performans Değerlendirmesi . . . . .	38
3.4.1 10, 30 ve 50 boyutlu tek-modlu test fonksiyonlarında ( $F_1 - F_7$ ) performans değerlendirme . . . . .	38

3.4.2	10, 30 ve 50 boyutlu çok-modlu test fonksiyonlarında ( $F_8 - F_{13}$ ) performans değerlendirilmesi . . . . .	38
3.4.3	Sabit boyutlu çok-modlu test fonksiyonlarında ( $F_{14} - F_{23}$ ) performans değerlendirilmesi . . . . .	39
3.4.4	Bileşik test fonksiyonlarında ( $F_{24} - F_{29}$ ) performans değerlendirilmesi	39
3.4.5	CEC 2019 test fonksiyonlarında ( $F_{30} - F_{39}$ ) performans değerlendirilmesi . . . . .	40
3.4.6	Wilcoxon İşaretili Sıra Testi . . . . .	40
3.4.7	Yakınsama Analizi . . . . .	41
3.4.8	Zaman Verimliliği Analizi . . . . .	42
<b>4.</b>	<b>MÜHENDİSLİK PROBLEMLERİ ÜZERİNDE YAPILAN DENEYSEL ÇALIŞMALAR . . . . .</b>	<b>68</b>
4.1	AISA ile IIR Sistem Tanılama Probleminin Çözümü . . . . .	68
4.2	AISA ile 7-DOF Robot Manipülörün Ters Kinematik Probleminin Çözümü . . . . .	75
4.3	AISA ile Görüntü İyileştirme Probleminin Çözümü . . . . .	80
<b>5.</b>	<b>SONUÇLAR VE TARTIŞMALAR . . . . .</b>	<b>90</b>
<b>6.</b>	<b>KAYNAKLAR . . . . .</b>	<b>93</b>

## KISALTMALAR LİSTESİ

<b>GA</b>	: Genetik Algoritma (Genetic Algorithm)
<b>DE</b>	: Diferansiyel Evrim (Differential Evolution)
<b>ES</b>	: Evrim Stratejisi (Evolution Strategy)
<b>EDA</b>	: Dağılım Tahmini Algoritması (Estimation of Distribution Algorithm)
<b>BBO</b>	: Biyocoğrafya Tabanlı Optimizasyonu (Biogeography Based Optimization)
<b>PSO</b>	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
<b>ACO</b>	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
<b>ABC</b>	: Yapay Arı Kolonisi (Artificial Bee Colony)
<b>CSO</b>	: Kedi Sürüsü Optimizasyonu (Cat Swarm Optimization)
<b>CS</b>	: Guguk Kuşu Arama (Cuckoo Search)
<b>FA</b>	: Ateşböceği Algoritması (Firefly Algorithm)
<b>BA</b>	: Yarasa Algoritması (Bat Algorithm)
<b>KH</b>	: Karides Sürüsü (Krill Herd)
<b>FOA</b>	: Meyve Sineği Optimizasyon Algoritması (Fruit Fly Optimization Algorithm)
<b>GWO</b>	: Gri Kurt Optimizasyon Algoritması (Grey Wolf Optimizer)
<b>WSA</b>	: Ağırlıklı Süperpozisyon Çekimi (Weighted Superposition Attraction)
<b>AAA</b>	: Yapay Alg Algoritması (Artificial Algae Algorithm)
<b>CSA</b>	: Karga Arama Algoritması (Crow Search Algorithm)
<b>WOA</b>	: Balina Optimizasyon Algoritması (Whale Optimization Algorithm)
<b>GOA</b>	: Çekirge Optimizasyon Algoritması (Grasshopper Optimisation Algorithm)
<b>SSA</b>	: Salp Sürüsü Algoritması (Salp Swarm Algorithm)
<b>EPO</b>	: İmparator Penguen Optimizasyonu (Emperor Penguin Optimizer)
<b>SSA</b>	: Sincap Arama Algoritması (Squirrel Search Algorithm)
<b>PFA</b>	: Yol Bulucu (Kaşif) Algoritması (Pathfinder Algorithm)
<b>SA</b>	: Tavlama Benzetimi (Simulated Annealing)
<b>BB-BC</b>	: Büyük Patlama-Büyük Çöküş (Big-Bang Big-Crunch)
<b>GSA</b>	: Yerçekimi Arama Algoritması (Gravitational Search Algorithm)
<b>CRO</b>	: Kimyasal Reaksiyon Optimizasyonu (Chemical Reaction Optimization)
<b>CSS</b>	: Yüklü Sistem Arama (Charged System Search)
<b>RA</b>	: Işın Optimizasyonu (Ray Optimization)
<b>BH</b>	: Kara Delik (Black Hole)
<b>GBMO</b>	: Gaz Brownian Hareketi Optimizasyonu (Gases Brownian Motion Optimization)
<b>IMO</b>	: İyon Hareket Optimizasyonu (Ions Motion Optimization)
<b>VS</b>	: Girdap Arama (Vortex Search)
<b>MVO</b>	: Çoklu Evren Optimizasyonu (Multi-Verse Optimizer)

<b>TS</b>	: Tabu Arama (Tabu Search)
<b>TLBO</b>	: Öğretme-Öğrenme Tabanlı Optimizasyon (Teaching–Learning-Based Optimization)
<b>MBA</b>	: Maden Patlatma Algoritması (Mine Blast Algorithm)
<b>ISA</b>	: İç Arama Algoritması (Interior Search Algorithm)
<b>EMA</b>	: Borsa Piyasası Algoritması (Exchange Market Algorithm)
<b>HBBO</b>	: İnsan Davranışı Tabanlı Optimizasyon (Human Behavior-Based Optimization)
<b>VPL</b>	: Voleybol Premier Lig (Volleyball Premier League)
<b>SEO</b>	: Sosyal Mühendislik Optimizasyonu (Social Engineering Optimizer)
<b>NFL</b>	: No Free Lunch
<b>AISA</b>	: Ergen Kimlik Arama Algoritması (Adolescent Identity Search Algorithm)
<b>DSÖ</b>	: Dünya Sağlık Örgütü
<b>LSE</b>	: En Küçük Kareler Tahmini (Least Squares Estimation)
<b>CFLN</b>	: Chebyshev fonksiyonel bağlantı ağı (Chebyshev functional-link network)
<b>SOS</b>	: Simbiyotik Organizmalar Arama (Symbiotic Organisms Search)
<b>BSA</b>	: Geri-izleme Arama Optimizasyon Algoritması (Backtracking Search Optimization Algorithm)
<b>CEC 2005</b>	: 2005 IEEE Evrimsel Hesaplama Kongresi (2005 IEEE Congress on Evolutionary Computation)
<b>CEC 2019</b>	: 2019 IEEE Evrimsel Hesaplama Kongresi (2019 IEEE Congress on Evolutionary Computation)
<b>MFO</b>	: Güve-Alev Optimizasyonu (Moth–Flame Optimization)
<b>SCA</b>	: Sinüs Kosinüs Algoritması (Sine Cosine Algorithm)
<b>TSA</b>	: Ağaç-tohum Algoritması (Tree-seed Algorithm)
<b>Ort.</b>	: Ortalama Değer
<b>Std.</b>	: Standart Sapma
<b>KD</b>	: Kazanma Durumu
<b>IIR</b>	: Sonsuz Darbe Cevaplı (Infinite Impulse Response)
<b>MSE</b>	: Ortalama Karesel Hata (Mean Squared Error)
<b>DOF</b>	: Serbestlik Derecesi (Degree of Freedom)

## ÖNSÖZ

Doktora tez çalışmam boyunca bilgi ve birikimlerini benden esirgemeyen danışmanım Doç. Dr. Selami Beyhan'a, tez izleme komite üyeleri Prof. Dr. Serdar İplikçi, Prof. Dr. Kadir Kavaklıođlu, Prof. Dr. Sezai Tokat hocalarıma, tez savunma jürisinde bulunarak değerli katkılar sunan Doç. Dr. Savaş Şahin ve Doç. Dr. Emre Çomak hocalarıma teşekkür ederim.

Eđitim hayatım boyunca verdikleri desteklerden dolayı tüm aileme ve en çok da doktora çalışmalarım boyunca ihmal ettiđim kızım Asya, ođlum Ardıç ve eşim Zeynep'e sonsuz teşekkürlerimi sunarım.

Ayrıca doktora eğitimim boyunca 2211/A Genel Yurt İçi Doktora Burs Programı kapsamında destek veren Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna (TÜBİTAK) teşekkür ederim.

## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1: Önerilen ortogonal Chebyshev ağı . . . . .	14
Şekil 3.1: İki boyutlu tek-modlu test fonksiyonlarının grafikleri. . . . .	27
Şekil 3.2: İki boyutlu çok-modlu test fonksiyonlarının grafikleri. . . . .	28
Şekil 3.3: İki boyutlu sabit boyutlu çok-modlu test fonksiyonlarının grafikleri.	30
Şekil 3.4: İki boyutlu bileşik test fonksiyonlarının grafikleri. . . . .	31
Şekil 3.5: İki boyutlu CEC 2019 test fonksiyonlarının ( $F_{33} - F_{39}$ ) grafikleri.	31
Şekil 3.6: <u>10</u> boyutlu $F_1 - F_{13}$ fonksiyonları üzerinde algoritmaların yakınsama eğrileri. . . . .	62
Şekil 3.7: <u>30</u> boyutlu $F_1 - F_{13}$ fonksiyonları üzerinde algoritmaların yakınsama eğrileri. . . . .	63
Şekil 3.8: <u>50</u> boyutlu $F_1 - F_{13}$ fonksiyonları üzerinde algoritmaların yakınsama eğrileri. . . . .	64
Şekil 3.9: $F_{14} - F_{26}$ fonksiyonları üzerinde algoritmaların yakınsama eğrileri.	65
Şekil 3.10: $F_{27} - F_{39}$ fonksiyonları üzerinde algoritmaların yakınsama eğrileri.	66
Şekil 3.11: Algoritmaların toplam çalışma süreleri. . . . .	67
Şekil 4.1: IIR filtre sistem tanılama konfigürasyonu. . . . .	69
Şekil 4.2: IIR sistem tanılama problemleri için algoritmaların yakınsama eğrileri (a) Uygulama 1 - Durum 1; (b) Uygulama 1 - Durum 2; (c) Uygulama 2 - Durum 1; (d) Uygulama 2 - Durum 2 . . . . .	75
Şekil 4.3: Ters kinematik problemleri için algoritmaların yakınsama eğrileri (a) $P_{desired1}$ ; (b) $P_{desired2}$ . . . . .	80
Şekil 4.4: Bölgesel/bütünsel kontrast iyileştirme için kullanılan görüntüler. .	84
Şekil 4.5: İyileştirme uygulanan görüntüler üzerinde algoritmaların yakınsama eğrileri. . . . .	87

Şekil 4.6: AISA ile iyileştirilen görüntüler ve histogramları (a) orijinal görüntü; (b) orijinal görüntünün histogramı; (c) iyileştirilen görüntü; (d) iyileştirilen görüntünün histogramı. . . . . 88



## TABLO LİSTESİ

	<u>Sayfa</u>
Tablo 3.1: Tek-modlu test fonksiyonları. . . . .	27
Tablo 3.2: Çok-modlu test fonksiyonları. . . . .	27
Tablo 3.3: Sabit boyutlu çok-modlu test fonksiyonları. . . . .	28
Tablo 3.4: Bileşik test fonksiyonları. . . . .	29
Tablo 3.5: CEC 2019 test fonksiyonları. . . . .	30
Tablo 3.6: Algoritmaların kontrol parametreleri. . . . .	37
Tablo 3.7: <u>10</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	43
Tablo 3.8: <u>30</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	44
Tablo 3.9: <u>50</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	45
Tablo 3.10: <u>10</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	46
Tablo 3.11: <u>30</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	47
Tablo 3.12: <u>50</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	48
Tablo 3.13: Sabit boyutlu çok-modlu test fonksiyonları ( $F_{14} - F_{23}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	49
Tablo 3.14: Bileşik test fonksiyonları ( $F_{24} - F_{29}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	50
Tablo 3.15: CEC 2019 test fonksiyonları ( $F_{30} - F_{39}$ ) üzerinde algoritmaların elde ettiği sonuçlar. . . . .	51
Tablo 3.16: <u>10</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	52

Tablo 3.17:	<u>30</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	53
Tablo 3.18:	<u>50</u> boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	54
Tablo 3.19:	<u>10</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	55
Tablo 3.20:	<u>30</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	56
Tablo 3.21:	<u>50</u> boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	57
Tablo 3.22:	Sabit boyutlu çok-modlu test fonksiyonları ( $F_{14} - F_{23}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	58
Tablo 3.23:	Bileşik test fonksiyonları ( $F_{24} - F_{29}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	59
Tablo 3.24:	CEC 2019 test fonksiyonları ( $F_{30} - F_{39}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	60
Tablo 3.25:	Test kategorilerine ait Wilcoxon işaretli sıra testi sonuçları ( $\alpha = 0.05$ ). . . . .	61
Tablo 4.1:	Uygulama 1 - Durum 1 için elde edilen MSE and MSE (dB) değerleri. . . . .	73
Tablo 4.2:	Uygulama 1 - Durum 2 için elde edilen MSE and MSE (dB) değerleri. . . . .	73
Tablo 4.3:	Uygulama 2 - Durum 1 için elde edilen MSE and MSE (dB) değerleri. . . . .	73
Tablo 4.4:	Uygulama 2 - Durum 2 için elde edilen MSE and MSE (dB) değerleri. . . . .	74
Tablo 4.5:	IIR sistem tanılama problemleri için Wilcoxon işaretli sıra testi sonuçları. . . . .	74
Tablo 4.6:	Yedi eklemli robotun DH parametreleri. . . . .	76
Tablo 4.7:	Algoritmaların $P_{desired1} = [-25 \ 100 \ 50]^T$ (cm) için 30 koşulda elde ettikleri karşılaştırma sonuçları. . . . .	79

Tablo 4.8:	Algoritmaların $P_{desired2} = [50 \quad -25 \quad 75]^T$ (cm) için 30 koşulda elde ettikleri karşılaştırma sonuçları. . . . .	79
Tablo 4.9:	$P_{desired1} = [-25 \quad 100 \quad 50]^T$ (cm) için Wilcoxon işaretli sıra testi sonuçları. . . . .	79
Tablo 4.10:	$P_{desired2} = [50 \quad -25 \quad 75]^T$ (cm) için Wilcoxon işaretli sıra testi sonuçları. . . . .	79
Tablo 4.11:	Bölgesel/bütünsel kontrast iyileştirme uygulanan görüntüler üzerinde algoritmaların elde ettiği sonuçlar. . . . .	85
Tablo 4.12:	Bölgesel/bütünsel kontrast iyileştirme uygulanan görüntüler için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ). . . . .	86

## 1. GİRİŞ

Optimizasyon, bir amaç fonksiyonunun kısıtsız veya belirli kısıtlar altında aldığı değeri en küçük veya en büyük yapan bir dizi değişkenin, en iyi değerini bulma süreci olarak tanımlanabilir. Optimizasyon konusu, bir sistemdeki mevcut kaynakları en verimli şekilde kullanarak belirli hedeflere ulaşmak için uygulamada ve teoride sıkça çalışılan önemli bir alan haline gelmiştir. Gerçek dünya problemlerine bakıldığında mühendislik, işletme, ekonomi, ulaşım gibi birçok farklı alanda optimizasyon problemleriyle karşılaşmaktadır. Genel olarak optimizasyon problemleri, kısıtlı veya kısıtsız, ayrık veya sürekli, statik veya dinamik, tek veya çok-amaçlı olma durumlarına göre çeşitli kategorilere ayrılmaktadırlar (Dhiman ve Kumar 2018).

Pek çok gerçek dünya optimizasyon problemi yüksek karmaşıklığa, doğrusal olmayan kısıtlara, değişkenler arasında karşılıklı bağımlılığa ve geniş bir çözüm aralığına sahip olduklarından dolayı, bu tür optimizasyon problemlerinin çözümünü bulmak zor bir görev olarak kabul edilir (Gogna ve Tayal 2013). Bu problemleri çözmek için araştırmacılar genel olarak matematiksel programlama yaklaşımlarını ve metasezgisel yöntemler olarak adlandırılan optimizasyon yöntemlerini önermişlerdir. Matematiksel programlama yaklaşımları çok-modlu, sürekli olmayan ve türevlenemeyen büyük ölçekli gerçek dünya problemlerinin birçoğunun üstesinden gelmede her zaman verimli değildirler (Heidari ve diğ. 2019). Kesin bir çözümü garanti etmeyen ancak makul bir sürede kabul edilebilir sonuçlar sağlayan metasezgisel yaklaşımlar ise alternatif çözüm arayıcılar olarak tasarlanmıştır. Yaklaşık son kırk yıldır uygulamada olan metasezgisel algoritmalar basitlik/sadelik, esneklik, türev içermeyen mekanizmaları ve yerel optimumdan kaçınma gibi güçlü yanlara sahip olmaları nedeniyle her geçen gün daha da yaygın kullanılmaktadırlar (Mirjalili ve diğ. 2014).

Metasezgisel yöntemler oldukça basittir; çoğunlukla temel kavramlardan

esinlenilerek geliştirilmişlerdir. Bu yöntemlerdeki ilham kaynakları genelde fiziksel fenomenler, canlı davranışları veya evrimsel kavramlarla ilgilidir. Basitlik, bilgisayar bilimcilerinin farklı doğal kavramları simüle etmelerine, yeni metasezgiseller önermelerine, iki veya daha fazla metasezgiseli hibritlemelerine veya mevcut metasezgiselleri geliştirmelerine olanak tanır. Ayrıca, diğer bilim insanlarının metasezgiselleri hızlı bir şekilde öğrenmelerine ve bunları problemlerine uygulamalarına da yardımcı olur.

İkinci güçlü yan esneklik ise, bir algoritmanın yapısında herhangi bir özel değişiklik olmaksızın metasezgisellerin farklı problemlere uygulanabilirliğini ifade eder. Metasezgisel yöntemler, optimizasyon problemlerini çoğunlukla kara kutular (black boxes) olarak kabul ettiğinden, farklı problemlere kolaylıkla uygulanabilirler. Başka bir deyişle, metasezgisel bir yöntem için önemli olan, bir sistemin yalnızca girdileri ve çıktılarıdır. Bir tasarımcının ihtiyacı problemin iç yapısından ziyade, problemi metasezgisellere nasıl aktaracağını bilmektir.

Metasezgisel yöntemler için bir diğer önemli özellik ise çoğunun türev içermeyen mekanizmalara sahip olmasıdır. Gradyan/türev-tabanlı optimizasyon yaklaşımlarının aksine, metasezgisel yöntemler problemleri stokastik olarak optimize eder. Metasezgisel yöntemlerle optimizasyon süreci, rastgele çözüm(ler) ile başlar ve optimum çözümü bulmak için fonksiyonun türevini hesaplamaya ihtiyaç duymaz. Bu durumda metasezgisel yöntemler ile çözüm elde edilirken, zahmetli türev alma işlemleri ve/veya türevlenemeyen gerçek dünya problemlerinin zorlukları ortadan kaldırılmış olur.

Son olarak metasezgisel yöntemler, geleneksel optimizasyon tekniklerine kıyasla yerel optimumdan kaçınmada üstün yeteneklere sahiptir. Bu özellik, metasezgisel algoritmaların yerel çözümlerde durağanlıktan kaçınma ve tüm arama alanını kapsamlı bir şekilde araştırmalarına izin veren stokastik doğalarından kaynaklanmaktadır. Gerçek dünya problemlerin arama alanı genellikle bilinmez ve çok sayıda yerel optimuma sahip oldukları için karmaşık bir yapıya sahiptir. Bu nedenle

metasezgisel yöntemler bu zorlu problemleri optimize etmek için iyi seçeneklerdir.

Metasezgisel yaklaşımlarda, bir algoritmanın arama kabiliyetiyle doğrudan ilgili iki vazgeçilmez bileşen vardır: keşif (exploration, diversification) ve sömürü (exploitation, intensification). Keşif, keşfedilmemiş bölgelere derinlemesine yayılarak umut verici çözümler bulmaya çalışmaktır. Başka bir deyişle keşif, çözüm çeşitliliğini artırmaya odaklanır. Aksine sömürü, keşifle elde edilen ve gelecek vadeden çözümler etrafında yerel arama yaparak çözümlerin kalitesini artırmayı amaçlar. Bu bileşenler birbiriyle çatışır ve birbirini engeller (Alba ve Dorronsoro 2005). Bu nedenle bir optimizasyon algoritması, keşif ve sömürü arasında doğru ve makul bir denge gözetilerek tasarlanmalıdır. Aksi takdirde yerel optimum çözüme takılma ve olgunlaşmamış veya yavaş yakınsama gibi istenmeyen senaryolar ortaya çıkabilir.

Metasezgisel algoritmalar farklı şekillerde sınıflandırılabilir (Talbi 2009; Fister ve diğ. 2013); ancak sınıflandırma ile ilgili olarak metasezgisel çalışmaların odağı ve perspektifi, genellikle başlangıç çözümü türü ve ilham kaynaklarıdır.

Başlangıç çözümü türüne bağlı olarak metasezgiseller, tek çözüm tabanlı ve popülasyon (çoklu çözüm) tabanlı olmak üzere iki kategoriye ayrılmaktadır. Yörünge yöntemleri olarak da adlandırılan tek çözüm tabanlı yöntemlerde, arama süreci bir aday çözümle manipüle edilirken, popülasyon tabanlı yöntemlerde optimizasyon sürecinin her yinelemesinde birden çok çözüme dayalı ilerleme sağlanır. Genel olarak birbirlerine göre avantajları ve dezavantajları aşağıdaki gibidir:

- Kodlama süreci ile ilgili olarak, tek çözüme dayalı metasezgiseller, popülasyon tabanlı olanlara göre daha basittir ve daha az bellek gerektirir.
- Tek çözüm tabanlı metasezgisellerin sömürü kabiliyeti, popülasyon tabanlı metasezgisellerden daha iyidir.
- Popülasyon tabanlı metasezgisellerde çözümler, arama alanı içinde birbirleri arasında bilgi alışverişi yapabilen çok sayıda arama ajanı tarafından yürütülür. Böylece, arama alanını tek çözüm tabanlı olanlara göre daha etkili bir şekilde gerebilir ve yerel çözümden kaçınılırlar.

- Popülasyon tabanlı metasezgisellerin keşif yeteneği, tek çözüm tabanlılara göre daha iyidir.

- En önemlisi ise popülasyon tabanlı metasezgiseller, keşif ve sömürü arasında daha iyi bir dengeye sahiptirler.

Diğer bir sınıflandırma türü olan ilham kaynağına bağlı metasezgiseller, genel olarak evrimsel algoritmalar, sürü zekası tabanlı, fizik/kimya tabanlı ve insan tabanlı algoritmalar olmak üzere dört ana kategoriye ayrılabilirler (Mirjalili ve diğ. 2016).

Evrimsel algoritmalar, Darwin'in evrim teorisinden esinlenilerek geliştirilmiş olup; rekombinasyon, mutasyon ve seleksiyon gibi doğadaki biyolojik evrimsel davranışları taklit ederler. Bu kategoriye ait algoritmaların en popüler olanlarından bazıları şunlardır; Genetik Algoritma (Genetic Algorithm - GA) (Holland 1975), Diferansiyel Evrim (Differential Evolution - DE) algoritması (Storn ve Price 1997), Evrim Stratejisi (Evolution Strategy - ES) (Rechenberg 1973), Dağılım Tahmini Algoritması (Estimation of Distribution Algorithm - EDA) (Larrañaga ve Lozano 2002) ve Biyocoğrafya Tabanlı Optimizasyon (Biogeography Based Optimization - BBO) (Simon 2008).

Sürü zekası tabanlı algoritmalar ise yiyecek arama, çiftleşme, göç gibi sürü davranışlarından ilham alınarak geliştirilmişlerdir. Kennedy ve Eberhart (1995) tarafından kuş sürüsünün akıllı sosyal davranışından esinlenilerek önerilmiş olan Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO) bu kategoriye ait en popüler algoritmadır. Karınca Kolonisi Optimizasyonu (Ant Colony Optimization - ACO) algoritması (Dorigo ve diğ. 1996), Yapay Arı Kolonisi (Artificial Bee Colony - ABC) algoritması (Karaboga ve Basturk 2007), Kedi Sürüsü Optimizasyonu (Cat Swarm Optimization - CSO) (Chu ve diğ. 2006), Guguk Kuşu Arama (Cuckoo Search - CS) algoritması (Yang ve Dep 2009), Ateşböceği Algoritması (Firefly Algorithm - FA) (Yang 2009), Yarasa Algoritması (Bat Algorithm - BA) (Yang 2010), Karides Sürüsü (Krill Herd - KH) algoritması (Gandomi ve Alavi 2012), Meyve Sineği Optimizasyon Algoritması (Fruit Fly Optimization Algorithm - FOA) (Pan 2012),

Gri Kurt Optimizasyon Algoritması (Grey Wolf Optimizer - GWO) (Mirjalili ve diğ. 2014), Ağırlıklı Süperpozisyon Çekimi (Weighted Superposition Attraction - WSA) algoritması (Baykasoğlu ve Akpınar 2015), Yapay Alg Algoritması (Artificial Algae Algorithm - AAA) (Uymaz ve diğ. 2015), Karga Arama Algoritması (Crow Search Algorithm - CSA) (Askarzadeh 2016), Balina Optimizasyon Algoritması (Whale Optimization Algorithm - WOA) (Mirjalili ve Lewis 2016), Çekirge Optimizasyon Algoritması (Grasshopper Optimisation Algorithm - GOA) (Saremi ve diğ. 2017), Salp Sürüsü Algoritması (Salp Swarm Algorithm - SSA) (Mirjalili ve diğ. 2017), İmparator Penguen Optimizasyonu (Emperor Penguin Optimizer - EPO) (Dhiman ve Kumar 2018), Sincap Arama Algoritması (Squirrel Search Algorithm - SSA) (Jain ve diğ. 2019), Yol Bulucu (Kaşif) Algoritması (Pathfinder Algorithm - PFA) (Yapici ve Cetinkaya 2019) bu kategorideki diğer algoritma örnekleridir.

Fizik/kimya tabanlı algoritmalar elektrik yükleri, kimyasal reaksiyonlar, yerçekimi, nehir sistemleri, hareket gibi belirli fiziksel veya kimyasal kanunlardan esinlenilerek geliştirilmişlerdir. Tavlama Benzetimi (Simulated Annealing - SA) algoritması (Kirkpatrick ve diğ. 1983), Büyük Patlama-Büyük Çöküş (Big-Bang Big-Crunch - BB-BC) (Erol ve Eksin 2006), Yerçekimi Arama Algoritması (Gravitational Search Algorithm - GSA) (Rashedi ve diğ. 2009), Kimyasal Reaksiyon Optimizasyonu (Chemical Reaction Optimization - CRO) (Lam ve Li 2010), Yüklü Sistem Arama (Charged System Search - CSS) algoritması (Kaveh ve Talatahari 2010), Işın Optimizasyonu (Ray Optimization - RA) (Kaveh ve Khayatazad 2012), Kara Delik (Black Hole - BH) algoritması (Hatamlou 2013), Gaz Brownian Hareketi Optimizasyonu (Gases Brownian Motion Optimization - GBMO) (Abdechiri ve diğ. 2009), İyon Hareket Optimizasyonu (Ions Motion Optimization -IMO) (Javidy ve diğ. 2015), Girdap Arama (Vortex Search - VS) algoritması (Doğan ve Ölmez 2015) ve Çoklu Evren Optimizasyonu (Multi-Verse Optimizer - MVO) (Mirjalili ve diğ. 2016) bu kategoriye ait bazı metasezgisel algoritmalarıdır. Bu kategoriye ait diğer algoritmalar Fister ve diğ. (2013) ve Salcedo-Sanz (2016) çalışmalarında sunulmuştur.



Son kategoride yer alan insan tabanlı algoritmalar ise insan davranışları, etkinlikleri ve özelliklerinden ilham alınarak geliştirilmişlerdir. Bu algoritmaların iyi bilinenlerinden bazıları şunlardır; Tabu Arama (Tabu Search - TS) (Glover 1986), Öğretme-Öğrenme Tabanlı Optimizasyon (Teaching-Learning-Based Optimization - TLBO) algoritması (Rao ve diğ. 2011), Maden Patlatma Algoritması (Mine Blast Algorithm - MBA) (Sadollah ve diğ. 2013), İç Arama Algoritması (Interior Search Algorithm - ISA) (Gandomi 2014), Borsa Piyasası Algoritması (Exchange Market Algorithm - EMA) (Ghorbani ve Babaei 2014), İnsan Davranışı Tabanlı Optimizasyon (Human Behavior-Based Optimization - HBBO) algoritması (Ahmadi 2017), Voleybol Premier Lig (Volleyball Premier League - VPL) algoritması (Moghdani ve Salimifard 2018) ve Sosyal Mühendislik Optimizasyonu (Social Engineering Optimizer - SEO) algoritması (Fathollahi-Fard ve diğ. 2018).

Literatürde gerçek dünya optimizasyon problemlerini çözmeye başarılı olan birçok optimizasyon algoritması olmasına rağmen, bilim insanları halen mevcut problemlerin çözümlerini geliştirmek ve benzeri görülmemiş karmaşık problemleri çözmek için yeni optimizasyon algoritmaları geliştirmeye devam etmektedirler. Buradaki sürekli gelişimin nedeni “No Free Lunch (NFL)” teoreminde yatmaktadır (Wolpert ve Macready 1997). Bu teorem mantıksal olarak tüm optimizasyon problemlerini çözmek için evrensel bir algoritma olmadığını kanıtlamaktadır. Diğer bir ifadeyle, bir optimizasyon algoritması bazı problemler için tatmin edici sonuçlar üretirken, başka problemler için başarılı sonuçlar üretemeyebilir (Mirjalili 2015). Bu nedenle bilimsel optimizasyon topluluğu, bu teoreme dayanarak daha geniş bir problem kümesini veya çözülmemiş belirli problem türlerini çözmeye umuduyla, yeni optimizasyon tekniklerini geliştirmeye çalışmaktadırlar.

Bu noktadan hareketle, bu tez çalışmasının ana amacı gerçek dünya optimizasyon problemlerini çözmek için Ergen Kimlik Arama Algoritması (Adolescent Identity Search Algorithm - AISA) adı verilen yeni, ilgi çekici ve güçlü bir popülasyon tabanlı metasezgisel optimizasyon algoritması geliştirerek bilimsel yazına

katkıda bulunmaktadır. Erikson'un ergenlik teorisine ve ergenler üzerindeki kişisel gözlemlere dayalı olarak geliştirilmiş olan bu algoritma, adından da anlaşılacağı üzere ergenlerin kimlik arama sürecini taklit etmektedir. Bu çalışmanın diğer bir amacı ise, önerilen yöntemin performansını mühendislikte sıklıkla karşılan IIR sistem tanılama, ters kinematik ve görüntü iyileştirme problemlerinde test etmek ve problemlerin optimum çözümlerini bulmaktır.

Tez çalışmasının ilerleyen kısmının organizasyonu aşağıdaki gibi özetlenebilir.

İkinci bölümde; önerilen AISA algoritmasının geliştirilmesinde ilham alınan ergen bir bireyin, akran grubundaki kimlik arama/geliştirme davranışları hakkında bilgiler verilmiş, önerilen algoritmanın modellenmesi, tanıtılması, matematiksel formülasyonu ve kodlanması detaylı olarak anlatılmıştır. Önerilen algoritmanın arama mekanizması ve matematiksel temellerine bağlı olarak diğer metasezgisellerden ayırt edilen özellikleri sunulmuştur.

Üçüncü bölümde; AISA algoritmasının performansını test etmek amacıyla tek-modlu, çok-modlu, sabit boyutlu çok-modlu, bileşik ve CEC 2019 olmak üzere 39 kıyaslama fonksiyonu üzerinde iyi bilinen 13 metasezgisel optimizasyon algoritmayla titiz bir karşılaştırması gerçekleştirilmiştir. Karşılaştırmada kullanılan performans göstergeleri istatistiksel analiz, ölçeklenebilirlik analizi, Wilcoxon işaretli sıra testi, yakınsama analizi ve zaman verimliliği analizidir. Bu standart kıyaslama fonksiyonlarında AISA'nın göstermiş olduğu performans detaylı olarak analiz edilmiştir.

Dördüncü bölümde ise AISA algoritmasının gerçek dünya problemlerini çözümedeki başarısını ispat etmek amacıyla IIR sistem tanılama, yedi serbestlik dereceli robot manipülatörün ters kinematiği ve görüntü iyileştirme optimizasyon problemleri çözdürülmüştür. Yine bu bölümde de AISA'nın performansı iyi bilinen 13 metasezgisel algoritmayla farklı performans göstergelerine göre karşılaştırılmış ve sonuçlar analiz edilmiştir.

Beşinci ve son bölümde ise elde edilen sonuçlar ortaya konmuştur. Algoritmanın güçlü ve zayıf yönleri elde edilen bulgular ışığında tartışılmış olup, algoritmanın gelişime açık yönlerine ve gelecek çalışmalara yer verilmiştir.

## 2. ERGEN KİMLİK ARAMA ALGORİTMASI (AISA)

Bu bölümde, algoritmanın geliştirilmesinde ilham alınan bir ergen bireyin, akran grubundaki kimlik arama/geliştirme davranışları hakkında bilgiler verilmiş, önerilen algoritmanın yapısı tanıtılmış ve önerilen algoritmayı diğer metasezgisel algoritmalarından ayıran özellikler detaylı olarak açıklanmıştır.

### 2.1 Algoritmanın İlham Kaynağı

Latince büyüme, olgunlaşma süreci anlamına gelen ergenlik, tipik olarak çocukluk ve yetişkinlik arasında fiziksel ve psikolojik geçiş dönemi olan bir olgunlaşma evresi olarak tanımlanır (Al-Ani ve Kadhum 2013). Dünya Sağlık Örgütü'ne (DSÖ) göre ise, ergen 10 ile 19 yaşları arasındaki herhangi bir kişi olarak tanımlanmaktadır.

Ergen bir birey, ergenlik döneminde derin ve heyecan verici fiziksel, sosyal, duygusal ve entelektüel değişiklikler yaşar. Dahası ergen kendini bulmaya, kendini kanıtlamaya, bir yaşam tarzı oluşturmaya, hedeflerini belirlemeye, cinsel ilişkiler kurmaya ve insan ilişkilerini düzenlemeye çalışır. Ergen bu süreç boyunca sık sık "Ben kimim?" sorusuna cevap ararken gizemli bir yolculuk yapar (Grotevant 1997). Bu nedenle kimlik, temelde bir kişinin bu soruya nasıl cevap verdiğini ifade eder (Tsang ve diğ. 2012). Başka bir deyişle, kimlik bireyin bu süreçte kazandığı davranışlar, tercihler, düşünceler, yetenekler ve inançlar gibi farklı kimlik seçeneklerinin/özelliklerinin bir kombinasyonudur.

Önde gelen bir gelişim kuramcısı olan Erikson'a göre, ergenlik döneminde ergen daha bağımsız hale gelir ve topluluklardaki farklı alternatifleri veya rolleri araştırır (Erikson 1963). Buna ek olarak Marcia, Erikson'ın kimlik kavramını detaylandırarak, keşif ve bağlanma/adanmışlık gibi iki temel ögeye odaklanarak dağımık kimlik, ipotekli kimlik, başarılı kimlik ve moratoryum olmak üzere dört kimlik statüsü öne sürmüştür (Marcia 1966; Atak 2011). Buradaki her bir kimlik statüsü,

farklı seviyelerde keşif ve bağlanmanın yani karar vermenin bir kombinasyonunu ifade etmektedir.

Ergen, toplumun diğer üyeleriyle etkileşime girer ve bu sosyal etkileşimlerden gelen geribildirimleri kullanarak kimliğini oluşturmaya çalışır. Genel olarak ergen, bağımsız bir birey olmak için aileden uzaklaşmaya meyilliyken, zamanının çoğunu okul ve oyun alanı gibi yerlerde geçirdiği için akranları tarafından kimliğinin şekillenmesine açıktır. Yapı ve normlara bakılmaksızın, herhangi bir akran grubu ergen gelişiminde önemli rol oynar. Bu grubun ergen üzerinde olumlu veya olumsuz bir etkisi olabilir ve ergenin kimlik oluşumu için bir istasyon görevi görür. Ayrıca, akran grubundaki deneyimlerin öz-yeterlik, öz-kontrol, öz-farkındalık ve benlik saygısına önemli bir katkısı vardır (Campbell 2006). Akran grubundaki ergenler, karar verme ve sorumluluk alma yeteneklerini özellikle yetişkinlerin bulunmadığı bir ortamda geliştirirler.

Grup yapısı; yaş, cinsiyet, yer, etnik köken, kültür, eğitim düzeyi ve sosyo-ekonomik durum gibi birçok faktörden etkilenebileceğinden, akran grubundaki ergenlerin kimlik oluşturma süreçlerinin modellenmesi aslında zor bir görevdir. Bu nedenle, gelişim teorilerine ve kendi ergenlik dönemimizdeki deneyimlerimize dayanarak, bir akran grubundaki ergen bireyin üç ana davranışı/durumu benimseyerek kimliğini oluşturabileceği varsayılmış olup; bu durumlar aşağıda verilmiştir:

- Ergen birey toplumun davranışlarını, değerlerini, inançlarını ve tutumlarını gözlemleyerek ve akıl yürüterek kendi kimliğini oluşturabilir (Bandura 1986).
- Ergen birey kimliğini yüksek statüye, güce ve prestije sahip olarak algılanan bir rol modelini taklit ederek oluşturabilir (Bandura 1971; Hurd ve diğ. 2011).
- Ergen birey gruptaki sigara kullanımı, madde kullanımı, erken cinsel davranış ve zorbalık gibi olumsuz/istenmeyen kimlik seçeneklerini benimseyebilir (Ragelienė 2016).

Aktif bir keşif döneminden sonra, yüksek bağlanımda bulunan bireylere atıfta bulunan Marcia'nın başarılı kimlik statüsü ile karakterize edilen bir ergenin kimlik

araması, sayısal optimizasyon problemlerini çözmek için aşağıdaki gibi matematiksel olarak modellenmiştir.

## 2.2 AISA Algoritması

Bu bölümün geri kalan kısmında matematiksel gösterim için genel bir simgelenim (notation) dikkate alınmıştır. Vektörler; harf veya simgenin kalın puntosunda, matrisler; büyük harf veya simgenin kalın puntosunda yazılmıştır. Ayrıca,  $(.)^{-1}$  ve  $(.)^T$  üstsimgeleri sırasıyla ters (inverse) ve devrik (transpose) operatörlerini belirtmektedir.

Önerilen AISA algoritmasının kısıtsız ve tek-amaçlı bir optimizasyon problemi üzerinde uygulandığı varsayıldığında, minimize edilecek problem

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

$$\underline{b}_j \leq x_j \leq \bar{b}_j, \quad j = 1, 2, \dots, n \quad (2.2)$$

şeklinde yazılabilir.  $f(.)$  minimize edilecek amaç fonksiyonunu,  $x_j$ ,  $j$ . karar değişkenini,  $n$  karar değişkeni sayısını (problem boyutu),  $\underline{b}_j$  ve  $\bar{b}_j$  ise sırasıyla  $j$ . değişkenin alt ve üst sınır değerlerini ifade etmektedir.

