

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



**ARTIMLI POPÜLASYON ATEŞBÖCEĞİ OPTİMİZASYON  
ALGORİTMASI**

**MURAT İŞMAN**

**YÜKSEK LİSANS TEZİ**

**Jüri Üyeleri:**    **Dr. Öğr. Üyesi Gültekin KUVAT**                      **(Tez Danışmanı)**  
                         **Prof. Dr. Metin DEMİRTAŞ**  
                         **Doç. Dr. Serdar ÖZYÖN**

**BALIKESİR, NİSAN – 2025**

## **ETİK BEYAN**

Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak tarafımda hazırlanan “**Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması**” başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

**Murat İŞMAN**

## ÖZET

**ARTIMLI POPÜLASYON ATEŞBÖCEĞİ OPTİMİZASYON ALGORİTMASI**  
**YÜKSEK LİSANS TEZİ**  
**MURAT İŞMAN**  
**BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**  
**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**  
**(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ GÜLTEKİN KUVAT)**  
**BALIKESİR, NİSAN - 2025**

Ateşböceği Algoritması (FA), doğadaki ateşböceklerinin ışık yayma ve eş bulma davranışlarından ilham alan bir metasezgisel optimizasyon algoritmasıdır. Bu çalışmada FA'nın yerel optimumlara takılma ihtimalini azaltmak ve daha başarılı sonuçlar üretebilmesini sağlamak amacıyla Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması (IFA) geliştirilmiştir. Artımlı popülasyon, belirli aralıklarla popülasyona yeni çözümlerin katılmasını ifade eder. Popülasyona katılacak yeni çözümler belirlenirken farklı yöntemler kullanılır. Bu çalışmada dört farklı yöntem kullanılarak yeni çözümler üretilmiş ve üretilen çözümler 5 ve 10 iterasyon aralıklarıyla popülasyona ilave edilmiştir. Bunun yanında, sosyal öğrenmenin IFA üzerindeki etkisini araştırmak amacıyla artım işleminin 100. iterasyondan sonra başlatıldığı Sosyal Öğrenmeli Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması (SLIFA) uygulanmıştır. Elde edilen verilere göre, IFA ve SLIFA sonuçlarının birçok durum için FA sonuçlarından daha başarılı olduğu gösterilmiştir.

**ANAHTAR KELİMELER:** Ateşböceği optimizasyon algoritması, artımlı popülasyon, sosyal öğrenme, artım aralığı

## **ABSTRACT**

### **INCREMENTAL POPULATION FIREFLY OPTIMIZATION ALGORITHM**

**MSC THESIS**

**MURAT İŞMAN**

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**ASSIST. PROF. DR. GÜLTEKİN KUVAT**

**BALIKESİR, APRIL - 2025**

The Firefly Algorithm (FA) is a metaheuristic optimization algorithm inspired by the light-emitting and mate-finding behaviors of fireflies in nature. In this study, Incremental Population Firefly Optimization Algorithm (IFA) was developed to reduce the probability of the FA getting stuck in local optima and to enhance its optimization performance. The concept of incremental population refers to the periodic inclusion of new solutions into the population. Various methods are employed to determine the new solutions to be added. In this study, four different methods were used to generate new solutions, which were then incorporated into the population at intervals of 5 and 10 iterations. Additionally, to investigate the impact of social learning on IFA, a variant called the Social Learning-Based Incremental Population Firefly Optimization Algorithm (SLIFA) was implemented, where the increment process begins after the 100th iteration. According to the obtained results, it was demonstrated that the IFA and SLIFA yielded better performance than the standard FA in many scenarios.

**KEYWORDS:** Firefly optimization algorithm, incremental population, social learning, increment interval

Science Code / Codes : 93438

Page Number: 52

# İÇİNDEKİLER

## Sayfa

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>İÇİNDEKİLER</b> .....	<b>iii</b>
<b>KISALTMALAR LİSTESİ</b> .....	<b>vi</b>
<b>ÖNSÖZ</b> .....	<b>vii</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. ATEŞBÖCEĞİ OPTİMİZASYON ALGORİTMASI</b> .....	<b>5</b>
<b>3. ARTIMLI POPÜLASYON ATEŞBÖCEĞİ OPTİMİZASYON ALGORİTMASI</b> 9	
3.1 Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması Uygulaması .....	11
3.1.1 Deney 1 .....	11
3.1.2 Deney 2 .....	12
3.1.3 Deney 3 .....	13
3.1.4 Deney 4 .....	14
3.2 Çalışmada Kullanılan Test Fonksiyonları ve Grafikler .....	17
3.2.1 Rastrigin Fonksiyonu .....	17
3.2.2 Sphere Fonksiyonu.....	18
3.2.3 Ackley Fonksiyonu .....	18
3.2.4 Rosenbrock Fonksiyonu.....	19
3.2.5 Griewank Fonksiyonu .....	20
3.2.6 Schwefel Fonksiyonu .....	21
3.2.7 Zakharov Fonksiyonu .....	22
3.2.8 Levy Fonksiyonu.....	23
3.2.9 Sum Squares Fonksiyonu .....	24
3.2.1 Dixon Price Fonksiyonu.....	25
3.2.2 Bent Cigar Fonksiyonu .....	26
3.2.3 Weierstrass Fonksiyonu .....	27
3.3 Deneysel Çalışmalar .....	29
3.4 Deney Çalışmalarının Grafikselsel Analizi .....	33
<b>4. SONUÇ VE ÖNERİLER</b> .....	<b>46</b>
<b>5. KAYNAKLAR (IEEE)</b> .....	<b>47</b>
<b>ÖZGEÇMİŞ</b> .....	<b>52</b>

## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1: Ateşböceği algoritması akış diagramı.....	8
Şekil 3.1: Artımlı ateşböceği algoritması akış diagramı. ....	15
Şekil 3.2: Rastrigin fonksiyonunun görünümü .....	17
Şekil 3.3: Sphere fonksiyonunun görünümü .....	18
Şekil 3.4: Ackley fonksiyonunun görünümü .....	19
Şekil 3.5: Rosenbrock fonksiyonunun görünümü .....	20
Şekil 3.6: Griewank fonksiyonunun görünümüleri .....	21
Şekil 3.7: Schwefel fonksiyonunun görünümü .....	22
Şekil 3.8: Zakharov fonksiyonunun görünümü.....	23
Şekil 3.9: Levy fonksiyonunun görünümü.....	24
Şekil 3.10: Sum Squares fonksiyonunun görünümü .....	25
Şekil 3.11: Dixon Price fonksiyonunun görünümü .....	26
Şekil 3.12: Bent Cigar fonksiyonunun görünümü .....	27
Şekil 3.13: Weierstrass fonksiyonunun görünümü .....	28
Şekil 3.14: Rastrigin fonksiyonunun Deney 1,2,3,4 değerleri. ....	33
Şekil 3.15: Sphere fonksiyonunun Deney 1,2,3,4 değerleri.....	34
Şekil 3.16: Ackley fonksiyonunun Deney 1,2,3,4 değerleri. ....	35
Şekil 3.17: Rosenbrock fonksiyonunun Deney 1,2,3,4 değerleri.....	36
Şekil 3.18: Griewank fonksiyonunun Deney 1,2,3,4 değerleri. ....	37
Şekil 3.19: Schwefel fonksiyonunun Deney 1,2,3,4 değerleri. ....	38
Şekil 3.20: Zakharov fonksiyonunun Deney 1,2,3,4 değerleri. ....	40
Şekil 3.21: Levy fonksiyonunun Deney 1,2,3,4 değerleri. ....	41
Şekil 3.22: Sum Squares fonksiyonunun Deney 1,2,3,4 değerleri.....	42
Şekil 3.23: Dixon Price fonksiyonunun Deney 1,2,3,4 değerleri.....	43
Şekil 3.24: Bent Cigar fonksiyonunun Deney 1,2,3,4 değerleri. ....	44
Şekil 3.25: Weierstrass fonksiyonunun Deney 1,2,3,4 değerleri. ....	45

## TABLO LİSTESİ

### Sayfa

<b>Tablo 3.1:</b> Artımlı popülasyon ateşböceği optimizasyon algoritmasında popülasyona yeni birey ekleme kuralları. ....	11
<b>Tablo 3.2:</b> Kullanılan fonksiyonlar.....	16
<b>Tablo 3.3:</b> Deney 1 sonuçları.....	29
<b>Tablo 3.4:</b> Deney 2 sonuçları.....	30
<b>Tablo 3.5:</b> Deney 3 sonuçları.....	31
<b>Tablo 3.6:</b> Deney 4 sonuçları.....	32

## KISALTMALAR LİSTESİ

<b>ACO</b>	:Karıncı Kolonisi Optimizasyonu
<b>GA</b>	:Genetik Algoritmalar
<b>DEA</b>	:Diferansiyel Gelişim Algoritması
<b>PSO</b>	:Parçacık Sürü Optimizasyonu
<b>COA</b>	:Guguk Kuşu Optimizasyonu
<b>GWO</b>	:Gri Kurt Optimizasyon Algoritması
<b>FA</b>	:Ateşböceği Algoritması
<b>ATP</b>	:Adenozin Trifosfat
<b>PBIL</b>	:Popülasyon Tabanlı Artımlı Öğrenme
<b>GSA</b>	:Yerçekimi Arama Algoritması
<b>IGSA</b>	:Artımlı Yerçekimi Algoritması
<b>IPOP</b>	:Artımsal Popülasyon
<b>PBIL</b>	:Popülasyon Tabanlı Artımlı Öğrenme
<b>CSA</b>	:Karga Arama Algoritması
<b>ABC</b>	:Yapay Arı Kolonisi Algoritması
<b>IFA</b>	:Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması
<b>SLIFA</b>	:Sosyal Öğrenmeli Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması
<b>FA20</b>	:Popülasyon Büyüklüğü 20 Olan Ateşböceği Optimizasyon Algoritması
<b>FA45</b>	:Popülasyon Büyüklüğü 45 Olan Ateşböceği Optimizasyon Algoritması
<b>FA70</b>	:Popülasyon Büyüklüğü 70 Olan Ateşböceği Optimizasyon Algoritması
<b>FA36</b>	:Popülasyon Büyüklüğü 36 Olan Ateşböceği Optimizasyon Algoritması
<b>FA52</b>	:Popülasyon Büyüklüğü 52 Olan Ateşböceği Optimizasyon Algoritması

## ÖNSÖZ

Yüksek lisans eğitimim ve tez çalışmam boyunca değerli bilgi birikimi ve tecrübelerini her daim benimle paylaşan ve bana yol gösteren çok kıymetli hocam ve danışmanım Sayın Dr.Öğr. Üyesi Gültekin KUVAT'a, uygulama geliştirme aşamasında bana desteğini esirgemeyen kıymetli arkadaşım Mustafa GÜÇKAN'a ve çalışmalarım boyunca her zaman desteğiyle yanımda olan sevgili eşim Şeyma'ya saygı, sevgi ve teşekkürlerimi sunarım.

**Balıkesir, 2025**

**Murat İŞMAN**

# 1. GİRİŞ

Optimizasyon, belirli bir amaca ulaşmak için sistematik bir şekilde ilerleyen ve çeşitli kısıtlar altında en uygun çözümü arayan bir süreçtir. Bu süreçte, mevcut problemler analiz edilerek hedefe yönelik adımlar atılır ve farklı koşullar dikkate alınır. Temel amaç, elde edilebilecek en verimli ve mantıklı sonuçlara ulaşmaktır.

Her optimizasyon probleminin temelinde bir amaç fonksiyonu bulunur ve bu fonksiyon, belirli sınırlamalar çerçevesinde optimize edilmeye çalışılır. Burada önemli olan, fonksiyonun maksimum veya minimum değerini belirleyerek en iyi performansı sağlamaktır. Örneğin, bir üretim firması işletme maliyetlerini en aza indirmeyi hedeflerken, bir lojistik şirketi teslimat sürelerini kısaltmayı amaçlayabilir.

Optimizasyon sürecinin başarılı olabilmesi için doğru bir matematiksel modelin kurulması ve uygun analiz yöntemlerinin seçilmesi kritik öneme sahiptir. Karmaşık problemlerin çözümünde etkili stratejiler belirlenmeli ve farklı senaryolar göz önünde bulundurularak en iyi sonuç elde edilmelidir.

Teknoloji hızla gelişmektedir. Böylelikle günlük yaşamdan sanayiye, sağlıktan eğitime kadar pek çok alanda büyük değişimler ortaya çıkmıştır. Sağlık alanındaki ilerlemeler, akıllı cihazların artışı, dijitalleşmenin hızlanması ve yapay zeka gibi yenilikler, yaşam kalitemizi iyileştirmiştir. Ancak bu hızlı dönüşüm, daha önce karşılaşılmayan kompleks ve çok boyutlu sorunları da ortaya çıkarmıştır. Bu sorunlar, birçok sektörü ve alanı etkileyen büyük zorluklar doğurmaktadır. Veri güvenliği, iş gücü değişimleri ve çevresel sorunlar gibi konular, çözüm bekleyen başlıca sorunlar arasında yer alır. Bu karmaşık problemleri çözmek için yenilikçi yöntemler geliştirilmiştir. Özellikle doğa esinli, fizik veya matematik, tek çözüm veya yinelemeli, evrimsel algoritma, sürü zekası ve popülasyon tabanlı dikkat çeken çözüm yöntemleri arasındadır [1], [2], [3].

Doğa ve yaşama süreçleri, detaylı bir şekilde gözlemlendiğinde birçok farklı sürecin olduğu görülmektedir. Bu süreçlerin içerisinde bulunan canlıların öğrenmeye veya tecrübeye dayanmayan, kendiliğinden ortaya çıkan davranış süreçleri incelenerek üretilen doğa esinli optimizasyon algoritmaları karmaşık optimizasyon problemlerini çözmeyi amaçlamaktadır. Böylelikle doğa esinli optimizasyon algoritmaları en uygun, en rahat, en ekonomik, en kısa, en sağlam çözümlere ulaşmayı sağlamış olmaktadır [1].

Geçen yıllar boyunca birçok doğadan esinlenen optimizasyon algoritması ortaya çıkarılmıştır [4]. Örnek olarak aşağıdaki algoritmalar gösterilebilir.

- Karınca Kolonisi Optimizasyonu (Ant Colony Opt.) (ACO): Karıncaların yiyecek ararken ve yuvalarına geri dönerken izledikleri yolları taklit ederek üretilmiş olan bir algoritmadır [5].

- Genetik Algoritmalar (Genetic algorithms) (GA): Doğal seçim ve genetik evrim prensiplerine dayanan bir doğa esinli optimizasyon algoritmasıdır [6].

- Karga Arama Algoritması (CSA): Kargaların sürü halinde hareket etme ve besin arama davranışlarını taklit eder [7].

- Yapay Arı Kolonisi Optimizasyonu (Artificial Bee Colony Opt.)(ABC): Arıların yiyecek arama davranışları esas alınarak geliştirilmiştir [2].

- Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm) (DEA): Evrimsel bir algoritma olarak geliştirilmiştir [8].

- Parçacık Sürü Optimizasyonu (Partical Swarm Optimization) (PSO): Bir popülasyon içindeki her bireyin çözüm uzayında diğer bireylerle etkileşimde bulunarak optimum çözüme ulaşmasını amaçlamaktadır [9].

- Guguk Kuşu Optimizasyonu (Cuckoo Optimization Algorithm) (COA): Guguk kuşların, yumurtalarını başka kuşların yuvalarına bırakarak hayatta kalma şansını artırmasını taklit eder [10].

- Gri Kurt Optimizasyon Algoritması (Grey wolf optimizer) (GWO): Gri kurtların avlanma davranışlarından ilham almıştır [11].

- Ateşböceği Algoritması (FA): Ateşböceklerinin parklaklığıyla orantılı olarak birbirini çekim gücünden hareketle ortaya çıkarılmıştır [12], [13], [14].

Bunlar ve bunların benzerleri optimizasyon algoritmaları birçok problemde iyi sonuçlar üretmişlerdir [15]. Her ne kadar iyi sonuçlar üretseler de esinlenen doğa olaylarının birçok farklı yönü olabileceği için bu algoritmaların performansını arttıracak çalışmalar yapılmaktadır [14]. Optimizasyon unsurlarının ayarlanması, hibrit yaklaşımlar, dinamik güncelleme ve artırım bu yetenekleri geliştirmek için kullanılan yöntemlerden bazılarıdır [1].

Çalışmamızda performans arttırmak için odaklandığımız yöntem artımlı optimizasyon yöntemidir. Karmaşık veya büyük ölçekli problemleri çözmekte karşılaşılan zorlukları

aşmak için ihtiyaç duyulan bu yaklaşımlar; genellikle doğrudan veya tek bir adımda optimize edilemeyecek kadar kapsamlıdır. Bu yöntem, bir problemi adım adım iyileştirme sürecini ifade eder ve her adımda yapılan yerel iyileştirmelerle çözümün genel olarak daha iyi bir hale getirilmesini amaçlar. Problemi zamanla küçük parçalara bölerek ele alır ve her adımda çözümün dinamik olarak güncellenmesini sağlayarak değişken koşullara hızlı bir şekilde uyum sağlar. Özellikle karmaşık yapıdaki problemlerde çözümün daha erişilebilir ve etkin hale getirilmesini sağladığı için birçok alanda yaygın çözüm sağlayacağı düşünülmektedir [16].

Her doğa esinli optimizasyon algoritmasının optimum ulaşma sürecinde kendine özgü zayıf yönleri bulunmaktadır. FA'nın bu kapsamda zayıf yönleri aşağıdaki gibidir.

- Ateşböceklerinin parlaklığına bağlı çekiciliğinden kaynaklı bir noktaya hızlı yönelme çözüm aralığını yeterince keşfedememesine yol açmaktadır. Bunun sonucu olarak kompleks problemlerle karşı karşıya kalındığında yerel optimuma takılma handikapıyla karşı karşıya kalmaktadır [17], [18], [19].

- Ateşböcekleri çekicilik ve parlaklık parametrelerine bağlı olarak optimum çözüme ulaşırken yavaş yakınsama problemiyle karşı karşıya kalınmaktadır [18], [20] .

