T. C.
SİVAS CUMHURİYET ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

# GÜNCEL METASEZGİSEL ALGORİTMALARIN PERFORMANS ANALİZİ

**YÜKSEK LİSANS TEZİ**

**Metin KALYON**

**(20219257012)**

**Bilgisayar Mühendisliği Ana Bilim Dalı**

**Tez Danışmanı: Doç. Dr. Sibel ARSLAN**

**SİVAS**

**AĞUSTOS 2024**

**Metin KALYON** ’un hazırladığı ve **“GÜNCEL METASEZGİSEL ALGORİTMALARIN PERFORMANS ANALİZİ”** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**’ nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

| | | |
|---|---|---|
| **Tez Danışmanı** | **Doç. Dr. Sibel ARSLAN**<br>Sivas Cumhuriyet Üniversitesi | …… |
| **Jüri Üyesi** | **Doç. Dr. Hidayet TAKÇI**<br>Sivas Cumhuriyet Üniversitesi | …… |
| **Jüri Üyesi** | **Doç. Dr. Turgut ÖZSEVEN**<br>Tokat Gaziosmanpaşa Üniversitesi | …… |

Bu tez, Sivas Cumhuriyet Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

**Prof. Dr. Nevcihan GÜRSOY**
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

Bu tez, Sivas Cumhuriyet Üniversitesi Senatosu'nun 20.08.2014 tarihli ve 7 sayılı kararı ile kabul edilen Fen Bilimleri Enstitüsü Lisansüstü Tez Yazım Kılavuzu (Yönerge)' nda belirtilen kurallara uygun olarak hazırlanmıştır.

**ETİK**

Sivas Cumhuriyet Üniversitesi Fen Bilimleri Enstitüsü, Tez Yazım Kılavuzu (Yönerge)' nda belirtilen kurallara uygun olarak hazırladığım bu tez çalışmasında;

- ✓ Bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- ✓ Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- ✓ Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere, bilimsel normlara uygun olarak atıfta bulunduğumu ve atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- ✓ Bütün bilgilerin doğru ve tam olduğunu, kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ✓ Tezin herhangi bir bölümünü, Sivas Cumhuriyet Üniversitesi veya bir başka üniversitede, bir başka tez çalışması olarak sunmadığımı; beyan ederim.

12.08.2024

Metin KALYON

## KATKI BELİRTME VE TEŞEKKÜR

Çalışmalarım boyunca beni yönlendiren, hiçbir konuda yardımlarını esirgemeyen, bilgi ve deneyimlerinden sürekli yararlandığım kıymetli danışman hocam Doç. Dr. Sibel ARSLAN' a saygılarımı sunar, teşekkürü bir borç bilirim.

Çalışmalarımda hiç tereddüt etmeden her türlü desteği sağlayan sevgili eşim Ümmühan BULUT KALYON' a çok teşekkür ederim.

Son olarak hayatım boyunca her zaman yanımda olan, bugünlere gelmemi sağlayan babam Halil KALYON başta olmak üzere tüm aileme en içten şükranlarımı sunarım.

# ÖZET

## GÜNCEL METASEZGİSEL ALGORİTMALARIN PERFORMANS ANALİZİ

**Metin KALYON**

**Yüksek Lisans Tezi**
**Bilgisayar Mühendisliği Ana Bilim Dalı**
**Danışman: Doç. Dr. Sibel ARSLAN**
**2024, 113+xvi sayfa**

Günümüzde, metasezgiseller optimizasyon problemlerinin çözümünde çok önemli bir rol oynamaktadır. Bu tez çalışmasında, son 5 yılda önerilen popülasyon tabanlı 10 metasezgisel (Harris Şahinleri Optimizasyonu-HHO, İsli Sumru Optimizasyon Algoritması-STOA, Karadul Optimizasyonu-BWO, Aritmetik Optimizasyon Algoritması-AOA, Afrika Akbabaları Optimizasyon Algoritması-AVOA, Kaya Kartalı Optimizasyon Algoritması-AO, Yapay Tavşan Optimizasyonu-ARO, Dağ Ceylanı Optimizasyonu-MGO, Çayır Köpeği Optimizasyonu-PDO, Kerevit Optimizasyon Algoritması-COA) kıyaslanmıştır. Algoritmalar ile kalite test fonksiyonları ve mühendislik tasarım problemleri çözülmüştür. Bildiğimiz kadarıyla, bu algoritmaların performansları ilk kez karşılaştırılmıştır. Simülasyon sonuçları, yakınsama grafikleri ve istatistiksel test sonuçlarına göre en yüksek başarıya sahip üç algoritma sırasıyla AVOA, MGO ve AO' dur. Gelecekteki çalışmalarda çeşitli metasezgisellerden yararlanılarak bu üç algoritmanın daha sağlam versiyonları ile farklı mühendislik problemlerinin çözülmesi hedeflenmektedir.

**Anahtar kelimeler:** Metasezgiseller, Afrika Akbabaları Optimizasyon Algoritması, Dağ Ceylanı Optimizasyonu, Kaya Kartalı Optimizasyonu, Kalite Test Fonksiyonları, Mühendislik Tasarım Problemleri

## ABSTRACT

### PERFORMANCE ANALYSIS OF CURRENT METAHEURISTIC ALGORITHMS

**Metin KALYON**

**Master of Science Thesis**

**Department of Computer Engineering**

**Supervisor: Asst. Prof. Sibel ARSLAN**

**2024, 113+ xvi pages**

Nowadays, metaheuristics play a very important role in solving optimization problems. In this thesis study, 10 population-based metaheuristics proposed in the last 5 years (Harris Hawks Optimization-HHO, Sooty Tern Optimization Algorithm-STOA, Black Widow Optimization -BWO, Arithmetic Optimization Algorithm-AOA, African Vultures Optimization Algorithm-AVOA, Aquila Optimization Algorithm). -AO, Artificial Rabbit Optimization-ARO, Mountain Gazelle Optimization-MGO, Prairie Dog Optimization-PDO, Crayfish Optimization Algorithm-COA) were compared. Quality test functions and engineering design problems are solved with algorithms. To our knowledge, this is the first time the performances of these algorithms have been compared. According to simulation results, convergence graphs and statistical test results, the three most successful algorithms are AVOA, MGO and AO, respectively. In future studies, it is aimed to solve different engineering problems with more robust versions of these three algorithms by utilizing various metaheuristics.



**Keywords:** Metaheuristics, African Vultures Optimization Algorithm, Mountain Gazelle Optimization, Aquilla Optimization, Quality Test Functions, Engineering Design Problems

# İÇİNDEKİLER

# ŞEKİLLER DİZİNİ

**Sayfa**

# ÇİZELGELER DİZİNİ

**Sayfa**

# SİMGELER DİZİNİ

| | |
|---|---|
| $\mathbf{X_{tavşan}}$ | Tavşanın anlık konumu |
| $\mathbf{X_{rand}}$ | Tavşanın rastgele konumu |
| $\mathbf{X_m}$ | Şahinlerin ortalama konumu |
| $\mathbf{X(t)}$ | Şahinlerin anlık konumu |
| $\mathbf{r}$ | Bir avın başarılı şekilde kaçma olasılığı değeri |
| $\mathbf{r_1, r_2, r_3, r_4,}$ | Şahin konumunu belirleyen $0-1$ arasındaki rastgele sayılardır |
| $\mathbf{t}$ | Problem çözümünde uygulanan adımlardan her biri |
| $\mathbf{LB}$ | Problemin alt sınırı |
| $\mathbf{UB}$ | Problemin üst sınırı |
| $\mathbf{T}$ | Problem çözümünde uygulanan toplam adım sayısı |
| $\mathbf{N}$ | Toplam şahin sayısı |
| $\mathbf{\Delta X(t)}$ | Tavşanın $t$'inci iterasyondaki pozisyon vektörü |
| $\mathbf{J}$ | Tavşan konumu belirleyen bir parametre |
| $\mathbf{r_5}$ | Tavşan konumu belirleyen $0-1$ aralığındaki rastgele bir sayı |
| $\mathbf{X_i(t)}$ | t. iterasyondaki şahin konumu |
| $\mathbf{E}$ | Avın kaybolan enerjisi |
| $\mathbf{E_0}$ | Avın ilk enerjisi |
| $\mathbf{D}$ | Problem boyutu |
| $\mathbf{s}$ | Rastgele bir vektör |
| $\mathbf{Z}$ | Harris şahinin hesaplanan ani dalış değeri |
| $\mathbf{Y}$ | Harris şahinin hesaplanan yumuşak kuşatma değeri |
| $\mathbf{LF}$ | Levy uçuşu fonksiyonu |
| $\vec{\mathbf{C}}_{\mathbf{st}}$ | İsli Sumrular ile çarpışmayan kuşun konumunu |
| $\vec{\mathbf{P}}_{\mathbf{st}}(\mathbf{z})$ | İsli Sumru'nun geçerli konumu |
| $\vec{\mathbf{P}}_{\mathbf{bst}}(\mathbf{z})$ | İsli Sumru'nun en iyi konumu |
| $\vec{\mathbf{D}}_{\mathbf{st}}$ | İsli Sumru'nun en iyi konumu ile geçerli konumu arasındaki fark |
| $\mathbf{S_A}$ | Geçerli yineleme |
| $\mathbf{C_f}$ | Geçerli yinelemeyi ayarlayan kontrol değişkeni |
| $\vec{\mathbf{M}}_{\mathbf{st}}$ | İsli Sumru'nun ulaştığı farkı konum |
| $\mathbf{R_{adius}}$ | İsli Sumru'nun spiral dönüş yarıçapı |
| $u$ | İsli Sumru'nun yaptığı spiral hareketin sabiti |
| $v$ | İsli Sumru'nun yaptığı spiral hareketin sabiti |
| $\mathbf{C_B}$ | Keşiften sorumlu değişken |
| $\mathbf{y_1}$ , $\mathbf{y_2}$ | Genç örümcekler |
| $x_1$ , $\mathbf{x_2}$ | Ebeveyn örümcekler |
| $\mathbf{X}$ | Çözüm dizisi |
| $\mathbf{C_{iterasyon}}$ | Geçerli iterasyon adımı |
| $\mathbf{M_{iterasyon}}$ | Maksimum iterasyon adımı |
| $\boldsymbol{\varepsilon}$ | Adım boyu katsayısı |
| $\boldsymbol{\alpha}$ | Adım doğruluk katsayısı |
| $\mathbf{x_{i,j}(C_{iterasyon})}$ | Geçerli iterasyondaki $i$'nci çözümün $j$'nci konumunu verir. |
| $\mathbf{R(i)}$ | Akbaba uygunluk değeri |
| $\mathbf{L_1}$ , $\mathbf{L_2}$ | Akbaba uygunluk değer parametreleri |
| $\mathbf{p_i}$ | Akbaba grupları için en iyi çözüm seçilme parametresi |

| | |
|---|---|
| $\mathbf{F_i}$ | $i$ . akbabanın uygunluk değeri |
| $\mathbf{n}$ | Toplam akbaba sayısı |
| $\mathbf{t_c}$ | Değerlendirme sayısı |
| $\mathbf{F}$ | Akbaba doygunluğunu gösteren parametre |
| $\mathbf{w}$ | Keşif ve operasyon bilgisi veren parametre |
| $\mathbf{P}$ | Akbaba konum vektörü |
| $\mathbf{R(i)}$ | Seçilen en iyi akbaba |
| $\mathbf{D(i)}$ | En iyi akbaba ile mevcut akbaba arasındaki konum |
| $\mathbf{d(t)}$ | Akbabanın gruplardaki en iyi akbabaya olan mesafesi |
| $\mathbf{S_1}$ | 1. en iyi akbaba dönme hareket değeri |
| $\mathbf{S_2}$ | 2. en iyi akbaba dönme hareket değeri |
| $\mathbf{LB_j}$ | Problem alt sınırı |
| $\mathbf{UB_j}$ | Problem üst sınırı |
| $\mathbf{QF}$ | Keşif yöntemlerini dengeleyen kalite fonksiyonu |
| $\vec{\mathbf{x}}_{\mathbf{i,d}}$ | $i$. tavşanın konum vektörü |
| $\mathbf{f}$ | Uygunluk değeri |
| $\vec{\mathbf{v}}_{\mathbf{i}}\mathbf{(t+1)}$ | $t+1$. anındaki $i$. tavşan konumu |
| $\mathbf{L}$ | Hareket hızını temsil eden koşu uzunluğu |
| $\mathbf{H}$ | Gizlenme parametresi |
| $\vec{\mathbf{b}}_{\mathbf{i,j}}$ | $i$. tavşanın $j$. yuva konumu |
| $\vec{\mathbf{b}}_{\mathbf{i,r}}$ | $i$. tavşan tarafında rastgele seçilen yuva konumu |
| $\mathbf{A(t)}$ | Tavşanın enerji seviyesi |
| $\mathbf{BH}$ | Ceylan genç erkek sürüsü etki faktör katsayısı |
| $\mathbf{Cof}_r$ | Arama yeteneğini artırmak için kullanılan bir katsayı vektörü |
| $\mathbf{n_1}$ | Standart dağılım sayısı |
| $\mathbf{CT_{i,j}}$ | Kolonideki $i$. zümrenin $j$. boyutunun konumu |
| $\mathbf{PD_{i,j}}$ | $i$. çayır köpeğinin $j$. boyutunun konumu |
| $\mathbf{GBest}_{\mathbf{i,j}}$ | Global en iyi çözüm |
| $\mathbf{eCBest}_{\mathbf{i,j}}$ | Mevcut en iyi çözümün etkisi |
| $\Delta$ | Çayır köpekleri arasındaki tam sayı farkı |
| $\mathbf{PD_\varepsilon}$ | Çayır köpekleri besin kaynağının kalitesini gösteren parametre |
| $\mathbf{CPD_{i,j}}$ | Kolonideki tüm çayır köpeklerinin toplam etkisi |
| $\mathbf{PE}$ | Avcı etkisi |
| $\boldsymbol{\mu}$ | Kerevitler için en uygun sıcaklık değeri |
| $\mathbf{X_G}$ | Kerevitler için en uygun konum |
| $\mathbf{X_L}$ | Anlık kerevit konumu |
| $\mathbf{Q}$ | Kerevit yiyecek boyutu |
| $\mathbf{C_3}$ | En büyük kerevit yiyecek faktörü |
| $\mathbf{d\ x_1}$ | Tel çapı |
| $\mathbf{D\ x_2}$ | Ortalama bobin çapı |
| $\mathbf{N\ x_3}$ | Aktif bobin sayısı |
| $\mathbf{g_1}$ | Minimum sapma |
| $\mathbf{g_2}$ | Kayma gerilimi |
| $\mathbf{g_3}$ | Darbe frekansı |
| $\mathbf{g_4}$ | Dış çap sınırı |
| $\mathbf{T_s\ (X_1)}$ | Kabuğun kalınlığı |
| $\mathbf{T_h\ (X_2)}$ | Kafa kalınlığı |
| $\mathbf{R\ (X_3)}$ | İç yarıçap |

| | |
|---|---|
| $\mathbf{L}\ (\mathbf{X_4})$ | Kafa göz alınmadan silindirik bölümün uzunluğu |
| $\mathbf{h}\ (\mathbf{x_1})$ | Kabuğun kalınlığı |
| $\mathbf{l}\ (\mathbf{x_2})$ | Kaynak bağlantı uzunluğu |
| $\mathbf{t}\ (\mathbf{x_3})$ | Kirişin genişliği |
| $\mathbf{b}\ (\mathbf{x_4})$ | Kiriş kalınlığı |
| $\mathbf{b}\ \mathbf{x_1}$ | Yüz genişliği |
| $\mathbf{m}\ \mathbf{x_2}$ | Diş modülü |
| $\mathbf{z}\ \mathbf{x_3}$ | Pinyondaki diş sayısı |
| $\mathbf{l_1 x_4}$ | Yataklar arasındaki ilk milin uzunluğu |
| $\mathbf{l_2}\ \mathbf{x_5}$ | Yataklar arasındaki ikinci milin uzunluğu |
| $\mathbf{d_1}\ \mathbf{x_6}$ | Birinci milin çapı |
| $\mathbf{d_2}\ \mathbf{x_7}$ | İkinci milin çapı |
| $\mathbf{T_a}$ | 1.çarkın diş sayısı |
| $\mathbf{T_b}$ | 2.çarkın diş sayısı |
| $\mathbf{T_c}$ | 3.çarkın diş sayısı |
| $\mathbf{T_d}$ | 4.çarkın diş sayısı |
| $\mathbf{A_1}$ | Kafes çapraz kesit alanı |
| $\mathbf{A_2}$ | Kafes dikey kesit alanı |

## KISALTMALAR DİZİNİ

| | |
|---|---|
| **ABC** | Yapay Arı Koloni (Artificial Bee Colony) |
| **AO** | Kaya Kartalı Optimizasyon Algoritması (Aquila Optimizer) |
| **AOA** | Aritmetik Optimizasyon Algoritması (Aritmetich Optimization Algorithm) |
| **ARO** | Yapay Tavşan Optimizasyonu ( Artificial Rabbits Optimization) |
| **AVOA** | Afrika Akbabaları Optimizasyon Algoritması (African Vulture Optimization Algorithm) |
| **BWO** | Karadul Optimizasyonu (Black Widow Optimization) |
| **COA** | Kerevit Optimizasyon Algoritması (Crayfish Optimization Algorithm) |
| **GA** | Genetik Algoritma (Genetic Algorithm) |
| **HHO** | Harris Hawks Optimizasyonu (Harris Hawks Optimization) |
| **MGO** | Dağ Ceylanı Optimizasyonu (Mountain Gazelle Optimizer) |
| **PDO** | Çayır Köpeği Optimizasyonu (Prairie Dog Optimization) |
| **STOA** | İsli Sumru Optimizasyon Algoritması (Sooty Tern Optimization Algorithm) |

# 1. GİRİŞ

## 1.1 Tezin Amacı, Kapsamı ve Katkıları

Optimizasyon, bir problemin belirli koşullar altında var olan çözümlerinden en iyisini seçme işlemidir [1]. Optimizasyon problemlerinde genellikle matematiksel ve klasik sezgisel yöntemler kullanılmaktadır. Karmaşık sistemler için matematiksel modelin kurulması zorluğunun yanında, çözüm uzayının geniş olduğu problemlerde matematiksel yöntemler maliyet açısından dezavantajlı oldukları için tercih edilmezler. Klasik optimizasyon yöntemleri ise sürekli ve türevlenebilir amaç fonksiyonlarının optimum çözümünü bulmakta kullanışlıdır. Değişken sayısı arttıkça problemlerin klasik optimizasyon yöntemleri ile çözümü zorlaşmakta veya çok uzun süre almaktadır [2]. Optimizasyon problemlerinin çözümünde kullanılan matematiksel ve klasik sezgisel yöntemlerin yerine çok daha başarılı sonuçlar veren metasezgisel optimizasyon algoritmaları geliştirilmiştir.

Metasezgisel optimizasyon algoritmaları, bir optimizasyon problemindeki matematiksel modelin oluşturulmasının zor olduğu durumlarda veya büyük ölçekli, çok değişkenli optimizasyon problemlerine ait kabul edilebilir çözümler üreten bir optimizasyon yaklaşımıdır. Metasezgisel optimizasyon algoritmaların temelinde doğa olayları, canlı türlerinin sosyal davranışları ve evrimsel kavramlar yer almaktadır [3]. Bu algoritmalar çözüme ulaşmak için iki temel yapı kullanır. Bunlar: Keşif (exploration) ve sömürü (exploitation) mekanizmalarıdır. Keşif aşaması çözüme dair arama uzayını, arama sürecini ifade ederken; sömürü aşaması ise keşif ile elde edilen alanda daha doğru bir arama sürecini ifade eder. Metasezgisel bir optimizasyon algoritmasında keşif aşamasının yeteneği yerel optimumdan kurtulmasını ve farklı çözümler bulmasını sağlarken; sömürü aşamasının yeteneği ise çözümlerin optimuma yakınlığını sağlamaktadır. Bu nedenle metasezgisel optimizasyon algoritmalarında iki aşamanın dengeli olması gerekmektedir [4]. Bu algoritmalar; ekonomi, ticaret, finans, enerji, çizelgeleme, görüntü işleme, optimal kontrol ve mühendislik tasarım uygulamalarında çeşitli alanlarda kullanılmaktadır [5].

En temelde metasezgisel optimizasyon algoritmaları 4 kategoride ele alınır: Evrimsel optimizasyon algoritmaları, sürü zekasına dayalı optimizasyon algoritmaları, fizik

tabanlı optimizasyon algoritmaları ve insan tabanlı optimizasyon algoritmalarıdır [6]. Şekil 1.1.1' de gösterilmiştir [7].

Bu tez çalışmasında son zamanlarda ortaya çıkmış 10 farklı metasezgisel optimizasyon algoritması ayrıntılı olarak ele alınmış ve çalışma prensipleri hakkında bilgi verilmiştir. Daha sonra ise bu algoritmalar 23 klasik kalite testi fonksiyonları ile karşılaştırılmıştır. Bu metasezgisellerin seçilen gerçek dünya problemleri karşısındaki performansları da bu çalışma ile ortaya çıkarılmıştır. Çalışmanın temel katkıları aşağıda sıralanmıştır:

- Literatür taramasına göre bu çalışmada kullanılan algoritmalar ilk kez kıyaslanmıştır.
- Algoritmaların performansı çeşitli işlevler üzerinde test edilmiştir.
- Deneysel çalışmalara göre en iyi performans gösteren metasezgiseller değerlendirilmiştir.


**Şekil 1.1.1** Metasezgisel optimizasyon algoritmalarının sınıflandırılması

Çalışmada yer alan metasezgisel optimizasyon algoritmaları Çizelge 1.1.1' de yer verilmiştir.

**Çizelge 1.1.1** Metasezgisel optimizasyon algoritmaları

| **Algoritma Adı** | **Yıl** | **Sınıflandırma** | **Önerildiği Yayın** |
|---|---|---|---|
| Harris Şahinleri Optimizasyon Algoritması | 2019 | Popülasyon tabanlı | [8] |
| İsli Sumru Optimizasyon Algoritması | 2019 | Popülasyon tabanlı | [9] |
| Karadul Optimizasyon Algoritması | 2020 | Popülasyon tabanlı | [10] |
| Aritmetik Optimizasyon Algoritması | 2020 | Popülasyon tabanlı | [11] |
| Afrika Akbabaları Optimizasyon Algoritması | 2021 | Popülasyon tabanlı | [12] |
| Kaya Kartalı Optimizasyon Algoritması | 2022 | Popülasyon tabanlı | [13] |
| Yapay Tavşan Optimizasyon Algoritması | 2022 | Popülasyon tabanlı | [14] |
| Dağ Ceylanı Optimizasyonu | 2022 | Popülasyon tabanlı | [15] |
| Çayır Köpeği Optimizasyon Algoritması | 2022 | Popülasyon tabanlı | [16] |
| Kerevit Optimizasyon Algoritması | 2023 | Popülasyon tabanlı | [17] |

## 1.2 Tezin Bölümleri

İlk bölümde; tezin amaç, kapsam ve katkıları açıklanmıştır. Tez çalışmasında kullanılan yöntemlerden kısa bir şekilde bahsedilmiştir.

Tezin ikinci bölümünde, karşılaştırılan güncel 10 metasezgisel optimizasyon algoritması hakkında detaylı bir literatür çalışması yapılmış, algoritmaların çalışma prensipleri ve sözde kodları ayrıntılı şekilde açıklanmıştır.

Üçüncü bölümde, öncelikle kıyaslama için kullanılacak standart test fonksiyonları açıklanmış, bu çalışmada kullanılan metasezgiselllerin seçilen standart test fonksiyonları ile karşılaştırma sonuçlarını ortaya koyan deneysel çalışmalar ve sonuçları, algoritmalarda yer alan parametrelere dair ayrıntılı bilgiler sunulmuştur. Ayrıca bu bölümde standart test fonksiyonları sonucu ortaya çıkan sonuçların belirgin olup olmadığı ortaya koyabilmek için t test yapılmış ve sonuçları yer almaktadır.

Dördüncü bölümde, seçilen algoritmaların gerçek dünya problemleri üzerindeki başarısını ortaya koyabilmek için seçilen problemlere yer verilmiş, bu algoritmaların seçilen problemler üzerindeki performans sonuçları sunulmuştur. Üçüncü bölümde ifade edildiği gibi algoritmaların gerçek dünya problemleri karşısındaki başarı oranları arasındaki farkın belirgin olup olmadığını daha net ortaya koyabilmek için yine t test tekniği uygulanmış ve sonuçları bu bölümde aktarılmıştır.

Beşinci bölümde, tez çalışmasında elde edilen tüm sonuçlar değerlendirilmiştir. Bununla birlikte ilerideki çalışmalarda yapılabilecekler tartışılmıştır.

# 2. METASEZGİSELLLER

Bu bölümde, karşılaştırma yapılacak metasezgisel optimizasyon algoritmalarına ait literatür çalışması yapılarak algoritmalar hakkında detaylı bilgiye yer verilmiştir

## 2.1 Harris Şahinleri Optimizasyonu (HHO)

HHO, Harris şahinlerinin davranışlarından ve avlanma metotlarından esinlenerek oluşturulmuş sürü tabanlı bir optimizasyon algoritmasıdır [8]. Harris şahinlerinin avı keşfetme, ava sürpriz saldırı ve enerji durumuna göre çeşitli saldırı stratejileri bu algoritmanın keşif ve sömürü aşamalarını kapsamaktadır. Algoritmanın aşamaları aşağıdaki gibidir.

### 2.1.1 Keşif Aşaması

Harris şahinlerinin en büyük fiziksel özelliklerinden birisi, avlarını kısa sürede fark edebilmesini sağlayan görme yetenekleridir. Bu durum her zaman geçerli olmayabilir, Harris şahinleri bu gibi durumlarda av bölgelerinde bekleyip gözlem yapmak zorunda kalırlar. HHO' da keşif aşaması, Harris şahinlerinin avlarını keşfetmek adına gözlem yapma sürecini anlatır. Harris şahinleri rastgele av bölgesine yerleşirler ve avlarını tespit edebilmek için 2 farklı strateji ile beklemeye geçerler. Bu stratejiler Denklem 2.1' de yer alan $q$ değerine göre belirlenir. Harris şahinlerinin her birisi aday çözümü oluştururken, avın konumu ise amaçlanan aday çözümlerin en iyisidir. Keşif aşaması Denklem 2.1' de gösterilmiştir.

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)| & q \geq 0.5 \\ \left(X_{\text{tavşan}}(t) - X_m(t)\right) - r_3 (LB + r_4 (UB - LB)) & q < 0.5 \end{cases} \quad [2.1]$$

Burada eşit olasılıkla $q$ değerine göre stratejiler belirlenir. Denklem 2.1' de, $X(t+1)$ $t$'inci iterasyonundaki şahinin konum vektörü, $X_{\text{tavşan}}(t)$ tavşanın anlık konumu, $X(t)$ şahinlerin anlık pozisyonları, $r_1, r_2, r_3, r_4$ ve $q, 0-1$ arasında belirlenen rastgele sayılardır ve her iterasyonda değişmektedirler. Denklem 2.1' de, $LB$ (lower bound, alt sınır) ve $UB$(upper bound, üst sınır) ise problemin alt sınır ve üst sınır değerleri, $X_m$ ise o anki konumlanan şahinlerin ortalama değeri olmaktadır ve Denklem 2.2' de gösterildiği gibi hesaplanmaktadır.

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \quad [2.2]$$

Burada $X_i(t)$ $t$'inci iterasyondaki şahinin konumunu, $N$ toplam şahin sayısını ifade eder [11].

HHO algoritması, keşiften sömürüye geçiş yapar ve avın kaçma esnasında azalan enerjisine bağlı olarak farklı saldırı davranışları gösterir. Avın kaçışı sırasında önemli ölçüde enerjisi azalır. Azalan avın enerjisi Denklem 2.3'te hesaplanmıştır.

$$E = 2E_0\left(1 - \frac{t}{T}\right) \quad [2.3]$$

Denklem 2.3' de, $E$ avın kaybolan enerjisini, $T$ toplam iterasyon sayısını ve $E_0$ avın başlangıç enerjisini ifade eder. $E_0$ rastgele olarak $(-1,1)$ aralığında her iterasyonda değişmektedir. $E_0$' ın değeri 0 dan $-1$' e düştüğünde av fiziksel olarak zayıf düşerken, $E_0$ 'ın değeri 0' dan 1' e çıktığında, tavşan güçlenmektedir. Avlar, her zaman tehdit edici durumlardan kaçmaya çalışmaktadırlar. Dinamik kaçış enerjisi $E$ iterasyonlar sırasında azalan bir eğilime sahiptir.

$|E| \geq 1$ olduğunda şahinler bir tavşanın yerini keşfetmek için farklı bölgeleri aramaktadır. Dolayısıyla HHO, keşif aşamasını gerçekleştirmektedir ve $|E| < 1$ olduğu zaman, algoritma sömürü aşaması boyunca çözümlerin komşuluğundan yararlanmaya çalışmaktadır [18].

$r$ bir avın başarılı bir şekilde kaçma olasılığı olarak varsayılırsa, sürpriz saldırıdan önce $r < 0.5$ durumunda av başarılı bir şekilde kaçarken; $r \geq 0.5$ durumunda ise, av başarılı bir şekilde kaçamamaktadır. Avın davranışına göre şahinler, avı yakalamak için sert veya yumuşak kuşatma yapacaklardır [8].

### 2.1.2 Sömürü Aşaması

Bir önceki aşamada tespit edilen ava, Harris şahinleri sürpriz saldırı gerçekleştirir. Avın kaçma davranışlarına ve Harris şahinlerinin kovalama stratejilerine göre 4 farklı saldırı türü gerçekleştirilir [19]:

- **Yumuşak Kuşatma**

$r \geq 0.5$ ve $|E| \geq 0.5$ durumunda, tavşanın yeterli enerjisi olduğundan kaçma davranışları göstermekte; fakat bunu başaramamaktadır. Harris şahinleri ise buna

karşın avın etrafını sarar ve avı yorarak sürpriz saldırı davranışlarında bulunurlar. Denklem 2.4' te bu davranış gösterilmiştir.

$$X(t+1) = \Delta X(t) - E\left|JX_{\text{tavşan}}(t) - X(t)\right|$$
$$\Delta X(t) = X_{\text{tavşan}}(t) - X(t) \quad [2.4]$$

$\Delta X(t)$ tavşanın $t$' inci iterasyondaki pozisyon vektörüdür ve o anki pozisyonun farkını ifade etmektedir, $r_5,\ 0-1$ arasında bir rastgele bir sayıdır ve $J = 2(1-r_5)$ olarak ifade edilmektedir. $J$ değeri, her iterasyonda rastgele olarak değişmektedir.

- **Sert Kuşatma**

$r \geq 0.5$ ve $|E| < 0.5$ durumunda, av yorulmuştur ve kaçış enerjisi çok düşüktür. Harris şahinleri, sürpriz saldırıyı gerçekleştirmek için avı kuşatmaktadırlar. Bu durum Denklem 2.5' teki gibidir.

$$X(t+1) = X_{\text{tavşan}}(t) - E|\Delta X(t) \quad [2.5]$$

- **Aşamalı Hızlı Dalışlarla Yumuşak Kuşatma**

$|E| \geq 0.5$ ve $r < 0.5$ durumunda, av kaçmak için için yeterli enerjiye sahiptir. Bu nedenle sürpriz saldırıdan önce, yumuşak bir kuşatma inşa edilmelidir. Yumuşak bir kuşatma gerçekleştimek için, şahinlerin bir sonraki hamleleri Denklem 2.6'da gösterilmiştir.

$$Y = X_{\text{tavşan}}(t) - E\left|JX_{\text{tavşan}}(t) - X(t)\right| \quad [2.6]$$

Avın kaçış hareketlerine karşın iyi bir dalış olup olmayacağını tespit etmek için böyle bir hareketin olası sonucu önceki dalışla karşılaştırılmaktadır. Bu karşılaştırma sonucu Denklem 2.6' da hesaplanan $Y$ başarılı olmaz ise ani bir dalış gerçekleştiririler. Bu ani dalış olan Lévy uçuşu tabanlı dalış, Denklem 2.7'de modellenmiştir.

$$Z = Y + S \times LF(D) \quad [2.7]$$

Denklem 2.7'de, $D$ problemin boyutu, $s$ bir rastsal vekör ve $LF$, Lévy uçuşu fonksiyonudur. Yumuşak kuşatma aşamasında şahinlerin konumlarını güncellemek için Denklem 2.8 kullanılmaktadır.

$$X(t+1) = \begin{cases} Y \text{ eğer } F(Y) < F(X(t)) \\ Z \text{ eğer } F(Z) < F(Xt)) \end{cases} \quad [2.8]$$

Her adımda, sadece daha iyi konum $Y$ veya $Z$ bir sonraki konum olarak seçilmektedir.