### 2.2.1 Başlangıç Çözüm Üretme

AISA algoritması diğer popülasyon tabanlı optimizasyon algoritmalarına benzer şekilde, çözüm alanı sınırları içerisinde rastgele bir başlangıç popülasyon oluşturarak optimizasyon sürecine başlar. Oluşturulan başlangıç popülasyonu, yapay bir akran grubuna karşılık gelmektedir. Bu grupta  $N$  sayıda ergen birey olduğu ve her bir ergen kimliğinin  $n$  sayıda kimlik özelliğine (değişken sayısı) sahip olduğu varsayılmaktadır.  $i$ . ergenin kimliği  $\{\mathbf{x}^i\}_{i=1,2,\dots,N}$  bir vektör ile temsil edilmekte ve aşağıda gösterildiği gibi elde edilmiştir:

$$x_j^i = \underline{b}_j + U(0, 1)_j \times (\bar{b}_j - \underline{b}_j), \quad i = 1, 2, \dots, N; \quad j = 1, 2, \dots, n \quad (2.3)$$

burada  $x_j^i$ ,  $i$ . ergenin  $j$ . kimlik özelliği ve  $U(0, 1)$  değerini  $[0, 1]$  aralığında alan düzgün dağılımlı rastgele bir değişkendir. Tüm ergenlerin kimliklerini içeren bir popülasyon matrisi ise aşağıdaki gibi gösterilmiştir:

$$\mathbf{X} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_n^N \end{bmatrix}_{N \times n} \quad (2.4)$$

burada  $\mathbf{X}$  popülasyon matrisini temsil etmektedir.

### 2.2.2 Uygunluk Hesabı

Her bir ergenin uygunluk değeri, ergen kimliğinin (karar değişkenlerinin değerleri) önceden tanımlanmış amaç fonksiyonunda yerine koyulmasıyla hesaplanır. Hesaplanan uygunluk değerleri aşağıdaki gibi  $N \times 1$  boyutlu bir sütun vektöründe tutulur:

$$\mathbf{f}(\mathbf{X}) = \begin{bmatrix} f_1 [x_1^1, x_2^1, \cdots, x_n^1] \\ f_2 [x_1^2, x_2^2, \cdots, x_n^2] \\ \vdots \\ f_N [x_1^N, x_2^N, \cdots, x_n^N] \end{bmatrix}_{N \times 1} \quad (2.5)$$

burada  $\mathbf{f}(\mathbf{X})$  mevcut çözüm popülasyonunun uygunluk vektörünü temsil etmektedir.

### 2.2.3 Yeni Kimlik (Çözüm) Üretme

AISA algoritmasında yeni arama ajanlarının oluşturulması, pek çok metasezgisel algıtmada olduğu gibi kullanıcı tarafından önceden tanımlanmış bir durdurma kriteri sağlanıncaya kadar iteratif olarak gerçekleştirilir. Burada amaç, iteratif bir süreç boyunca çözüm vektörlerini geliştirmektir.

Daha önce de belirtildiği üzere akran grubundaki ergenlerin kimlik keşfi sırasında genel olarak üç davranış biçimi ortaya çıkmaktadır. Her bir ergenin her yinelemede bu üç durumdan yalnızca birini rastgele seçtiği varsayılmıştır. Bu seçim bireyin dürtülerine dayalı dinamik bir seçim stratejisine sahip olduğu anlamına gelir. Akran grubundaki ergenlerin kimlik oluşumu üç durum altında aşağıda verildiği gibi

matematiksel olarak modellenmiştir.

**Durum 1.** Bir ergen, akran grubunun davranışlarını, değerlerini, inançlarını ve tutumlarını gözlemleyerek ve ardından akıl yürüterek kendi kimliğini oluşturabilir. Ergenlerin gruptaki en iyi özellikleri (en iyi saç modeli, en çalışkan, en iyi futbol becerileri vb.) belirleyerek onları taklit edebileceği varsayılmıştır. Algoritmaya özgü grubun en iyi kimlik özellikleri, ortogonal bir fonksiyon yaklaşımı kullanan bir tür özellik seçim yapısı ile bulunmaktadır.

Her bir iterasyonda mevcut kimlik vektörleri ve bunların uygunluk değerleri arasında yaklaşık bir model oluşturulur. Bu model, kısmen kara kutu optimizasyon probleminin çevrimiçi bir modellemesi olarak düşünülebilir. Yaklaşık modelde bireylerin kimlikleri ayrı ayrı model girdileri, uygunluk değerleri ise model çıktıları olarak kullanılmaktadır. Ortogonal yaklaşıklama fonksiyon modeli kestirildikten sonra, kısmi uygunluk değerleri olarak ifade edilen tüm kimlik özelliklerinin amaç fonksiyonuna katkıları ayrı ayrı hesaplanmaktadır. Tüm bireylerin herhangi bir kimlik özelliğine karşılık gelen kısmi uygunluk değerleri arasından, problem en küçükleme ise en küçük uygunluk değerine, en büyükleme ise de en büyük uygunluk değerine karşılık gelen kimlik özelliği en iyi kimlik özelliği olarak belirlenir. Daha sonra bu prosedür tüm kimlik özelliklerine teker teker uygulanarak en iyi kimlik yapısı oluşturulur. Modelleme sürecinde fonksiyon yaklaşımı performansının özellik seçimi için doğrudan bir önemi olmadığını belirtmek gerekir. Bunun yerine asıl görev, bireyleri problemin yerel çözümüne yönlendiren en iyi kimlik yapısını oluşturmaktır.

Bireyler ve uygunluk değerleri arasında yaklaşık bir model oluşturmak için En Küçük Kareler Tahmin (Least Squares Estimation - LSE) yöntemi ile parametreleri kestirilen bir Chebyshev fonksiyonel bağlantı ağı (Chebyshev functional-link network - CFLN) kullanılmıştır. İyi bilinen ortogonal fonksiyon yaklaşıklama modellerinden biri olan bu ağ, özellikle çevrimiçi problemler için hızlı ve doğru yaklaşıklama yeteneğine sahiptir (Çetin ve diğ. 2019). Ek olarak modelin nihai çıktısı girdi değişkenlerinin kısmi çıktılarının doğrusal kombinasyonu olarak ifade edildiğinden bu



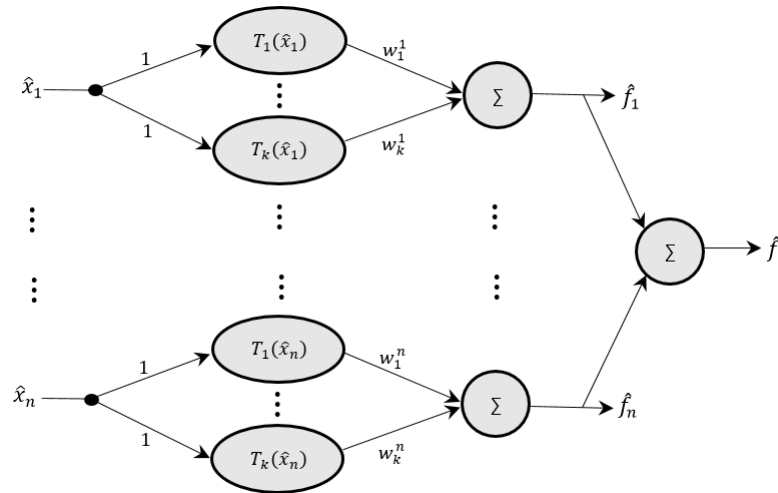
ağ yapısı kısmi uygunluk değerlerinin bulunmasına izin vermektedir.

Chebyshev polinomları ortogonal olmaları, yinelemeli ilişkilere sahip olmaları ve  $[-1, 1]$  tanım aralığında yoğun bir alanı kapsamaları nedeniyle fonksiyon yaklaşıklama için oldukça uygundur (Lee ve Jeng 1998; Patra ve Kot 2002).  $x \in [-1, 1]$  için  $\{T_k(x)\}_{k=0,1,2,\dots}$  Chebyshev polinomları,

$$T_k(x) := \begin{cases} 1, & ,k = 0 \\ x, & ,k = 1 \\ 2xT_{k-1}(x) - T_{k-2}(x), & ,k \geq 2 \end{cases} \quad (2.6)$$

rekürans bağıntısı ile tanımlanır ve özyinelemeli olarak kolaylıkla elde edilirler. Burada  $k$  Chebyshev polinomunun derecesini ifade etmektedir.

Kısmi uygunluk değerlerini bulmak için Çetin ve diğ. (2019) tarafından geliştirilmiş olan yüksek performanslı modelleme yeteneğine sahip bir Chebyshev fonksiyonel bağlantı ağ modelinin sabit terimsiz versiyonu kullanılmıştır. Bu ağ modeli optimizasyon problemini iyi bir doğrulukla yaklaştığı, iteratif bir prosedür gerektirmediği ve her girdi değişkeninin kısmi uygunluk değerini bulmaya imkan sağladığı için tercih edilmiştir. Şekil 2.1'de çok girişli-tek çıkışlı ileri yönlü ortogonal Chebyshev ağı görülmektedir.



**Şekil 2.1:** Önerilen ortogonal Chebyshev ağı

Modelin girdisini oluşturacak kimlik örneklerinin (popülasyon) değerleri

herhangi bir sonlu aralıkta olabileceğinden, ağ modelinde baz fonksiyonu olarak kullanılan Chebysyhev polinomları için, örneklerin  $[-1, 1]$  aralığına dönüştürülmeleri gerekir. Bu nedenle modelleme sürecinde ilk olarak modelin girdisini oluşturan kimlik örnekleri  $[-1, 1]$  aralığında aşağıdaki eşitlik kullanılarak normalleştirilmiştir,

$$\hat{x}_j^i = 2 \frac{(x_j^i - \underline{b}_j)}{(\overline{b}_j - \underline{b}_j)} - 1, \quad i = 1, 2, \dots, N; \quad j = 1, 2, \dots, n. \quad (2.7)$$

burada  $\hat{x}_j^i$ ,  $i$ . ergenin  $j$ . kimlik özelliğinin normalleştirilmiş değerini ifade etmektedir ve Şekil 2.1'de görüldüğü gibi her girdi vektörü  $\{\hat{x}^i\}_{i=1,2,\dots,N}$  Chebysyhev polinomlarına doğrudan uygulanır. Normalleştirilmiş girişler aşağıdaki matris formunda gösterilmiştir:

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{x}_1^1 & \hat{x}_2^1 & \dots & \hat{x}_n^1 \\ \hat{x}_1^2 & \hat{x}_2^2 & \dots & \hat{x}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_1^N & \hat{x}_2^N & \dots & \hat{x}_n^N \end{bmatrix}_{N \times n} \quad (2.8)$$

burada  $\hat{\mathbf{X}}$  ortogonal ağ modeline uygulanacak giriş matrisidir. Normalleştirme işleminden sonra, model girdileri doğrudan Chebysyhev polinomlarına uygulanır, regresyon matrisi ( $\Psi$ ) ve her giriş elemanına ait altregresyon vektörleri ( $\psi$ )

$$\begin{aligned} \Psi &= \begin{bmatrix} T_1(\hat{x}_1^1) & \dots & T_k(\hat{x}_1^1) & \dots & \dots & T_1(\hat{x}_n^1) & \dots & T_k(\hat{x}_n^1) \\ T_1(\hat{x}_1^2) & \dots & T_k(\hat{x}_1^2) & \dots & \dots & T_1(\hat{x}_n^2) & \dots & T_k(\hat{x}_n^2) \\ \vdots & \ddots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ T_1(\hat{x}_1^N) & \dots & T_k(\hat{x}_1^N) & \dots & \dots & T_1(\hat{x}_n^N) & \dots & T_k(\hat{x}_n^N) \end{bmatrix}_{N \times (n \times k)} \\ &= \begin{bmatrix} \psi_1^1 & \dots & \dots & \psi_n^1 \\ \psi_1^2 & \dots & \dots & \psi_n^2 \\ \vdots & \ddots & \ddots & \vdots \\ \psi_1^N & \dots & \dots & \psi_n^N \end{bmatrix}. \end{aligned} \quad (2.9)$$

şeklinde elde edilir.

Şekil 2.1'deki ağırlık faktörleri, yani yaklaşık modelin parametreleri mevcut iterasyondaki hem regresyon matrisi ( $\Psi$ ) hem de uygunluk vektörü ( $f$ ) kullanılarak en

küçük kareler tahmin yöntemiyle aşağıdaki gibi kestirilir,

$$\begin{aligned}\hat{\mathbf{w}} &= (\mathbf{\Psi}^T \mathbf{\Psi})^{-1} \mathbf{\Psi}^T \mathbf{f} \\ &= [w_1^1 \ \cdots \ w_k^1 \ \cdots \ \cdots \ w_1^n \ \cdots \ w_k^n]_{1 \times (n \times k)}^T \\ &= [\mathbf{w}^1 \ \cdots \ \cdots \ \mathbf{w}^n]^T\end{aligned}\quad (2.10)$$

burada  $\mathbf{w}^j \in \mathfrak{R}^{1 \times k}$ ,  $j$ . girişin ağırlık vektörüdür.

Eşitlik (2.8)'de verilen her bir bireyin kimlik özelliğini belirten matrisin her bir elemanının eşitlik (2.11) ile kısmi uygunluk değeri hesaplanır ve bulunan değerler eşitlik (2.12)'deki matriste tutulur.

$$\hat{f}_j^i = \mathbf{\Psi}_j^i \mathbf{w}^i. \quad (2.11)$$

$$\hat{\mathbf{F}} = \begin{bmatrix} \hat{f}_1^1 & \hat{f}_2^1 & \cdots & \hat{f}_n^1 \\ \hat{f}_1^2 & \hat{f}_2^2 & \cdots & \hat{f}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{f}_1^N & \hat{f}_2^N & \cdots & \hat{f}_n^N \end{bmatrix}_{N \times n} \quad (2.12)$$

Son olarak eşitlik (2.12)'deki kısmi uygunluk matrisinin ( $\hat{\mathbf{F}}$ ) her bir sütununda ( $j = 1, 2, \dots, n$ ) en küçük değere karşılık gelen satır indisi ( $m^j$ ) bulunur ve popülasyon matrisinin ( $\mathbf{X}$ )  $j$ . sütununun  $m^j$ . satırındaki eleman  $j$ . en iyi kimlik özelliği ( $x_j^*$ ) olarak belirlenir (eşitlik (2.13)).

$$x_j^* = x_j^{m^j}, \quad m^j = \arg \min_l \{ \hat{f}_j^l \mid l = 1, 2, \dots, N \}, \quad \forall j \quad (2.13)$$

Bu durumda  $i$ . ergenin yeni kimliği ( $\mathbf{x}_{new}^i$ ) aşağıdaki gibi elde edilir:

$$\mathbf{x}_{new}^i = \mathbf{x}^i - r_1 (\mathbf{x}^i - \mathbf{x}^*) \quad (2.14)$$

burada  $r_1$ , 0 ile 1 arasında rastgele üretilen bir sayıdır.  $\mathbf{x}^*$  ise akran grubunun sahip olduğu en iyi kimlik özellikleri dizisini temsil etmektedir. Eşitlik (2.14), ergenin akran grubundan iyi davranışları gözlemleyerek daha nitelikli yeni kimlikler edinmeye çalıştığını gösterir.

**Durum 2.** Ergen, kimliğini yüksek statüye, güce ve prestije sahip algıladığı bir rol modeli taklit ederek oluşturabilir. Buradan yola çıkarak akran grubundaki yüksek

statüye, güce ve prestije sahip birey, popülasyondaki en iyi uygunluk değerine sahip birey olarak kabul edilmiş olup, rol model olarak seçilmiştir. Ergenin rol modeli taklit ederek oluşturacağı yeni kimliği aşağıdaki gibi elde edilmektedir:

$$\mathbf{x}_{new}^i = \mathbf{x}^i - r_2 (\mathbf{x}^p - \mathbf{x}^{rm}) \quad (2.15)$$

burada  $r_2$ ,  $[0, 1]$  aralığında üretilen rastgele bir sayıdır.  $\mathbf{x}^{rm}$  ise rol modeli yani popülasyondaki en iyi bireyi belirtmektedir.  $p \neq rm$  olmak üzere,  $\mathbf{x}^p$  popülasyondan rastgele seçilen  $p$ . bireydir. Eşitlik (2.15)'teki  $(\mathbf{x}^p - \mathbf{x}^{rm})$  ifadesi olgunlaşmamış yakınsamayı engellemektedir. Bu ifadenin, Cheng ve Prayogo (2014) tarafından geliştirilmiş olan Simbiyotik Organizmalar Arama (Symbiotic Organisms Search - SOS) algoritması ile benzerlik gösterdiğini belirtmekte fayda vardır.

**Durum 3.** Ergen birey sigara kullanımı, madde kullanımı, erken cinsel davranış ve zorbalık gibi olumsuz/istenmeyen kimlik seçeneklerini benimseyebilir. Algoritmanın yüksek keşif kabiliyetine sahip olabilmesi için olumsuz/istenmeyen kimlik özelliğinin ( $x^u$ ) popülasyon matrisinden rastgele seçilen bir eleman olduğu varsayılmıştır. Bu durumda,  $i$ . ergenin yeni kimliği şu şekilde elde edilmektedir:

$$\mathbf{x}_{new}^i = \mathbf{x}^i - \mathbf{r}_3 (\mathbf{x}^i - \mathbf{x}^q), \quad (2.16)$$

burada  $\mathbf{r}_3$ ,  $1 \times n$  boyutlu ve değerlerini  $[0, 1]$  aralığında alan düzgün dağılımlı rastgele bir satır vektörüdür.  $\mathbf{x}^q$  ise istenmeyen kimlik vektörünü göstermekte olup aşağıdaki gibi ifade edilmiştir:

$$\mathbf{x}^q = [x^u \quad x^u \quad \dots \quad x^u]_{1 \times n}^T. \quad (2.17)$$

Bahsedilen bu üç duruma ait denklemler birleştirildiğinde, önerilen yaklaşımın yeni kimlik oluşturma aşamasının genel ifadesi,

$$\mathbf{x}_{new}^i = \begin{cases} \text{Durum 1:} & \mathbf{x}^i - r_1 (\mathbf{x}^i - \mathbf{x}^*), & r_4 \leq 1/3 \\ \text{Durum 2:} & \mathbf{x}^i - r_2 (\mathbf{x}^p - \mathbf{x}^{rm}), & 1/3 < r_4 \leq 2/3 \\ \text{Durum 3:} & \mathbf{x}^i - \mathbf{r}_3 (\mathbf{x}^i - \mathbf{x}^q), & 2/3 < r_4 \end{cases} \quad (2.18)$$

şeklinde yazılır. Burada  $r_4$ , bireyin dürtülerine dayalı dinamik bir seçim stratejisine sahip olduğunu işaret eden  $[0, 1]$  aralığındaki rastgele bir değişkendir.

Algoritmanın farklı arama yeteneklerine sahip olabilmesi için, algoritma bu üç durum düşünülerek özenle tasarlanmıştır. İlk durumda en iyi kimlik özelliklerine karşı gelen  $\mathbf{x}^*$  vektörünün doğru tahminlenmesi, algoritmanın yerel arama yeteneğini (sömürü) arttırmaktadır. Bu durumda model tahmini ne olursa olsun mevcut popülasyondaki minimum ve maksimum değişken değerleri arasında yeni çözümler elde edildiğini ifade etmek gerekir. İkinci durumda ise en iyi çözüme (rol model) karşılık gelen ( $\mathbf{x}^m$ ) vektörüne doğru bir yönelim söz konusudur. Yani bu durumda algoritma sömürü yapmaya odaklanır. Bu iki durum algoritmanın sömürü yeteneğine hizmet ettiğinden dolayı, ikinci durumda popülasyondan herhangi bir  $\mathbf{x}^p$  bireyi rastgele seçilerek olgunlaşmamış bir yakınsama probleminin oluşmasının önüne geçilmiştir. Son olarak algoritmanın yerel minimumlardan kurtulabilmesi için bir keşif yeteneği üçüncü durum ile sağlanmıştır.

#### **2.2.4 Sınır Kontrol Mekanizması**

Eğer yeni üretilen bir birey verilen arama alanının sınırları dışına çıkarsa, bir sınır kontrol mekanizması aktif hale gelir. Bu mekanizmada değişkenlerden herhangi biri sınır/limit aralığını ihlal ederse, bu değişkenler kendi limit aralıkları içinde rastgele yerleştirilir, (2.3) eşitliğiyle uygun sınır aralığına getirilir. Bu mekanizma mevcut ve gelecek nesillerde etkin aramalara olanak tanıyarak popülasyon çeşitliliğinin artmasını sağlamaktadır. Bu sınır kontrol mekanizmasının Civicioglu (2013) tarafından geliştirilen Geri-izleme Arama Optimizasyon Algoritmasında (Backtracking Search Optimization Algorithm - BSA) kullanıldığını belirtmekte fayda vardır.

#### **2.2.5 Güncelleme Mekanizması**

Güncelleme mekanizması, yeni kimliğin yani yeni oluşturulan arama ajanının popülasyona dahil edilip edilmeyeceğine karar vermek için kullanılır. Güncelleme mekanizması uygulanmadan önce ergenin yeni kimliğinin uygunluk değeri hesaplanır. Yeni kimlik mevcut kimlikten daha iyiye, mevcut kimlik popülasyondan çıkarılır

ve bunun yerine yeni kimlik popülasyona dahil edilir. Aynı prosedür her yeni kimlik için ayrı ayrı uygulanır. Böylece arama ajanları sonraki iterasyonlarda kullanılmak üzere gelişirler. Birçok metasezgisel her yinelemenin sonunda benzer bir güncelleme mekanizması kullanırken, AISA her fonksiyon değerlendirmesinde güncelleme mekanizmasını kullanmaktadır. Burada asıl amaç, daha nitelikli bir çözümü popülasyona hemen dahil ederek çözümlerin kalitesini arttırmaktır.

### **2.2.6 Durdurma/Sonlandırma Kriteri**

Genel olarak optimizasyon algoritmaları, amaç veya hata fonksiyonu toleransı, maksimum fonksiyon değerlendirme/işlem sayısı, maksimum çalışma süresi ve maksimum iterasyon sayısı gibi bir veya birden fazla farklı durdurma kriterinin bir kombinasyonunu kullanarak arama görevini tamamlarlar. Bu çalışmada ergenlik dönemi iteratif/yinelemeli süreçle ilişkilendirilerek, maksimum iterasyon sayısı ( $Max_{iter}$ ) ergenlik sürecinin bitişi anlamına gelen sonlandırma kriteri olarak kabul edilmiştir.

AISA'nın sözde kodu Algoritma 1'de verilmiştir.

---

**Algoritma 1** Ergen Kimlik Arama Algoritmasının (AISA) sözde kodu

---

```
1: Başla:
2: Algoritmaya özgü parametreleri ( $N, k, Max_{iter}$ ) belirle
3: Eşitlik (2.3)'ü kullanarak başlangıç popülasyonunu ( $[\mathbf{X}]_{N \times n}$ ) oluştur
4: Her bireyin uygunluk değerini hesapla
5: while durdurma kriteri sağlanmadığı sürece do
6:   Eşitlik (2.7)'yi kullanarak eşitlik (2.8)'deki normalize edilmiş  $\hat{\mathbf{X}}$  matrisini
   oluştur
7:   Eşitlik (2.9) ile  $\Psi$  regresyon matrisini ve onun altregresyon vektörlerini
   ( $\Psi_1^1, \dots, \Psi_n^N$ ) oluştur
8:   Eşitlik (2.10) ile ağırlık vektörlerini ( $\mathbf{w}^1, \dots, \mathbf{w}^n$ ) hesapla
9:   Eşitlik (2.11) ile kısmi uygunluk matrisini ( $\hat{\mathbf{F}}$ ) hesapla
10:  Eşitlik (2.13) ile en iyi özellik vektörünü ( $\mathbf{x}^*$ ) bul
11:  for  $i = 1$  to  $N$  do
12:     $r_4 \sim U(0, 1)$  parametresini güncelle
13:    if  $r_4 \leq 1/3$  then
14:       $r_1 \sim U(0, 1)$  parametresini güncelle
15:       $\mathbf{x}_{new}^i = \mathbf{x}^i - r_1 (\mathbf{x}^i - \mathbf{x}^*)$ 
16:    else if  $r_4 > 1/3 \wedge r_4 \leq 2/3$  then
17:       $r_2 \sim U(0, 1)$  parametresini güncelle
18:      Rol modeli/en iyi çözümü ( $\mathbf{x}^m$ ) bul
19:      Bireylerden birini rastgele seç  $p \mid p \neq rm$ 
20:       $\mathbf{x}_{new}^i = \mathbf{x}^i - r_2 (\mathbf{x}^p - \mathbf{x}^m)$ 
21:    else
22:       $\mathbf{r}_3 \sim U(0, 1)^n$  vektörünü güncelle
23:      Negatif kimlik vektörünü ( $\mathbf{x}^q$ ) oluştur
24:       $\mathbf{x}_{new}^i = \mathbf{x}^i - \mathbf{r}_3 (\mathbf{x}^i - \mathbf{x}^q)$ 
25:    end if
26:    Sınır kontrol mekanizmasını uygula
27:    Güncelleme mekanizmasını uygula
28:  end for
29: end while
30: En iyi çözümü çıktı olarak ver
31: Bitir
```

---

### 2.3 AISA'nın Diğer Metasezgiseller ile Yapısal Karşılaştırması

Doğadan ilham alınarak geliştirilen yeni metasezgisel optimizasyon algoritmalarının sayıları literatürde çarpıcı bir şekilde artmaktadır. Optimizasyon literatüründeki bazı çalışmalarda, görünüşte çekirdek/klasik metasezgisel algoritmalara benzeyen şüpheli, sözde yeni algoritmaların geliştirildiği tartışılmıştır

(Sörensen 2015; Weyland 2010; Molina ve diğ. 2020; Del Ser ve diğ. 2019). Bu çalışmalar, yeni algoritmaların birçoğunun aslında yeni bir algoritma olmadığını, çekirdek algoritmalarından esinlenilerek onların değiştirilmiş versiyonları olduğunu ileri sürmüşlerdir. Molina ve diğ. (2020) çalışmalarında metasezgiseller ile ilgili kapsamlı bir karşılaştırma yapmış olup, birçok yeni metasezgiselin GA, ACO, PSO, DE ve ABC gibi çekirdek algoritmaların yeni sürümleri olarak kabul edilebileceğini belirtmişlerdir. Yang (2020) tarafından yapılan çalışmada ise algoritmalar prosedür-tabanlı ve denklem-tabanlı olmak üzere iki kategoriye ayrılarak karşılaştırılmıştır. Birçok soyut karşılaştırmanın aksine bahsedilen çalışmada, denklem-tabanlı algoritmaların hem somut hem de daha anlaşılır bir şekilde karşılaştırmaları matematiksel olarak yapılmıştır. Bu tartışmalar ışığında tez kapsamında önerilen algoritmanın diğer algoritmalarından farklı ve yeni bir algoritma olduğunu göstermek amacıyla, DE, PSO ve ABC çekirdek algoritmaları ile arama mekanizmaları ve matematiksel temelleri açısından karşılaştırma yapılmıştır. Bu algoritmaların AISA ile karşılaştırmada tercih edilmesinin nedeni tamamının denklem-tabanlı olmasıdır.

Genel olarak denklem-tabanlı metasezgisel optimizasyon algoritmaları, gradyan tabanlı optimizasyon algoritmalarında olduğu gibi vektör doğrusal kombinasyonunu kullanırlar ve yinelemeli güncelleme denklemleri aşağıdaki gibi ifade edilir:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta \mathbf{x}_i^t \quad (2.19)$$

burada  $\mathbf{x}_i^t$ ,  $\mathbf{x}_i^{t+1}$  ve  $\Delta \mathbf{x}_i^t$ ; sırasıyla mevcut çözüm, yeni çözüm ve arama (değişim veya mutasyon) vektörleridir.

Matematiksel açıdan AISA'nın DE, PSO ve ABC algoritmalarından farklı olduğunu göstermek amacıyla bu dört algoritmanın çözüm güncelleme denklemleri kısaca aşağıdaki gibi tanımlanmıştır.

- Diferansiyel Evrim (DE): Bu algoritmanın ana mutasyon denklemi

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + F(\mathbf{x}_j^t - \mathbf{x}_k^t) \quad (2.20)$$



şeklinde olup, (2.19) formunda alttaki gibi yazılabilir.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta \mathbf{x}_i^t, \quad \Delta \mathbf{x}_i^t = F(\mathbf{x}_j^t - \mathbf{x}_k^t) \quad (2.21)$$

$\mathbf{x}_i^t$ ,  $\mathbf{x}_j^t$  ve  $\mathbf{x}_k^t$  popülasyondan seçilen üç ayrı çözüm vektörüdür.  $F \in (0,2)$  ise ölçeklendirme faktörü olarak ifade edilen bir kontrol parametresidir. Mutasyon işleminden sonra hedef/ebeveyn vektör ile mutasyona uğramış vektörün çaprazlama işlemi gerçekleştirilir.

- Parçacık Sürü Optimizasyonu (PSO): Eşitlik (2.19) formunda,  $t$ . iterasyondaki  $i$ . parçacığın hız vektörü ( $\mathbf{v}_i^t$ ) ve konum vektörü ( $\mathbf{x}_i^t$ )

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + \Delta \mathbf{v}_i^t \quad (2.22)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta \mathbf{x}_i^t \quad (2.23)$$

denklemleri ile iteratif olarak güncellenir. Burada  $w$  atalet ağırlığıdır ve

$$\Delta \mathbf{x}_i^t = \mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + \Delta \mathbf{v}_i^t \quad (2.24)$$

olup hız değişim vektörü

$$\Delta \mathbf{v}_i^t = c_1 r_1 (\mathbf{xBest}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{gBest}_i^t - \mathbf{x}_i^t) \quad (2.25)$$

şeklinde yazılabilir.  $\mathbf{xBest}_i^t$  ve  $\mathbf{gBest}_i^t$  sırasıyla ilgili parçacığa ait o ana kadarki en iyi konumu ve popülasyonun en iyi parçacığının konumunu ifade eden vektörlerdir.  $r_1$  ve  $r_2$  değerlerini  $[0, 1]$  aralığından alan rassal değişkenlerdir.  $c_1$  ve  $c_2$  ise ivme katsayılarıdır.

- Yapay Arı Kolonisi (ABC): Bu algoritma işçi, gözcü ve kâşif arılar olmak üzere üç kaskat aşamalı bir yapıya sahiptir. İşçi arıların yeni besin kaynaklarına doğru hareket denklemi

$$\mathbf{x}_{ij}^{t+1} = \mathbf{x}_{ij}^t + \phi_{ij}(\mathbf{x}_{ij}^t - \mathbf{x}_{kj}^t) \quad (2.26)$$

şeklindedir. (2.19) formunda

$$\mathbf{x}_{ij}^{t+1} = \mathbf{x}_{ij}^t + \Delta \mathbf{x}_{ij}^t, \quad \Delta \mathbf{x}_{ij}^t = \phi_{ij}(\mathbf{x}_{ij}^t - \mathbf{x}_{kj}^t) \quad (2.27)$$

yazılabilir. Bu denklemlerde  $k \neq i$  olmak üzere  $k$  ve  $j$  rastgele seçilen indis değerleridir.  $\phi_{ij}$  ise  $[-1,1]$  aralığında düzgün dağılımlı rastgele bir değişkendir. Rulet tekerliği gibi bir seçim mekanizmasına sahip olan gözcü arılar, yine (2.26)'deki denklemi kullanarak konumlarını güncellerler. Ayrıca ABC'de popülasyonu çeşitlendirmek amacıyla önceden tanımlanmış *Limit* parametresi ile kontrol edilen bir kâşif arı aşaması bulunur (Karaboga ve Basturk 2007).

- Ergen Kimlik Arama Algoritması (AISA): AISA'nın çözüm vektörleri, aşağıda verilen eşitlikler kullanılarak yinelemeli olarak güncellenmektedir.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta \mathbf{x}_i^t \quad (2.28)$$

burada

$$\Delta \mathbf{x}_i^t = \begin{cases} r_1 (\mathbf{x}_*^t - \mathbf{x}_i^t), & r_4 \leq 1/3 \\ r_2 (\mathbf{x}_{rm}^t - \mathbf{x}_p^t), & 1/3 < r_4 \leq 2/3 \\ r_3 (\mathbf{x}_q^t - \mathbf{x}_i^t), & 2/3 < r_4 \end{cases} \quad (2.29)$$

şeklinde.  $r_1, r_2$  ve  $r_4$ ;  $[0,1]$  aralığındaki düzgün dağılımlı rastgele sayılardır.  $r_3$  değerlerini  $[0,1]$  aralığında alan düzgün dağılımlı rastgele üretilmiş bir vektördür.  $\mathbf{x}_{rm}^t$  global en iyi çözüm/rol modeli belirtmekte,  $rm \neq p$  olmak üzere  $\mathbf{x}_p^t$  popülasyondan rastgele seçilen bir çözümdür. Son olarak  $\mathbf{x}_*^t$  ve  $\mathbf{x}_q^t$  popülasyondaki çözüm vektörlerinden farklı iki vektördür ve bu vektörler bölüm 2.2'de detaylı olarak anlatılmıştır.

Metasezgiseller, genellikle çözüm güncelleme stratejilerine göre farklılık gösterirler. AISA algoritması DE, PSO ve ABC algoritmalarından aşağıdaki yönleriyle farklılık gösterir:

- Eşitlikler (2.21), (2.23), (2.27) ve (2.28)'de verilmiş olan  $\mathbf{x}_i^{t+1}$  ifadesi tüm algoritmalarda benzer görünmesine rağmen, tüm algoritmaların sahip olduğu arama vektörü ( $\Delta \mathbf{x}_i^t$ ) birbirinden farklıdır. Yani tüm algoritmaların popülasyon matrisleri aynı olsa bile, bir sonraki iterasyondaki popülasyon matrisleri birbirlerinden çok

büyük olasılıkla farklı olacaktır. AISA'nın arama vektörleri çözüm uzayını farklı şekilde gerdiğinden matematiksel olarak diğer algoritmalarından ayrıldığı açıkça ortadadır.

- Metasezgisellerin çoğu sömürü süreçlerinde yeni arama ajanlarını oluşturmak için global en iyi çözümü kullanırlar. Aksine AISA, ayırt edici ve kendine özgü bir özelliği olan ortogonal bir fonksiyon yaklaşımı ile bulunan bir rehber vektörü ( $\mathbf{x}_*^t$ ) kullanır.
- AISA ve PSO algoritmaları güncelleme denklemlerinde global en iyi çözüm kullanırlar. PSO'da  $i$ . parçacığın yeni konumu  $\mathbf{xBest}_i^t$  ve  $\mathbf{gBest}_i^t$  vektörleri tarafından yönetilir ve eşitlik (2.25)'de görüldüğü gibi her ikisinin de kümülatif etkisi dikkate alınır. Fakat AISA'da  $\mathbf{x}_{rm}^t$  ve bir başka çözüm vektörü ( $\mathbf{x}_p^t$ ) kullanılarak güncelleme yapılır.
- Metasezgisellerin çoğu çözüm çeşitliliğini arttırmak için bazı çözümleri rastgele arama uzayına yerleştiren bir keşif stratejisine sahipken, AISA bu çeşitliliği rastgele üretilmiş bir vektörü ( $\mathbf{r}_3$ ) kullanarak  $\mathbf{x}_q^t$  vektörüne doğru bir yönelim ile sağmaya çalışır.
- DE sadece popülasyondaki çözüm vektörlerini kullanarak mutasyon işlemini gerçekleştirir; ancak AISA, popülasyondaki çözüm vektörlerinden farklı iki vektörü ( $\mathbf{x}_*^t$  ve  $\mathbf{x}_q^t$ ) kullanmaktadır. Ek olarak AISA, DE'de bulunan bir çaprazlama operatörü kullanmamaktadır.
- ABC algoritmasında çözüm matrisi yani popülasyon, işçi ve gözcü arılar için ikiye ayrılır. AISA'da ise böyle bir durum söz konusu değildir. Ayrıca ABC; işçi, gözcü ve kâşif arılar olmak üzere üç kaskat aşamaya sahipken, AISA rastsal bir seçim stratejisiyle kontrol edilen kaskat olmayan üç mekanizmayı kullanmaktadır.
- AISA algoritması, sınır dışına çıkan değişkenlerin sınır aralığı içine rastgele yerleştirilmesi yönüyle, DE, PSO ve ABC algoritmalarından farklıdır.

### 3. STANDART TEST FONKSİYONLARI ÜZERİNDE YAPILAN DENEYSEL ÇALIŞMALAR

Bu bölümde, tez kapsamında kullanılan test fonksiyonları ve önerilen algoritma ile performansları kıyaslanan gelişmiş ve yaygın kullanılan metasezgisel algoritmaların kısa bir açıklaması yer almaktadır. Bölümün devamında ise hesaplamalı deney sonuçları verilmiş olup, sonuçlar analiz edilmiştir.

#### 3.1 Standart Kıyaslama Test Fonksiyonları

Önerilen algoritmanın uygulanabilirliğini ve etkinliğini kanıtlamak amacıyla, algoritma Tablolar 3.1–3.5’te verilen 39 standart test problemi üzerinde sınanmıştır. Bu tablolarda  $n$  fonksiyonun boyutunu (karar değişkeni sayısını), *Aralık* fonksiyonun araştırma uzayını ve  $F_{min}$  minimize edilecek fonksiyonunun optimum değerini ifade etmektedir. Bu test problemleri karakteristik özelliklerine göre tek-modlu (unimodal), çok-modlu (multimodal), sabit boyutlu çok-modlu (fixed-dimension multimodal), bileşik/karma/kompozit (composite) ve CEC 2019 olmak üzere beş kategoride ele alınmıştır.

İlk kategori olan tek-modlu test problemleri, Tablo 3.1’de listelenen yedi fonksiyondan ( $F_1 - F_7$ ) oluşmakta ve iki boyutlu grafikleri Şekil 3.1’de görüldüğü gibidir. Bu test problemleri tek bir global optimum değere ve geniş bir arama alanına sahip olduklarından, algoritmaların sömürü ve yakınsama hızlarını karşılaştırmak için oldukça uygundur.

İkinci kategoride yer alan çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) Tablo 3.2’de, iki boyutlu grafikleri ise Şekil 3.2’de verilmiştir. Bu fonksiyonlar çok sayıda yerel optimum değere sahip olmaları nedeniyle, algoritmaların yerel optimuma takılmama ve keşif performanslarını değerlendirmek için kullanılmaktadırlar.

Üçüncü kategoride Tablo 3.3’te verilen ve on fonksiyondan ( $F_{14} - F_{23}$ ) oluşan sabit boyutlu çok-modlu test problemleri yer almaktadır. Grafikleri Şekil 3.3’te verilen

bu fonksiyonlarda kullanılan sabit parametrelerin detaylı anlatımı için Rashedi ve diğ. (2009) incelenebilir. Bu grupta yer alan test fonksiyonları, birçok yerel optimum çözüme ve çok-modlu test fonksiyonları gibi tek bir global optimum çözüm içermeleri nedeniyle bir algoritmanın keşif kabiliyetini incelemek için oldukça elverişlidirler.

Dördüncü grup, CEC 2005 özel bölümünden (Liang ve diğ. (2005)) seçilen altı adet test fonksiyonundan oluşmaktadır. Bu test fonksiyonları, daha zorlu problemler oluşturmak için klasik fonksiyonların kaydırılmış, döndürülmüş, genişletilmiş ve birleştirilmiş türleridir. Bu karmaşık fonksiyonları oluşturan klasik fonksiyonlar, rastgele kaydırılmış, uzatılıp/kısaltılmış ve Salomon yöntemini kullanarak (Salomon (1996)) elde edilen ortogonal rotasyon matrislerine göre döndürülmüş ve bu fonksiyonların Gauss fonksiyonlarıyla da yayılma aralığı (coverage range) kontrol edilmiştir. Bu grupta ele alınan fonksiyonlar Tablo 3.4'te verilmiş olup,  $f_1, f_2, \dots, f_{10}$  birleştirme için kullanılan bireysel fonksiyonları göstermektedir.  $\sigma$  ve  $\lambda$  sırasıyla her bir bireysel fonksiyonun yayılma aralığını ve uzatma ( $\lambda > 1$ ) ya da kısaltma ( $\lambda < 1$ ) faktörünü göstermektedir. Tez kapsamında dikkate alınan fonksiyonların iki boyutlu grafikleri Şekil 3.4'te verildiği gibi olup, bu fonksiyonlara ait detaylı bilgi için Liang ve diğ. (2005) incelenebilir ve Matlab kodları <http://www.ntu.edu.sg/home/epnsugan/> bağlantısından indirilebilir.