- Keşif ve sömürü dengesinin optimize edilememesi. Optimum çözüme yakın bölgelerin daha ayrıntılı incelenmesi gereken durumlarda, bu denge eksik kalmaktadır [17]. Bu zayıflıkları gidermek için 12 farklı fonksiyonda farklı yaklaşımlar uygulanarak FA çalıştırılmış ve sonuçlar gözlemlenmiştir. Bu yaklaşımlar;

1. Sabit olan popülasyon büyüklüğüyle algoritmayı çalıştırmak.
2. Popülasyon büyüklüğünü belirlenen iterasyon tamamlanana kadar 5 ve 10 iterasyonda 1 ateşböceği arttırmak ve bu artımlarda yeni üretilen popülasyon üyesini 4 farklı şekilde belirlemek.
3. Popülasyon büyüklüğünün 5 ve 10 iterasyonda bir arttırıldığında algoritmanın çalışması boyunca toplam değerlendirilen birey sayısına eşdeğer olan, karşılık değer olarak adlandırılan sabit bir popülasyon büyüklüğüyle (70 ve 45) artım olmadan temel algoritmayı çalıştırmak.
4. Popülasyon büyüklüğünü ilk 100 iterasyonda sabit tutup sonrasında belirlenen iterasyon tamamlanana kadar 5 ve 10 iterasyonda 1 ateşböceği arttırmak, bu artımda yeni üretilen popülasyon üyesini 4 farklı durumda üreterek algoritmayı çalıştırmak.

5. Popülasyon büyüklüğünün ilk 100 iterasyonda sabit tutulduğu ve sonrasında belirlenen iterasyon sayısına ulaşana kadar her 5 ve 10 iterasyonda bir arttırıldığında algoritmanın çalışması boyunca toplam değerlendirilen birey sayısına denk olacak karşılık değerler için sabit bir popülasyon büyüklüğüyle (52 ve 36) artım olmadan temel algoritmayı çalıştırmak.

## 2. ATEŞBÖCEĞİ OPTİMİZASYON ALGORİTMASI

Ateşböceklerinin ışık yayma özelliği biyoluminesans adı verilen bir süreçle gerçekleşir ve bu süreç, karın bölgelerinde bulunan özel organlarda meydana gelir. Kimyasal bir reaksiyona dayanan biyoluminesans, luciferin adlı bir molekülün luciferaz enzimi yardımıyla oksijen ve ATP ile etkileşime girmesi sonucu oluşur. Bu etkileşim, enerjinin neredeyse tamamını görünür ışığa çevirir ve bu da "soğuk ışık" olarak adlandırılır, çünkü ısı üretimi minimum düzeydedir. Ateşböceklerinin ışığı genellikle yeşil, sarı veya turuncu tonlarında görünür [21], [22].

Bu ışık, öncelikli olarak eş bulmak için kullanılır. Erkek ateşböcekleri, belirli bir ritimde ışık yayarak dişilere sinyal gönderir, dişiler de bu ritmi tanıyarak yanıt verir. Ayrıca bazı türler, ışığı avcılarını caydırmak veya kimyasal savunmalarını göstermek için kullanabilirken, diğerleri avlarını cezbetmek için ışıklarını kullanır. Çeşitli türlerin ışık yayma desenleri birbirinden farklıdır ve bu desenler türlerin ayırt edilmesini sağlar. Ateşböceklerinin biyoluminesans yeteneği, başlangıçta yırtıcıları caydırmak için evrimleşmiş olabilir, ancak zamanla çiftleşme ve iletişim için uyarlanmıştır. Bu süreç, türler arasında farklı ışık desenlerinin gelişmesine yol açmış ve yeni türlerin ortaya çıkmasını desteklemiştir. Ayrıca, bu ritüeller, genetik olarak üstün bireylerin seçilmesine katkıda bulunarak evrimsel baskılar oluşturur [21], [23].

Ateşböceklerinin bu üstün yeteneği hem bilimsel hem de çevresel açıdan büyük bir ilgi uyandırmaya devam etmektedir. Ateşböceklerinin bu yeteneklerinden hareketle doğa esinli bir optimizasyon algoritması olan FA türetilmiştir [12], [20]. Bu algoritmada ışık yayma davranışı bir "çekim" kavramıyla ilişkilendirilir. Daha parlak bir ateşböceği, daha güçlü bir çözüme veya optimizasyon probleminde daha iyi bir noktaya karşılık gelir. Daha parlak ateşböceğinin parlaklığı problemin daha iyiye gitmesini sağlayacaktır [20]. Bu algoritmada parlaklığın yanında ateşböceklerinin arasındaki mesafe de çekicilik açısından önemlidir. Mesafe arttıkça ateşböceğinin çekiciliği azalacaktır [12], [20]. Bir sürü algoritması olan FA'da temel prensip diğer sürü algoritmalarında olduğu gibi sürünün en iyi pozisyona, çekiciliğe sahip bireyine yönelim vardır. FA'nın temelinde 3 kural vardır. Bu kurallar:

1. Değerlendirilen ateşböcekleri cinsiyetten bağımsız bir şekilde birbirlerini çekerek hareket etmektedir.

2. Ateşböceklerinin birbirlerine karşı çekiciliği sahip oldukları parlaklıkla doğru orantılı olarak kabul edilir. Bir ateşböceği tarafından gözlemlenen diğer ateşböceğinin parlaklığı aralarındaki mesafeye bağlı olarak değişir.
3. Algoritmada ateşböceklerinin parlaklığı uygunluk fonksiyonundan çıkan değerle eşleştirilmiştir [24].

Algoritmada parlaklık aradaki mesafe arttıkça ters kare yasasına uygun olarak azalmaktadır. Işığın kaynağından ( $I_s$ ) [14]  $r$  uzaklıkta oluşan ışığın şiddeti ( $I(r)$ ) Denklem(2.1)'de verilen denklem ile hesaplanır.

$$I(r) = \frac{I_s}{r^2} \quad (2.1)$$

Işık kaynağından uzaklaştıkça yoğunluğu azalır ve ışık ortamda soğurur [12], [13]. Bu formülasyon Denklem (2.2)'de formülize edilmiştir. Işığın soğurulması sabit olan bir ışık emilim katsayısı ( $\gamma$ ) dikkate alınarak gerçekleşmektedir. Bu katsayı ateşböceği algoritmasının yakınsama hızı ve tavrına etki edebilmektedir [24]. Önceki çalışmalarda bu katsayı genellikle sabit 1 ya da  $[0,10]$  aralığında değerlendirilmiştir [25].  $I_0$ ,  $r = 0$  olduğundaki ışık kaynağının şiddeti olarak değerlendirilir. Ayrıca sıfıra bölünmede karşılaşılabilecek tanımsız olma durumuna karşın mesafe gauss dağılımı olarak yazılabilir [26].

$$I(r) = I_0 e^{-\gamma r^2} \quad (2.2)$$

Algoritmada ateşböceklerinin birbirleri arasındaki çekicilik Denklem (2.3) ile hesaplanmaktadır. Bu denklemde  $\beta_0$  herhangi bir ateşböceğinin diğer ateşböceği ile aralarındaki mesafe  $r = 0$  olarak düşünüldüğündeki çekiciliği temsil ederken  $\beta(r)$  de  $\beta_0$  çekiciliğine sahip bir ateşböceğinin  $r$  mesafedeki çekiciliği temsil eder.

$$\beta(r) = \beta_0 \cdot e^{-\gamma r^2} \quad (2.3)$$

$i$  ve  $j$  iki ateşböceği olarak düşünülürken  $d$  boyutlu bir problemde ateşböcekleri arasındaki uzaklık Denklem (2.4) ile hesaplanmaktadır[27].

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2.4)$$

Çekici olana doğru yönelmiş bir ateşböceğinin yeni konumu Denklem (2.5) teki formül ile hesaplanmaktadır. Şekil 2.1’de FA’nın akış diyagramı görülebilmektedir.

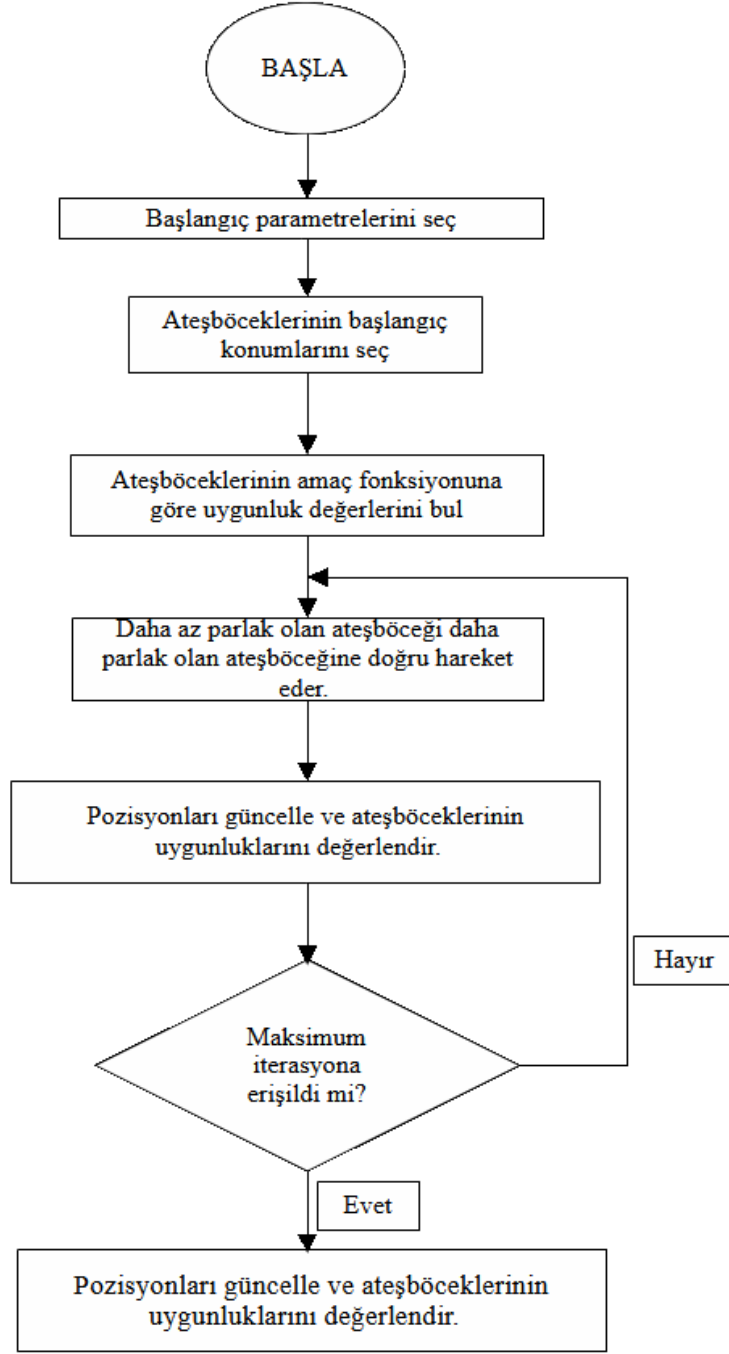
$$x_i^{(t+1)} = x_i^{(t)} + \beta \cdot e^{-\gamma r_{ij}^2} \cdot (x_j^{(t)} - x_i^{(t)}) + \alpha \cdot (\text{rand} - 0.5) \quad (2.5)$$

$x_i^{(t)}$  :  $t$  ‘inci iterasyondaki ateşböceği ( $i$ )’nin mevcut konumu.

$x_j^{(t)}$  :  $t$  ‘inci iterasyondaki ateşböceği ( $j$ )’nin mevcut konumu.

$\alpha$  :Rastgelelik faktörü [27]

rand :[0,1] aralığında bir uniform dağılımdan rastgele çekilen bir sayı [28].



Şekil 2.1: Ateşböceği algoritması akış diagramı.

### 3. ARTIMLI POPÜLASYON ATEŞBÖCEĞİ OPTİMİZASYON ALGORİTMASI

Artımlı popülasyon metasezgisel algoritmalar sabit bir popülasyon büyüklüğü yerine algoritmanın çalışması esnasında çalışma uzayında artım yaparak algoritmanın daha iyi bir sonuca ulaşmasını hedefleyen yöntemlerdir. Özellikle büyük veri setleri ve karmaşık hesaplama problemleriyle çalışırken temel yöntemlerden çok daha verimli sonuçlar elde edilebilmektedir. Küçük adımlarla çözümün sürekli olarak iyileştirilmesine olanak tanınmaktadır. Temel metasezgisel optimizasyon algoritmaları, genellikle tüm veri kümesini bir arada işleyerek çözüm üretirken artımlı optimizasyon bu süreci adım adım gerçekleştirir. Bu yaklaşım, verilerin tümünü aynı anda belleğe almak ya da dar bir çözüm uzayında çalışmak yerine adım adım çözüm uzayını ve popülasyonu büyüterek çözüm üretir, bu da verimliliği önemli ölçüde artırır. Bu sayede hesaplama maliyeti düşer ve büyük veri ile çalışırken zaman kazanılır [29]. Artımlı algoritmalar, her iterasyonda mevcut bilgiye dayanarak çözümü adım adım geliştirir. Bu esneklik, verinin sürekli olarak değiştiği dinamik olan sistemlerde de faydalıdır. Örneğin, zaman serisi verileri veya gerçek zamanlı veri akışları gibi sürekli güncellenen verilerle çalışırken artımlı algoritmalar, her yeni veri noktasında çözümü hemen güncelleyerek daha hızlı ve daha etkili sonuçlar sunar. Bu dinamik yapı, özellikle makine öğrenmesi ve derin öğrenme gibi büyük veri kümeleriyle yapılan model eğitimlerinde, paralel işleme ve hızlı hesaplama olanakları sağlar. Artımlı optimizasyonun temel avantajlarından bir başkası da çözüm uzayında gezinme hızıdır; geleneksel algoritmalar tüm çözüm uzayını geniş bir şekilde tararken artımlı algoritmalar bu süreci hızlandırır ve daha verimli bir şekilde önemli bölgelere odaklanır. Bu nedenle, özellikle büyük ve karmaşık veri setleriyle çalışırken artımlı optimizasyon hem hesaplama açısından daha verimlidir hem de daha hızlı sonuçlar elde edilmesini sağlar.

Artımlı optimizasyon yaklaşımlarının etkinliğini artırmaya yönelik önemli stratejilerden biri de artımlı sosyal öğrenme (ISL) mekanizmasının entegrasyonudur. Bu yöntem, popülasyona sonradan dâhil olan bireylerin, mevcut bireylerden bilgi edinerek daha hızlı bir adaptasyon süreci geçirmelerini sağlar. Böylece yeni bireyler, çözüm uzayını sıfırdan keşfetmek yerine, mevcut deneyimlerden yararlanarak arama sürecine daha donanımlı bir şekilde katılırlar [30].

ISL tabanlı algoritmalar, başlangıçta küçük bir popülasyonla çalışmaya başlar ve zamanla yeni bireylerin eklenmesiyle birlikte popülasyon büyür. Yeni eklenen bireyler, mevcut popülasyonla etkileşim kurarak bilgi paylaşımı gerçekleştirir. Bu yapı, çözüm uzayının daha etkin ve kapsamlı şekilde taranmasına imkân tanırken, aynı zamanda algoritmanın çeşitliliğini korumasına da katkıda bulunur. Artımlı optimizasyon algoritmalarından ISL’de bilgi edinimi sadece keşfe dayanmamakta, mevcut bireylerin deneyimlerinden türeyen kolektif bir öğrenme süreci de etkin bir biçimde kullanılmaktadır [30]

Popülasyon boyutuyla ilgili yapılan bir çalışmada PSO’nun popülasyon boyutu, çözümün kalitesine göre dinamik olarak ayarlanmış ve popülasyonun artışı ile çözüm kalitesinin iyileştiği gözlemlenmiştir [31]. Yapılan bir diğer çalışmada Popülasyon Tabanlı Artımlı Öğrenme (PBIL) yöntemi keşfedilmektedir. PBIL, jenerasyonel genetik algoritmanın mekanizmalarını, basit bir rekabetçi öğrenme ile birleştiren bir yaklaşımdır. Bu iki yöntemin birleşimi, GA’dan çok daha basit bir araç ortaya çıkarmakta ve büyük bir optimizasyon problemi kümesinde hız ve doğruluk açısından GA’yı geride bırakmaktadır [29]. Yerçekimi Arama Algoritması (GSA) üzerine yapılan bir diğer çalışmada da GSA geliştirilerek yeni ajanlar belirli adımlarla eklenmiş ve maksimum popülasyon sayısına ulaşıldığında ekleme işlemi sonlandırılmıştır. Arama, istenen fonksiyon çağrısı sağlanana kadar devam etmiştir. Bu geliştirilmiş algoritma, Artımlı Yerçekimi Arama Algoritması (IGSA) olarak adlandırılmıştır. Ajan ekleme süreci farklı yöntemlerle gerçekleştirilmiş ve bu yöntemlerin her biri için IGSA’nın sonuçları, literatürdeki GSA sonuçlarıyla karşılaştırılmıştır [32]. IGSA’ya benzer şekilde CSA üzerine yapılan bir diğer çalışmada da genişleyen bir popülasyonu temel alan PBIL stratejisiyle çözüm çeşitliliği sağlanması hedeflenmiştir. Geliştirilen dört CSA versiyonu, 100 boyutlu test fonksiyonuna uygulanarak performansları değerlendirilmiştir [1]. Bunun yanında farklı PSO [33], ACO [33] ve ABC [34] üzerine yapılan çalışmalarda da dinamik popülasyon yapısının sürü algoritmalarının performansı üzerinde olumlu bir etkisi ortaya konmuştur.