- **Aşamalı Hızlı Dalışlarla Sert Kuşatma**

Bu adımda, $|E| < 0{,}5$ ve $r < 0{,}5$ olduğunda tavşanın kaçmak için yeterli enerjisi yoktur ve sürpriz saldırıdan önce avını yakalamak ve öldürmek için sert bir kuşatma inşa edilmektedir. Şahinler, kaçan av ile ortalama konumlarının mesafesini azaltmaya çalışmaktadırlar. Bu durum Denklem 2.9' da modellenmektedir.

$$X(t+1) = \begin{cases} Y \text{ eğer } F(Y) < F(X(t)) \\ Z \text{ eğer } F(Z) < F(Xt)) \end{cases} \quad [2.9]$$

- Bu adımda $Y$ ve $Z$ aşağıdaki yeni kurallara dayanarak elde edilmektedir.

$$Y = X_{\text{tavşan}}(t) - E\left|JX_{\text{tavşan}}(t) - X_m(t)\right| \quad [2.10]$$

$$Z = Y + S \times LF(D) \quad [2.11]$$

HHO algoritması adımları aşağıda Çizelge 2.1.1' de gösterilmiştir [20].

**Çizelge 2.1.1** Harris Şahinleri Optimizasyonu Adımları

| **Harris Şahinleri Optimizasyonu** |
|---|
| **Adım 1.** Başlangıç parametrelerini tanımla |
| **Adım 2.** Başlangıç şahinlerini üret. |
| **Adım 3.** Şahinlerin uygunluk değerini hesapla. |
| **Adım 4.** Algoritma parametrelerini ve şahinlerin konumlarını güncelle. |
| **Adım 6.** Güncel uygunluk değerini hesapla. |
| **Adım 7.** Durdurma kriteri sağlanıyor mu? |
| a. Sağlanıyor: |
| - En iyi şahini çözüm olarak seç. |
| b. Sağlanmıyor: |
| - 4. Adıma geç |

## 2.2 İsli Sumru Optimizasyon Algoritması (STOA)

STOA, isli sumruların göç ve saldırı davranışlarını simüle eder [9]. Bu algoritmanın ana ilham kaynağı deniz kuşlarından biri olan isli sumruların göç ve saldırı davranışlarıdır. Bu algoritmada göç davranışı keşif aşamasını; saldırı davranışı ise sömürü (kullanım) aşamasını ifade eder.

### 2.2.1 Keşif Aşaması

Göç sırasında isli sumruların gösterdiği davranışlar algoritmada aşağıdaki gibi modellenmiştir:

- Çarpışmadan kaçınma: Göç sırasında isli sumrular arasındaki çarpışmayı önlemek için yeni isli sumrunun konumu Denklem 2.12. ile hesaplanır.

$$\vec{C}_{st} = S_A \times \vec{P}_{st}(z) \quad [2.12]$$

$\vec{C}_{st}$ Diğer isli sumrular ile çarpışmayan kuşun konumunu, $\vec{P}_{st}(z)$ isli sumrunun geçerli konumunu ve $S_A$ mevcut yinelemeyi gösterir ve Denklem 2.13 ile aşağıdaki gibi hesaplanır.

$$S_A = C_f - \left(z \times \left(C_f / M\text{aksimum}_{\text{iterasyon}}\right)\right) \quad [2.13]$$

Burada $z$ değeri Denklem 2.14 ile aşağıdaki gibi hesaplanır.

$$z = 0,1,2,\ldots, M\text{aksimum}_{\text{iterasyon}} \quad [2.14]$$

$C_f$, $S_A$'yı ayarlamak için kontrol değişkenidir.

- En iyi komşu yönüne yakınlaşma: Çarpışmadan kaçındıktan sonra isli sumrular, en iyi komşunun yönüne doğru hareket ederler. Hareket, Denklem 2.15 ile hesaplanmaktadır.

$$\vec{M}_{st} = C_B \cdot \vec{P}_{st}(z)\left(\vec{P}_{bst}(z) - \vec{P}_{st}(z)\right) \quad [2.15]$$

Burada $\vec{M}_{st}$ İsli sumrunun ulaştığı farklı konumlarını ifade eder.$\vec{P}_{bst}(z)$ İsli sumrunun en iyi konumunu verir. $\vec{P}_{st}(z)$ İsli sumrunun geçerli konumudur. $C_B$ daha iyi bir keşiften sorumlu bir değişkeni ifade eder ve aşağıdaki şekilde hesaplanır.

$$C_B = 0.5 \times R_{\text{and}} \quad [2.16]$$

Burada $R_{\text{and}}$ [0, 1] aralığında yer alan rastgele bir sayıdır.

- En iyi isli sumru konumuna göre güncelleme: En sonunda arama temsilcisi veya isli sumru, en iyi konuma sahip isli sumruya göre konumunu güncelleyebilir. Denklem 2.17' de bu güncelleme hesaplanmıştır.

$$\vec{D}_{st} = \vec{C}_{st} + \vec{M}_{st} \quad [2.17]$$

$\vec{D}_{st}$ En iyi konuma sahip isli sumru ile isli sumrunun o anki konumu arasındaki farkı ifade eder.

### 2.2.2 Sömürü Aşaması

Sömürü aşaması, isli sumruların göç sırasındaki hız ve saldırı açılarını değiştirebilme davranışlarını kapsamaktadır. İsli sumrular bu değişikliği kanatlarını hareket ettirerek irtifa kazanırlar. Avlarına saldırırken havada aşağıda spiral davranışlar sergilerler [21]. Bu süreç aşağıdaki denklemler ile gösterilmiştir.

$$x' = R_{\text{adius}} \times \sin(i) \quad [2.18]$$

$$y' = R_{\text{adius}} \times \cos(i) \quad [2.19]$$

$$z' = R_{\text{adius}} \times i \quad [2.20]$$

$$R_{\text{adius}} = u \times e^{kv} \quad [2.21]$$

$R_{\text{adius}}$ spiralin her dönüşünün yarıçapını, $i$, $[0 \leq k \leq 2\pi]$ aralığı arasında yer alan değişkeni, $u$ ve $v$ spiral şeklini tanımlayan sabitleri, $e$ doğal logaritmanın tabanıdır. Denklem 2.21'deki $u$ ve $v$ sabitlerinin değeri 1 olarak alınmıştır [21]. İsli sumrunun konum güncellemesi Denklem 2.22 ile hesaplanmaktadır.

$$\vec{P}_{st}(Z) = \left(\vec{D}_{st} \times (x' + y' + z')\right) \times \vec{P}_{bst}(Z) \quad [2.22]$$

Burada $\vec{P}_{bst}(z)$ İsli sumrunun en iyi konumunu, $\vec{P}_{st}(z)$ İsli sumrunun geçerli konumu iken $\vec{D}_{st}$ ise en iyi konuma sahip isli sumru ile isli sumrunun o anki konumu arasındaki farktır. $x', y'$ ve $z'$, göç sırasındaki isli sumruların değişen açılarıdır. STOA algoritması adımları aşağıda Çizelge 2.2.1 'de gösterilmiştir [9].

**Çizelge 2.2.1** İsli Sumru Optimizasyon Algoritması Adımları

| İsli Sumru Optimizasyon Algoritması |
|---|
| **Girdi**: İsli Sumru nufusu |
| **Çıktı**: En iyi isli sumru konumu |
| **Adım 1.** STOA prosedürünü başlat |
| **Adım 2.** $\boldsymbol{S_A}$ ve $\boldsymbol{C_B}$ parametrelerini başlat |
| **Adım 3.** Her bir isli sumrunun konumunu hesapla |
| **Adım 4.** En iyi isli sumru konumu |
| **Adım 5.** **while** ($\boldsymbol{z < Max_{iterations}}$) **do** |
| **Adım 6.** **for** her bir isli sumru için **do** |
| **Adım 7.** İsli Sumruların konumunu güncelle |
| **Adım 8.** **end for** |
| **Adım 9.** $\boldsymbol{S_A}$ ve $\boldsymbol{C_B}$ parametrelerini güncelleyin |
| **Adım 10.** Her bir isli sumrunun uygunluk değerini güncelleyin |
| **Adım 11.** En iyi isli sumru konumundan daha iyi bir konuma sahip isli sumru varsa konum güncellemesi yapın |
| **Adım 12.** $\boldsymbol{z = z + 1}$ |
| **Adım 13.** **end while** |
| **Adım 14.** En iyi isli konumunu al |
| **Adım 15.** Prosedürü bitir. |

## 2.3 Karadul Optimizasyonu (BWO)

BWO, Karadul örümceklerinin çiftleşme davranışlarından ilham alınarak geliştirilmiş popülasyon tabanlı bir algoritmadır [10].

Algoritma, genel anlamda karadul örümceklerine ait olan üreme ve sonrasındaki yamyamlık davranışı olan Darwin' in evrim teorisinin yani en uygun olanın hayatta kalması ve en güçlü olanın üstünlüğü düşüncelerini yansıtır. Özel anlamda ise örümcek popülasyonundaki mutasyonu; ayrıntı boyutunda ise genetik değişiklikleri ifade eder [22]. BWO; popülasyon başlatma, üreme, yamyamlık ve mutasyon olmak üzere dört aşamadan meydana gelmektedir.

### 2.3.1 Popülasyonun başlatılması

Karadul optimizasyon algoritmasında karadul örümceklerinin her birisi çözüme dair potansiyel bir çözümü ifade eder. $D$ boyutlu bir problemde karadullar Denklem 2.23' te gösterilmiştir.

$$\text{Örümcek } = [x_1, x_2, \dots, x_{\mathrm{N_{var}}}] \quad [2.23]$$

Burada $x_1$ problemdeki her bir karadulu ifade ederken, $x_{\mathrm{N_{var}}}$ problem boyudur. Dizi şeklinde gösterilen her bir karadulun uygunluk değeri Denklem 2.24 ile hesaplanmaktadır.

$$\text{Fitness } = f(\text{örümcek}) = f\left(x_1, x_2, \dots, x_{\mathrm{N_{var}}}\right) \quad [2.24]$$

Popülasyonu başlatmak için $\mathrm{N_{pop}} \times \mathrm{N_{var}}$ boyutlarında bir matris oluşturulur. Daha sonra üreme aşaması için çiftler rastgele seçilerek bir sonraki aşamaya geçilir.

### 2.3.2 Üreme

Yeni oluşturulan nesil, karadulların kendilerine özgü olan çiftleşme yöntemi ile oluşturulmaktadır. Algoritmada, yeni nesil oluşturmak için $0-1$ arasında rastgele değerler alabilen $\alpha$ adında $D$ boyutlu bir dizi oluşturulur. Bu dizideki değerler karadullara karşılık gelmektedir. Yeni nesil örümcekler aşağıdaki Denklem 2.25 ile oluşturulmaktadır.

$$\begin{aligned} y_1 &= \alpha \times x_1 + (1-\alpha) \times x_2 \\ y_2 &= \alpha \times x_2 + (1-\alpha) \times x_1 \end{aligned} \quad [2.25]$$

Denklem 2.25' te yer alan $Y_1$ ve $Y_2$ üreme sonucu gelen genç örümcekleri, $x_1$ ve $x_2$ ise anne ve baba örümcekleri temsil eder. Yukarıdaki işlem $D/2$ kez tekrar eder ve $\alpha$ dizisi elemanlarından her adımda farklı seçim yapılarak hesaplama yapılır.

### 2.3.3 Yamyamlık

3 çeşit yamyamlık türü vardır [10]. Bunlar:

- **Cinsel yamyamlık:** Dişi karadulun çiftleşme sırasında veya sonrasında erkek örümceği yemesi. Bu yamyamlık dişi ve erkek karadulun uygunluk değerine göre uygulanır.

- **Kardeş yamyamlık:** Güçlü örümcek yavrularının zayıf kardeşlerini yediği yamyamlık türüdür. Uygunluk değeri yüksek genç örümcekler popülasyonda kalır ve diğerleri popülasyondan atılır.
- **Yavru ve anne arasındaki yamyamlık:** Bazı güçlü yavru örümcekler annelerini yerler ve yeni nesle geçerler.

Bütün bu süreçte, güçlü veya zayıf örümcekleri belirlemek için uygunluk değerlerinden yararlanılmaktadır.

### 2.3.4 Mutasyon

Algoritmada mutasyon, belirlenen sayıda genç örümceğin rastgele olarak popülasyondan seçilmesiyle başlar. Her bir seçilen örümceğin iki elementi rastgele olarak yer değiştirmektedir. BWO algoritması adımları aşağıda çizelgede gösterilmiştir [20].

**Çizelge 2.3.1** Karadul Optimizasyonu Adımları

| Karadul Optimizasyonu |
|---|
| **Adım 1**. Başlangıç parametrelerini tanımla |
| **Adım 2**. Başlangıç örümcek popülasyonunu üret. |
| **Adım 3**. Popülasyonun uygunluk değerini hesapla. |
| **Adım 4**. Durdurma kriteri sağlanıyor mu? |
| a. Sağlanıyor: |
| - En iyi örümceği çözüm olarak seç. |
| b. Sağlanmıyor: |
| - Popülasyonu güncelle. |
| - 2. Adıma geç |

## 2.4 Aritmetik Optimizasyon Algoritması (AOA)

AOA, matematiksel işlemlerden dört işlem olarak adlandırılan çarpma, bölme, çıkarma ve toplama işlemlerinin kullanım durumlarını modelleyen popülasyon tabanlı bir optimizasyon algoritmasıdır [11]. Ayrıca, optimizasyon problemlerini türev hesabı yapmadan çözer [23]. AOA iki temel bölümden meydana gelir. Bunlar keşif ve sömürü

(kullanım) aşamalarıdır. Keşif aşamasında, aday çözümlerin konumları çarpma ve bölme işlemlerinde kullanılırken, sömürü aşamasında toplama ve çıkarma işlemleri yapılır. AOA' nın matematiksel modeli şu şekildedir:

### 2.4.1 Başlatma Aşaması

AOA' da optimizasyon süreci, rastgele oluşturulmuş bir dizi aday çözüm ile başlar ve her yinelemede en iyi aday çözüm, şimdiye kadar elde edilen en iyi çözüm kabul edilir. Denklem 2.47 ile aday çözüm dizisi $(X)$ gösterilir.

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad [2.47]$$

AOA' da arama sürecinden önce hangi aşamada çalışacağı belirlenmelidir. Bu belirleme Denklem 2.48 ile hesaplanmaktadır.

$$\text{MOA}\left(C_{\text{iterasyon}}\right) = \text{Min} + C_{\text{iterasyon}} \times \left(\frac{\text{maks} - \text{min}}{M_{\text{iterasyon}}}\right) \quad [2.48]$$

MOA **(Hızlandırılmış Matematik Optime Edici)** çalışılacak aşamayı gösteren denklemdir. $\text{MOA}\left(C_{\text{iterasyon}}\right)$ $C_{\text{iterasyon}}$. deki hesaplanan fonksiyon değeridir. $C_{\text{iterasyon}}$ geçerli yinelemeyi, $M_{\text{iterasyon}}$ ise maks yinelemeyi ifade eder. Min ve Maks fonksiyonun alabileceği en küçük ve en büyük değerleri tanımlar.

### 2.4.2 Keşif Aşaması

Keşif aşamasında, aritmetik operatörlerden bölme (D) ve çarpma (M) operatörü, keşif aşamasını yüksek dağılımlarından dolayı kullanılır [11]. AOA' da arama alanı birkaç bölgede rastgele keşfedilir ve Denklem 2.49' da modellenen iki ana arama stratejisine bağlı olarak daha iyi bir çözüm bulmaya çalışır.

$$x_{i,j}\left(C_{\text{iterasyon}} + 1\right) = \begin{cases} \text{en iyi}\left(x_j\right) \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), \text{için } r_2 > 0.5, \\ en\ iyi\left(x_j\right) \div (MOP + \varepsilon) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), \text{için } r_2 < 0.5, \end{cases} \quad [2.49]$$

$x_i(C_{\text{iterasyon}}+1)$ sonraki yinelemedeki çözümü gösterir. $x_{i,j}(C_{\text{iterasyon}})$ geçerli yinelemedeki $i$'nci çözümün $j$'nci konumunu verir. $\varepsilon$ pozitif bir tam sayı olup adım boyunu ifade eder. $UB_j$ ve $LB_j$ $j$'nci pozisyonun alt ve üst sınırlarıdır.

$$\text{MOP}\left(C_{\text{iterasyon}}\right) = 1 - \left(\frac{C_{\text{iterasyon}}}{M_{\text{iterasyon}}}\right)^{1/\alpha} \quad [2.50]$$

$\text{MOP}\left(C_{\text{iterasyon}}\right)$ mevcut iterasyondaki işlev değerini, $\alpha$ iterasyondaki doğruluğu ifade eder 5 olarak kabul edilmiştir [24].

### 2.4.3 Sömürü (Kullanım) Aşaması

Aritmetik operatörlerden toplama (A) ve çıkarma (S) düşük dağılımlı olduklarından dolayı (A) veya (S) hedefe daha kolay yaklaşırlar. Bundan dolayı hedefe kolay yaklaştıkları için sömürü aşamasında kullanılırlar. (A) ve (S) arama alanını birkaç yoğun bölgede Denklem 2.51' deki gibi derinlemesine araştırır.

$$x_{i,j}(C_{\text{Iter}}+1) = \begin{cases} \text{best}\left(x_j\right) - (MOP) \times \left(\left(\text{U}B_j - \text{L}B_j\right) \times \mu + \text{L}B_j\right), r3 < 0.5 \\ \text{best}\left(x_j\right) + MOP \times \left(\left(\text{U}B_j - \text{L}B_j\right) \times \mu + \text{L}B_j\right), \text{ diğer} \end{cases} \quad [2.51]$$

AOA algoritmasının sözde kodu Çizelge 2.4.1 'de verilmiştir [11].

**Çizelge 2.4.1** Aritmetik Optimizasyon Algoritması Sözde Kodu

| Aritmetik Optimizasyon Algoritması | |
|---|---|
| **Adım 1.** | **Giriş**: $\alpha, \mu$ $ve$ $M_{\text{Iter}}$ |
| **Adım 2.** | Başlangıç değeri için $(X_i i = 1,2, \dots, N)$ |
| **Adım 3.** | **While** ($C_{\text{Iter}} < M_{\text{Iter}}$ ) **do** |
| **Adım 4.** | Her bir temsilci için uygunluk değeri hesaplayın |
| **Adım 5.** | En iyi temsilciyi best $(x)$ olarak belirleyin |
| **Adım 6.** | MOA ve $MOP$ denklemlerini sırasıyla güncelleyin |
| **Adım 7.** | **for** $i = 1$ $to$ $N$ **do** |
| **Adım 8.** | **for** $j = 1$ $to$ $Dim$ **do** |
| **Adım 9.** | $r_1, r_2$ ve $r_3$değerlerini güncelleyin |
| **Adım 10.** | **if** ($r_1 >$ MOA ) **then** |
| **Adım 11.** | **Keşif Aşaması** |
| **Adım 12.** | $X_i$'yi güncellemek için Denklem 2.51'i kullanın |
| **Adım 13.** | **else** |
| **Adım 14.** | **Sömürü Aşaması** |
| **Adım 15.** | $X_i$'yi güncellemek için Denklem 2.49'u kullanın |
| **Adım 20.** | **end if** |
| **Adım 21.** | **end for** |
| **Adım 22.** | **end for** |
| **Adım 23.** | $C_{\text{Iter}} = C_{\text{Iter}} + 1$ |
| **Adım 24.** | **end while** |
| **Adım 25.** | $X_{best}$' alın |

## 2.5 Afrika Akbabaları Optimizasyon Algoritması (AVOA)

AVOA, Afrika akbabalarının yiyecek arama davranışları ve sosyal davranışlarından ilham alan doğa tabanlı bir metasezgiseldir [12]. Afrika akbabalarının bu davranışları matematiksel olarak modellenirken aşağıdaki temel noktalara göre şekillenmiştir [25].

- Probleme bağlı olarak belirlenecek olan ortamdaki akbaba sayısı (N), popülasyon büyüklüğü ile eşittir [25].
- Akbabalar fiziksel olarak iki kategoriye ayrılır. Akbabalar kategorilere ayrılmadan önce her birinin uygunluk değeri hesaplanır. Uygunluk değeri en

yüksek olan en uygun tepkiyi temsil için; ikinci en yüksek uygunluk değerine sahip olan ise ikinci en iyi akbabayı temsil ediyor. Her adımda en üstteki iki akbabadan biri başka akbaba popülasyonuyla değiştirilir veya değiştirilir [26].

- Akbabalarda yiyecek arama davranışı popülasyonun ortak hareketi sonucu gerçekleşir. Bunun nedeni, bir popülasyondaki her bir akbabanın yiyecek avlama ve yakalama becerilerinin farklı olmasıdır.[25]
- AVOA' da uygunluk değerlerine göre grup üyeleri uygunluk değeri en yüksek akbabaya yaklaşırken; uygunluk değeri en düşük olan akbabadan ise uzaklaşırlar. [25]

AVOA' nın matematiksel modeli aşağıdaki adımlar ile gösterilmiştir.

### 2.5.1 En İyi Akbaba Seçimi

Başlangıç popülasyonu oluşturulduktan sonra akbabaların uygunluk değeri hesaplanır. En yüksek uygunluk değerine sahip akbaba, birinci grubun en iyi akbabası; ikinci en yüksek uygunluk değerine sahip akbaba ise, ikinci grubun en iyi akbabası olarak belirlenir. Her bir adımda bu seçim tekrarlanır. Bu durum Denklem 2.52 ile aşağıdaki gibi gösterilmiştir [12].

$$R(i) = \begin{cases} \text{En iyi akbaba}_{\,1} \text{ eğer } p_i = L_1 \\ \text{En iyi akbaba}_{\,2} \text{ eğer } p_i = L_2 \end{cases} \quad [2.52]$$

$L_1$ ve $L_2$, 0-1 arasındaki parametrelerdir ve $L_1 + L_2 = 1$ dir. Denklem 2.53 kullanılarak her grup için en iyi çözümlerin, $p_i$ seçme olasılığı her birinin seçilmesi amacıyla rulet çarkı kullanılarak elde edilir.

$$p_i = \frac{F_i}{\sum_{i=1}^{n} F_i} \quad [2.53]$$

$F_i$ , $i$ . akbabanın uygunluk değerini, $n$ ise toplam akbaba sayısını ifade eder.

### 2.5.2 Akbabaların Açlık Oranı

Akbabalar yüksek enerjiye sahip olduklarında genellikle yiyecek arama davranışı gösterirler. Bu durum onların yiyecek aramak için daha uzun mesafeler gidebilmelerini sağlar; fakat enerjileri olmadığında uzun mesafelere gidemezler. Akbabaların yeterli enerjiye sahip olmadığında kendilerinden güçlü olan akbabaları takip ettikleri durumu, Denklem 2.54 ile aşağıdaki gibi gösterilmiştir.[12]

$$F = (2 \times \text{ rand }_1 + 1) \times z \times \left(1 - \frac{\text{iterasyon}_i}{maksimum_{\text{ iterasyon}}}\right) + t \quad [2.54]$$

$F$ akbabaların doyduğunu, iteration $_i$, geçerli yinelemeyi, max $_{\text{iterasyon}}$ ve $z$ yinelemeri değiştiren -1 ile +1 arasındaki rastgele bir değerdir. Rand ise 0 ile 1 arasındaki rastgele bir sayıdır. Denklem 2.54' te yer alan $t$ aşağıda yer alan Denklem 2.55 ile hesaplanmaktadır.

$$t = h \times \left(\sin^w \left(\frac{\pi}{2} \times \frac{\text{iterasyon}_i}{maksimum_{\text{ iterasyon}}}\right) + \cos\left(\frac{\pi}{2} \times \frac{\text{iterasyon}_i}{maksimum_{\text{ iterasyon}}}\right) - 1\right) \quad [2.55]$$

$h$, -2 ile +2 arasındaki rastgele bir sayıdır. $w$ , sabit bir sayı olup keşif ve operasyon aşamaları hakkında bilgi veren bir parametredir. $w$ değeri arttıkça keşif aşamasına girme olasılığı artarken; azalması durumunda ise arama aşamasına girme olasılığı azalır [12].

### 2.5.3 Keşif Aşaması

Denklem 2.54' te hesaplanan $F$ değerinin 1' den büyük olduğu durumda akbabalar keşif aşamasındadırlar. Bu aşamadaki akbabaların görme yetenekleri, yiyecek tespit edebilme ve ölmekte olan hayvanları fark edebilme yetenekleri oldukça başarılıdır. Akbabalar uzun süre yiyecek bulamadıklarında uzun mesafeler alabilirler. Algoritma başlatıldıktan sonra keşif aşamasında, alan taraması yapmak için öncelikle 0 ve 1 arasında değer alan $p_1$ parametresini göre belirlerler. Daha sonra kullanacakları iki farklı keşif yöntemini seçebilmek için [0,1] aralığında değer alan $rand_{p1}$ değeri seçilir [27]. Kullanılacak keşif yöntemini belirleme denklemleri aşağıdaki gibidir [12].

$$P(i+1) = \begin{cases} \text{Denklem(2.57) eğer } P_1 \geq \text{ rand }_{P1} \\ \text{Denklem(2.59) eğer } P_1 < \text{rand}_{P1} \end{cases} \quad [2.56]$$

$$P(i+1) = R(i) - D(i) \times F \quad [2.57]$$

$$D(i) = |X \times R(i) - P(i)| \quad [2.58]$$

$$P(i+1) = R(i) - F + \text{ rand }_2 \times ((ub - lb) \times \text{ rand }_3 + lb) \quad [2.59]$$

Burada $P(i+1)$, bir sonraki yinelemede konum vektörüdür. $F$ , akbaba doyurulma oranını, $R(i)$ seçilen en iyi akbabalardan birisini, $X$ ise akbabaların yiyeceklerini diğer akbabalardan korumak için hareket ettikleri konumu ifade eder. $ub - lb$, üst sınır ve alt sınırları tanımlar. $D(i)$ ise en iyi akbaba ile mevcut akbaba arasındaki konumu ifade eder. rand $_1$, rand $_2$ ve rand $_3$ rastegele sayılardır

### 2.5.4 Sömürü Aşaması

Denklem 2.54' te hesaplanan $F$ değerinin 1' den küçük olduğu durumda akbabalar sömürü aşamasındadırlar. AVOA' da sömürü aşaması iki aşamadan oluştur ve her bir aşama için iki farklı strateji kullanılır [26]. Sömürü aşamasında yer alan iki aşamada kullanılan stratejiler için $p_2$ ve $p_3$ parametreleri kullanılır. $p_2$, birinci aşamadaki strateji seçimi için, $p_3$ ise ikinci aşamadaki strateji seçimi için kullanılır [12]. $F$ değeri 1 ile 0,5 arasında olduğunda sömürüde ilk aşama gerçekleşir. Sömürünün ilk aşamasında akbabalar dönüşümlü uçuş ve kuşatma savaşı stratejilerini uygularlar. $p_2$ parametresinin değeri bu aşama öncesinde tanımlı hale gelir, daha sonra rand $_2$ üretilir. Denklem 2.60 ile dönel uçuş ya da kuşatma stratejileri belirlenir.

$$P(i+1) = \begin{cases} \text{Denklem (2.61) eğer } P_2 \geq \text{rand}_{P2} \\ \text{Denklem (2.65) eğer } P_2 < \text{rand}_{P2} \end{cases} \quad [2.60]$$

$F$, değerinin $1 - 0{,}5$ arasında olduğu durumda akbaba tok ve yeterli enerjiye sahiptir. Akbabaların bu şekilde bir besin kaynağına yönelmeleri kendi aralarında çatışmalara neden olur. Bu davranış matematiksel olarak Denklem 2.61 ile gösterilmiştir.

$$P(i+1) = D(i) \times (F + rand_4) - d(t) \quad [2.61]$$

$$d(t) = R(i) - P(i) \quad [2.62]$$

$rand_4$, 0 ve 1 arasındaki rastgele bir sayıdır ve rastgele katsayısını artırmak için kullanılır.

Akbabalar dönel uçuş stratejileri ile dönme hareketi yaparlar. Bu hareket spiral model ile gösterilmiştir. Bu yöntemde tüm akbabalar ile en iyi iki akbabadan biri arasında sarmal bir denklem yaratılır. Bu denklemler Denklem 2.63, Denklem 2.64 ve Denklem 2.65 ile aşağıdaki gibi gösterilmiştir.

$$S_1 = R(i) \times \left(\frac{rand_5 \times P(i)}{2\pi}\right) \times \cos\left(P(i)\right) \quad [2.63]$$

$$S_2 = R(i) \times \left(\frac{\text{rand}_6 \times P(i)}{2\pi}\right) \times \sin\left(P(i)\right) \quad [2.64]$$

$$P(i+1) = R(i) - (S_1 - S_2) \quad [2.65]$$

$Sin$ ve $cos$ parametreleri sinüs ve kosinüs fonksiyonlarına ait temsilleridir. $\text{rand}_5$ ve $\text{rand}_6$ 0 ve 1 arasında rastgele değerler alan parametrelerdir. Akbabaların konumu denklem 2.65 ile güncellenir.

$F$ değerinin 0,5' ten daha küçük olduğu durumda sömürünün ikinci aşaması gerçekleştirilir. Bu aşamada, iki akbabanın hareketleri diğer akbabaların yiyecek kaynağı üzerinde birikmelerine sebep olur ve bu durum rekabetin yanında çatışmalara yol açar. Aşama öncesinde $\text{rand}_{p3}$ olarak tanımlanan 0 ve 1 arasında değer alan parameter üretilir. Eğer $\text{rand}_{p3}$, $p_3$ değerinden büyük veya ona eşitse besin kaynağı etrafında akbabaların toplanması, eğer üretilen değer $p_3$ parametresinden küçükse agresif kuşatma savaşı uygulaması yapılır [12]. Denklem 2.66 ile gösterilmiştir.

$$P(i+1) = \begin{cases} \text{Denklem 2.68} & \text{eğer } \text{P}_3 \geq \text{rand}_{P3} \\ \text{Denklem 2.69} & \text{eğer } \text{P}_3 < \text{rand}_{P3} \end{cases} \quad [2.65]$$

Akbabaların beslenme kaynağına doğru tüm hareketleri gözlemlenir. Akbabaların yiyecek bulmada zorluk çektiği dönemlerde, akbabalar besin kaynağı için yoğun bir rekabete yol açar ve bu durum farklı ağababa türlerinin aynı besin kaynağında toplanması ile sonuçlanır. Bu durum Denklem 2.66, Denklem 2.67 ve 2.68 ile aşağıdaki gibi gösterilmiştir.

$$A_1 = \text{ En iyi akbaba }_1(i) - \frac{\text{En iyi akbaba }_1(i) \times P(i)}{\text{En iyi akbaba }_1(i) - P(i)^2} \times F \quad [2.66]$$

$$A_2 = \text{ En iyi akbaba }_2(i) - \frac{\text{En iyi akbaba }_2(i) \times P(i)}{\text{En iyi akbaba }_2(i) - P(i)^2} \times F \quad [2.67]$$

$\text{En iyi akbaba}_1(i)$, mevcut yinelemede birinci gruptaki en uygun 1. akbabayı, $\text{En iyi akbaba}_2(i)$ ise ikinci gruptaki en uygun 1. Akbabayı temsil eder. $P(i)$ akbabanın mevcut pozisyonunu tanımlar. Tüm akbabaları birleştirme işlem Denklem 2.68 ile aşağıdaki gibidir.

$$P(i+1) = \frac{A_1 + A_2}{2} \quad [2.68]$$

Gruplardaki baş akbabaların yeterli beslenmeleri sonucunda zayıf düşerler ve diğer akbabalarla mücadele için yeterince enerjiye sahip olamazlar. Diğer akbabalar da beslenemediklerinden saldırganlaşırlar. Bu durum Denklem 2.69 ile tanımlanmıştır.

$$P(i+1) = R(i) - |d(t)| \times F \times \text{Levy}\,(d) \quad [2.69]$$

$d(t)$, akbabanın gruplardaki en iyi akbabaya olan mesafeyi tanımlar. Levy ise Denklem 2.70 ile aşağıdaki gibi hesaplanmaktadır.

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left( \frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma(1+\beta 2) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad [2.70]$$

$d$ problem boyutunu, $u$ ve v 0 ve 1 arasında değer alabilen rastgele değişkenlerdir. $\beta$ ise değeri 1,5 olan bir sabittir [12]. AVOA algoritmasının sözde kodu Çizelge 2.5.1' de verilmiştir [26].