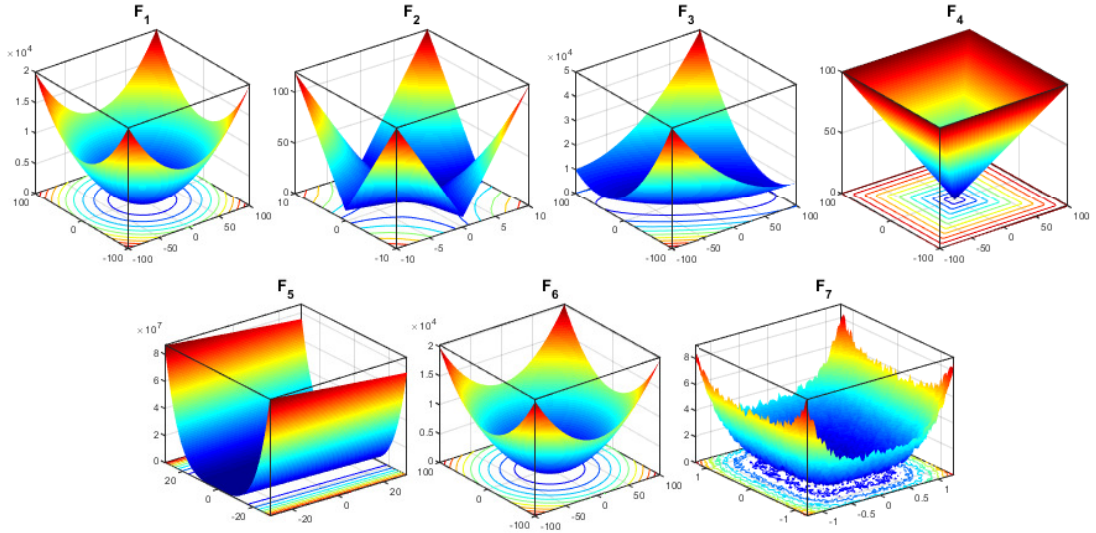
Son grup CEC 2019 test problemleri ise “100 Basamak Problemi (The 100-Digit Challenge)” olarak bilinen 10 modern fonksiyondan oluşmaktadır. Bu tek amaçlı test problemleri çok-modlu olma ve ayrışamama (non-separable) gibi farklı özellikler içeren karmaşık özellikleri bakımından oldukça iyi tasarlanmış fonksiyonlardır. Bu fonksiyonlar ( $F_{30} - F_{39}$ ) Tablo 3.5'te kısaca verilmiş olup, detaylı matematiksel ifadeleri ve karakteristikleri için Price ve diğ. (2018) incelenebilir. Şekil 3.5'te yer alan iki boyutlu grafiklerde bu 10 fonksiyonun son 7 tanesi yer almakta olup, bu fonksiyonlar temel fonksiyonların kaydırılmış ve döndürülmüş versiyonlarıdır. Yer almayan diğer 3 fonksiyon ise tamamen parametre bağımlı oldukları için döndürülmemişler ve iki boyutlu olarak tanımlı değildirler.

**Tablo 3.1:** Tek-modlu test fonksiyonları.

Fonksiyon	$n$	Aralık	$F_{min}$
$F_1(x) = \sum_{i=1}^n x_i^2$	10, 30, 50	$[-100, 100]^n$	0
$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	10, 30, 50	$[-10, 10]^n$	0
$F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	10, 30, 50	$[-100, 100]^n$	0
$F_4(x) = \max_i \{  x_i , 1 \leq i \leq n \}$	10, 30, 50	$[-100, 100]^n$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	10, 30, 50	$[-30, 30]^n$	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	10, 30, 50	$[-100, 100]^n$	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	10, 30, 50	$[-1.28, 1.28]^n$	0

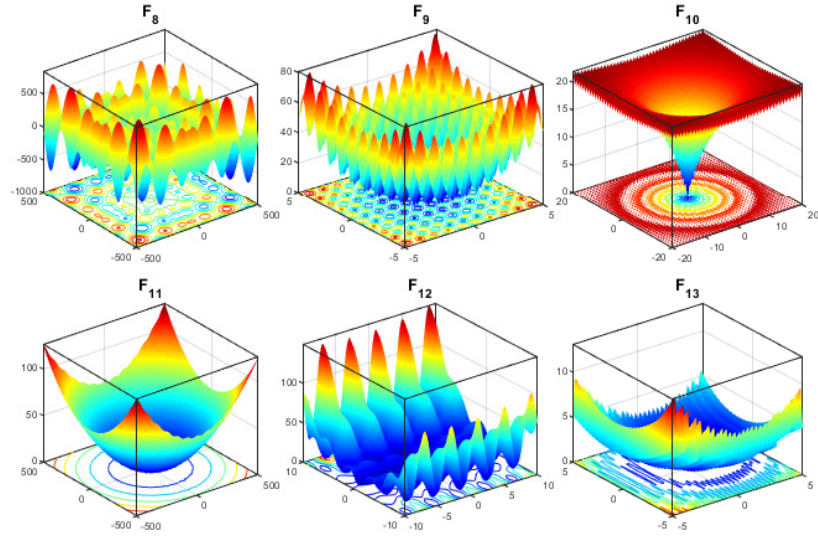
**Tablo 3.2:** Çok-modlu test fonksiyonları.

Fonksiyon	$n$	Aralık	$F_{min}$
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	10, 30, 50	$[-500, 500]^n$	$-418.98 \times n$
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	10, 30, 50	$[-5.12, 5.12]^n$	0
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	10, 30, 50	$[-32, 32]^n$	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10, 30, 50	$[-600, 600]^n$	0
$F_{12}(x) = \frac{\pi}{n} \left\{ \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	10, 30, 50	$[-50, 50]^n$	0
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi y_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi y_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n + 1)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	10, 30, 50	$[-50, 50]^n$	0

**Şekil 3.1:** İki boyutlu tek-modlu test fonksiyonlarının grafikleri.

**Tablo 3.3:** Sabit boyutlu çok-modlu test fonksiyonları.

Fonksiyon	$n$	Aralık	$F_{min}$
$F_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65.536, 65.536]^n$	0.998
$F_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]^n$	0.00030
$F_{16}(x) = 4x_i^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.0316
$F_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10$	2	$[-5, 5]^n$	0.398
$F_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[ 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	2	$[-2, 2]^n$	3
$F_{19}(x) = - \sum_{i=1}^4 c_i \exp \left( - \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	3	$[1, 3]^n$	-3.86
$F_{20}(x) = - \sum_{i=1}^4 c_i \exp \left( - \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$	6	$[0, 1]^n$	-3.32
$F_{21}(x) = - \sum_{i=1}^5 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]^n$	-10.1532
$F_{22}(x) = - \sum_{i=1}^7 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]^n$	-10.4028
$F_{23}(x) = - \sum_{i=1}^{10} \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]^n$	-10.5363



**Şekil 3.2:** İki boyutlu çok-modlu test fonksiyonlarının grafikleri.

**Tablo 3.4:** Bileşik test fonksiyonları.

Fonksiyon	$n$	Aralık	$F_{min}$
$F_{24}(CF_1)$ : $f_1, f_2, \dots, f_{10} =$ Sphere fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10	$[-5, 5]^n$	0
$F_{25}(CF_2)$ : $f_1, f_2, \dots, f_{10} =$ Griewank fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10	$[-5, 5]^n$	0
$F_{26}(CF_3)$ : $f_1, f_2, f_3, \dots, f_{10} =$ Griewank fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1, 1, 1, \dots, 1]$	10	$[-5, 5]^n$	0
$F_{27}(CF_4)$ : $f_1, f_2 =$ Ackley fonksiyonu $f_3, f_4 =$ Rastrigin fonksiyonu $f_5, f_6 =$ Weierstrass fonksiyonu $f_7, f_8 =$ Griewank fonksiyonu $f_9, f_{10} =$ Sphere fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	$[-5, 5]^n$	0
$F_{28}(CF_5)$ : $f_1, f_2 =$ Rastrigin fonksiyonu $f_3, f_4 =$ Weierstrass fonksiyonu $f_5, f_6 =$ Griewank fonksiyonu $f_7, f_8 =$ Ackley fonksiyonu $f_9, f_{10} =$ Sphere fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	$[-5, 5]^n$	0
$F_{29}(CF_6)$ : $f_1, f_2 =$ Rastrigin fonksiyonu $f_3, f_4 =$ Weierstrass fonksiyonu $f_5, f_6 =$ Griewank fonksiyonu $f_7, f_8 =$ Ackley fonksiyonu $f_9, f_{10} =$ Sphere fonksiyonu $[\sigma_1, \sigma_2, \dots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [0.1*1/5, 0.2*1/5, 0.3*5/0.5, 0.4*5/0.5, 0.5*5/100, \dots, 0.6*5/100, 0.7*5/32, 0.8*5/32, 0.9*5/100, 1*5/100]$	10	$[-5, 5]^n$	0

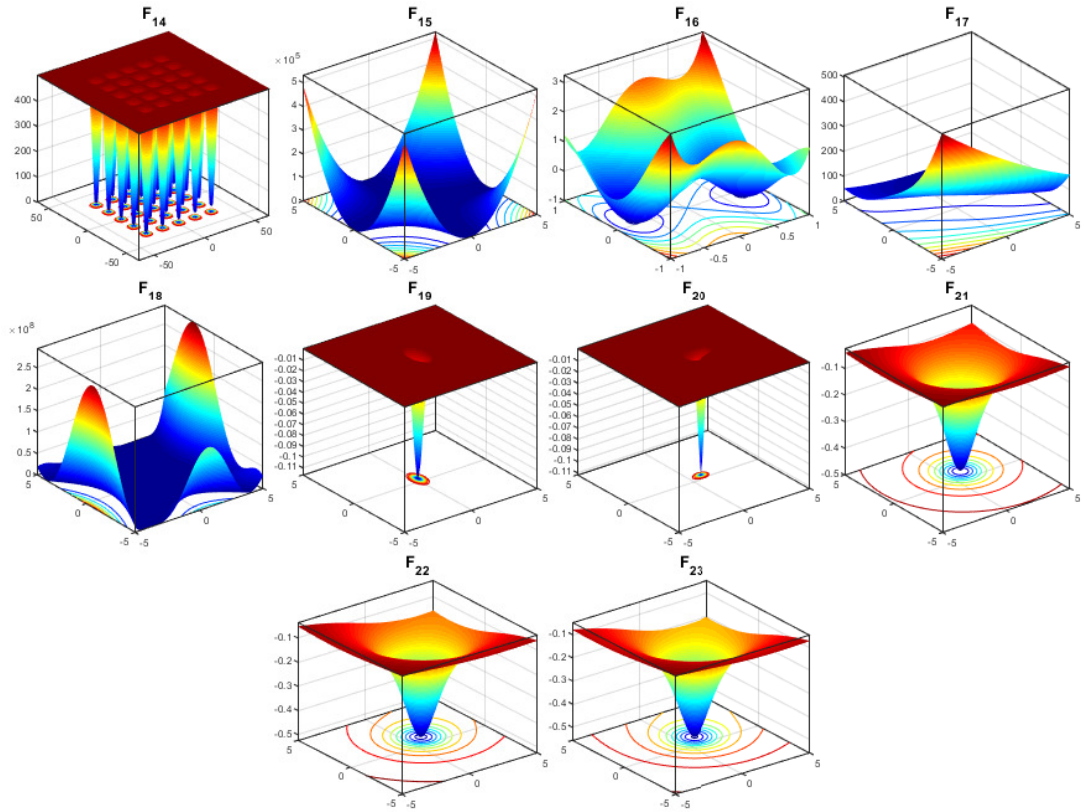
**Not:** Sphere, Rastrigin, Ackley ve Griewank fonksiyonları, Tablolar 3.1 ve 3.2’de verilmiş olan  $F_1(x)$ ,  $F_9(x)$ ,  $F_{10}(x)$  ve  $F_{11}(x)$  fonksiyonlarını belirtmektedir. Weierstrass fonksiyonunun matematiksel formülasyonuna Liang ve diğ. (2005) teknik raporundan ulaşılabilir.

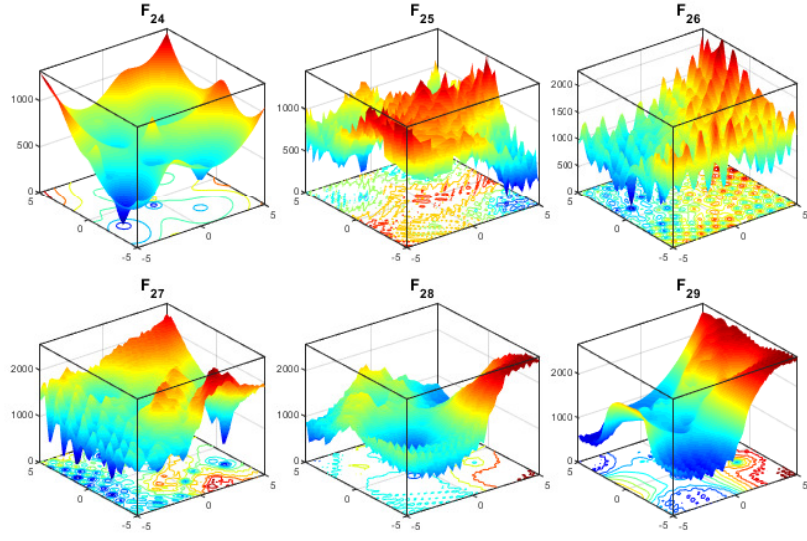


**Tablo 3.5:** CEC 2019 test fonksiyonları.

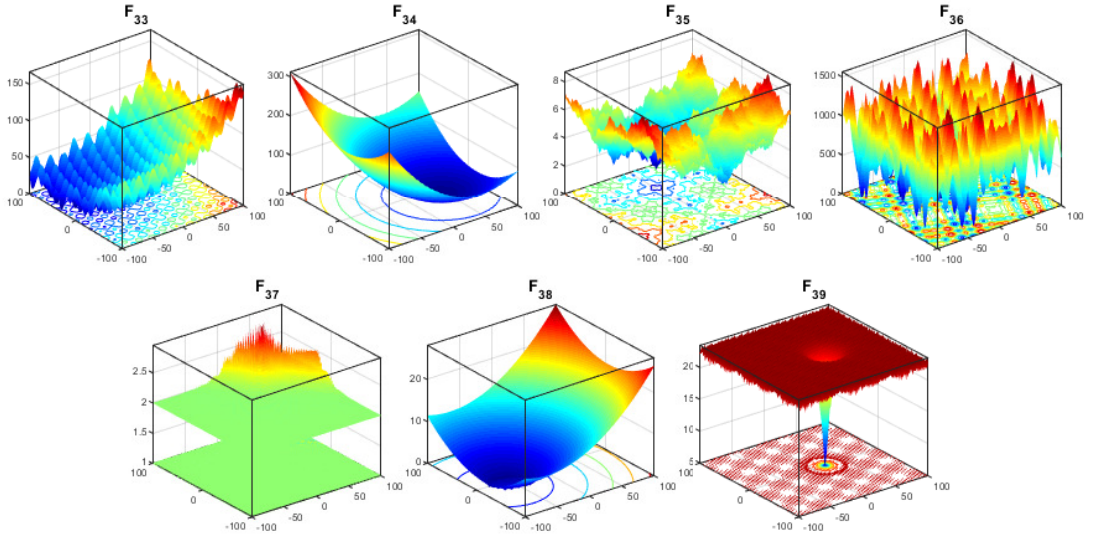
Fonksiyon	$n$	Aralık	$F_{min}$
$F_{30}$ (CEC01_2019) : Storn Chebyshev polinomsal uyum problemi	9	$[-8192, 8192]^n$	1
$F_{31}$ (CEC02_2019) : Ters Hilbert matris problemi	16	$[-16384, 16384]^n$	1
$F_{32}$ (CEC03_2019) : Lennard-Jones minimum enerji kümesi	18	$[-4, 4]^n$	1
$F_{33}$ (CEC04_2019) : Rastrigin fonksiyonu ( $F_9(x)$ )	10	$[-100, 100]^n$	1
$F_{34}$ (CEC05_2019) : Griewank fonksiyonu ( $F_{11}(x)$ )	10	$[-100, 100]^n$	1
$F_{35}$ (CEC06_2019) : Weierstrass fonksiyonu	10	$[-100, 100]^n$	1
$F_{36}$ (CEC07_2019) : Modifiye edilmiş Schwefel fonksiyonu	10	$[-100, 100]^n$	1
$F_{37}$ (CEC08_2019) : Genişletilmiş Schaffer F6 fonksiyonu	10	$[-100, 100]^n$	1
$F_{38}$ (CEC09_2019) : Happy Cat fonksiyonu	10	$[-100, 100]^n$	1
$F_{39}$ (CEC10_2019) : Ackley fonksiyonu ( $F_{10}(x)$ )	10	$[-100, 100]^n$	1

**Not:** Daha ayrıntılı bilgi için Price ve diğ. (2018) teknik raporu incelenebilir.

**Şekil 3.3:** İki boyutlu sabit boyutlu çok-modlu test fonksiyonlarının grafikleri.



Şekil 3.4: İki boyutlu bileşik test fonksiyonlarının grafikleri.



Şekil 3.5: İki boyutlu CEC 2019 test fonksiyonlarının ( $F_{33} - F_{39}$ ) grafikleri.

### 3.2 Karşılaştırma için Kullanılan Metasezgisel Algoritmalar

AISA algoritmasının performansını doğrulamak amacıyla, 13 iyi bilinen metasezgisel optimizasyon algoritması karşılaştırma için kullanılmıştır. Bu bölümde önerilen algoritmanın performansının kıyaslandığı algoritmaların kısa bir açıklaması sunulmuştur.

- *Parçacık Sürü Optimizasyonu (PSO)*: Kennedy ve Eberhart (1995) tarafından önerilen PSO algoritması, oldukça popüler bir popülasyon tabanlı stokastik optimizasyon algoritmasıdır. Bu algoritma balık ve kuş sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir. Her parçacık mevcut konumunu arama uzayının sınırları içerisinde üç faktöre bağlı olarak güncellemektedir: (i) parçacık tarafından o ana kadar elde edilen en iyi çözüm/konum (kendi kendini ayarlama), (ii) sürü tarafından elde edilen en iyi çözüm (sosyal ayarlama) ve (iii) önceki iterasyondaki parçacığın hızı (atalet ayarlama).

- *Yapay Arı Kolonisi (ABC)*: Bir diğer popülasyon tabanlı metasezgisel algoritma olan ABC algoritması, arı sürülerinin yiyecek arama davranışlarından esinlenilerek Karaboga ve Basturk (2007) tarafından önerilmiştir. ABC algoritması üç grup arıdan oluşmaktadır: işçi arılar, gözcü arılar ve kaşif arılar. Bu algoritma, keşif (işçi arılar ve gözcü arılar tarafından yürütülen) ve sömürü süreçlerini (kaşif arılar tarafından yürütülen) dengelemeye çalışan bir konseptte dayandığından, birçok gerçek dünya optimizasyon problemini çözmek için başarıyla uygulanmıştır.

- *Yerçekimi Arama Algoritması (GSA)*: Newton'un yerçekimi ve hareket yasasından yola çıkan Rashedi ve arkadaşlarının önerdiği GSA algoritması doğadan ilham almaktadır (Rashedi ve diğ. 2009). Bu algoritmada her kütle konum, ataletli kütle, pasif ve aktif yerçekimi kütlesi olmak üzere dört parametreye sahiptir. Bu algoritma, modifiye edilmiş ve hibrit versiyonları, global optimumu bulma yeteneğindeki başarısı nedeniyle çeşitli optimizasyon problemlerini çözmek için

literatürde kullanılmıştır (Rashedi ve diğ. 2018).

- *Yarasa Algoritması (BA)*: Popülasyon tabanlı metasezgisel algoritmalarından biri olan bu algoritma, yarasaların ekolokasyon (sesin yankılanması) davranışlarından esinlenilerek Yang tarafından geliştirilmiş ve performansı klasik iyi bilinen test fonksiyonları üzerinde kanıtlanmıştır (Yang 2010). Yarasalar avlanmak ve yer değiştirmek için sonar sinyalleri kullanır ve bir nesnenin veya avın kendisine göre konumunu belirleyebilirler. BA algoritması da yarasaların bu davranışına dayalı olarak tasarlanmış ve matematiksel olarak modellenmiştir.

- *Gri Kurt Optimizasyonu (GWO)*: Bu algoritma sürü zekasına dayalı tekniklerin en önemlilerinden bir tanesidir. Gri kurtların doğadaki davranışlarından esinlenilerek geliştirilen bu algoritma, gri kurtların liderlik hiyerarşisini ve avlanma mekanizmasını taklit edecek şekilde matematiksel olarak modellenip, iyi bilinen test fonksiyonlarına ve gerçek dünya kısıtlı optimizasyon problemlerine uygulanarak başarılı sonuçlar elde edilmiştir (Mirjalili ve diğ. 2014). Bu algoritmada liderlik hiyerarşisinin yanı sıra avlanma, arama, kuşatma ve saldırı mekanizmalarını simüle etmek için alfa, beta, delta ve omega olmak üzere dört tür gri kurt kullanılır.

- *Güve-Alev Optimizasyonu (Moth-Flame Optimization - MFO)*: Mirjalili (2015) tarafından yakın zamanda geliştirilmiş, doğadan ilham alan popülasyon temelli bir optimizasyon algoritmasıdır. Güvelerin enine yönelim (transverse orientation) olarak adlandırılan doğada yön bulma yöntemlerinden esinlenilerek modellenen bu algoritma, hem yerel hem de global arama yeteneklerine sahiptir. Zorlu optimizasyon problemleriyle başa çıkmada etkinliği kanıtlanan bu algoritmanın iki ana bileşeni, çözümleri temsil etmek için kullanılan güveler ve alevlerdir (Mirjalili 2015). Güveler, arama alanında en uygun çözümleri bulmak için gerçek arama ajanları olarak tanımlanırken, alevler o ana kadar elde edilen güvelerin en iyi pozisyonlarını temsil etmektedir.

- *Girdap Arama Algoritması (VS)*: Doğan ve Ölmez (2015) tarafından nümerik optimizasyon problemlerini çözmek için önerilen tek-çözüm tabanlı bir metasezgisel algoritma olan VS, karıştırılan sıvılarda oluşan girdap deseninden esinlenmiştir.

Uyarlanabilir bir adım boyutu ayarlama stratejisi kullanarak keşif ve sömürü arasında iyi bir dengeye sahip olan VS algoritması, arama sürecinde daha az hafızaya ihtiyaç duyması bakımından oldukça avantajlıdır.

- *Jaya Algoritması (JA)*: Rao (2016) tarafından, her arama adayının en iyi çözüme doğru hareket etme ve en kötü çözümden uzaklaşması yaklaşımına dayandırılarak geliştirilen popülasyon tabanlı bir metasezgisel algoritmadır. Pek çok metasezgisel algoritmanın aksine, JA, yalnızca popülasyon boyutu ve maksimum yineleme/iterasyon sayısı gibi ortak kontrol parametreleri dışında algoritmaya özgü herhangi bir parametre gerektirmez. Bu algoritmanın popüleritesinin ardındaki sebep, uygun algoritma parametrelerini ayarlama ve seçme zorluklarının üstesinden gelebilme kabiliyetidir.

- *Sinüs Kosinüs Algoritması (Sine Cosine Algorithm - SCA)*: 2016 yılında Mirjalili tarafından önerilen SCA, sinüs ve kosinüs fonksiyonlarına dayalı matematiksel bir model kullanılarak geliştirilmiş, gelecek vaat eden popülasyon tabanlı bir stokastik optimizasyon algoritmasıdır (Mirjalili (2016)). Bu algoritma arama ajanlarını rastgele oluşturur ve ardından hedef nokta (şimdiye kadar elde edilen en iyi çözüm/ajan) etrafındaki sinüs ve kosinüs dalgalarının salınımlarını kullanarak çözüm alanı sınırları içinde yeni çözümler oluşturur. Bu yeni çözümler dikkate alınarak en iyi çözüm güncellenir ve iteratif olarak evrilir. Algoritmada kullanılan rastgele ve uyarlanabilir parametreler, keşif ve sömürü için önemli bir role sahiptir.

- *Salp Sürüsü Algoritması (SSA)*: Mirjalili ve diğ. (2017) tarafından tek ve çok amaçlı optimizasyon problemlerini çözmek için geliştirilen, yüksek boyutlu ve farklı özelliklere sahip optimizasyon problemlerinin çözümünde yüksek performans sergileyen bu algoritma, derin okyanuslarda gezinen ve yiyecek arayan salpların sürü davranışlarını taklit eder. Salplar, yiyecek arama sırasında daha iyi hareket sağlamaya yardımcı olan salp zinciri adı verilen bir sürü oluşturur. Salp zincir mekanizması, popülasyon iki gruba ayrılarak matematiksel olarak modellenmiştir: (i) zincirin önünde olan ve sürüye rehberlik eden lider ve (ii) salpların geri kalanını oluşturan ve birbirini takip eden takipçiler.

- *Ağaç-tohum Algoritması (Tree-seed Algorithm - TSA)*: Sürekli optimizasyon problemlerini çözmek için Kiran (2015) tarafından ağaçlar ve onların tohumları arasındaki ilişkiden esinlenilerek geliştirilen, henüz yeni ancak gelecek vaat eden popülasyon tabanlı bir metasezgisel algoritmadır. Bu algoritma, arama eğilimi (ST) parametresi tarafından yönetilen iki mekanizma üzerine tasarlanmıştır. Bu mekanizmalar, TSA'nın keşif ve sömürü yeteneklerini dengelemeyi sağlar. Bu algoritmanın güçlü yanları etkili global arama yeteneğine, uygulama basitliğine ve hesaplama süresi verimliliğine sahip olmasıdır.

- *Diferansiyel Evrim (DE)*: 1997 yılında Storn ve Price tarafından önerilen bu algoritma, basit yapısı, kolay uygulanabilirliği ve zorlu optimizasyon problemleriyle başa çıkmadaki etkinliği nedeniyle evrimsel stokastik tekniklerde önemli bir yere sahiptir (Storn ve Price 1997). Başlangıçta, popülasyondaki vektörlerle temsil edilen bireyler rastgele oluşturulur ve daha sonra mutasyon, çaprazlama ve seçim olmak üzere üç ana evrimsel operatör kullanılarak yinelemeli olarak çözümler geliştirilir. Ayrıca, çok sayıda gerçek hayat problemini çözmek için farklı mutasyon stratejilerine sahip birçok DE varyantı olduğunu da belirtmek gerekir (Das ve diğ. 2016).

- *Evrimsel Strateji (ES)*: Doğal evrimden esinlenen bir arama paradigmasına sahip olan ES, popüler ve çok yönlü evrimsel algoritmalarından biridir (Rechenberg 1973; Schwefel 1993). ES algoritması sürekli veya ayrık optimizasyon problemlerini çözmek için mutasyon, rekombinasyon ve seçim operatörlerini kullanır. Bu algoritmanın farklı varyantları  $(\mu + \lambda)$ -ES,  $(\mu, \lambda)$ -ES gibi kısaltmalarla tanımlanmaktadır. Burada  $\mu$  ebeveyn,  $\lambda$  ise yavru/çocuk sayılarını temsil etmektedir.

### 3.3 Deneysel Plan ve Girdiler

Tüm benzetim çalışmaları, Intel Core I7 2.30 GHz işlemci, 8 GB RAM ve 64 bit Windows 10 işletim sistemine sahip bilgisayarda MATLAB (R2018a) platformunda gerçekleştirilmiştir.

Önerilen algoritmayı diğer algoritmalar ile adil olarak karşılaştırmak için

her algoritmanın maksimum fonksiyon deęerlendirme sayısı, optimize edilecek fonksiyonun boyutuna ( $n$ ) gre deęiřecek řekilde ( $n \times 10,000$ ) ayarlanmıřtır. Ayrıca PSO'nun kontrol parametreleri Hu (2006), DE ve ES'nin parametreleri Simon (2008), dięer algoritmaların parametre deęerleri ise orijinal alıřmalarında nerildięi veya mevcut kodlarında kullanıldıęı řekilde seilmiřtir. Btn algoritmaların parametreleri Tablo 3.6'da listelenmiřtir. Tm istatistiksel sonular algoritmaların 30 kez baęımsız kořumu ile elde edilmiřtir.

**Tablo 3.6:** Algoritmaların kontrol parametreleri.

Algoritma	Parametreler	Değerler
AISA	Popülasyon büyüklüğü	30
	Chebyshev polinomlarının sayısı ( $k$ )	3
PSO	Popülasyon büyüklüğü	30
	Bilişsel ve sosyal sabitler	2, 2
	Atalet ağırlığı	[0.9 0.4]
ABC	Popülasyon büyüklüğü (koloni boyutu)	40
	Kaynak sayısı	Koloni boyutu/2
	Limit	100
GSA	Popülasyon büyüklüğü	50
	Rnorm, Rpower, alpha ve $G_0$	2, 1, 20, 100
BA	Popülasyon büyüklüğü	20
	Ses şiddeti ( $A$ ), sinyal yayılım oran ( $r$ )	0.5, 0.5
	Minimum ve maksimum frekanslar	0, 2
GWO	Popülasyon büyüklüğü	30
	Kontrol parametresi ( $a$ )	[2 0]
MFO	Popülasyon büyüklüğü	30
	Yakınsama sabiti ( $r$ )	[-1 -2]
VS	Spiral faktör ( $b$ )	1
	Komşu çözüm sayısı	50
JA	$gammaincinv$ fonksiyonu parametresi ( $x$ )	0.1
SCA	Popülasyon büyüklüğü	25
	Parametre $a$	30
SSA	Parametre $c_1$	2
	Popülasyon büyüklüğü	30
TSA	Popülasyon büyüklüğü (Ağaç sayısı)	20
	Arama eğilimi ( $ST$ )	0.1
	Tohum sayısı	Ağaç sayısının %10-25
DE	Popülasyon büyüklüğü	50
	Şema	DE/rand/1
	Ölçeklendirme faktörü	0.5
	Çaprazlama olasılığı	0.5
ES	Popülasyon büyüklüğü	50
	Strateji	$(\mu + \lambda)$
	$\lambda$	10
	$\sigma$	1



### 3.4 Test Fonksiyonları Üzerinde Performans Değerlendirmesi

Önerilen algoritmanın etkinliğini kanıtlamak amacıyla, algoritma tek-modlu, çok-modlu, sabit boyutlu çok-modlu, bileşik ve CEC 2019 kıyaslama fonksiyonları üzerinde 13 iyi bilinen optimizasyon algoritması ile karşılaştırılmıştır. Ayrıca problem boyutunun AISA'nın performansı üzerindeki etkisini incelemek amacıyla tek-modlu ve çok-modlu fonksiyonların 10, 30 ve 50 boyutlu versiyonları üzerinde bir ölçeklenebilirlik çalışması yapılmıştır. Algoritmaların nicel karşılaştırması için 30 bağımsız koşumda elde edilen çözümlerin en iyi (En İyi), ortalama (Ort.) ve standart sapma (Std.) değerleri gibi farklı performans göstergeleri kullanılmıştır.

#### 3.4.1 10, 30 ve 50 boyutlu tek-modlu test fonksiyonlarında ( $F_1 - F_7$ ) performans değerlendirilmesi

Tek-modlu test fonksiyonları ( $F_1 - F_7$ ), metasezgisel algoritmaların sömürü yeteneklerini test etmek için kullanılmaktadırlar. Tüm algoritmaların 10, 30 ve 50 boyutlu tek-modlu test fonksiyonları üzerinde elde ettikleri performans sonuçları sırasıyla Tablolar 3.7 - 3.9'da sunulmuştur. Bu tablolar incelendiğinde AISA'nın farklı boyutlardaki  $F_1$ ,  $F_2$ ,  $F_3$ ,  $F_4$ ,  $F_6$  ve  $F_7$  fonksiyonlarında en iyi sonuçları elde edebildiği gözlenmektedir. Ek olarak, AISA'nın 10, 30 ve 50 boyutlu  $F_2$ ,  $F_3$ ,  $F_4$  ve  $F_7$  fonksiyonlarında diğer rakiplerinden önemli ölçüde daha iyi olduğu sonucu çıkarılabilir. Sonuçlar, önerilen algoritmanın diğer algoritmalara kıyasla daha başarılı ve sömürü açısından üstün yeteneklere sahip olduğunu ortaya koymaktadır.

#### 3.4.2 10, 30 ve 50 boyutlu çok-modlu test fonksiyonlarında ( $F_8 - F_{13}$ ) performans değerlendirilmesi

Tek-modlu fonksiyonların aksine çok-modlu fonksiyonlar çok sayıda yerel minimuma sahiptirler. Bu fonksiyonların boyutları arttıkça üstel olarak yerel minimum

sayıları da artmaktadır. Bu karakteristik özelliklerinden dolayı bu fonksiyonlar algoritmaların keşif yeteneğini test etmek için oldukça elverişlidirler. Algoritmaların 10, 30 ve 50 boyutlu çok-modlu test fonksiyonlarına ilişkin sonuçları sırasıyla Tablo 3.10, Tablo 3.11 ve Tablo 3.12’de verilmiştir. Bu tablolardaki sonuçlar, AISA’nın 10, 30 ve 50 boyutlu altı fonksiyonun beşini ( $F_9 - F_{13}$ ) çözmeye başarısında genel olarak ilk sırada olduğunu ve diğer güçlü metasezgisel algoritmalara kıyasla daha iyi sonuçlar verdiğini göstermektedir. Elde edilen sonuçlar göstermektedir ki, AISA yerel optimumdan kurtulabilme ve problemin boyutu ne olursa olsun yüksek performanslı bir keşif kabiliyetine sahiptir.

### **3.4.3 Sabit boyutlu çok-modlu test fonksiyonlarında ( $F_{14} - F_{23}$ ) performans değerlendirmesi**

Algoritmaların bu test problemleri üzerinde elde ettikleri istatistiksel sonuçlar Tablo 3.13’te listelenmiş olup, AISA’nın bu on fonksiyonun yedisinde ( $F_{14}, F_{15}, F_{17}, F_{19}, F_{21}, F_{22}$ ) en iyi performansı gösterdiği ve diğer fonksiyonlarda da rekabetçi sonuçlar elde ettiği görülmektedir. DE’nin, büyük boyutlu çok-modlu fonksiyonlarda sergilediği performansın aksine, bu test problemlerinde çok daha iyi performans gösterdiği ve başarı değerlendirmesinde ilk genel sırayı aldığını belirtmek gerekir. Genel olarak, AISA’nın yüksek keşif kabiliyetine sahip olduğu sonucuna varılabilir.

### **3.4.4 Bileşik test fonksiyonlarında ( $F_{24} - F_{29}$ ) performans değerlendirmesi**

Standart test fonksiyonlarından üretilmiş olan bileşik test fonksiyonları bir algoritmanın hem keşif hem de sömürü davranışını eş zamanlı olarak test etmek için kullanılırlar. Keşif ve sömürü arasında kurulan iyi bir denge yerel optimumdan kaçınmayı mümkün kıldığından, bu fonksiyonların optimizasyonları pratikte son derece zordur. Tablo 3.14 tüm algoritmaların bileşik test fonksiyonları üzerindeki optimizasyon sonuçlarını göstermektedir. Bu tabloya bakarak AISA’nın dört fonksiyonda ( $F_{24}, F_{25}, F_{27}$  ve  $F_{28}$ ) en başarılı algoritma olduğu görülmektedir. Bu

bulgular ışığında, AISA'nın optimizasyon sürecinin iki temel yapı taşı olan keşif ve sömürü davranışlarını başarılı bir şekilde dengelendiği söylenebilir.

### 3.4.5 CEC 2019 test fonksiyonlarında ( $F_{30} - F_{39}$ ) performans değerlendirmesi

CEC 2019 test fonksiyonları modern ve zorlu optimizasyon problemleridir. Algoritmaların bu problem üzerindeki nicel karşılaştırma sonuçları Tablo 3.15'te sunulmuştur. AISA algoritmasının beş zorlu test fonksiyonunda ( $F_{30}, F_{31}, F_{32}, F_{37}$  ve  $F_{39}$ ) en iyi performansı sergilediği görülmektedir. Sonuçlar, AISA'nın karmaşık CEC 2019 kıyaslama fonksiyonlarını çömede diğer rakip metasezgisel algoritmalara göre üstün performans ve güçlü bir optimizasyon yeteneğine sahip olduğunu göstermektedir.

### 3.4.6 Wilcoxon İşaretli Sıra Testi

Bundan önce yapılmış olan performans analizinde çözümlerin en iyisi, ortalaması ve standart sapması gibi temel istatistiksel sonuçlar sunulmuş olmasına rağmen, algoritmaların performansını daha titiz ve ayrıntılı olarak analiz etmek amacıyla ikili kıyaslamalar için Wilcoxon işaretli sıra testi kullanılmıştır. Bu test iki örneğin istatistiksel olarak farklı popülasyonlara/gruplara ait olup olmadığını belirlemek için kullanılan parametrik olmayan istatistiksel hipotez testidir (Derrac ve diğ. 2011). Bir hipotezin reddedilip edilmeyeceği bir anlam düzeyinde ( $\alpha$ ) gerçekleştirilir. Bu testte  $H_0 : \mu_1 = \mu_2$  (null hipotezi) "Aynı problem iki farklı algoritma tarafından çözdürüldükten sonra elde edilen çözümlerin medyanları arasında bir fark yoktur yani algoritmalar istatistiksel olarak farklı değildir",  $H_1 : \mu_1 \neq \mu_2$  (alternatif hipotez) "Algoritmaların elde ettikleri çözümlerin medyanları farklı olup, algoritmalar istatistiksel olarak farklıdır".

AISA'nın diğer algoritmalara karşı Wilcoxon işaretli sıra testine göre parametrik olmayan istatistiksel sonuçları Tablolar 3.16 - 3.24'te verilmiştir. Bu sonuçlar elde edilirken her algoritma 30 koşuma tabi tutulmuş ve testlerde anlam

düzeyi  $\alpha = 0.05$  olarak kullanılmıştır. Tablolarda  $p$ -değeri, T+ ve T- değerleri raporlanmıştır.  $p$ -değeri kurulan hipotezin doğru olması durumunda  $H_0$  hipotezinin reddedilme olasılığını, T+ ilk algoritmanın, T- ise ikinci algoritmanın daha iyi olduğu sonuçlardaki sıralama değerlerinin toplamlarını ifade etmektedir. Bunlara ek olarak '+' hem  $H_0$  hipotezinin reddedildiğini hem de AISA'nın istatistiksel olarak diğer algoritmaya göre daha üstün performans gösterdiğini, '-'  $H_0$  hipotezinin reddedildiğini ve kıyaslanan algoritmanın AISA'ya göre daha başarılı performans gösterdiğini, '=' ise  $H_0$  hipotezinin reddedilmediğini yani iki algoritma arasında istatistiksel bir fark olmadığını belirtmektedir. KD ise AISA'nın kazanıp kazanmama durumunu göstermektedir. Sonuçları daha kolay gözlemleyebilmek için Tablolar 3.16 - 3.24'te yer alan veriler derlenerek Tablo 3.25'te özetlenmiştir. Bu tablodaki her hücre, her test kategorisinde AISA'nın diğer algoritma ile ikili kıyaslanmasıyla elde edilen üç duruma ait (+/=/-) toplam sonuçları göstermektedir. Sonuçlar incelendiğinde, önerilen AISA algoritmasının diğerlerinden daha iyi performans gösterdiği sonucuna ulaşılabilir. Bununla birlikte, sabit boyutlu çok-modlu test yataklarında AISA, TSA ve DE algoritmalarıyla rekabet içindedir. Wilcoxon işaretli sıra testinde elde edilen sonuçlara dayanarak AISA'nın istatistiksel olarak diğer algoritmalarından farklı ve üstün performans sergilediği söylenebilir.