Artımlı sosyal öğrenme süreci, popülasyonlardaki bireylerin çevrelerinden ve birbirlerinden bilgi alarak davranışlarını iyileştirdiği bir süreç [35] olduğundan bu yaklaşım kullanılarak geliştirilen artımlı popülasyon ateşböceği algoritmasının akış diyagramı aşağıda Şekil 3.1’de [1] verilmiştir. Çalışmamızda FA üzerine geliştirmeler yapılmıştır. Önerilen algoritmada sabit popülasyon yerine, artımlı popülasyon örneklerine benzer şekilde iterasyon sayısı arttıkça belirli bir artım aralığı kullanılarak popülasyon büyüklüğü arttırılmıştır. Bu

bağlamda, popülasyona yeni birey eklenmesi durumu dört farklı senaryo için uygulanmıştır [1]. Aşağıda verilen Tablo 3.1'de belirtilen bu dört durum için algoritmanın performansı ölçülmüştür. Bu analiz, 12 farklı test fonksiyonu ve iki farklı artım aralığı (5 ve 10) kullanılarak gerçekleştirilmiştir. Bunun yanında artım ile yapılan aramayı temsil eden karşılık değer sabit popülasyon büyüklüğü olacak şekilde alınmış ve FA çalıştırılmıştır. Tüm bu süreçler sonucunda, farklı artım aralıkları ve popülasyon büyüklükleri için ayrıntılı analiz yapılmıştır. Bu durumlar arasındaki karşılaştırmalar, algoritmanın performansını optimize etmek ve en verimli senaryoyu belirlemek için önemli veriler sağlamıştır.

**Tablo 3.1:** Artımlı popülasyon ateşböceği optimizasyon algoritmasında popülasyona yeni birey ekleme kuralları.

Durumlar	Yeni Birey Ekleme Kuralı
<i>Durum-1</i>	$X_{yeni,G} = X_{eniye,G}$
<i>Durum-2</i>	$X_{yeni,G} = X_{rand,G}$
<i>Durum-3</i>	$X_{yeni,G} = X_{eniye,G} + rand(0,1) * (X_{rand,G} - X_{eniye,G})$
<i>Durum-4</i>	$X_{yeni,G} = X_{rand,G} + rand(0,1) * (X_{eniye,G} - X_{rand,G})$

Tablo 3.1 'de belirtilen yeni birey ekleme kuralındaki değerler şöyledir.

$X_{yeni,G}$ :  $G$ . jenerasyonda popülasyona eklenecek yeni ateşböceğini ifade etmektedir.

$X_{eniye,G}$ :  $G$ . jenerasyonda popülasyondaki en iyi uygunluk değerine sahip ateşböceğini ifade etmektedir.

$X_{rand,G}$ : Rastgele üretilen bireyi ifade etmektedir.

$rand(0,1)$ : 0,1 aralığında üretilen rastgele bir sayı

### 3.1 Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması Uygulaması

Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması (IFA) için Deney 1 ve Deney 2, Sosyal Öğrenmeli Artımlı Popülasyon Ateşböceği Optimizasyon Algoritması (SLIFA) için Deney 3 ve Deney 4 uygulamaları yapılmıştır.

#### 3.1.1 Deney 1

Uygulamamızda popülasyon başlangıç büyüklüğü 20, iterasyon sayısı ise 500 olarak belirlenmiştir. Bu çalışmada artım aralığı 5 alınarak her beş iterasyonda bir algoritmaya yeni bireylerin eklenmesi sağlanmıştır. Popülasyona bireyler Tablo 3.1' de belirtilen 4 farklı

durum için eklenmiş ve Tablo 3.2'deki 12 fonksiyon için değerlendirme yapılmıştır. Her durum için algoritma 50 kez Python yazılım geliştirme uygulamasında çalıştırılmış ve bu çalıştırmaların sonucunda ortaya çıkan en büyük, en küçük, ortalama ve standart sapma değerleri kayıt altına alınmıştır. Artım yapılmadan 50 kez çalıştırılan temel algoritmada çıkan değerlerle de karşılaştırma yapılmıştır.

Artım aralığı 5 ve 10 olarak belirlenmiş algoritmanın tüm çalışması boyunca değerlendirilen toplam birey sayısı hesaplanmış ve temel algoritma çalıştırılarak da bu toplam değerlendirilen birey sayısına ulaşmak için sabit bir popülasyon büyüklüğü belirlenmiştir. Bu büyüklüğe karşılık değer denmiştir. Bu deneyimizde artım aralığı 5 olduğunda bulunan karşılık değere (70) sahip popülasyon temel algoritmada çalıştırılmış ve bu da FA70 olarak kıyaslamalara eklenmiştir. FA20, IFA1, IFA2, IFA3, IFA4 ve karşılık değer popülasyon büyüklüğüne sahip FA70 durumlarına ait değerler çalışmamızda kullanılan tüm fonksiyonlar için Tablo 3.3 'te kıyaslanmış ve bu durumlar grafiklendirilmiştir. IFA1, IFA2, IFA3 ve IFA4 sırasıyla yeni birey ekleme kuralları Durum-1, Durum-2, Durum-3 ve Durum-4 uygulanarak elde edilen sonuçlardır.

Tablo 3.3'te verilen sonuçlara göre; FA20 ve FA70 ile IFA durumları karşılaştırıldığında, genel olarak IFA durumlarının daha başarılı sonuçlar verdiği görülmektedir. Birçok test fonksiyonunda hem en küçük değer hem de ortalama değer açısından IFA1 ile IFA4 durumları öne çıkmıştır. Hem ortalama değer hem de en küçük değerde IFA4'ün en iyi çıktığı fonksiyonlar; Ackley, Griewank, Levy ve Bent Cigar olurken IFA1'in en iyi çıktığı fonksiyonlar ise Schewefel, Zakharov ve Dixon Price olmuştur. Sabit popülasyonlu FA70, FA20'ye göre çoğu fonksiyonda daha iyi performans sergilese bile artımlı yapılar kadar etkili olamamıştır. Bu genel tablo, artımlı yapıların optimizasyon başarısını belirgin şekilde artırdığını göstermektedir.

### **3.1.2 Deney 2**

Başlangıçta 20 bireyden oluşan bir popülasyonla 500 iterasyonluk bir çalışma gerçekleştirilmiştir. Bu senaryoda bireyler her 10 iterasyonda bir algoritmaya dâhil edilerek popülasyon artışı yapılmıştır. Bu artış Tablo 3.1'de belirtilen 4 farklı senaryo kapsamında sağlanmış ve her bir senaryo Tablo 3.2'de yer alan 12 farklı fonksiyon üzerinde değerlendirilmiştir. Algoritma 50 kez çalıştırılmış; bu çalıştırmalar sonucunda elde edilen en küçük ve ortalama değerler kayıt altına alınmıştır.

Referans karşılaştırma amacıyla, artım yapılmadan sabit popülasyonla 50 kez çalıştırılan temel algoritmanın çıktıları da değerlendirilmiştir. Buna ek olarak, artım aralığı 10 olduğunda bulunan karşılık değer (45) popülasyon büyüklüğü belirlenmiş ve temel algoritma bu sabit büyüklükle çalıştırılarak FA45 olarak kıyaslamaya dahil edilmiştir.

FA20, IFA1, IFA2, IFA3 ve IFA4 ve karşılık değer popülasyon büyüklüğüne sahip FA45 senaryoları, çalışmamızda kullanılan tüm test fonksiyonları için karşılaştırılmış ve elde edilen sonuçlar Tablo 3.4'te verilmiştir.

FA45 popülasyon büyüklüğü ile gerçekleştirilen klasik FA algoritması ile IFA durumları karşılaştırıldığında, Deney 1'e benzer şekilde, IFA algoritmalarının genel olarak daha yüksek performans sergilediği görülmektedir. Özellikle karmaşık yapılı ve çoklu yerel minimum içeren fonksiyonlarda (örneğin Ackley, Rosenbrock, Bent Cigar ve Levy) IFA algoritmaları üstün performans göstermiştir. IFA1 ve IFA4, en başarılı sonuçlar üreten durumlar olmuştur. Ortalama ve en küçük değerlerde en iyi sonuca ulaşan IFA1 durumları Sphere, Rosenbrock, Schewefel, Zakharov fonksiyonlarında olurken, IFA4 durumları ise Ackley, Griewank, Levy ve Bent Cigar fonksiyonlarında olmuştur. Ayrıca sabit popülasyonlu FA45'in, FA20'ye göre birçok fonksiyonda daha iyi sonuçlar ürettiği de gözlemlenmiştir.

### 3.1.3 Deney 3

500 iterasyon süresince uygulanan çalışmada başlangıç popülasyonu yine 20 olarak belirlenmiştir. İlk 100 iterasyonda artım yapılmaksızın sosyal öğrenme aşamasına odaklanılmış, ardından popülasyon her 5 iterasyonda bir büyütülmüştür. Popülasyona eklenen bireyler, Tablo 3.1'de tanımlanan dört farklı senaryo kapsamında uygulanmış ve her bir senaryo, Tablo 3.2'de yer alan 12 farklı test fonksiyonu üzerinde ayrı ayrı değerlendirilmiştir.

Karşılaştırma amacıyla, artım uygulanmadan sabit popülasyonla 50 kez çalıştırılan temel algoritmanın sonuçları da analiz edilmiştir. Buna ek olarak, belirlenen karşılık değer (52) kullanılarak algoritma uygulanmış ve elde edilen sonuçlar FA52 olarak adlandırılarak analiz kapsamına dahil edilmiştir.

Sonuç olarak SLIFA1, SLIFA2, SLIFA3, SLIFA4, FA52 ve yeniden uygulanan FA20 senaryoları; çalışmada kullanılan tüm test fonksiyonları üzerinde karşılaştırılmış ve elde edilen çıktılar grafiksel olarak görselleştirilmiştir.

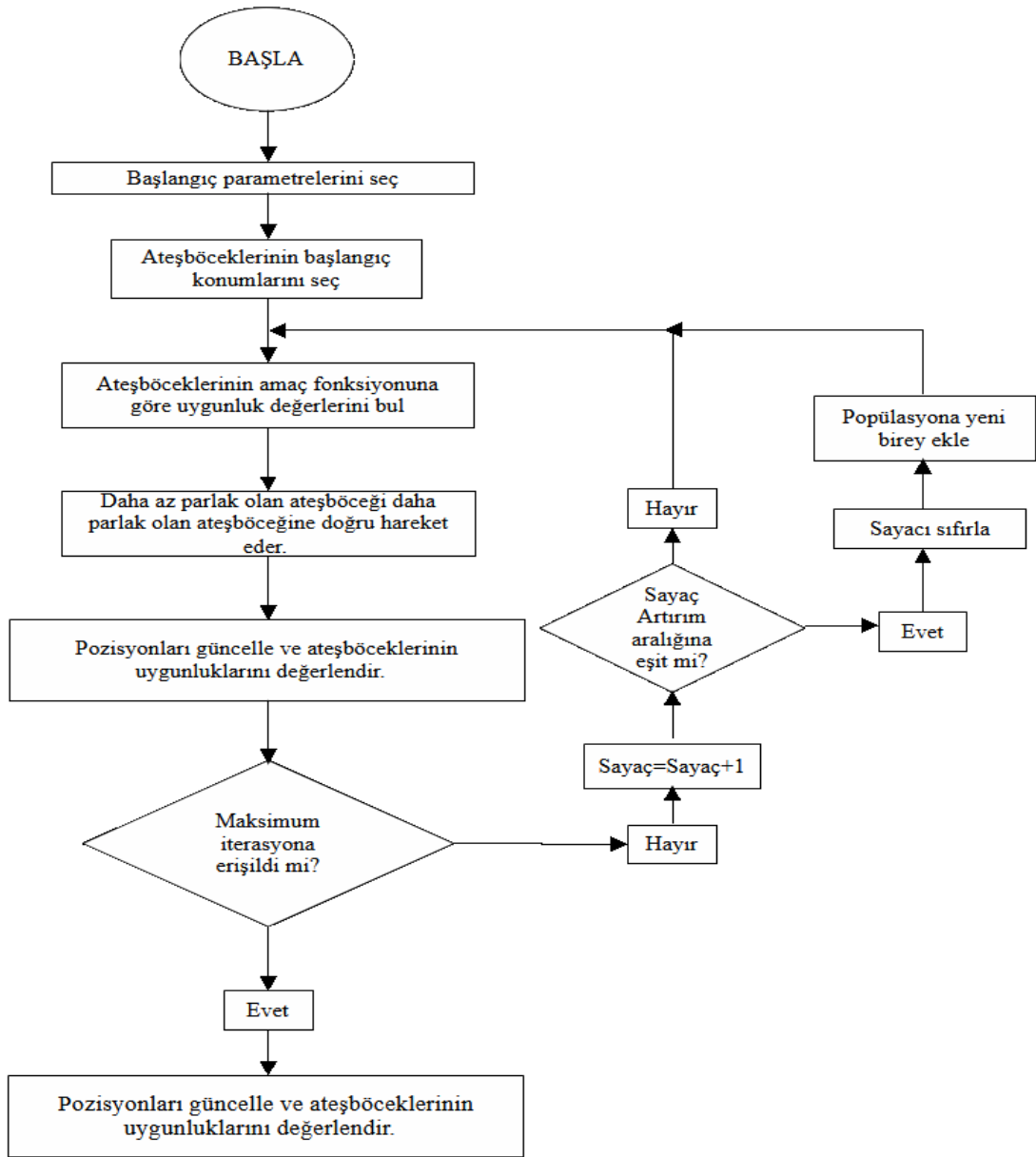
Tablo 3.5'te bulunan sonuçlara göre; özellikle karmaşık ve çok boyutlu fonksiyonlarda SLIFA4, hem en küçük hem de ortalama değerler bakımından dikkat çekici bir performans sergilemiştir. Bu fonksiyonlarda istikrarlı ve düşük sonuçlar elde etmesi, SLIFA4'ün klasik FA varyantlarına kıyasla önemli bir üstünlük sağladığını göstermektedir. Benzer şekilde, SLIFA1 algoritması da zorlu problemlerde anlamlı iyileştirmeler sunmuştur. Bu bulgular, sosyal öğrenme bileşeniyle zenginleştirilmiş artımlı yapının, özellikle karmaşık ve yüksek boyutlu fonksiyonlarda çözüm kalitesini artırarak etkili bir optimizasyon süreci sağladığını ortaya koymaktadır.

#### **3.1.4 Deney 4**

Bu deneyimizde de 20 bireyden oluşan bir başlangıç popülasyonu kullanılmış ve algoritma toplamda 500 iterasyon boyunca çalıştırılmıştır. İlk 100 iterasyondan sonra her 10 iterasyonda bir birey eklenerek artımlı yapı devreye alınmıştır. Bireyler popülasyona, Tablo 3.1'de belirtilen dört farklı durum için eklenmiş ve her bir durum, Tablo 3.2'de yer alan 12 farklı fonksiyon üzerinden değerlendirilmiştir.

Artım aralığı 10 olduğundaki karşılık değer (36) için uygulanan FA36 durumu, SLIFA1, SLIFA2, SLIFA3, SLIFA4 durumları ve yeniden uygulanan FA20 durumu çalışmamızda kullanılan tüm fonksiyonlar için karşılaştırılmış ve elde edilen sonuçlar grafiklerle görselleştirilmiştir.

Tablo 3.6 sonuçlarına göre SLIFA4 durumu diğer deneylere benzer şekilde en küçük ve ortalama değerlerde en başarılı sonuçları üretmiştir. SLIFA4 Rosenbrock, Griewank, Levy, Dixon Price ve Bent Cigar fonksiyonlarının hepsinde hem ortalama hem de en küçük değerlerde en iyi sonuçlara ulaşmıştır.



Şekil 3.1: Artımlı ateşböceği algoritması akış diagramı.

**Tablo 3.2:** Kullanılan fonksiyonlar.

Fonksiyon Formülasyonu	Fonksiyon İsmi	Çalışma Aralığı	Özellik
$f_1(x) = 10 \cdot d + \sum_{i=1}^d (x_i^2 - A \cdot \cos(2\pi x_i))$	Rastrigin	$[-5.12, 5.12]^d$	Multimodal
$f_2(x) = \sum_{i=1}^d x_i^2$	Sphere	$[-5.12, 5.12]^d$	Unimodal
$f_3(x) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos 2\pi \cdot x_i\right) + 20 + \exp(1)$	Ackley	$[-32.768, 32.768]^d$	Multimodal
$f_3(x) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos 2\pi \cdot x_i\right) + 20 + \exp(1)$	Rosenbrock	$[-5, 10]^d$	Unimodal
$f_5(x) = 1 + \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right)$	Griewank	$[-600, 600]^d$	Multimodal
$f_6(x) = 418.9829 \cdot d - \sum_{i=1}^d x_i \cdot \sin(\sqrt{ x_i })$	Schwefel	$[-500, 500]^d$	Multimodal
$f_7(x) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5 \cdot i \cdot x_i\right)^2 + \left(\sum_{i=1}^d 0.5 \cdot i \cdot x_i\right)^4$	Zakharov	$[-5, 10]^d$	Unimodal
$f_8(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)]$	Levy	$[-10, 10]^d$	Multimodal
$f_9(x) = \sum_{i=1}^d i \cdot x_i^2$	Sum Squares	$[-5.12, 5.12]^d$	Unimodal
$f_{10}(x) = (x_1 - 1)^2 + \sum_{i=2}^d i \cdot (2x_i^2 - x_{i-1})^2$	Dixon-Price	$[-10, 10]^d$	Unimodal
$f_{11}(x) = x_1^2 + 10^6 \sum_{i=2}^d x_i^2$	Bent Cigar	$[-100, 100]^d$	Unimodal
$f_{12}(x) = \sum_{i=1}^d \sum_{n=0}^k a^n \cos(2\pi b^n (x_i + 0.5)) - d \cdot \sum_{n=0}^k a^n \cos(2\pi b^n \cdot 0.5)$	Weierstrass	$[-5, 5]^d$	Multimodal

### 3.2 Çalışmada Kullanılan Test Fonksiyonları ve Grafikler

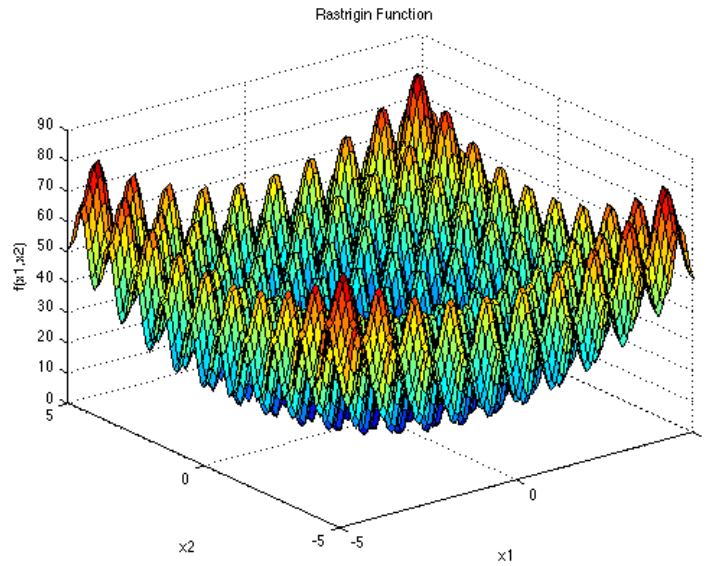
Bu bölümde uygulamalarımızda kullanılan test fonksiyonlarının özellikleri açıklanmış ve 3 boyutlu gösterim grafikleri verilmiştir.