**Çizelge 2.5.1** Afrika Akbabaları Optimizasyon Algoritması Sözde Kodu

| Afrika Akbabaları Optimizasyon Algoritması | |
|---|---|
| **Adım 1.** | **Giriş**: Popülasyon büyüklüğü $N$ ve maksimum yineleme sayısı $T$ |
| **Adım 2.** | Akbabaların konumu ve uygunluk değeri |
| **Adım 3.** | $(P_i i = 1,2, \dots, N)$ popülasyonu rastgele başlatın |
| **Adım 4.** | **While** (durma koşulu tamamlanmıyor) **do** |
| **Adım 5.** | Akbabanın uygunluk değerini hesaplayın |
| **Adım 6.** | **Konum ayarla** Akbaba $_1$ (1. gruptaki en iyi akbabanın) |
| **Adım 7.** | **Konum ayarla** Akbaba $_2$ (2. gruptaki en iyi akbabanın) |
| **Adım 8.** | **for (**Her akbaba için $P(i)$ yi hesapla **) do** |
| **Adım 9.** | Denklem 2.52' yi kullanarak $R(i)$' yi seçin |
| **Adım 10.** | Denklem 2.54' ü kullanarak $F$' yi güncelleyin |
| **Adım 11.** | **if(** $\|F\| \geq 1$) **then** |
| **Adım 12.** | **if(** $p_1 \geq rand_{p1}$) **then** |
| **Adım 13.** | Denklem 2.57 ile akbabanın konumunu güncelle |
| **Adım 14.** | **else** |
| **Adım 15.** | Denklem 2.59 ile akbabanın konumunu güncelle |
| **Adım 20.** | **if(** $\|F\| <1$) **then** |
| **Adım 21.** | **if(** $\|F\| \geq 0.5$) **then** |
| **Adım 22.** | **if(**$p_2 \geq rand_{p2}$)) **then** |
| **Adım 23.** | Denklem 2.61 ile akbabanın konumunu güncelle |
| **Adım 24.** | **else** |
| **Adım 25.** | Denklem 2.65 ile akbabanın konumunu güncelle |
| **Adım 26.** | **else** |
| **Adım 27.** | **if(**$p_3 \geq rand_{p3}$)) **then** |
| **Adım 28.** | Denklem 2.68 ile akbabanın konumunu güncelle |
| **Adım 29.** | **else** |
| **Adım 30.** | Denklem 2.69 ile akbabanın konumunu güncelle |
| **Adım 31.** | **P** Akbaba $_1$'i getir |

## 2.6 Kaya Kartalı Optimizasyon Algoritması (AO)

AO, kaya kartallarının avlanma ve avlarını yakalama becerilerinden ilham alan popülasyon tabanlı bir optimizasyon algoritmasıdır [13]. Kaya kartalları tarafından kullanılan avlanma ve av yakalama yöntemleri şu şekildedir:

- Birinci yöntem, dikey eğimli yüksek uçma, kaya kartallarının yükseklerde uçuş esnasındaki avlanma yöntemidir. Kaya kartalı, av bulduklarında, ava doğru dikey bir dalış gerçekleştirir [28].
- İkinci yöntem, kısa süzülme saldırısıyla kontur uçuşu, kaya kartallarının yerden düşük bir seviyedeki uçuşu esnasındaki avlanma yöntemidir. En sık olarak tercih edilen avlanma yöntemidir. Orman tavuğu, yer sincapları ve deniz kuşları avı için bu yöntem özellikle tercih edilmektedir [29].
- Üçüncü yöntem, yavaş alçalma saldırısı ile alçak uçuş, kaya kartallarının yere inerek ve sonrasında avına saldırarak avlanmasıdır. Kaya kartalları bu yöntem ile avını boynundan ve sırtından yakalayarak avlar. Daha çok yavaş hareket eden avlar için kullanılan bir yöntemdir [30].
- Dördüncü yöntem, Kaya kartalının yürüyerek avını yakalaması, Kaya kartalları av için yere inerek avı yakalar. Bu yöntem daha çok büyük avların yavrularını avlamak için tercih edilir [13].

Kaya kartalı optimizasyonunda keşif ve sömürü aşaması aşağıda yer alan Denklem 2.62' deki şarta bağlı olarak seçilir [13].

$$t \leq \left(\frac{2}{3}\right) * T \qquad [2.62]$$

$t$ anlık iterasyonu, $T$ ise maksimumum iterasyon sayısını tanımlar. $t \leq \left(\frac{2}{3}\right) * T$ koşulu sağlandığı durumda keşif aşaması, sağlanmadığı durumda ise algoritması sömürü aşamasındadır.

### 2.6.1 Başlatma

AOA, Denklem 2.63' te gösterildiği gibi rastgele oluşturulan bir aday çözüm havuzu ile başlatılır [28].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D-1} & x_{1,D} \\ x_{2,1} & \ddots & \cdots & \ddots & x_{2,D} \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ x_{N-1,1} & \ddots & \cdots & \ddots & x_{N-1,D} \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D-1} & x_{N,D} \end{bmatrix}_{N \times D} \quad [2.63]$$

$X$ her bir bireyi, $N$ aday çözüm sayısını, $D$ ise problem boyutunu ifade etmektedir. Denklem 2.63' te yer alan her bir aday çözüm aşağıda yer alan Denklem 2.64 ile ortaya çıkarılmaktadır.

$$X_{ij} = \text{rand} * \left(UB_j - LB_j\right) + LB_j, i = 1,2, \ldots, Nj = 1,2, \ldots, D \quad [2.64]$$

$X_i$ $i$. çözümün değerini, $LB_j$ $j$. alt sınırı, $UB_j$ $j$. üst sınırı ve rand ise rastgele bir sayıyı tanımlar.

### 2.6.2 Genişletilmiş keşif

Kaya kartalları, av için bölgesini yüksekten uçarak tanır ve en iyi avlanma alanını seçer. Burada geniş çapta keşifler yapılır. Av bulunduğu zaman dikey bir dalış gerçekleştirir. Bu hareket Denklem 2.65 ile gösterilmiştir [30].

$$X_1(t+1) = X_{\text{en iyi}}(t) \times \left(\frac{1-t}{T}\right) + \left(X_M(t) - X_{\text{en iyi}}(t) \times \text{ rand }\right) \quad [2.65]$$

$X_1(t+1)$ $t$'den bir sonraki adım çözümünü, $X_1$ ilk aday çözümü, $X_{\text{en iyi}}(t)$ $t$. adımdaki en iyi aday çözümü, $\left(\frac{1-t}{T}\right)$ ise keşif kontrolü tanımlar. $X_M(t)$ $t$. adımdaki çözümlerin ortalamasıdır. Rand 0 ile 1 arasındaki rastgele bir sayıdır.

### 2.6.3 Daraltılmış keşif

Kaya kartalları yüksek uçuş ile av bölgesini belirlediklerinde, kaya kartalı, hedefin üzerinde daireler çizerek inmeye hazırlanır ve ardından avına saldırır. Bu kaya kartallarının 2.avlanma yöntemi olan kısa süzülme saldırısıyla kontur uçuşudur [30]. Bu davranış Denklem 2.66 ile gösterilmiştir

$$X_2(t+1) = X_{\text{en iyi}}(t) * \text{Levy}(D) + X_R(t) + (y - x) * \text{ rand} \quad [2.66]$$

$X_2(t+1)$ , $t+1$.adımındaki çözümdür. $X_R(t)$, $t$. iterasyonundaki [1 N] aralığındaki rastgele bir çözümdür. Levy $(D)$ levy flight dağılım fonksiyonudur ve Denklem 2.67 ile hesaplanmaktadır.

$$\text{Levy}\,(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \qquad [2.67]$$

$s$ değeri 0,01 olarak kabul edilir . $u$ ve $v$, 0 ve 1 arasındaki rastgele değerlerdir. $\sigma$ parametresi ise Denklem 2.68 ile aşağıdaki gibi hesaplanmaktadır.

$$\left(\frac{\Gamma(1+\beta) \times \text{sine}\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}}\right) \qquad [2.68]$$

$\beta$ değeri 1,5 olan bir sabittir. Denklem 2.66' daki $x$ ve $y$ aramadaki spiral şekli ifade ederler. $x$ ve $y$ Denklem 2.69 ile aşağıdaki şekilde hesaplanır.

$$y = r \times \cos(\theta) \text{ ve } x = r \times \sin(\theta),$$
$$r = r_1 + \text{U} \times D_1 \text{ ve } \theta = -\omega \times D_1 + \theta_1, \theta_1 = \frac{3\pi}{2} \qquad [2.69]$$

$r_1$, 1 ile 20 arasındaki arama döngü sayısı olup sabit bir sayıdır. U, 0,00565 değerine eşittir [29]. $D_1$, 1 ile arama uzayının boyutu ($D$) arasındaki bir tam sayıdır ve $\omega$ sayısı 0,005' e eşittir [30].

### 2.6.4 Genişletilmiş sömürü

Kaya kartalları, av alanını belirledikten sonra saldırı için hazır hale gelirler. Bu bir ön saldırıdır ve bu saldırı ile avın tepkisi ölçülür. Kaya kartalları burada, yavaş alçalma saldırısı ile alçak uçuş yöntemini kullanarak ava daha fazla yaklaşmış olurlar. Kaya kartallarının bu davranışı Denklem 2.70 ile gösterildiği gibidir [31].

$$X_3(t+1) = \left(X_{\text{en iyi}}(t) - X_M(t)\right) \times \alpha - r_4 + ((UB - LB) \times r_5 + LB) \times \delta \qquad [2.70]$$

$X_3(t+1)$ , $t+1$. iterasyondaki çözümü, $\alpha$ ve $\delta$ 0,1 değerlerine sahip parametrelerdir.

### 2.6.5 Daraltılmış sömürü

Kaya kartallarının, avın konumuna en yakın olduğu anda gerçekleştirmiş olduğu saldırı türüdür. Bu saldırı karada gerçekleşir ve yürüyerek avın yakalanması yöntemi uygulanır. Bu davranış Denklem 2.71 ile aşağıdaki gibi gösterilmiştir [31].

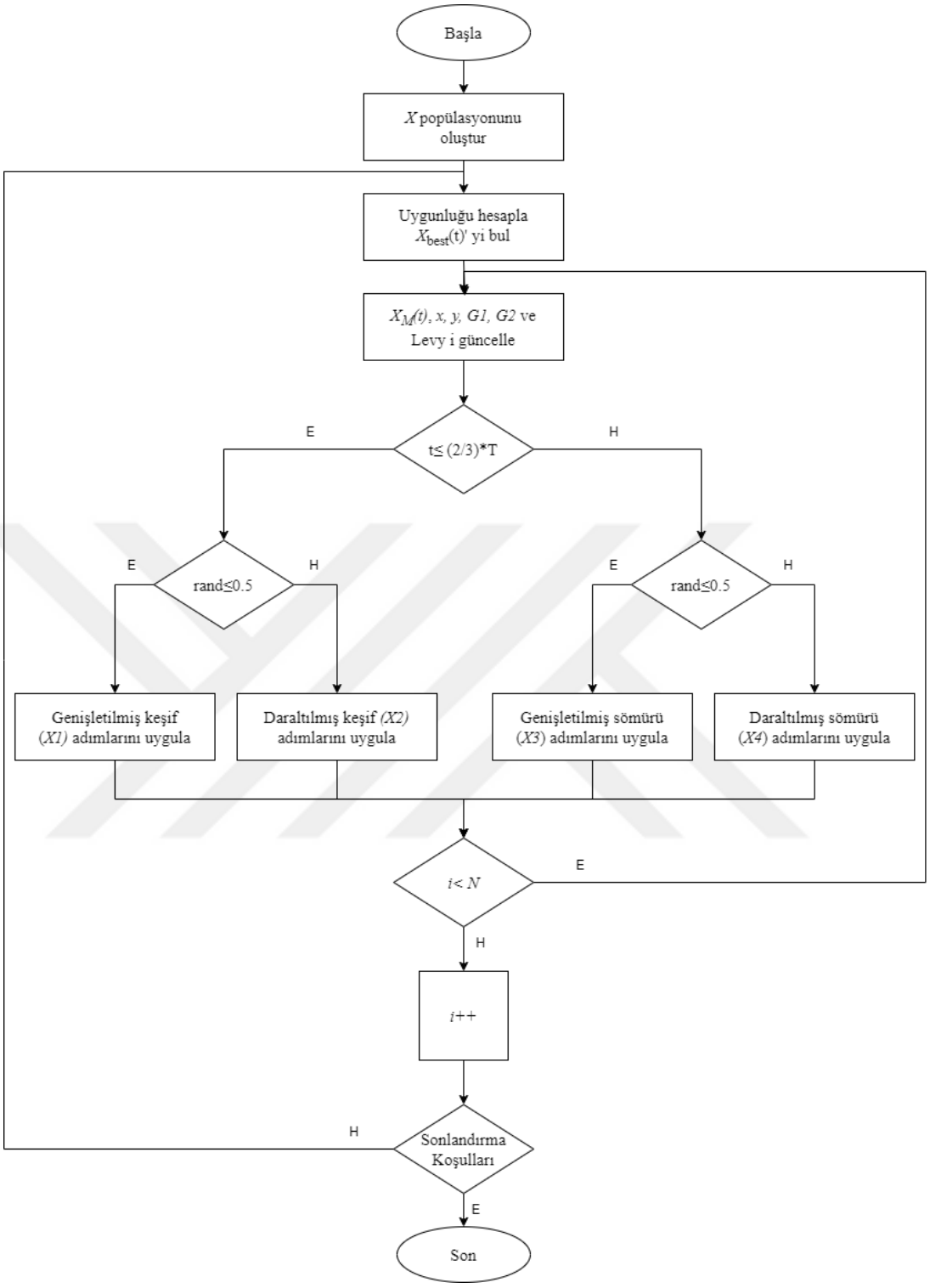
$$X_4(t+1) = QF \times X_{\text{en iyi}}(t) - (G_1 \times X(t) \times \text{ rand }) - G_2 \times \text{Levy}\,(D) + \text{ rand } \times G_1 \qquad [2.71]$$

$$QF(t) = t^{\frac{2\times \text{rand}\,()-1}{(1-T)^2}} \quad [2.72]$$

$$G_1 = 2 \times \quad rand - 1 \quad [2.73]$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad [2.74]$$

$X_4(t+1)$ , $t+1$. iterasyondaki çözümü, $QF$ keşif yöntemlerini dengeleyen kalite fonksiyonunu temsil eder. $G_1$, $[-1,+1]$ aralığındaki kaçma sırasında avı takip için, $G_2$ 2'den 0' doğru azalan, kaçma sırasında avı takip için kullanılan eğim parametreleridir. $X(t)$, $t$.adımdaki çözümdür. $r_6, r_7$ ve $r_8$ 0 ve 1 arasında rastgele bir değer alan parametrelerdir. AOA' nın akış şeması Şekil 2.6.1 ile aşağıdaki gibi gösterilmiştir [32].




**Şekil 2.6.1** AO akış şeması

## 2.7 Yapay Tavşan Optimizasyon Algoritması (ARO)

2022 yılında önerilen ARO; tavşanların yiyecek arama, rastgele saklanma ve enerji küçültme stratejilerini taklit eden bir optimizasyon algoritmasıdır [14]. Optimizasyon algoritmalarının temeli olan keşif aşaması ARO 'da yiyecek arama davranışını, sömürü aşaması ise rastgele saklanma aşamasına karşılık gelir. ARO' da keşif

aşamasından sömürü aşamasına geçiş, enerji küçültme stratejisi ile gerçekleştirilmektedir [33]. ARO' nun matematiksel modeli şu şekildedir:

### 2.7.1 Başlatma Aşaması

AROA' da ilk adım, aday çözümlerin(tavşan) rastgele olarak konumlandırılıp bunların uygunluk değerlerinin hesaplanmasıdır. Aşağıda Denklem 2.73 ile aday çözümlerin rastgele konumlandırması, Denklem 2.74 ile ise aday çözümlerin uygunluk değeri hesaplanması yer almaktadır.

$$\vec{x}_{i,d} = lb_d + \text{ rand } \times (ub_d - lb_d) i = (1,2,\cdots,N) \text{ ve } d = (1,2,\cdots,D) \quad [2.75]$$

$\vec{x}_{i,d}$ $i$. tavşanın konum vektörü, $N$ ve $D$ sırasıyla tavşan sayısı problem boyutu , tasarım parametreleri ve $ub_d$ ve $lb_d$ $d$. boyutun alt ve üst sınırlarıdır.

$$P \equiv \begin{bmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,D} \end{bmatrix}_{NxD} \quad f = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}_{Nx1} \quad [2.76]$$

$P$ tavşan popülasyonunu, $f$ uygunluk değerini ifade eder.

### 2.7.2 Dolambaçlı yiyecek arama

AROA' da popülasyondaki her bir tavşan başlatma aşaması ile rastgele olarak konumlandırılmıştır. Her bir konumdaki tavşanın yiyecek ve saklanma için bir yuvaya sahip olduğu varsayılmaktadır. Tavşanlar yiyecek arama işlemlerini kendi bulundukları konumda değil de komşu tavşanların konumlarında yaparlar. Tavşanların bu hareketine dolambaçlı yiyecek arama (keşif) denir [14]. Bu davranış modeli aşağıdaki gibi hesaplanmaktadır.

$$\vec{v}_i(t+1) = \vec{x}_j(t) + R \cdot \left(\vec{x}_i(t) - \vec{x}_j(t)\right) + \text{round}\left(0.5 \cdot (0.05 + r_1)\right) \cdot n_1 \quad [2.77]$$
$$i, j = 1, \ldots, n \text{ ve } j \neq i$$

$$R = L \cdot c \quad [2.78]$$

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right) \cdot \sin(2\pi r_2) \quad [2.79]$$

$$c(k) = \begin{cases} 1 & \text{if } k == g(l) \\ 0 & \text{else} \end{cases} k = 1, \ldots, d \text{ and } l = 1, \ldots, \lceil r_3 \cdot d \rceil \quad [2.80]$$

$$g = rand_{perm}(d) \quad [2.81]$$

$$n_1 \sim N(0,1) \quad [2.82]$$

$\vec{v}_i(t+1)$ , $t+1$. anındaki $i$. tavşan konumunu, $\vec{x}_i(t)$, $t$ anındaki tavşan konumunu, $N$ popülasyon büyüklüğünü, $t$ o anki iterasyonu, $T$ max iterasyon sayısını ifade eder. $rand_{perm}$ 1' den d' ye kadarki tam sayıların geriye döndürüldüğü bir permütasyon değeridir. $r_1$, $r_2$ ve $r_3$ 0 ile 1 arasındaki rastgele bir sayıdır. $L$ modeldeki hareket hızını temsil eden koşu uzunluğudur. $n_1$ ise standart normal bir dağılımdır [34].

### 2.7.3 Rastgele saklanma

ARO' da her bir yineleme boyunca her bir tavşan kendi çevresinde rastgele yuva oluştururlar. Tehditlere karşı tavşanlar saklanmak amacıyla kendi yuvaları yerine komşu tavşanların oluşturmuş olduğu yuvalardan birini rastgele seçerler [36]. Denklem 2.83 ile bir tavşanın çevresinde oluşturmuş olduğu yuvalar gösterilmiştir [32]. Diğer denklemler (2.84-2.86) bu denklemin hesaplanması için kullanılmaktadır.

$$\vec{b}_{i,j}(t) = \vec{x}_i(t) + H \cdot g \cdot \vec{x}_i(t), i = 1, \dots, n \text{ and } j = 1, \dots, d \quad [2.83]$$

$$H = \frac{T - t + 1}{T} \cdot r_4 \quad [2.84]$$

$$n_2 \sim N(0,1) \quad [2.85]$$

$$g(k) = \begin{cases} 1 & \text{if } k == j \\ 0 & \text{else} \end{cases} k = 1, \dots, d \quad [2.86]$$

$H$, gizleme parametresidir ve doğrusal olarak yinelemeler boyunca 1' den $1/T$'ye düşer. $\vec{b}_{i,j}$, $i$. tavşanın $j$. yuvasıdır. $n_2$ ise standart normal bir dağılımdır.

Tehditlere karşı tavşanlar rastgele olarak komşu tavşanların yapmış oldukları yuvalardan birini rastgele seçerler [35]. Bu seçme işlemi Denklem 2.87, Denklem 2.88 ve Denklem 2.59 ile gösterilmiştir [14].

$$\vec{v}_i(t+1) = \vec{x}_i(t) + R \cdot \left(r_4 \cdot \vec{b}_{i,r}(t) - \vec{x}_i(t)\right), i = 1, \dots, n \quad [2.87]$$

$$g_r(k) = \begin{cases} 1 & \text{if } k == \lceil r_5 \cdot d \rceil \\ 0 & \text{else} \end{cases} k = 1, \dots, d \quad [2.88]$$

$$\vec{b}_{i,r}(t) = \vec{x}_i(t) + H \cdot g_r \cdot \vec{x}_i(t) \quad [2.89]$$

$\vec{b}_{i,r}$ , $i$. tavşan tarafında rastgele seçilmiş yuvayı, $r_4$ ve $r_5$ 0 ile 1 arasındaki rastgele bir sayıdır.

AROA' da dolambaçlı yiyecek arama ve rastgele saklanma gerçekleştikten sonra konum güncellemesi yapılır. Konum güncelleme denklemi Denklem 2.90 ile ortaya çıkarılmaktadır [14].

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_i(t) & f(\vec{x}_i(t)) \leq f(\vec{v}_i(t+1)) \\ \vec{v}_i(\mathrm{t}+1) & f(\vec{x}_i(t)) > f(\vec{v}_i(t+1)) \end{cases} \quad [2.90]$$

Denklem 2.90 ile mevcut konum uygunluk değeri aday çözüme ait uygunluk değerinden daha iyiyse yer değiştirme yapılmaz, iyi değilse yer değiştirme yapılır.

### 2.7.4 Enerji küçültme

ARO' da tavşanlar enerji seviyesine göre ilerlemeler arttıkça keşif aşamasından sömürü aşamasına geçerler [36]. Denklem 2.91 geçişi temsil edilmektedir [14].

$$A(t) = 4\left(1 - \frac{t}{T}\right)\ln\frac{1}{r} \quad [2.91]$$

$r$ değeri 0 ile 1 arasında rastgele bir değerdir. Tavşanlar $A(t) > 1$ ise, yiyecek arama yani keşif aşaması; $A(t) < 1$ olduğunda ise rastgele saklanma yani sömürü aşamasındadırlar. ARO algoritmasının temel adımları Çizelge 2.7.1 ile aşağıdaki gibi gösterilmiştir [14].

**Çizelge 2.7.1** Yapay Tavşan Algoritması Adımları

| Yapay Tavşan Algoritması | |
|---|---|
| **Adım 1.** | Başlatma |
| **Adım 2.** | $P$, rastgele bir tavşan popülasyonu oluşturun |
| **Adım 3.** | **for** $i$ =1: $N$ (tavşan sayısı) **do** |
| **Adım 4.** | $f_i$ : Her bir tavşanın uygunluk değerini hesaplayın |
| **Adım 5.** | **end** |
| **Adım 6.** | **repeat** |
| **Adım 7.** | **Adım 1: Seçim aşaması** |
| **Adım 8.** | $P$ başlangıç tavşanlarını seçin |
| **Adım 9.** | **Adım 2: Arama aşaması** |
| **Adım 10.** | Dolambaçlı yiyecek arama (Keşif) |
| **Adım 11.** | Rastgele saklanma (Sömürü) |
| **Adım 12.** | **Adım 3: Güncelleme aşaması** |
| **Adım 13.** | Tavşanların uygunluk değerlerine göre $P$ popülasyonu güncelle |
| **Adım 14.** | **until** |
| **Adım 15.** | **end** |

## 2.8 Dağ Ceylanı Optimizasyonu (MGO)

MGO, yabani dağ ceylanlarının sosyal yaşamlarını ve kendi aralarındaki hiyerarşilerini ilham alan metasezgisel bir algoritmadır [15]. Bu algoritma dağ ceylanlarının yaşamlarındaki dört temel noktayı ele alır. Bunlar: Bekar erkek sürüleri, doğum sürüleri, bölgesel bekar erkekler ve yiyecek bulmak için gerçekleştirilen göç davranışı [37]. MGO' da her bir ceylan $X_i$, bekar erkek sürüsü, doğum sürüsü ya da bölgesel erkek sürüsü üyelerinden birisi olabilir [38]. MGO algoritmasının matematiksel modeli aşağıdaki gibi sunulmuştur.

### 2.8.1 Bölgesel Bekar Erkekler

Erkek dağ ceylanları yetişkin olup güçlendiklerinde kendilerine bir bölge oluştururlar. Bu bölgeler birbirlerinden oldukça mesafelidir ve bu şekilde dağ ceylanları bölgesel olarak yaşarlar. Erkek dağ ceylanları, dişi ceylanlar üzerinde hakimiyet ve bölge kontrolü mücadelesi verirler. Burada genç erkek ceylanlar ise bölge işgali ve dişi

hakimiyeti kurmak istediklerinde dağ ceylanlarının koruma davranışı ile karşılaşırlar [39]. Yetişkin dağ ceylanını bölge modellemesi Denklem 2.92 ile aşağıdaki gibidir [15].

$$TSM = \text{erkek}_{\text{ceylan}} - |(ri_1 \times BH - ri_2 \times X(t)) \times F| \times \text{Cof}_r \quad [2.92]$$

Burada $\text{erkek}_{\text{ceylan}}$, en iyi küresel çözümün konum vektörü; $ri_1$, $ri_2$ parametreleri 1 veya 2 rastgele tamsayılarıdır. $BH$ genç erkek sürü katsayısı, $\text{Cof}_r$ her yinelemede güncellenen, arama yeteneğini artırmak için kullanılan rastgele bir katsayı vektörüdür. $BH$ Denklem 2.93 ile aşağıdaki gibi hesaplanır [15].

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lceil r_2 \rceil, ra = \left\{ \left\lceil \frac{N}{3} \right\rceil \dots N \right\} \quad [2.93]$$

$X_{ra}$ , Genç bir ceylanın $ra$ aralığındaki rastgele bir çözümüdür. $r_1$ ve $r_2$ 0 ve 1 arasındaki rastgele sayılardır. Rastgele seçilmiş çözüm adaylarının ortalama değeridir. $N$ ise toplam dağ ceylanı sayısıdır. $F$ değeri Denklem 2.94 ile aşağıdaki gibi hesaplanmaktadır [15].

$$F = N_1(D) \times \exp\left(2 - \text{iterasyon} \times \left(\frac{2}{\text{Maksimum}_{\text{iterasyon}}}\right)\right) \quad [2.94]$$

$N_1$, standart dağılımdan rastgele bir sayıdır. Exp üstel fonksiyon, $\text{Maksimum}_{\text{iterasyon}}$, maksimum adım sayısıdır. $\text{Cof}_r$, Denklem 2.95 ile aşağıdaki gibi hesaplanmaktadır [15].

$$\text{Cof}_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos\left((r_4 \times 2) \times N_3(D)\right) \end{cases} \quad [2.95]$$

$r_3$, $r_4$ ve $rand$ 0 ve 1 arasındaki rastgele sayılardır. $N_2$, $N_3$ ve $N_4$ normal aralığa ve problem boyutlarına göre rastgele sayılardır. $a$ parametresi ise Denklem 2.96 ile ifade edilir [15].

$$a = -1 + \text{iterasyon} \times \left(\frac{-1}{Maksimum_{\text{iterasyon}}}\right) \quad [2.96]$$

### 2.8.2 Doğum Sürüleri

Dağ ceylanları, erkek yavrular üretmek için doğum sürülerine bağımlı olan bir türdür. Erkek ceylanlar, dişi elde etmeye çalışan genç erkeklere yardımlarının yanında dişi ceylanların doğumuna da yardımcı olabilirler [40]. Bu davranışları Denklem 2.97 ile tanımlanır [15].

$$MH = \left(BH + \mathrm{Cof}_{1,r}\right) + \mathrm{ri}_{3} \times \mathrm{erkek}_{G} - ri_4 \times X_{\mathrm{rand}}) \times \mathrm{Cof}_{2,r} \quad [2.97]$$

$BH$, genç erkeklerin etki faktörü, $\mathrm{Cof}_{1,r}$ ve $\mathrm{Cof}_{2,r}$ bağımsız olarak hesaplanan rastgele seçilmiş katsayı vektörleridir. $\mathrm{ri}_{3}$ ve $ri_4$ 1 ya da 2 değerini alan rastgele tamsayılardır. $X_{\mathrm{rand}}$, tüm popülasyondan seçilen rastgele bir ceylanın konum vektörüdür.

### 2.8.3 Bekar Erkek Sürüleri

Erkek ceylanlar olgunlaştıkça bölge üretme ve dişi ceylanlar üzerinde kontrolü ele geçirme eğilimindedir. Bu noktada genç erkek ceylanlar dişiler üzerinde hâkimiyet kurmak için yetişkin erkeklerle mücadele etmeye başlar ve şiddet ortaya çıkabilir [41]. Bu davranış matematiksel olarak Denklem 2.98 ile aşağıdaki şekilde ifade edilir [15].

$$BMH = (X(t) - D) + (ri_5 \times \mathrm{erkek}_{\mathrm{ceylan}} - ri_6 \times BH) \times \mathrm{Cof}_{r} \quad [2.98]$$

$X(t)$, anlık ceylan vektörel konumu, $ri_5$ ve $ri_6$ 1 ya da 2 tamsayılarını alan rastgele sayılardır. $\mathrm{erkek}_{\mathrm{ceylan}}$, erkek ceylanın vektörel konumudur. $BH$ ise genç erkek sürüsünün etki faktörü, $\mathrm{Cof}_{r}$ rastgele seçilmiş katsayı vektörüdür. $D$ Denklem 2.99 ile aşağıdaki gibi tanımlanır.

$$D = \left(|X(t)| + |\ \mathrm{erkek}_{\mathrm{gazelle}}\ |\right) \times (2 \times r_6 - 1) \quad [2.99]$$

$r_6$ 0 ve 1 arasındaki rastgele bir sayıdır.

### 2.8.4 Yiyecek Aramaya Geçiş

Dağ ceylanları, otlamak ve yiyecek kaynağı bulabilmek için devamlı uzun yolculuklar yaparlar. Bu yolculukları, yüksek hızlı koşma ve atlama becerileri yardımıyla gerçekleştirirler [41]. Dağ ceylanlarının bu davranışı Denklem 2.100 ile aşağıdaki gibi gösterilir [15].

$$MSF = (ub - lb) \times r_7 + lb \quad [2.100]$$

$ub$ ve $lb$ sırasıyla problemin üst ve alt sınırlarıdır. $r_7$ 0 ve 1 arasındaki rastgele bir sayıdır.

Yeni nesil ceylanlar üretmek için yukarıdaki dört mekanizma tüm ceylanlara uygulanır. Popülasyona yeni bir nesil eklenir ve her nesil bir kopyaya eşittir. Her bir nesilde ceylanlar çözüm kalitelerine göre sıralanmaktadır. Kaliteli ve gelecek vaat eden çözümlere sahip, maliyeti daha düşük olan en iyi ceylanlar toplam popülasyonda korunmaktadır. Yaşlı veya zayıf olduğu düşünülen diğer ceylanlar ise tüm popülasyondan uzaklaştırılır. En iyi ceylan aynı zamanda bölgenin sahibi olan yetişkin erkek ceylan olarak da kabul edilir [15]. MGO' nun sözde kodu Çizelge 2.8.1 ile aşağıdaki gibi gösterilmiştir [37].