### 3.4.7 Yakınsama Analizi

Bu bölümde, önerilen algoritmanın performansını daha da derinlemesine incelemek için algoritmaların yakınsama davranışları tüm kıyaslama fonksiyonları üzerinde incelenmiştir. Şekiller 3.6 - 3.10, AISA ve diğer algoritmaların test fonksiyonlarını optimize ederken oluşturduğu yakınsama eğrilerini göstermektedir. Bu şekillerde "En iyilerin ortalaması", 30 bağımsız koşum üzerinde her bir fonksiyon değerlendirmesinde o ana kadar elde edilen en iyi çözümlerin ortalaması olduğuna dikkat edilmelidir.

Şekil 3.6 algoritmaların 10 boyutlu tek-modlu ve çok-modlu fonksiyonlar

üzerindeki yakınsama performansını göstermektedir. AISA algoritmasının,  $F_1$  fonksiyonu dışında diğer algoritmalarından çok daha iyi bir yakınsama performansı gösterdiği görülmektedir. Şekil 3.7 ve Şekil 3.8 sırasıyla 30 ve 50 boyutlu tek-modlu ve çok-modlu fonksiyonlardaki yakınsama eğrileri göstermektedir. AISA'nın  $F_5$  fonksiyonu dışında daha az sayıda fonksiyon değerlendirmesi ile hızlı bir şekilde optimum çözüme yakınsadığı görülmektedir. Bu sonuçlar, AISA'nın farklı boyutlara sahip hem tek-modlu hem de çok-modlu fonksiyonlarda tutarlı yakınsama başarısı elde ettiğini göstermektedir. Ek olarak Şekiller 3.9 ve 3.10'da fonksiyonlar üzerinde tüm algoritmaların yakınsama geçmişleri sunulmuştur. AISA'nın fonksiyonların yarısından fazlasında dikkate değer bir yakınsama oranı sergilediği ve sürekli hal durumuna ulaştığı açıkça görülmektedir. Genel yakınsama analizi çerçevesinde sonuçlar, AISA'nın yakınsama oranı açısından diğer rakip algoritmalara kıyasla çok daha başarılı olduğunu ortaya koymaktadır.

### **3.4.8 Zaman Verimliliği Analizi**

Algoritmaların performansını değerlendirmek için bir başka değerlendirme kriteri olarak, algoritmaların çalışma süreleri araştırılmıştır. Şekil 3.11'de her bir algoritma için toplam çalışma süreleri karşılaştırılmıştır. Bu çalışma süreleri; 10, 30 ve 50 boyutlu tek-modlu ve çok-modlu, sabit boyutlu çok-modlu, bileşik ve CEC 2019 test problemlerinin 30 bağımsız koşumuyla elde edilen toplam sürelerini belirtmektedir. Tüm problemlerde AISA'nın toplam çalışma süresinin 12545 saniye olduğu görülmektedir. AISA'nın GSA, DE ve ES gibi önemli algoritmalarından daha az zaman kullandığı görülmektedir. AISA'nın çalışma süresi maliyetinin, LSE tarafından parametreleri kestirilen CFLN'nin gerektirdiği ekstra hesaplamalardan kaynaklandığı unutulmamalıdır. Genel olarak, tüm deneysel analizler düşünüldüğünde AISA'nın tatmin edici bir performansa sahip olduğu açıkça ifade edilebilir.

**Tablo 3.7:** 10 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_1$	En iyi	<b>0.00E+00</b>	7.75E-89	2.79E-17	1.34E-18	1.22E+02	<b>0.00E+00</b>	1.34E-114	3.71E-33	1.02E-110	3.68E-114	1.58E-10	1.43E-110	1.21E-100	3.92E-02
	Ort.	<b>0.00E+00</b>	1.32E-81	5.09E-17	3.04E-18	3.45E+03	<b>0.00E+00</b>	6.40E-108	8.13E-28	3.73E-106	5.81E-97	7.24E-10	5.30E-107	1.39E-97	1.84E-01
	Std.	0.00E+00	7.04E-81	1.36E-17	9.38E-19	1.75E+03	0.00E+00	2.39E-107	3.24E-27	7.73E-106	2.17E-96	2.68E-10	2.37E-106	5.13E-97	1.16E-01
$F_2$	En iyi	<b>8.86E-270</b>	8.71E-50	1.56E-16	3.66E-09	1.73E-03	7.51E-232	3.29E-68	6.96E-16	1.86E-57	6.37E-73	3.24E-06	1.09E-69	4.62E-56	3.13E-02
	Ort.	<b>2.46E-256</b>	5.77E-46	2.22E-16	5.19E-09	1.61E+01	7.45E-228	1.08E-63	6.50E-13	5.14E-56	4.46E-63	8.48E-06	7.68E-68	2.91E-55	8.17E-02
	Std.	0.00E+00	1.64E-45	4.85E-17	7.96E-10	1.58E+01	0.00E+00	3.08E-63	1.87E-12	8.81E-56	2.00E-62	5.45E-06	1.27E-67	2.86E-55	2.53E-02
$F_3$	En iyi	<b>0.00E+00</b>	1.27E-31	7.46E-01	1.70E-18	5.65E+02	5.26E-206	3.97E-36	2.56E-19	8.37E-21	4.22E-59	3.85E-10	1.13E-18	2.24E-15	3.09E+01
	Ort.	<b>5.72E-285</b>	6.81E-27	4.79E+00	5.60E-18	5.31E+03	2.18E-186	3.89E+02	1.76E-15	2.95E-18	1.61E-39	1.34E-09	4.45E-16	3.22E-14	3.93E+02
	Std.	0.00E+00	1.41E-26	3.13E+00	2.02E-18	3.19E+03	0.00E+00	1.50E+03	4.47E-15	4.78E-18	8.79E-39	6.24E-10	1.03E-15	4.94E-14	3.42E+02
$F_4$	En iyi	<b>2.19E-226</b>	2.79E-24	4.39E-02	6.73E-10	1.55E+01	7.61E-134	9.20E-14	2.83E-14	4.06E-31	5.05E-37	1.10E-05	3.00E-21	2.40E-26	3.98E-01
	Ort.	<b>1.22E-215</b>	8.13E-22	1.51E-01	1.11E-09	3.36E+01	1.20E-127	1.65E-01	1.48E-11	9.39E-30	5.42E-29	1.55E-05	1.11E-19	2.16E-25	9.27E-01
	Std.	0.00E+00	1.60E-21	6.51E-02	2.00E-10	7.35E+00	4.23E-127	7.84E-01	4.92E-11	1.46E-29	2.35E-28	2.82E-06	9.65E-20	1.74E-25	3.44E-01
$F_5$	En iyi	<b>0.00E+00</b>	4.91E-03	1.10E-03	2.90E+00	4.12E+00	5.24E+00	1.69E-01	3.91E-01	1.01E-08	5.97E+00	7.03E-01	2.85E-02	3.62E-03	3.49E+00
	Ort.	<b>0.00E+00</b>	2.15E+00	1.88E-01	3.08E+00	5.03E+04	6.33E+00	4.35E+01	1.71E+01	1.05E-01	6.61E+00	8.76E+01	2.17E+00	1.96E+00	5.58E+01
	Std.	0.00E+00	1.12E+00	2.25E-01	1.28E-01	1.17E+05	7.01E-01	1.11E+02	4.49E+01	4.14E-01	4.18E-01	1.95E+02	1.42E+00	7.00E-01	4.05E+01
$F_6$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	3.50E-17	1.40E-18	7.98E+02	5.79E-06	<b>0.00E+00</b>	1.23E-32	8.16E-02	4.84E-02	1.47E-10	<b>0.00E+00</b>	<b>0.00E+00</b>	2.33E-02
	Ort.	<b>0.00E+00</b>	<b>0.00E+00</b>	6.19E-17	2.98E-18	3.47E+03	1.24E-05	2.60E-32	2.04E-28	1.24E-01	2.14E-01	6.40E-10	<b>0.00E+00</b>	<b>0.00E+00</b>	2.05E-01
	Std.	0.00E+00	0.00E+00	2.09E-17	7.68E-19	1.43E+03	2.69E-06	5.77E-32	4.27E-28	2.07E-02	1.08E-01	3.03E-10	0.00E+00	0.00E+00	1.51E-01
$F_7$	En iyi	1.14E-05	3.64E-04	3.17E-03	7.64E-04	2.49E-01	<b>2.03E-06</b>	1.80E-04	1.07E-04	2.07E-04	2.44E-05	2.08E-04	3.32E-04	2.60E-03	1.48E-03
	Ort.	<b>7.60E-05</b>	1.49E-03	1.24E-02	3.31E-03	6.90E-01	8.07E-05	3.49E-03	1.36E-03	1.14E-03	3.28E-04	1.44E-03	1.13E-03	8.40E-03	4.02E-03
	Std.	5.36E-05	9.63E-04	4.06E-03	1.71E-03	4.80E-01	8.50E-05	2.57E-03	7.69E-04	4.77E-04	3.39E-04	9.23E-04	5.00E-04	3.18E-03	2.09E-03

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.8:** 30 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_1$	En iyi	<b>0.00E+00</b>	9.00E-67	4.20E-16	4.98E-18	1.75E+03	<b>0.00E+00</b>	2.81E-88	6.01E-30	1.89E-109	1.03E-73	2.55E-09	1.56E-57	2.30E-89	2.57E-01
	Ort.	<b>0.00E+00</b>	1.23E-61	5.65E-16	8.49E-18	9.63E+03	<b>0.00E+00</b>	2.00E+03	5.67E-27	5.40E-106	6.36E-57	3.95E-09	7.78E-56	1.66E-87	4.66E-01
	Std.	0.00E+00	5.89E-61	8.41E-17	1.96E-18	5.17E+03	0.00E+00	4.84E+03	1.69E-26	1.35E-105	1.64E-56	7.24E-10	1.83E-55	2.37E-87	1.85E-01
$F_2$	En iyi	<b>0.00E+00</b>	9.44E-42	1.12E-15	1.21E-08	1.71E+00	<b>0.00E+00</b>	3.71E-55	1.20E-12	7.93E-61	1.67E-68	5.35E-05	1.08E-42	8.89E-51	1.74E-01
	Ort.	<b>0.00E+00</b>	2.99E-38	1.48E-15	1.67E-08	3.19E+03	<b>0.00E+00</b>	4.20E+01	2.86E-07	1.48E-59	4.18E-58	4.31E-01	8.70E-42	4.48E-50	2.42E-01
	Std.	0.00E+00	6.32E-38	1.94E-16	2.42E-09	1.29E+04	0.00E+00	2.44E+01	1.17E-06	2.12E-59	2.21E-57	7.09E-01	9.00E-42	3.51E-50	4.58E-02
$F_3$	En iyi	<b>0.00E+00</b>	5.28E-05	8.68E+02	3.82E-17	7.88E+03	9.49E-211	9.97E-10	4.53E-09	8.80E+02	7.66E-08	3.55E-08	8.90E+02	1.81E+03	5.10E+03
	Ort.	<b>0.00E+00</b>	7.22E-04	2.77E+03	1.50E-02	2.84E+04	1.26E-181	1.83E+04	1.24E-07	7.78E+03	1.23E+01	6.74E-08	3.20E+03	4.39E+03	8.88E+03
	Std.	0.00E+00	8.31E-04	8.39E+02	2.63E-02	2.13E+04	0.00E+00	1.18E+04	1.85E-07	6.05E+03	3.59E+01	2.17E-08	1.80E+03	1.78E+03	2.40E+03
$F_4$	En iyi	<b>0.00E+00</b>	7.52E-03	2.50E+01	1.65E-09	3.09E+01	7.51E-158	4.10E+01	5.45E-08	1.23E-07	4.48E-08	3.08E-04	5.45E-01	1.55E-10	2.81E+00
	Ort.	<b>0.00E+00</b>	2.29E-02	3.51E+01	2.01E-09	4.50E+01	8.47E-153	6.26E+01	4.54E-06	9.13E-06	2.74E-02	1.81E-01	1.36E+00	1.37E-07	4.91E+00
	Std.	0.00E+00	1.51E-02	5.27E+00	2.32E-10	8.72E+00	3.73E-152	1.11E+01	7.73E-06	1.01E-05	5.81E-02	3.05E-01	6.16E-01	7.50E-07	9.82E-01
$F_5$	En iyi	<b>0.00E+00</b>	5.73E-02	5.72E-03	1.86E+01	1.28E+01	2.51E+01	5.69E-02	1.95E+01	2.66E-14	2.63E+01	2.13E+01	1.92E+01	1.58E+01	3.24E+01
	Ort.	2.55E+00	3.41E+01	1.18E-01	2.35E+01	3.84E+03	2.63E+01	2.68E+06	8.94E+01	<b>2.54E-05</b>	2.75E+01	8.43E+01	2.42E+01	1.77E+01	1.81E+02
	Std.	2.10E+00	2.92E+01	1.32E-01	2.36E+01	1.21E+04	6.74E-01	1.46E+07	1.34E+02	9.74E-05	7.19E-01	1.34E+02	1.06E+01	6.25E-01	2.83E+02
$F_6$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	3.80E-16	7.62E-18	2.25E+03	1.07E-05	3.50E-30	1.57E-29	1.71E+00	2.98E+00	2.86E-09	<b>0.00E+00</b>	<b>0.00E+00</b>	2.25E-01
	Ort.	<b>0.00E+00</b>	2.34E-32	5.65E-16	1.20E-17	9.42E+03	3.88E-01	2.67E+03	2.84E-27	2.54E+00	3.57E+00	4.76E-09	2.05E-34	<b>0.00E+00</b>	4.80E-01
	Std.	0.00E+00	3.12E-32	8.47E-17	3.06E-18	4.88E+03	2.94E-01	5.23E+03	4.84E-27	5.24E-01	2.73E-01	1.14E-09	7.82E-34	0.00E+00	2.00E-01
$F_7$	En iyi	<b>5.38E-06</b>	6.68E-03	8.85E-02	7.34E-03	4.70E-01	1.26E-05	1.15E-02	2.07E-03	2.62E-03	2.42E-04	2.75E-03	3.36E-03	5.15E-02	7.47E-03
	Ort.	<b>4.34E-05</b>	1.12E-02	1.58E-01	1.16E-02	1.01E+00	4.45E-05	5.28E+00	5.74E-03	6.81E-03	2.94E-03	7.55E-03	9.47E-03	7.94E-02	1.58E-02
	Std.	2.72E-05	3.58E-03	3.82E-02	2.95E-03	3.64E-01	2.28E-05	1.06E+01	2.22E-03	3.38E-03	4.44E-03	2.46E-03	2.82E-03	1.47E-02	5.54E-03

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.9:** 50 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_1$	En iyi	<b>0.00E+00</b>	4.22E-52	8.47E-16	1.39E-17	3.37E+03	<b>0.00E+00</b>	1.22E-69	3.37E-28	4.55E-97	2.65E-50	8.79E-09	2.07E-25	6.28E-74	1.29E-01
	Ort.	<b>0.00E+00</b>	2.68E-48	1.33E-15	2.31E-17	1.30E+04	<b>0.00E+00</b>	9.67E+03	1.32E-26	9.55E-93	2.60E-33	1.10E-08	1.66E-24	1.50E-72	8.07E-01
	Std.	0.00E+00	9.79E-48	2.51E-16	4.28E-18	6.74E+03	0.00E+00	1.10E+04	2.05E-26	1.79E-92	1.36E-32	1.30E-09	2.03E-24	2.00E-72	2.72E-01
$F_2$	En iyi	<b>0.00E+00</b>	1.25E-33	2.58E-15	2.27E-08	1.54E+00	<b>0.00E+00</b>	1.00E+01	1.52E-09	6.77E-55	6.75E-66	2.22E-02	9.65E-23	9.82E-44	3.18E-01
	Ort.	<b>0.00E+00</b>	1.46E-28	3.20E-15	2.88E-08	2.87E+09	<b>0.00E+00</b>	6.73E+01	5.37E-06	2.10E-47	9.85E-55	1.12E+00	4.02E-22	5.40E-42	4.12E-01
	Std.	0.00E+00	7.46E-28	3.55E-16	3.49E-09	1.56E+10	0.00E+00	3.42E+01	2.53E-05	1.15E-46	4.25E-54	1.14E+00	2.46E-22	3.11E-42	6.46E-02
$F_3$	En iyi	<b>0.00E+00</b>	3.31E-01	1.24E+04	8.92E-01	1.08E+04	2.59E-212	5.00E+03	9.78E-05	2.21E+04	8.42E+00	6.16E-07	2.06E+04	4.46E+04	1.73E+04
	Ort.	<b>0.00E+00</b>	1.16E+00	1.64E+04	5.37E+00	5.57E+04	5.60E-179	3.71E+04	4.79E-04	5.68E+04	3.83E+03	1.06E-06	5.62E+04	6.70E+04	2.58E+04
	Std.	0.00E+00	5.65E-01	2.30E+03	3.82E+00	4.19E+04	0.00E+00	2.44E+04	3.40E-04	1.45E+04	4.79E+03	2.23E-07	1.07E+04	7.54E+03	4.19E+03
$F_4$	En iyi	<b>0.00E+00</b>	5.82E-01	5.91E+01	2.22E-09	3.19E+01	9.36E-178	7.25E+01	3.09E-04	1.26E+01	1.21E-01	1.00E+00	3.49E+01	2.33E-04	6.10E+00
	Ort.	<b>0.00E+00</b>	8.65E-01	6.64E+01	2.97E-09	4.52E+01	6.98E-171	8.35E+01	5.94E-03	1.97E+01	9.93E+00	8.40E+00	5.44E+01	4.90E-03	8.96E+00
	Std.	0.00E+00	1.84E-01	3.52E+00	3.74E-10	7.05E+00	0.00E+00	4.32E+00	7.12E-03	2.95E+00	7.44E+00	3.85E+00	8.92E+00	2.07E-02	2.06E+00
$F_5$	En iyi	<b>0.00E+00</b>	6.32E+00	4.61E-03	3.54E+01	2.85E+01	4.50E+01	1.46E-01	4.28E+01	2.33E-13	4.70E+01	3.82E+01	4.29E+01	3.04E+01	7.37E+01
	Ort.	3.89E+01	6.82E+01	2.02E-01	3.61E+01	5.15E+01	4.67E+01	5.37E+06	7.90E+01	<b>6.37E-09</b>	4.80E+01	7.24E+01	6.79E+01	3.56E+01	2.06E+02
	Std.	1.33E+01	4.15E+01	3.46E-01	3.21E-01	4.23E+01	1.09E+00	2.03E+07	9.77E+01	1.62E-08	5.82E-01	4.25E+01	3.61E+01	9.41E+00	5.09E+01
$F_6$	En iyi	<b>0.00E+00</b>	7.40E-32	1.11E-15	1.53E-17	1.61E+03	1.25E+00	2.37E-26	3.40E-28	8.15E+00	7.42E+00	8.08E-09	2.03E-25	<b>0.00E+00</b>	2.77E-01
	Ort.	<b>0.00E+00</b>	2.13E-30	1.38E-15	2.13E-17	1.33E+04	2.13E+00	8.32E+03	5.14E-27	1.45E+01	8.04E+00	1.11E-08	2.10E-24	<b>0.00E+00</b>	8.50E-01
	Std.	0.00E+00	3.63E-30	1.80E-16	5.72E-18	8.14E+03	6.17E-01	9.85E+03	6.15E-27	6.09E+00	3.32E-01	1.68E-09	2.08E-24	0.00E+00	2.48E-01
$F_7$	En iyi	<b>4.78E-06</b>	1.04E-02	3.13E-01	1.41E-02	4.19E-01	1.20E-05	3.39E-01	5.22E-03	1.16E-02	1.46E-03	4.49E-03	2.44E-02	1.53E-01	1.47E-02
	Ort.	<b>2.27E-05</b>	2.91E-02	4.72E-01	2.27E-02	6.22E-01	6.35E-05	1.55E+01	1.09E-02	2.51E-02	1.07E-02	1.61E-02	4.30E-02	1.91E-01	2.91E-02
	Std.	1.32E-05	1.10E-02	6.13E-02	4.70E-03	1.27E-01	2.93E-05	2.09E+01	3.65E-03	7.97E-03	8.09E-03	4.38E-03	1.09E-02	2.17E-02	6.46E-03

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.



**Tablo 3.10:** 10 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	En iyi	<b>-4.19E+03</b>	-3.16E+03	<b>-4.19E+03</b>	-2.07E+03	-3.12E+03	-3.50E+03	-3.72E+03	<b>-4.19E+03</b>	<b>-4.19E+03</b>	-2.83E+03	-3.32E+03	<b>-4.19E+03</b>	<b>-4.19E+03</b>	-4.19E+03
	Ort.	<b>-4.19E+03</b>	-2.54E+03	<b>-4.19E+03</b>	-1.53E+03	-1.89E+03	-2.82E+03	-3.10E+03	-3.92E+03	-3.75E+03	-2.42E+03	-2.92E+03	-4.12E+03	-4.18E+03	-4.19E+03
	Std.	2.78E-12	3.01E+02	2.53E-12	2.32E+02	3.69E+02	3.37E+02	2.98E+02	1.96E+02	4.85E+02	1.74E+02	1.93E+02	1.01E+02	3.54E+01	4.33E-01
$F_9$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	9.95E-01	1.19E+01	<b>0.00E+00</b>	6.96E+00	1.99E+00	9.05E+00	<b>0.00E+00</b>	6.96E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	1.41E-02
	Ort.	<b>0.00E+00</b>	2.02E+00	<b>0.00E+00</b>	3.08E+00	4.41E+01	<b>0.00E+00</b>	2.73E+01	1.59E+01	1.95E+01	<b>0.00E+00</b>	1.64E+01	1.64E+00	<b>0.00E+00</b>	9.77E-02
	Std.	0.00E+00	1.27E+00	0.00E+00	1.58E+00	1.67E+01	0.00E+00	1.57E+01	7.88E+00	6.21E+00	0.00E+00	4.76E+00	1.55E+00	0.00E+00	6.30E-02
$F_{10}$	En iyi	<b>8.88E-16</b>	4.44E-15	4.44E-15	1.66E-09	1.29E+01	4.44E-15	<b>8.88E-16</b>	4.44E-15	4.44E-15	<b>8.88E-16</b>	7.88E-06	4.44E-15	<b>8.88E-16</b>	9.17E-02
	Ort.	<b>8.88E-16</b>	4.56E-15	7.99E-15	2.54E-09	1.57E+01	4.44E-15	2.58E-01	1.66E-14	4.44E-15	3.38E-15	6.56E-01	4.44E-15	4.09E-15	2.61E-01
	Std.	0.00E+00	6.49E-16	9.33E-16	4.19E-10	1.69E+00	0.00E+00	8.10E-01	1.69E-14	0.00E+00	1.66E-15	7.91E-01	0.00E+00	1.08E-15	9.87E-02
$F_{11}$	En iyi	<b>0.00E+00</b>	2.95E-02	<b>0.00E+00</b>	<b>0.00E+00</b>	1.47E+01	<b>0.00E+00</b>	4.67E-02	1.01E-01	1.51E-01	<b>0.00E+00</b>	9.10E-02	<b>0.00E+00</b>	<b>0.00E+00</b>	1.15E-01
	Ort.	<b>0.00E+00</b>	1.19E-01	2.10E-03	4.11E-03	5.42E+01	3.06E-03	1.40E-01	2.78E-01	3.65E-01	7.01E-04	3.13E-01	6.07E-02	<b>0.00E+00</b>	3.28E-01
	Std.	0.00E+00	1.05E-01	4.72E-03	5.65E-03	2.26E+01	6.07E-03	6.91E-02	1.23E-01	1.11E-01	3.72E-03	1.59E-01	5.33E-02	0.00E+00	1.21E-01
$F_{12}$	En iyi	<b>4.71E-32</b>	<b>4.71E-32</b>	3.48E-17	2.09E-20	5.63E+00	5.97E-07	<b>4.71E-32</b>	2.30E-25	4.73E-03	1.46E-02	3.33E-12	<b>4.71E-32</b>	<b>4.71E-32</b>	5.29E-04
	Ort.	<b>4.71E-32</b>	4.72E-32	5.17E-17	5.58E-20	1.10E+05	3.30E-03	1.14E-01	1.58E-22	2.20E-02	4.78E-02	1.24E-01	<b>4.71E-32</b>	<b>4.71E-32</b>	5.92E-03
	Std.	1.67E-47	2.49E-34	1.01E-17	2.04E-20	3.66E+05	1.05E-02	2.08E-01	3.16E-22	8.57E-03	2.76E-02	2.66E-01	1.67E-47	1.67E-47	7.04E-03
$F_{13}$	En iyi	<b>1.35E-32</b>	<b>1.35E-32</b>	2.62E-17	1.06E-19	1.26E+01	3.36E-06	<b>1.35E-32</b>	5.67E-25	<b>1.35E-32</b>	8.04E-02	1.59E-11	<b>1.35E-32</b>	<b>1.35E-32</b>	9.05E-04
	Ort.	<b>1.35E-32</b>	<b>1.35E-32</b>	5.33E-17	5.71E-05	8.62E+05	1.32E-02	4.03E-03	6.35E-22	3.66E-04	2.06E-01	3.66E-04	<b>1.35E-32</b>	<b>1.35E-32</b>	1.58E-02
	Std.	5.57E-48	5.57E-48	1.57E-17	3.13E-04	2.39E+06	3.42E-02	5.39E-03	2.70E-21	2.01E-03	7.87E-02	2.01E-03	5.57E-48	5.57E-48	1.06E-02

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.11:** 30 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	En iyi	<b>-1.26E+04</b>	-8.31E+03	-1.26E+04	-3.87E+03	-6.85E+03	-7.50E+03	-1.02E+04	-1.16E+04	<b>-1.26E+04</b>	-4.86E+03	-9.21E+03	-1.19E+04	<b>-1.26E+04</b>	-1.26E+04
	Ort.	<b>-1.26E+04</b>	-6.59E+03	-1.26E+04	-2.66E+03	-4.92E+03	-6.24E+03	-8.49E+03	-1.03E+04	-1.12E+04	-4.39E+03	-7.86E+03	-1.06E+04	-1.23E+04	-1.26E+04
	Std.	1.85E-12	9.11E+02	3.06E-03	3.70E+02	1.13E+03	6.36E+02	8.15E+02	5.35E+02	2.27E+03	2.46E+02	7.31E+02	1.15E+03	5.20E+02	7.80E-01
$F_9$	En iyi	<b>0.00E+00</b>	1.19E+01	<b>0.00E+00</b>	9.95E+00	4.48E+01	<b>0.00E+00</b>	6.67E+01	4.97E+01	1.99E+01	<b>0.00E+00</b>	5.07E+01	6.96E+00	7.39E+01	1.00E-01
	Ort.	<b>0.00E+00</b>	2.49E+01	1.44E-13	1.37E+01	1.01E+02	<b>0.00E+00</b>	1.51E+02	7.92E+01	8.31E+01	4.33E-05	7.23E+01	2.54E+01	9.15E+01	2.44E-01
	Std.	0.00E+00	7.44E+00	1.37E-13	2.85E+00	3.68E+01	0.00E+00	4.13E+01	1.66E+01	5.95E+01	2.26E-04	1.62E+01	2.09E+01	7.68E+00	9.99E-02
$F_{10}$	En iyi	<b>8.88E-16</b>	7.99E-15	4.35E-14	2.01E-09	1.57E+01	4.44E-15	7.99E-15	2.93E-14	7.99E-15	4.44E-15	1.52E-05	4.44E-15	4.44E-15	9.79E-02
	Ort.	<b>1.01E-15</b>	1.04E-14	5.13E-14	2.64E-09	1.71E+01	7.64E-15	1.60E+01	3.10E-02	4.04E+00	1.12E+01	1.81E+00	7.52E-15	6.10E-15	2.35E-01
	Std.	6.49E-16	3.41E-15	6.75E-15	2.60E-10	8.79E-01	1.08E-15	6.79E+00	1.70E-01	7.28E+00	9.25E+00	8.10E-01	1.23E-15	1.80E-15	7.08E-02
$F_{11}$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	9.77E+01	<b>0.00E+00</b>	<b>0.00E+00</b>	1.11E-16	<b>0.00E+00</b>	<b>0.00E+00</b>	9.43E-09	<b>0.00E+00</b>	<b>0.00E+00</b>	3.06E-01
	Ort.	<b>0.00E+00</b>	1.36E-02	3.34E-15	<b>0.00E+00</b>	1.99E+02	<b>0.00E+00</b>	1.81E+01	1.71E-02	1.14E-02	<b>0.00E+00</b>	1.23E-02	2.47E-04	<b>0.00E+00</b>	5.25E-01
	Std.	0.00E+00	1.20E-02	6.88E-15	0.00E+00	6.89E+01	0.00E+00	4.37E+01	1.51E-02	1.26E-02	0.00E+00	1.21E-02	1.35E-03	0.00E+00	1.26E-01
$F_{12}$	En iyi	<b>1.57E-32</b>	<b>1.57E-32</b>	2.63E-16	3.25E-20	9.09E+00	6.55E-03	1.23E-31	4.09E-26	2.42E-01	2.01E-01	1.09E-11	<b>1.57E-32</b>	<b>1.57E-32</b>	4.85E-04
	Ort.	<b>1.57E-32</b>	1.64E-32	5.54E-16	7.72E-20	6.62E+03	2.80E-02	2.49E-01	6.84E-02	5.39E+00	3.33E-01	1.24E+00	1.57E-32	<b>1.57E-32</b>	2.15E-03
	Std.	5.57E-48	7.59E-34	1.15E-16	2.71E-20	3.49E+04	1.72E-02	7.29E-01	2.11E-01	3.42E+00	4.63E-02	2.13E+00	5.89E-35	5.57E-48	1.76E-03
$F_{13}$	En iyi	<b>1.35E-32</b>	<b>1.35E-32</b>	3.91E-16	6.08E-19	6.87E+01	6.63E-06	1.95E-30	4.24E-26	<b>1.35E-32</b>	1.83E+00	1.31E-10	<b>1.35E-32</b>	<b>1.35E-32</b>	8.97E-03
	Ort.	<b>1.35E-32</b>	3.66E-04	5.60E-16	1.20E-18	4.16E+05	3.96E-01	1.09E-01	2.56E-03	4.00E-03	2.01E+00	3.63E-03	1.36E-32	<b>1.35E-32</b>	2.58E-02
	Std.	5.57E-48	2.01E-03	7.82E-17	2.87E-19	1.33E+06	2.00E-01	3.87E-01	4.73E-03	6.01E-03	1.03E-01	5.91E-03	3.13E-34	5.57E-48	8.68E-03

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.12:** 50 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	En iyi	<b>-2.09E+04</b>	-1.31E+04	-2.09E+04	-4.54E+03	-1.27E+04	-1.08E+04	-1.65E+04	-1.82E+04	-2.09E+04	-6.59E+03	-1.47E+04	-1.91E+04	-1.42E+04	-2.09E+04
	Ort.	-2.08E+04	-1.08E+04	-2.09E+04	-3.43E+03	-8.43E+03	-9.02E+03	-1.39E+04	-1.66E+04	-1.96E+04	-5.91E+03	-1.25E+04	-1.73E+04	-1.20E+04	<b>-2.09E+04</b>
	Std.	1.08E+03	1.36E+03	4.09E+01	5.06E+02	1.35E+03	9.54E+02	1.33E+03	9.04E+02	2.26E+03	3.77E+02	7.02E+02	1.70E+03	6.51E+02	7.12E-01
$F_9$	En iyi	<b>0.00E+00</b>	3.58E+01	1.71E-13	1.89E+01	9.95E+01	<b>0.00E+00</b>	1.82E+02	9.05E+01	5.17E+01	<b>0.00E+00</b>	6.17E+01	2.49E+01	2.30E+02	2.25E-01
	Ort.	<b>0.00E+00</b>	6.06E+01	2.20E-11	2.56E+01	1.77E+02	<b>0.00E+00</b>	2.85E+02	1.42E+02	1.03E+02	5.31E-01	1.13E+02	4.73E+01	2.47E+02	3.71E-01
	Std.	0.00E+00	1.23E+01	8.84E-11	4.06E+00	7.27E+01	0.00E+00	5.08E+01	2.97E+01	2.94E+01	2.81E+00	1.97E+01	1.04E+01	9.15E+00	1.17E-01
$F_{10}$	En iyi	<b>8.88E-16</b>	1.51E-14	9.33E-14	2.05E-09	1.53E+01	7.99E-15	1.25E+01	5.77E-14	5.77E-14	7.99E-15	1.37E+00	1.96E-13	7.99E-15	1.63E-01
	Ort.	<b>1.60E-15</b>	2.28E-14	1.14E-13	2.82E-09	1.74E+01	8.47E-15	1.94E+01	1.45E-01	3.38E+00	1.22E+01	2.39E+00	4.64E-13	7.99E-15	2.42E-01
	Std.	1.45E-15	9.00E-15	1.47E-14	2.91E-10	1.07E+00	1.54E-15	1.42E+00	4.49E-01	3.37E+00	9.82E+00	5.44E-01	2.17E-13	0.00E+00	4.67E-02
$F_{11}$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	8.88E-16	<b>0.00E+00</b>	1.80E+02	<b>0.00E+00</b>	<b>0.00E+00</b>	3.33E-16	<b>0.00E+00</b>	<b>0.00E+00</b>	1.19E-08	<b>0.00E+00</b>	<b>0.00E+00</b>	3.79E-01
	Ort.	<b>0.00E+00</b>	5.26E-03	6.26E-15	5.45E-03	3.57E+02	<b>0.00E+00</b>	8.75E+01	5.01E-03	7.29E-03	1.45E-12	6.32E-03	<b>0.00E+00</b>	<b>0.00E+00</b>	6.06E-01
	Std.	0.00E+00	6.43E-03	7.51E-15	1.87E-02	1.20E+02	0.00E+00	9.88E+01	6.45E-03	1.16E-02	6.76E-12	9.74E-03	0.00E+00	0.00E+00	1.38E-01
$F_{12}$	En iyi	<b>9.42E-33</b>	1.02E-32	1.04E-15	5.06E-20	1.02E+01	3.53E-02	5.07E-29	1.25E-26	9.08E+00	4.86E-01	7.98E-01	1.38E-13	<b>9.42E-33</b>	6.41E-04
	Ort.	<b>9.42E-33</b>	7.39E-32	1.40E-15	2.07E-03	2.46E+01	6.65E-02	2.56E+07	2.46E-01	1.94E+01	5.93E-01	6.52E+00	3.17E-01	<b>9.42E-33</b>	2.18E-03
	Std.	2.78E-48	1.72E-31	1.82E-16	1.14E-02	8.57E+00	2.28E-02	7.81E+07	4.26E-01	5.20E+00	8.34E-02	3.19E+00	1.12E+00	2.78E-48	2.82E-03
$F_{13}$	En iyi	<b>1.35E-32</b>	2.83E-32	7.87E-16	1.57E-18	1.33E+02	7.23E-01	2.70E-26	6.50E-26	<b>1.35E-32</b>	3.73E+00	3.23E-10	9.73E-13	<b>1.35E-32</b>	1.84E-02
	Ort.	<b>1.35E-32</b>	2.56E-03	1.34E-15	2.26E-18	8.72E+03	1.45E+00	4.10E+07	3.30E-03	1.98E+00	4.19E+00	5.49E-03	4.01E-04	<b>1.35E-32</b>	4.15E-02
	Std.	5.57E-48	4.73E-03	2.19E-16	4.29E-19	3.18E+04	3.11E-01	1.25E+08	5.12E-03	5.82E+00	1.98E-01	5.59E-03	2.00E-03	5.57E-48	1.29E-02