#### 3.2.1 Rastrigin Fonksiyonu

$$f_1(x) = 10 \cdot d + \sum_{i=1}^d (x_i^2 - A \cdot \cos(2\pi x_i)) \quad (3.1)$$

$x_i \in [-5.12, 5.12]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x = (0, \dots, 0), f(x) = 0$  [36]



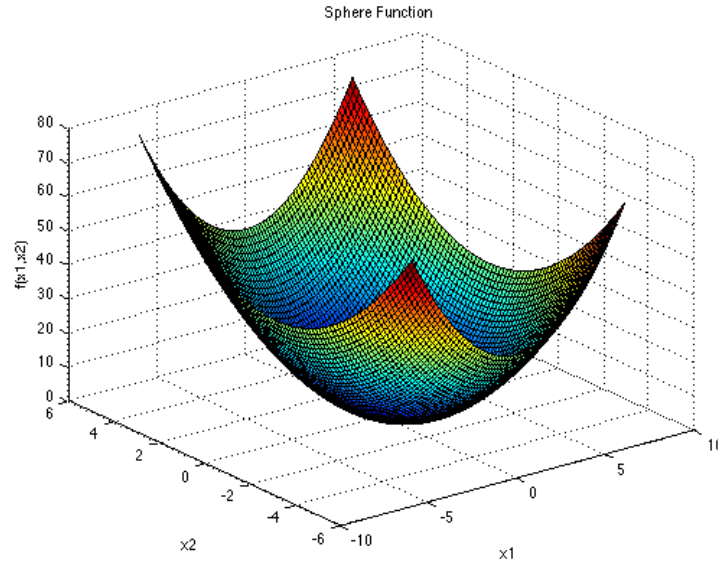
**Şekil 3.2:** Rastrigin fonksiyonun görünümü

### 3.2.2 Sphere Fonksiyonu

$$f_2(x) = \sum_{i=1}^d x_i^2 \quad (3.2)$$

$x_i \in [-5.12, 5.12]$  aralığında,  $i = 1, \dots, d$

Global minimum  $x^* = (0, \dots, 0)$ ,  $f(x^*) = 0$  [37]



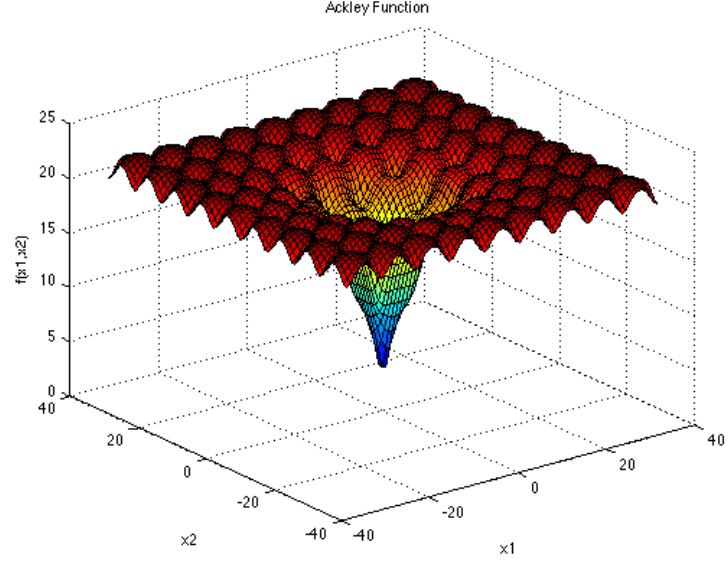
Şekil 3.3: Sphere fonksiyonunun görünümü

### 3.2.3 Ackley Fonksiyonu

$$f_3(x) = -20 \cdot \exp\left(-0,2 \cdot \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos 2\pi \cdot x_i\right) + 20 + \exp(1) \quad (3.3)$$

$x_i \in [-32.768, 32.768]$  aralığında,  $i = 1, \dots, d$

Global minimum  $x^* = (0, \dots, 0)$ ,  $f(x^*) = 0$  [38]



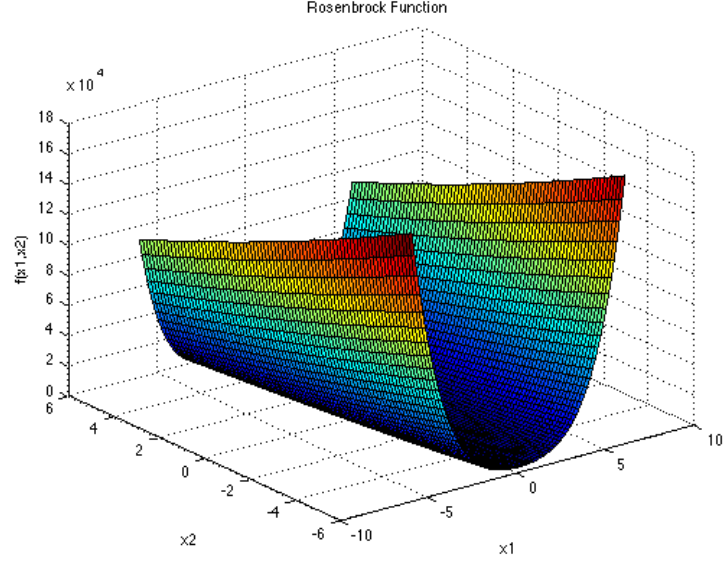
**Şekil 3.4:** Ackley fonksiyonunun görünümü

### 3.2.4 Rosenbrock Fonksiyonu

$$f_4(x) = \sum_{i=1}^{d-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad (3.4)$$

$x_i \in [-5, 10]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (1, \dots, 1), f(x^*) = 0$  [39]



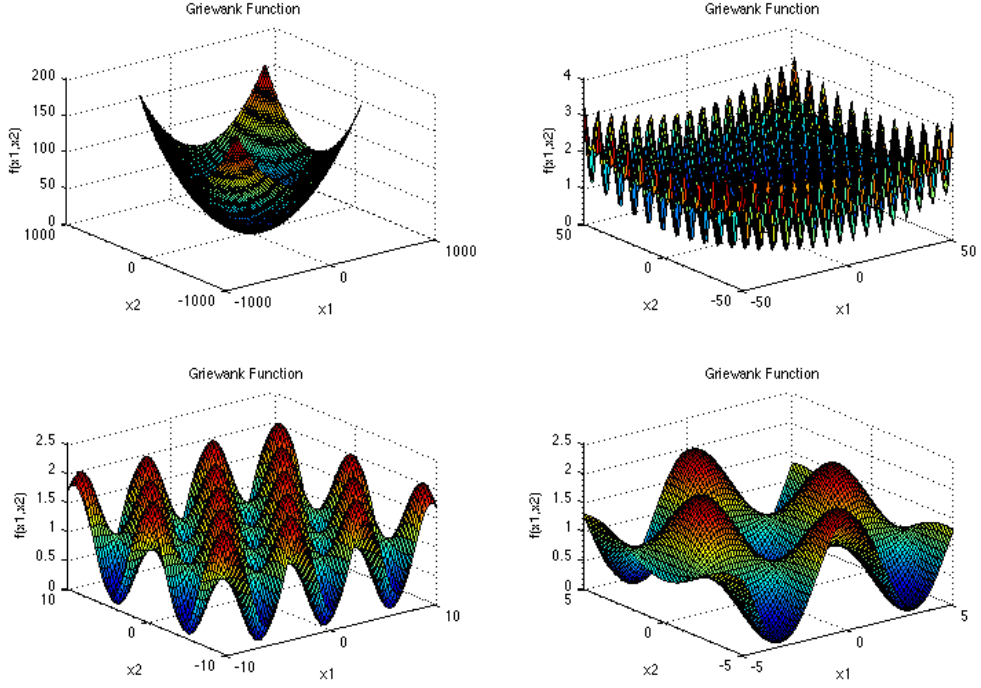
**Şekil 3.5:** Rosenbrock fonksiyonunun görünümü

### 3.2.5 Griewank Fonksiyonu

$$f_5(x) = 1 + \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (3.5)$$

$x_i \in [-600, 600]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (0, \dots, 0), f(x^*) = 0$  [40]



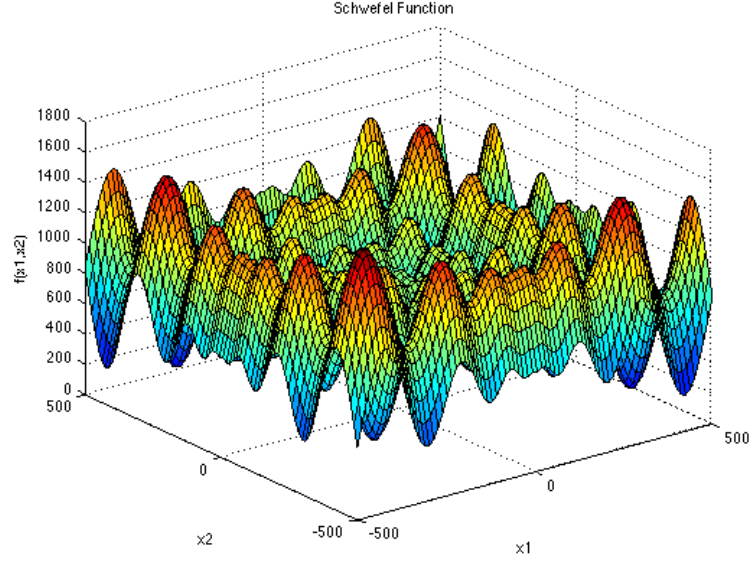
Şekil 3.6: Griewank fonksiyonunun görünümleri

### 3.2.6 Schwefel Fonksiyonu

$$f_6(x) = 418.9829 \cdot d - \sum_{i=1}^d x_i \cdot \sin(\sqrt{|x_i|}) \quad (3.6)$$

$x_i \in [-500, 500]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (420,9687, \dots, 420,9687), f(x^*) = 0$  [41]



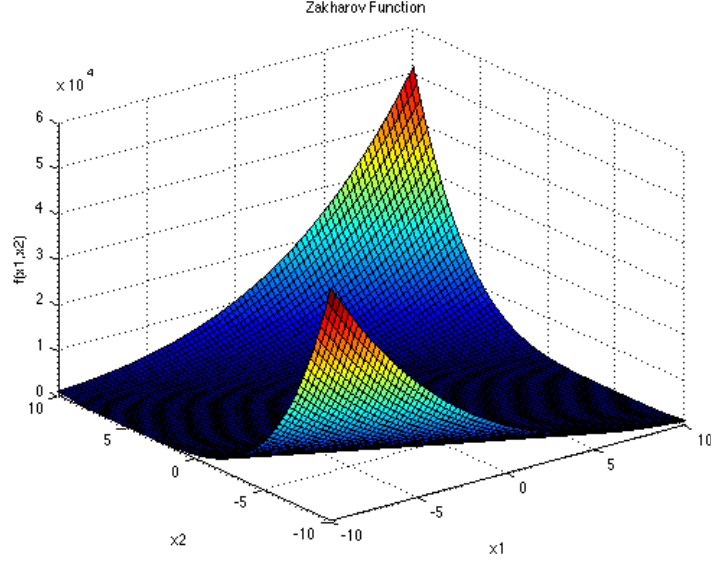
Şekil 3.7: Schwefel fonksiyonunun görünümü

### 3.2.7 Zakharov Fonksiyonu

$$f_7(x) = \sum_{i=1}^d x_i^2 + \left( \sum_{i=1}^d 0.5 \cdot i \cdot x_i \right)^2 + \left( \sum_{i=1}^d 0.5 \cdot i \cdot x_i \right)^4 \quad (3.7)$$

$x_i \in [-5, 10]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (0, \dots, 0), f(x^*) = 0$  [42]



Şekil 3.8: Zakharov fonksiyonunun görünümü

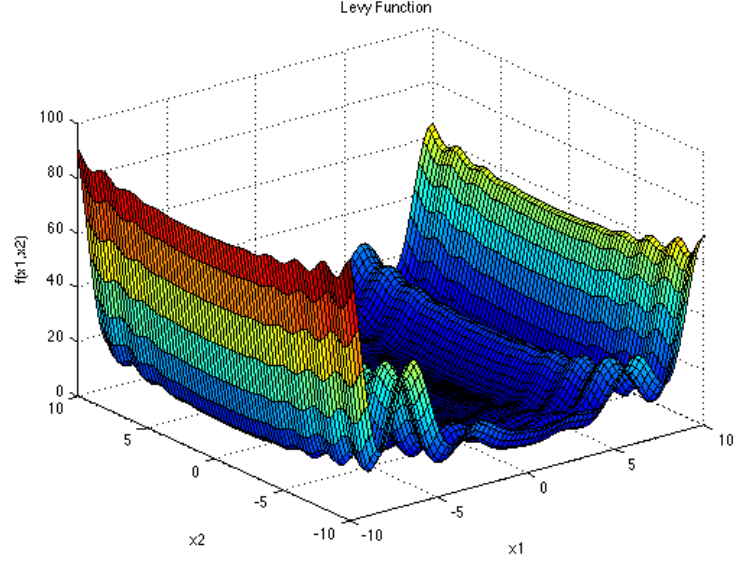
### 3.2.8 Levy Fonksiyonu

$$f_8(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)] \quad (3.8)$$

$$w_i = 1 + \frac{x_i - 1}{4} \text{ ve } x = (x_1, x_2, \dots, x_d)$$

$$x_i \in [-10, 10] \text{ aralığında, } i = 1, \dots, d$$

$$\text{Global minimum } x^* = (1, \dots, 1), f(x^*) = 0 \quad [43]$$



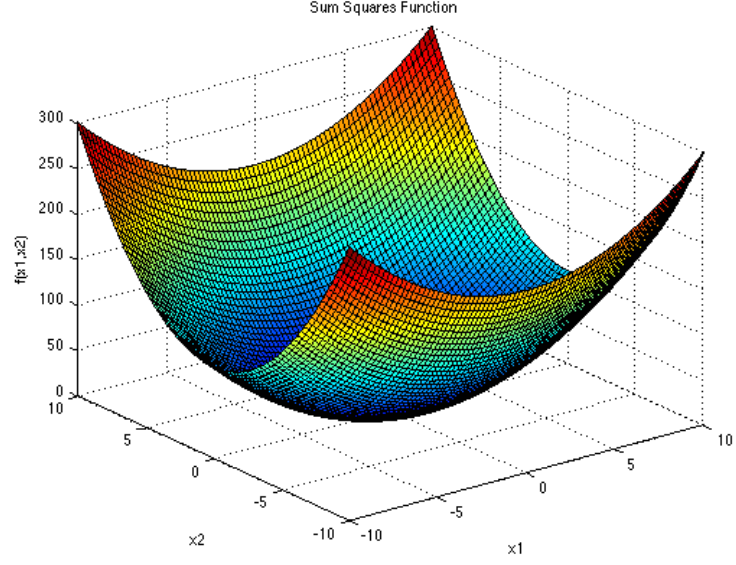
Şekil 3.9: Levy fonksiyonunun görünümü

### 3.2.9 Sum Squares Fonksiyonu

$$f_9(x) = \sum_{i=1}^d i \cdot x_i^2 \quad (3.9)$$

$x_i \in [-5.12, 5.12]$  aralığında,  $i = 1, \dots, d$

Global minimum  $x^* = (0, \dots, 0), f(x^*) = 0$  [44]



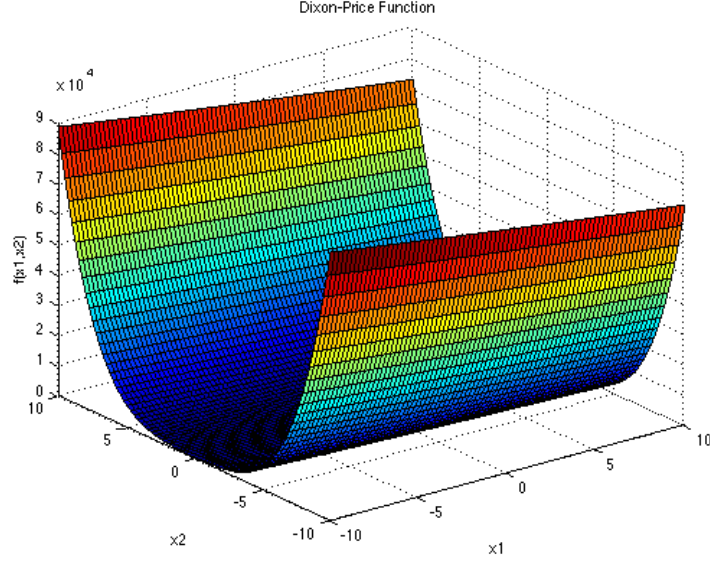
Şekil 3.10: Sum Squares fonksiyonunun görünümü

### 3.2.1 Dixon Price Fonksiyonu

$$f_{10}(x) = (x_1 - 1)^2 + \sum_{i=2}^d i \cdot (2x_i^2 - x_{i-1})^2 \quad (3.10)$$

$x_i \in [-10, 10]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $i \geq 2$  için  $x_1 = 1$ ,  $x_i = 2^{-(2^{i-2})}$ ,  $f(x^*) = 0$  [45]