**Çizelge 2.8.1** Dağ Ceylanı Optimizasyonu Sözde Kodu

| Dağ Ceylanı Optimizasyonu | |
|---|---|
| **Adım 1.** | **Girdi**: Popülasyon büyüklüğü $N$ ve maksimum yineleme sayısı $T$ |
| **Adım 2.** | **Çıktı**: Ceylanın konumu ve uygunluk potansiyeli |
| **Adım 3.** | $X_i(i = 1,1, \dots, N)$' yi kullanarak rastgele bir popülasyon oluşturun |
| **Adım 4.** | Ceylanların uygunluk değerlerini hesaplayın |
| **Adım 5.** | **While** durma koşulu sağlanmadığı sürece |
| **Adım 6.** | **for** (her bir ceylan için $X_i$ hesapla) |
| **Adım 7.** | Bölgesel bekar erkekler hesapla |
| **Adım 8.** | Doğum sürüleri hesapla |
| **Adım 9.** | Bekar erkek sürüleri hesapla |
| **Adım 10.** | Yiyecek aramaya geçiş hesapla |
| **Adım 11.** | Bölgesel bekar erkekler, Doğum sürüleri, Bekar erkek sürüleri ve Yiyecek aramaya geçiş aşamalarının uygunluk değerlerini hesaplayın ve bunları alana ekleyin |
| **Adım 12.** | **end for** |
| **Adım 13.** | Popülasyonu artan sırada sıralayın |
| **Adım 14.** | $en\ iyi_{ceylan}$ güncelle |
| **Adım 15.** | Maksimum popülasyon da $en\ iyi_{ceylan}$' ı kaydedin |
| **Adım 16.** | **end while** |
| **Adım 17.** | **Return** $en\ iyi_{ceylan}$ |

## 2.9 Çayır Köpeği Optimizasyon Algoritması (PDO)

PDO, çayır köpeklerinin yiyecek arama ve yuva yapma davranışlarından ilham alarak geliştirilmiş bir optimizasyon algoritmasıdır [16]. PDO' da çayır köpeklerinin yiyecek arama ve yuva inşa etme faaliyetleri, optimizasyon için keşif aşamasıdır. Çayır köpeklerinin en önemli özelliklerinden biri de iletişim becerileridir. Bu iletişim becerileri ile avcı tehditleri, yiyecek ile ilgili çeşitli bilgiler gibi seslere sahiptirler. PDO' da sömürü aşaması ise çayır köpeklerinin iletişim ve avcılara karşı mücadelelerini kapsamaktadır [42]. PDO' da keşif ve sömürü aşamaları $maks_{iterasyon}$ sayısına bağlı olarak uygulanır. İlk iki bölüm, keşif aşamasını; son iki bölüm ise sömürü aşamasını tanımlar [16]. Algoritmanın keşif ve sömürü aşamaları sırasıyla Denklem 2.103 ve Denklem 2.104 olarak aşağıda gösterilmiştir [42].

$$\text{iterasyon} < \frac{Maksimum_{\text{iterasyon}}}{4} \text{ ve } \frac{Maksimum_{\text{iterasyon}}}{4} \leq \text{ iterasyon} < \frac{Maksimum_{\text{iterasyon}}}{2} \quad [2.103]$$

$$\frac{Maksimum_{\text{iterasyon}}}{2} \leq \text{iterasyon} \leq 3\frac{Maksimum_{\text{iterasyon}}}{4} \text{ ve } 3\frac{Maksimum_{\text{iterasyon}}}{4} \leq \text{ iterasyon} \leq \textit{M}\text{aksimum}_{\text{iterasyon}} \quad [2.104]$$

PDO' nın matematiksel modeli aşağıdaki alt bölümlerde tanımlanmıştır.

### 2.9.1 Başlatma Aşaması

PDO' da çayır köpekleri (PD) rastgele olarak konumlandırılarak başlangıç popülasyonu oluşturulur. Çayır köpeği popülasyonları, arama aracılarıdır ve her bir çayır köpeğini temsil etmek için d-boyutlu uzayda vektör kullanılır [16]. Bir PDO kolonisi, $m$ adet zümre ve bu $m$ adet zümrede, $n$ adet çayır köpeğinin yer almasıyla oluşur. Zümredeki her bir çayır köpeğinin (PD) konumu vektör ile gösterilir. Denklem 2.105 ile bir kolonideki zümrelerin konumları gösterilir.

$$CT = \begin{bmatrix} CT_{1,1} & CT_{1,2} & \cdots & CT_{1,d} & CT_{1,d} \\ CT_{2,1} & CT_{2,2} & \cdots & CT_{2,d-1} & CT_{2,d} \\ \vdots & \vdots & CT_{i,j} & \vdots & \vdots \\ CT_{m,1} & CT_{m,2} & \cdots & CT_{m,d-1} & CT_{m,d} \end{bmatrix} \quad [2.105]$$

$CT_{i,j}$ kolonideki $i$. zümrenin $j$. boyutunu ifade eder. Bir zümredeki çayır köpeklerinin konumu ise Denklem 2.106 ile aşağıdaki gibidir.

$$PD = \begin{bmatrix} PD_{1,1} & PD_{1,2} & \cdots & PD_{1,d-1} & PD_{1,d} \\ PD_{2,1} & PD_{2,2} & \cdots & PD_{2,d-1} & PD_{2,d} \\ \vdots & \vdots & PD_{i,j} & \vdots & \vdots \\ PD_{n,1} & PD_{n,2} & \cdots & PD_{n,d-1} & PD_{n,d} \end{bmatrix} \quad [2.106]$$

$PD_{i,j}$ $i$. çayır köpeğinin $j$. boyutunu verir. $n \leq m$ koşulu sağlanmalıdır. Her bir çayır köpeği ve zümrenin konumu sırasıyla Denklem 2.107 ve Denklem 2.108 ile aşağıdaki gibi gösterilir.

$$CT_{i,j} = U(0,1) * \left(UB_j - LB_j\right) + LB_j \quad [2.107]$$

$$PD_{i,j} = U(0,1) * \left(ub_j - lb_j\right) + lb_j \quad [2.108]$$

$UB_j$ ve $LB_j$, optimizasyon problemine dair $j$. boyutun üst ve alt sınırlarıdır. $ub_j = UB_j/m$, $lb_j = LB_j/m$ ve $U$, 0 ile 1 arasındaki rastgele bir sayıdır. Her bir çayır köpeğinin konumuna ait uygunluk değeri yiyeceğin kalitesini, yuva kurma potansiyelini ve avcılardan korunma derecesini gösterir. Uygunluk değeri hesabı her bir çayır köpeği için yapıldıktan sonra bir dizi şeklinde sıralama yapılır ve optimizasyon problemine göre minimum 4 ya da maksimum 4 değer alınır. Minimizasyon problemlemi için alınan 4 değerden minimum olanı en iyi çözüm kabul edilir. Maksimizasyon problemi için ise 4 değerden maksimum olanı alınarak en iyi kabul edilir. Geriye kalan 3 değer çayır köpeklerinin yuva inşa etme davranışı için dikkate alınır [16]. PDO' da her bir çayır köpeğinin uygunluk değeri Denklem 2.209' de olduğu gibi hesaplanmaktadır [42].

$$f(PD) = \begin{bmatrix} f_1\left(\left[PD_{1,1} & PD_{1,2} & \ldots & PD_{1,d-1} & PD_{1,d}\right]\right) \\ f_2\left(\left[PD_{2,1} & PD_{2,2} & \ldots & PD_{2,d-1} & PD_{2,d}\right]\right) \\ \vdots \quad \vdots \quad \ldots \quad \vdots \quad \vdots \\ f_n\left(\left[PD_{n,1} & PD_{n,2} & \ldots & PD_{n,d-1} & PD_{n,d}\right]\right) \end{bmatrix} \quad [2.109]$$

Burada $f(PD_{n,d})$ $n$. çayır köpeğinin $d$. boyutundaki uygunluk değerini verir.

### 2.9.2 Keşif Aşaması

Çayır köpeklerinin yiyecek arama ve yuva inşa etme faaliyetleri bu aşamada gerçekleştirilir. Çayır köpekleri yuvalarını besin kaynağının bol olduğu alanlara inşa ederler. Besin kaynakları tükendiği anda yeni besin kaynağı keşfederek, besin kaynağı çevresine yeni yuvalarını inşa ederler. Denklem 2.101 ile algoritmanın keşif aşaması tanımlanmıştır. Çayır köpeklerinin yiyecek arama davranışı Levy uçuş hareketi ile gösterilir. Bu hareket ile çayır köpekleri geniş bir alanda besin kaynakları için keşif yaparlar. Yiyecek kaynakları keşfedildiğinde, çayır köpekleri kullandıkları sesler ile sürünün diğerlerine bilgi verilir. Besin kaynağının miktarı ve kalitesine göre yeni yuva inşa edilir. Denklem 2.108 ile yiyecek arama için konum güncellemesi aşağıdaki gibi hesaplanmıştır [16].

$$PD_{i+1,j+1} = G\ \text{En iyi}_{\ i,j} - \ \text{eCEn iyi}_{\ i,j} \times \rho - CPD_{i,j} \times \text{Levy}\ (n) \forall\ \text{iterasyon} < \frac{\text{Maksimum}_{\ \text{iterasyon}}}{4} \quad [2.108]$$

Denklem 2.109 ile yeni yuva inşa konum güncellemesi aşağıdaki gibi gösterilmiştir.

$$PD_{i+1,j+1} = \ \text{GBest}_{\ i,j} \times rPD \times DS \times \text{Levy}(n)\ \forall \frac{\text{Maksimum}_{\ \text{iterasyon}}}{4} \leq\ \text{iterasyon} < \frac{\text{Maksimum}_{\ \text{iterasyon}}}{2} \quad [2.109]$$

$\text{GBest}_{\ i,j}$, global anlamdaki en iyi çözüm olduğu durumda $\text{eCBest}_{\ i,j}$ mevcut en iyi çözümün etkinini tanımlar ve Denklem 2.110 ile aşağıdaki hesaplanır.

$$eCEn\ iyi_{\ i,j} = \ GEn\ iyi_{\ i,j} \times \Delta + \frac{PD_{i,j} \times \ mean\left(PD_{n,m}\right)}{GEn\ iyi_{\ i,j} \times \left(UB_j - LB_j\right) + \Delta} \quad [2.110]$$

$$CPD_{i,j} = \frac{\text{GEn iyi}_{\ i,j} - rPD_{i,j}}{\text{GEn iyi}_{\ i,j} + \Delta} \quad [2.111]$$

$$DS = 1.5 \times r \times \left(1 - \frac{\text{iterasyon}}{\text{Maksimum}_{\ \text{iterasyon}}}\right)^{\left(2\frac{\text{iterasyon}}{\text{Maksimum}_{\ \text{iterasyon}}}\right)} \quad [2.112]$$

$r$, keşif için kullanılan $-1$ veya $+1$ değerlerini alan bir parametredir. $\Delta$ ise çayır köpekleri arasındaki farkı ifade eden bir bir tam sayıdır.

### 2.9.3 Sömürü Aşaması

Çayır köpeklerinin iletişim ve avcılara karşı mücadeleleri bu aşamada gerçekleştirilir. PDO' da kullanılan sömürü mekanizmalarının amacı, keşif aşamasında keşfedilen bölgelerin yoğun bir şekilde araştırmaktır. Bu aşama Denklem 2.102 ile tanımlanmıştır. Çayır köpeklerinin iki özel çağrısı bulunmaktadır. Bunlar bir yırtıcı hayvan önleme ve yeni besin kaynağı (yeni yuva inşa etme) çağrısıdır. Sömürü aşamasında kullanılan stratejiler Denklem 2.113 ve Denklem 2.114 ile aşağıdaki gibi gösterilmiştir [43].

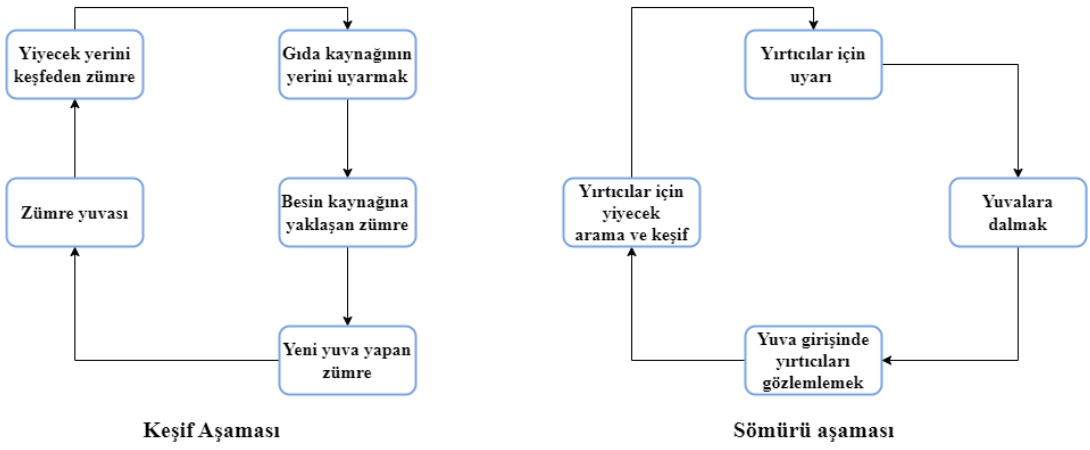
$$PD_{i+1,j+1} = GEn\ iyi_{i,j} - eCEn\ iyi_{i,j} \times \varepsilon - CPD_{i,j} \times rand\forall \frac{Maks_{\text{iterasyon}}}{2} \leq \text{iterasyon} < 3\frac{Maks_{\text{iterasyon}}}{4} \quad [2.113]$$

$$PD_{i+1,j+1} = \text{GEn iyi}_{i,j} \times PE \times \text{rand}\ \forall 3\frac{\text{Maks}_{\text{iterasyon}}}{4} \leq \text{iterasyon} < \text{Maks}_{\text{iterasyon}} \quad [2.114]$$

$\varepsilon$, besin kaynağının kalitesini gösteren parametre, rand ise 0 ve 1 arasındaki rastgele bir sayıyı, $CPD_{i,j}$ kolonideki tüm çayır köpeklerinin toplam etkisini tanımlar. $PE$ avcı etkisi olup Denklem 2.115 ile aşağıdaki gibi hesaplanır.

$$PE = 1.5 \times \left(1 - \frac{\text{iterasyon}}{Maksimum_{\text{iterasyon}}}\right)^{\left(2\frac{\text{iterasyon}}{Maksimumx_{\text{iterasyon}}}\right)} \quad [2.115]$$

PDO' nun keşif ve sömürü aşamaları Şekil 2.5 ile aşağıdaki gibi gösterilmiştir [44].




**Şekil 2.9.1** PDO' nun keşif ve sömürü aşama modelleri

PDO' nın sözde kodu Çizelge 2.9.1 ile aşağıdaki gibi gösterilmiştir [16].

**Çizelge 2.9.1** Çayır Köpeği Optimizasyonu Sözde Kodu

| Çayır Köpeği Optimizasyonu | |
|---|---|
| **Adım 1.** | Başlatma |
| **Adım 2.** | $n, m, \rho, \varepsilon$ parametrelerini ayarla |
| **Adım 3.** | $GBest$ ve $CBest$ $\emptyset$ olarak ayarla |
| **Adım 4.** | $CT$ ve $PD$ aday çözümleri başlat |
| **Adım 5.** | **While** (iterasyon< $Maks_{iterasyon}$) **do** |
| **Adım 6.** | **for** ($i = 1$ to $m$) **do** |
| **Adım 7.** | **for** ($j = 1$ to $n$) **do** |
| **Adım 8.** | $PD$'nin uygunluk değerini hesapla |
| **Adım 9.** | Şu ana kadar ki en iyi çözümü bul ($CBest$) |
| **Adım 10.** | $GBest$ ' i güncelle |
| **Adım 11.** | $DS$ ve $PE$' yi güncelle |
| **Adım 12.** | $CPD_{i,j}$ ' yi güncelle |
| **Adım 13.** | **if**(iterasyon $< \frac{\text{Maks}_{\text{iterasyon}}}{4}$)**then** {yiyecek arama} |
| **Adım 14.** | Denklem [2.108]'ı çalıştır |
| **Adım 15.** | **else if**($\frac{Maks_{\text{iterasyon}}}{4} \leq$ iterasyon $< \frac{Maks_{\text{iterasyon}}}{2}$) **then** {yeni yuva oluşturma aktivitesi} |
| **Adım 16.** | Denklem [2.109]' ı çalıştır. |
| **Adım 17.** | **else if**($\frac{Maks_{\text{iterasyon}}}{2} \leq$ iterasyon $\leq 3\frac{Maks_{\text{iterasyon}}}{4}$ )**then** {yiyecek alarmı} |
| **Adım 18.** | Denklem [2.113]' ü çalıştır. |
| **Adım 19.** | **else** {avcı alarmı} |
| **Adım 20.** | Denklem [2.114]' ü çalıştır. |
| **Adım 21.** | **end if** |
| **Adım 22.** | **end for** |
| **Adım 23.** | **end for** |
| **Adım 24.** | iterasyon = iterasyon + 1 |
| **Adım 25.** | **end while** |
| **Adım 26.** | **Return** en iyi çözüm ($GBest$) |
| **Adım 27.** | **end** |

## 2.10 Kerevit Optimizasyon Algoritması (COA)

COA, kerevitlerin yiyecek arama, yaz tatili ve rekabetçi davranışlarından ilham alan bir metasezgiseldir. Yiyecek arama aşaması ve rekabetçi davranış aşaması COA' nın kullanım aşamasına, yazlık tatil yeri aşaması COA' nın keşif aşamasına karşılık gelir [17]. COA algoritması popülasyon tabanlı bir algoritmadır, bu nedenle öncelikle bir dizi aday çözüm oluşturulur [45]. COA' daki aday çözüm Denklem 2.116 'da aşağıdaki gibidir [17].

$$X = [X_1, X_2, \cdots, X_N] = \begin{bmatrix} X_{1,1} & \cdots & X_{1,j} & \cdots & X_{1,\dim} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ X_{i,1} & \cdots & X_{i,j} & \cdots & X_{i,\dim} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ X_{N,1} & \cdots & X_{N,j} & \cdots & X_{N,\dim} \end{bmatrix} \quad [2.115]$$

$X$, başlangıç popülasyon konumu, $N$ popülasyon sayısı, $dim$ ise popülasyon boyutudur. $X_{i,j}$, $i$'nin $,j$ boyutundaki konumudur ve Denklem 2.116 ile hesaplanmaktadır [17].

$$X_{i,j} = lb_j + \left(ub_j - lb_j\right) \times \text{ rand} \quad [2.116]$$

$lb_j, ub_j$ sırasıyla $j$. boyutun alt ve üst sınırlarıdır. Rand rastgele bir tamsayıdır.

COA' da keşif ve kullanım aşamaları sıcaklığa göre düzenlenir. Sıcaklığın değişmesi kerevitlerin davranışlarını etkileyerek farklı aşamalara geçmelerini sağlamaktadır. $sıcaklık \geq 30℃$ olduğunda kerevitler yazlık tatil yeri aşamasına geçerler. Kerevitler $sıcaklık < 30℃$ ve $sıcaklık > 20℃$ aralığında ideal yiyecek arama davranışı gösterirler. Kerevitlerin beslenme miktarıda sıcaklık ile bağlantılıdır. İdeal sıcaklık aralığında beslenme miktarları artar [45]. Sıcaklık Denklem 2.117 ile, kerevit alımı Denklem 2.118 ile aşağıdaki gibi hesaplanmaktadır [17].

$$\text{sıcaklık} = \text{rand} \times 15 + 20 \quad [2.117]$$

sıcaklık , kerevitin bulunduğu ortam sıcaklığı, rand, rastgele bir tamsayıyı ifade eder.

$$p = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma)} \times \exp\left(-\frac{(\text{ sıcaklık } - \mu)^2}{2\sigma^2}\right)\right) \quad [2.118]$$

$\mu$, kerevitler için en uygun sıcaklığı, $\sigma$ ve $C_1$ farklı sıcaklıklarda kerevit alımını kontrol etmek için kullanılan parametreleri tanımlar.

### 2.10.1 Yazlık Tatil Yeri Aşaması (Keşif)

$sıcaklık \geq 30℃$ olduğunda, bu sıcaklık değeri kerevitler için çok yüksektir. Bu durumda kerevitler genellikle gölgeli ve serin mağaralara sığınır ve yazlık tatil yeri sağlanmış olur [46]. Denklem 2.119 ile bu durum tanımlanmıştır [17].

$$X_{\text{shade}} = (X_G + X_L)/2 \quad [2.119]$$

$X_G$ en uygun konumu, $X_L$ anlık konumu ifade eder.

Kerevitlerin serin mağaralar için savaşması rastgele bir olaydır. Bu rastgeleliği tanımlayan $rand < 0{,}5$ ise mağara için rekabetin olmadığı, kerevitin mağaraya doğrudan girebileceğini tanımlar [45]. Bu durum Denklem 2.220 ile aşağıdaki gibi gösterilir [17].

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times \text{ rand } \times \left(X_{\text{shade}} - X_{i,j}^{t}\right) \quad [2.220]$$

$t$ mevcut yineleme sayısını, $t+1$ bir sonraki yinelemeyi, $C_2$ ise azalan bir eğridir ve Denklem 2.221 ile aşağıdaki gibi tanımlanır.

$$C_2 = 2 - (t/T) \quad [2.221]$$

$T$ maksimum iterasyon sayısınıdır. Yazlık tatil yeri aşamasında kerevitler çözüme en uygun mağaraya yaklaşmayı hedeflerler. Bu bireyleri etkili bir şekilde en iyi çözüme yaklaştırır. Bu COA' nın kullanım yeteneğini daha da artırarak algoritma yakınsamasını hızlandırır [45].

### 2.10.2 Rekabetçi Davranış Aşaması (Sömürü)

$sıcaklık \geq 30$ ℃ ve $rand \geq 0{,}5$ olduğunda kerevitler mağara seçimi için mücadele içerisine gireceklerdir[17]. Kerevitlerin mağara için mücadeleleri Denklem 2.222 ile aşağıdaki gibidir [17].

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{\text{shade}} \quad [2.222]$$

$z$ rastgele kereviti ifade eder ve Denklem 2.223 ile hesaplanır.

$$z = \text{round } (\text{rand} \times (N-1)) + 1 \quad [2.223]$$

$N$ popülasyon sayısıdır.

### 2.10.3 Yiyecek Toplama Aşaması (Sömürü)

$sıcaklık \leq 30$, kerevitlerin beslenmeleri için ideal bir sıcaklıktır. Bu sıcaklıkta kerevitler yiyeceğe doğru hareket ederler ve yiyecek bulunduktan sonra, yiyeceğin büyüklüğüne karar verirler. Yiyeceğin çok büyük olması durumunda kerevit, yiyeceği parçalar ve yer [47]. Yiyeceğin konumu Denklem 2.224 ile aşağıdaki gibi gösterilmiştir [17].

$$X_{yiyecek} = X_G \quad [2.224]$$

Yiyeceğin boyutu $Q$ ile gösterilir ve Denklem 2.225 ile aşağıdaki gibi hesaplanır [12].

$$Q = C_3 \times \text{ rand } \times \left( \text{en iyi }_i / \text{ en iyi }_{\text{yiyecek}} \right) \quad [2.225]$$

$C_3$ en büyük temsil eden yiyecek faktörüdür ve sabit değerli olup değeri 3 tür. $\text{en iyi }_i$, $\text{en iyi}_{\text{yiyecek}}$ , sırasıyla $i$. kerevitin ve yiyeceğin uygunluk değerleridir. Eğer $Q > (C_3 + 1)/2$ ise yiyecek boyutu çok büyüktür ve ayağıyla kerevit yiyeceği parçalar. Bu işlem Denklem 2.226 ile aşağıdaki gibi gösterilmiştir [17].

$$X_{\text{yiyecek}} = \exp\left(-\frac{1}{Q}\right) \times X_{\text{yiyecek}} \quad [2.226]$$

Yiyecek parçalanıp küçüldüğünde, ikinci ve üçüncü ayaklar sırasıyla parçayı tüketirler [47]. Yiyecek arama denklemi aşağıdaki Denklem 2.227 ile gösterilmiştir [17].

$$X_{i,j}^{t+1} = X_{i,j}^{t} + X_{\text{food}} \times p \times (\cos(2 \times \pi \times \text{ rand }) - \sin(2 \times \pi \times \text{ rand })) \quad [2.227]$$

Eğer $Q \leq (C_3 + 1)/2$ ise kerevit doğrudan yiyeceğe yönelir yiyeceği tüketir. Bu Denklem 2.228 ile gösterilmiştir [17].

$$X_{i,j}^{t+1} = \left(X_{i,j}^{t} - X_{\text{yiyecek}}\right) \times p + p \times \text{ rand } \times X_{i,j}^{t} \quad [2.228]$$

COA' nın sözde kodu Çizelge 2.10.1 ile aşağıdaki gibi gösterilmiştir [17].

**Çizelge 2.10.1** Kerevit Optimizasyon Algoritması Sözde Kodu

| Kerevit Optimizasyon Algoritması | |
|---|---|
| **Adım 1.** | Başlatma, $T$ toplam adım sayısı, $N$ popülasyon sayısı, dim problem boyutu |
| **Adım 2.** | Rastgele bir başlangıç popülasyonu oluşturun |
| **Adım 3.** | $X_G, X_L$ uygunluk değerlerini hesaplayın |
| **Adım 4.** | **While** ($t < T$) |
| **Adım 6.** | Denklem 2.117' yi kullanarak sıcaklığı hesaplayın |
| **Adım 7.** | **if** sıcaklık $> 30$ ise |
| **Adım 8.** | Denklem 2.119' u kullanarak $x_{shade}$'i hesaplayın |
| **Adım 9.** | **if** rand $< 0{,}5$ ise |
| **Adım 10.** | Yazlık Tatil Yeri Aşaması |
| **Adım 11.** | **else** |
| **Adım 12.** | Rekabetçi Davranış Aşaması |
| **Adım 13.** | **end** |
| **Adım 14.** | **else** |
| **Adım 15.** | $p$, $Q$ değerlerini hesaplayın |
| **Adım 16.** | **if** $Q > 2$ ise |
| **Adım 17.** | Denklem 2.226 ile yiyeceği parçala |
| **Adım 18.** | Denklem 2.227 ile yiyeceği tüket |
| **Adım 19.** | **else** |
| **Adım 20.** | Denklem 2.228 ile yiyeceğe yönel |
| **Adım 21.** | **end** |
| **Adım 22.** | **end** |
| **Adım 23.** | $X_G, X_L$ uygunluk değerlerini güncelleyin |
| **Adım 24.** | $t = t + 1$ |
| **Adım 25.** | **End** |

# 3. TEST PROBLEMLERİNİN ÇÖZÜMÜNDE METASEZGİSELLERİN UYGULANMASI

## 3.1 Giriş

Bu bölüm, 2. bölümde önerilen güncel metasezgisel algoritmaların 23 farklı test fonksiyonları üzerindeki karşılaştırmalarını içermektedir. Bölümün temel katkıları aşağıda sıralanmıştır:

- Literatür taramasına göre bu metasezgisellerin simülasyon sonuçları, yakınsama hızları gibi farklı değerlendirme kriterleri ilk kez karşılaştırılmıştır.
- Algoritmaların simülasyon sonuçlarına göre en başarılı düşünülen beş metasezgiselin diğerleriyle başarısının kıyaslanması için tek kuyruk t-testi yapılmıştır.
- Deneysel çalışmalara göre AVOA ve MGO' nun diğer metasezgisellere üstünlük sağladığı değerlendirilmiştir.

Bölümün geri kalanında kıyaslama yapılan test fonksiyonları, parametreler, simülasyon sonuçları ve istatiksel test sonuçlarının verildiği deneysel çalışmalar sunulmuştur.

## 3.2 Deneysel Tasarım

Bu bölümde test fonksiyonlarına ve karşılaştırmada kullanılan algoritmaların kendilerine ait parametrelerine yer verilmiştir.

### 3.2.1 Fonksiyonlar

Bu bölümde, algoritmalar birçok çalışmada kullanılan 23 farklı test fonksiyonu ile kıyaslanmıştır [14]. Fonksiyonlar üç temel gruba ayrılmaktadır: Tek modlu, çok modlu ve farklı boyutlu çok modlu. Her bir grup sırasıyla Çizelge 3.2.1, Çizelge 3.2.2 ve Çizelge 3.2.3' te sunulmuştur. Tablolarda verilen aralık fonksiyonların arama uzayının sınırlarını $f_{min}$ optimum değerini temsil etmektedir.

**Çizelge 3.2.1** Tek modlu test fonksiyonları

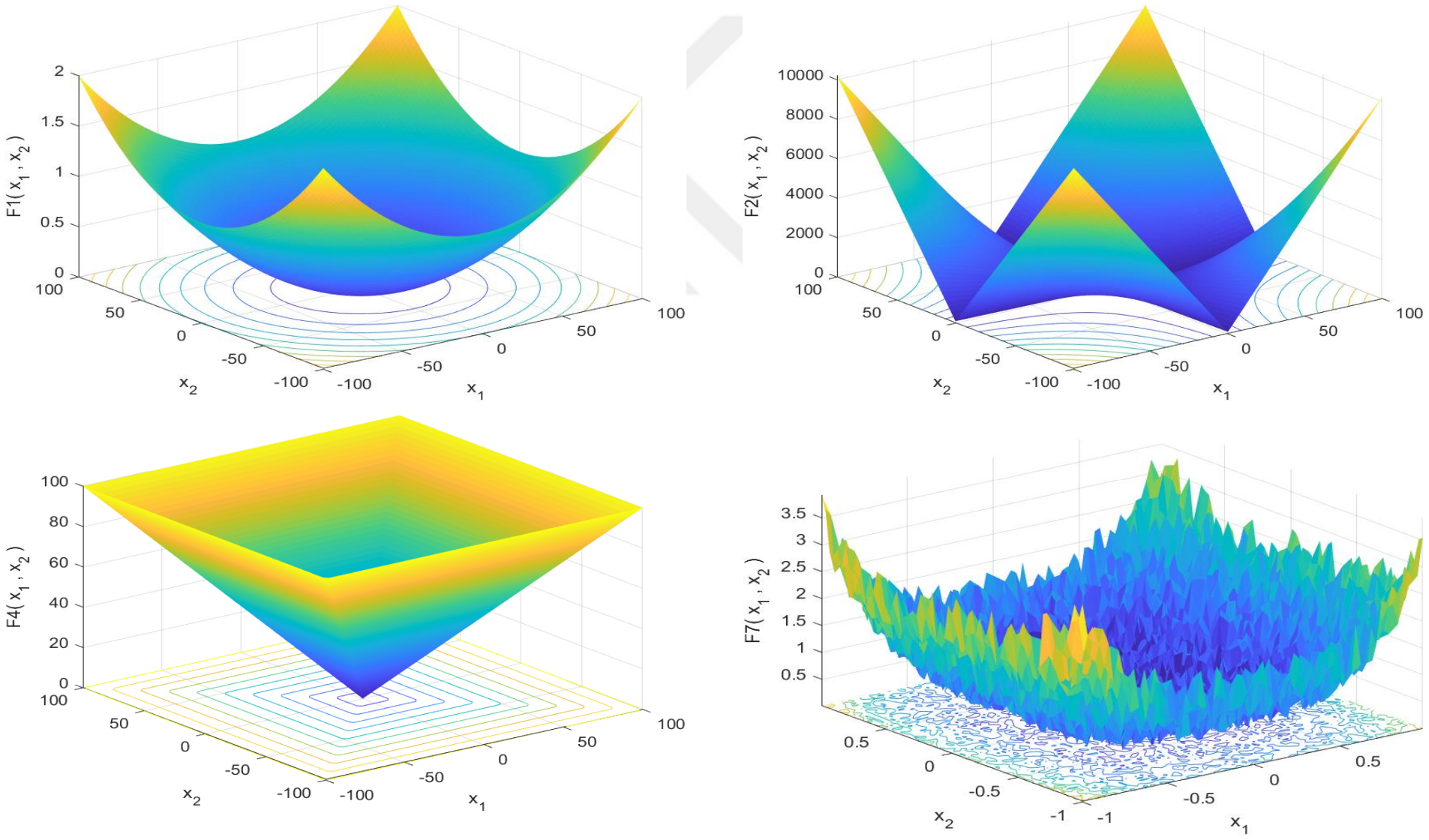
| **Fonksiyon** | **Boyut** | **Aralık** | $f_{min}$ |
|---|---|---|---|
| $f_1(x) = \sum_{i=1}^{n} x_i^2$ | 30 | [-100,100] | 0 |
| $f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | 30 | [-10,10] | 0 |
| $f_3(x) = \sum_{i=1}^{n} \left( \sum_{j-1}^{i} x_j \right)^2$ | 30 | [-100,100] | 0 |
| $f_4(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$ | 30 | [-100,100] | 0 |
| $f_5(x) = \sum_{i=1}^{n-1} \left[ 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$ | 30 | [-30,30] | 0 |
| $F_6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$ | 30 | $[-100,100]$ | 0 |
| $F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}\,[0,1)$ | 30 | $[-1{,}28, +1{,}28]$ | 0 |