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.13:** Sabit boyutlu çok-modlu test fonksiyonları ( $F_{14} - F_{23}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_{14}$	En iyi	<b>9.98E-01</b>	<b>9.98E-01</b>	<b>9.98E-01</b>	9.99E-01	9.98E-01	9.98E-01	<b>9.98E-01</b>	<b>9.98E-01</b>	9.98E-01	9.98E-01	9.98E-01	<b>9.98E-01</b>	<b>9.98E-01</b>	9.98E-01
	Ort.	<b>9.98E-01</b>	2.71E+00	9.98E-01	5.53E+00	1.21E+01	5.73E+00	2.02E+00	9.98E-01	1.00E+00	1.92E+00	1.06E+00	<b>9.98E-01</b>	<b>9.98E-01</b>	4.29E+00
	Std.	0.00E+00	2.53E+00	1.13E-16	4.08E+00	6.89E+00	4.53E+00	1.95E+00	1.91E-16	5.89E-03	1.91E+00	2.52E-01	0.00E+00	0.00E+00	5.30E+00
$F_{15}$	En iyi	<b>3.07E-04</b>	3.07E-04	4.09E-04	1.11E-03	7.32E-04	3.07E-04	5.85E-04	3.08E-04	3.07E-04	3.32E-04	4.01E-04	3.82E-04	3.07E-04	7.34E-04
	Ort.	<b>3.07E-04</b>	7.54E-04	7.15E-04	2.55E-03	6.61E-03	4.35E-03	1.45E-03	7.85E-04	4.65E-04	9.65E-04	8.29E-04	6.29E-04	3.83E-04	3.17E-03
	Std.	1.71E-19	2.16E-04	1.86E-04	1.27E-03	2.13E-02	8.14E-03	1.90E-03	3.14E-04	3.56E-04	4.19E-04	2.64E-04	9.82E-05	2.42E-04	3.84E-03
$F_{16}$	En iyi	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.03E+00	-1.03E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.03E+00	-1.03E+00	-1.03E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.03E+00
	Ort.	-1.03E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.00E+00	-1.03E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.03E+00	-1.03E+00	-1.03E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>	-1.03E+00
	Std.	1.08E-15	6.32E-16	4.97E-16	5.05E-16	1.49E-01	3.18E-08	6.78E-16	5.00E-16	2.32E-05	3.06E-05	3.02E-14	6.71E-16	6.78E-16	1.30E-05
$F_{17}$	En iyi	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	3.98E-01	3.98E-01	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	3.98E-01	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	3.98E-01
	Ort.	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	3.98E-01	3.98E-01	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	3.99E-01	3.98E-01	<b>3.98E-01</b>	<b>3.98E-01</b>	3.98E-01
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.98E-10	7.02E-07	0.00E+00	0.00E+00	0.00E+00	1.00E-03	2.32E-14	0.00E+00	0.00E+00	1.64E-04
$F_{18}$	En iyi	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	<b>3.00E+00</b>	3.00E+00
	Ort.	3.00E+00	3.00E+00	3.01E+00	3.00E+00	8.40E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	<b>3.00E+00</b>	3.00E+00
	Std.	1.41E-15	1.47E-15	4.26E-02	4.04E-15	2.06E+01	2.41E-05	2.33E-15	3.70E-15	4.01E-04	4.89E-05	2.18E-13	1.23E-15	1.81E-15	7.63E-04
$F_{19}$	En iyi	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.86E+00	-3.86E+00	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.86E+00	-3.86E+00	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.86E+00
	Ort.	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.84E+00	-3.86E+00	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.86E+00	-3.86E+00	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.86E+00
	Std.	2.71E-15	2.68E-15	2.34E-15	2.40E-15	1.41E-01	2.61E-03	2.71E-15	2.07E-15	2.71E-15	2.56E-03	6.64E-14	2.71E-15	2.71E-15	5.32E-05
$F_{20}$	En iyi	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	<b>-3.32E+00</b>	-3.32E+00	<b>-3.32E+00</b>	-3.23E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
	Ort.	-3.31E+00	-3.28E+00	<b>-3.32E+00</b>	<b>-3.32E+00</b>	-3.27E+00	-3.25E+00	-3.26E+00	-3.27E+00	-3.25E+00	-2.88E+00	-3.22E+00	<b>-3.32E+00</b>	-3.29E+00	-3.24E+00
	Std.	4.11E-02	5.83E-02	1.76E-15	1.36E-15	6.03E-02	1.07E-01	6.40E-02	6.03E-02	5.92E-02	4.22E-01	4.11E-02	1.37E-15	5.54E-02	5.54E-02
$F_{21}$	En iyi	<b>-1.02E+01</b>	<b>-1.02E+01</b>	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	<b>-1.02E+01</b>	<b>-1.02E+01</b>	<b>-1.02E+01</b>	-6.64E+00	-1.02E+01	<b>-1.02E+01</b>	<b>-1.02E+01</b>	-1.02E+01
	Ort.	<b>-1.02E+01</b>	-8.55E+00	-1.02E+01	-7.39E+00	-5.47E+00	-9.48E+00	-6.80E+00	<b>-1.02E+01</b>	-7.45E+00	-2.83E+00	-8.06E+00	-1.01E+01	<b>-1.02E+01</b>	-6.23E+00
	Std.	6.27E-15	2.52E+00	6.82E-07	3.50E+00	3.06E+00	1.75E+00	3.31E+00	5.86E-15	2.77E+00	2.18E+00	3.09E+00	5.18E-01	7.23E-15	3.74E+00
$F_{22}$	En iyi	<b>-1.04E+01</b>	<b>-1.04E+01</b>	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	<b>-1.04E+01</b>	<b>-1.04E+01</b>	<b>-1.04E+01</b>	-7.09E+00	-1.04E+01	<b>-1.04E+01</b>	<b>-1.04E+01</b>	-1.04E+01
	Ort.	<b>-1.04E+01</b>	-1.00E+01	-1.04E+01	<b>-1.04E+01</b>	-5.62E+00	-9.87E+00	-8.05E+00	<b>-1.04E+01</b>	-8.18E+00	-3.89E+00	-9.01E+00	-1.03E+01	<b>-1.04E+01</b>	-5.86E+00
	Std.	6.60E-16	1.34E+00	1.42E-03	8.08E-16	3.33E+00	1.62E+00	3.21E+00	1.19E-15	2.97E+00	1.93E+00	2.61E+00	3.00E-01	1.40E-15	3.54E+00
$F_{23}$	En iyi	-1.05E+01	<b>-1.05E+01</b>	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	<b>-1.05E+01</b>	-1.05E+01	-1.05E+01	-8.34E+00	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
	Ort.	<b>-1.05E+01</b>	-9.92E+00	-1.05E+01	-1.05E+01	-4.62E+00	-1.03E+01	-7.09E+00	-1.05E+01	-8.14E+00	-4.87E+00	-9.74E+00	-1.05E+01	<b>-1.05E+01</b>	-6.11E+00
	Std.	2.56E-15	1.91E+00	5.76E-03	1.98E-15	3.41E+00	1.48E+00	3.81E+00	1.62E-15	3.07E+00	1.95E+00	2.09E+00	2.70E-05	1.81E-15	3.70E+00

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.14:** Bileşik test fonksiyonları ( $F_{24} - F_{29}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_{24}$	En iyi	<b>0.00E+00</b>	<b>0.00E+00</b>	1.59E-16	3.48E-18	4.92E-05	6.01E-01	<b>0.00E+00</b>	<b>0.00E+00</b>	6.74E+00	4.17E+01	2.83E-11	<b>0.00E+00</b>	<b>0.00E+00</b>	2.50E-02
	Ort.	<b>8.44E-30</b>	8.00E+01	4.56E-03	9.48E-18	1.37E+02	8.64E+01	9.08E+01	2.33E+01	5.19E+01	1.03E+02	6.00E+01	5.81E+00	4.20E+01	2.67E+01
	Std.	1.73E-29	8.05E+01	9.47E-03	3.53E-18	1.33E+02	1.03E+02	5.85E+01	5.04E+01	4.57E+01	3.81E+01	6.75E+01	1.90E+01	5.89E+01	4.50E+01
$F_{25}$	En iyi	1.77E+00	3.18E+00	4.99E+00	1.00E+02	3.57E+01	6.87E+00	0.00E+00	1.03E+01	4.74E+01	7.60E+01	6.18E+00	1.88E-01	<b>0.00E+00</b>	1.74E+00
	Ort.	<b>1.90E+01</b>	1.16E+02	2.05E+01	1.89E+02	3.96E+02	1.30E+02	4.39E+01	2.54E+01	1.11E+02	1.03E+02	2.19E+01	4.31E+01	5.88E+01	1.00E+02
	Std.	2.82E+01	9.48E+01	1.24E+01	3.06E+01	1.88E+02	1.11E+02	4.87E+01	1.08E+01	7.34E+01	8.53E+00	1.00E+01	4.52E+01	7.27E+01	8.99E+01
$F_{26}$	En iyi	1.11E+02	9.60E+01	9.79E+01	<b>0.00E+00</b>	2.53E+02	9.51E+01	1.26E+02	1.01E+02	1.88E+02	2.21E+02	1.01E+02	1.06E+02	9.43E+01	1.05E+02
	Ort.	1.58E+02	1.62E+02	1.31E+02	<b>5.70E+01</b>	6.28E+02	1.93E+02	2.49E+02	1.85E+02	2.86E+02	3.21E+02	1.92E+02	1.49E+02	1.31E+02	1.73E+02
	Std.	2.17E+01	3.78E+01	1.61E+01	7.09E+01	1.94E+02	5.49E+01	7.62E+01	4.44E+01	6.19E+01	6.65E+01	5.31E+01	2.81E+01	2.37E+01	3.74E+01
$F_{27}$	En iyi	2.00E+02	2.28E+02	2.94E+02	2.00E+02	5.02E+02	2.24E+02	3.16E+02	2.28E+02	3.40E+02	3.71E+02	2.62E+02	2.18E+02	<b>2.00E+02</b>	2.97E+02
	Ort.	<b>2.86E+02</b>	3.10E+02	3.62E+02	4.84E+02	7.53E+02	4.03E+02	3.54E+02	3.07E+02	3.62E+02	4.09E+02	3.17E+02	3.14E+02	2.92E+02	4.09E+02
	Std.	2.50E+01	2.32E+01	2.88E+01	1.50E+02	1.45E+02	1.49E+02	2.83E+01	3.38E+01	2.53E+01	2.02E+01	2.74E+01	2.59E+01	2.07E+01	9.87E+01
$F_{28}$	En iyi	<b>0.00E+00</b>	1.19E+00	5.78E+00	3.94E-15	1.96E+02	1.34E+00	4.37E+00	<b>0.00E+00</b>	1.12E+01	4.45E+01	1.19E+00	2.25E-03	<b>0.00E+00</b>	6.25E+00
	Ort.	<b>2.48E+00</b>	9.71E+01	1.01E+01	2.17E+02	5.47E+02	1.43E+02	8.39E+01	3.52E+00	1.03E+02	9.90E+01	2.43E+01	3.19E+01	5.60E+01	8.99E+01
	Std.	1.24E+00	8.97E+01	3.25E+00	4.74E+01	2.02E+02	1.63E+02	7.40E+01	2.25E+00	9.19E+01	5.32E+01	3.92E+01	3.58E+01	5.20E+01	1.14E+02
$F_{29}$	En iyi	<b>4.01E+02</b>	5.00E+02	4.10E+02	5.58E+02	5.67E+02	5.01E+02	4.41E+02	4.05E+02	5.25E+02	4.30E+02	4.02E+02	4.37E+02	5.00E+02	5.01E+02
	Ort.	5.24E+02	7.74E+02	5.03E+02	7.16E+02	8.53E+02	8.60E+02	7.09E+02	<b>4.98E+02</b>	7.90E+02	5.17E+02	5.25E+02	5.19E+02	6.52E+02	6.45E+02
	Std.	1.04E+02	1.84E+02	6.51E+01	8.09E+01	1.09E+02	1.22E+02	1.90E+02	1.77E+01	1.64E+02	7.10E+01	1.22E+02	7.47E+01	1.89E+02	1.90E+02

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.15:** CEC 2019 test fonksiyonları ( $F_{30} - F_{39}$ ) üzerinde algoritmaların elde ettiği sonuçlar.

$F$		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_{30}$	En iyi	<b>1.00E+00</b>	9.03E+03	4.60E+05	1.70E+08	1.32E+08	8.22E+00	1.07E+03	3.04E+04	1.32E+06	2.98E+01	7.56E+03	2.09E+05	4.10E+00	5.24E+05
	Ort.	<b>1.00E+00</b>	2.87E+07	2.03E+06	4.04E+08	9.68E+08	5.88E+03	1.64E+07	5.27E+05	4.49E+06	4.51E+05	5.24E+05	1.10E+06	1.39E+04	1.23E+07
	Std.	0.00E+00	3.54E+07	8.30E+05	1.80E+08	6.87E+08	2.94E+04	3.33E+07	3.17E+05	2.02E+06	9.84E+05	4.28E+05	8.86E+05	5.13E+04	9.70E+06
$F_{31}$	En iyi	<b>2.83E+00</b>	1.54E+03	7.38E+02	1.23E+04	1.10E+04	4.84E+00	1.12E+02	1.20E+02	1.20E+03	1.40E+02	1.00E+02	1.07E+03	5.15E+02	1.17E+03
	Ort.	<b>3.21E+00</b>	6.40E+03	2.07E+03	2.24E+04	2.15E+04	1.62E+02	2.05E+03	3.08E+02	2.77E+03	1.59E+03	3.71E+02	2.33E+03	1.06E+03	3.94E+03
	Std.	2.14E-01	4.18E+03	5.92E+02	5.41E+03	8.11E+03	1.75E+02	2.79E+03	9.59E+01	6.60E+02	1.14E+03	2.46E+02	6.42E+02	2.60E+02	2.06E+03
$F_{32}$	En iyi	<b>1.00E+00</b>	<b>1.00E+00</b>	1.06E+00	1.41E+00	1.41E+00	1.00E+00	1.41E+00	1.00E+00	3.98E+00	3.72E+00	1.00E+00	3.37E+00	3.28E+00	1.07E+00
	Ort.	<b>1.33E+00</b>	1.50E+00	1.52E+00	2.21E+00	7.86E+00	1.88E+00	6.55E+00	3.07E+00	7.46E+00	7.26E+00	2.41E+00	6.90E+00	4.97E+00	2.24E+00
	Std.	1.66E-01	6.07E-01	1.94E-01	1.56E+00	3.17E+00	1.18E+00	2.58E+00	2.02E+00	1.65E+00	1.50E+00	1.45E+00	1.64E+00	7.89E-01	1.28E+00
$F_{33}$	En iyi	4.98E+00	1.69E+01	5.13E+00	2.39E+01	3.28E+01	5.00E+00	1.49E+01	6.97E+00	2.03E+01	3.25E+01	5.97E+00	5.13E+00	<b>1.00E+00</b>	8.13E+00
	Ort.	1.29E+01	4.04E+01	9.93E+00	3.40E+01	8.81E+01	1.59E+01	3.03E+01	1.64E+01	3.23E+01	4.41E+01	2.25E+01	1.60E+01	<b>9.56E+00</b>	1.70E+01
	Std.	4.41E+00	1.32E+01	1.87E+00	7.56E+00	3.39E+01	7.74E+00	1.35E+01	6.75E+00	7.86E+00	6.03E+00	9.52E+00	5.83E+00	4.37E+00	6.30E+00
$F_{34}$	En iyi	1.01E+00	1.03E+00	1.00E+00	<b>1.00E+00</b>	3.75E+01	1.17E+00	1.04E+00	1.04E+00	2.10E+00	2.75E+00	1.06E+00	1.01E+00	<b>1.00E+00</b>	1.22E+00
	Ort.	1.07E+00	1.15E+00	1.02E+00	<b>1.00E+00</b>	1.13E+02	1.69E+00	4.17E+00	1.25E+00	2.27E+00	6.44E+00	1.28E+00	1.17E+00	1.05E+00	1.49E+00
	Std.	3.87E-02	8.88E-02	1.03E-02	5.35E-03	5.04E+01	6.37E-01	8.03E+00	1.20E-01	1.10E-01	2.42E+00	1.10E-01	1.17E-01	6.30E-02	1.55E-01
$F_{35}$	En iyi	1.00E+00	1.00E+00	1.43E+00	1.00E+00	1.18E+01	1.12E+00	2.22E+00	1.00E+00	3.72E+00	4.96E+00	1.20E+00	<b>1.00E+00</b>	<b>1.00E+00</b>	1.41E+00
	Ort.	1.76E+00	3.66E+00	2.34E+00	2.98E+00	1.42E+01	2.46E+00	5.38E+00	1.35E+00	5.85E+00	7.02E+00	3.41E+00	<b>1.21E+00</b>	1.85E+00	3.33E+00
	Std.	5.49E-01	1.37E+00	5.13E-01	1.36E+00	1.10E+00	1.19E+00	1.76E+00	6.25E-01	1.36E+00	1.47E+00	1.63E+00	4.06E-01	7.67E-01	1.04E+00
$F_{36}$	En iyi	2.42E+02	1.35E+02	3.38E+01	7.07E+02	5.37E+02	1.80E+02	3.59E+02	1.33E+02	7.74E+02	6.22E+02	1.28E+02	2.93E+02	<b>3.30E+01</b>	3.72E+01
	Ort.	3.95E+02	8.01E+02	<b>1.88E+02</b>	1.16E+03	1.36E+03	5.09E+02	1.03E+03	5.11E+02	1.18E+03	1.14E+03	7.63E+02	8.94E+02	3.09E+02	3.41E+02
	Std.	7.05E+01	2.88E+02	7.50E+01	2.70E+02	4.22E+02	1.77E+02	3.73E+02	2.36E+02	2.00E+02	2.08E+02	3.63E+02	2.80E+02	1.73E+02	2.27E+02
$F_{37}$	En iyi	2.23E+00	3.14E+00	3.04E+00	4.60E+00	4.36E+00	2.67E+00	3.42E+00	<b>1.96E+00</b>	3.76E+00	3.37E+00	3.06E+00	3.54E+00	2.53E+00	2.67E+00
	Ort.	<b>3.12E+00</b>	4.04E+00	3.41E+00	5.27E+00	5.25E+00	3.62E+00	4.53E+00	3.55E+00	4.20E+00	4.19E+00	3.75E+00	3.87E+00	3.24E+00	3.69E+00
	Std.	3.88E-01	4.85E-01	2.32E-01	2.17E-01	2.73E-01	4.03E-01	3.42E-01	5.27E-01	2.30E-01	3.14E-01	3.70E-01	1.89E-01	2.86E-01	4.47E-01
$F_{38}$	En iyi	1.04E+00	1.09E+00	1.10E+00	<b>1.01E+00</b>	2.41E+00	1.09E+00	1.10E+00	1.03E+00	1.20E+00	1.22E+00	1.10E+00	1.11E+00	1.09E+00	1.16E+00
	Ort.	1.09E+00	1.19E+00	1.15E+00	<b>1.03E+00</b>	4.22E+00	1.17E+00	1.35E+00	1.15E+00	1.41E+00	1.49E+00	1.21E+00	1.17E+00	1.14E+00	1.37E+00
	Std.	2.27E-02	7.68E-02	2.55E-02	1.30E-02	9.55E-01	8.41E-02	1.66E-01	7.44E-02	8.38E-02	1.15E-01	9.41E-02	2.91E-02	2.61E-02	9.64E-02
$F_{39}$	En iyi	2.16E+00	2.10E+01	2.83E+00	1.00E+00	2.10E+01	7.07E+00	2.10E+01	<b>1.00E+00</b>	2.11E+01	2.12E+01	2.16E+00	2.12E+01	2.02E+01	2.10E+01
	Ort.	<b>1.22E+01</b>	2.11E+01	1.85E+01	1.83E+01	2.10E+01	2.09E+01	2.12E+01	1.90E+01	2.13E+01	2.14E+01	2.04E+01	2.13E+01	2.11E+01	2.10E+01
	Std.	9.09E+00	9.21E-02	5.91E+00	6.91E+00	5.61E-05	2.61E+00	1.73E-01	6.10E+00	9.05E-02	7.17E-02	3.44E+00	6.57E-02	2.30E-01	1.71E-02

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 3.16:** 10 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_1$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	=	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_2$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_3$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_4$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.72E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_5$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	3.51E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1	1.73E-06
$F_6$ (T+, T-)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 406)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 0)	(0, 465)
KD	=	+	+	+	+	+	+	+	+	+	=	=	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	0.39333	1.73E-06	1.92E-06	1.73E-06	5.79E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_7$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(274, 191)	(0, 465)	(1, 464)	(0, 465)	(37, 428)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	=	+	+	+	+	+	+	+	+
<b>Toplam (+/=/-)</b>	<b>6/1/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>5/2/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>6/1/0</b>	<b>6/1/0</b>	<b>7/0/0</b>

**Tablo 3.17:** 30 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1.72E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_1$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	=	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1.59E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_2$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	=	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_3$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_4$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	3.52E-06	2.61E-04	1.73E-06	1.73E-06	1.73E-06	1.02E-05	1.73E-06	2.61E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_5$ (T+, T-)	(7, 458)	(410, 55)	(0, 465)	(0, 465)	(0, 465)	(18, 447)	(0, 465)	(410, 55)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	-	+	+	+	+	+	-	+	+	+	+	+
$p$ -değeri	2.45E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	0.5	1	1.73E-06
$F_6$ (T+, T-)	(0, 435)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 3)	(0, 0)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	=	=	+
$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	0.46528	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
$F_7$ (T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(197, 268)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	=	+	+	+	+	+	+	+	+
<b>Toplam (+/=/-)</b>	<b>7/0/0</b>	<b>6/0/1</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>4/3/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>6/0/1</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>6/1/0</b>	<b>6/1/0</b>	<b>7/0/0</b>



**Tablo 3.18:** 50 boyutlu tek-modlu test fonksiyonları ( $F_1 - F_7$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES	
$F_1$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1 (0, 0) =	1.36E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	
$F_2$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1 (0, 0) =	1.68E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	
$F_3$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	
$F_4$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	
$F_5$	$p$ -değeri (T+, T-) KD	3.38E-03 (90, 375) +	3.18E-06 (459, 6) -	2.77E-03 (378, 87) -	0.42843 (271, 194) =	1.73E-06 (0, 465) +	2.05E-04 (52, 413) +	1.73E-06 (0, 465) +	3.18E-06 (459, 6) -	1.73E-06 (0, 465) +	1.15E-04 (45, 420) +	3.16E-03 (89, 376) +	0.01480 (351, 114) -	1.73E-06 (0, 465) +
$F_6$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.66E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1 (0, 0) =	1.73E-06 (0, 465) +
$F_7$	$p$ -değeri (T+, T-) KD	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.64E-05 (23, 442) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +	1.73E-06 (0, 465) +
<b>Toplam (+/-/-)</b>	<b>7/0/0</b>	<b>6/0/1</b>	<b>6/0/1</b>	<b>6/1/0</b>	<b>5/2/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>6/0/1</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>7/0/0</b>	<b>5/1/1</b>	<b>7/0/0</b>	

**Tablo 3.19:** 10 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$		PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	$p$ -değeri	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	1.72E-06	2.65E-05	2.93E-04	1.73E-06	1.73E-06	1.46E-04	0.25	1.73E-06
	(T+, T-)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 276)	(0, 153)	(0, 465)	(0, 465)	(0, 171)	(0, 6)	(0, 465)
	KD	+	=	+	+	+	+	+	+	+	+	+	=	+
$F_9$	$p$ -değeri	1.11E-05	1	1.56E-06	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	1	1.73E-06	1.19E-05	1	1.73E-06
	(T+, T-)	(0, 325)	(0, 0)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 325)	(0, 0)	(0, 465)
	KD	+	=	+	+	=	+	+	+	=	+	+	=	+
$F_{10}$	$p$ -değeri	6.80E-08	1.01E-07	1.73E-06	1.73E-06	4.32E-08	2.40E-07	1.24E-06	4.32E-08	4.59E-06	1.73E-06	4.32E-08	2.03E-07	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 435)	(0, 465)	(0, 465)	(0, 231)	(0, 465)	(0, 465)	(0, 378)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_{11}$	$p$ -değeri	1.73E-06	5.60E-06	4.88E-04	1.73E-06	0.01563	1.73E-06	1.73E-06	1.73E-06	0.5	1.73E-06	2.56E-06	1	1.73E-06
	(T+, T-)	(0, 465)	(0, 378)	(0, 78)	(0, 465)	(0, 28)	(0, 465)	(0, 465)	(0, 465)	(0, 3)	(0, 465)	(0, 435)	(0, 0)	(0, 465)
	KD	+	+	+	+	+	+	+	+	=	+	+	=	+
$F_{12}$	$p$ -değeri	0.125	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.78E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1	1.73E-06
	(T+, T-)	(0, 10)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 300)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 0)	(0, 465)
	KD	=	+	+	+	+	+	+	+	+	+	=	=	+
$F_{13}$	$p$ -değeri	1	1.73E-06	1.73E-06	1.73E-06	1.73E-06	6.81E-06	1.73E-06	1	1.73E-06	1.73E-06	1	1	1.73E-06
	(T+, T-)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 351)	(0, 465)	(0, 1)	(0, 465)	(0, 465)	(0, 0)	(0, 0)	(0, 465)
	KD	=	+	+	+	+	+	+	+	+	+	=	=	+
<b>Toplam (+/=/-)</b>		<b>4/2/0</b>	<b>4/2/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>4/2/0</b>	<b>6/0/0</b>	<b>4/2/0</b>	<b>1/5/0</b>	<b>6/0/0</b>

**Tablo 3.20:** 30 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$		PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	$p$ -değeri	1.73E-06	1.54E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	6.10E-05	1.73E-06	1.73E-06	1.73E-06	1.61E-05	1.73E-06
	(T+, T-)	(0, 465)	(0, 300)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 120)	(0, 465)	(0, 465)	(0, 465)	(0, 300)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_9$	$p$ -değeri	1.72E-06	2.58E-06	1.68E-06	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	0.25	1.73E-06	1.73E-06	1.73E-06	1.73E-06
	(T+, T-)	(0, 465)	(0, 406)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 6)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
	KD	+	+	+	+	=	+	+	+	=	+	+	+	+
$F_{10}$	$p$ -değeri	6.47E-07	1.36E-06	1.73E-06	1.73E-06	1.96E-07	1.73E-06	1.51E-06	1.70E-06	1.71E-06	1.73E-06	2.57E-07	1.24E-06	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 435)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_{11}$	$p$ -değeri	1.18E-05	2.52E-06	1	1.73E-06	1	3.99E-05	1.73E-06	1.31E-04	1	1.73E-06	0.5	1	1.73E-06
	(T+, T-)	(0, 325)	(0, 435)	(0, 0)	(0, 465)	(0, 0)	(0, 253)	(0, 465)	(0, 190)	(0, 0)	(0, 465)	(0, 3)	(0, 0)	(0, 465)
	KD	+	+	=	+	=	+	+	+	=	+	=	=	+
$F_{12}$	$p$ -değeri	3.65E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1	1.73E-06
	(T+, T-)	(0, 406)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 1)	(0, 0)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	=	=	+
$F_{13}$	$p$ -değeri	1.16E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.37E-06	1.73E-06	7.61E-05	1.73E-06	1.73E-06	0.5	1	1.73E-06
	(T+, T-)	(0, 325)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 210)	(0, 465)	(0, 465)	(0, 3)	(0, 0)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	=	=	+
<b>Toplam (+/=/-)</b>		<b>6/0/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>6/0/0</b>	<b>4/2/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>4/2/0</b>	<b>6/0/0</b>	<b>3/3/0</b>	<b>3/3/0</b>	<b>6/0/0</b>

**Tablo 3.21:** 50 boyutlu çok-modlu test fonksiyonları ( $F_8 - F_{13}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$		PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$F_8$	$p$ -değeri	1.73E-06	3.11E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.92E-06	4.90E-05	1.73E-06	1.73E-06	3.18E-06	1.73E-06	3.11E-05
	(T+, T-)	(0, 465)	(30, 435)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(1, 464)	(25, 381)	(0, 465)	(0, 465)	(6, 459)	(0, 465)	(30, 435)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_9$	$p$ -değeri	1.73E-06	1.73E-06	1.71E-06	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	0.01563	1.73E-06	1.73E-06	1.73E-06	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 28)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
	KD	+	+	+	+	=	+	+	+	+	+	+	+	+
$F_{10}$	$p$ -değeri	1.53E-06	1.68E-06	1.73E-06	1.73E-06	4.11E-07	1.73E-06	1.69E-06	1.73E-06	1.71E-06	1.73E-06	1.73E-06	3.26E-07	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_{11}$	$p$ -değeri	6.10E-05	1.72E-06	0.25	1.73E-06	1	8.30E-06	1.72E-06	5.54E-06	0.5	1.73E-06	1	1	1.73E-06
	(T+, T-)	(0, 120)	(0, 465)	(0, 6)	(0, 465)	(0, 0)	(0, 351)	(0, 465)	(0, 378)	(0, 3)	(0, 465)	(0, 0)	(0, 0)	(0, 465)
	KD	+	+	=	+	=	+	+	+	=	+	=	=	+
$F_{12}$	$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.71E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	=	+
$F_{13}$	$p$ -değeri	1.67E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.61E-06	1.72E-06	2.36E-06	1.73E-06	1.73E-06	1.73E-06	1	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 435)	(0, 465)	(0, 465)	(0, 465)	(0, 0)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	=	+
<b>Toplam (+/=/-)</b>		<b>6/0/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>6/0/0</b>	<b>4/2/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>3/3/0</b>	<b>6/0/0</b>

**Tablo 3.22:** Sabit boyutlu çok-modlu test fonksiyonları ( $F_{14} - F_{23}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES	
$F_{14}$	$p$ -değeri	1.38E-05	1	1.73E-06	1.73E-06	1.73E-06	6.10E-05	0.01563	1.73E-06	1.73E-06	2.19E-06	1	1	1.73E-06
	(T+, T-)	(0, 253)	(0, 0)	(0, 465)	(0, 465)	(0, 465)	(0, 120)	(0, 28)	(0, 465)	(0, 465)	(0, 300)	(0, 0)	(0, 0)	(0, 465)
	KD	+	=	+	+	+	+	+	+	+	+	=	=	+
$F_{15}$	$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.72E-06	1.73E-06	8.11E-06	1.73E-06	1.73E-06	1.73E-06	8.00E-04	1.73E-06
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(3.5, 374.5)	(0, 465)	(0, 465)	(0, 465)	(16.5, 193.5)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$F_{16}$	$p$ -değeri	7.81E-03	0.03125	0.01563	1.73E-06	1.73E-06	3.91E-03	0.03125	1.73E-06	1.73E-06	5.90E-06	3.91E-03	3.91E-03	1.73E-06
	(T+, T-)	(36, 0)	(21, 0)	(28, 0)	(0, 465)	(0, 465)	(45, 0)	(21, 0)	(0, 465)	(0, 465)	(15, 450)	(45, 0)	(45, 0)	(0, 465)
	KD	-	-	-	+	+	-	-	+	+	+	-	-	+
$F_{17}$	$p$ -değeri	1	1	1	1.73E-06	1.73E-06	1	1	1	1.73E-06	1.15E-05	1	1	1.73E-06
	(T+, T-)	(0, 0)	(0, 0)	(0, 0)	(0, 465)	(0, 465)	(0, 0)	(0, 0)	(0, 0)	(0, 465)	(0, 325)	(0, 0)	(0, 0)	(0, 465)
	KD	=	=	=	+	+	=	=	=	+	+	=	=	+
$F_{18}$	$p$ -değeri	0.65472	2.47E-04	2.55E-06	1.73E-06	1.73E-06	1	1.57E-04	1.73E-06	1.73E-06	1.73E-06	3.47E-04	4.22E-06	1.73E-06
	(T+, T-)	(115.5, 94.5)	(36, 315)	(12.5, 393.5)	(0, 465)	(0, 465)	(105, 105)	(37.5, 287.5)	(0, 465)	(0, 465)	(0, 465)	(189, 21)	(312, 13)	(0, 465)
	KD	=	+	+	+	+	=	+	+	+	+	-	-	+
$F_{19}$	$p$ -değeri	1	1	1	1.73E-06	1.73E-06	1	0.06250	1	1.73E-06	1.42E-06	1	1	1.73E-06
	(T+, T-)	(0, 0)	(0, 0)	(0, 0)	(0, 465)	(0, 465)	(0, 0)	(0, 15)	(0, 0)	(0, 465)	(0, 465)	(0, 0)	(0, 0)	(0, 465)
	KD	=	=	=	+	+	=	=	=	+	+	=	=	+
$F_{20}$	$p$ -değeri	0.11847	0.125	0.125	4.86E-05	4.53E-04	0.01071	1.39E-03	4.65E-04	1.73E-06	3.52E-06	0.125	0.22656	5.75E-06
	(T+, T-)	(32, 88)	(10, 0)	(10, 0)	(35, 430)	(62, 403)	(27, 126)	(9, 144)	(8.5, 127.5)	(0, 465)	(7, 458)	(10, 0)	(18, 48)	(12, 453)
	KD	=	=	=	+	+	+	+	+	+	+	=	=	+
$F_{21}$	$p$ -değeri	3.91E-03	0.125	4.88E-04	1.73E-06	1.73E-06	3.82E-04	1	8.67E-05	1.73E-06	1.73E-06	0.03125	1	1.73E-06
	(T+, T-)	(0, 45)	(0, 10)	(0, 78)	(0, 465)	(0, 465)	(0, 136)	(0, 0)	(0, 210)	(0, 465)	(0, 465)	(0, 21)	(0, 0)	(0, 465)
	KD	+	=	+	+	+	+	=	+	+	+	+	=	+
$F_{22}$	$p$ -değeri	0.5	0.25	1	1.73E-06	1.73E-06	9.77E-04	1	2.44E-04	1.73E-06	1.73E-06	0.03125	1	1.73E-06
	(T+, T-)	(0, 3)	(0, 6)	(0, 0)	(0, 465)	(0, 465)	(0, 66)	(0, 0)	(0, 91)	(0, 465)	(0, 465)	(0, 21)	(0, 0)	(0, 465)
	KD	=	=	=	+	+	+	=	+	+	+	+	=	+
$F_{23}$	$p$ -değeri	0.25	9.77E-04	1	1.73E-06	1.73E-06	1.22E-04	1	2.44E-04	1.73E-06	1.73E-06	0.125	1	1.73E-06
	(T+, T-)	(0, 6)	(0, 66)	(0, 0)	(0, 465)	(0, 465)	(0, 105)	(0, 0)	(0, 91)	(0, 465)	(0, 465)	(0, 10)	(0, 0)	(0, 465)
	KD	=	+	=	+	+	+	=	+	+	+	=	=	+
<b>Toplam (+/=-)</b>	<b>3/6/1</b>	<b>3/6/1</b>	<b>4/5/1</b>	<b>10/0/0</b>	<b>10/0/0</b>	<b>6/3/1</b>	<b>4/5/1</b>	<b>8/2/0</b>	<b>10/0/0</b>	<b>10/0/0</b>	<b>3/5/2</b>	<b>1/7/2</b>	<b>10/0/0</b>	

**Tablo 3.23:** Bileşik test fonksiyonları ( $F_{24} - F_{29}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$p$ -değeri	2.38E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	7.63E-06	0.01449	1.73E-06	1.73E-06	1.73E-06	4.67E-03	0.01178	1.73E-06
$F_{24}$ (T+, T-)	(15, 238)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 351)	(104.5, 330.5)	(0, 465)	(0, 465)	(0, 465)	(45, 231)	(49, 204)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	1.80E-05	0.02849	1.73E-06	1.73E-06	1.02E-05	1.48E-03	2.77E-03	6.98E-06	2.88E-06	9.27E-03	0.07190	0.36953	3.06E-04
$F_{25}$ (T+, T-)	(24, 441)	(126, 339)	(0, 465)	(0, 465)	(18, 447)	(78, 387)	(87, 378)	(14, 451)	(5, 460)	(106, 359)	(145, 320)	(176, 259)	(57, 408)
KD	+	+	+	+	+	+	+	+	+	+	=	=	+
$p$ -değeri	0.53044	8.19E-05	1.36E-05	1.73E-06	4.68E-03	1.36E-05	0.01044	1.73E-06	1.73E-06	3.85E-03	0.12544	8.94E-04	0.17791
$F_{26}$ (T+, T-)	(202, 263)	(424, 41)	(444, 21)	(0, 465)	(95, 370)	(21, 444)	(108, 357)	(0, 465)	(0, 465)	(92, 373)	(307, 158)	(394, 71)	(167, 298)
KD	=	-	-	+	-	+	+	+	+	+	=	-	=
$p$ -değeri	5.29E-04	2.35E-06	1.36E-05	1.73E-06	1.04E-03	1.73E-06	6.84E-03	1.73E-06	1.73E-06	3.72E-05	2.41E-04	0.17138	1.92E-06
$F_{27}$ (T+, T-)	(64, 401)	(3, 462)	(21, 444)	(0, 465)	(73, 392)	(0, 465)	(101, 364)	(0, 465)	(0, 465)	(32, 433)	(54, 411)	(166, 299)	(1, 464)
KD	+	+	+	+	+	+	+	+	+	+	+	=	+
$p$ -değeri	8.47E-06	1.73E-06	1.92E-06	1.73E-06	4.29E-06	1.73E-06	0.03501	1.73E-06	1.73E-06	2.84E-05	2.05E-04	7.73E-04	1.73E-06
$F_{28}$ (T+, T-)	(16, 449)	(0, 465)	(1, 464)	(0, 465)	(9, 456)	(0, 465)	(120, 315)	(0, 465)	(0, 465)	(29, 436)	(52, 413)	(62, 373)	(0, 465)
KD	+	+	+	+	+	+	+	+	+	+	+	+	+
$p$ -değeri	2.60E-05	2.58E-03	3.41E-05	2.35E-06	2.13E-06	4.45E-05	0.01790	6.339E-06	0.79710	0.12044	0.06511	4.27E-03	1.92E-06
$F_{29}$ (T+, T-)	(28, 437)	(86, 379)	(31, 434)	(3, 462)	(2, 463)	(34, 431)	(108, 327)	(13, 452)	(220, 245)	(157, 308)	(122, 284)	(50, 250)	(1, 464)
KD	+	+	+	+	+	+	+	+	=	=	=	+	=
<b>Toplam (+/=-)</b>	<b>5/1/0</b>	<b>5/0/1</b>	<b>5/0/1</b>	<b>6/0/0</b>	<b>5/0/1</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>6/0/0</b>	<b>5/1/0</b>	<b>5/1/0</b>	<b>3/3/0</b>	<b>3/2/1</b>	<b>4/2/0</b>

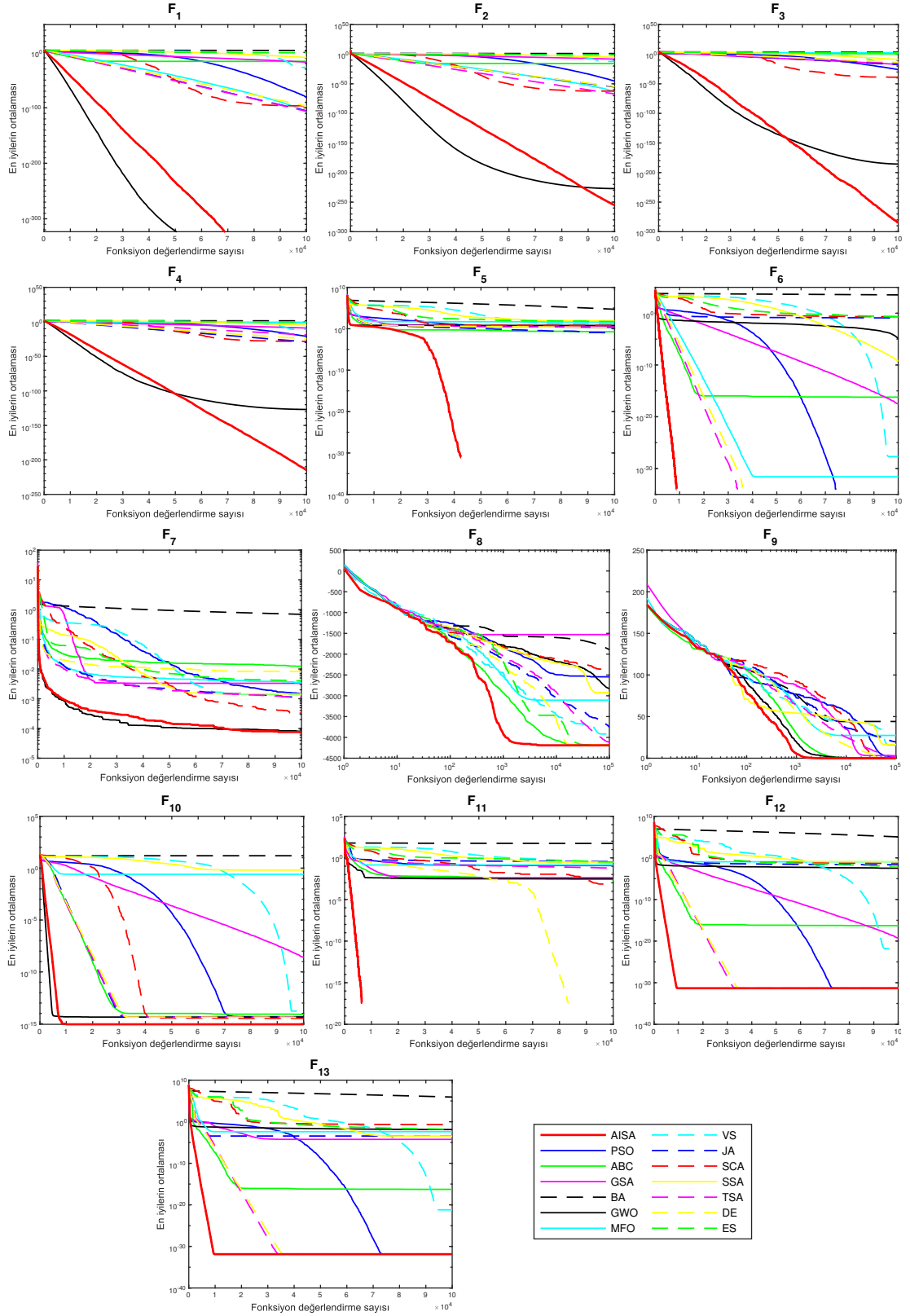
**Tablo 3.24:** CEC 2019 test fonksiyonları ( $F_{30} - F_{39}$ ) için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