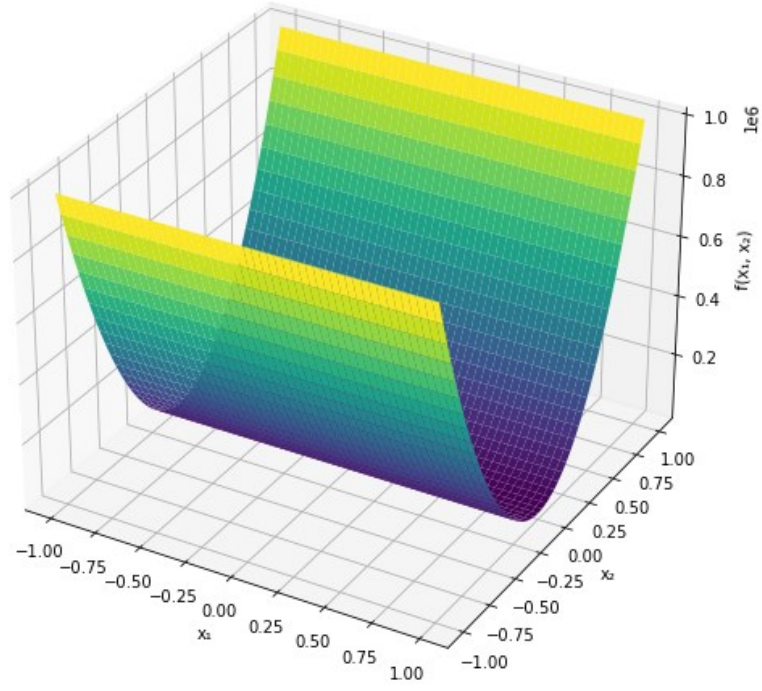
**Şekil 3.11:** Dixon Price fonksiyonunun görünümü

### 3.2.2 Bent Cigar Fonksiyonu

$$f_{11}(x) = x_1^2 + 10^6 \sum_{i=2}^d x_i^2 \quad (3.11)$$

$x_i \in [-100, 100]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (0, \dots, 0), f(x^*) = 0$  [46]



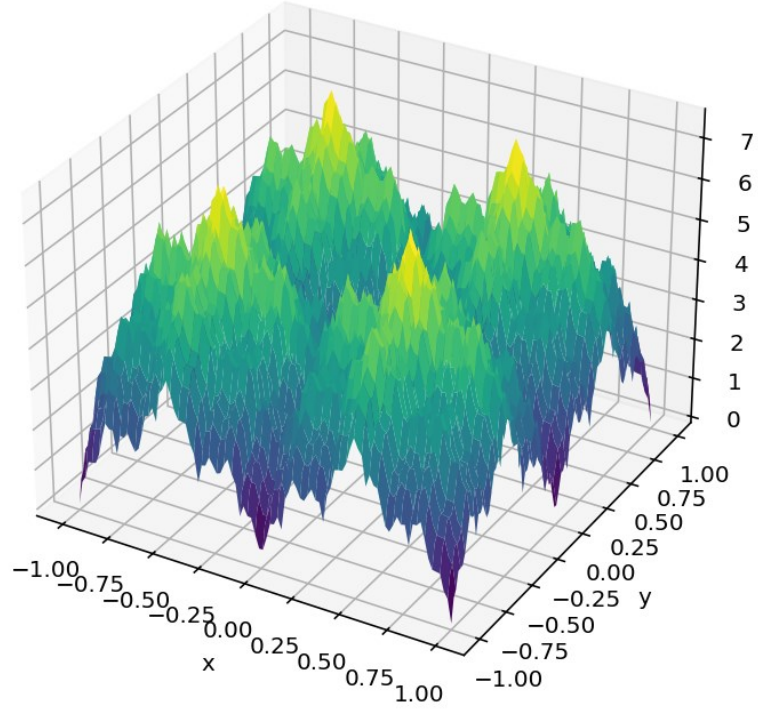
**Şekil 3.12:** Bent Cigar fonksiyonunun görünümü

### 3.2.3 Weierstrass Fonksiyonu

$$f_{12}(x) = \sum_{i=1}^d \sum_{n=0}^k a^n \cos(2\pi b^n (x_i + 0.5)) - d \cdot \sum_{n=0}^k a^n \cos(2\pi b^n \cdot 0.5) \quad (3.12)$$

$x_i \in [-5, 5]$  aralığında,  $i = 1, \dots, d$

*Global minimum*  $x^* = (0, \dots, 0)$ ,  $f(x^*) = 0$  [47]



**Şekil 3.13:** Weierstrass fonksiyonunun görünümü

### 3.3 Deneysel Çalışmalar

**Tablo 3.3:** Deney 1 sonuçları.

<i>Fonksiyon</i>	<i>İndeks</i>	<i>FA20</i>	<i>FA70</i>	<i>IFA1</i>	<i>IFA2</i>	<i>IFA3</i>	<i>IFA4</i>
<i>Rastrigin</i>	<i>En Küçük</i>	4,477E+01	30,844	42,783	35,819	90,828	22,412
	<i>Ortalama</i>	7,848E+01	56,374	87,934	57,111	207,734	58,694
<i>Sphere</i>	<i>En Küçük</i>	1,078E-10	6,115E-11	4,585E-11	4,283E-11	4,656E-11	5,203E-11
	<i>Ortalama</i>	1,410E-10	7,484E-11	5,768E-11	6,004E-11	6,018E-11	6,207E-11
<i>Ackley</i>	<i>En Küçük</i>	2,027E+01	2,009E+01	1,811E+01	2,010E+01	1,909E+01	1,069E+01
	<i>Ortalama</i>	2,055E+01	2,035E+01	1,886E+01	2,044E+01	1,975E+01	1,372E+01
<i>Rosenbrock</i>	<i>En Küçük</i>	1,449E+06	2,730E+04	3,553E+01	1,017E+06	1,854E+02	7,020E+00
	<i>Ortalama</i>	2,209E+06	1,334E+06	7,451E+01	1,912E+06	2,990E+04	2,869E+02
<i>Griewank</i>	<i>En Küçük</i>	8,483E+02	7,079E+02	5,135E+02	7,662E+02	7,596E+02	5,514E+01
	<i>Ortalama</i>	9,614E+02	8,425E+02	6,754E+02	9,032E+02	9,172E+02	8,447E+01
<i>Schewefel</i>	<i>En Küçük</i>	1,422E+04	1,397E+04	1,025E+04	1,393E+04	1,333E+04	1,436E+04
	<i>Ortalama</i>	1,560E+04	1,490E+04	1,152E+04	1,530E+04	1,515E+04	1,529E+04
<i>Zakharov</i>	<i>En Küçük</i>	7,591E+02	5,223E+02	1,090E+02	5,521E+02	1,267E+02	1,101E+02
	<i>Ortalama</i>	1,786E+06	8,863E+02	2,275E+02	1,287E+05	3,763E+02	3,795E+02
<i>Levy</i>	<i>En Küçük</i>	2,520E+02	1,933E+02	3,903E+01	2,203E+02	5,042E+01	1,737E+01
	<i>Ortalama</i>	3,081E+02	2,521E+02	6,383E+01	2,733E+02	7,237E+01	3,478E+01
<i>Sum Squares</i>	<i>En Küçük</i>	1,163E-06	2,159E-09	1,422E-09	1,310E-09	1,322E-09	1,373E-09
	<i>Ortalama</i>	9,710E-03	1,205E-08	7,462E-09	5,507E-09	5,794E-09	5,433E-09
<i>Dixon Price</i>	<i>En Küçük</i>	2,897E+06	1,952E+06	5,000E-01	2,660E+06	7,369E-01	5,000E-01
	<i>Ortalama</i>	4,084E+06	3,119E+06	6,645E-01	3,750E+06	8,230E+04	1,535E+01
<i>Bent Cigar</i>	<i>En Küçük</i>	8,353E+10	7,971E+10	2,208E+10	7,418E+10	8,342E+10	6,116E+09
	<i>Ortalama</i>	1,046E+11	9,066E+10	3,636E+10	9,695E+10	9,678E+10	1,089E+10
<i>Weierstrass</i>	<i>En Küçük</i>	9,196E+00	6,885E-01	3,259E+00	2,069E+00	1,358E-02	9,879E-01
	<i>Ortalama</i>	1,611E+01	5,758E+00	9,789E+00	6,928E+00	5,493E+00	7,007E+00

**Tablo 3.4:** Deney 2 sonuçları.

<i>Fonksiyon</i>	<i>İndeks</i>	<i>FA20</i>	<i>FA45</i>	<i>IFA1</i>	<i>IFA2</i>	<i>IFA3</i>	<i>IFA4</i>
<i>Rastrigin</i>	<i>En Küçük</i>	4,477E+01	26,864	46,763	25,869	61,831	38,804
	<i>Ortalama</i>	7,848E+01	59,160	86,024	68,692	139,501	65,038
<i>Sphere</i>	<i>En Küçük</i>	1,078E-10	6,752E-11	4,457E-11	4,630E-11	5,082E-11	5,250E-11
	<i>Ortalama</i>	1,410E-10	8,980E-11	5,792E-11	6,140E-11	6,172E-11	6,333E-11
<i>Ackley</i>	<i>En Küçük</i>	2,027E+01	2,013E+01	1,840E+01	2,008E+01	1,909E+01	1,282E+01
	<i>Ortalama</i>	2,055E+01	2,041E+01	1,901E+01	2,047E+01	2,000E+01	1,514E+01
<i>Rosenbrock</i>	<i>En Küçük</i>	1,449E+06	1,178E+06	3,162E+01	1,579E+05	5,624E+02	8,008E+01
	<i>Ortalama</i>	2,209E+06	1,767E+06	1,907E+02	2,034E+06	2,153E+05	5,947E+03
<i>Griewank</i>	<i>En Küçük</i>	8,483E+02	7,425E+02	6,558E+02	7,570E+02	6,751E+02	6,979E+01
	<i>Ortalama</i>	9,614E+02	8,870E+02	8,445E+02	9,213E+02	9,136E+02	1,374E+02
<i>Schewefel</i>	<i>En Küçük</i>	1,422E+04	1,411E+04	1,250E+04	1,439E+04	1,339E+04	1,466E+04
	<i>Ortalama</i>	1,560E+04	1,509E+04	1,401E+04	1,542E+04	1,526E+04	1,550E+04
<i>Zakharov</i>	<i>En Küçük</i>	7,591E+02	7,677E+02	2,558E+02	5,698E+02	3,729E+02	3,498E+02
	<i>Ortalama</i>	1,786E+06	1,242E+03	5,024E+02	1,730E+05	7,212E+02	6,822E+02
<i>Levy</i>	<i>En Küçük</i>	2,520E+02	2,101E+02	4,395E+01	2,428E+02	4,570E+01	1,227E+01
	<i>Ortalama</i>	3,081E+02	2,682E+02	6,908E+01	2,905E+02	7,807E+01	3,716E+01
<i>Sum Squares</i>	<i>En Küçük</i>	1,163E-06	2,658E-09	2,058E-09	3,230E-09	2,191E-09	4,387E-09
	<i>Ortalama</i>	9,710E-03	2,156E-07	2,183E-08	2,048E-08	2,103E-08	2,120E-08
<i>Dixon Price</i>	<i>En Küçük</i>	2,897E+06	2,736E+06	5,042E-01	2,823E+06	3,521E+04	1,961E+00
	<i>Ortalama</i>	4,084E+06	3,519E+06	3,206E+03	3,926E+06	6,432E+05	2,506E+03
<i>Bent Cigar</i>	<i>En Küçük</i>	8,353E+10	7,909E+10	4,875E+10	8,173E+10	8,039E+10	7,802E+09
	<i>Ortalama</i>	1,046E+11	9,548E+10	6,963E+10	9,934E+10	9,942E+10	1,691E+10
<i>Weierstrass</i>	<i>En Küçük</i>	9,196E+00	1,927E+00	4,954E+00	3,586E+00	1,401E-02	2,597E+00
	<i>Ortalama</i>	1,611E+01	8,004E+00	1,101E+01	9,114E+00	8,117E+00	1,024E+01

**Tablo 3.5:** Deney 3 sonuçları.

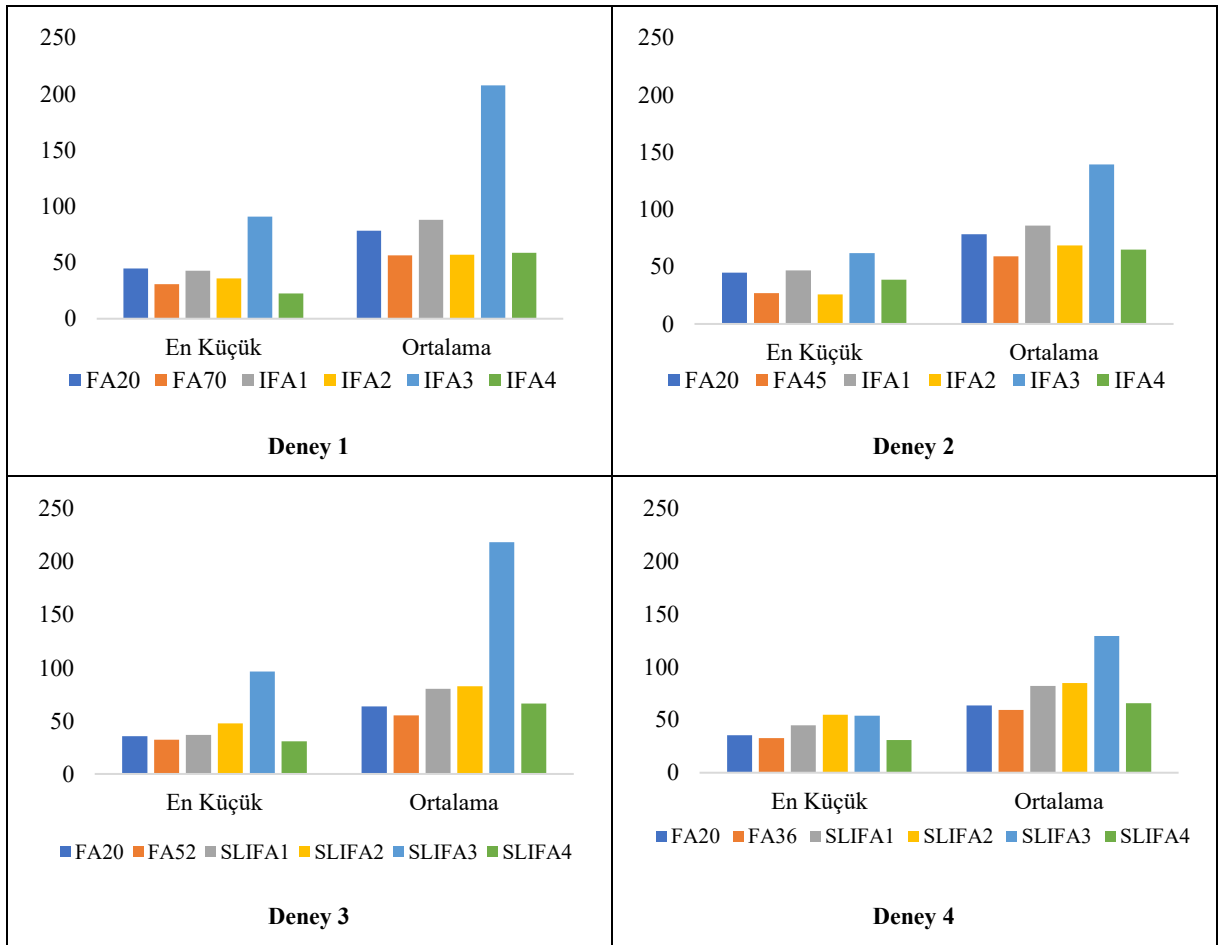
<i>Fonksiyon</i>	<i>İndeks</i>	<i>FA20</i>	<i>FA52</i>	<i>SLIFA1</i>	<i>SLIFA2</i>	<i>SLIFA3</i>	<i>SLIFA4</i>
<i>Rastrigin</i>	<i>En Küçük</i>	3,547E+01	32,299	36,813	47,758	96,513	30,849
	<i>Ortalama</i>	6,355E+01	55,187	80,114	82,661	218,352	66,337
<i>Sphere</i>	<i>En Küçük</i>	1,086E-10	6,551E-11	5,175E-11	5,594E-11	5,325E-11	4,703E-11
	<i>Ortalama</i>	1,420E-10	8,095E-11	6,195E-11	6,677E-11	6,581E-11	6,496E-11
<i>Ackley</i>	<i>En Küçük</i>	2,012E+01	2,012E+01	1,857E+01	2,028E+01	1,915E+01	1,113E+01
	<i>Ortalama</i>	2,054E+01	2,035E+01	1,912E+01	2,051E+01	1,976E+01	1,374E+01
<i>Rosenbrock</i>	<i>En Küçük</i>	1,189E+06	5,509E+05	4,542E+01	6,298E+05	6,797E+04	6,382E+01
	<i>Ortalama</i>	2,177E+06	1,640E+06	6,524E+02	2,215E+06	5,989E+05	8,698E+03
<i>Griewank</i>	<i>En Küçük</i>	8,224E+02	7,477E+02	6,027E+02	8,064E+02	8,427E+02	5,194E+01
	<i>Ortalama</i>	9,687E+02	8,495E+02	8,123E+02	9,514E+02	9,645E+02	9,514E+01
<i>Schewefel</i>	<i>En Küçük</i>	1,463E+04	1,387E+04	9,958E+03	1,491E+04	1,466E+04	1,488E+04
	<i>Ortalama</i>	1,568E+04	1,497E+04	1,191E+04	1,558E+04	1,556E+04	1,563E+04
<i>Zakharov</i>	<i>En Küçük</i>	5,988E+02	6,002E+02	3,133E+02	8,323E+02	3,681E+02	1,763E+02
	<i>Ortalama</i>	3,709E+05	9,504E+02	5,037E+02	1,426E+06	8,995E+02	2,028E+05
<i>Levy</i>	<i>En Küçük</i>	2,391E+02	2,043E+02	5,875E+01	2,677E+02	4,766E+01	1,397E+01
	<i>Ortalama</i>	2,969E+02	2,546E+02	7,302E+01	3,094E+02	7,628E+01	3,471E+01
<i>Sum Squares</i>	<i>En Küçük</i>	1,556E-07	2,063E-09	7,189E-09	1,717E-09	2,111E-09	1,367E-09
	<i>Ortalama</i>	1,197E-02	2,259E-08	1,044E-08	9,026E-09	2,753E+00	9,134E-09
<i>Dixon Price</i>	<i>En Küçük</i>	2,908E+06	2,382E+06	2,333E+01	2,268E+06	1,888E+03	5,003E-01
	<i>Ortalama</i>	4,098E+06	3,302E+06	1,413E+03	4,112E+06	1,124E+05	1,888E+02
<i>Bent Cigar</i>	<i>En Küçük</i>	8,490E+10	7,259E+10	3,877E+10	8,366E+10	8,379E+10	4,152E+09
	<i>Ortalama</i>	1,035E+11	9,086E+10	4,929E+10	1,015E+11	1,012E+11	9,097E+09
<i>Weierstrass</i>	<i>En Küçük</i>	9,207E+00	8,970E-01	4,940E+00	4,476E+00	1,203E-02	2,342E+00
	<i>Ortalama</i>	1,487E+01	7,190E+00	1,268E+01	1,026E+01	6,913E+00	1,064E+01