**Çizelge 3.2.2** Çok modlu test fonksiyonları

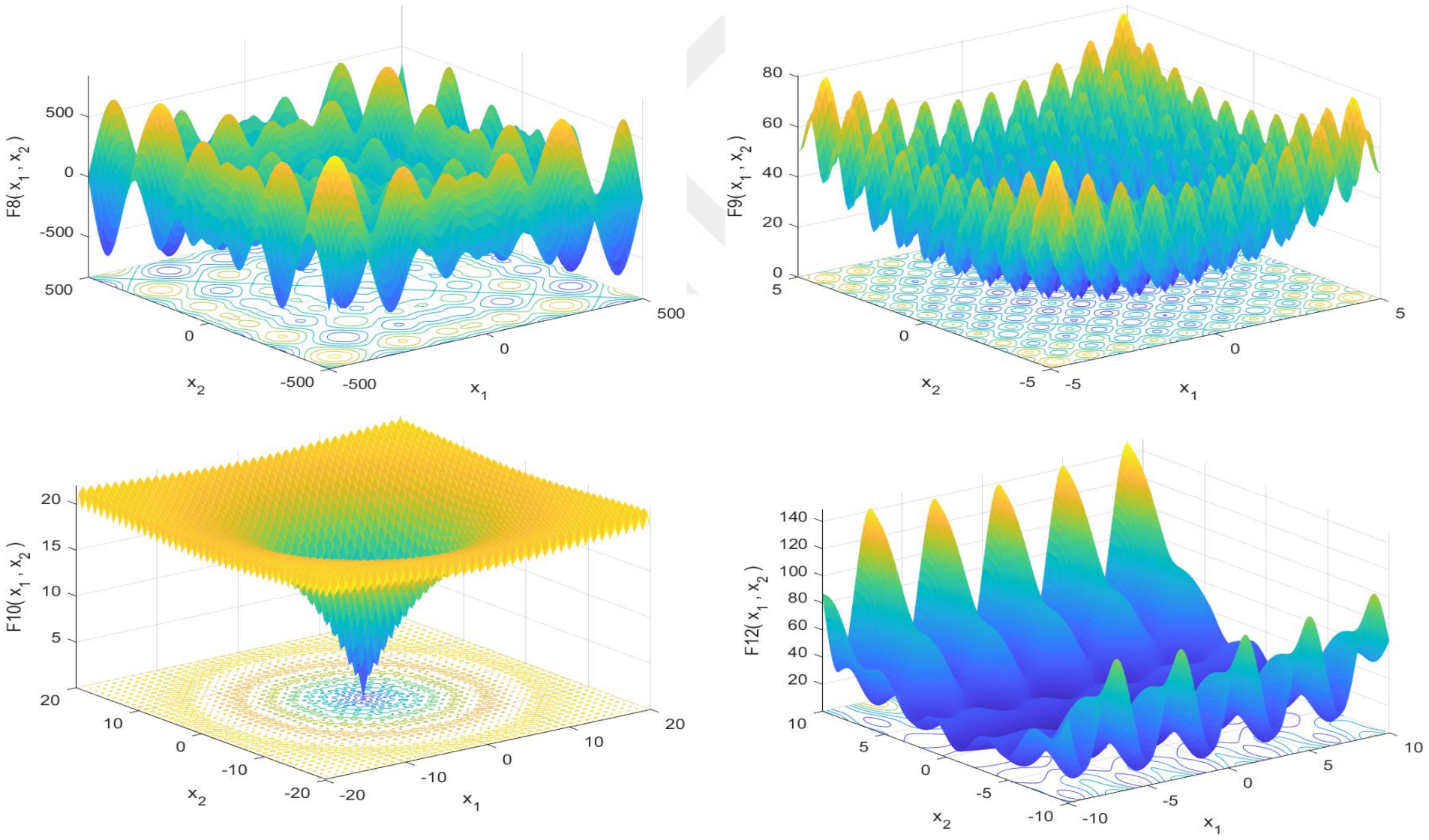
| Fonksiyon | Boyut | Aralık | $f_{min}$ |
|---|---|---|---|
| $F_8(x) = -\sum_{i=1}^{n} \left(x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$ | 30 | $[-500,500]$ | $-12569,5$ |
| $F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$ | 30 | $[-5.12,5.12]$ | 0 |
| $F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$ | 30 | $[-32,32]$ | 0 |
| $F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | 30 | $[-600,600]$ | 0 |
| $F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ | 30 | [-50,50] | 0 |
| $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ | | | |
| $F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | 30 | $[-50,50]$ | 0 |

**Çizelge 3.2.3** Farklı boyutlu çok modlu test fonksiyonları

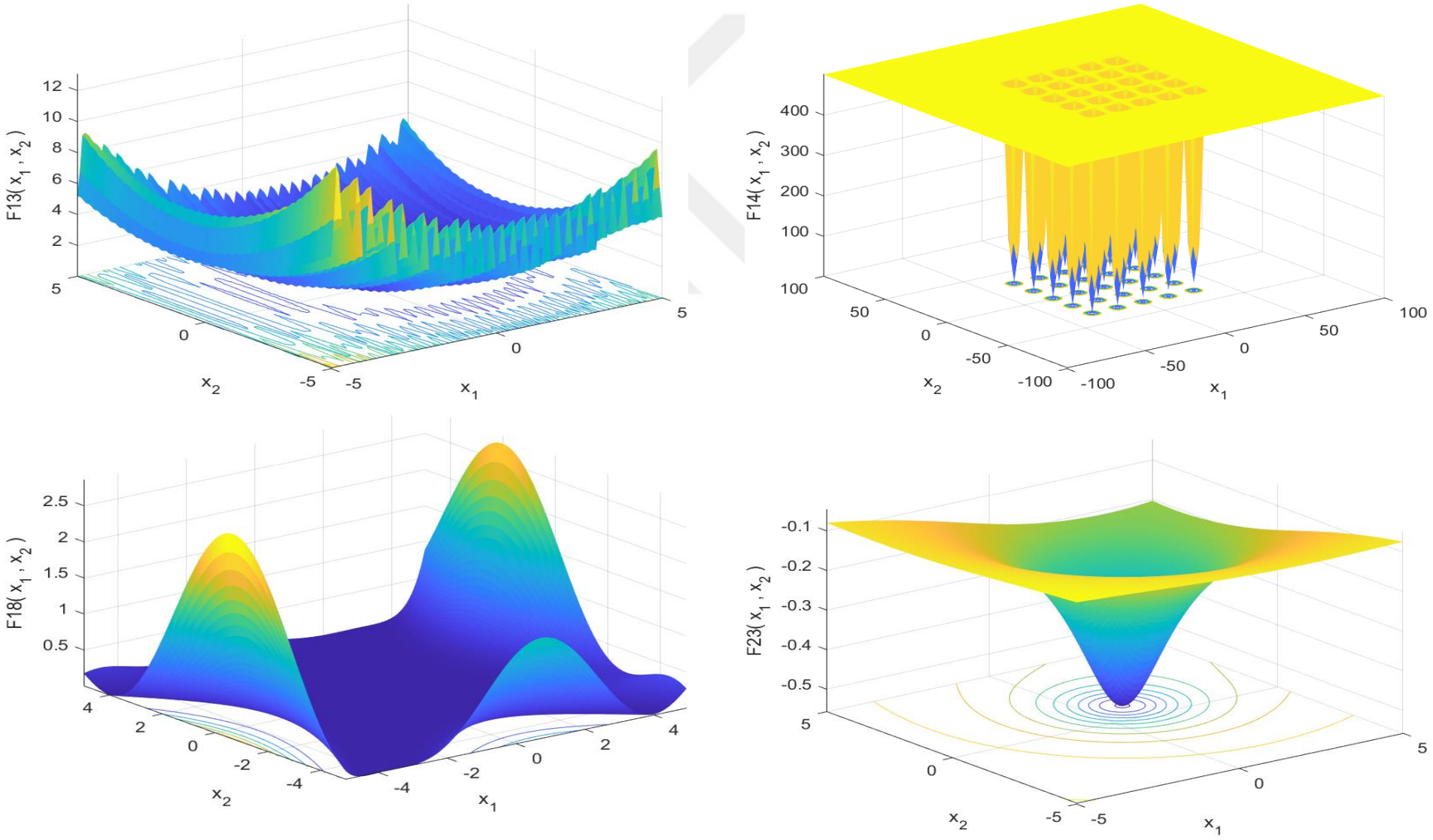
| Fonksiyon | Boyut | Aralık | $f_{min}$ |
|---|---|---|---|
| $F_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{j=1}^{2} (x_i - a_{ij})^6}\right]^{-1}$ | 2 | $[-65{,}536, 65{,}536]$ | 0,998 |
| $F_{15}(x) = \sum_{i=1}^{11} \left\lvert a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right\rvert^2$ | 4 | $[-5, 5]$ | $3{,}075 \times 10^{-4}$ |
| $F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ | 2 | $[-5, 5]$ | -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 | $[-5, 5]\ x\ [0, 15]$ | 0,398 |
| $F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 + 1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$ | 2 | $[-2, 2]$ | 3 |
| $F_{19}(x) = -\sum_{i=1}^{4} \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$ | 3 | $[0, 1]$ | -3,86 |
| $F_{20}(x) = -\sum_{i=1}^{4} \exp\left[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right]$ | 6 | $[0, 1]$ | $-3{,}322$ |
| $F_{21}(x) = -\sum_{i=1}^{5} \lvert(x_i - a_i)(x_i - a_i)^T + c_i\rvert^{-1}$ | 4 | [0, 10] | $-10{,}1532$ |
| $F_{22}(x) = -\sum_{i=1}^{7} \lvert(x_i - a_i)(x_i - a_i)^T + c_i\rvert^{-1}$ | 4 | $[0, 10]$ | $-10{,}4028$ |
| $F_{23}(x) = -\sum_{i=1}^{10} \lvert(x_i - a_i)(x_i - a_i)^T + c_i\rvert^{-1}$ | 4 | $[0, 10]$ | $-10{,}5363$ |




**Şekil 3.2.1** Tek modlu test fonksiyon grafikleri ($F_1, F_2, F_4, F_7$)




**Şekil 3.2.2** Çok modlu test fonksiyon grafikleri ($F_8, F_9, F_{10}, F_{12}$)




**Şekil 3.2.3** Farklı boyutlu çok modlu test fonksiyon grafikleri ($F_{13}, F_{14}, F_{18}, F_{23}$)

### 3.2.2 Parametreler

Çizelge 3.2.4 'teki parametre değerleri, kullanıldığı algoritmanın önerildiği makalelere ait kalite test fonksiyonları için tavsiye edilen değerlerdir. Bu çalışmada da aynı test fonksiyonları kullanıldığından aynı değerler tercih edilmiştir. Adil bir karşılaştırma olması için tüm algoritmalarda popülasyon büyüklüğü 50, maksimum iterasyon sayısı 1000 olarak belirlenmiştir.

**Çizelge 3.2.4** Algoritma parametreleri ve değerleri

| Algoritma | Simge | Anlam | Değer |
|---|---|---|---|
| **AO** | $\alpha$ | Keşif parametresi | 0,1 |
| | $\delta$ | Sömürü parametresi | 0,1 |
| **AOA** | $\alpha$ | İterasyon doğruluğu | 5 |
| | $\mu$ | Arama doğruluğu düzenleyicisi | 0,5 |
| **BWO** | $P_P$ | Üreme hızı | 0,6 |
| | $C_R$ | Yamyamlık oranı | 0,44 |
| | $P_M$ | Mutasyon oranı | 0,4 |
| **STOA** | $C_f$ | Çarpışma kontrolü | 2 |
| | $C_B$ | Keşif parametresi | $[0-0,5]$ |
| | $u$ | Spiral davranış parametresi | 1 |
| | $v$ | Spiral davranış parametresi | 1 |
| **AVOA** | $L_1$ | Arama öncesi parametre | 0,8 |
| | $L_2$ | Arama öncesi parametre | 0,2 |
| | $w$ | Operasyon aşama bilgisi | 2,5 |
| | $p_1$ | Keşif yöntemi belirleme parametresi | 0,6 |
| | $p_2$ | Birinci aşama seçimi | 0,4 |
| | $p_3$ | İkinci aşama seçimi | 0,6 |
| **PDO** | $\rho$ | Yiyecek kaynağı alarmı | 0,1 |
| | $\varepsilon$ | Besin kaynağı kalite göstergesi | $2,22E-12$ |
| | $\Delta$ | Çayır köpekleri konum farkı | 0,005 |
| **COA** | $C_1$ | Kerevit alım kontrolü | 0,2 |
| | $C_3$ | En büyük yiyecek | 3 |
| | $\mu$ | Uygun sıcaklık değeri | 25 |
| | $\sigma$ | Kerevit alım kontrolü | 3 |

## 3.3 Sonuçlar

### 3.3.1 Simülasyon Sonuçları

Çalışmada tüm algoritmalar 100 kez bağımsız koşturulmuştur. Algoritmalar MATLAB 2022b platformu kullanılarak yazılmıştır. Kullanılan bilgisayar, Ryzen 9 3950X, 3.5 GHz hıza ve 64 GB RAM' e sahiptir. Simülasyon sonuçları sırasıyla tek modlu fonksiyonlar için Çizelge 3.3.1 ve Çizelge 3.3.2' da çok modlu fonksiyonlar için Çizelge 3.3.3 ve farklı boyutlu çok modlu fonksiyonlar için Çizelge 3.3.4 ve Çizelge 3.3.5' da paylaşılmıştır. Her fonksiyon için koşmaların ortalama, standart sapma, en iyi ve en kötü sonuçları bu çizelgelerde verilmiştir. Ayrıca algoritmaların test fonksiyonlarında göstermiş olduğu performans sıralamaları ve genel toplamları Çizelge 3.3.6' da yer almaktadır.

**Çizelge 3.3.1** Tek modlu test fonksiyonları ($F_1$-$F_4$)

| | | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_1$ | **ORTALAMA** | 6,34E-208 | 0,00E+00 | 3,23E-124 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 5,82E-192 | 4,48E-167 | 0,00E+00 | 8,13E-19 |
| | **STD SAPMA** | 0,00E+00 | 0,00E+00 | 3,20E-123 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 2,30E-18 |
| | **MİNİMUM** | 0,00E+00 | 0,00E+00 | 4,33E-144 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 6,82E-219 | 1,68E-183 | 0,00E+00 | 1,40E-22 |
| | **MAKSİMUM** | 6,34E-206 | 0,00E+00 | 3,20E-122 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 4,45E-190 | 4,16E-165 | 0,00E+00 | 1,68E-17 |
| | **SIRALAMA** | 2 | 1 | 5 | 1 | 1 | 1 | 3 | 4 | 1 | 6 |
| $F_2$ | **ORTALAMA** | 1,28E-121 | 0,00E+00 | 2,07E-68 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 9,29E-100 | 3,72E-96 | 0,00E+00 | 8,69E-13 |
| | **STD SAPMA** | 1,28E-120 | 0,00E+00 | 1,39E-67 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 9,27E-99 | 1,74E-95 | 0,00E+00 | 1,23E-12 |
| | **MİNİMUM** | 7,38E-158 | 0,00E+00 | 4,79E-77 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,51E-120 | 1,12E-102 | 0,00E+00 | 9,90E-15 |
| | **MAKSİMUM** | 1,28E-119 | 0,00E+00 | 1,13E-66 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 9,27E-98 | 1,41E-94 | 0,00E+00 | 7,55E-12 |
| | **SIRALAMA** | 2 | 1 | 5 | 1 | 1 | 1 | 3 | 4 | 1 | 6 |
| $F_3$ | **ORTALAMA** | 2,86E-196 | 0,00E+00 | 1,63E-94 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,86E-153 | 2,01E-21 | 0,00E+00 | 1,35E-09 |
| | **STD SAPMA** | 0,00E+00 | 0,00E+00 | 1,63E-93 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,86E-152 | 1,70E-20 | 0,00E+00 | 2,71E-09 |
| | **MİNİMUM** | 0,00E+00 | 0,00E+00 | 8,78E-117 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 1,52E-198 | 3,64E-33 | 0,00E+00 | 3,65E-12 |
| | **MAKSİMUM** | 2,86E-194 | 0,00E+00 | 1,63E-92 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,86E-151 | 1,69E-19 | 0,00E+00 | 2,15E-08 |
| | **SIRALAMA** | 2 | 1 | 4 | 1 | 1 | 1 | 3 | 5 | 1 | 6 |
| $F_4$ | **ORTALAMA** | 7,91E-145 | 0,00E+00 | 1,60E-52 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,95E-96 | 6,38E-55 | 0,00E+00 | 4,86E-06 |
| | **STD SAPMA** | 7,89E-144 | 0,00E+00 | 7,09E-52 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 7,26E-95 | 4,82E-54 | 0,00E+00 | 8,43E-06 |
| | **MİNİMUM** | 2,41E-158 | 0,00E+00 | 4,14E-62 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,88E-111 | 5,51E-65 | 0,00E+00 | 2,51E-07 |
| | **MAKSİMUM** | 7,89E-143 | 0,00E+00 | 4,54E-51 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 7,21E-94 | 4,74E-53 | 0,00E+00 | 7,19E-05 |
| | **SIRALAMA** | 2 | 1 | 5 | 1 | 1 | 1 | 3 | 4 | 1 | 6 |

**Çizelge 3.3.2** Tek modlu test fonksiyonları ($F_5$-$F_7$)

| | | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_5$ | **ORTALAMA** | 5,35E-04 | 5,20E+00 | 4,60E-03 | 1,71E-06 | 2,89E+01 | 2,44E+01 | 1,44E-03 | 8,76E-31 | 7,87E+00 | 2,77E+01 |
| | **STD SAPMA** | 9,16E-04 | 2,77E-01 | 8,56E-03 | 1,70E-06 | 3,50E-02 | 4,84E-01 | 2,30E-03 | 3,08E-30 | 1,06E+01 | 6,37E-01 |
| | **MİNİMUM** | 6,11E-07 | 4,53E+00 | 7,78E-05 | 4,22E-08 | 2,88E+01 | 2,34E+01 | 4,50E-07 | 0,00E+00 | 2,90E-01 | 2,61E+01 |
| | **MAKSİMUM** | 6,19E-03 | 5,92E+00 | 6,39E-02 | 9,54E-06 | 2,90E+01 | 2,63E+01 | 1,35E-02 | 2,50E-29 | 2,90E+01 | 2,88E+01 |
| | **SIRALAMA** | 3 | 6 | 5 | 2 | 10 | 8 | 4 | 1 | 7 | 9 |
| $F_6$ | **ORTALAMA** | 0,00E+00 | 1,41E-02 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 2,16E+00 |
| | **STD SAPMA** | 0,00E+00 | 4,80E-03 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 5,04E-01 |
| | **MİNİMUM** | 0,00E+00 | 3,83E-03 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 7,52E-01 |
| | **MAKSİMUM** | 0,00E+00 | 2,65E-02 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 3,27E+00 |
| | **SIRALAMA** | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| $F_7$ | **ORTALAMA** | 3,39E-05 | 1,98E-05 | 2,62E-04 | 4,86E-05 | 4,45E-05 | 2,14E-05 | 4,94E-05 | 1,45E-04 | 2,92E-05 | 1,55E-03 |
| | **STD SAPMA** | 3,11E-05 | 1,88E-05 | 1,97E-04 | 4,91E-05 | 3,70E-05 | 2,23E-05 | 4,94E-05 | 1,07E-04 | 2,64E-05 | 1,22E-03 |
| | **MİNİMUM** | 1,91E-07 | 1,81E-07 | 8,32E-06 | 5,53E-07 | 1,45E-06 | 1,61E-07 | 7,97E-08 | 6,40E-06 | 4,17E-09 | 1,73E-04 |
| | **MAKSİMUM** | 1,43E-04 | 1,01E-04 | 1,07E-03 | 2,42E-04 | 1,68E-04 | 1,11E-04 | 3,34E-04 | 5,68E-04 | 1,32E-04 | 6,01E-03 |
| | **SIRALAMA** | 4 | 1 | 9 | 6 | 5 | 2 | 7 | 8 | 3 | 10 |

**Çizelge 3.3.3** Çok modlu test fonksiyonları ($F_8$-$F_{13}$)

| | | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_8$ | ORTALAMA | -9,85E+03 | -3,44E+03 | -1,11E+04 | -1,25E+04 | -4,94E+03 | -9,36E+03 | -1,26E+04 | -1,26E+04 | -4,07E+03 | -5,67E+03 |
| | STD SAPMA | 3,31E+03 | 2,53E+02 | 4,29E+02 | 2,00E+02 | 6,24E+02 | 5,74E+02 | 4,76E+01 | 1,04E-09 | 2,98E+02 | 6,18E+02 |
| | MİNİMUM | -1,26E+04 | -3,95E+03 | -1,21E+04 | -1,26E+04 | -6,46E+03 | -1,09E+04 | -1,26E+04 | -1,26E+04 | -4,92E+03 | -7,98E+03 |
| | MAKSİMUM | -4,58E+03 | -2,86E+03 | -1,02E+04 | -1,09E+04 | -3,54E+03 | -8,25E+03 | -1,21E+04 | -1,26E+04 | -3,41E+03 | -4,63E+03 |
| | SIRALAMA | 5 | 10 | 4 | 3 | 8 | 6 | 2 | 1 | 9 | 7 |
| $F_9$ | ORTALAMA | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 7,29E-01 |
| | STD SAPMA | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 2,16E+00 |
| | MİNİMUM | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 |
| | MAKSİMUM | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 1,39E+01 |
| | SIRALAMA | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| $F_{10}$ | ORTALAMA | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 5,15E-16 | 4,44E-16 | 2,00E+01 |
| | STD SAPMA | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 5,00E-16 | 0,00E+00 | 1,83E-03 |
| | MİNİMUM | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 2,00E+01 |
| | MAKSİMUM | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,44E-16 | 4,00E-15 | 4,44E-16 | 2,00E+01 |
| | SIRALAMA | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 3 |
| $F_{11}$ | ORTALAMA | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 7,03E-03 |
| | STD SAPMA | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 1,32E-02 |
| | MİNİMUM | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 |
| | MAKSİMUM | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 8,35E-02 |
| | SIRALAMA | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| $F_{12}$ | ORTALAMA | 1,95E-07 | 5,94E-01 | 7,58E-08 | 1,06E-11 | 6,74E-01 | 2,56E-04 | 8,41E-07 | 1,57E-32 | 5,89E-01 | 1,61E-01 |
| | STD SAPMA | 3,60E-07 | 3,48E-02 | 6,51E-08 | 7,66E-12 | 2,29E-01 | 1,24E-03 | 1,50E-06 | 5,50E-48 | 5,35E-01 | 8,58E-02 |
| | MİNİMUM | 7,94E-12 | 4,78E-01 | 7,82E-09 | 1,70E-12 | 2,40E-01 | 6,71E-08 | 1,99E-11 | 1,57E-32 | 1,04E-02 | 5,67E-02 |
| | MAKSİMUM | 1,95E-06 | 7,11E-01 | 3,76E-07 | 4,64E-11 | 1,34E+00 | 6,50E-03 | 1,13E-05 | 1,57E-32 | 1,60E+00 | 6,17E-01 |
| | SIRALAMA | 4 | 9 | 3 | 2 | 10 | 6 | 5 | 1 | 8 | 7 |
| $F_{13}$ | ORTALAMA | 2,45E-06 | 8,22E-01 | 9,25E-04 | 5,22E-10 | 2,94E+00 | 1,49E+00 | 8,52E-06 | 1,35E-32 | 2,93E+00 | 1,56E+00 |
| | STD SAPMA | 3,67E-06 | 1,46E-01 | 4,87E-03 | 3,88E-10 | 1,83E-01 | 3,29E-01 | 1,27E-05 | 5,50E-48 | 2,33E-01 | 2,26E-01 |
| | MİNİMUM | 2,17E-09 | 4,56E-01 | 3,44E-08 | 3,59E-11 | 1,94E+00 | 5,47E-01 | 2,16E-09 | 1,35E-32 | 1,61E+00 | 9,08E-01 |
| | MAKSİMUM | 2,77E-05 | 9,95E-01 | 4,39E-02 | 1,82E-09 | 3,00E+00 | 2,67E+00 | 8,63E-05 | 1,35E-32 | 3,00E+00 | 2,09E+00 |
| | SIRALAMA | 3 | 6 | 5 | 2 | 10 | 7 | 4 | 1 | 9 | 8 |

**Çizelge 3.3.4** Farklı boyutlu çok modlu test fonksiyonları ($F_{14}$-$F_{18}$))

| | | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_{14}$ | ORTALAMA | 1,68E+00 | 9,08E+00 | 9,98E-01 | 1,04E+00 | 1,97E+00 | 2,64E+00 | 1,05E+00 | 9,98E-01 | 3,55E+00 | 1,33E+00 |
| | STD SAPMA | 2,17E+00 | 3,94E+00 | 0,00E+00 | 2,79E-01 | 2,18E+00 | 3,30E+00 | 2,18E-01 | 1,48E-16 | 3,58E+00 | 1,15E+00 |
| | MİNİMUM | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 | 9,98E-01 |
| | MAKSİMUM | 1,27E+01 | 1,27E+01 | 9,98E-01 | 2,98E+00 | 1,27E+01 | 1,27E+01 | 1,99E+00 | 9,98E-01 | 1,27E+01 | 1,08E+01 |
| | SIRALAMA | 5 | 9 | 1 | 2 | 6 | 7 | 3 | 1 | 8 | 4 |
| $F_{15}$ | ORTALAMA | 4,17E-04 | 1,26E-02 | 3,07E-04 | 3,26E-04 | 5,38E-03 | 2,07E-03 | 3,20E-04 | 3,35E-04 | 9,68E-04 | 1,18E-03 |
| | STD SAPMA | 1,07E-04 | 2,43E-02 | 1,77E-16 | 1,29E-04 | 1,25E-02 | 5,43E-03 | 1,24E-05 | 1,57E-04 | 4,53E-04 | 2,01E-04 |
| | MİNİMUM | 3,19E-04 | 3,20E-04 | 3,07E-04 | 3,07E-04 | 3,07E-04 | 3,07E-04 | 3,08E-04 | 3,07E-04 | 3,08E-04 | 3,08E-04 |
| | MAKSİMUM | 1,24E-03 | 1,30E-01 | 3,07E-04 | 1,22E-03 | 6,33E-02 | 2,04E-02 | 3,60E-04 | 1,22E-03 | 2,26E-03 | 1,31E-03 |
| | SIRALAMA | 5 | 10 | 1 | 3 | 7 | 8 | 4 | 2 | 6 | 9 |
| $F_{16}$ | ORTALAMA | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 |
| | STD SAPMA | 1,25E-04 | 4,90E-08 | 6,53E-16 | 4,99E-16 | 4,99E-16 | 4,76E-16 | 1,36E-12 | 6,41E-16 | 4,33E-04 | 3,45E-07 |
| | MİNİMUM | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 |
| | MAKSİMUM | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 | -1,03E+00 |
| | SIRALAMA | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $F_{17}$ | ORTALAMA | 3,98E-01 | 4,03E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 4,02E-01 | 3,98E-01 |
| | STD SAPMA | 1,46E-04 | 4,13E-03 | 0,00E+00 | 0,00E+00 | 0,00E+00 | 3,57E-10 | 1,29E-07 | 0,00E+00 | 1,92E-02 | 4,94E-05 |
| | MİNİMUM | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 |
| | MAKSİMUM | 3,99E-01 | 4,20E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 3,98E-01 | 5,80E-01 | 3,98E-01 |
| | SIRALAMA | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| $F_{18}$ | ORTALAMA | 3,01E+00 | 7,23E+00 | 3,00E+00 | 3,00E+00 | 3,81E+00 | 3,00E+00 | 3,00E+00 | 3,00E+00 | 3,00E+00 | 3,00E+00 |
| | STD SAPMA | 9,11E-03 | 9,75E+00 | 9,30E-16 | 1,33E-07 | 4,63E+00 | 3,76E-14 | 1,70E-09 | 1,21E-15 | 1,16E-13 | 1,13E-05 |
| | MİNİMUM | 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 |
| | MAKSİMUM | 3,04E+00 | 3,00E+01 | 3,00E+00 | 3,00E+00 | 3,00E+01 | 3,00E+00 | 3,00E+00 | 3,00E+00 | 3,00E+00 | 3,00E+00 |
| | SIRALAMA | 2 | 4 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |

**Çizelge 3.3.5** Farklı boyutlu çok modlu test fonksiyonları ($F_{19}$-$F_{23}$)

| | | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $F_{19}$ | ORTALAMA | -3,86E+00 | -3,85E+00 | -3,86E+00 | -3,86E+00 | -3,85E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 |
| | STD SAPMA | 2,39E-03 | 2,86E-03 | 2,23E-15 | 1,73E-15 | 1,09E-01 | 3,52E-15 | 8,52E-04 | 2,17E-15 | 3,71E-03 | 9,35E-04 |
| | MİNİMUM | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 |
| | MAKSİMUM | -3,85E+00 | -3,84E+00 | -3,86E+00 | -3,86E+00 | -3,09E+00 | -3,86E+00 | -3,86E+00 | -3,86E+00 | -3,85E+00 | -3,85E+00 |
| | SIRALAMA | 3 | 6 | 1 | 1 | 7 | 1 | 2 | 1 | 4 | 5 |
| $F_{20}$ | ORTALAMA | -3,20E+00 | -3,12E+00 | -3,31E+00 | -3,27E+00 | -3,18E+00 | -3,28E+00 | -3,18E+00 | -3,26E+00 | -3,05E+00 | -2,98E+00 |
| | STD SAPMA | 7,34E-02 | 6,72E-02 | 3,74E-02 | 5,94E-02 | 1,28E-01 | 5,84E-02 | 8,37E-02 | 5,93E-02 | 2,62E-01 | 2,96E-01 |
| | MİNİMUM | -3,31E+00 | -3,28E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 | -3,32E+00 |
| | MAKSİMUM | -2,98E+00 | -2,87E+00 | -3,20E+00 | -3,20E+00 | -2,75E+00 | -3,20E+00 | -2,89E+00 | -3,20E+00 | -1,58E+00 | -1,16E+00 |
| | SIRALAMA | 5 | 8 | 1 | 3 | 7 | 2 | 6 | 4 | 9 | 10 |
| $F_{21}$ | ORTALAMA | -1,02E+01 | -4,02E+00 | -1,02E+01 | -1,02E+01 | -8,01E+00 | -7,71E+00 | -5,26E+00 | -1,02E+01 | -5,59E+00 | -5,17E+00 |
| | STD SAPMA | 2,00E-03 | 1,18E+00 | 1,06E-05 | 2,94E-15 | 2,58E+00 | 2,56E+00 | 9,94E-01 | 4,46E-15 | 2,29E+00 | 4,29E+00 |
| | MİNİMUM | -1,02E+01 | -8,04E+00 | -1,02E+01 | -1,02E+01 | -1,02E+01 | -1,02E+01 | -1,01E+01 | -1,02E+01 | -1,02E+01 | -1,02E+01 |
| | MAKSİMUM | -1,01E+01 | -1,58E+00 | -1,02E+01 | -1,02E+01 | -2,63E+00 | -5,06E+00 | -5,05E+00 | -1,02E+01 | -2,73E+00 | -3,51E-01 |
| | SIRALAMA | 2 | 8 | 1 | 1 | 3 | 4 | 6 | 1 | 5 | 7 |
| $F_{22}$ | ORTALAMA | -1,04E+01 | -4,57E+00 | -1,03E+01 | -1,04E+01 | -7,76E+00 | -8,00E+00 | -5,62E+00 | -1,04E+01 | -6,06E+00 | -7,14E+00 |
| | STD SAPMA | 4,64E-03 | 1,52E+00 | 7,48E-01 | 3,70E-15 | 2,87E+00 | 2,75E+00 | 1,59E+00 | 3,23E-15 | 2,91E+00 | 4,13E+00 |
| | MİNİMUM | -1,04E+01 | -9,11E+00 | -1,04E+01 | -1,04E+01 | -1,04E+01 | -1,04E+01 | -1,04E+01 | -1,04E+01 | -1,04E+01 | -1,04E+01 |
| | MAKSİMUM | -1,04E+01 | -1,47E+00 | -5,09E+00 | -1,04E+01 | -9,10E-01 | -2,77E+00 | -5,09E+00 | -1,04E+01 | -2,18E+00 | -5,21E-01 |
| | SIRALAMA | 3 | 9 | 2 | 1 | 5 | 4 | 8 | 1 | 7 | 6 |
| $F_{23}$ | ORTALAMA | -1,05E+01 | -4,41E+00 | -1,05E+01 | -1,05E+01 | -8,24E+00 | -8,35E+00 | -5,34E+00 | -1,05E+01 | -5,94E+00 | -7,96E+00 |
| | STD SAPMA | 2,10E-03 | 1,57E+00 | 6,70E-01 | 1,73E-15 | 2,87E+00 | 2,92E+00 | 1,06E+00 | 1,44E-15 | 2,94E+00 | 3,68E+00 |
| | MİNİMUM | -1,05E+01 | -1,02E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 | -1,05E+01 |
| | MAKSİMUM | -1,05E+01 | -1,53E+00 | -3,84E+00 | -1,05E+01 | -2,43E+00 | -2,42E+00 | -5,13E+00 | -1,05E+01 | -2,55E+00 | -5,54E-01 |
| | SIRALAMA | 2 | 9 | 3 | 1 | 5 | 4 | 8 | 1 | 7 | 6 |

**Çizelge 3.3.6** Test fonksiyonları karşılaştırması

| ALGORİTMALAR | AO | AOA | ARO | AVOA | BWO | COA | HHO | MGO | PDO | STOA |
|---|---|---|---|---|---|---|---|---|---|---|
| **TEK MODLU SIRALAMA TOPLAM** | 16 | 13 | 34 | 13 | 20 | 15 | 24 | 27 | 15 | 46 |
| **ÇOK MODLU SIRALAMA TOPLAM** | 15 | 28 | 15 | 10 | 31 | 22 | 14 | 7 | 29 | 29 |
| **FARKLI BOYUTLU ÇOK MODLU SIRALAMA TOPLAM** | 30 | 67 | 13 | 15 | 45 | 33 | 40 | 14 | 50 | 50 |
| **GENEL TOPLAM** | **61** | **108** | **62** | **38** | **96** | **70** | **78** | **48** | **94** | **125** |

Tek modlu test fonksiyon sonuçlarına göre aşağıdaki değerlendirmeler Tablo 3.3.1 ve Tablo 3.3.2' den çıkarılmıştır:

- $F_1$, $F_2$, $F_{13}$ ve $F_{14}$ fonksiyonlarında AOA, AVOA, BWO, COA ve PDO algoritmaları tüm koşmalarda optimum sonuca ulaşmıştır.
- Algoritmaların başarı sıralamasını değiştiren iki fonksiyon $F_5$ ve $F_6$ dır. $F_5$' te MGO en başarılı olurken, $F_6$' da AOA ve STOA en başarısız olmuştur.
- STOA genel olarak bakıldığında ya sonuncu ya da sondan bir önceki olmuştur. Ayrıca başarı sıralamalarında 46 değeri ile en kötü sıralamaya sahiptir.
- AOA ve AVOA başarı sıralamasında eşit puanlar alarak 1. olmuşlardır. Bu iki algoritmayı 15 sıralama değeri ile PDO, 16 sıralama değeri ile AO takip etmektedir.
- $F_5$ ve $F_7$' de hiçbir algoritma $F_{min}$ değerine ulaşamamıştır. $F_7$' de ortalamada sırasıyla en iyi ve en kötü değeri üreten AOA ve STOA arasında $0{,}0015$' lik çok küçük bir fark bulunmaktadır.