$F$	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES	
$F_{30}$	$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	
	KD	+	+	+	+	+	+	+	+	+	+	+	+	
$F_{31}$	$p$ -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	
	(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	
	KD	+	+	+	+	+	+	+	+	+	+	+	+	
$F_{32}$	$p$ -değeri	0.53648	6.64E-04	0.02840	1.73E-06	2.77E-03	2.35E-06	9.09E-04	1.73E-06	1.73E-06	7.69E-06	1.73E-06	1.73E-06	
	(T+, T-)	(128.5, 171.5)	(67, 398)	(66, 210)	(0, 465)	(87, 378)	(3, 462)	(45, 306)	(0, 465)	(0, 465)	(15, 450)	(0, 465)	(0, 465)	
	KD	=	+	+	+	+	+	+	+	+	+	+	+	
$F_{33}$	$p$ -değeri	1.73E-06	4.68E-03	1.73E-06	1.73E-06	0.05193	1.92E-06	0.03160	1.92E-06	1.73E-06	1.60E-04	3.61E-03	7.27E-03	6.04E-03
	(T+, T-)	(0, 465)	(370, 95)	(0, 465)	(0, 465)	(138, 327)	(1, 464)	(128, 337)	(1, 464)	(0, 465)	(49, 416)	(91, 374)	(363, 102)	(99, 366)
	KD	+	-	+	+	=	+	+	+	+	+	+	-	+
$F_{34}$	$p$ -değeri	1.48E-04	2.35E-06	1.73E-06	1.73E-06	1.73E-06	2.37E-05	3.88E-06	1.73E-06	1.73E-06	3.18E-06	4.53E-04	0.06564	1.73E-06
	(T+, T-)	(48, 417)	(462, 3)	(465, 0)	(0, 465)	(0, 465)	(27, 438)	(8, 457)	(0, 465)	(0, 465)	(6, 459)	(62, 403)	(322, 143)	(0, 465)
	KD	+	-	-	+	+	+	+	+	+	+	+	=	+
$F_{35}$	$p$ -değeri	8.47E-06	7.51E-05	3.32E-04	1.73E-06	0.01852	1.73E-06	4.11E-03	1.73E-06	1.73E-06	7.51E-05	9.71E-05	0.65833	9.32E-06
	(T+, T-)	(16, 449)	(40, 425)	(58, 407)	(0, 465)	(118, 347)	(0, 465)	(372, 93)	(0, 465)	(0, 465)	(40, 425)	(422, 43)	(211, 254)	(17, 448)
	KD	+	+	+	+	+	+	-	+	+	+	-	=	+
$F_{36}$	$p$ -değeri	8.47E-06	1.73E-06	1.73E-06	1.73E-06	1.38E-03	1.73E-06	0.01319	1.73E-06	1.73E-06	6.89E-05	2.60E-06	0.03501	0.07865
	(T+, T-)	(16, 449)	(465, 0)	(0, 465)	(0, 465)	(77, 388)	(0, 465)	(112, 353)	(0, 465)	(0, 465)	(39, 426)	(4, 461)	(335, 130)	(318, 147)
	KD	+	-	+	+	+	+	+	+	+	+	+	-	=
$F_{37}$	$p$ -değeri	4.73E-06	0.03160	1.73E-06	1.73E-06	1.06E-04	1.73E-06	7.27E-03	1.73E-06	1.92E-06	1.49E-05	1.73E-06	0.16503	6.89E-05
	(T+, T-)	(10, 455)	(128, 337)	(0, 465)	(0, 465)	(44, 421)	(0, 465)	(102, 363)	(0, 465)	(1, 464)	(22, 443)	(0, 465)	(165, 300)	(39, 426)
	KD	+	+	+	+	+	+	+	+	+	+	+	=	+
$F_{38}$	$p$ -değeri	2.13E-06	2.88E-06	2.13E-06	1.73E-06	2.88E-06	1.73E-06	3.59E-04	1.73E-06	1.73E-06	2.35E-06	1.73E-06	7.69E-06	1.73E-06
	(T+, T-)	(2, 463)	(5, 460)	(463, 2)	(0, 465)	(5, 460)	(0, 465)	(59, 406)	(0, 465)	(0, 465)	(3, 462)	(0, 465)	(15, 450)	(0, 465)
	KD	+	+	-	+	+	+	+	+	+	+	+	+	+
$F_{39}$	$p$ -değeri	3.61E-03	0.03327	0.24519	0.01566	7.69E-06	4.20E-04	0.14139	1.73E-06	1.92E-06	0.03001	1.73E-06	1.83E-03	0.01108
	(T+, T-)	(91, 374)	(129, 336)	(176, 289)	(115, 350)	(15, 450)	(61, 404)	(161, 304)	(0, 465)	(1, 464)	(127, 338)	(0, 465)	(81, 384)	(109, 356)
	KD	+	+	=	+	+	+	=	+	+	+	+	+	+
<b>Toplam (+/=-)</b>	<b>9/1/0</b>	<b>7/0/3</b>	<b>7/1/2</b>	<b>10/0/0</b>	<b>9/1/0</b>	<b>10/0/0</b>	<b>8/1/1</b>	<b>10/0/0</b>	<b>10/0/0</b>	<b>10/0/0</b>	<b>9/0/1</b>	<b>5/3/2</b>	<b>9/1/0</b>	

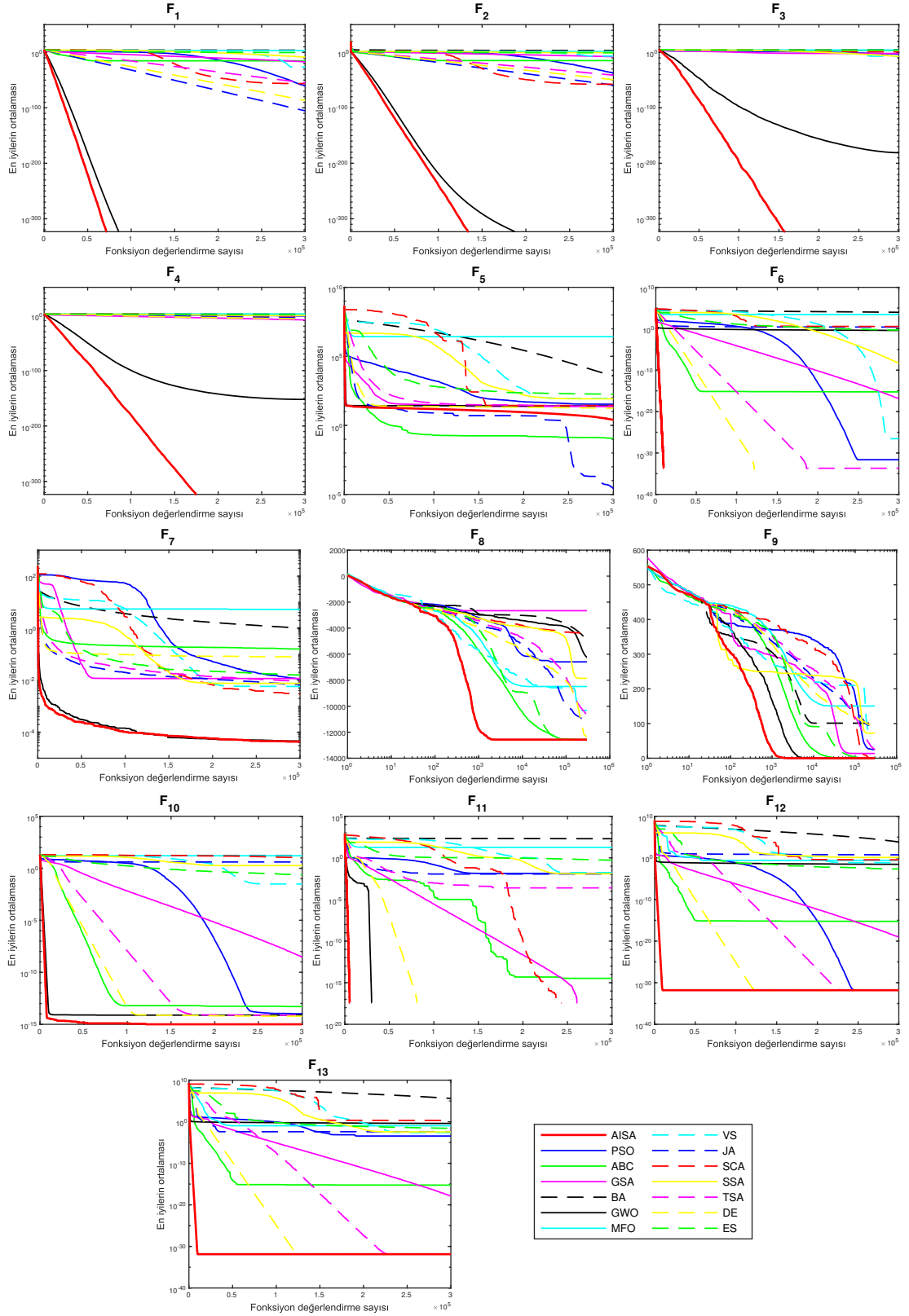
**Tablo 3.25:** Test kategorilerine ait Wilcoxon işaretli sıra testi sonuçları ( $\alpha = 0.05$ ).

Kategori	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
Tek-modlu ( $n = 10$ )	6/1/0	7/0/0	7/0/0	7/0/0	5/2/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0	6/1/0	6/1/0	7/0/0
Tek-modlu ( $n = 30$ )	7/0/0	6/0/1	7/0/0	7/0/0	4/3/0	7/0/0	7/0/0	6/0/1	7/0/0	7/0/0	6/1/0	6/1/0	7/0/0
Tek-modlu ( $n = 50$ )	7/0/0	6/0/1	6/0/1	6/1/0	5/2/0	7/0/0	7/0/0	6/0/1	7/0/0	7/0/0	7/0/0	5/1/1	7/0/0
Çok-modlu ( $n = 10$ )	4/2/0	4/2/0	6/0/0	6/0/0	5/1/0	6/0/0	6/0/0	6/0/0	4/2/0	6/0/0	4/2/0	1/5/0	6/0/0
Çok-modlu ( $n = 30$ )	6/0/0	6/0/0	5/1/0	6/0/0	4/2/0	6/0/0	6/0/0	6/0/0	4/2/0	6/0/0	3/3/0	3/3/0	6/0/0
Çok-modlu ( $n = 50$ )	6/0/0	6/0/0	5/1/0	6/0/0	4/2/0	6/0/0	6/0/0	6/0/0	5/1/0	6/0/0	5/1/0	3/3/0	6/0/0
Sabit boyutlu çok-modlu	3/6/1	3/6/1	4/5/1	10/0/0	10/0/0	6/3/1	4/5/1	8/2/0	10/0/0	10/0/0	3/5/2	1/7/2	10/0/0
Bileşik	5/1/0	5/0/1	5/0/1	6/0/0	5/0/1	6/0/0	6/0/0	6/0/0	5/1/0	5/1/0	3/3/0	3/2/1	4/2/0
CEC 2019	9/1/0	7/0/3	7/1/2	10/0/0	9/1/0	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	9/0/1	5/3/2	9/1/0
<b>Toplam (+/=/-)</b>	<b>53/11/1</b>	<b>50/8/7</b>	<b>52/8/5</b>	<b>64/1/0</b>	<b>61/13/1</b>	<b>61/3/1</b>	<b>57/6/2</b>	<b>61/2/2</b>	<b>59/6/0</b>	<b>64/1/0</b>	<b>46/16/3</b>	<b>33/26/6</b>	<b>62/3/0</b>

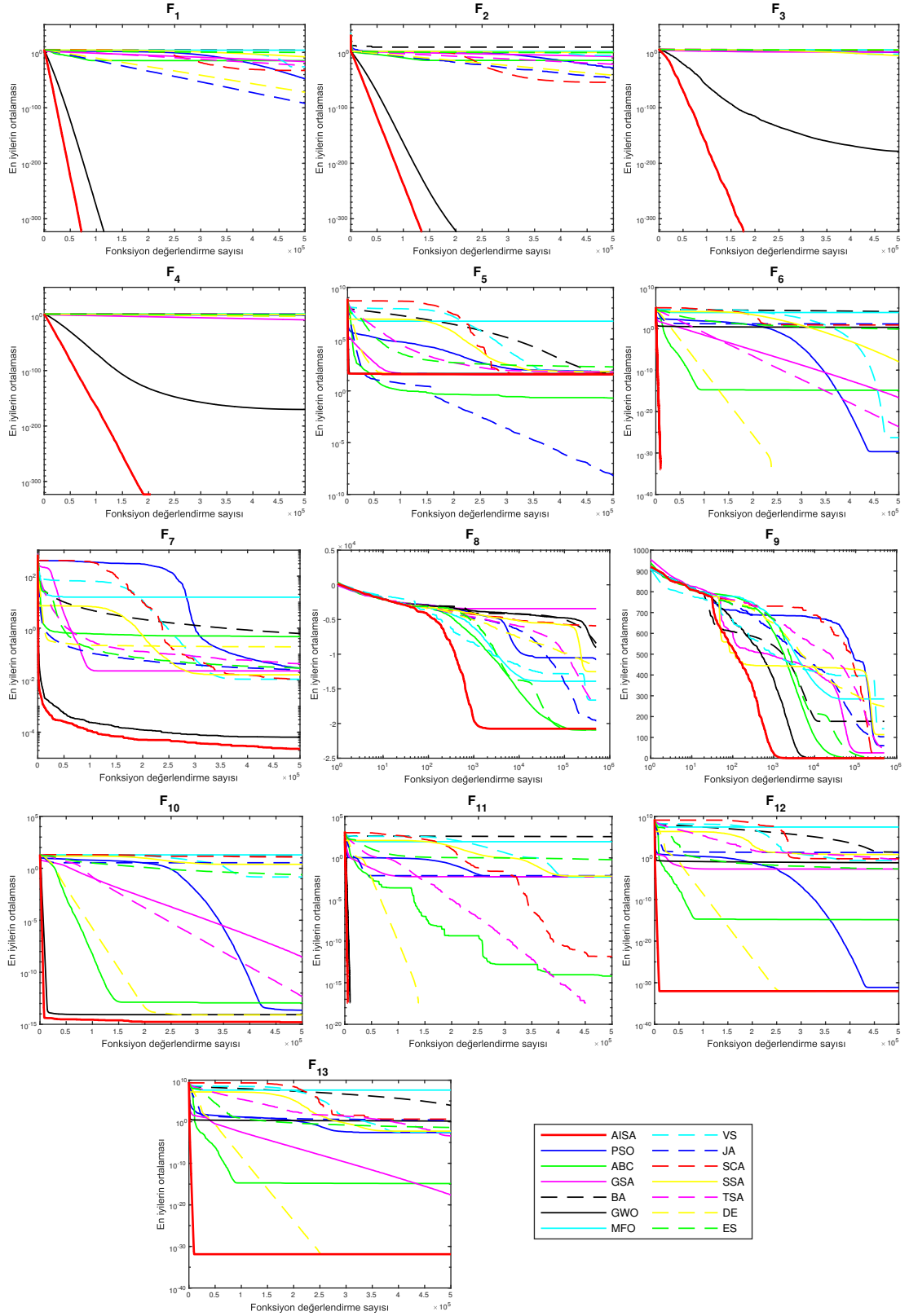




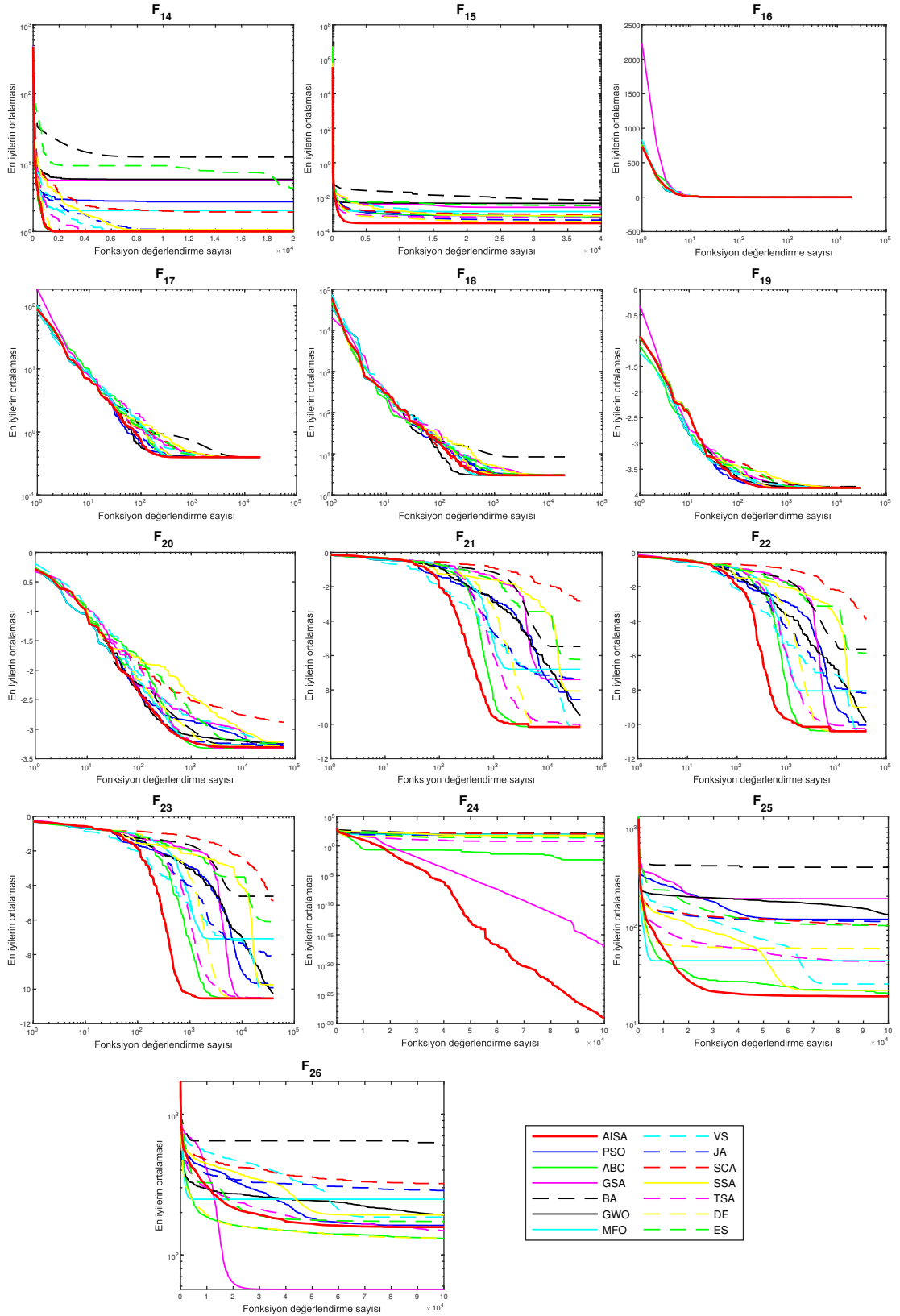
**Şekil 3.6:** 10 boyutlu  $F_1 - F_{13}$  fonksiyonları üzerinde algoritmaların yakınsama eğrileri.



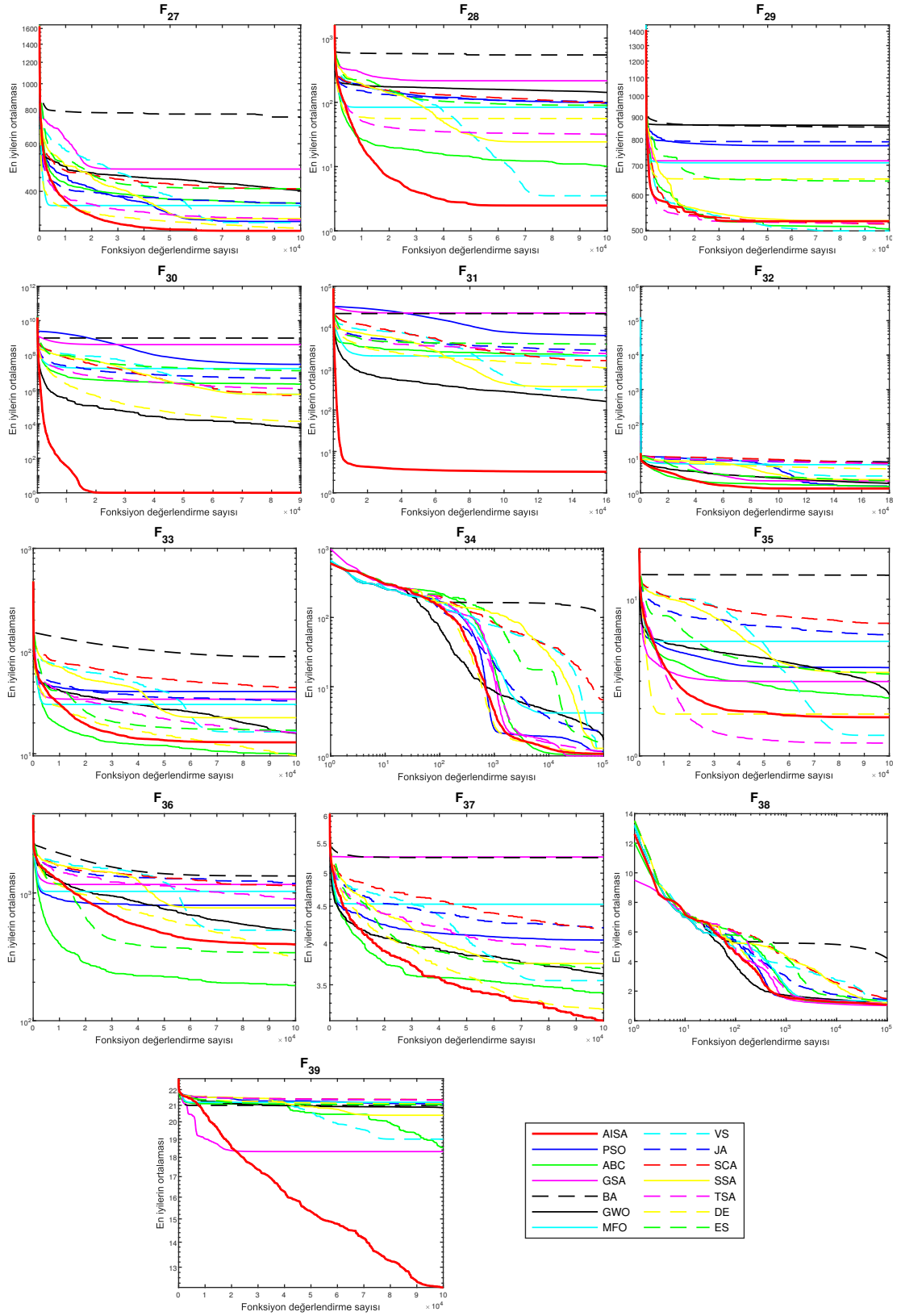
Şekil 3.7: 30 boyutlu  $F_1 - F_{13}$  fonksiyonları üzerinde algoritmaların yakınsama eğrileri.



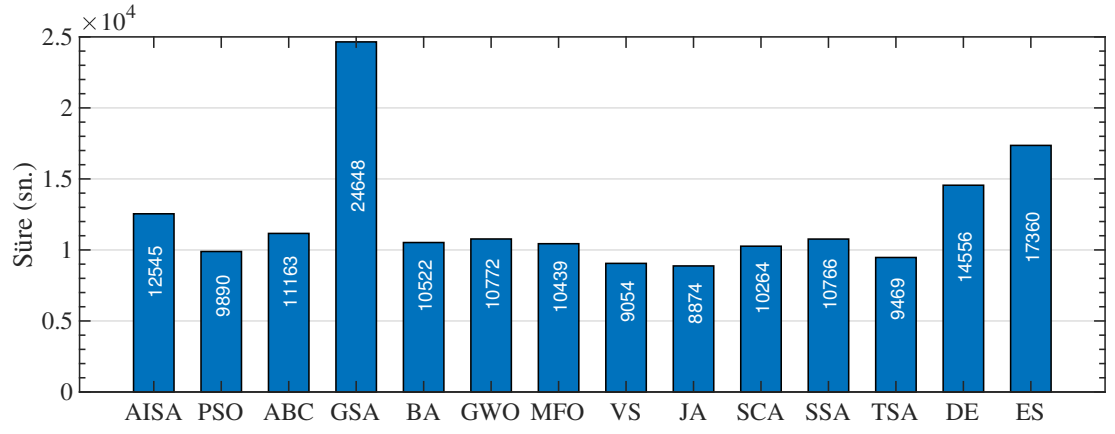
Şekil 3.8: 50 boyutlu  $F_1 - F_{13}$  fonksiyonları üzerinde algoritmaların yakınsama eğrileri.



Şekil 3.9:  $F_{14} - F_{26}$  fonksiyonları üzerinde algoritmaların yakınsama eğrileri.



Şekil 3.10:  $F_{27} - F_{39}$  fonksiyonları üzerinde algoritmaların yakınsama eğrileri.



**Şekil 3.11:** Algoritmaların toplam çalışma süreleri.

## 4. MÜHENDİSLİK PROBLEMLERİ ÜZERİNDE YAPILAN DENEYSEL ÇALIŞMALAR

Bu bölüme kadar, önerilen AISA algoritmasının performansı standart test problemleri üzerinde farklı performans ölçütlerine göre kapsamlı olarak incelenmiş ve başarısı gösterilmiştir. Buna ek olarak önerilen yaklaşımın performansını gerçek zamanlı uygulamalar için de test etmek amacıyla, IIR filtre tabanlı sistem tanımlama problemi, yedi serbestlik dereceli (7-DOF) seri robot manipülatörün ters kinematik problemi ve görüntü kontrast iyileştirme problemi ele alınmış ve bu problemler çözülmüştür.

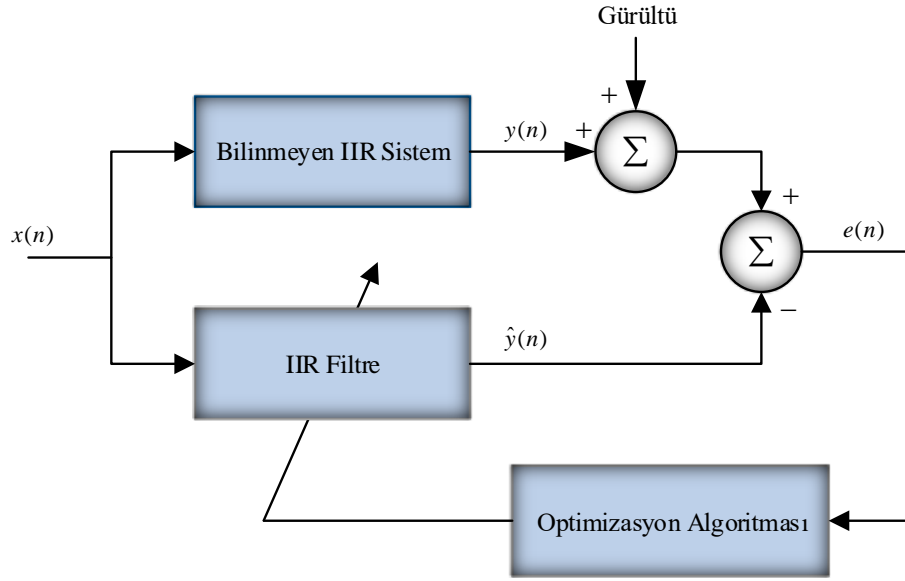
### 4.1 AISA ile IIR Sistem Tanılama Probleminin Çözümü

Sayısal filtre türlerinden biri olan sonsuz dürtü cevaplı (Infinite Impulse Response - IIR) filtre; sinyal işleme, kontrol, haberleşme ve pek çok mühendislik alanında sıklıkla kullanılmaktadır. IIR filtresi özyinelemeli bir yapıya sahip olduğundan, mevcut çıkışı sadece mevcut girişe bağlı değil, aynı zamanda önceki giriş ve çıkışlara da bağlıdır. Genel olarak IIR sistem tanılamada ana görev, aynı girdinin hem IIR filtreye hem de bilinmeyen sisteme eş zamanlı uygulanarak filtre çıkışı bilinmeyen sistemin çıkışıyla eşleşene kadar bir optimizasyon algoritması yardımıyla filtrenin parametrelerini belirlemektir. Tanılanacak sistemin hata yüzeyi genellikle doğrusal olmayan, türevlenemez ve çok modludur (Lagos-Eulogio ve diğ. 2017). IIR sistem tanılama ve filtre tasarımı problemlerinde bu zorlukların üstesinden gelmek için birçok metasezgisel algoritma sıklıkla kullanılmıştır (Karaboga 2009; Gotmare ve diğ. 2017; Kalinli ve Karaboga 2005; Dai ve diğ. 2009). Şekil 4.1’de IIR filtresi kullanan bir sistem tanılama yapılandırması gösterilmektedir.

IIR filtresinin giriş-çıkış ilişkisi aşağıdaki fark denklemi şeklinde yazılabilir,

$$y(n) + \sum_{i=1}^M a_i y(n-i) = \sum_{j=0}^N b_j x(n-j) \quad (4.1)$$

burada  $a_i$  ( $i = 1, 2, \dots, M$ ) ve  $b_j$  ( $j = 0, 1, 2, \dots, N$ ) filtre katsayıları olup,  $x(n)$  ve  $y(n)$



**Şekil 4.1:** IIR filtre sistem tanılama konfigürasyonu.

sırasıyla sistemin  $n$  anındaki giriş ve çıkış sinyallerini ifade etmektedir.  $M (\geq N)$  ise filtrenin derecesidir. IIR filtrenin transfer fonksiyonu (4.2) eşitliğinde verilmiştir.

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + \dots + a_Mz^{-M}} \quad (4.2)$$

IIR sistem tanınması, bilinmeyen sistem davranışını taklit eden eşdeğer bir modeli tahmin etmek için kullanılır. Tasarım sürecinde giriş örnekleri önce her iki sistemde kullanılmak üzere üretilir ve daha sonra hem filtreye hem de bilinmeyen sisteme senkronize olarak uygulanır. Son olarak, IIR filtresinin katsayıları her iki sistemden gelen çıktılarla yakından eşleşecek şekilde bir optimizasyon algoritması yardımıyla kestirilir. Diğer bir deyişle, arzulanan ve tahmin edilen çıktılar arasındaki hatayla ilişkili amaç fonksiyonu minimize edilir. Bu çalışmada, ortalama karesel hata (mean square error - MSE) amaç fonksiyonu,

$$MSE = \frac{1}{N_s} \sum_{n=1}^{N_s} e(n)^2 \quad (4.3)$$

biçiminde ele alınmış olup  $N_s$  giriş-çıkış çifti sayısı,  $e(n) = y(n) - \hat{y}(n)$  o anki tanılama hata sinyalidir. Buradaki  $\hat{y}(n)$  ise filtrenin o anki tahmin edilen cevabını işaret



etmektedir. Bunlara ek olarak, MSE pratik uygulamalarda desibel (dB) cinsinden (4.4) eşitliği ile ifade edilebilir.

$$MSE(\text{dB}) = 10 \log \left( \frac{1}{N_s} \sum_{n=1}^{N_s} e(n)^2 \right) \quad (4.4)$$

Tez kapsamında yapılan sistem tanılama benzetim çalışmasında, önerilen ASIA algoritmasının arama yeteneğini araştırmak için iki IIR filtre uygulaması kullanılmıştır. 256 örnekleme ( $N_s = 256$ ) sahip giriş/test işareti; sıfır ortalamalı, birim varyanslı ve uniform dağılımlı bir beyaz gürültüdür. AISA ve diğer rakip algoritmaları adil bir şekilde karşılaştırmak için maksimum fonksiyon değerlendirme sayısı Bölüm 3'te kullanıldığı gibi ayarlanmış olup, algoritmaların kontrol parametreleri ise Tablo 3.6'da verildiği şekilde kullanılmıştır.

### Uygulama 1

Bu uygulamada bilinmeyen sistem olarak Panda ve diğ. (2011) tarafından kullanılan dördüncü dereceden bir sistem ele alınmıştır ve bu sistemin transfer fonksiyonu (4.5) eşitliğinde verilmiştir.

$$H_s(z) = \frac{1 - 0.9z^{-1} + 0.81z^{-2} - 0.729z^{-3}}{1 + 0.04z^{-1} + 0.2775z^{-2} - 0.2101z^{-3} + 0.14z^{-4}} \quad (4.5)$$

*Durum 1:* Bu durumda (4.5) eşitliğinde verilen sistem aynı dereceden IIR filtre kullanılarak modellenmiştir. Bu IIR filtre modelinin transfer fonksiyonu ( $H_f(z)$ ) aşağıdaki eşitlikte verilmiştir.

$$H_f(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}{1 - a_1z^{-1} - a_2z^{-2} - a_3z^{-3} - a_4z^{-4}} \quad (4.6)$$

*Durum 2:* Bu durumda (4.5) eşitliğinde verilen sistem üçüncü dereceden IIR filtre ile modellenmiştir. Bu sistemi modelleyecek olan IIR filtrenin transfer fonksiyonu ( $H_f(z)$ ) aşağıda verilmiştir.

$$H_f(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 - a_1z^{-1} - a_2z^{-2} - a_3z^{-3}} \quad (4.7)$$

## Uygulama 2

Bu uygulamada bilinmeyen sistem olarak Panda ve diğ. (2011) çalışmasındaki beşinci dereceden bir sistem ele alınmıştır ve bu sistemin transfer fonksiyonu (4.8) eşitliğinde verilmiştir.

$$H_s(z) = \frac{0.1084 + 0.5419z^{-1} + 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}}{1 + 0.9853z^{-1} + 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}} \quad (4.8)$$

*Durum 1:* Bu durumda (4.8) eşitliğinde verilen sistem aynı dereceden IIR filtre kullanılarak modellenmiştir. Bu filtre modelinin transfer fonksiyonu ( $H_f(z)$ ) aşağıdaki eşitlikte verilmiştir.

$$H_f(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5}}{1 - a_1z^{-1} - a_2z^{-2} - a_3z^{-3} - a_4z^{-4} - a_5z^{-5}} \quad (4.9)$$

*Durum 2:* Bu durumda (4.8) eşitliğinde verilen sistem dördüncü dereceden IIR filtre ile modellenmiştir. Bu sistemi modelleyecek olan IIR filtrenin transfer fonksiyonu ( $H_f(z)$ ) aşağıda verilmiştir.

$$H_f(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}{1 - a_1z^{-1} - a_2z^{-2} - a_3z^{-3} - a_4z^{-4}} \quad (4.10)$$

Yukarıda verilen ardışık uygulamalar ve onların durumlarının herbiri için tüm algoritmalar 30 kez çalıştırılmış, MSE ve MSE (dB) cinsinden testlerden elde edilen en iyi, ortalama ve standart sapma sonuçları sırasıyla Tablolar 4.1 - 4.4'te rapor edilmiştir. Tablolardan ortalama sonuçlar açısından AISA ve JA algoritmalarının Uygulama 1 - Durum 2'de en iyi sonuçları elde ettiği, diğer uygulama ve durumlarda ise AISA'nın diğer algoritmalarından daha iyi performans gösterdiği açıkça görülmektedir.

Başka bir karşılaştırma kriteri olarak 30 bağımsız koşumdan elde edilen sonuçlara dayalı parametrik olmayan Wilcoxon işaretli sıra testi uygulanmıştır. Bu teste ilişkin sonuçlar Tablo 4.5'te verilmiştir. Uygulama 1 - Durum 2'de, JA algoritmasına karşılık gelen  $p$ -değeri 0.05'ten büyüktür. Bu durum için AISA ve JA arasında anlamlı bir fark yoktur. Diğer durumlarda  $p$ -değerleri 0.05'ten küçüktür. Bu,

önerilen yaklaşımın önemli derecede üstünlüğünü ortaya koyan  $H_0$  hipotezine karşı güçlü bir kanıttır.

IIR filtre tabanlı sistem tanımlama problemlerine ilişkin algoritmaların yakınsama eğrileri Şekil 4.2’de gösterilmiştir. Ölçeklendirme sorunlarını önlemek için ilk 500 fonksiyon değerlendirmesinin sonuçlarının gösterilmediğine dikkat edilmelidir. Şekil 4.2’de görüldüğü gibi, AISA’nın yakınsaması diğer rakip algoritmalarından çok daha hızlıdır. Tüm sonuçlar değerlendirildiğinde, önerilen algoritmanın IIR sistem tanımlama problemlerinde dikkate değer bir başarıya sahip olduğu sonucuna varılabilir.

**Tablo 4.1:** Uygulama 1 - Durum 1 için elde edilen MSE and MSE (dB) değerleri.

MSE	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En iyi	<b>3.58E-30</b>	1.39E-18	1.60E-05	4.18E-19	2.32E-07	8.60E-07	2.39E-19	1.66E-18	2.10E-03	1.90E-02	3.78E-14	9.05E-15	7.93E-21	1.93E-03
En iyi (dB)	<b>-294.4604</b>	-178.5740	-47.9577	-183.7881	-66.3454	-60.6553	-186.2170	-177.7988	-26.7786	-17.2119	-134.2239	-140.4340	-201.0078	-27.1339
Ort.	<b>6.14E-27</b>	8.36E-07	3.01E-04	1.24E-18	4.10E-01	2.63E-02	2.52E-09	2.10E-03	6.64E-03	6.67E-02	3.39E-03	8.16E-09	8.36E-19	1.59E-02
Ort. (dB)	<b>-262.1154</b>	-60.7771	-35.2213	-179.0833	-3.8713	-15.8069	-85.9780	-26.7690	-21.7760	-11.7557	-24.7029	-80.8846	-180.7754	-17.9776
Std.	1.49E-26	3.29E-06	3.14E-04	6.07E-19	1.35E+00	5.12E-02	1.08E-08	1.15E-02	5.08E-03	3.94E-02	1.31E-02	3.40E-08	2.74E-18	1.39E-02
Std. (dB)	-258.2687	-54.8233	-35.0346	-182.1663	1.3157	-12.9069	-79.6833	-19.3834	-22.9413	-14.0486	-18.8233	-74.6853	-175.6244	-18.5708

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 4.2:** Uygulama 1 - Durum 2 için elde edilen MSE and MSE (dB) değerleri.

MSE	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En iyi	0.07186	0.07186	0.07224	0.07186	0.07186	0.07500	0.07186	0.07186	<b>0.07186</b>	0.08347	0.07186	0.07186	0.07186	0.07613
En iyi (dB)	-11.4353	-11.4353	-11.4120	-11.4353	-11.4353	-11.2493	-11.4353	-11.4353	<b>-11.4353</b>	-10.7849	-11.4353	-11.4353	-11.4353	-11.1844
Ort.	<b>0.07186</b>	0.09419	0.07875	0.13471	0.58313	0.13005	0.11183	0.07558	<b>0.07186</b>	0.14374	0.07930	0.07189	0.07186	0.17529
Ort. (dB)	<b>-11.4353</b>	-10.2602	-11.0374	-8.7059	-2.3423	-8.8590	-9.5145	-11.2161	<b>-11.4353</b>	-8.4242	-11.0073	-11.4331	-11.4353	-7.5625
Std.	3.47E-17	4.54E-02	5.00E-03	5.30E-02	1.19E+00	4.79E-02	1.79E-01	2.04E-02	3.98E-17	4.64E-02	2.83E-02	7.60E-05	4.09E-16	7.43E-02
Std. (dB)	-164.6004	-13.4279	-23.0087	-12.7601	0.7470	-13.1962	-7.4737	-16.9077	-163.9968	-13.3383	-15.4787	-41.1898	-153.8830	-11.2886

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 4.3:** Uygulama 2 - Durum 1 için elde edilen MSE and MSE (dB) değerleri.