**Tablo 3.6:** Deney 4 sonuçları.

<i>Fonksiyon</i>	<i>İndeks</i>	<i>FA20</i>	<i>FA36</i>	<i>SLIFA1</i>	<i>SLIFA2</i>	<i>SLIFA3</i>	<i>SLIFA4</i>
<i>Rastrigin</i>	<i>En Küçük</i>	3,547E+01	32,839	44,773	54,723	53,754	30,859
	<i>Ortalama</i>	6,355E+01	59,168	81,965	84,671	129,027	65,654
<i>Sphere</i>	<i>En Küçük</i>	1,086E-10	4,867E-11	5,976E-11	6,219E-11	6,575E-11	5,890E-11
	<i>Ortalama</i>	1,420E-10	9,669E-11	7,738E-11	8,012E-11	7,993E-11	7,974E-11
<i>Ackley</i>	<i>En Küçük</i>	2,012E+01	2,007E+01	1,843E+01	2,028E+01	1,889E+01	2,025E+01
	<i>Ortalama</i>	2,054E+01	2,043E+01	1,915E+01	2,053E+01	2,001E+01	2,053E+01
<i>Rosenbrock</i>	<i>En Küçük</i>	1,189E+06	1,429E+06	2,801E+04	1,272E+06	4,758E+05	6,020E+03
	<i>Ortalama</i>	2,177E+06	1,890E+06	8,854E+04	2,191E+06	1,292E+06	4,453E+04
<i>Griewank</i>	<i>En Küçük</i>	8,224E+02	7,560E+02	7,261E+02	7,879E+02	8,101E+02	6,859E+01
	<i>Ortalama</i>	9,687E+02	9,073E+02	9,088E+02	9,571E+02	9,688E+02	1,354E+02
<i>Schewefel</i>	<i>En Küçük</i>	1,463E+04	1,423E+04	1,229E+04	1,428E+04	1,435E+04	1,476E+04
	<i>Ortalama</i>	1,568E+04	1,519E+04	1,407E+04	1,569E+04	1,553E+04	1,573E+04
<i>Zakharov</i>	<i>En Küçük</i>	5,988E+02	5,926E+02	5,236E+02	6,805E+02	5,388E+02	3,180E+02
	<i>Ortalama</i>	3,709E+05	1,330E+03	7,273E+02	7,784E+04	1,115E+03	7,621E+05
<i>Levy</i>	<i>En Küçük</i>	2,391E+02	2,071E+02	5,329E+01	2,329E+02	6,044E+01	1,459E+01
	<i>Ortalama</i>	2,969E+02	2,754E+02	7,657E+01	3,014E+02	8,346E+01	4,888E+01
<i>Sum Squares</i>	<i>En Küçük</i>	1,556E-07	3,050E-09	2,420E-09	1,991E-09	4,390E-09	2,095E-08
	<i>Ortalama</i>	1,197E-02	2,026E-06	2,140E-05	5,403E-06	2,669E-04	4,431E-06
<i>Dixon Price</i>	<i>En Küçük</i>	2,908E+06	2,623E+06	1,091E+05	3,071E+06	3,056E+05	4,989E+02
	<i>Ortalama</i>	4,098E+06	3,508E+06	3,119E+05	4,218E+06	1,113E+06	7,171E+03
<i>Bent Cigar</i>	<i>En Küçük</i>	8,490E+10	7,971E+10	6,306E+10	9,312E+10	8,198E+10	9,754E+09
	<i>Ortalama</i>	1,035E+11	9,570E+10	7,859E+10	1,046E+11	1,044E+11	1,586E+10
<i>Weierstrass</i>	<i>En Küçük</i>	9,207E+00	5,231E+00	6,661E+00	4,749E+00	1,775E-02	4,908E+00
	<i>Ortalama</i>	1,487E+01	1,036E+01	1,428E+01	1,173E+01	1,257E+01	1,259E+01

### 3.4 Deney Çalışmalarının Grafikselle Analizi

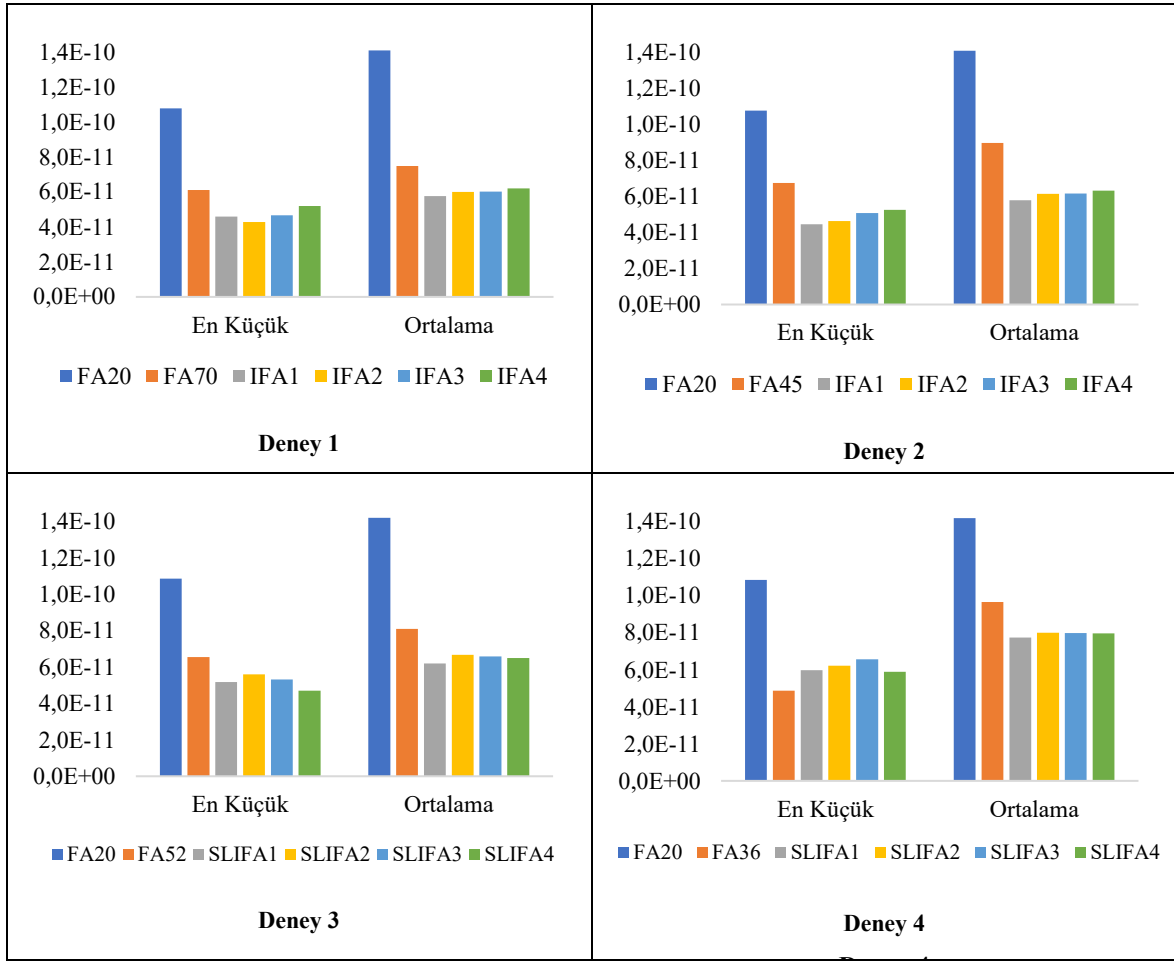
Çok modlu yapıya sahip Rastrigin fonksiyonu ile gerçekleştirilen deneylerde hem en küçük değerler hem de ortalama değerlerin performansı göz önünde bulundurulduğunda, IFA4 ve SLIFA4 durumlarının algoritmada başarılı sonuçlar ürettiği görülmektedir. Ancak, Rastrigin fonksiyonunun her boyutta periyodik olarak değişen kosinüs terimi nedeniyle çok sayıda yerel minimuma sahip olması [36], sabit popülasyonlu algoritmalar olan FA70, FA45, FA52 ve FA36'in genel başarı düzeyini artırmakta ve bu algoritmaların daha istikrarlı performanslar sergilemesine olanak tanımaktadır.



Şekil 3.14: Rastrigin fonksiyonunun Deney 1,2,3,4 değerleri.

Unimodal yapıya sahip Sphere fonksiyonu üzerinde elde edilen deneysel sonuçlar grafiksel olarak Şekil 3.15'te sunulmuştur. Ortalama değerler açısından en başarılı sonuçların IFA1 ve SLIFA1 konfigürasyonlarında elde edildiği gözlemlenmektedir. En küçük değerlere bakıldığında ise, IFA1 ve SLIFA1 durumları en iyi performansı sergilememiş olsa da tatmin edici sonuçlar üretmiştir. Sphere fonksiyonu hem SLIFA hem de IFA'da artım

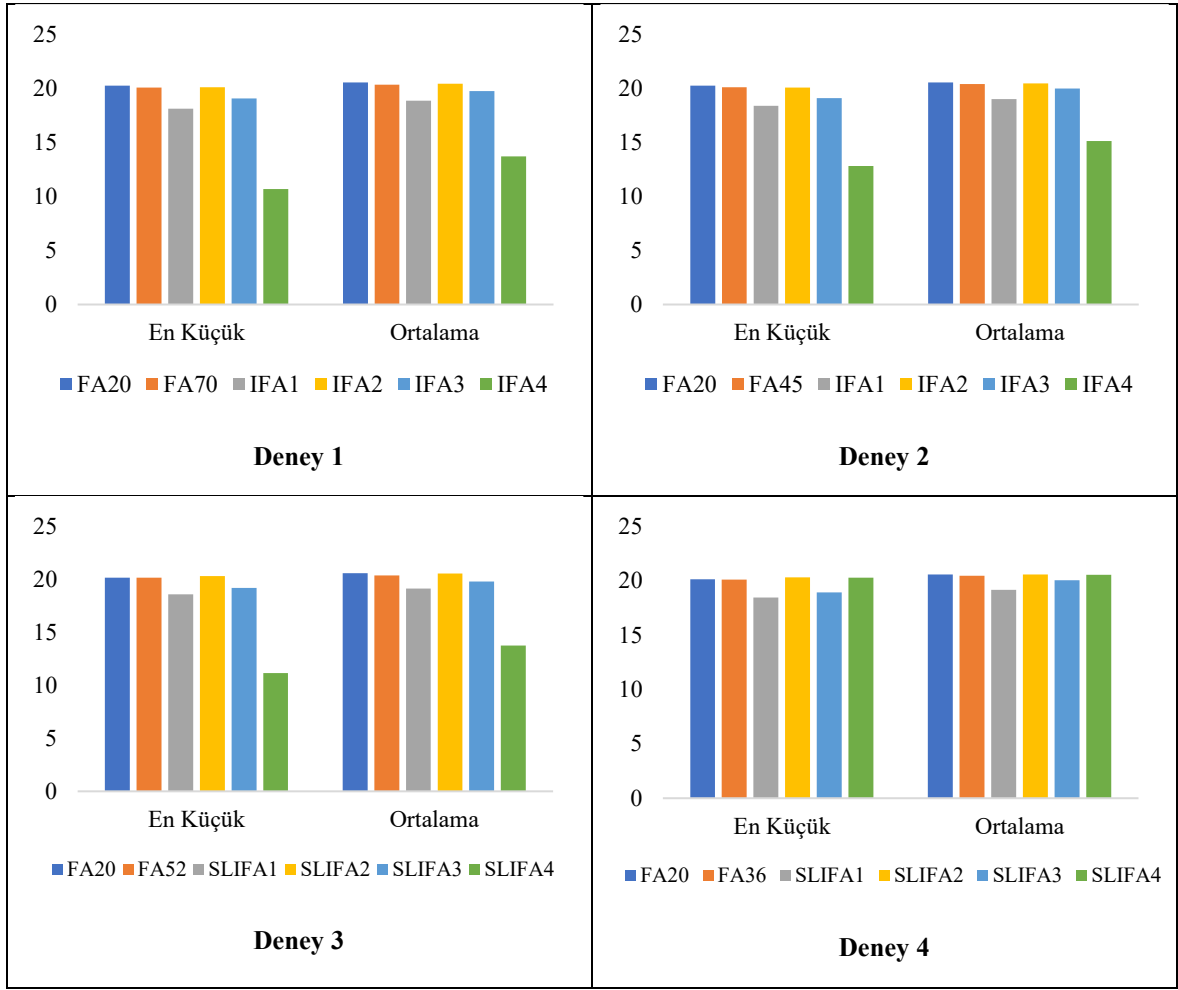
uygulanmayan algoritmalara kıyasla daha iyi sonuçlar vermiştir. Ayrıca IFA'da elde edilen sonuçların SLIFA'daki sonuçlara kıyasla optimuma daha yakın olduğu tespit edilmiştir.



Şekil 3.15: Sphere fonksiyonunun Deney 1,2,3,4 değerleri.

Ackley fonksiyonu üzerinde gerçekleştirilen ve Şekil 3.16'da verilen deney sonuçlarına göre en başarılı küçük ve ortalama değerlerin, Deney 1 ve Deney 2 için IFA4, Deney 3 için SLIFA4 algoritmasıyla elde edildiği gözlemlenmiştir. Özellikle, küçük artım aralıklarının çözüm uzayını daha ayrıntılı tarayabilme potansiyeli sayesinde daha yüksek çözüm doğruluğu sağladığı anlaşılmaktadır. IFA (Deney 1 ve Deney 2) ve SLIFA (Deney 3 ve Deney 4) karşılaştırıldığında ise, daha fazla örnekleme noktasına sahip olan IFA4 yapılandırmalarının, Ackley fonksiyonunun çok sayıda yerel minimum içeren karmaşık doğası karşısında daha başarılı çözümler ürettiği görülmektedir. Bu durum, artımlı arama stratejilerinin ve uygun popülasyon yapılandırmalarının, Ackley gibi çok modlu ve yüksek

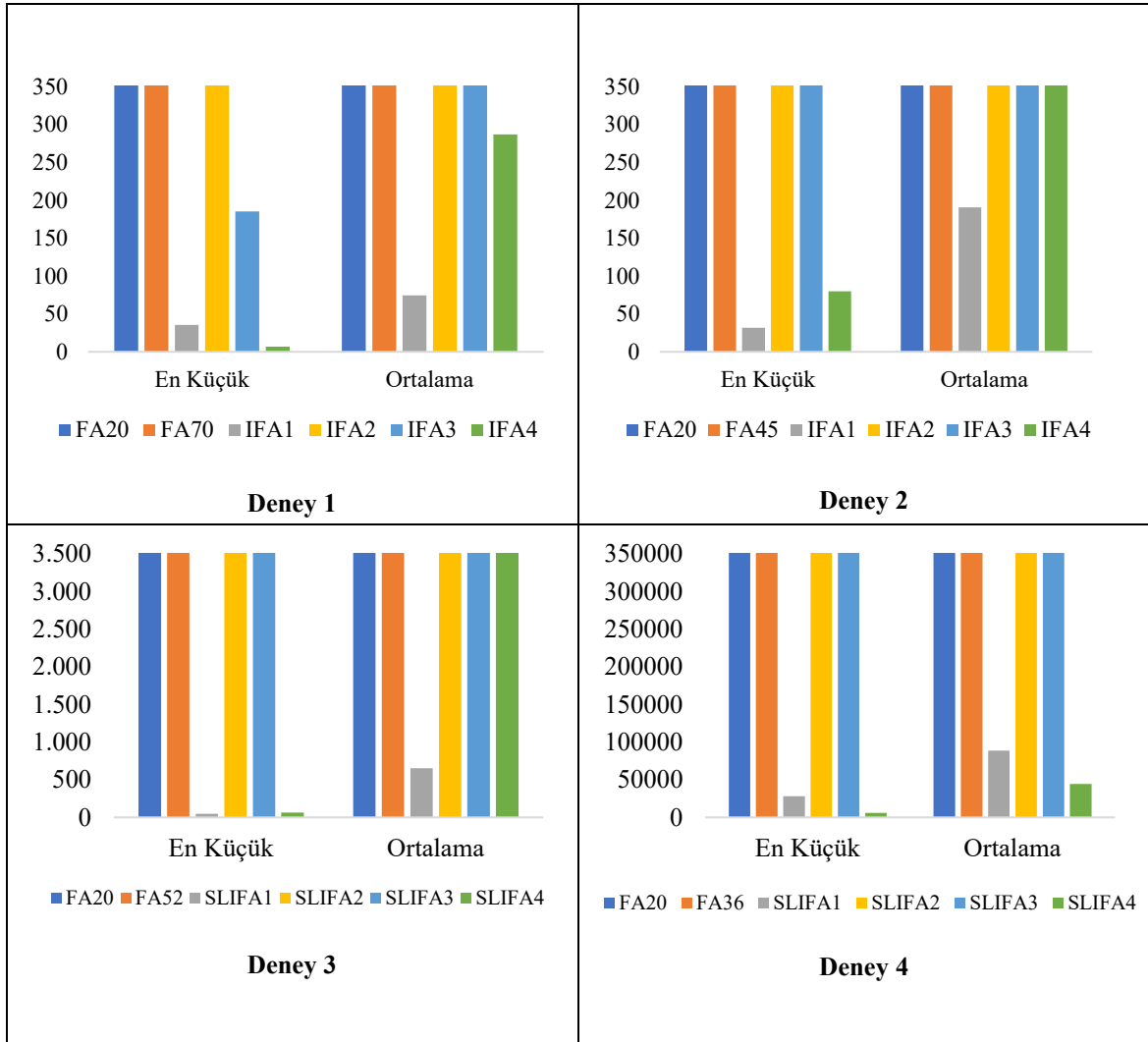
karmaşıklıkla sahip fonksiyonlarda çözüm kalitesini anlamlı ölçüde artırabileceğini ortaya koymaktadır.