Çok modlu test fonksiyon sonuçlarına göre aşağıdaki değerlendirmeler Tablo 3.3.3' den çıkarılmıştır:

- $F_8$ fonksiyonunun optimum değeri olan $-12569{,}5$' i hiçbir algoritma elde edememiştir; ancak MGO yaklaşık $0{,}01$' lik farkla bu değere yaklaşmıştır.
- Sıralama toplamlarına bakıldığında MGO, 6 fonksiyonun sadece 1 tanesinde 2. diğerlerinde 1. olmuştur.
- MGO' yu en yakın 3 puan farkla AVOA takip etmektedir.
- $F_9$ ve $F_{11}$ için sadece STOA optimum değeri elde edememiştir.
- Tek modlu fonksiyonlarda PDO 15 sıralama toplamı ile 2. sırada tamamlarken çok modlu fonksiyonlarda 29 sıralama puanı ile 7. sırada tamamlamıştır.

Farklı boyutlu çok modlu test fonksiyon sonuçlarına göre aşağıdaki değerlendirmeler Tablo 3.3.4 ve Tablo 3.3.5' ten çıkarılmıştır:

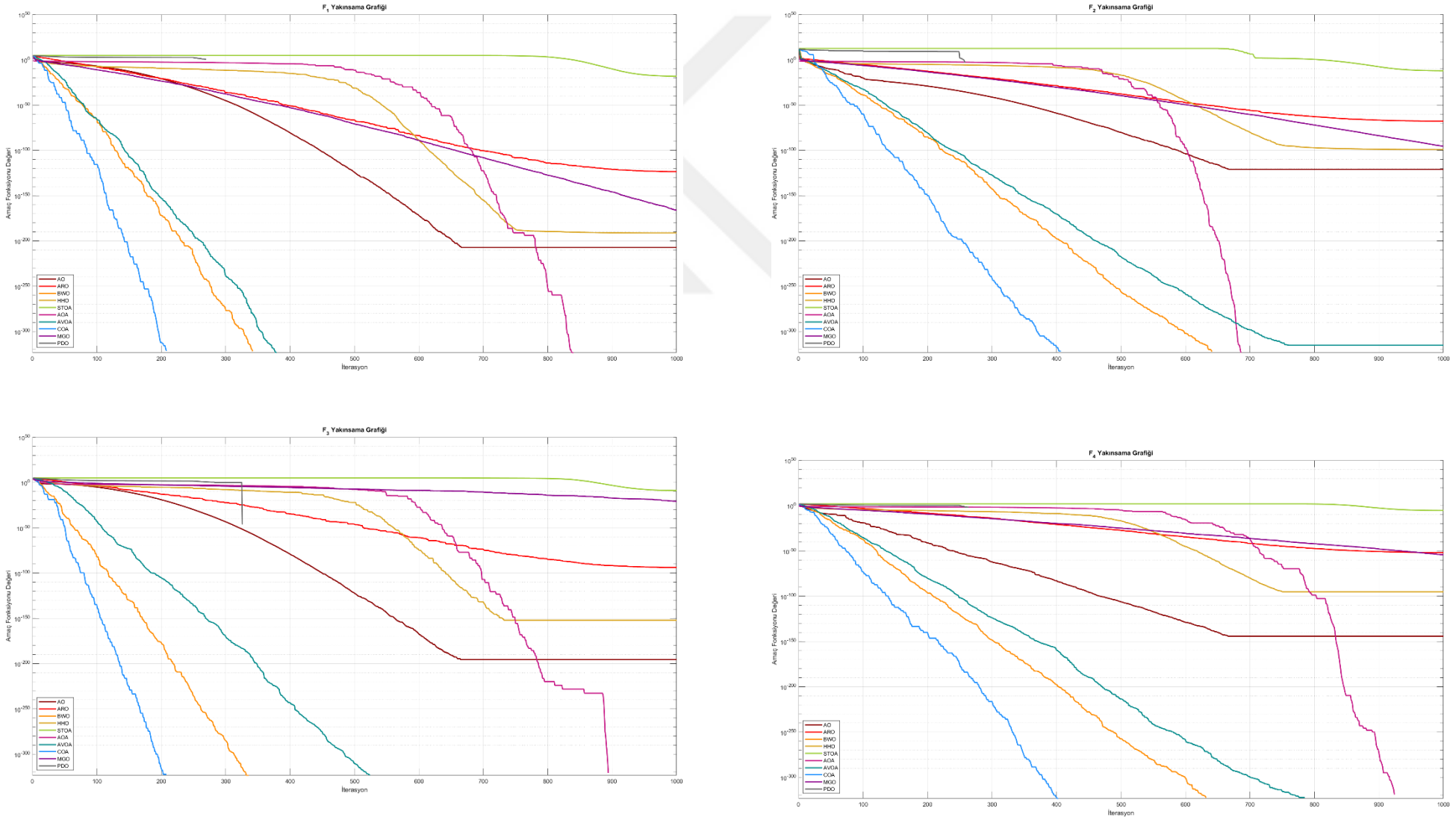
- $F_{16}$, $F_{17}$ ve $F_{18}$ fonksiyonlarında neredeyse tüm algoritmalar optimum sonucu elde etmiştir.
- $F_{20}$ fonksiyonunda sadece optimum değere ulaşan algoritma ARO olmuştur.

- $F_{21}$, $F_{22}$ ve $F_{23}$ fonksiyonlarında AVOA ve MGO optimum değerleri elde etmişlerdir.
- Sıralama toplamlarına bakıldığında ilk üç sırasıyla ARO, MGO ve AVOA' dır.

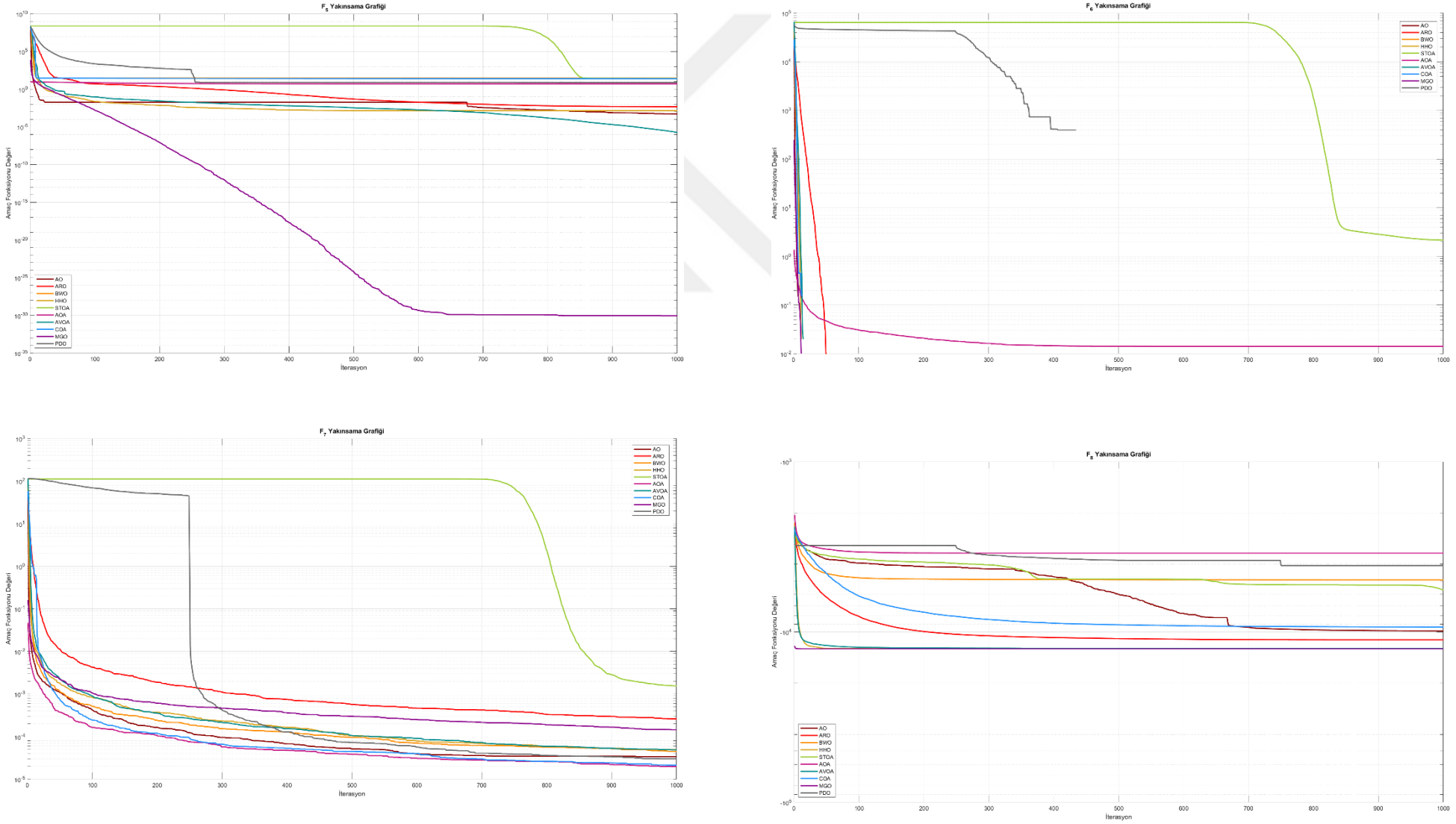
Tek modlu, çok modlu ve farklı boyutlu çok modlu fonksiyon sıralama değeri birlikte değerlendirildiğinde 1. AVOA olmuştur. AVOA' yı sırasıyla MGO, AO, ARO ve COA takip etmiştir. Tüm algoritmalar içerisinde toplam 125 sıralama puanı ile en başarısız olan algoritma STOA' dır

### 3.3.2 Yakınsama Grafikleri Analizi

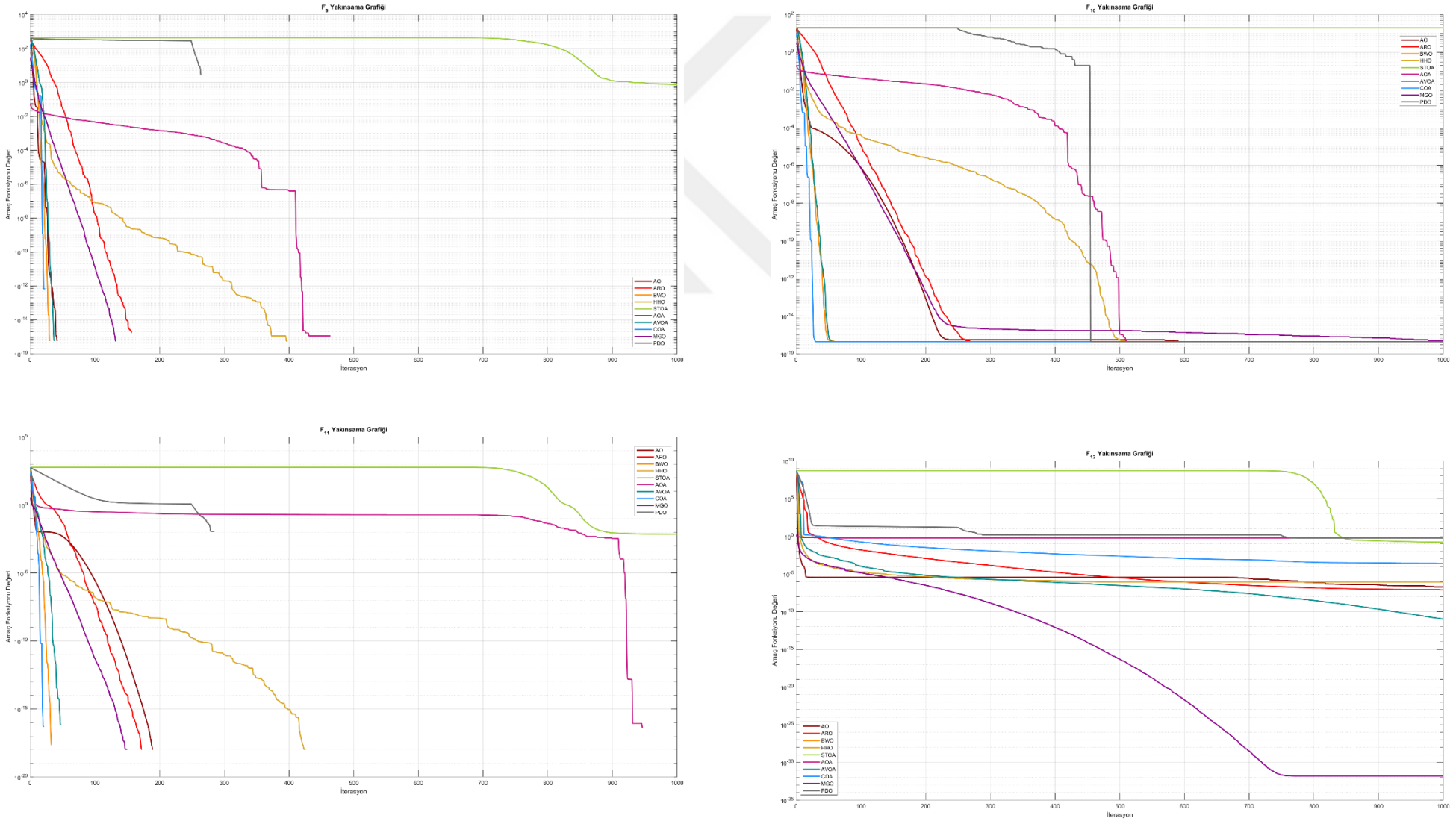
Yakınsama grafikleri, bir algoritmanın başlangıç çözümlerinin iterasyonlar boyunca gelişimini takip ederek çözümlerin değişimini gösteren grafiklerdir. Bu alt bölümde çalışmada kullanılan tüm algoritmaların her bir test fonksiyonu üzerindeki yakınsama grafikleri çizilerek şekillerle sunulmuştur. Yakınsama grafikleri algoritmaların tüm koşmalarının her iterasyondaki ortalamaları alınarak çizdirilmiştir.



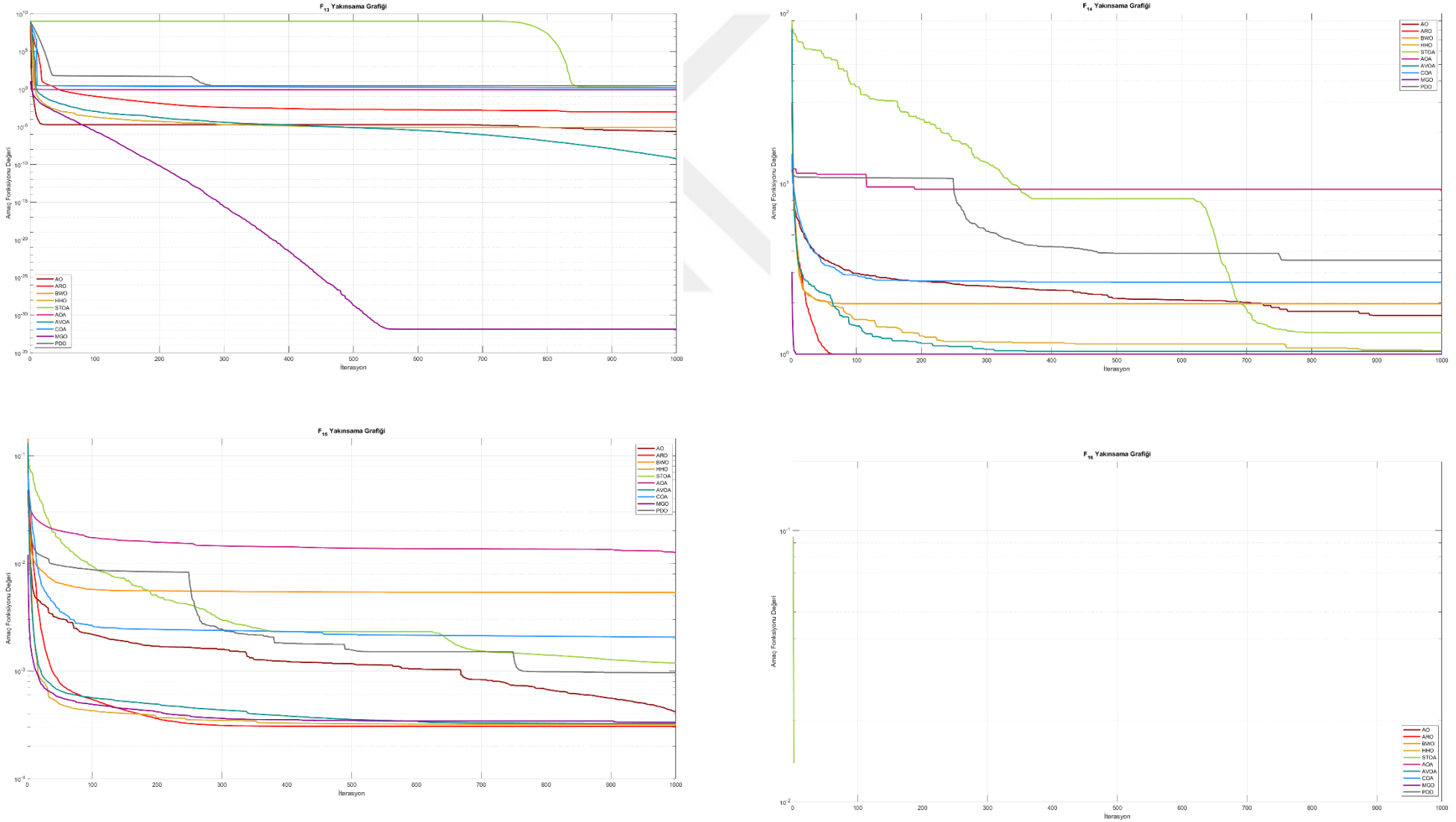

**Şekil 3.3.1** Fonksiyonların yakınsama grafikleri ($F_1 - F_4$)



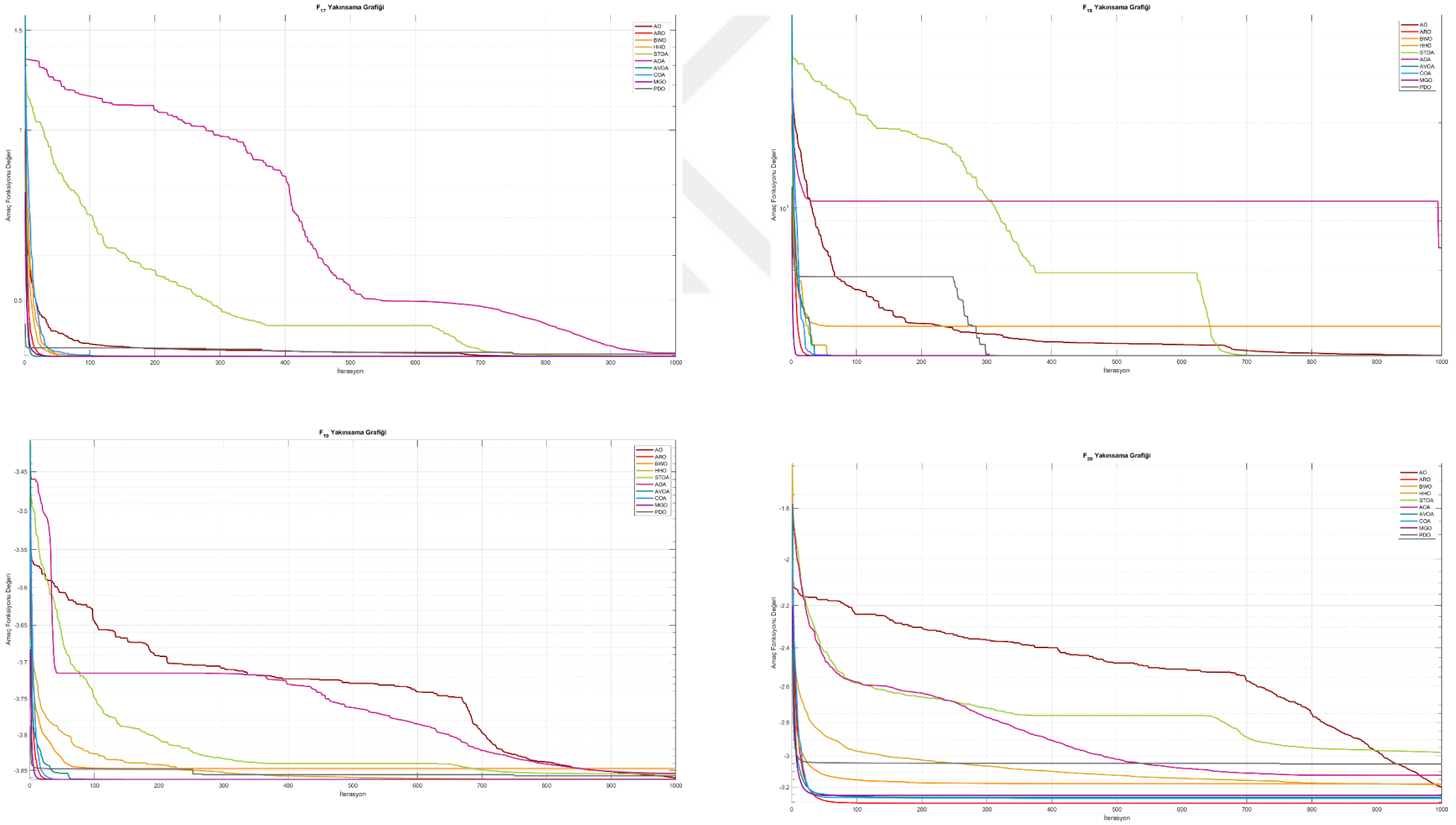

**Şekil 3.3.2** Fonksiyonların yakınsama grafikleri ($F_5 - F_8$)



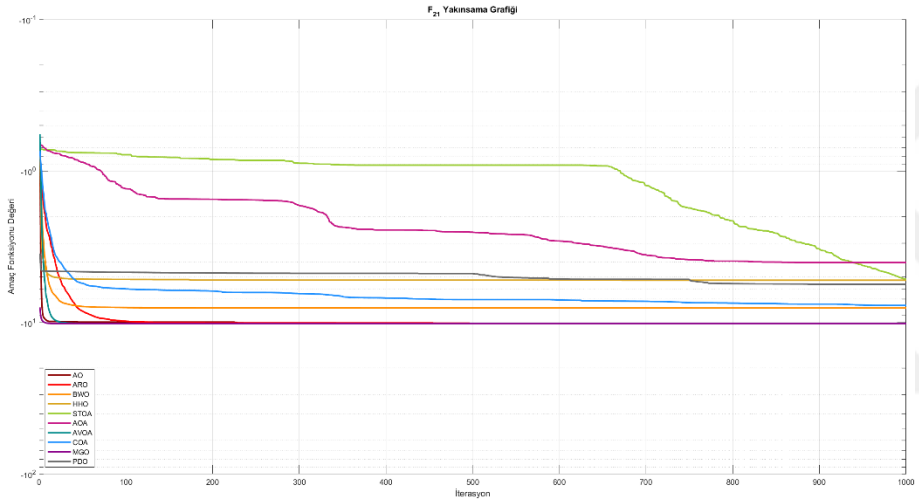

**Şekil 3.3.3** Fonksiyonların yakınsama grafikleri ($F_9 - F_{12}$)

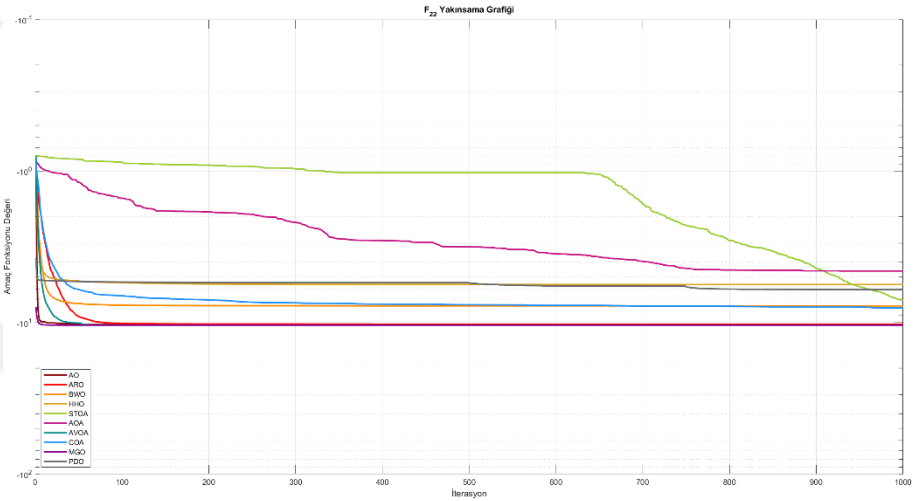


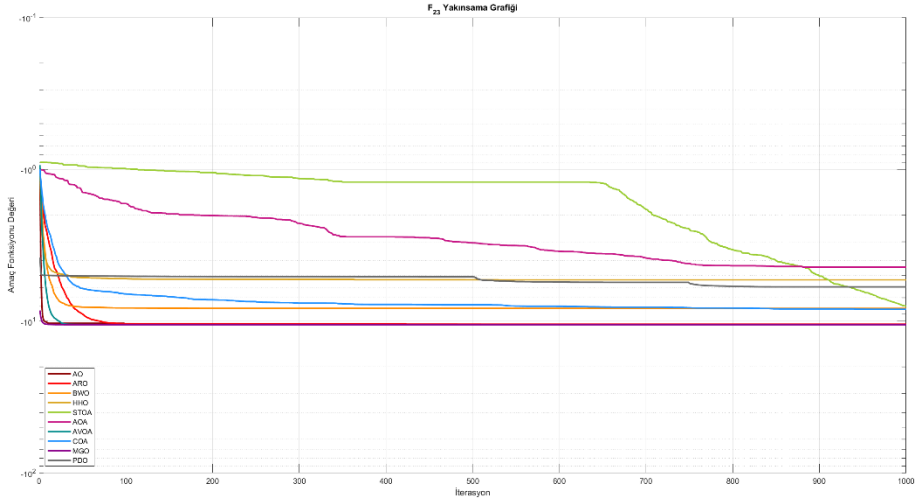

**Şekil 3.3.4** Fonksiyonların yakınsama grafikleri ($F_{13} - F_{16}$)




**Şekil 3.3.5** Fonksiyonların yakınsama grafikleri ($F_{17} - F_{20}$)




**Şekil 3.3.6** Fonksiyonların yakınsama grafikleri ($F_{21} - F_{23}$)

Yakınsama grafiklerine göre aşağıdaki yorumlar çıkarılabilir.

- $F_1$ fonksiyonu için yakınsama grafiği incelendiği en hızlı yakınsayan algoritma COA' dır. COA' yı sırasıyla BWO ve AVOA takip etmiştir. AO yaklaşık 650. iterasyondan itibaren, HHO yaklaşık 750. iterasyondan itibaren kendi optimum değerlerine ulaşarak yakınsamalarını tamamlamışlardır. PDO ve STOA ilk iterasyonlardan itibaren çok yavaş yakınsama hızlarına sahiptir. 3. en yavaş yakınsama hızına sahip olan ARO' dur.
- $F_2$ fonksiyonunda $F_1$ ile benzer olarak sırasıyla COA ve BWO en hızlı yakınsayan algoritmalardır. Bu iki algoritmayı AOA ve AVOA takip etmiştir. Diğer algoritmalar bu 4 dört algoritma kadar hızlı yakınsayamamıştır.
- $F_3$ fonksiyonunda MGO, STOA' dan sonra en yavaş yakınsayan ikinci algoritmadır.
- $F_4$ fonksiyonunda diğer üç fonksiyonda olduğu gibi PDO yakınsayamamıştır.
- $F_5$ fonksiyonunda rakiplerine göre çok hızlı yakınsayan MGO' dur. Diğer algoritmalar çok daha yavaş yakınsama hızına ve daha yüksek amaç fonksiyonu değerine sahiptir.
- $F_6$ fonksiyonunda MGO ve ARO en hızlı yakınsayan algoritmalardır. Bu iki algoritmayı AVOA takip eder.
- $F_7$ fonksiyonunda STOA ve PDO haricindeki tüm algoritmalar ilk 300 iterasyonda çok hızlı yakınsamışlardır. Ancak daha sonrasında daha yavaş yakınsama hızlarına sahiplerdir. Burada en hızlı yakınsayanlar COA ve AOA' dır.
- $F_8$ fonksiyonunda MGO ilk iterasyonlarda optimum değere ulaşmıştır. Bu nedenle bir yakınsama hızına sahip değildir. İlk 100 iterasyonda HHO ve AVOA çok yüksek yakınsama hızına sahiptir.
- $F_9$ fonksiyonunda STOA, PDO, AOA ve HHO hariç tüm algoritmalar yaklaşık ilk 200 iterasyonda çok hızlı yakınsama göstermişlerdir.
- $F_{10}$ fonksiyonunda en hızlı yakınsayan ilk üç algoritma sırasıyla COA, BWO ve AVOA' dır.
- $F_{11}$ fonksiyonunda $F_{10}$ ile benzer şekilde algoritmaların neredeyse tamamı çok hızlı yakınsamıştır. En yavaş yakınsayan iki algoritma sırasıyla STOA ve PDO' dur.

- $F_{12}$ ve $F_{13}$ fonksiyonunda MGO rakiplerine nazaran oldukça yüksek bir yakınsama hızı ile birinci olmuştur.
- $F_{14}$ fonksiyonunda MGO ve ARO optimum değere ilk 100 iterasyonda ulaşmışlardır. Bu iki algoritmayı sırasıyla AVOA ve HHO takip eder.
- Farklı boyutlu çok modlu fonksiyon sonuçları tablosu $F_{15}$ fonksiyonu yakınsama grafiğini doğrulamaktadır. Buna göre HHO, ARO, MGO, AVOA algoritmalarının yakınsama hızları benzer ve ulaştıkları optimum değerler birbirlerine oldukça yakındır.
- $F_{16}$ fonksiyonunda tüm algoritmaların bütün koşmaları optimum değere benzer hızlarla ulaştığından hız eğrileri kesişmiştir.
- $F_{17}$ fonksiyonunda tüm algoritmalar optimum değere ulaşsa da AOA ve STOA en yavaş yakınsayan algoritmalardır.
- $F_{18}$ fonksiyonunda AOA' nın iterasyonlar boyunca yakınsama hızı sahiptir. Burada en hızlı yakınsayan algoritma MGO' dur.
- $F_{19}$' da algoritmaların çoğu çok hızlı yakınsama hızları ile birbirlerine yakın amaç fonksiyonu değerlerine ulaşmışlardır. Burada en yavaş yakınsayanlar AO ve AOA' dır.
- $F_{20}$' de ARO, MGO ve COA en hızlı yakınsayan algoritmalardır. Burada en yavaş yakınsayan AO olmuştur.
- $F_{21}$ ve $F_{22}$ ve $F_{23}$ fonksiyonunda ilk 100 iterasyonda çok hızlı yakınsayarak amaç fonksiyonu değerine ulaşan dört algoritma MGO, AO, AVOA ve ARO' dur.