MSE	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En iyi	<b>1.34E-06</b>	1.99E-06	8.54E-05	1.87E-05	2.32E-06	9.71E-05	7.70E-06	1.39E-06	2.02E-06	4.70E-03	1.71E-06	5.01E-05	3.63E-06	2.02E-04
En iyi (dB)	<b>-58.7154</b>	-57.0168	-40.6843	-47.2913	-56.3362	-40.1260	-51.1338	-58.5660	-56.9497	-23.2754	-57.6609	-42.9980	-54.4048	-36.9459
Ort.	<b>1.34E-06</b>	1.49E-03	3.98E-04	8.28E-05	8.59E-01	5.31E-03	5.16E-01	1.85E-04	5.27E-04	1.37E-02	1.93E-04	2.67E-04	7.16E-05	1.62E-02
Ort. (dB)	<b>-58.7154</b>	-28.2778	-34.0056	-40.8183	-0.6605	-22.7476	-2.8765	-37.3266	-32.7852	-18.6172	-37.1471	-35.7430	-41.4507	-17.9131
Std.	1.36E-20	5.26E-03	2.21E-04	3.16E-05	1.68E+00	4.65E-03	1.20E+00	8.70E-05	8.56E-04	5.26E-03	1.34E-04	1.35E-04	4.08E-05	3.29E-02
Std. (dB)	-198.6741	-22.7901	-36.5503	-44.9966	2.2489	-23.3273	0.8044	-40.6040	-30.6752	-22.7914	-38.7222	-38.7010	-43.8889	-14.8291

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

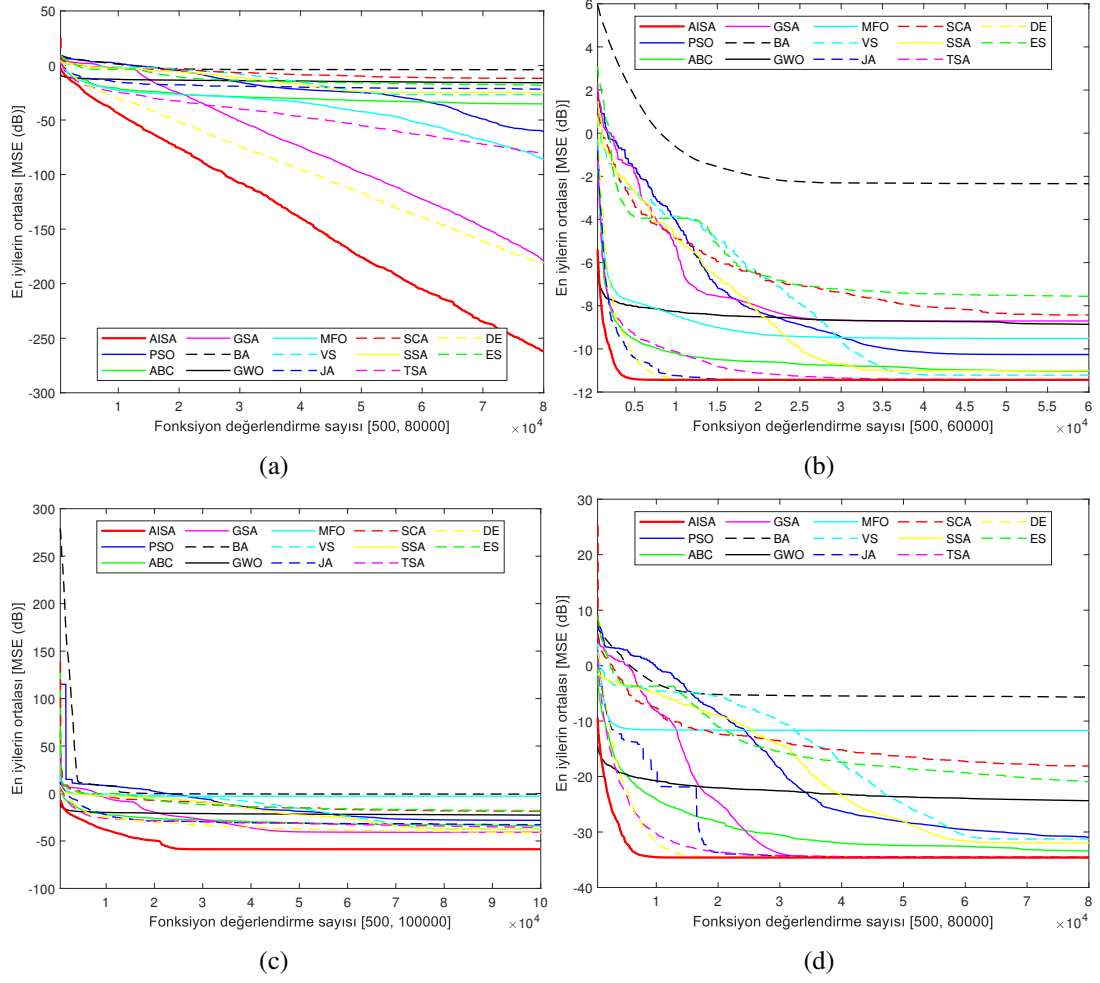
**Tablo 4.4:** Uygulama 2 - Durum 2 için elde edilen MSE and MSE (dB) değerleri.

MSE	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En iyi	<b>3.47E-04</b>	3.47E-04	3.55E-04	3.47E-04	3.47E-04	3.55E-04	3.47E-04	3.47E-04	3.47E-04	5.78E-03	3.47E-04	3.47E-04	3.47E-04	8.58E-04
En iyi (dB)	<b>-34.6023</b>	-34.6023	-34.5020	-34.6023	-34.5996	-34.5008	-34.6023	-34.6022	-34.5939	-22.3826	-34.6023	-34.6023	-34.6023	-30.6641
Ort.	<b>3.47E-04</b>	8.00E-04	4.56E-04	3.47E-04	2.71E-01	3.67E-03	6.76E-02	7.49E-04	3.50E-04	1.55E-02	6.32E-04	3.48E-04	3.47E-04	8.03E-03
Ort. (dB)	<b>-34.6023</b>	-30.9703	-33.4089	-34.6023	-5.6706	-24.3541	-11.7009	-31.2563	-34.5607	-18.0975	-31.9913	-34.5886	-34.6023	-20.9525
Std.	1.65E-19	1.98E-03	1.24E-04	4.92E-19	6.16E-01	4.48E-03	3.68E-01	1.40E-03	1.68E-06	9.09E-03	9.60E-04	2.48E-06	2.28E-15	7.27E-03
Std. (dB)	-187.8305	-27.0252	-39.0501	-183.0801	-2.1030	-23.4854	-4.3376	-28.5332	-57.7371	-20.4153	-30.1779	-56.0515	-146.4270	-21.3818

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 4.5:** IIR sistem tanılama problemleri için Wilcoxon işaretli sıra testi sonuçları.

		PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES	
Uygulama 1	Durum 1	<i>p</i> -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
		(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
		KD	+	+	+	+	+	+	+	+	+	+	+	+	+
	Durum 2	<i>p</i> -değeri	1.73E-06	1.73E-06	1.37E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1	1.73E-06	1.73E-06	1.73E-06	4.18E-07	1.73E-06
		(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(9, 6)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
		KD	+	+	+	+	+	+	+	=	+	+	+	+	+
Uygulama 2	Durum 1	<i>p</i> -değeri	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
		(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
		KD	+	+	+	+	+	+	+	+	+	+	+	+	+
	Durum 2	<i>p</i> -değeri	1.73E-06	1.73E-06	1.01E-07	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
		(T+, T-)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
		KD	+	+	+	+	+	+	+	+	+	+	+	+	+
<b>Toplam (+/=-)</b>		<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>3/1/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	<b>4/0/0</b>	



**Şekil 4.2:** IIR sistem tanılama problemleri için algoritmaların yakınsama eğrileri (a) Uygulama 1 - Durum 1; (b) Uygulama 1 - Durum 2; (c) Uygulama 2 - Durum 1; (d) Uygulama 2 - Durum 2

## 4.2 AISA ile 7-DOF Robot Manipülatorün Ters Kinematik Probleminin Çözümü

Yedi serbestlik dereceli (7-DOF) seri robot manipülatorler engelden kolayca kaçabilmeleri, esnek hareket edebilmeleri ve geniş bir alanda çalışabilmelerinden dolayı endüstride yaygın olarak kullanılmaktadır.

Robot kinematiği, eklem açıları ve bu eklemler arasındaki mesafe ile uç elemanın çalışma uzayındaki konumu arasındaki ilişkinin analitik olarak incelenmesidir (Dereli 2019). Robot kinematiği ileri/düz (forward) ve ters (inverse) kinematik olmak üzere ikiye ayrılır. İleri kinematik, eklem açılarına göre robotun

uç elemanının kartezyen uzayda konumunu bulma işlemi; ters kinematik ise istenen uç elemanın konumuna göre eklem parametrelerini bulma olarak tanımlanabilir. Ters kinematik problemleri, karmaşık doğrusal olmayan yapıları nedeniyle zorlu optimizasyon problemleri olarak düşünülebilir (Toz 2020) ve bu problemleri çözmek için üstün arama kabiliyetine sahip metasezgisel algoritmalar robotik çalışmalarda optimizasyon aracı olarak kullanılmaktadır (Akyıldız ve Çetinkaya 2012; Rokbani ve diğ. 2015; Dereli ve Köker 2020<sup>b</sup>).

Bir seri endüstriyel robotun eklem değişkenlerini belirlemek için Denavit–Hartenberg (DH) yöntemi yaygın olarak kullanılır (Karaçalı 2012). Bu çalışmada, Dereli ve Köker (2020<sup>a</sup>) tarafından Tablo 4.6’da verilen DH parametreleri kullanılmıştır. Burada,  $a_i$ ,  $\alpha_i$ ,  $d_i$  ve  $\theta_i$  sırasıyla iki eksen arasındaki bağ uzunluğunu (link length),  $(i - 1)$  ile  $i$  eksenleri arasındaki bağ açısını (link twist), üst üste çakışan bağlar arasındaki bağ kaçıklığını (link offset) ve iki bağ arasında oluşan eklem açısını (joint angle) ifade etmektedir.

**Tablo 4.6:** Yedi eklemli robotun DH parametreleri.

$i$	$a_i(m)$	$\alpha_i(^{\circ})$	$d_i(m)$	$\theta_i(^{\circ})$ (Aralık)
1	0	-90	$l_1 = 0.5$	$-180 < \theta_1 < 180$
2	$l_2 = 0.2$	90	0	$-90 < \theta_2 < 30$
3	$l_3 = 0.25$	-90	0	$-90 < \theta_3 < 120$
4	$l_4 = 0.3$	90	0	$-90 < \theta_4 < 90$
5	$l_5 = 0.2$	-90	0	$-90 < \theta_5 < 90$
6	$l_6 = 0.2$	0	0	$-90 < \theta_6 < 90$
7	$l_7 = 0.1$	0	$d_7 = 0.05$	$-30 < \theta_7 < 90$

DH değişkenlerinin belirlenmesiyle genel dönüşüm matrisi

$${}_{i-1}^i T = \begin{bmatrix} c\theta_i & -c\alpha_i \cdot s\theta_i & s\alpha_i \cdot s\theta_i & a_i \cdot c\theta_i \\ s\theta_i & c\alpha_i \cdot c\theta_i & -c\theta_i \cdot s\alpha_i & a_i \cdot s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

kullanılarak her bir eklem dönüşüm matrisi belirlenir. Burada  $s$  ve  $c$  sırasıyla sinüs ve kosinüs fonksiyonlarını ifade etmektedir. Tablo 4.6’da verilen DH değişkenleri (4.11)

eşitliğinde yerlerine konulduğunda,

$$\begin{aligned}
{}^1_0T &= \begin{bmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & {}^2_1T &= \begin{bmatrix} c\theta_2 & 0 & s\theta_2 & l_2c\theta_2 \\ s\theta_2 & 0 & -c\theta_2 & l_2s\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^3_2T &= \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 & l_3c\theta_3 \\ s\theta_3 & 0 & c\theta_3 & l_3s\theta_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & {}^4_3T &= \begin{bmatrix} c\theta_4 & 0 & s\theta_4 & l_4c\theta_4 \\ s\theta_4 & 0 & -c\theta_4 & l_4s\theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^5_4T &= \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 & l_5c\theta_5 \\ s\theta_5 & 0 & c\theta_5 & l_5s\theta_5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & {}^6_5T &= \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & l_6c\theta_6 \\ s\theta_6 & c\theta_6 & 0 & l_6s\theta_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^7_6T &= \begin{bmatrix} c\theta_7 & -s\theta_7 & 0 & l_7c\theta_7 \\ s\theta_7 & c\theta_7 & 0 & l_7s\theta_7 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{4.12}$$

açık formda her bir eklemün dönüşüm matrisi elde edilir.

Sonrasında manipülatörün ileri yön kinematik matrisi, bireysel dönüşüm matrislerinin çarpılmasıyla,

$$T_{\text{End-Effector}} = {}^7_0T = {}^1_0T \cdot {}^2_1T \cdot {}^3_2T \cdot {}^4_3T \cdot {}^5_4T \cdot {}^6_5T \cdot {}^7_6T = \begin{bmatrix} R_{3 \times 3} & \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

şeklinde elde edilir. Burada  $R_{3 \times 3}$  oryantasyon matrisini ve  $[P_x \ P_y \ P_z]^T$  uç elemanın konum vektörünü gösterir.

7-DOF seri robot manipülatörü ters kinematik probleminin ana görevi, Kartezyen koordinat sisteminde uç elemanın istenen pozisyonlarına göre eklem açılarını  $(\theta_1, \theta_2, \dots, \theta_7)$  belirlemektir. İstenen ve tahmin edilen konum vektörleri arasındaki Öklid mesafesine göre hesaplanan konum hatası, en küçüklenmesi gereken amaç fonksiyonu olarak kabul edilir ve aşağıdaki gibi yazılabilir:

$$error = \|P_{\text{desired}} - P_{\text{predicted}}\| \tag{4.14}$$



burada  $\|\cdot\|$  Öklid mesafesini,  $P_{desired}$  robot manipulatörün istenen pozisyon vektörünü,  $P_{predicted}$  ise algoritmaların bir aday çözümünü ileri kinematik denklemlerinde yerine koyarak bulunan tahmini konum vektörünü temsil eder.

Benzetim bölümünde AISA'nın performansını doğrulamak için  $P_{desired1} = [-25 \ 100 \ 50]^T$  ve  $P_{desired2} = [50 \ -25 \ 75]^T$  olmak üzere iki farklı istenen pozisyon vektörü santimetre cinsinden seçilmiştir. Rakip algoritmalarla adil bir karşılaştırma yapabilmek için maksimum fonksiyon değerlendirme sayısı Bölüm 3'de kullanıldığı gibi ve algoritmaların kontrol parametreleri de Tablo 3.6'da verildiği şekilde kullanılmıştır.

Tablo 4.7 ve Tablo 4.8 uç elemanın iki farklı istenen konum koordinatı için ters kinematik problemleri üzerinde AISA ve diğer 13 rakip metasezgisel ile elde edilen optimizasyon sonuçlarını göstermektedir. Bu sonuçlara göre AISA ters kinematik problemlerinde ortalama performans sonuçları açısından diğer algoritmalarından daha iyi sonuç vermektedir. AISA'ya karşı diğer algoritmaların 30 koşum sonrasında Wilcoxon işaretli sıra testine ilişkin karşılaştırma sonuçları Tablo 4.9 ve Tablo 4.10'da verilmiştir. Önerilen AISA, birinci senaryoda MFO ve ikinci senaryoda VS dışında diğer algoritmalara göre daha iyi performans sergilemektedir. Şekil 4.3'te ise ters kinematik problemleri üzerinde algoritmaların yakınsama geçmişleri yer almakta olup, AISA'nın diğer rakip algoritmalara göre çok daha hızlı yakınsadığını görülmektedir. Ters kinematik problemlerine ilişkin genel optimizasyon sonuçları, bu mühendislik probleminde optimal değerleri elde eden AISA'nın gerçek mühendislik problemlerine uygulanabilir bir algoritma olduğunu ortaya koymaktadır.

**Tablo 4.7:** Algoritmaların  $P_{desired1} = [-25 \ 100 \ 50]^T$  (cm) için 30 koşulda elde ettikleri karşılaştırma sonuçları.

	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En İyi	<b>0.00E+00</b>	<b>0.00E+00</b>	1.45E-04	1.69E-11	1.11E-05	9.03E-06	<b>0.00E+00</b>	<b>0.00E+00</b>	5.37E-04	1.69E-03	1.58E-09	8.32E-07	3.26E-04	2.27E-03
Ort.	<b>0.00E+00</b>	6.40E-10	9.78E-04	7.33E-11	2.66E-05	9.75E-05	6.07E-04	4.84E-12	4.12E-03	7.62E-03	1.67E-08	1.20E-03	3.00E-03	3.10E-02
Std.	0.00E+00	3.27E-09	7.06E-04	3.38E-11	1.05E-05	8.50E-05	3.33E-03	1.94E-11	2.09E-03	3.93E-03	1.04E-08	1.78E-03	1.81E-03	1.93E-02

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 4.8:** Algoritmaların  $P_{desired2} = [50 \ -25 \ 75]^T$  (cm) için 30 koşulda elde ettikleri karşılaştırma sonuçları.

	AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
En İyi	<b>0.00E+00</b>	<b>0.00E+00</b>	8.70E-05	7.30E-12	9.53E-06	1.46E-05	<b>0.00E+00</b>	<b>0.00E+00</b>	4.40E-03	2.42E-03	2.41E-09	8.99E-07	4.20E-05	1.41E-03
Ort.	<b>0.00E+00</b>	2.05E-05	3.46E-04	2.81E-03	2.53E-05	1.78E-03	5.27E-04	5.55E-18	9.51E-03	1.11E-02	1.20E-08	7.24E-04	1.85E-03	1.63E-02
Std	0.00E+00	1.12E-04	1.94E-04	1.54E-02	9.80E-06	9.22E-03	2.89E-03	2.11E-17	5.94E-03	4.60E-03	6.04E-09	7.18E-04	2.31E-03	1.00E-02

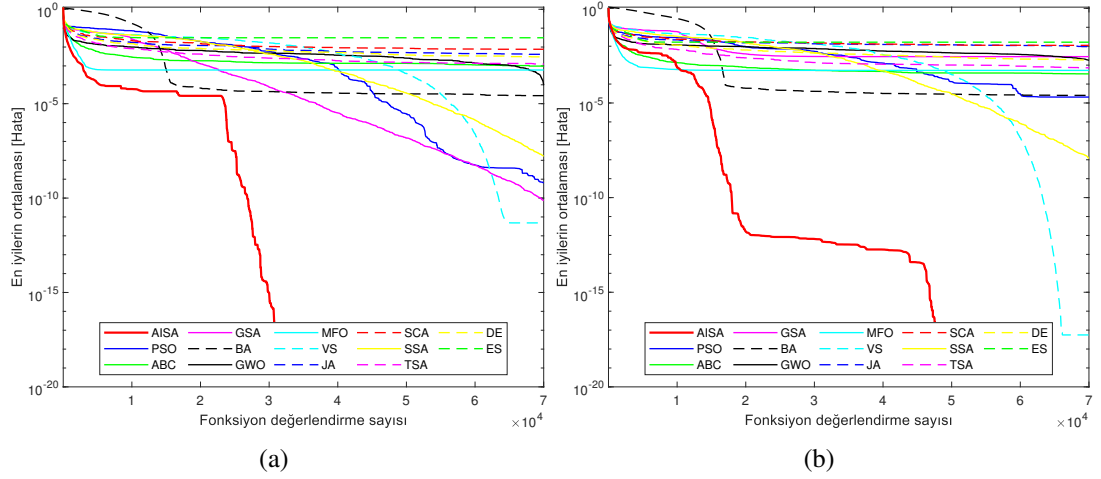
\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

**Tablo 4.9:**  $P_{desired1} = [-25 \ 100 \ 50]^T$  (cm) için Wilcoxon işaretli sıra testi sonuçları.

	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$p$ -değeri	4.88E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	0.0625	1.93E-04	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
(T+, T-)	(0, 78)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 15)	(0, 171)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	=	+	+	+	+	+	+	+

**Tablo 4.10:**  $P_{desired2} = [50 \ -25 \ 75]^T$  (cm) için Wilcoxon işaretli sıra testi sonuçları.

	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
$p$ -değeri	8.81E-05	1.73E-06	1.73E-06	1.73E-06	1.73E-06	7.81E-03	0.25	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06	1.73E-06
(T+, T-)	(0, 210)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 36)	(0, 6)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)	(0, 465)
KD	+	+	+	+	+	+	=	+	+	+	+	+	+



**Şekil 4.3:** Ters kinematik problemleri için algoritmaların yakınsama eğrileri (a)  $P_{desired1}$  ; (b)  $P_{desired2}$

### 4.3 AISA ile Görüntü İyileştirme Probleminin Çözümü

Görüntü işleme gelişmiş bir görüntü elde etmek veya ondan bazı yararlı bilgiler çıkarmak için bir görüntü üzerinde belirli işlemleri sayısal bilgisayarlarla gerçekleştirme olarak tanımlanabilir. Görüntü işleme tıptan ulaştırmaya ve hatta sanayiye kadar günlük yaşamımızda birçok alanda uygulamaya sahiptir.

Görüntü işleme tekniklerinin en önemlilerinden biri olan görüntü iyileştirme, görüntü üzerinde hem nitelik hem de nicelik açısından üstün bilgi içeriklerine sahip gelişmiş bir görüntü oluşturma işlemidir. Diğer bir ifadeyle, bir görüntünün kontrastı gibi fotometrik özelliklerini değiştirerek daha iyi görünmesini, yorumlanmasını ve herhangi bir görüntü işleme sisteminin sonraki adımlarında başarımı artırmayı amaçlayan bir tekniktir (Draa ve Bouaziz 2014). Görüntü iyileştirmeye yönelik süzgeçleme/filtreleme ve kontrast ile yapılan iyileştirme olmak üzere iki temel yaklaşım vardır. Süzgeçleme bulanıklaşma ve gürültü gibi bozucu etkilerin giderilmesi için kullanılır. Kontrasta dayalı olarak yapılan iyileştirmenin amacı ise görüntüdeki en açık ve en koyu renkler arasındaki fark olarak ifade edilen kontrastı artırmaktır.

Kontrasta dayalı olarak doğrusal kontrast germe ve histogram eşitleme/dengeleme olmak üzere iki temel iyileştirme yöntemi vardır. Doğrusal

kontrast germe, görüntüdeki gri düzeylerin dinamik aralığını artırmak için görüntünün gri seviyelerini yeniden eşleyen doğrusal bir dönüşüm kullanır. Histogram eşitleme ise görüntü histogramının kümülatif yoğunluk fonksiyonuna göre giriş gri seviyesi değerlerini eşitlemek için girdi görüntüsüne bir dönüşüm uygular. Çok basit ve kullanışlı olan bu yöntemler histogram tabanlı bütünsel iyileştirme yöntemleridir. Bu yöntemlerde görüntünün bütünsel olarak işlenmesi nedeniyle, görüntünün genel görünürlüğünü artırabilirler. Ancak bu yöntemler çoğu zaman küçük bölgeler üzerindeki ayrıntıların kaybolması ve gürültünün artması gibi dezavantajlara sahiptirler. Bu dezavantajları ortadan kaldırmak için hem bütünsel hem de bölgesel görüntünün bilgilerini kullanan bölgesel/bütünsel görüntü iyileştirme yöntemi kullanılmıştır (Munteanu ve Rosa 2004; Maurya ve diğ. 2017).

Genel olarak bölgesel/bütünsel görüntü iyileştirme yönteminde aşağıda verilen dönüşüm fonksiyonu kullanılarak gri ton bir görüntü iyileştirilmeye çalışılır. Bu dönüşüm fonksiyonu

$$O_{(i,j)} = k \frac{\mu^G}{\sigma_{(i,j)}^L + b} \left( I_{(i,j)} - c \times \mu_{(i,j)}^L \right) + (\mu_{(i,j)}^L)^a. \quad (4.15)$$

şeklinde ifade edilir. Burada

$I_{(i,j)}$  : her bir  $(i, j)$  piksel konumundaki giriş görüntüsü

$O_{(i,j)}$  : her bir  $(i, j)$  piksel konumundaki iyileşmiş/çıkış görüntüsü

$\mu^G$  : giriş görüntüsünü oluşturan piksellerin gri seviye değerlerinin ortalaması (bütünsel ortalama)

$\mu_{(i,j)}^L$  :  $(i, j)$  merkezli komşuluktaki piksellerin gri seviye değerlerinin ortalaması (bölgesel ortalama)

$\sigma_{(i,j)}^L$  :  $(i, j)$  merkezli komşuluktaki piksellerin gri seviye değerlerinin standart sapması

$a, b, c, k$  : bilinmeyen karar değişkenleri

olup,  $M \times N$  boyutlu bir görüntünün ortalaması,  $n \times n$  pencere/kalıp büyüklüğüne sahip

bölgenin ortalaması ve standart sapması aşağıdaki eşitlikler kullanılarak hesaplanabilir.

$$\mu^G = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N I_{(i,j)} \quad (4.16)$$

$$\mu_{(i,j)}^L = \frac{1}{n \times n} \sum_{i=1}^n \sum_{j=1}^n I_{(i,j)} \quad (4.17)$$

$$\sigma_{(i,j)}^L = \sqrt{\left( \frac{1}{n \times n} \sum_{i=1}^n \sum_{j=1}^n \left( I_{(i,j)} - \mu_{(i,j)}^L \right)^2 \right)}. \quad (4.18)$$

Bölgesel/bütünsel görüntü iyileştirme yönteminde kullanılan bu dönüşüm fonksiyonu  $a, b, c, k$  olmak üzere dört bilinmeyen değişken içermektedir. Görüntü kontrastı iyileştirme süreci belli bir amaç fonksiyonuna göre giriş görüntüsünün görsel kontrastını artırarak istenmeyen yapaylıkların en aza indirildiği doğrusal olmayan bir optimizasyon problemi olarak ele alındığından, bu parametreler metasezgisel optimizasyon algoritmaları ile kestirilerek birçok farklı alandaki görüntülere uygulanmışlardır (Mahapatra ve diğ. 2015; Suresh ve diğ. 2017; Kamoona ve Patra 2019; Yue ve Zhang 2020; Suresh ve Lal 2017).

Bilinmeyen  $a, b, c$  ve  $k$  parametrelerini bulabilmek için maksimize edilecek olan amaç fonksiyonu,

$$F(O) = \ln \left( \ln(E(O^s) + e) \right) \cdot \frac{n_{edge}(O^s)}{M \times N} \cdot e^{H(O)} \quad (4.19)$$

şeklinde ifade edilebilir. Burada  $e$  Euler sabitini,  $H(O)$  iyileştirilen görüntünün entropisini,  $n_{edge}(O^s)$  iyileştirilen görüntüye Sobel kenar bulma algoritması uygulanması sonucu elde edilen kenar sayısını ve  $E(O^s)$  iyileştirilen görüntüye Sobel kenar işlemcisi uygulanması sonucu elde edilen tüm piksel yoğunluğu değerlerinin toplamını belirtmektedir. İyileştirilen görüntünün entropisi

$$H(O) = \begin{cases} - \sum_{i=0}^{255} p(i) \log_2(p(i)), & \text{eğer } p_i \neq 0 \\ 0, & \text{eğer } p_i = 0 \end{cases} \quad (4.20)$$

şeklinde hesaplanır ve burada  $p_i$  görüntünün  $i$ . yoğunluk değerinin oluşma olasılığıdır.

Ek olarak Sobel kenar işlemcisi uygulanması sonucu elde edilen  $O^s$  görüntüsü

$$O_{(i,j)}^s = \sqrt{(\nabla_x O_{(i,j)})^2 + (\nabla_y O_{(i,j)})^2} \quad (4.21)$$

$$\nabla_x O_{(i,j)} = (g_{(i-1,j+1)} + 2g_{(i,j+1)} + g_{(i+1,j+1)}) - (g_{(i-1,j-1)} + 2g_{(i,j-1)} + g_{(i+1,j-1)}) \quad (4.22)$$

$$\nabla_y O_{(i,j)} = (g_{(i+1,j+1)} + 2g_{(i+1,j)} + g_{(i+1,j-1)}) - (g_{(i-1,j+1)} + 2g_{(i-1,j)} + g_{(i-1,j-1)}) \quad (4.23)$$

eşitlikleri ile elde edilir. Burada  $g_{(i,j)}$  iyileştirilen görüntünün  $(i, j)$  koordinatındaki gri seviye yoğunluk değerini belirtmektedir.  $\nabla_x O_{(i,j)}$  ve  $\nabla_y O_{(i,j)}$  ise sırasıyla yatay ve dikey türevleri ifade etmektedir.  $O^s$  görüntüsünün tüm piksel yoğunluğu değerlerinin toplamı aşağıdaki eşitlikte verilmiştir.

$$E(O^s) = \sum_{i=1}^M \sum_{j=1}^N O_{(i,j)}^s. \quad (4.24)$$

AISA algoritmasının performansını test etmek amacıyla; insan, doğa ve biyomedikal gibi farklı alanlardaki görüntülere AISA tabanlı bölgesel/bütünsel kontrast iyileştirme uygulanmıştır. Bu 12 görüntü Şekil 4.4'te verilmiştir. Tüm algoritmaların popülasyon sayısı veya başlangıç çözüm sayısı 10 ve fonksiyon değerlendirme sayısı 500 olacak şekilde ayarlanmıştır. Algoritmalara özgü diğer parametreler ise Tablo 3.6'da verildiği gibidir. Kestirilmesi gereken parametrelerin aralığı Kamoon ve Patra (2019) tarafından önerildiği gibi  $a \in (0, 1.5)$ ,  $b \in (\mu^G/2, \mu^G)$ ,  $c \in (0.5, 1.5)$  ve  $k \in (0.5, 1.5)$  olarak kullanılmıştır.

Karşılaştırmada kullanılan tüm algoritmaların 30 koşumu sonucunda elde ettikleri en iyi, ortalama ve standart sapma gibi temel istatistiksel sonuçları Tablo 4.11'de listelenmiştir. Bu tablodan görüldüğü üzere AISA, on iki görüntünün on tanesinde diğer algoritmalara kıyasla daha iyi performans sergilemiştir. Diğer iki görüntü olan *cameraman* ve *mri* görüntülerinde sırasıyla üçüncü ve ikinci başarı sırasına sahiptir. Wilcoxon işaretli sıra testine bağlı karşılaştırma sonuçları ise Tablo 4.12'de sunulmuş olup, sonuçlar AISA'nın üstün optimizasyon yeteneğine sahip

olduğunu kanıtlamaktadır. Diğer bir kıyaslamada ise algoritmaların bu görüntülerde yakınsama davranışları incelenmiş ve sonuçlar Şekil 4.5'te verilmiştir. Burada da AISA'nın yüksek yakınsama hızına sahip olduğu görülmektedir. Şekil 4.6'da ise kontrastı düşük olan orijinal görüntüler, bu görüntülerin histogramları, AISA ile iyileştirilmiş görüntüler ve histogramları verilmiştir. Açıkça görülmektedir ki; AISA temelli iyileştirilen görüntüler orijinal görüntülere göre daha yüksek kontrasta sahiptirler.



**Şekil 4.4:** Bölgesel/bütünsel kontrast iyileştirme için kullanılan görüntüler.

**Tablo 4.11: Bölgesel/bütünsel kontrast iyileştirme uygulanan görüntüler üzerinde algoritmaların elde ettiği sonuçlar.**

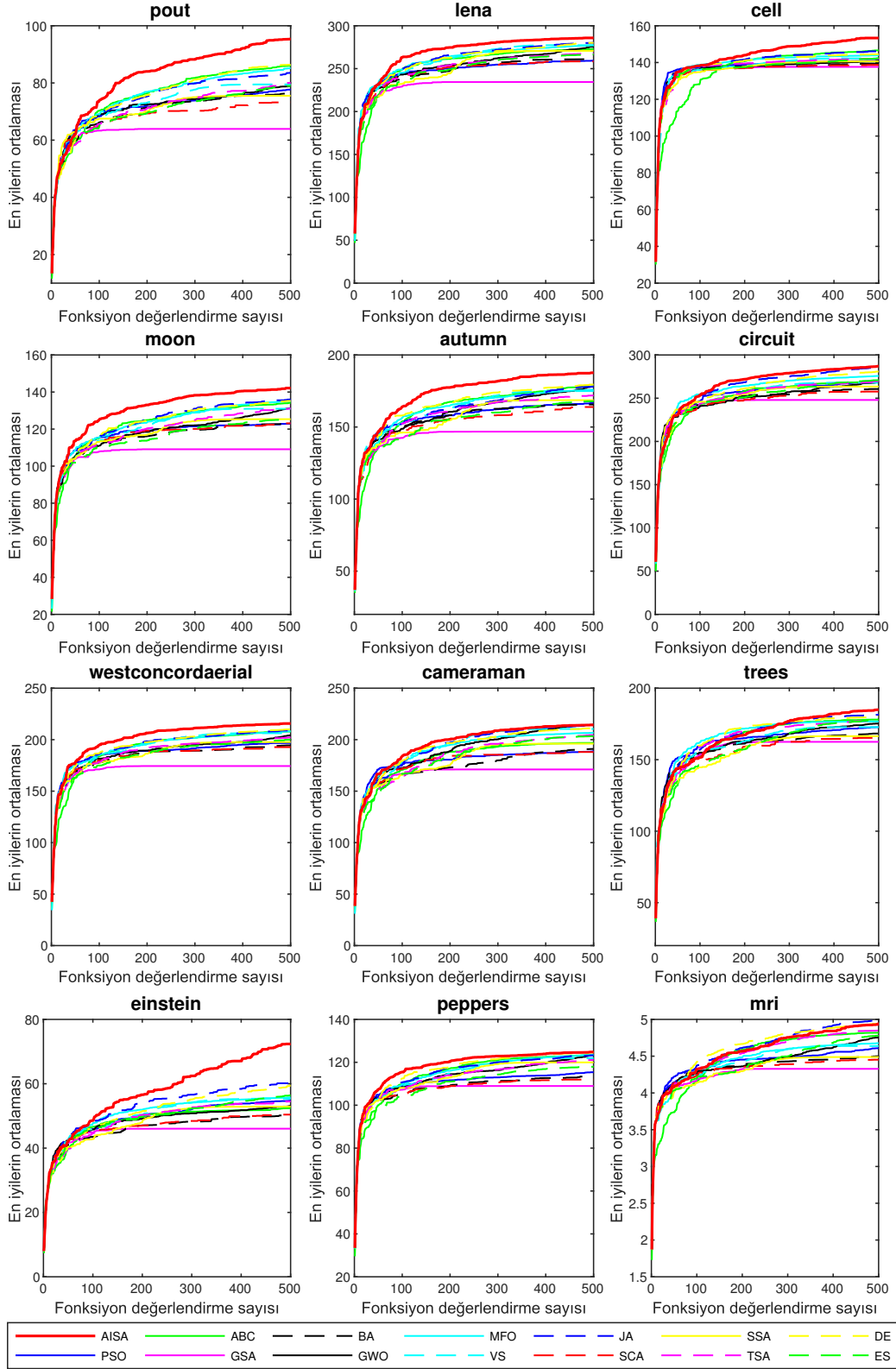
Görüntü		AISA	PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
pout	En iyi	<b>104.2920</b>	91.5824	101.9447	72.6436	89.9216	92.0811	98.4610	99.0389	99.9173	86.7730	87.3853	98.4790	100.5374	86.2286
	Ort.	<b>95.3548</b>	77.6700	86.0347	63.9259	76.3276	78.9299	85.1892	79.4804	83.5810	73.2412	75.5260	80.0291	86.3903	78.9482
	Std.	7.4644	4.4436	5.8171	4.1890	6.4158	5.1335	4.7407	7.4635	5.2966	5.4183	7.2628	4.9836	6.6159	4.0264
moon	En iyi	<b>150.0774</b>	134.3500	147.2315	125.5198	137.9286	139.8192	148.5574	142.3736	142.8567	135.1090	138.0004	144.7879	145.5529	135.1100
	Ort.	<b>142.1359</b>	123.0714	134.1692	109.1113	122.7743	131.3230	135.2276	130.9757	135.9792	123.1054	125.3470	131.1747	135.2031	125.2257
	Std.	4.8049	6.7883	6.7826	8.8742	8.9877	4.3724	5.6825	8.9185	4.7635	6.5756	8.5428	5.9016	6.3108	9.0961
westconcordaerial	En iyi	<b>226.5170</b>	210.5812	224.8097	194.4533	213.2162	213.4268	222.1491	216.4543	221.3796	211.4148	214.8539	208.9356	222.3915	215.6831
	Ort.	<b>215.6872</b>	196.7009	207.9308	174.3858	194.4303	204.1615	208.1125	206.2522	209.0389	192.7491	197.6774	202.0559	210.1063	200.0578
	Std.	6.5813	8.1926	8.7303	13.7181	10.0567	3.6275	6.8583	5.4976	5.7217	8.4503	9.9152	4.8049	5.2812	8.2721
einstein	En iyi	<b>79.7346</b>	61.6842	67.9408	62.7299	61.9908	60.6531	73.3648	75.2276	73.3696	61.5144	66.1514	66.7674	74.9703	76.9380
	Ort.	<b>72.3925</b>	54.7561	52.4021	46.0332	50.6037	53.0929	55.7566	55.3511	60.1502	50.4121	52.8976	54.4768	59.4040	56.3691
	Std.	7.0441	2.4582	4.2773	5.6306	4.1887	3.1209	5.4938	6.5461	6.0636	3.2989	4.5602	3.9580	7.1858	6.7178
lena	En iyi	<b>290.5617</b>	278.4870	284.1377	281.9046	281.0028	286.1477	289.1079	290.0145	287.2707	277.5964	287.2976	285.2521	287.6661	284.5636
	Ort.	<b>285.8641</b>	259.2456	272.8838	234.4272	261.0592	275.6532	277.2034	277.3567	280.0313	258.4233	270.7481	268.0361	280.1793	266.5271
	Std.	3.0558	13.6066	14.6464	18.9856	15.8421	10.6861	8.3479	17.9278	5.3415	9.5715	11.4962	8.6860	5.7682	13.6512
autumn	En iyi	<b>194.5710</b>	186.7078	190.8171	160.7265	187.5488	186.5469	189.0780	189.1086	186.7348	186.5246	186.6082	185.0218	190.1896	186.4105
	Ort.	<b>187.6448</b>	166.6471	178.2778	146.7980	165.6488	178.1695	175.8384	174.9356	178.0428	163.8937	168.6414	171.8638	179.4342	167.8682
	Std.	4.2007	8.5136	9.3878	10.7292	12.7290	5.4393	9.5289	12.7216	5.4529	10.1346	10.8219	7.8767	7.6939	11.4114
cameraman	En iyi	220.9048	216.3957	219.4715	201.4601	218.8378	218.9412	219.5302	<b>220.9338</b>	220.8470	216.0009	217.9594	218.6122	220.1751	218.1416
	Ort.	214.3214	188.8334	196.8344	171.0805	190.9915	214.3951	206.8029	210.3919	<b>214.4540</b>	188.0258	196.0913	204.3868	211.0836	203.3629
	Std.	6.6397	20.2273	15.4100	16.2218	18.0116	3.8105	14.0804	16.4400	6.3437	12.8608	17.7239	8.4001	7.6765	17.6688
peppers	En iyi	126.9675	123.3340	126.1310	119.4985	120.7811	126.4224	<b>127.1761</b>	126.8314	126.1736	118.9705	125.8962	124.3138	127.0826	125.5927
	Ort.	<b>124.8055</b>	115.3712	123.4890	108.9485	113.3931	123.5179	123.6100	121.9532	123.3400	111.9631	119.9574	121.0833	124.5615	117.9209
	Std.	1.9069	5.5997	2.2536	5.1327	5.7067	1.5249	2.7004	5.5862	2.1827	3.8448	4.3578	2.5417	2.8826	5.4452
cell	En iyi	<b>158.6663</b>	145.7942	152.6843	141.0806	144.1819	143.7241	158.1316	158.1283	156.5940	142.1347	150.7563	147.9582	157.1474	150.3981
	Ort.	<b>153.3267</b>	140.9156	146.6119	137.6342	139.4330	140.9571	143.9486	142.2430	146.1953	138.6616	140.6135	142.0666	144.5365	142.3209
	Std.	4.0945	1.9189	4.0613	1.5877	2.1014	1.5974	4.5528	4.4456	4.3127	1.6364	2.8133	2.5063	3.8794	2.5278
circuit	En iyi	<b>294.9119</b>	285.8726	290.9187	272.5074	284.2344	289.2413	287.8656	290.8951	292.7530	273.7566	287.0006	284.3502	294.4911	291.8051
	Ort.	<b>287.0621</b>	270.2700	268.6187	247.9610	260.6770	268.3719	275.7124	268.0289	285.4551	257.6560	263.4991	269.6885	280.8199	270.8087
	Std.	6.6599	7.8460	13.2738	9.2861	13.7939	12.5884	8.6644	13.0012	5.4097	10.7784	14.4965	8.9913	10.6770	11.0760
trees	En iyi	<b>188.3158</b>	184.9080	188.1858	181.0691	187.6241	185.3546	187.2771	187.4596	187.8079	184.7561	184.8724	185.8973	187.2952	186.7602
	Ort.	<b>184.9236</b>	172.7124	178.1161	162.4759	168.3072	175.2622	176.7447	172.7728	181.6995	165.2423	166.4514	177.2310	179.9124	177.7716
	Std.	3.8677	7.2901	11.0921	7.2213	10.8863	7.5208	8.6531	9.2206	4.5294	7.9771	10.2653	6.2472	6.9522	8.9228
mri	En iyi	5.1106	4.9027	<b>5.1145</b>	4.6791	4.9669	4.9856	5.0774	5.0448	5.0874	4.7688	4.9822	5.0283	5.1047	5.0020
	Ort.	4.9423	4.6084	4.8192	4.3277	4.4985	4.7557	4.6776	4.6380	<b>4.9856</b>	4.4519	4.4886	4.8483	4.9409	4.7795
	Std.	0.1304	0.1471	0.2574	0.1314	0.2295	0.2440	0.2866	0.3033	0.0706	0.1332	0.2658	0.1020	0.1295	0.1955

\* En iyi sonuçlar kalın yazı tipi ile vurgulanmıştır.