Şekil 3.16: Ackley fonksiyonunun Deney 1,2,3,4 değerleri.

Rosenbrock fonksiyonu üzerinde gerçekleştirilen 50 bağımsız çalıştırma sonucunda elde edilen bulgular, Şekil 3.17’de de görülebileceği üzere, IFA1, IFA4, SLIFA1 ve SLIFA4 durumlarının en küçük ve ortalama değerler bakımından diğer senaryolardan anlamlı şekilde ayrıştığını ortaya koymuştur. Fonksiyonun doğası gereği, durumlar arasındaki sonuç farklılıkları oldukça yüksek olduğundan, bazı değerler grafik üzerinde uygun şekilde gösterilememiştir. Söz konusu dört durumun gösterdiği üstün performans, artımlı popülasyon yapısının bu fonksiyon özelinde etkili çözümler üretebildiğini ortaya koymaktadır. Rosenbrock fonksiyonu, karakteristik olarak unimodal yapıda olup dar ve eğimli bir vadi içerisinde küresel minimuma sahiptir. Bu yapı, algoritmanın keşif ve sömürü süreçleri arasında dengeli bir ilişki kurmasına olanak tanımaktadır [48]. Dolayısıyla hem

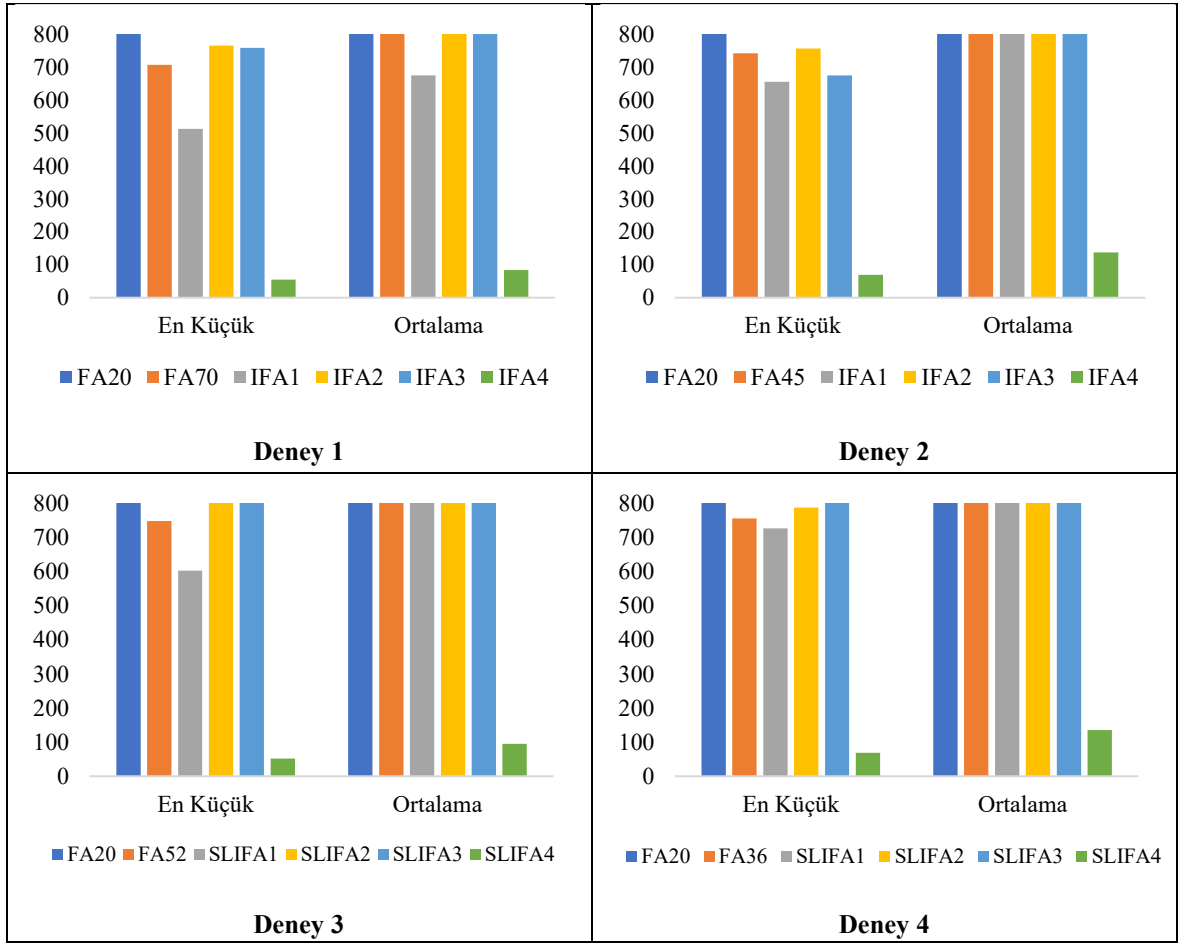
IFA hem de SLIFA algoritmalarının birinci ve dördüncü durumlarında, bu fonksiyon için hızlı ve kararlı çözümler elde edilmiştir.



Şekil 3.17: Rosenbrock fonksiyonunun Deney 1,2,3,4 değerleri.

Şekil 3.18’de görüldüğü üzere Griewank fonksiyonunda artım aralığı 5 ve 10 olduğunda, en küçük ve ortalama değerde IFA4 ve SLIFA4 durumlarının en iyi performansı sergilediği görülmektedir. Griewank fonksiyonu, multimodal bir yapıya sahip olup çözüm uzayında çok sayıda yerel minimum barındırır. Ancak, IFA4 ve SLIFA4 durumları bu zorlu yapıda kararlı bir performans sergilemiştir. Artım aralıkları kıyaslandığında, artım aralığının azalması ve buna paralel olarak popülasyon büyüklüğünün artması pozitif yönde katkı sağlamıştır. Bunun temel nedeni, artım aralığının küçülmesiyle birlikte algoritmanın çözüm uzayını daha detaylı tarayabilmesi ve popülasyon büyüklüğünün artmasıyla global minimuma ulaşma

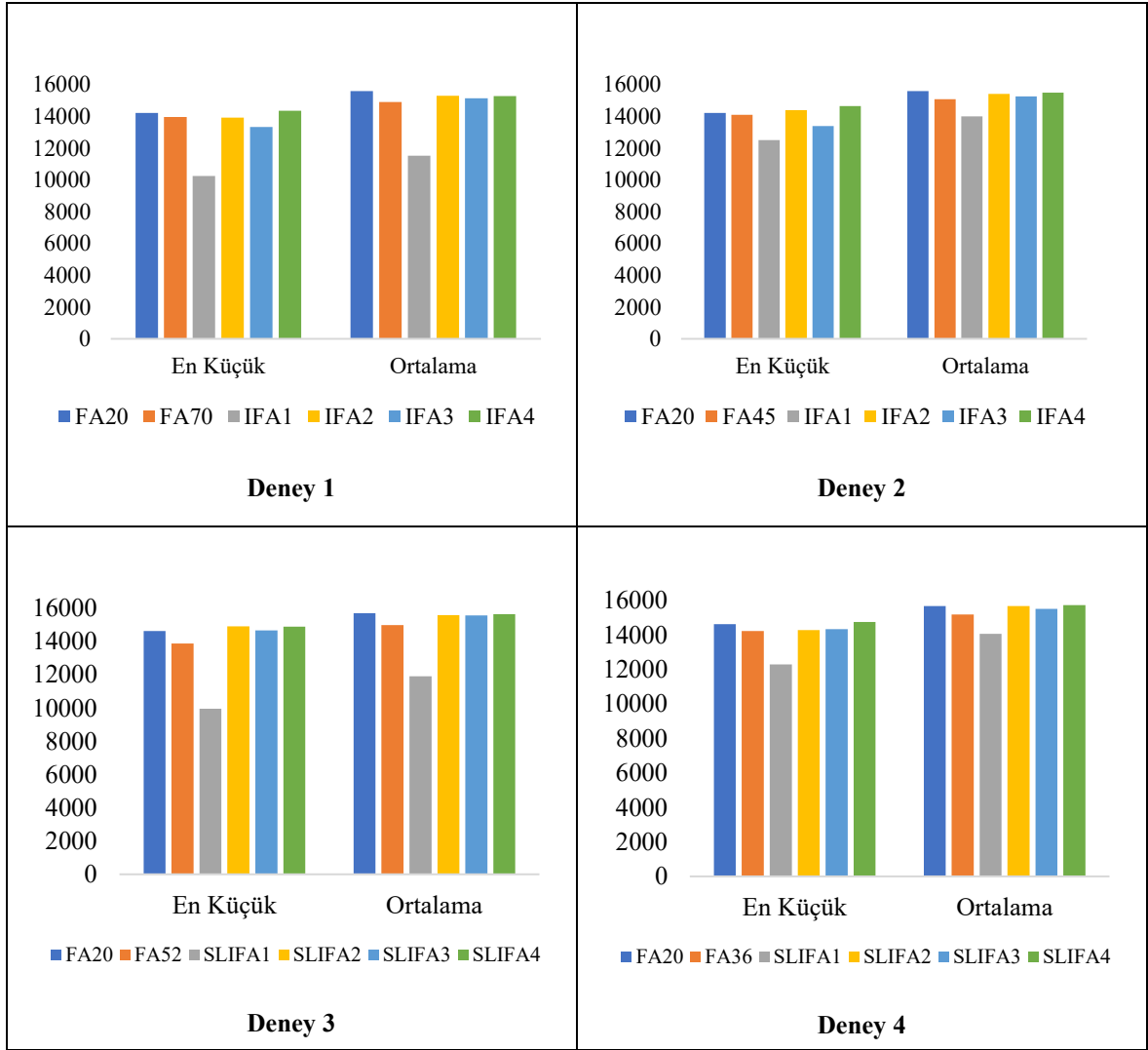
olasılığının yükselmesidir. Grafiklerdeki bazı değerler grafiğe sığdırılmadığı için kırılmıştır.



Şekil 3.18: Griewank fonksiyonunun Deney 1,2,3,4 değerleri.

Multimodal yapıya sahip bir test fonksiyonu olan Schewefel fonksiyonundan elde edilen sonuçlar Şekil 3.19’da verilmektedir. Artım aralıkları 5 ve 10 olarak belirlendiğinde, en düşük ve ortalama değerlerin IFA1 ve SLIFA1 durumlarında elde edildiği görülmektedir. Özellikle bu iki durumda, her artım adımında algoritmaya çözüm uzayının en iyi bireyinin eklenmesi, algoritmanın yerel minimumlara sıkışma olasılığını azaltarak küresel minimuma ulaşma başarımını artırmıştır. IFA ve SLIFA algoritmalarının karşılaştırmasında, her ne kadar en küçük değerler birbirine yakın olsa da, IFA’nın elde ettiği sonuçların optimum çözüme daha yakın olduğu gözlemlenmiştir. Benzer şekilde, artım aralığı açısından yapılan değerlendirmede, 5 iterasyonda bir gerçekleştirilen artımların 10 iterasyonda bir yapılanlara göre daha başarılı sonuçlar verdiği tespit edilmiştir. Bu durumun temel nedeni, daha sık yapılan artımlar sayesinde algoritmanın çözüm uzayında daha fazla noktayı değerlendirme

şansı bulmasıdır. Özellikle çok sayıda yerel minimum içeren Schewefel fonksiyonu gibi fonksiyonlarda, çözüm uzayının geniş bir şekilde taranması, küresel optimumun bulunma olasılığını artırmaktadır [49]. Dolayısıyla hem IFA'nın hem de kısa aralıklarla gerçekleştirilen artımların, fonksiyonun doğasına uygun olarak daha başarılı çözümler ürettiği sonucuna varılabilir.



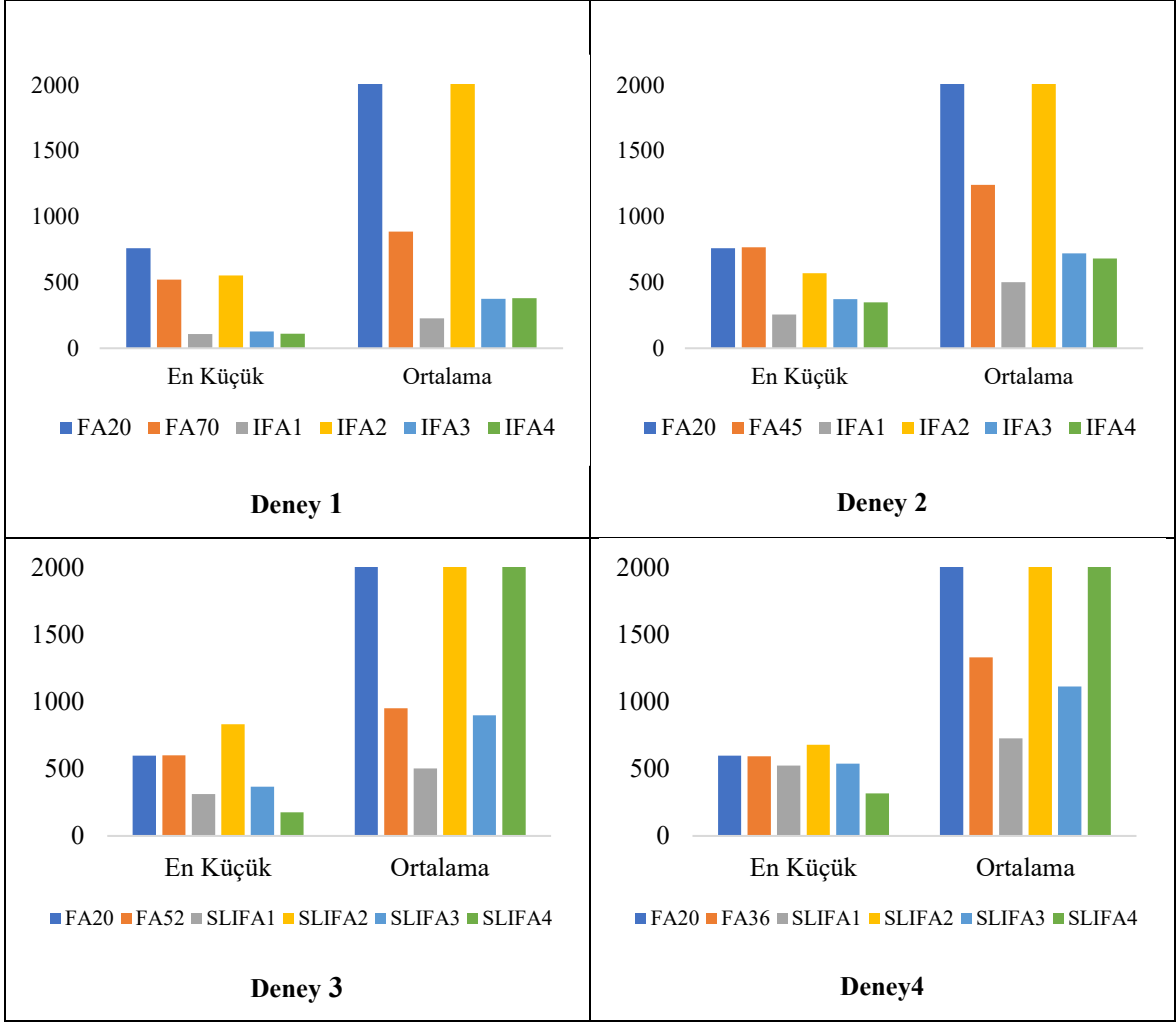
Şekil 3.19: Schewefel fonksiyonunun Deney 1,2,3,4 değerleri.

Zakharov fonksiyonu üzerinde yapılan değerlendirmelerde, elde edilen ortalama değerler dikkate alındığında IFA1 ve SLIFA1 durumları en başarılı sonuçları üretmiştir. Bu durum, artımlı popülasyon stratejisinin sabit popülasyon büyüklüğüne sahip yaklaşımlara kıyasla daha etkili olduğunu ortaya koymaktadır. Özellikle IFA1 ve SLIFA1 durumlarında, popülasyona eklenen yeni bireylerin mevcut en iyi bireyin doğrudan kopyası olması,

algoritmanın başarı oranını Şekil 3.20’de görüleceği üzere önemli ölçüde arttırmıştır. En iyi bireyin popülasyona yeniden kazandırılması, mevcut optimum çözümün korunmasını sağlarken, aynı zamanda diğer bireylerin bu çözüme daha fazla odaklanmasına imkân tanımıştır. Bu durum, çözüm sürecinde sömürü yeteneğini artıran önemli bir etkidir [50]. Bununla birlikte, IFA4 durumunda uygulanan ve yeni bireylerin rastgele bir çözümden mevcut en iyi bireye yönlendirildiği vektör temelli ekleme yöntemi, özellikle en küçük değerler üzerinden yapılan değerlendirmelerde, arama sürecine hem keşif hem de sömürü kapasitesi kazandıran etkili bir strateji olarak öne çıkmıştır. Bu yöntemde yer alan rastgelelik faktörü, bireylerin tek bir noktada toplanmasını engelleyerek, en iyi birey etrafında daha yoğun ancak çeşitli bir arama yapılmasını sağlamaktadır [20]. Ayrıca popülasyon büyüklüğünün arttırılma sıklığı da çözüm kalitesi üzerinde belirleyici bir rol oynamaktadır. 5 iterasyonda bir yapılan arttırımın, 10 iterasyonda bir yapılan arttırmaya göre daha başarılı sonuçlar verdiği gözlemlenmiştir. Daha sık gerçekleştirilen bu arttırmaların, algoritmanın çözüm uzayını daha hızlı ve etkili bir şekilde taramasına olanak tanıdığı, böylece çözüm kalitesini arttırdığı değerlendirilmektedir.