## 3.4 İstatistiksel Test Sonuçları

T testi, iki bağımsız örneklem arasındaki farkın istatistiksel olarak belirgin olup olmadığını test etmek için kullanılan parametrik bir tekniktir [48]. Bu çalışmada algoritmaların koşmalarda elde ettiği değerler kullanılarak t test yapılmıştır. Her bir algoritma 100 kez bağımsız koşturulduğundan burada örneklem büyüklüğü 100 olarak belirlenmiştir. Bu çalışmada tek kuyruklu ve anlamlılık düzeyi %5 olarak belirlenmiş t test kullanılmıştır. Sonuçlarda h değerinin 1' e eşit olması ve p değerinin 0,05'ten küçük olması karşılaştırılan iki örneklem arasında anlamlı bir fark olduğunu gösterir [49]. Ancak Çizelge 3.3.1- Çizelge 3.3.5'deki farklı fonksiyonlarda anlamlı fark olduğu düşünülen algoritma çiftlerinin istatistiksel test sonuçları Çizelge 3.4.1' de verilmiştir.

**Çizelge 3.4.1** Fonksiyon t test sonuçları

| Algoritmalar | STOA-AVOA | | AO-AVOA | | PDO-AVOA | |
|---|---|---|---|---|---|---|
| **Fonksiyonlar** | **p** | **h** | **p** | **h** | **p** | **h** |
| $F_1$ | 3,16E-04 | 1 | 0,00E+00 | 1 | | |
| $F_2$ | 1,27E-10 | 1 | 1,60E-01 | 0 | 1,00E+00 | 0 |
| $F_3$ | 1,33E-06 | 1 | 0,00E+00 | 1 | | |
| $F_4$ | 4,80E-08 | 1 | 1,59E-01 | 0 | | |
| $F_5$ | 1,52E-164 | 1 | 3,57E-08 | 1 | 2,09E-11 | 1 |
| $F_6$ | 5,46E-66 | 1 | | | | |
| $F_7$ | 1,23E-21 | 1 | 9,95E-01 | 0 | 1,00E+00 | 0 |
| $F_8$ | 2,71E-104 | 1 | 1,09E-12 | 1 | 2,07E-139 | 1 |
| $F_9$ | 5,15E-04 | 1 | | | | |
| $F_{10}$ | 0,00E+00 | 1 | | | | |
| $F_{11}$ | 2,98E-07 | 1 | | | | |
| $F_{12}$ | 8,95E-35 | 1 | 2,09E-07 | 1 | 3,62E-19 | 1 |
| $F_{13}$ | 7,57E-86 | 1 | 6,94E-10 | 1 | 2,26E-111 | 1 |
| $F_{14}$ | 6,21E-03 | 1 | 2,02E-03 | 1 | 1,44E-10 | 1 |
| $F_{15}$ | 2,11E-59 | 1 | 2,75E-07 | 1 | 1,45E-25 | 1 |
| $F_{16}$ | 1,12E-16 | 1 | 1,84E-13 | 1 | 1,48E-01 | 0 |
| $F_{17}$ | 1,33E-13 | 1 | 1,42E-05 | 1 | 2,34E-02 | 1 |
| $F_{18}$ | 1,51E-07 | 1 | 2,48E-17 | 1 | 1,00E+00 | 0 |
| $F_{19}$ | 1,00E-93 | 1 | 1,42E-16 | 1 | 5,12E-26 | 1 |
| $F_{20}$ | 2,08E-16 | 1 | 1,95E-11 | 1 | 2,26E-13 | 1 |
| $F_{21}$ | 1,73E-20 | 1 | 4,51E-10 | 1 | 8,76E-37 | 1 |
| $F_{22}$ | 1,84E-12 | 1 | 1,00E-05 | 1 | 2,19E-27 | 1 |
| $F_{23}$ | 1,54E-10 | 1 | 6,28E-11 | 1 | 8,97E-29 | 1 |

Çizelgedeki t test sonuçlarına göre tüm fonksiyonlar için AVOA, STOA' dan daha düşük amaç fonksiyonu değerleri ile daha iyi sonuçlar ürettiği kanıtlanmıştır. Bununla birlikte AVOA 16 fonksiyonla AO' dan, 12 fonksiyonla PDO' dan daha düşük değerler sahiptir.

# 4. MÜHENDİSLİK PROBLEMLERİNİN METASEZGİSELLER İLE ÇÖZÜMÜ

## 4.1 Giriş

Bu bölümde metasezgiseller ile 6 farklı mühendislik problemi çözülmüştür. İlerleyen alt bölümlerde çözülen mühendislik tasarım problemleri detaylı olarak anlatılmış; simülasyon sonuçları, yakınsama grafikleri ve algoritmaların elde ettiği parametreler sunulmuştur.

## 4.2 Mühendislik Problemleri

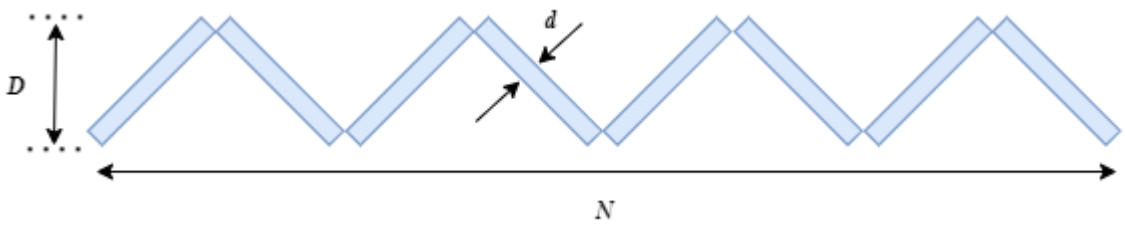
Gerçek dünya mühendislik tasarım problemleri çok sayıda kısıtı olan oldukça karmaşık amaç fonksiyonlarına sahip, çözümü zor olan problemlerdir. Yaygın olarak kullanılan bu problemler özellikle endüstriyel alanlar ve disiplinler arası çalışmalarda oldukça kullanılmaktadır [50]. Burada en çok kullanılan 6 gerçek dünya mühendislik tasarım problemi ele alınarak bunlar hakkında detaylı bilgi verilmiştir.

### 4.2.1 Germe/Sıkıştırma Yayı Tasarımı

Germe/sıkıştırma yay tasarım problemi, minimum ağırlığa sahip bir yay tasarımı oluşturmayı amaçlayan bir problemdir [51]. Germe/sıkıştırma yayı probleminin üç adet karar değişkeni bulunmaktadır [52].

- $d\ x_1$ : Tel çapı
- $D\ x_2$ : Ortalama bobin çapı
- N $x_3$ : Aktif bobin sayısı

Germe/sıkıştırma yayı tasarımı probleminin şematik yapısı Şekil 4.2.1'de aşağıdaki gibi gösterilmiştir.




**Şekil 4.2.1** Germe/Sıkıştırma Yayı Tasarımı

Burada dört adet kısıtlama yer alır. Bunlar: Minimum sapma ($g_1$), kayma gerilimi ($g_2$), darbe frekansı ($g_3$) ve dış çap sınırı ($g_4$). Problemin matematiksel tanımı aşağıdaki gibidir [52].

Amaç fonksiyonu(minimize):

$$f(x) = (x_3 + 2)x_2x_1^2 \quad [4.1]$$

Kısıtlayıcılar:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0 \quad [4.2]$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad [4.3]$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad [4.4]$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad [4.5]$$

Değişken aralıkları:

$$0{,}05 \leq x_1 \leq 2 \quad [4.6]$$

$$0{,}25 \leq x_2 \leq 1{,}3 \quad [4.7]$$

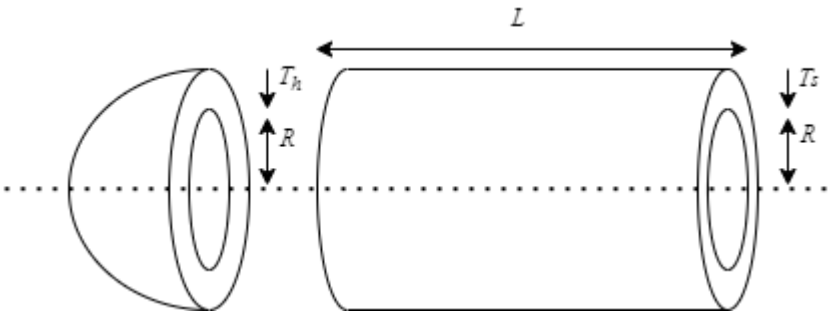
$$2 \leq x_3 \leq 15 \quad [4.8]$$

### 4.2.2 Basınçlı Kap Tasarımı

Basınçlı kap tasarım probleminde amaç silindirik basınçlı kapların toplam maliyetini en aza indirmektir [17]. Bu tasarım probleminin dört adet karar değişkeni bulunmaktadır [52].

- $T_s$ ($X_1$): Kabuğun kalınlığı
- $T_h$($X_2$): Kafanın kalınlığı
- $R$ ($X_3$): İç yarıçap
- $L$ ($X_4$): Kafa göz alınmadan silindirik bölümün uzunluğu

Basınçlı kap tasarım probleminin şematik yapısı Şekil 4.2.2' de aşağıdaki gibi gösterilmiştir.




**Şekil 4.2.2** Basınçlı Kap Tasarımı

Problemin matematiksel modeli aşağıdaki gibidir [17]. Basınçlı kap tasarım probleminin karar değişkenleri ve kısıtlayıcıları ile birlikte modeli Denklem 4.9 ile aşağıdaki gibi gösterilmiştir.

$$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [T_s \quad T_h \quad R \quad L] \tag{4.9}$$

Amaç fonksiyonu:

$$f(\vec{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{4.10}$$

Kısıtlar:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \tag{4.11}$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0 \tag{4.12}$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \tag{4.13}$$

$$g_4(x) = x_4 - 240 \leq 0 \tag{4.14}$$

Değişken aralıkları:

$$0 \leq x_1 \leq 99{,}0 \leq x_2 \leq 99{,}10 \leq x_3 \leq 200{,}10 \leq x_4 \leq 200 \tag{4.15}$$

### 4.2.3 Kaynaklı Kiriş Tasarım Problemi

Kaynaklı kiriş tasarım probleminin temel amacı, belirli kısıtlar altında minimum maliyete bir kiriş üretimi sağlamaktır [53]. Problem, dört tasarım değişkeninden ve beş kısıtlamadan oluşmaktadır. Tasarım değişkenleri şunlardır [50]:

- $h$ $(x_1)$: Kaynak kalınlığı
- $l$ $(x_2)$: Kaynak bağlantı uzunluğu
- $t$ $(x_3)$: Kirişin genişliği
- $b$ $(x_4)$: Kiriş kalınlığı

Problemin matematiksel modeli aşağıdaki gibidir [50]. Kaynaklı kiriş tasarım probleminin karar değişkenleri ve kısıtlayıcıları ile birlikte modeli Denklem 4.16 ile aşağıdaki gibi gösterilmiştir.

$$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [h \quad l \quad t \quad b] \tag{4.16}$$

Amaç fonksiyonu:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{4.17}$$

Kısıtlar:

$$g_1(x) = \tau(\mathrm{x}) - \tau_{max} \leq 0 \tag{4.18}$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \tag{4.19}$$

$$g_3(x) = x_1 - x_4 \leq 0 \tag{4.20}$$

$$g_4(x) = \&1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \tag{4.21}$$

$$g_5(x) = 0.125 - x_1 \leq 0 \tag{4.22}$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0 \tag{4.23}$$

$$g_7(x) = P - P_c(x) \leq 0 \tag{4.24}$$

Burada;

$$\tau(x) = \sqrt{(\tau')^2 + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2} \tag{4.25}$$

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2} \quad [4.26]$$

$$\tau'' = \frac{MR}{J} \quad [4.27]$$

$$M = 6000\left(14.0 + \frac{x_2}{2}\right) \quad [4.28]$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad [4.29]$$

$$J = \left\{x_1x_2\sqrt{2}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \quad [4.30]$$

$$\sigma(x) = \frac{504000}{x_4x_3^2} \quad [4.31]$$

$$\delta(x) = \frac{2.1952}{x_4x_3^3} \quad [4.32]$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{196}\left(1 - \frac{x_3\sqrt{\frac{E}{4G}}}{28}\right) \quad [4.33]$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{\sqrt{2}x_4^6}{36}}}{196}\left(1 - \frac{x_3\sqrt{\frac{E}{4G}}}{28}\right) \quad [4.34]$$

$$\begin{aligned} &\tau_{max} = 13600psi \\ &\sigma_{max} = 30000psi \\ &\delta_{max} = 0.25in \\ &E = 30 \times 10^6 psi \\ &G = 12 \times 10^6 psi \end{aligned} \quad [4.35]$$

Değişken aralıkları:

$$0.125 \leq x_1 \leq 5 \text{ ve } 0.1 \leq x_2, x_3, x_4 \leq 10 \quad [4.36]$$

### 4.2.4 Hız Düşürücü Tasarımı

Hız düşürücü tasarım problemi, hava taşıtı motorunun en verimli hızda dönmesini sağlayan basit bir vites kutusu problemidir. Bu tasarım problemi, hız düşürücünün minimum maliyet ağırlığını bulmayı amaçlamaktadır [54]. Tasarım değişkenleri şunlardır [50]:

- $b\ x_1$: Yüz genişliği
- $m\ x_2$ : Diş modülü
- $z\ x_3$ : Pinyondaki diş sayıs
- $l_1\ x_4$ : Yataklar arasındaki ilk milin uzunluğu
- $l_2\ x_5$: Yataklar arasındaki ikinci milin uzunluğu
- $d_1\ x_6$ : Birinci milin çap 1
- $d_2\ x_7$ : İkinci milin çap 1

Problemin matematiksel modeli aşağıdaki gibidir [50]. Hız düşürücü tasarım probleminin karar değişkenleri ve kısıtlayıcıları ile birlikte modeli Denklem 4.37 ile aşağıdaki gibi gösterilmiştir.

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_l, d_2] \quad [4.37]$$

Amaç fonksiyonu:

$$f(x) = 0.785x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 42.0934) \\ -1.508x_1(x_6^2 + x_7^2) + 7.4777x_1(x_6^3 + x_7^3) + 1.508x_1(x_4x_6^2 + x_5x_7^2) \quad [4.38]$$

Kısıtlar:

$$g_1(\vec{x}) = \frac{27}{x_1 \times x_2^2 \times x_3} - 1 \leq 0 \quad [4.39]$$

$$g_2(\vec{x}) = \frac{397.5}{x_1 \times x_2^2 \times x_3^2} - 1 \leq 0 \quad [4.40]$$

$$g_3(\vec{x}) = \frac{1.93 \times x_4^3}{x_2 \times x_3 \times x_6^4} - 1 \leq 0 \quad [4.41]$$

$$g_4(\vec{x}) = \frac{1.93 \times x_5^3}{x_2 \times x_3 \times x_7^4} - 1 \leq 0 \quad [4.42]$$

$$g_5(\vec{x}) = \frac{1}{110 \times x_6^3} \times \sqrt{\left(\frac{745 \times x_4}{x_2 \times x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \quad [4.43]$$

$$g_6(\vec{x}) = \frac{1}{85 \times x_7^3} \times \sqrt{\left(\frac{745 \times x_5}{x_2 \times x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \quad [4.44]$$

$$g_7(\vec{x}) = \frac{x_2 \times x_3}{40} - 1 \leq 0 \quad [4.45]$$

$$g_8(\vec{x}) = \frac{5 \times x_2}{x_1} - 1 \leq 0 \quad [4.46]$$

$$g_9(\vec{x}) = \frac{x_1}{12 \times x_2} - 1 \leq 0 \quad [3.47]$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \quad [4.48]$$

$$g_{11}(\vec{x}) = \frac{1.1 \times x_7 + 1.9}{x_5} - 1 \leq 0 \quad [4.49]$$

Değişken aralıkları:

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, \\ 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5. \quad [4.50]$$

### 4.2.5 Dişli Takımı Tasarım Problemi

Dişli takımı tasarım problemi, kısıtsız ayrık bir tasarım optimizasyon problemidir. Bu problemde amaç dişli takımının dişli oran maliyetini en aza indirmektir. Bu problemdeki tasarım değişkenleri $T_a, T_b, T_c, T_d$ sırasıyla olmak üzere dişli çarkların diş sayılarıdır. Problemin matematiksel modeli aşağıdaki gibidir [16].

Dişli takımı tasarım probleminin tasarım değişkenleri yapısının modeli Denklem 4.51 ile aşağıdaki gibi gösterilmiştir.

$$\vec{x} = [x_1, x_2, x_3, x_4] = [T_a, T_b, T_d, T_f] \quad [4.51]$$

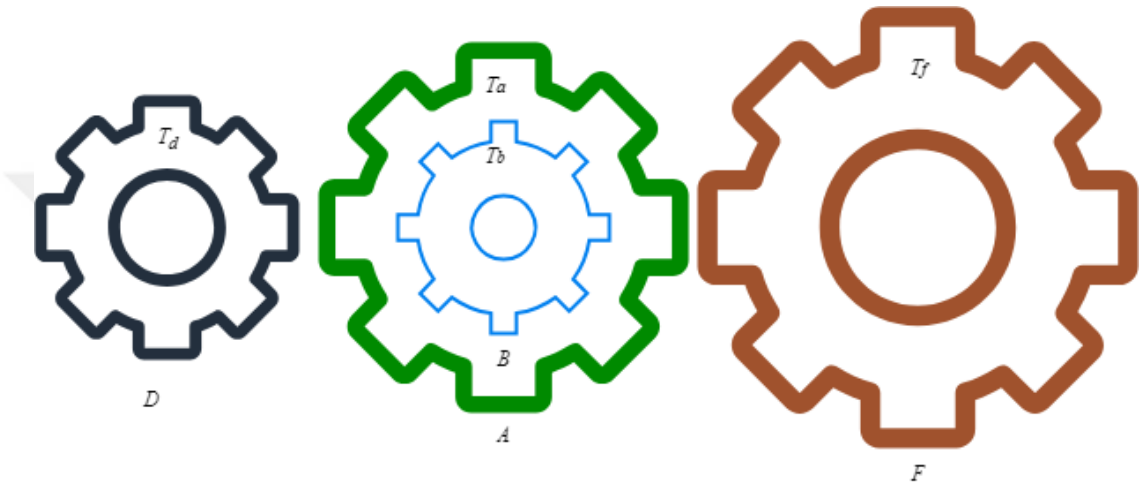
Amaç fonksiyonu:

$$\text{Minimize } f_5(\vec{x}) = \left(\frac{1}{6.931} - \frac{T_b \cdot T_d}{T_a \cdot T_f}\right)^2 \quad [4.52]$$

Değişken aralıkları:

$$x_1, x_2, x_3, x_4 \in \{12,13,14, \ldots ,60\} \quad [4.53]$$

Dişli takımı tasarım probleminin şematik yapısı Şekil 4.2.3' te aşağıdaki gibi gösterilmiştir [14].




**Şekil 4.2.3** Dişli Takımı Tasarımı

### 4.2.6 Üç Çubuklu Kafes Tasarım Problemi

Üç çubuklu kafes tasarım problemi, statik olarak yüklenen üç çubuklu bir kafes kirişin ağırlığını en aza indirmektir [55]. Bu problemdeki tasarım değişkenleri $x_1, x_2$ olmak üzere çapraz kesit alanlarıdır [52]. Problemin matematiksel modeli aşağıdaki gibidir [12].

Üç çubuklu kafes tasarım probleminin tasarım değişkenleri yapısının modeli Denklem 4.54 ile aşağıdaki gibi gösterilmiştir.

$$\vec{X} = [x_1 x_2] = [A_1 A_2] \quad [4.54]$$

Amaç fonksiyonu:

$$f(\vec{X}) = \left(2\sqrt{2}X_1 + X_2\right) \times L \quad [4.55]$$

Kısıtlar:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \quad [4.56]$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \quad [4.57]$$

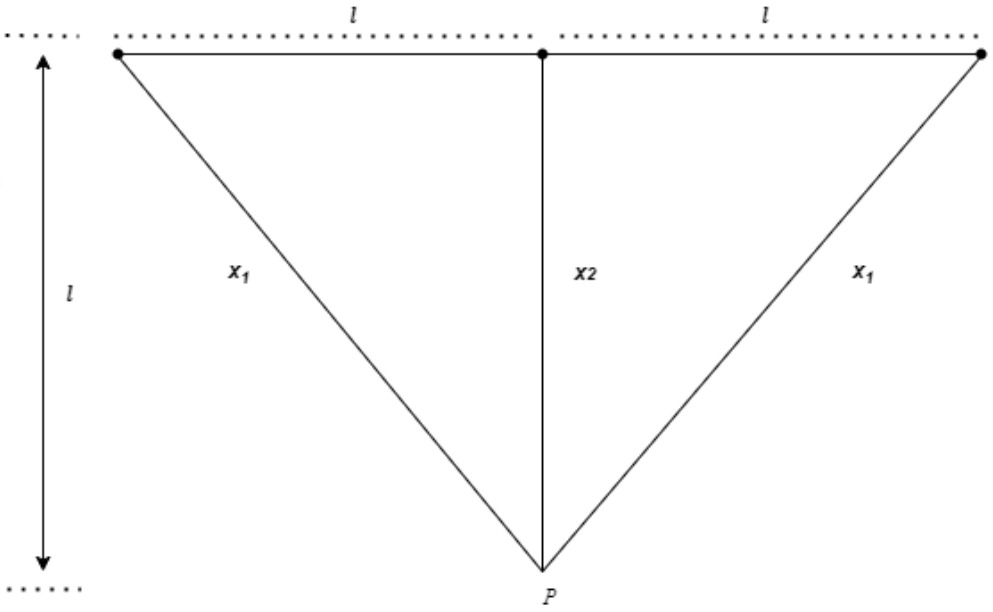
$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0 \quad [4.58]$$

Değişken aralıkları:

$$0 \leq x_i \leq 1, i = 1,2 \quad [4.59]$$

Burada; $l = 100\text{cm}, P = 2\text{kN/cm}^2, \sigma = \text{kN/cm}^2$, dir.

Üç çubuklu kafes tasarım probleminin şematik yapısı Şekil 4.2.4' te aşağıdaki gibi gösterilmiştir [55].




**Şekil 4.2.4** Üç Çubuklu Kafes Tasarımı

## 4.3 Sonuçlar

Çalışmada tüm algoritmalar 100 kez bağımsız koşturulmuştur. Algoritmalar MATLAB 2024a platformu kullanılarak yazılmıştır. Kullanılan bilgisayar, 13th Gen Intel(R) Core (TM) i9-13900H 2.60 GHz hıza ve 16 GB RAM özelliklerine sahiptir. Tezin 3.bölümünde kullanılan parametreler ve değerleri Çizelge 3.4' te yer aldığı şekliyle bu çalışma içinde aynı kullanılmıştır.

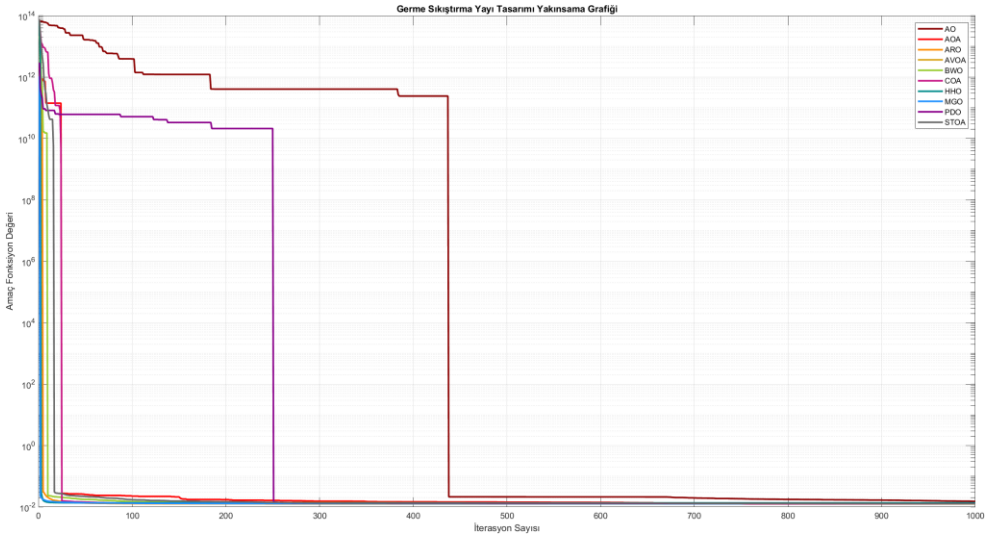
### 4.3.1 Simülasyon sonuçları

Problemlere göre algoritmaların ortalama, standart sapma, en iyi, en kötü ve sıralama değerleri Çizelge 4.3.1-4.3.6' da sunulmuştur. Tabloda ortalama ve en iyi değerler renklendirilerek sunulmuştur. Çizelge 4.3.1-4.3.6' ya göre aşağıdaki çıkarımlara ulaşılır:

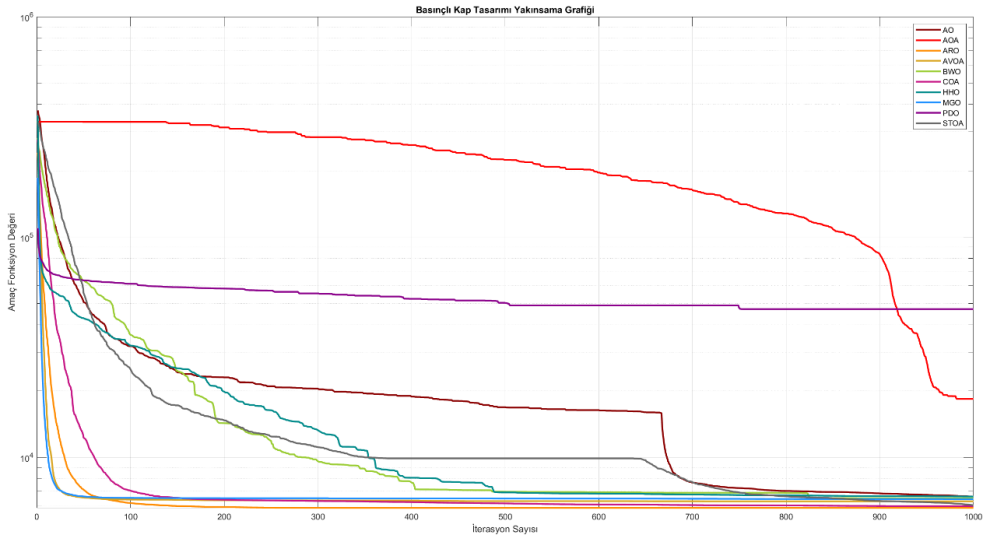
- Tüm problemler için ARO oldukça düşük amaç fonksiyonları değerlerine sahiptir. Ayrıca toplamda 9 sıralama puanına sahiptir. Bu da tüm algoritmalar içerisinde mühendislik problemlerinin en iyi çözen algoritma olduğunun göstergesidir. ARO' dan sonra en başarılı 3 algoritma BWO, MGO ve AVOA dır.
- Dişli takımı tasarımı problemini tüm algoritmalar başarı ile çözmüştür.
- Test fonksiyonlarının aksine STOA, germe sıkıştırma yayı tasarım problemi için oldukça düşük bir amaç fonksiyonu değeri ile 6. sırada yer almaktadır.
- Algoritmalar içerisinde en yüksek sıralama puanına (55) sahibi olan AOA mühendislik problemleri çözümünde sonuncudur.

### 4.3.2 Yakınsama Grafikleri Analizi

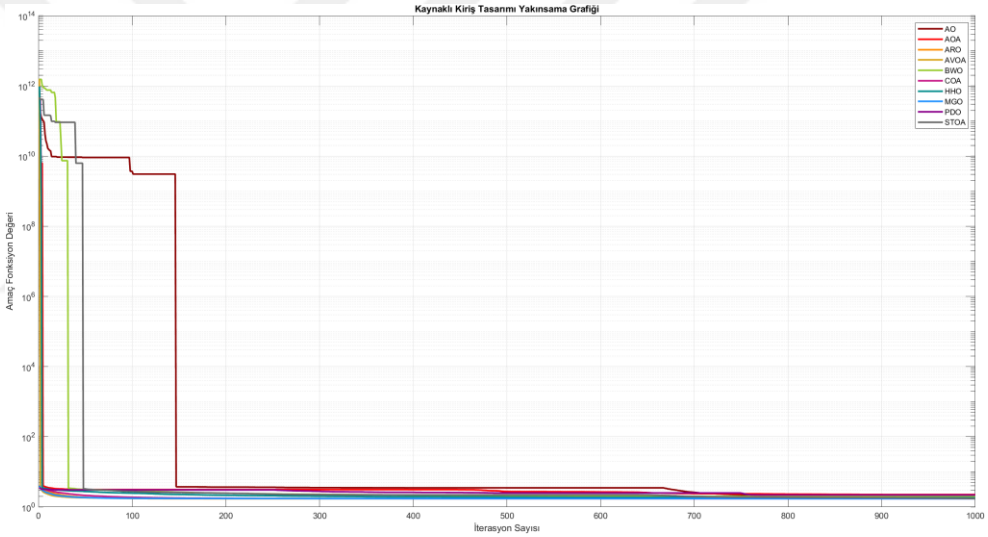
Mühendislik problemleri için çıkarılan yakınsama grafikleri Şekil 4.3.1-4.3.6' da sunulmuştur.