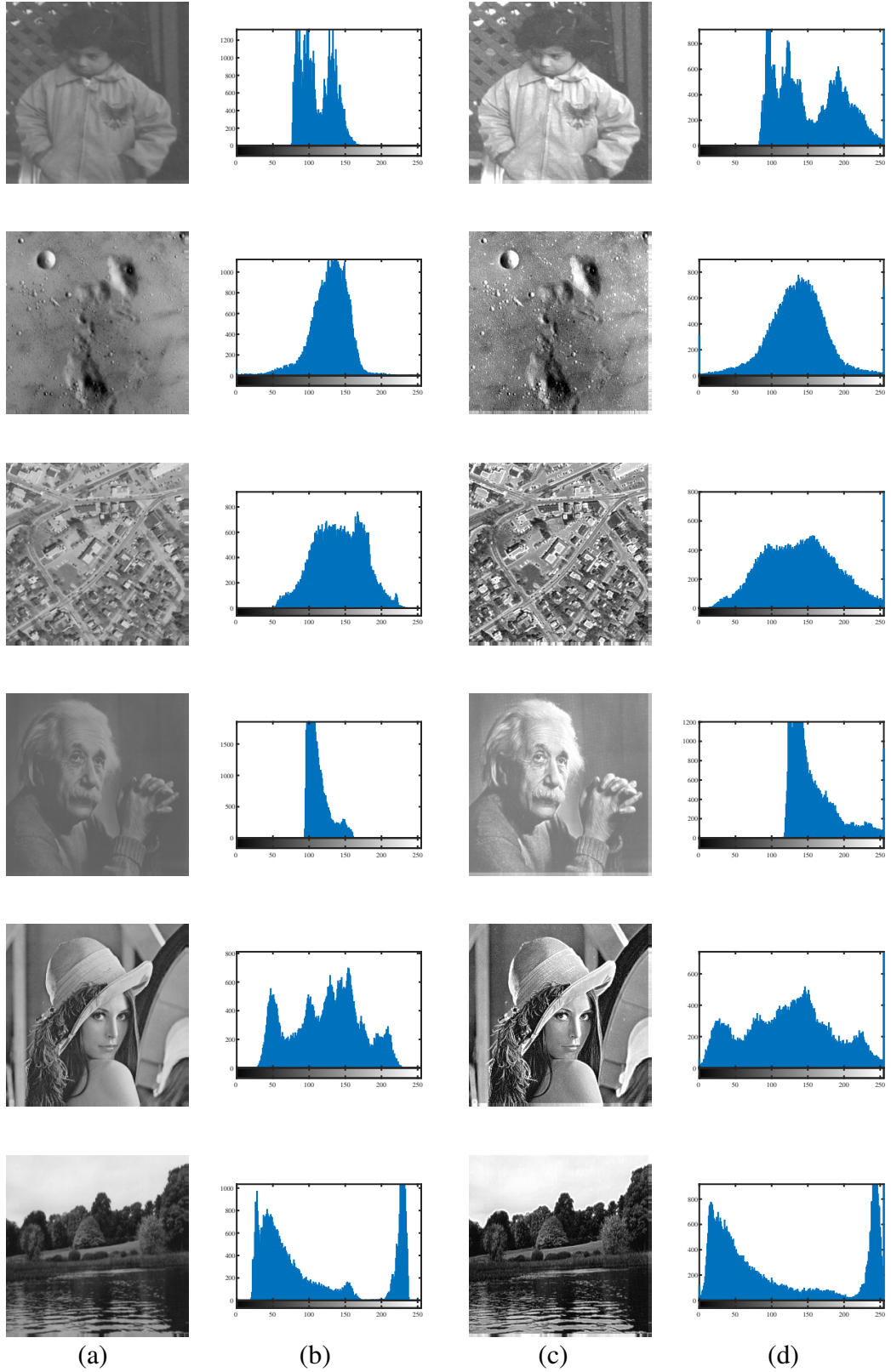


**Tablo 4.12:** Bölgesel/bütünsel kontrast iyileştirme uygulanan görüntüler için Wilcoxon işaretli sıra testinin sonuçları ( $\alpha = 0.05$ ).

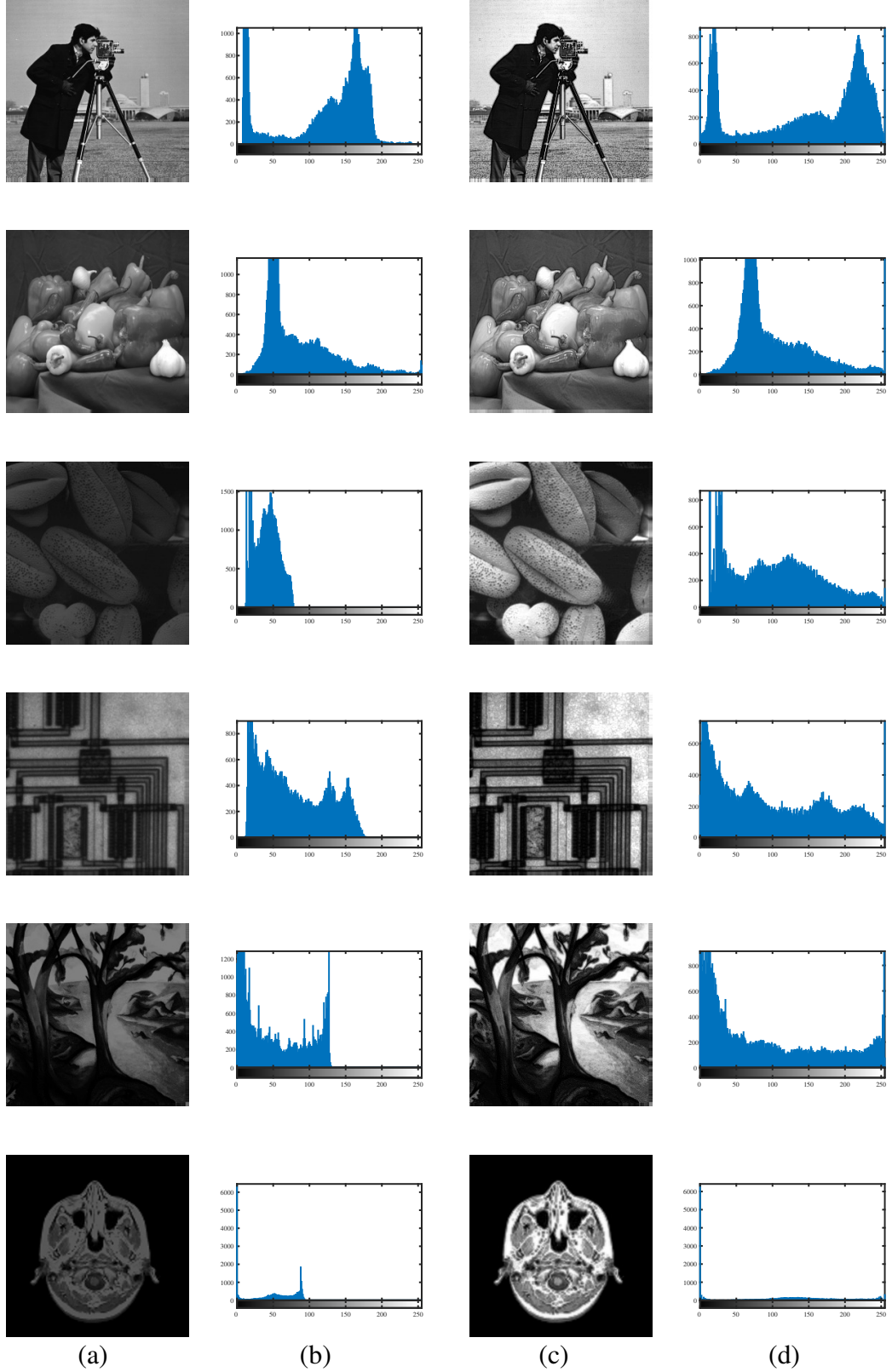
Görüntü		PSO	ABC	GSA	BA	GWO	MFO	VS	JA	SCA	SSA	TSA	DE	ES
pout	<i>p</i> -değeri	3.52E-06	3.06E-04	1.73E-06	1.73E-06	1.73E-06	2.60E-05	1.02E-05	5.22E-06	1.73E-06	1.73E-06	4.29E-06	3.32E-04	4.73E-06
	(T+, T-)	(7, 458)	(57, 408)	(0, 465)	(0, 465)	(0, 465)	(28, 437)	(18, 447)	(11, 454)	(0, 465)	(0, 465)	(9, 456)	(58, 407)	(10, 455)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
moon	<i>p</i> -değeri	1.73E-06	6.89E-05	1.73E-06	2.13E-06	2.60E-06	1.15E-04	6.98E-06	2.37E-05	1.73E-06	1.73E-06	8.47E-06	6.89E-05	1.73E-06
	(T+, T-)	(0, 465)	(39, 426)	(0, 465)	(2, 463)	(4, 461)	(45, 420)	(14, 451)	(27, 438)	(0, 465)	(0, 465)	(16, 449)	(39, 426)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
westconcordaerial	<i>p</i> -değeri	1.92E-06	1.20E-03	1.73E-06	1.92E-06	6.98E-06	1.74E-04	9.32E-06	4.20E-04	1.73E-06	3.52E-06	1.73E-06	5.32E-03	2.88E-06
	(T+, T-)	(1, 464)	(75, 390)	(0, 465)	(1, 464)	(14, 451)	(50, 415)	(17, 448)	(61, 404=)	(0, 465)	(7, 458)	(0, 465)	(97, 368)	(5, 460)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
einstein	<i>p</i> -değeri	2.13E-06	1.92E-06	1.73E-06	1.73E-06	1.92E-06	2.13E-06	4.29E-06	4.86E-05	1.73E-06	1.73E-06	2.35E-06	4.29E-06	2.13E-06
	(T+, T-)	(2, 463)	(1, 464)	(0, 465)	(0, 465)	(1, 464)	(2, 463)	(9, 456)	(35, 430)	(0, 465)	(0, 465)	(3, 462)	(9, 456)	(2, 463)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
lena	<i>p</i> -değeri	1.73E-06	3.52E-06	1.73E-06	1.73E-06	1.36E-05	2.37E-05	0.1020	1.36E-05	1.73E-06	3.52E-06	1.92E-06	3.41E-05	2.13E-06
	(T+, T-)	(0, 465)	(7, 458)	(0, 465)	(0, 465)	(21, 444)	(27, 438)	(153, 312)	(21, 444)	(0, 465)	(7, 458)	(1, 464)	(31, 434)	(2, 463)
	KD	+	+	+	+	+	+	=	+	+	+	+	+	+
autumn	<i>p</i> -değeri	1.73E-06	1.06E-04	1.73E-06	1.73E-06	5.22E-06	6.98E-06	6.34E-06	5.22E-06	1.92E-06	2.35E-06	1.73E-06	5.75E-06	1.73E-06
	(T+, T-)	(0, 465)	(44, 421)	(0, 465)	(0, 465)	(11, 454)	(14, 451)	(13, 452)	(11, 454)	(1, 464)	(3, 462)	(0, 465)	(12, 453)	(0, 465)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
cameraman	<i>p</i> -değeri	4.73E-06	3.52E-06	1.73E-06	4.73E-06	0.5304	4.90E-04	0.8130	0.9426	1.73E-06	3.41E-05	1.36E-05	0.0545	1.83E-03
	(T+, T-)	(10, 455)	(7, 458)	(0, 465)	(10, 455)	(202, 263)	(63, 402)	(221, 244)	(236, 229)	(0, 465)	(31, 434)	(21, 444)	(139, 326)	(81, 384)
	KD	+	+	+	+	=	+	=	=	+	+	+	=	+
peppers	<i>p</i> -değeri	1.92E-06	0.0196	1.92E-06	1.92E-06	0.0104	0.0627	0.1306	6.04E-03	1.73E-06	3.41E-05	4.29E-06	0.7499	3.52E-06
	(T+, T-)	(1, 464)	(119, 346)	(1, 464)	(1, 464)	(108, 357)	(142, 323)	(159, 306)	(99, 366)	(0, 465)	(31, 434)	(9, 456)	(248, 217)	(7, 458)
	KD	+	+	+	+	+	=	=	+	+	+	+	=	+
cell	<i>p</i> -değeri	1.73E-06	1.80E-05	1.73E-06	1.73E-06	1.92E-06	2.88E-06	3.18E-06	5.22E-06	1.73E-06	2.13E-06	1.92E-06	1.24E-05	2.13E-06
	(T+, T-)	(0, 465)	(24, 441)	(0, 465)	(0, 465)	(1, 464)	(5, 460)	(6, 459)	(11, 454)	(0, 465)	(2, 463)	(1, 464)	(20, 445)	(2, 463)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
circuit	<i>p</i> -değeri	1.92E-06	2.60E-06	1.73E-06	2.35E-06	3.52E-06	1.80E-05	1.36E-05	0.1254	1.73E-06	5.22E-06	2.88E-06	9.84E-03	2.37E-05
	(T+, T-)	(1, 464)	(4, 461)	(0, 465)	(3, 462)	(7, 458)	(24, 441)	(21, 444)	(158, 307)	(0, 465)	(11, 454)	(5, 460)	(107, 358)	(27, 438)
	KD	+	+	+	+	+	+	+	=	+	+	+	+	+
trees	<i>p</i> -değeri	1.92E-06	0.0132	1.73E-06	5.75E-06	4.86E-05	2.05E-04	1.13E-05	3.16E-03	1.92E-06	1.92E-06	5.79E-05	7.27E-03	4.07E-05
	(T+, T-)	(1, 464)	(112, 353)	(0, 465)	(12, 453)	(35, 430)	(52, 413)	(19, 446)	(89, 376)	(1, 464)	(1, 464)	(37, 428)	(102, 363)	(33, 432)
	KD	+	+	+	+	+	+	+	+	+	+	+	+	+
mri	<i>p</i> -değeri	5.22E-06	0.0175	1.73E-06	2.13E-06	2.77E-03	1.60E-04	1.25E-04	0.1915	1.73E-06	8.47E-06	4.99E-03	0.9918	2.77E-03
	(T+, T-)	(11, 454)	(117, 348)	(0, 465)	(2, 463)	(87, 378)	(49, 416)	(46, 419)	(296, 169)	(0, 465)	(16, 449)	(96, 369)	(232, 233)	(87, 378)
	KD	+	+	+	+	+	+	+	=	+	+	+	=	+
<b>Toplam (+/=-)</b>		<b>12/0/0</b>	<b>12/0/0</b>	<b>12/0/0</b>	<b>12/0/0</b>	<b>11/1/0</b>	<b>11/1/0</b>	<b>9/3/0</b>	<b>9/3/0</b>	<b>12/0/0</b>	<b>12/0/0</b>	<b>12/0/0</b>	<b>9/3/0</b>	<b>12/0/0</b>



Şekil 4.5: İyileştirme uygulanan görüntüler üzerinde algoritmaların yakınsama eğrileri.



**Şekil 4.6:** AISA ile iyileştirilen görüntüler ve histogramları (a) orijinal görüntü; (b) orijinal görüntünün histogramı; (c) iyileştirilen görüntü; (d) iyileştirilen görüntünün histogramı.



**Şekil 4.6 (devam):** AISA ile iyileştirilen görüntüler ve histogramları (a) orijinal görüntü; (b) orijinal görüntünün histogramı; (c) iyileştirilen görüntü; (d) iyileştirilen görüntünün histogramı.

## 5. SONUÇLAR VE TARTIŞMALAR

Tez kapsamında mühendislik optimizasyon problemlerini çözebilmek için Ergen Kimlik Arama Algoritması (Adolescent Identity Search Algorithm - AISA) adı verilen yeni, güçlü, gürbüz ve gelecek vadeden bir metasezgisel algoritma önerilmiştir. Önerilen algoritma sınıflama açısından hem popülasyon hem de insan tabanlıdır. AISA ergenlerin kimlik oluşum süreçlerini taklit etmektedir. Ergenlerin belirgin kimlik davranışları, gruptaki bireylerin özelliklerini gözleme ve mantık yürütme, bir rol model taklit etme ve istenmeyen bir özelliği benimseme temellerine dayanılarak matematiksel olarak modellenmiştir. Ayrıca AISA'yı diğer metasezgisellerden ayıran en önemli karakteristik özellik, algoritmanın LSE tarafından optimize edilmiş bir CFLN kullanarak kısmi uygunluk değerleri bulmaya çalışmasıdır. İlk olarak, önerilen algoritmanın keşif, sömürü, yerel optimumdan kaçınma ve yakınsama performansını test etmek için tek-modlu, çok-modlu, sabit boyutlu çok-modlu, bileşik ve CEC 2019 dahil olmak üzere beş farklı türden oluşan 39 test problemi kullanılmıştır. Sonuçlar, AISA'nın 13 iyi bilinen meta-sezgisel algoritmaya (PSO, ABC, GSA, BA, GWO, MFO, VS, JA, SCA, SSA, TSA, DE ve ES) kıyasla standart istatistiksel analiz, ölçeklenebilirlik analizi, Wilcoxon işaretli sıra testi, yakınsama analizi ve zaman verimliliği analizi performans değerleri açısından çok daha iyi sonuçlar üretebildiğini ortaya koymuştur. Çalışmanın devamında AISA'nın performansı; IIR sistem tanımlama, 7-DOF robot manipülatörünün ters kinematik ve görüntü iyileştirme mühendislik problemleri üzerinde araştırılmıştır. Elde edilen genel sonuçlara göre AISA, üstün ve rekabetçi performansa sahip yeni bir metasezgisel algoritmadır.

Elde edilen tüm optimizasyon sonuçları değerlendirildiğinde, önerilen AISA ile diğer algoritmalarından daha iyi veya onlarla rekabet edebilir sonuçlar elde edilmiştir. Bu denli iyi sonuçlar elde edilmesinin ardında yatan sebep AISA'nın farklı arama yeteneklerine sahip olabilmesi için bilinçli bir şekilde iyi tasarlanmış olmasıdır. AISA,  $r_4$  parametresi tarafından yürütülen üç mekanizmaya sahiptir. İlk mekanizma, algoritmayı sömürü üzerine odaklamak için tasarlanmıştır. Bu

mekanizmada en iyi özellik vektörü ( $\mathbf{x}^*$ ), her iterasyonun başında en küçük kareler tahmini yöntemi tarafından parametreleri kestirilmiş bir Chebyshev fonksiyonel bağlantı ağı kullanılarak bulunur. Sonrasında birey,  $r_1$  adım uzunluğu kullanılarak bu vektöre doğru yönlendirilir. İkinci mekanizma, algoritmanın daha dengeli bir sömürü kabiliyetine sahip olmasını sağlamak için tasarlanmıştır. Bu mekanizma, algoritmanın en iyi çözüme/rol modele ( $\mathbf{x}^{r_m}$ ) doğru bir yönelime sahip olduğu ancak başka bir  $\mathbf{x}^p$  çözümü kullanarak da olgunlaşmamış bir yakınsamayı önlemek için mücadele ettiği anlamına gelir. Üçüncü mekanizma ise algoritmanın keşif kabiliyeti üzerine oluşturulmuştur. Bu mekanizmada algoritma, keşfedilmemiş bölgelere yayılmaya yardımcı olan  $r_3$  rastgele vektörünü kullanarak  $\mathbf{x}^q$  negatif kimlik vektörüne doğru bir yönelime sahiptir. Bu özellikler teorik olarak AISA'nın arama davranışının anlaşılmasına yardımcı olabilir. AISA hem standart kıyaslama test fonksiyonlarını hem de gerçek hayat mühendislik problemlerini çözmeye diğer rakip metasezgisel algoritmalara göre üstün optimizasyon performansı üretmede güçlü ve sağlam bir yeteneğe sahipken, kendine özgü bir modelleme süreci içerdiğinden ilave hesaplama yüküne sahiptir. Ayrıca en küçük kareler tahmin yönteminde, kötü koşullu bir matrisin tersinin sağlıklı alınamamasından kaynaklı bir sorun ortaya çıkabilir. Bu nedenle sonraki çalışmalarda Pseudo ters veya özyinelemeli versiyonu iyileştirmeler için kullanılabilir.

Bu çalışma, gelecekteki çalışmalar için yeni fikirlere ve geliştirilmeye pek çok yönden açıktır. Ergen kimlik gelişim süreci psikososyal çerçeveden ele alındığında karmaşık bir kavram olduğundan, farklı, yeni ve daha derin bakış açılarıyla algoritma zenginleştirilebilir. İkinci gelişime açık yön şudur ki; önerilen ortogonal Chebyshev ağı yerine farklı ağ yapıları entegre edilebilir ve algoritmanın performansına etkileri gözlemlenebilir. Bir başka çalışılmaya müsait yön ise AISA algoritmasının ikili ve çok amaçlı problemlere yönelik versiyonlarının geliştirilerek teorik ve gerçek hayat problemlerine daha gerçekçi ve kapsamlı çözümler sağlayabilecek olmasıdır. Yeni, özgün ve gelecek vadeden bir algoritma olma özelliği ile AISA, alternatif çözüm

yöntemleri ile çözdürülmüş olan diğer optimizasyon problemlerinde daha iyi sonuçlar elde etmek için kullanılabilir.

## 6. KAYNAKLAR

- Abdechiri, M., Meybodi, M. R. and Bahrami, H., “Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO)”, *Applied Soft Computing*, 13 (5), 2932-2946, (2013).
- Ahmadi, S. A., “Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems”, *Neural Computing and Applications*, 28 (1), 233-244, (2017).
- Akyıldız, M. and Çetinkaya, K., “Comparison of four different heuristic optimization algorithms for the inverse kinematics solution of a real 4-DOF serial robot manipulator”, *Neural Comput. Appl.*, 27 (4), 825–836, (2016).
- Al-Ani, W. and Kadhum, M. J., “The Study of Unhealthy Eating Habits among Secondary Schools Students in Babel Governorate”, *Mustansiriya Medical Journal*, 12 (1), (2013).
- Alba, E. and Dorronsoro, B., “The exploration/exploitation tradeoff in dynamic cellular genetic algorithms”, *IEEE Transactions on Evolutionary Computation*, 9 (2), 126-142, (2005).
- Askarzadeh, A., “A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm”. *Computers & Structures*, 169, 1-12, (2016).
- Atak, H., “Kimlik Gelişimi ve Kimlik Biçimlenmesi: Kuramsal Bir Değerlendirme”, *Psikiyatride Güncel Yaklaşımlar*, 3 (1), 163-213, (2011).
- Bandura, A., *Social foundations of thought and action: A social cognitive theory*, Englewood Cliffs, NJ: Prentice-Hall, (1986).
- Bandura, A., *Social Learning Theory*, General Learning Press, (1971).
- Baykasoğlu, A. and Akpınar, Ş., “Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems - Part 2: Constrained optimization”, *Applied Soft Computing*, 37, 396-415, (2015).
- Campbell, S. B., *Behavior problems in preschool children: clinical and developmental issues*, Guilford Press, (2006).
- Cheng, M. Y. and Prayogo, D., “Symbiotic organisms search: a new metaheuristic optimization algorithm, *Computers & Structures*, 139, 98-112, (2014).
- Chu, S. C., Tsai, P. W. and Pan, J. S., “Cat swarm optimization”, *Pacific Rim international conference on artificial intelligence*, Berlin Heidelberg, Springer, 854-858, (2006).



- Civicioglu, P., “Backtracking search optimization algorithm for numerical optimization problems”, *Applied Mathematics and Computation*, 219 (15), 8121-8144, (2013).
- Çetin, M., Bahtiyar, B. and Beyhan, S., “Adaptive uncertainty compensation-based nonlinear model predictive control with real-time applications”, *Neural Computing and Applications*, 31 (2), 1029-1043, (2019).
- Dai, C., Chen, W. and Zhu, Y., “Seeker optimization algorithm for digital IIR filter design”, *IEEE Transactions on Industrial Electronics*, 57 (5), 1710-1718, (2009).
- Das, S., Mullick, S. S. and Suganthan, P. N., “Recent advances in differential evolution-an updated survey”, *Swarm and Evolutionary Computation*, 27, 1-30, (2016).
- Del Ser, J., Osaba, E., Molina, D., Yang, X. S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello Coello, C. C. and Herrera, F., “Bio-inspired computation: Where we stand and what’s next”, *Swarm and Evolutionary Computation*, 48, 220-250, (2019).
- Dereli, S., “Robotların Ters Kinematik Çözümünde Sezgisel Optimizasyon Algoritmalarının Kullanılması ve FPGA ile Gerçekleştirilmesi”, Doktora Tezi, Sakarya Uygulamalı Bilimler Üniversitesi Lisansüstü Eğitim Enstitüsü, Mekatronik Mühendisliği Anabilim Dalı, Sakarya, (2019).
- Dereli, S., Köker, R., “A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm”, *Artif. Intell. Rev.* 53, 949–964, (2020<sup>a</sup>).
- Dereli, S. and Köker, R., “Simulation based calculation of the inverse kinematics solution of 7-DOF robot manipulator using artificial bee colony algorithm”, *SN Appl. Sci.*, 2 (27), (2020<sup>b</sup>).
- Derrac, J., García, S., Molina, D. and Herrera, F., “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms”, *Swarm and Evolutionary Computation*, 1 (1), 3-18, (2011).
- Dhiman, G. and Kumar V., “Emperor penguin optimizer: A bio-inspired algorithm for engineering problems”, *Knowledge-Based Systems*, 159, 20-50, (2018).
- Doğan, B. and Ölmez, T., “A new metaheuristic for numerical function optimization: Vortex Search algorithm”, *Information Sciences*, 293, 125-145, (2015).
- Dorigo, M., Maniezzo, V. and Colorni, A., “Ant system: optimization by a colony of cooperating agents”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26 (1), 29-41, (1996).

- Draa, A. and Bouaziz, A., “An artificial bee colony algorithm for image contrast enhancement”, *Swarm and Evolutionary computation*, 16, 69-84, (2014).
- Erikson, E. H., *Childhood and society (2nd ed.)*, New York: Norton, (1963).
- Erol, O. K. and Eksin, I., “A new optimization method: Big Bang–Big Crunch”, *Advances in Engineering Software*, 37 (2), 106-111, (2006).
- Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. and Tavakkoli-Moghaddam, R., “The Social Engineering Optimizer (SEO)”, *Engineering Applications of Artificial Intelligence*, 72, 267-293, (2018).
- Fister Jr, I., Yang, X.-S., Fister, I. Brest, J. and Fister, D., “A Brief Review of Nature-Inspired Algorithms for Optimization”, arXiv:1307.4186, (2013).
- Gandomi, A. H., “Interior search algorithm (ISA): a novel approach for global optimization”, *ISA Transactions*, 53 (4), 1168-1183, (2014).
- Gandomi, A. H. and Alavi, A. H., “Krill herd: a new bio-inspired optimization algorithm”, *Communications in Nonlinear Science and Numerical Simulation*, 17 (12), 4831-4845, (2012).
- Ghorbani, N. and Babaei, E., “Exchange market algorithm”, *Applied Soft Computing*, 19, 177-187, (2014).
- Glover, F., “Future paths for integer programming and links to artificial intelligence”, *Computers & Operations Research*, 13 (5), 533-549, (1986).
- Gogna, A. and Tayal, A., “Metaheuristics: review and application”, *Journal of Experimental & Theoretical Artificial Intelligence*, 25 (4), 503-526, (2013).
- Gotmare, A., Bhattacharjee, S. S., Patidar, R. and George N. V., “Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review”, *Swarm and Evolutionary Computation*, 32, 68-84, (2017).
- Grotevant, H. D., “Coming to terms with adoption: The construction of identity from adolescence into adulthood”, *Adoption Quarterly*, 1 (1), 3-27, (1997).
- Hatamlou, A., “Black hole: A new heuristic optimization approach for data clustering”, *Information Sciences*, 222, 175-184, (2013).
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., “Harris hawks optimization: Algorithm and applications”, *Future Generation Computer Systems*, 97, 849-872, (2019).
- Holland, J., *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, (1975).

- Hu, X., “Particle swarm optimization tutorial [online]”, (22 March 2021), <https://titan.csit.rmit.edu.au/e46507/publications/pso-tutorial-seal06.pdf>, (2006).
- Hurd, N. M., Wittrup, A. and Zimmerman, M. A., Role Models in Adolescent Development, *Encyclopedia of Adolescence*, New York: Springer, 2399-2404, (2011).
- Jain, M., Singh, V. and Rani, A., “A novel nature-inspired algorithm for optimization: Squirrel search algorithm”, *Swarm and Evolutionary Computation*, 44, 148-175, (2019).
- Javidy, B., Hatamlou, A. and Mirjalili, S., “Ions motion algorithm for solving optimization problems”, *Applied Soft Computing*, 32, 72-79, (2015).
- Kalinli, A. and Karaboga, N., “Artificial immune algorithm for IIR filter design”, *Engineering Applications of Artificial Intelligence*, 18 (8), 919-929, (2005).
- Kamoona, A. M. and Patra, J. C., “A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images”, *Applied Soft Computing*, 85, 105749, (2019).
- Karaboga, D. ve Basturk, B., “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *Journal of Global Optimization*, 39 (3), 459-471, (2007).
- Karaboga, N., “A new design method based on artificial bee colony algorithm for digital IIR filters”, *Journal of the Franklin Institute*, 346 (4), 328-348, (2009).
- Karaçalı, H., “Beş Eklemlı Çapak Alma Robotu Tasarımı”, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Anabilim Dalı, Eskişehir, (2012).
- Kaveh, A. and Khayatad, M., “A new meta-heuristic method: Ray Optimization”, *Computers & Structures*, 112, 283-294, (2012).
- Kaveh, A. and Talatahari, S., “A novel heuristic optimization method: charged system search”, *Acta Mechanica*, 213 (3), 267-289, (2010).
- Kennedy, J. and Eberhart, R., “Particle swarm optimization”, *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942-1948, (1995).
- Kiran, M. S., “TSA: Tree-seed algorithm for continuous optimization”, *Expert Systems with Applications*, 42 (19), 6686-6698, (2015).
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., “Optimization by simulated annealing”, *Science*, 220 (4598), 671-680, (1983).

- Lagos-Eulogio, P., Seck-Tuoh-Mora, J. C., Hernandez-Romero, N. and Medina-Marin, J., “A new design method for adaptive IIR system identification using hybrid CPSO and DE”, *Nonlinear Dynamics*, 88 (4), 2371-2389, (2017).
- Lam, A. Y. S. and Li, V. O. K., “Chemical-Reaction-Inspired Metaheuristic for Optimization”, *IEEE Transactions on Evolutionary Computation*, 14 (3), 381-399, (2009).
- Larrañaga, P. and Lozano, J. A., *Estimation of distribution algorithms: A new tool for evolutionary computation*, Kluwer Academic Publishers, (2002).
- Lee, T. T. and Jeng, J. T., “The Chebyshev-polynomials-based unified model neural networks for function approximation”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28 (6), 925-935, 1998.
- Liang, J. J., Suganthan, P. N. and Deb K., “Novel composition test functions for numerical global optimization”, *Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005*, 68-75, (2005).
- Marcia, J. E., “Development and validation of ego-identity status”, *Journal of Personality and Social Psychology*, 3 (5), 551-558, (1966).
- Mahapatra, P.K., Ganguli, S. and Kumar, A., “A hybrid particle swarm optimization and artificial immune system algorithm for image enhancement”, *Soft computing*, 19 (8), 2101–2109, (2015).
- Maurya, L., Mahapatra, P. K. and Kumar, A., “A social spider optimized image fusion approach for contrast enhancement and brightness preservation”, *Applied Soft Computing*, 52, 575-592, (2017).
- Mirjalili, S., “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm”, *Knowledge-Based Systems*, 89, 228-249, (2015).
- Mirjalili, S., “SCA: A Sine Cosine Algorithm for solving optimization problems”, *Knowledge-Based Systems*, 96, 120-133, (2016).
- Mirjalili, S. and Lewis, A., “The whale optimization algorithm”, *Advances in Engineering Software*, 95, 51-67, (2016).
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. and Mirjalili, S. M., “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems”, *Advances in Engineering Software*, 114, 163-191, (2017).
- Mirjalili, S., Mirjalili, S. M. and Hatamlou, A., “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization”, *Neural Computing and Applications*, 27 (2), 495-513, (2016).

- Mirjalili, S., Mirjalili, S. M. and Lewis, A., “Grey Wolf Optimizer”, *Advances in Engineering Software*, 69, 46-61, (2014).
- Moghdani, R. and Salimifard, K., “Volleyball Premier League Algorithm”, *Applied Soft Computing*, 64, 161-185, (2018).
- Molina, D., Poyatos, J., Del Ser, J., García, S., Hussain, A. and Herrera, F., “Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations”, *Cognitive Computation*, 12 (5), 897-939, (2020).
- Munteanu, C. and Rosa, A., “Gray-scale image enhancement as an automatic process driven by evolution”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34 (2), 1292-1298, (2004).
- Pan, W. T., “A new fruit fly optimization algorithm: taking the financial distress model as an example”, *Knowledge-Based Systems*, 26, 69-74, (2012).
- Panda, G., Pradhan, P. M. and Majhi, B., “IIR system identification using cat swarm optimization”, *Expert Systems with Applications*, 38 (10), 12671-12683 pp., (2011).
- Patra, J. C., and Kot, A. C., “Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32 (4), 505-511, (2002).
- Price, K. V., Awad, N. H., Ali, M. Z. and Suganthan, P. N., “Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization”, *Technical Report, Nanyang Technological University*, (2018).
- Ragelienė, T., Links of adolescents identity development and relationship with peers: A systematic literature review, *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, 25 (2), 97-105, (2016).
- Rao, R. V., “Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems”, *International Journal of Industrial Engineering Computations*, 7 (1), 19-34, (2016).
- Rao, R. V., Savsani, V. J. and Vakharia, D. P., “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems”, *Computer-Aided Design*, 43 (3), 303-315, (2011).
- Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S., “GSA: A Gravitational Search Algorithm”, *Information Sciences*, 179 (13), 2232-2248, (2009).
- Rashedi, E., Rashedi, E., and Nezamabadi-pour, H., “A comprehensive survey on gravitational search algorithm”, *Swarm and Evolutionary Computation*, 41, 141-158, (2018).

- Rechenberg, I., *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, (1973).
- Rokbani N., Casals A. and Alimi A. M., “IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm”, (eds: Azar A. and Vaidyanathan S.), *Computational Intelligence Applications in Modeling and Control*, 575, Springer, 369-395, (2015).
- Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M., “Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems”, *Applied Soft Computing*, 13 (5), 2592-2612, (2013).
- Salcedo-Sanz, S., “Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures”, *Physics Reports*, 655, 1-70, (2016).
- Salomon, R., “Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms”, *Biosystems*, 39 (3), 263-278, 1996.
- Saremi, S., Mirjalili, S. and Lewis, A., “Grasshopper Optimisation Algorithm: Theory and application”, *Advances in Engineering Software*, 105, 30-47, (2017).
- Schwefel, H. P., *Evolution and Optimum Seeking: The Sixth Generation*, John Wiley & Sons, Inc., (1993).
- Simon, D., “Biogeography-Based Optimization”, *IEEE Transactions on Evolutionary Computation*, 12 (6), (2008).
- Sörensen, K., “Metaheuristics—the metaphor exposed”, *International Transactions in Operational Research*, 22 (1), 3-18, (2015).
- Storn, R. and Price, K., “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, 11 (4), 341-359, (1997).
- Suresh, S. and Lal, S., “Modified differential evolution algorithm for contrast and brightness enhancement of satellite images”, *Applied Soft Computing*, 61, 622-641, (2017).
- Suresh, S., Lal, S., Reddy, C. S. and Kiran, M. S., “A novel adaptive cuckoo search algorithm for contrast enhancement of satellite images”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10 (8), 3665-3676, (2017).
- Talbi, E. G., *Metaheuristics: from design to implementation*, 74, John Wiley & Sons, (2009).

- Toz, M., “Chaos-based Vortex Search algorithm for solving inverse kinematics problem of serial robot manipulators with offset wrist”, *Applied Soft Computing*, 89, (2020).
- Tsang, S. K., Hui, E. K. and Law, B. C., “Positive Identity as a Positive Youth Development Construct: A Conceptual Review”, *The Scientific World Journal*, 2012, 1-7, (2012).
- Uymaz, S. A., Tezel, G. and Yel, E., “Artificial algae algorithm (AAA) for nonlinear global optimization”, *Applied Soft Computing*, 31, 153-171, (2015).
- Weyland, D., “A rigorous analysis of the harmony search algorithm: How the research community can be misled by a “novel” methodology”, *International Journal of Applied Metaheuristic Computing (IJAMC)*, 1 (2), 50-60, (2010).
- Wolpert, D. H. and Macready, W. G., “No free lunch theorems for optimization”, *IEEE Transactions on Evolutionary Computation*, 1 (1), 67-82, (1997).
- Yang, X.-S., “A New Metaheuristic Bat-Inspired Algorithm”, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65-74, (2010).
- Yang, X.-S., “Firefly Algorithms for Multimodal Optimization”, (eds: O. Watanabe, t. Zeugmann), *Stochastic Algorithms: Foundations and Applications*, 169-178, (2009).
- Yang, X.-S., “Nature-inspired optimization algorithms: Challenges and open problems”, *Journal of Computational Science*, 46, 101104, (2020).
- Yang, X.-S. and Dep, S., “Cuckoo Search via Lévy flights”, *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, 210-214, (2009).
- Yapici, H. and Cetinkaya, N., “A new meta-heuristic optimizer: Pathfinder algorithm”, *Applied Soft Computing*, 78, 545-568, (2019).
- Yue, X. and Zhang, H., “A Novel Industrial Image Contrast Enhancement Technique Based on an Improved Ant Lion Optimizer”, *Arabian Journal for Science and Engineering*, 1-12, (2020).