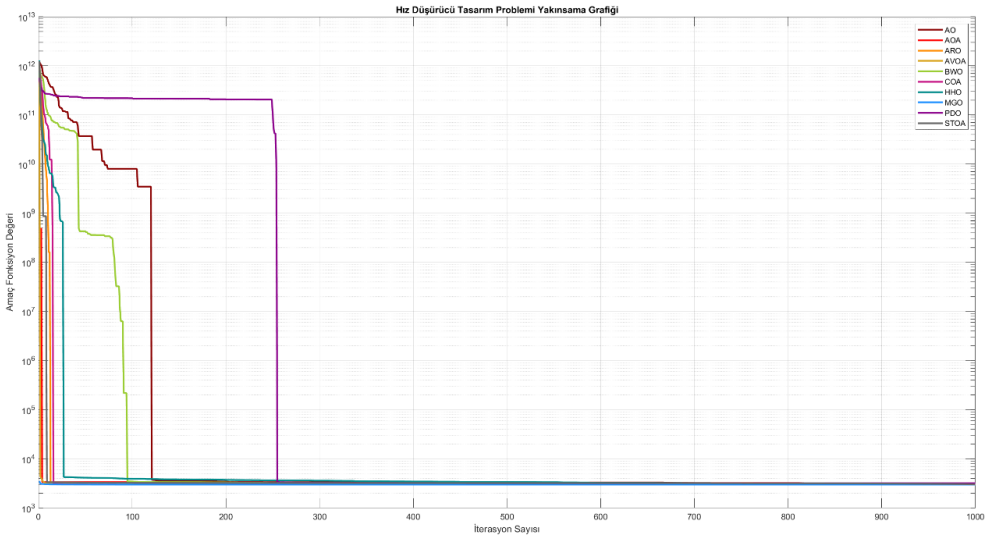

**Şekil 4.3.1** Germe Sıkıştırma yayı tasarımı yakınsama grafiği



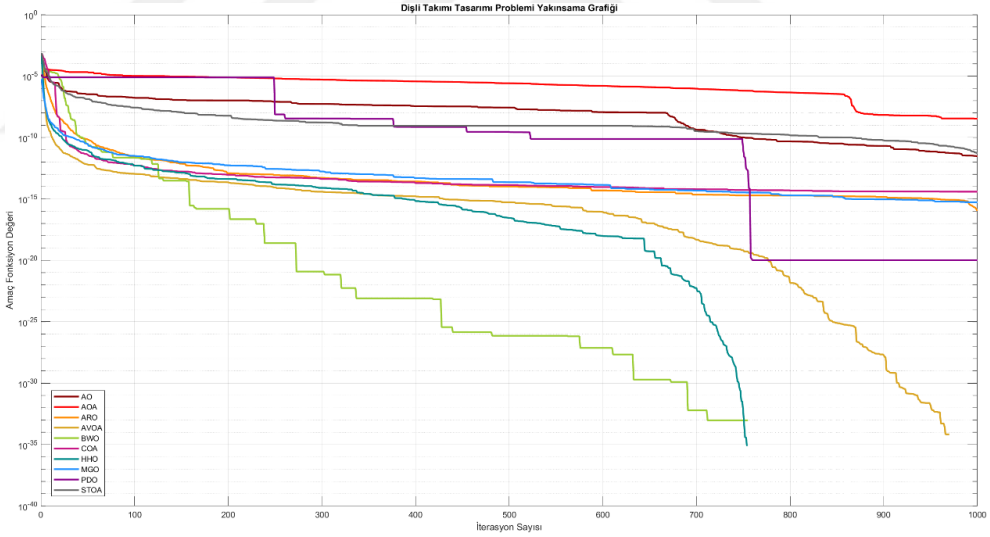

**Şekil 4.3.2** Basınçlı kap tasarımı yakınsama grafiği



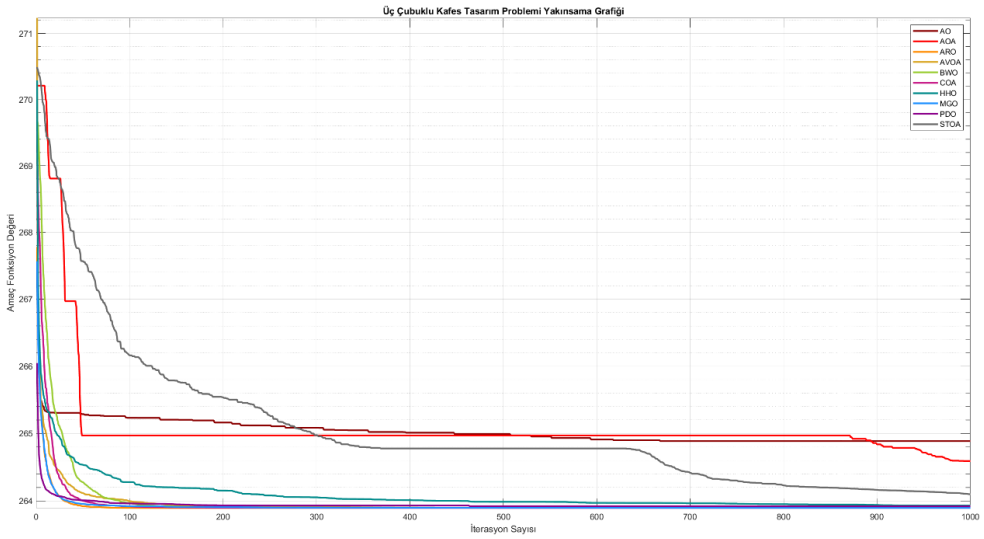

**Şekil 4.3.3** Kaynaklı kiriş tasarımı yakınsama grafiği




**Şekil 4.3.4** Hız düşürücü tasarım problemi yakınsama grafiği




**Şekil 4.3.5** Dişli takımı tasarımı problemi yakınsama grafiği




**Şekil 4.3.6** Üç çubuklu kafes tasarım problemi yakınsama grafiği

Yakınsama grafiklerine göre aşağıdaki yorumlar yapılabilir:

- Germe sıkıştırma yayı tasarım problemi için PDO ve AO hariç diğer tüm algoritmalar minimum amaç fonksiyonu değerlerine yaklaşık 30. iterasyonda ulaşmışlardır.
- Basınçlı kap tasarımı problemi için en hızlı yakınsayan iki algoritma MGO ve AVOA olmuştur; ancak ARO bu algoritmalardan daha yavaş yakınsamasına rağmen daha düşük amaç fonksiyonu değerine ulaşmıştır.
- Kaynaklı kiriş tasarımı problemi için BWO, STOA ve AO hariç diğer algoritmalar çok hızlı yakınsayarak amaç fonksiyon değerini elde etmiştir. Bu algoritmalarda yavaş yakınsamalarına rağmen daha sonra benzer amaç fonksiyon değerleri elde etmişlerdir.
- Hız düşürücü tasarım problemi için en yavaş yakınsayan algoritma sırasıyla PDO, AO ve BWO' dur.
- Dişli takımı tasarımı probleminde diğer problemlere nazaran algoritmalar farklı yakınsama hızları göstermişlerdir. BWO' dan daha yavaş yakınsayan HHO ve AVOA daha düşük amaç fonksiyonu değerine ulaşabilmişlerdir.
- Üç çubuklu kafes tasarım problemi için STOA en yavaş yakınsayan algoritma olmasına rağmen AO ve AOA daha düşük amaç fonksiyon değeri elde edebilmiştir.
- Tüm problemler birlikte değerlendirildiğinde AO'nun en yavaş yakınsama hızına sahip olduğu görülebilir.

### 4.3.3 En İyi Modellerin Analizi

Mühendislik tasarım problemleri için algoritmaların 100 koşma sonucunda çıkarılan en iyi modellerinin optimize ettiği parametre değerleri ve amaç fonksiyonu değerleri Çizelge 4.3.1- Çizelge 4.3.7 da sunulmuştur.

**Çizelge 4.3.1** Germe sıkıştırma yayı tasarım problemi sonuçları

| Algoritmalar | x1 | x2 | x3 | Amaç Fonksiyonu Değerleri | Rank |
|---|---|---|---|---|---|
| **AO** | 0,052592491 | 0,37470729 | 10,48502 | 0,012939839 | 9 |
| **AOA** | 0,05 | 0,3168629 | 14,649187 | 0,013188774 | 10 |
| **ARO** | 0,051680393 | 0,35650922 | 11,301204 | 0,012665238 | 2 |
| **AVOA** | 0,051834953 | 0,36023768 | 11,085547 | 0,01266562 | 4 |
| **BWO** | 0,051692255 | 0,35679458 | 11,284462 | 0,012665233 | 1 |
| **COA** | 0,051509285 | 0,35240065 | 11,549537 | 0,012668704 | 5 |
| **HHO** | 0,052900223 | 0,3865607 | 9,7321911 | 0,012691467 | 7 |
| **MGO** | 0,05160305 | 0,35465206 | 11,411106 | 0,012665368 | 3 |
| **PDO** | 0,050777043 | 0,33517065 | 12,695232 | 0,012699225 | 8 |
| **STOA** | 0,051987112 | 0,36377502 | 10,893929 | 0,012676797 | 6 |

**Çizelge 4.3.2** Basınçlı kap tasarımı problemi sonuçları

| Algoritmalar | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Amaç Fonksiyonu Değerleri | Sıralama |
|---|---|---|---|---|---|---|
| **ARO** | 0,778168652 | 0,38464917 | 40,319619 | 200 | 5885,332823 | 1 |
| **BWO** | 0,778168753 | 0,38464922 | 40,319622 | 200 | 5885,333375 | 2 |
| **MGO** | 0,778316101 | 0,38472205 | 40,327259 | 199,894 | 5885,584879 | 3 |
| **AVOA** | 0,77851532 | 0,38482053 | 40,337581 | 199,75 | 5885,925625 | 4 |
| **COA** | 0,77962882 | 0,38542321 | 40,38514 | 199,091 | 5889,372289 | 5 |
| **STOA** | 0,780186935 | 0,38738763 | 40,416082 | 198,677 | 5895,266668 | 6 |
| **AO** | 0,787601938 | 0,40980793 | 40,667112 | 196,283 | 6004,018483 | 7 |
| **HHO** | 0,806719602 | 0,43650485 | 41,780412 | 180,618 | 6055,476821 | 8 |
| **PDO** | 0,880778106 | 0,4352173 | 45,620253 | 137,421 | 6086,984962 | 9 |
| **AOA** | 0,791275935 | 0,44601592 | 40,554178 | 200 | 6199,048644 | 10 |

**Çizelge 4.3.3** Kaynaklı kiriş tasarımı problemi sonuçları

| Algoritmalar | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Amaç Fonksiyonu Değerleri | Sıralama |
|---|---|---|---|---|---|---|
| **ARO** | 0,2057296 | 3,4704887 | 9,0366239 | 0,20572964 | 1,72485230860 | 1 |
| **MGO** | 0,2057296 | 3,4704887 | 9,0366239 | 0,20572964 | 1,72485230861 | 2 |
| **BWO** | 0,2057296 | 3,4704888 | 9,0366241 | 0,205729639 | 1,72485234090 | 3 |
| **AVOA** | 0,2057288 | 3,4704997 | 9,03664 | 0,20572956 | 1,72485473370 | 4 |
| **COA** | 0,2057105 | 3,4709241 | 9,0366253 | 0,205730396 | 1,72488747435 | 5 |
| **STOA** | 0,2060635 | 3,4678117 | 9,0281662 | 0,206160973 | 1,72682471051 | 6 |
| **HHO** | 0,2040826 | 3,4891408 | 9,0808917 | 0,20571622 | 1,73235128058 | 7 |
| **PDO** | 0,1995249 | 3,6002381 | 9,058884 | 0,205618841 | 1,73555097437 | 8 |
| **AO** | 0,2009556 | 3,5984309 | 9,0545819 | 0,20772722 | 1,75300142909 | 9 |
| **AOA** | 0,202386 | 3,2197822 | 10 | 0,206040794 | 1,85262409521 | 10 |

**Çizelge 4.3.4** Hız düşürücü tasarımı problemi sonuçları

| Algoritmalar | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | Amaç Fonksiyonu Değerleri | Sıralama |
|---|---|---|---|---|---|---|---|---|---|
| **ARO** | 3,5 | 0,7 | 17 | 7,3 | 7,715 | 3,35 | 5,287 | 2994,471066 | 1 |
| **MGO** | 3,5 | 0,7 | 17 | 7,3 | 7,715 | 3,35 | 5,287 | 2994,471066 | 1 |
| **AVOA** | 3,5 | 0,7 | 17 | 7,3 | 7,715 | 3,35 | 5,287 | 2994,471068 | 2 |
| **BWO** | 3,5 | 0,7 | 17 | 7,3 | 7,715 | 3,35 | 5,287 | 2994,471073 | 3 |
| **COA** | 3,5 | 0,7 | 17 | 7,3 | 7,716 | 3,35 | 5,287 | 2994,499611 | 4 |
| **HHO** | 3,5 | 0,7 | 17 | 7,3 | 7,901 | 3,351 | 5,287 | 2998,88129 | 5 |
| **AO** | 3,507 | 0,7 | 17 | 7,397 | 7,783 | 3,351 | 5,292 | 3003,015979 | 6 |
| **STOA** | 3,502 | 0,7 | 17 | 7,372 | 7,746 | 3,363 | 5,295 | 3005,315459 | 7 |
| **PDO** | 3,522 | 0,7 | 17 | 7,3 | 8,299 | 3,454 | 5,302 | 3052,735098 | 8 |
| **AOA** | 3,6 | 0,7 | 17 | 7,602 | 8,3 | 3,354 | 5,297 | 3056,678063 | 9 |

**Çizelge 4.3.5** Dişli takımı tasarımı problemi sonuçları

| Algoritmalar | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Amaç Fonksiyonu Değerleri | Sıralama |
|---|---|---|---|---|---|---|
| **AVOA** | 38,89793379 | 16,8804144 | 12,455453 | 37,46381 | 0 | 1 |
| **BWO** | 31,87277374 | 12,7180715 | 12,718071 | 35,17374 | 0 | 1 |
| **HHO** | 40,26331843 | 16,6614264 | 20,868086 | 59,85234 | 0 | 1 |
| **PDO** | 52,2103518 | 19,52835 | 14,157349 | 36,70175 | 0 | 1 |
| **MGO** | 42,62577039 | 18,5483072 | 14,911688 | 44,9733 | 1,11101E-21 | 2 |
| **ARO** | 48,66466943 | 13,2402823 | 13,667962 | 25,77408 | 1,44834E-21 | 3 |
| **COA** | 59,51160909 | 42,6073522 | 12,091286 | 60 | 7,07504E-21 | 4 |
| **STOA** | 59,79186347 | 12 | 42,455804 | 59,0571 | 1,28743E-16 | 5 |
| **AO** | 58,39378178 | 15,9832017 | 26,984379 | 51,1924 | 1,76985E-16 | 6 |
| **AOA** | 60 | 21,1917568 | 12,510427 | 30,62561 | 1,07693E-13 | 7 |

**Çizelge 4.3.6** Üç çubuklu tasarım problemi sonuçları

| Algoritmalar | $x_1$ | $x_2$ | Amaç Fonksiyonu Değerleri | Sıralama |
|---|---|---|---|---|
| **ARO** | 0,788675137 | 0,40824828 | 263,8958434 | 1 |
| **BWO** | 0,788675134 | 0,40824829 | 263,8958434 | 1 |
| **PDO** | 0,788713316 | 0,40814031 | 263,8958444 | 2 |
| **MGO** | 0,788634467 | 0,40836333 | 263,8958446 | 3 |
| **AVOA** | 0,788600311 | 0,40845997 | 263,8958475 | 4 |
| **HHO** | 0,7885911 | 0,40848603 | 263,8958486 | 5 |
| **COA** | 0,788664618 | 0,40827811 | 263,8958505 | 6 |
| **STOA** | 0,789159675 | 0,40688784 | 263,896847 | 7 |
| **AO** | 0,790004005 | 0,40451246 | 263,8981219 | 8 |
| **AOA** | 0,794775673 | 0,39132284 | 263,9287907 | 9 |

**Çizelge 4.3.7** Mühendislik tasarım problemleri başarı sıralamaları

| | Başarı Sıralamaları | | | | | | |
|---|---|---|---|---|---|---|---|
| **Algoritmalar** | **Germe Sıkıştırma Yayı Tasarım** | **Basınçlı Kap Tasarımı** | **Kaynaklı Kiriş Tasarımı** | **Hız Düşürücü Tasarımı** | **Dişli Takımı Tasarımı Problemi** | **Üç Çubuklu Tasarım Problemi** | **Sıralama Toplam** |
| **ARO** | 2 | 1 | 1 | 1 | 3 | 1 | 9 |
| **BWO** | 1 | 2 | 3 | 3 | 1 | 1 | 11 |
| **MGO** | 3 | 3 | 2 | 1 | 2 | 3 | 14 |
| **AVOA** | 4 | 4 | 4 | 2 | 1 | 4 | 19 |
| **COA** | 5 | 5 | 5 | 4 | 4 | 6 | 29 |
| **HHO** | 7 | 8 | 7 | 5 | 1 | 5 | 33 |
| **PDO** | 8 | 9 | 8 | 8 | 1 | 2 | 36 |
| **STOA** | 6 | 6 | 6 | 7 | 5 | 7 | 37 |
| **AO** | 9 | 7 | 9 | 6 | 6 | 8 | 45 |
| **AOA** | 10 | 10 | 10 | 9 | 7 | 9 | 55 |

Tablolarda amaç fonksiyonu değeri en iyiden en kötüye doğru sıralanmıştır. Örneğin; germe sıkıştırma yayı tasarım problemi için en iyi amaç fonksiyonu değerine sahip algoritma BWO' dur. Tablolardan aşağıdaki yorumlar çıkarılabilir:

- Germe sıkıştırma yayı tasarım problemi için AOA ve AO hariç diğer algoritmalar birbirlerine oldukça yakın amaç fonksiyonu değeri bu üç parametreyi optimize etmişlerdir.
- Basınçlı kap tasarımı problemi için diğerlerine nazaran başarısız olan 4 algoritma AO, HHO, PDO ve AOA' dır.
- Kaynaklı kiriş tasarımı probleminde AOA' nın amaç fonksiyon değerleri en yakın rakibi olan AO' dan yaklaşık 0,1 yüksek amaç fonksiyon değerine sahip olarak en başarısız algoritma olmuştur.
- Hız düşürücü tasarım problemi için sırasıyla PDO ve AOA en yakın rakipleri STOA' dan oldukça yüksek bir farkla (47 sıralama değeri) sonuncu olmuşlardır.
- Dişli takımı tasarımı problemi ve üç çubuklu kafes tasarım problemlerini tüm algoritmalar başarıyla çözmüştür.

En iyi modeller amaç fonksiyon değerine göre en iyiden en kötüye sıralandığında elde edilen sıralama toplamları Çizelge 4.3.7' de sunulmuştur. Bu tabloya göre ortalamalarda olduğu gibi en iyi modellerde de ARO mühendislik problemlerini en iyi tahmin eden algoritma olmuştur. BWO ortalama değerler sıralamasında (Çizelge 4.3.7) 5. Sırada olmasına rağmen en iyi modellerde 2. sırada yer almaktadır.

## 4.4 T-Test Sonuçları

3.bölümde olduğu gibi bu bölümde t testinden yararlanılarak algoritmaların başarıları arasında istatistiksel anlamlı bir farkın olup olmadığı değerlendirilmiştir. Burada ortalamalarda fark gözüken AOA-ARO, AO-ARO ve PDO-ARO istatistiksel test sonuçları Çizelge 4.4.1' de sunulmuştur.

Tüm mühendislik problemleri için ARO, AOA ve AO' dan daha düşük amaç fonksiyon değerleri ile anlamlı olarak daha iyi sonuçlar üretmiştir. F5 için PDO-ARO çiftinin kıyaslamasında ise F5 problemi hariç diğer tüm problemlerde ARO anlamlı olarak PDO dan daha iyidir.

**Çizelge 4.4.1** t testi sonuçları

| Algoritmal | AOA-ARO | | AO-ARO | | PDO-ARO | |
|---|---|---|---|---|---|---|
| | **p** | **h** | **p** | **h** | **p** | **h** |
| **F1** | 1,18E-31 | 1 | 8,29E-144 | 1 | 1,10E-27 | 1 |
| **F2** | 3,93E-28 | 1 | 9,51E-14 | 1 | 9,24E-08 | 1 |
| **F3** | 8,20E-25 | 1 | 1,06E-41 | 1 | 2,46E-11 | 1 |
| **F4** | 1,51E-51 | 1 | 1,65E-53 | 1 | 1,60E-80 | 1 |
| **F5** | 3,79E-06 | 1 | 8,65E-10 | 1 | 1,00E+00 | 0 |
| **F6** | 1,08E-15 | 1 | 8,93E-27 | 1 | 2,28E-03 | 1 |

# 5. TARTIŞMA, SONUÇ VE ÖNERİLER

## 5.1 Tartışma ve Sonuç

Bu tez çalışmasında öncelikle metasezgisel algoritmaların öneminden ve son 5 yılda geliştirilen 10 güncel metasezgisel hakkında literatür taraması yapılmış ve bu metasezgisel algoritmalar ayrıntılı olarak verilmiştir. Tez kapsamında bu 10 algoritmanın test fonksiyonları ve mühendislik tasarım problemleri karşısındaki başarı oranları ilk kez ortaya koyulmuştur.

Tezin birinci bölümünde amaç, kapsam ve katkıları açıklanmıştır. Ek olarak tezde kullanılan metasezgisel algoritmaların çalışmaları hakkında kısa bir açıklama yapılmıştır.

İkinci bölümde, bu çalışmada yer alan metasezgiseller hakkında detaylı bir literatür çalışması yapılmıştır. Burada metasezgisel algoritmaların her birinin çalışma yapıları ve sözde kodları yer almaktadır.

Tezin üçüncü bölümünde, çalışmada kullanılan optimizasyon algoritmaların performanslarını ölçmek amacıyla 23 farklı test fonksiyonu ile kıyaslamaları yapılmıştır. Kıyaslama sonucunda her fonksiyon için koşmaların ortalama, standart sapma, en iyi ve en kötü sonuçları ortaya çıkarılmıştır. Ayıca tez çalışmasında kullanılan tüm algoritmaların her bir test fonksiyonu üzerindeki yakınsama grafikleri çizilerek şekillerle sunulmuştur. Çalışmada yer alan metasezgisellerin önerildiği makalelerdeki kalite test fonksiyonları için tavsiye edilen değerleri olan parametre değerleri almaktadır. Test fonksiyonları ile bu algoritmaların kıyaslanmasında tüm algoritmaların popülasyon büyüklüğü 50, maksimum iterasyon sayısı 1000 olarak belirlenmiştir. Çalışmada algoritmaların koşmalarda elde ettiği değerler arasındaki farkın anlamlı olup olmadığını daha net ortaya koyabilmek için t test yapılmış ve sonuçlar ortaya çıkarılmıştır. Bu çalışmada kullanılan 23 farklı test fonksiyonları (Tek modlu, çok modlu ve farklı boyutlu çok modlu fonksiyon) sıralama değerleri birlikte değerlendirildiğinde 1. AVOA olmuştur. AVOA' yı sırasıyla MGO, AO, ARO ve COA takip etmiştir. Tüm algoritmalar içerisinde toplam 125 sıralama puanı ile en başarısız olan algoritma STOA olduğu gözlemlenmiştir.

Dördüncü bölümde, gerçek dünya mühendislik tasarım problemleri ele alınmış ve ayrıntılı bir literatür taraması yapılmıştır. Çalışmada yer alan mühendislik tasarım problemleri 10 metasezgisel ile bu problemler optimize edilmiş ve problemlere göre algoritmaların ortalama, standart sapma, en iyi, en kötü ve sıralama değerlerine yer verilmiştir. Çalışma sonuçlarına göre tüm problemlerde ARO oldukça düşük amaç fonksiyonları değerlerine (9 sıralama puanına) sahiptir. Bu da tüm algoritmalar içerisinde mühendislik problemlerinin en iyi çözen algoritma olduğunun göstergesidir. ARO' dan sonra en başarılı 3 algoritmanın BWO, MGO ve AVOA olduğu ortaya çıkarılmıştır. Ayrıca mühendislik problemleri için çıkarılan yakınsama grafikleri ortaya çıkarılmıştır. 3.bölümde olduğu gibi bu bölümde de t testinden yararlanılarak algoritmaların başarıları arasında istatistiksel anlamlı bir farkın olup olmadığı değerlendirilmiştir.

Bu çalışmada son 5 yılda önerilen metasezgisellerin test fonksiyonlarını ve mühendislik problemlerini çözme kabiliyeti değerlendirilmiştir. Çalışma kapsamında yapılan tüm değerlendirmeler sonucunda sırasıyla AVOA, MGO ve ARO' nun en yüksek başarıya sahip algoritmalardır. Gelecekteki çalışmalarda bu algoritmalar farklı mühendislik problemlerine uyarlanabilir. Ayrıca algoritmaların keşif ve sömürü aşamalarında iyileştirmeler yapılarak daha iyi amaç fonksiyonu değerlerine ve daha yüksek yakınsama hızlarına ulaşılabilir. Ek olarak, algoritmalar güncel metasezgisellerin başarı kıyaslamalarında kullanılabilir.

## KAYNAKLAR


[1] **Emel, Gül Gökay, and Ç. Taşkın.** "Genetik algoritmalar ve uygulama alanları." *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* 21.1 (2002): 129-152.

[2] **Mitchell, Melanie, and StephanieForrest**. "Genetic algorithm sand artificial life." Artificial Life 1.3 (1994): 267-289. doi:10.1162/artl.1994.1.3.267

[3] **Öznur, İ. Ş. Ç. İ., &Korukoğlu, S.** (2003). Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama. *Yönetim ve Ekonomi Dergisi*, 10(2), 191-208.

[4] **Albadr, M. A., Tiun, S., Ayob, M., & Al-Dhief, F**. (2020). Genetic algorithm based on natural selection the oryfor optimization problems. *Symmetry*, *12*(11), 1758.

[5] **Karaboga, D., &Basturk, B.** (2007). Sayısal fonksiyon optimizasyonu için güçlü ve verimli bir algoritma: Yapay arı kolonisi (ABC) algoritması. *Journal Of Global Optimazition*, *39*(3).

[6] **Özdemir, R.** (2012). *Yapay arı kolonisi algoritması için yeni seçme ve arama mekanizmalarının geliştirilmesi* (Doctoraldissertation, Yüksek lisans Tezi. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı, Kayseri 69s).

[7] **Jia, H., Rao, H., Wen, C., & Mirjalili, S.** (2023). Crayfish optimization algorithm. Artificial Intelligence Review, 56(Suppl 2), 1919-1979.

[8] **Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., &Chen, H.** (2019). Harris hawksoptimization: Algorithm and applications. Futur egeneration computer systems, 97, 849-872.

[9] **Dhiman, G., &Kaur, A.** (2019). STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of ArtificialIntelligence*, *82*, 148-174.

[10] **Hayyolalam, V., & Kazem, A. A. P.** (2020). Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *87*, 103249.

[11] **Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H.** (2021). The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering, 376, 113609.

[12] **Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S.** (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering*, *158*, 107408.

[13] **Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H.** (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. Computers & Industrial Engineering, 157, 107250.

[14] **Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W.** (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *114*, 105082.

[15] **Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N., & Mirjalili, S**. (2022). Mountain gazelle optimizer: a new nature-inspired metaheuristic

algorithm for global optimization problems. *Advances in Engineering Software*, *174*, 103282.

[16] **Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandomi, A. H.** (2022). Prairie dog optimization algorithm. *Neural Computing and Applications*, *34*(22), 20017-20065.

[17] **Jia, H., Rao, H., Wen, C., & Mirjalili, S.** (2023). Crayfish optimization algorithm. *Artificial Intelligence Review*, *56*(Suppl 2), 1919-1979.

[18] **Altay, O.** (2022). Güncel Metasezgisel Yöntemlerin Standart Kalite Testi Fonksiyonlarında Karşılaştırılması. *International Journal of PureandAppliedSciences*, *8*(2), 286-301.

[19] **JC Bednarz**, Harris'in şahinlerinde kooperatif avcılığı (parabuteounicinctus), Science 239 (1988) 1525.

[20] **Kuyu, Y. Ç.** (2023). *Optimizasyon Problemleri Için Yeni Metasezgisel Yaklaşımlar* (Doctoraldissertation, Bursa UludagUniversity (Turkey)).

[21] **Singh, A., Sharma, A., Rajput, S., Mondal, A. K., Bose, A., & Ram, M.** (2022). Parameter extraction of solar module using the sooty tern optimization algorithm. *Electronics*, *11*(4), 564.

[22] **Hu, G., Du, B., Wang, X., & Wei, G.** (2022). An enhanced black widow optimization algorithm for feature selection. *Knowledge-Based Systems*, *235*, 107638.

[23] **Li, X. D., Wang, J. S., Hao, W. K., Zhang, M., & Wang, M.** (2022). Chaotic arithmetic optimization algorithm. Applied Intelligence, 52(14), 16718-16757.

[24] **Dhal, K. G., Sasmal, B., Das, A., Ray, S., & Rai, R.** (2023). A comprehensive survey on arithmetic optimization algorithm. *Archives of Computational Methods in Engineering*, *30*(5), 3379-3404.

[25] **Fan, J., Li, Y., & Wang, T.** (2021). An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism. *Plos one*, *16*(11), e0260725.

[26] **Sasmal, B., Das, A., Dhal, K. G., & Saha, R.** (2024). A Comprehensive Survey on African Vulture Optimization Algorithm. *Archives of Computational Methods in Engineering*, *31*(3), 1659-1700.

[27] **Zhang, J., Khayatnezhad, M., & Ghadimi, N.** (2022). Optimal model evaluation of the proton-exchange membrane fuel cells based on deep learning and modified African Vulture Optimization Algorithm. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, *44*(1), 287-305.

[28] **Turgut, O. E., & Turgut, M. S.** (2023). Local search enhanced Aquila optimization algorithm ameliorated with an ensemble of Wavelet mutation strategies for complex optimization problems. Mathematics and Computers in Simulation, 206, 302-374.

[29] **AKYOL, S.** (2021). Global optimizasyon için yeni bir hibrit yöntem: kaya kartalı optimizasyonu-tanjant arama algoritması. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 33(2), 721-733.

[30] **Zhang, Y. J., Yan, Y. X., Zhao, J., & Gao, Z. M.** (2022). AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer. IEEE Access, 10, 10907-10933.

[31] **AYDEMİR, S. B.** (2022). Küresel optimizasyon için gauss kaotik haritası ile kartal optimizasyonu. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 34(1), 85-104.

[32] **Sasmal, B., Hussien, A. G., Das, A., & Dhal, K. G.** (2023). A comprehensive survey on aquila optimizer. Archives of Computational Methods in Engineering, 30(7), 4449-4476.

[33] **Bakır, H.** (2024). Dynamic fitness-distance balance-based artificial rabbits optimization algorithm to solve optimal power flow problem. Expert Systems with Applications, 240, 122460.

[34] **Wang, Y., Xiao, Y., Guo, Y., & Li, J.** (2022). Dynamic chaotic opposition-based learning-driven hybrid aquila optimizer and artificial rabbits optimization algorithm: framework and applications. Processes, 10(12), 2703.

[35] **Alsaiari, A. O., Moustafa, E. B., Alhumade, H., Abulkhair, H., & Elsheikh, A.** (2023). A coupled artificial neural network with artificial rabbits optimizer for predicting water productivity of different designs of solar stills. *Advances in Engineering Software*, *175*, 103315.

[36] **Gülmez, B.** (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems with Applications, 227, 120346.

[37] **Khodadadi, N., El-Kenawy, E. S. M., De Caso, F., Alharbi, A. H., Khafaga, D. S., & Nanni, A.** (2023). The Mountain Gazelle Optimizer for truss structures optimization. *Applied Computing and Intelligence*, *3*(2), 116-144.

[38] **Abbassi, R., Saidi, S., Urooj, S., Alhasnawi, B. N., Alawad, M. A., & Premkumar, M.** (2023). An Accurate Metaheuristic Mountain Gazelle Optimizer for Parameter Estimation of Single-and Double-Diode Photovoltaic Cell Models. *Mathematics*, *11*(22), 4565.

[39] **Ekinci, S., & Izci, D.** (2023). Enhancing IIR system identification: Harnessing the synergy of gazelle optimization and simulated annealing algorithms. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *5*, 100225.

[40] **Sarangi, P., & Mohapatra, P.** (2023). Evolved opposition-based mountain gazelle optimizer to solve optimization problems. *Journal of King Saud University-Computer and Information Sciences*, *35*(10), 101812.

[41] **Alomoush, W., Houssein, E. H., Alrosan, A., Abd-Alrazaq, A., Alweshah, M., & Alshinwan, M.** (2024). Joint opposite selection enhanced Mountain Gazelle Optimizer for brain stroke classification. *Evolutionary Intelligence*, 1-19.

[42] **Sahoo, G. K., Choudhury, S., Rathore, R. S., & Bajaj, M.** (2023). A novel prairie dog-based meta-heuristic optimization algorithm for improved control, better transient response, and power quality enhancement of hybrid microgrids. Sensors, 23(13), 5973.

[43] **Abualigah, L., Diabat, A., Thanh, C. L., & Khatir, S.** (2023). Opposition-based Laplacian distribution with Prairie Dog Optimization method for industrial engineering design problems. Computer Methods in Applied Mechanics and Engineering, 414, 116097.

[44] **Izci, D., Ekinci, S., & Hussien, A. G.** (2024). Efficient parameter extraction of photovoltaic models with a novel enhanced prairie dog optimization algorithm. Scientific Reports, 14(1), 7945.

[45] **Shikoun, N. H., Al-Eraqi, A. S., & Fathi, I. S.** (2024). BinCOA: An Efficient Binary Crayfish Optimization Algorithm for Feature Selection. *IEEE Access*, *12*, 28621-28635.

[46] **Daulat, H., Varma, T., & Chauhan, K.** (2024, April). Augmenting the Crayfish Optimization with Gaussian Distribution Parameter for Improved

Optimization Efficiency. *In 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)* (pp. 462-470). IEEE.

[47] **Xiao, B., Wang, R., Deng, Y., Yang, Y., & Lu, D**. (2024, March). Simplified Crayfish Optimization Algorithm. *In 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 7, pp. 392-396). IEEE.

[48] **Kim, T. K.** (2015). T test as a parametric statistic. *Korean journal of anesthesiology*, *68*(6), 540-546.

[49] **Browne, R. H.** (2010). The t-test p value and its relationship to the effect size and P (X> Y). *The American Statistician*, *64*(1), 30-33.

[50] **Altay, E. V.** (2022). Gerçek dünya mühendislik tasarım problemlerinin çözümünde kullanılan metasezgisel optimizasyon algoritmalarının performanslarının incelenmesi. *International Journal of Innovative Engineering Applications*, *6*(1), 65-74.

[51] **Arora, J. S.** (2004). *Introduction to optimum design*. Elsevier.

[52] **ÖZBAY, F. A., & Özbay, E.** MARTI OPTİMİZASYON ALGORİTMASININ KISITLI MÜHENDİSLİK TASARIM PROBLEMLERİ İÇİN PERFORMANS ANALİZİ. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, *8*(15), 469-485.

[53**] He, X., & Zhou, Y.** (2018). Enhancing the performance of differential evolution with covariance matrix self-adaptation. *Applied Soft Computing*, *64*, 227-243.

[54] **Dhiman, G.** (2021). ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Engineering with Computers*, *37*, 323-353.

[55] **Zhao, W., Wang, L., & Mirjalili, S.** (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, 114194.