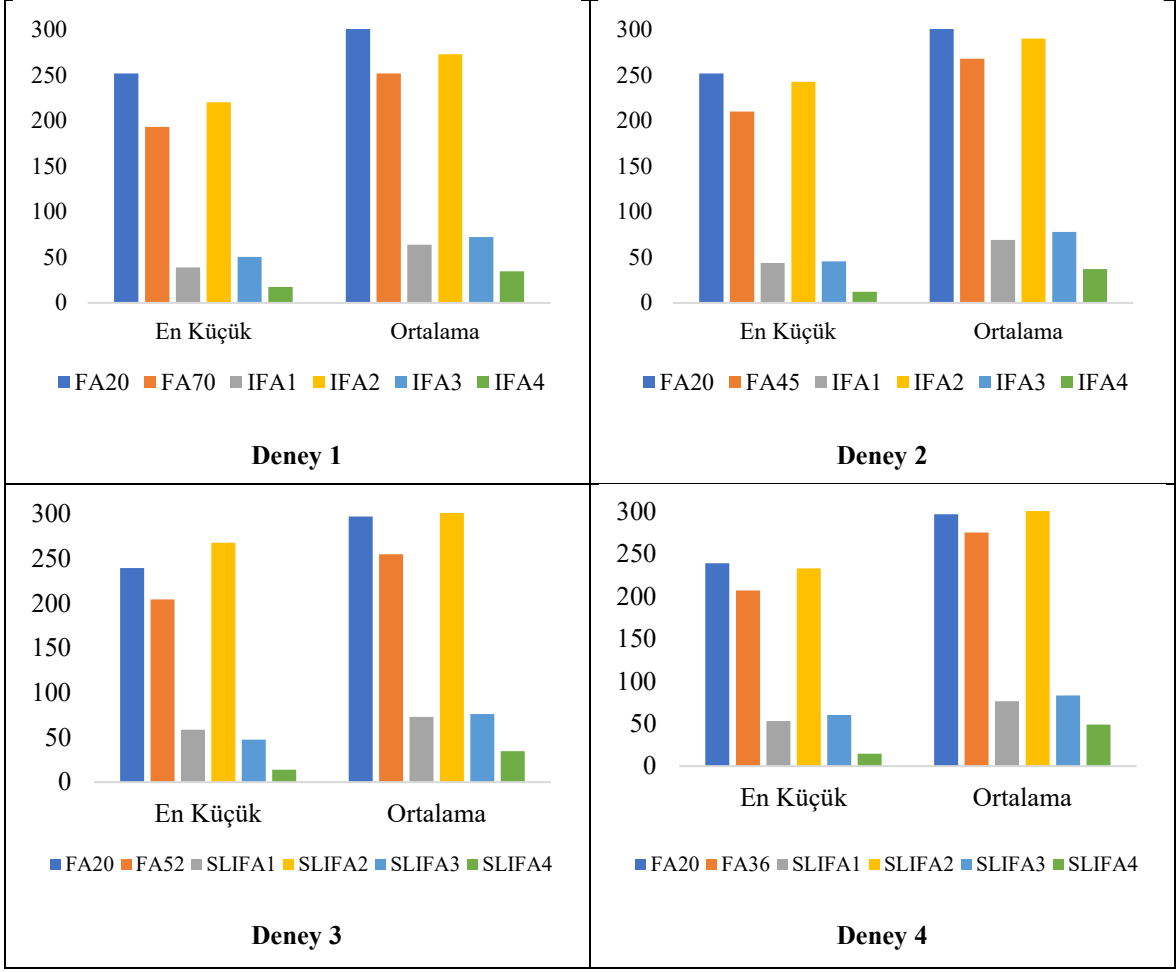
Levy fonksiyonu gibi karmaşık yapılı ve çok sayıda yerel optima içeren optimizasyon problemlerinde, popülasyona yeni bireyler ekleme stratejileri algoritmanın başarısı açısından kritik bir rol oynamaktadır. Şekil 3.21’de sunulan deneysel sonuçlar değerlendirildiğinde, en küçük ve ortalama değerler bakımından en iyi performansı IFA4 ve SLIFA4 yapılandırmalarının gösterdiği görülmektedir. Bu durum, yeni bireylerin en iyi bireye doğru yönlendirilmesinin çözüm uzayında yoğunlaşmayı arttırdığı ve dolayısıyla keşif ile sömürü arasındaki dengenin korunmasına katkı sağladığı şeklinde yorumlanabilir.

Her ne kadar en başarılı sonuçlar elde edilememiş olsa da, IFA1, SLIFA1 ve IFA3, SLIFA3 yapılandırmaları da iyi sonuçlar üretmiştir. En iyi bireyin doğrudan kopyalanmasına dayanan IFA1 ve SLIFA1 stratejileri, mevcut iyi çözümlerin korunmasını sağlayarak sömürü dinamiğini güçlendirmiş; ancak bu durum keşif kapasitesini sınırlamıştır. Öte yandan, IFA3 ve SLIFA3’te arama sürecine rastgelelik katılarak çeşitliliğin arttırılması hedeflenmiş olsa da bu stratejilerin en iyi bireyin etkisini yeterince optimize edemediği gözlemlenmiştir.



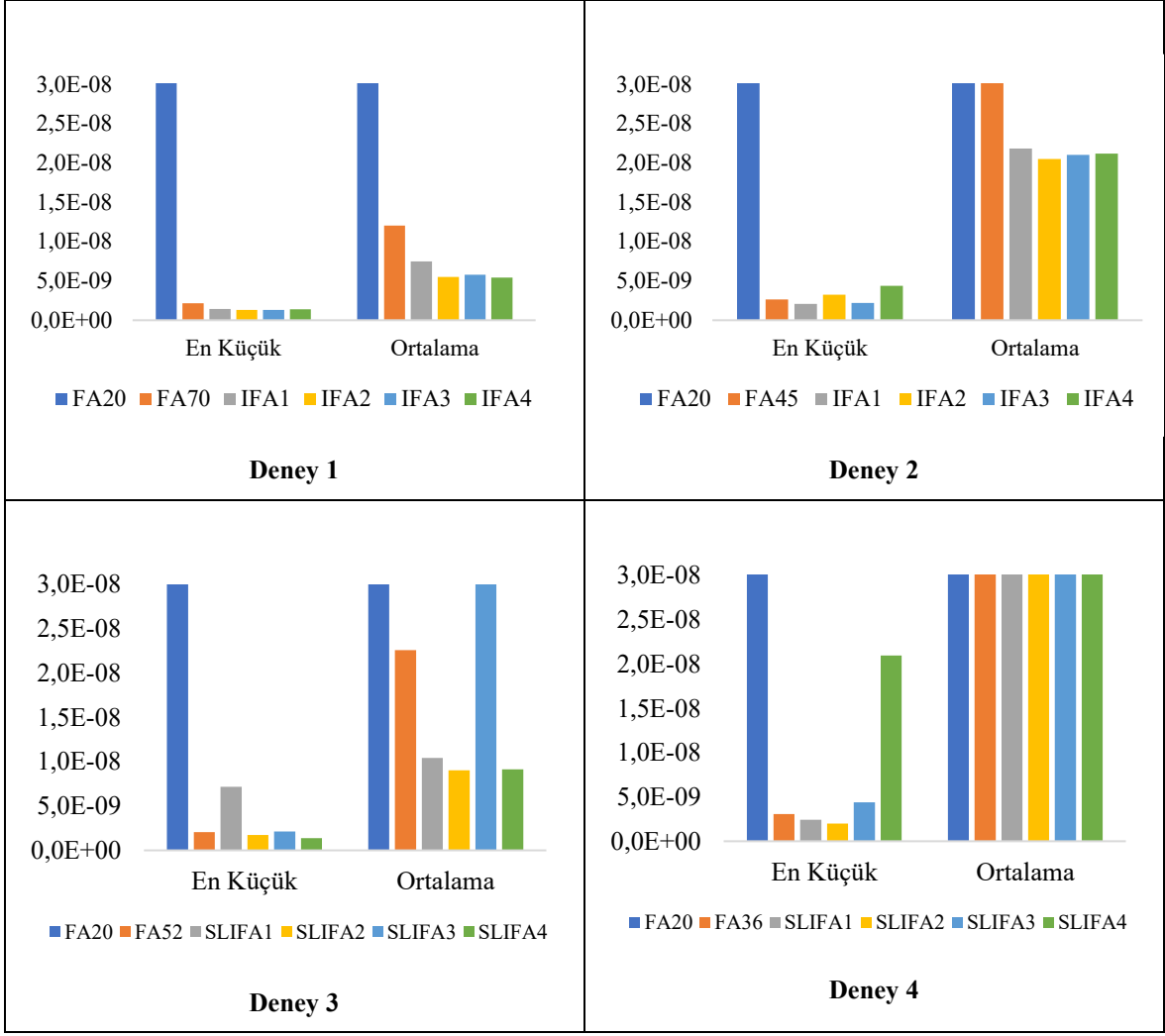
**Şekil 3.20:** Zakharov fonksiyonun Deney 1,2,3,4 değerleri.

Tüm deneysel bulgular birlikte değerlendirildiğinde, artımlı popülasyon yapısına sahip algoritmaların sabit popülasyonlu algoritmalara kıyasla üstün performans gösterdiği sonucuna ulaşılabilir. Bu sonuçlar, keşif ve sömürü arasındaki dengenin, özellikle çoklu yerel optimuma sahip problemlerde, algoritmanın genel başarısında belirleyici bir faktör olduğunu ortaya koymaktadır [51].



Şekil 3.21: Levy fonksiyonunun Deney 1,2,3,4 değerleri.

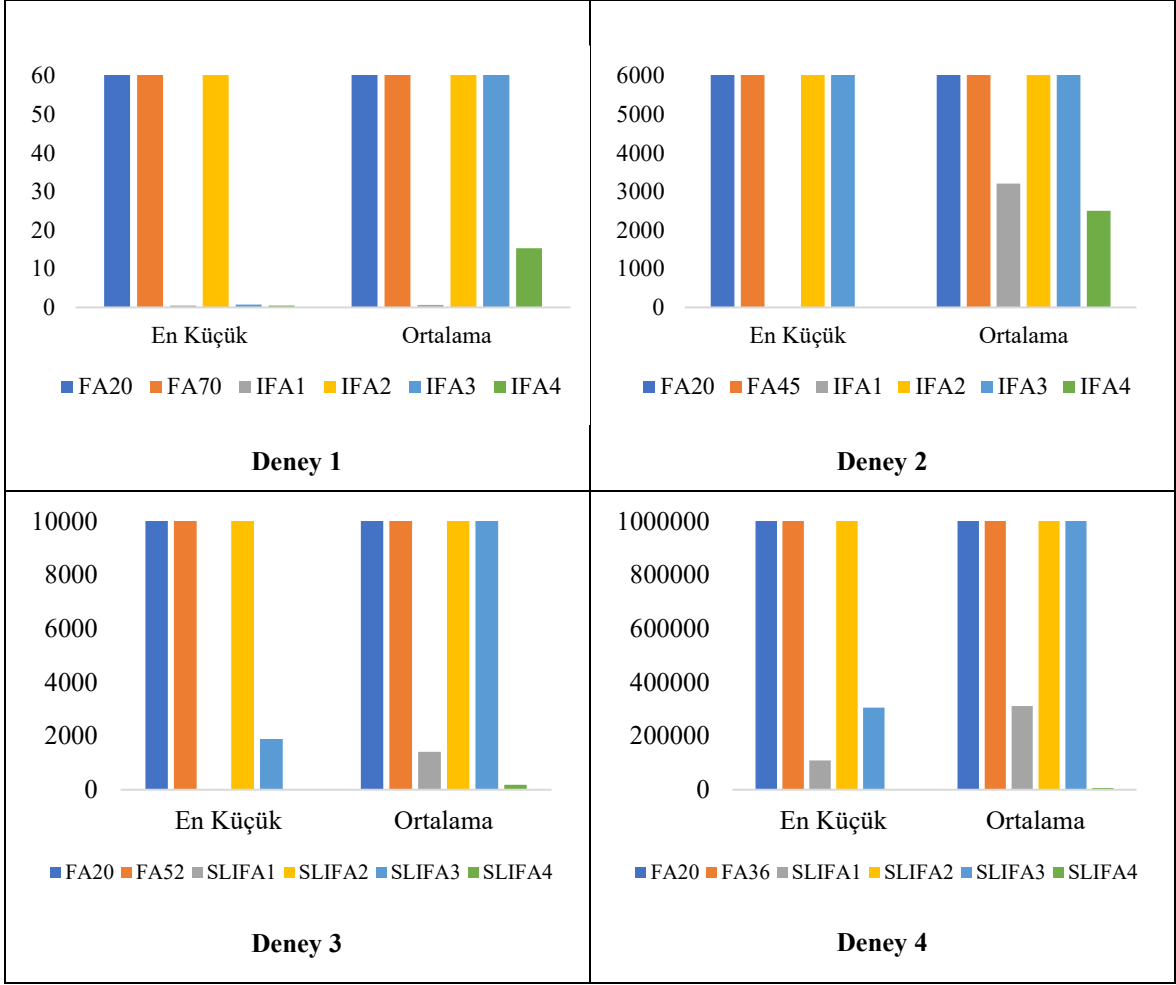
Sum Squares fonksiyonunun çalıştırılması sonucu elde edilen bulgular Şekil 3.22’de verilmiştir. Sum Squares fonksiyonunun unimodal yapısı gereği, algoritmanın sıfıra yakın çözümler üretmesi global optimuma doğru başarılı bir yakınsamaya işaret etmektedir. Bu fonksiyon türünde, küçük iyileştirmeler dahi çözüm uzayının merkezine (sıfıra) hızla yaklaşmayı mümkün kılmaktadır. Bu bağlamda, sıfıra yakın sonuçlar algoritmanın etkinliğini göstermektedir ve bu değerler, özellikle bazı durumlarda birbirine oldukça yakın seviyelere ulaşmıştır. Bazı grafiklerde yüksek sapma gösteren değerlerin baskın hale gelmesine veya grafik ölçeğine sığmamasına neden olmuştur. Dolayısıyla bazı çözüm değerleri, görsel karşılaştırmalarda grafik dışı kalmıştır.



**Şekil 3.22:** Sum Squares fonksiyonunun Deney 1,2,3,4 değerleri.

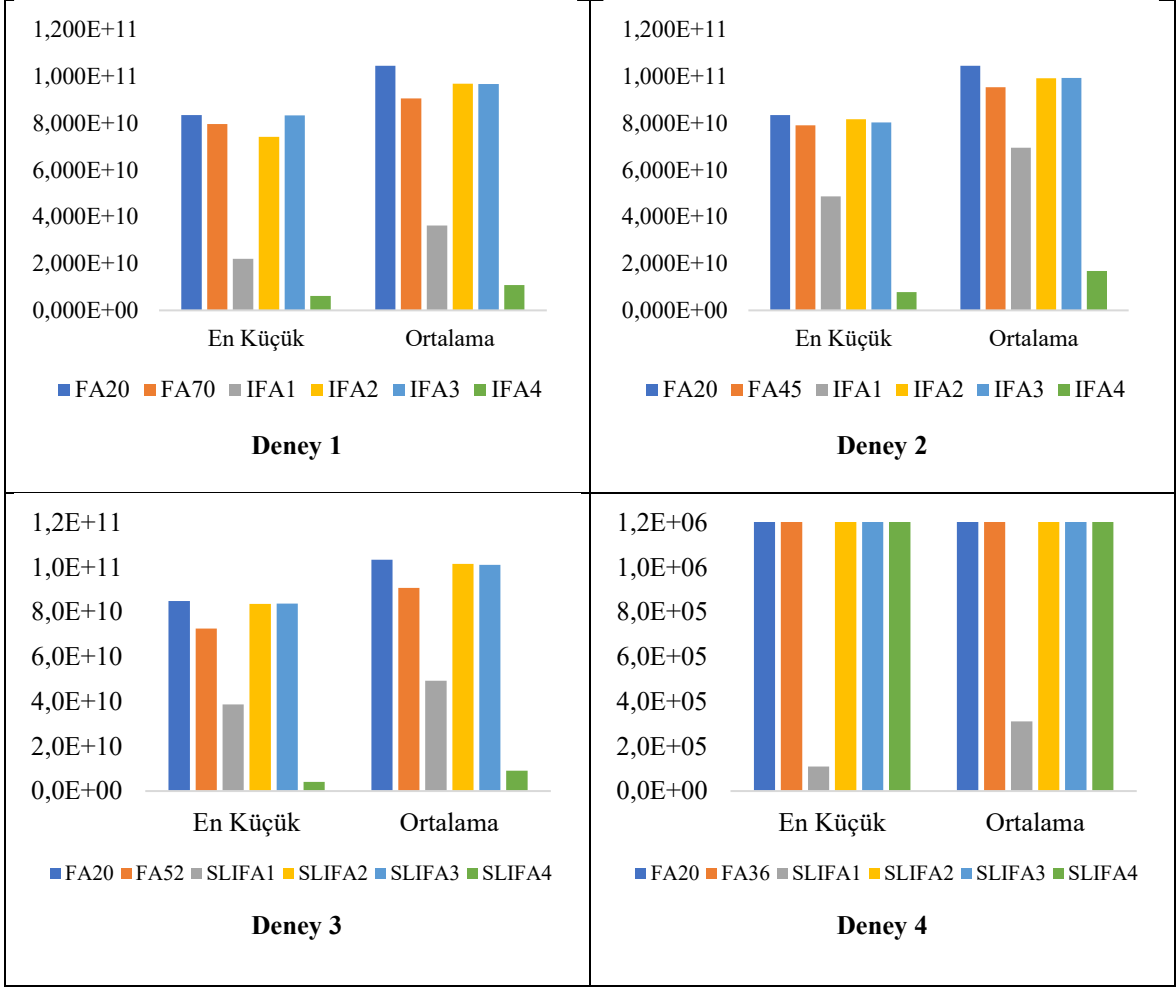
Dixon-Price fonksiyonu için elde edilen sonuçlar Şekil 3.23’de sunulmuştur. Algoritmanın tüm çalıştırmaları sonrasında ortaya çıkan ortalama değerler incelendiğinde, IFA4 ve SLIFA4 yapılandırmalarının ortalama değerler bakımından çoğunlukla en başarılı çözümlere ulaştığı görülmektedir.

Sonuç olarak, IFA4 ve SLIFA4 yapılandırmalarının hem en düşük hem de ortalama değerler açısından başarılı performans sergilemesi, bu yaklaşımın algoritmaya güvenilirlik ve kararlılık kazandırdığını göstermektedir. Özellikle çözüm uzayında belirgin minimumların bulunduğu ve algoritmanın odaklanması gereken bölgelerin net olduğu problemlerde, bu stratejilerin etkili bir şekilde çalıştığı söylenebilir.



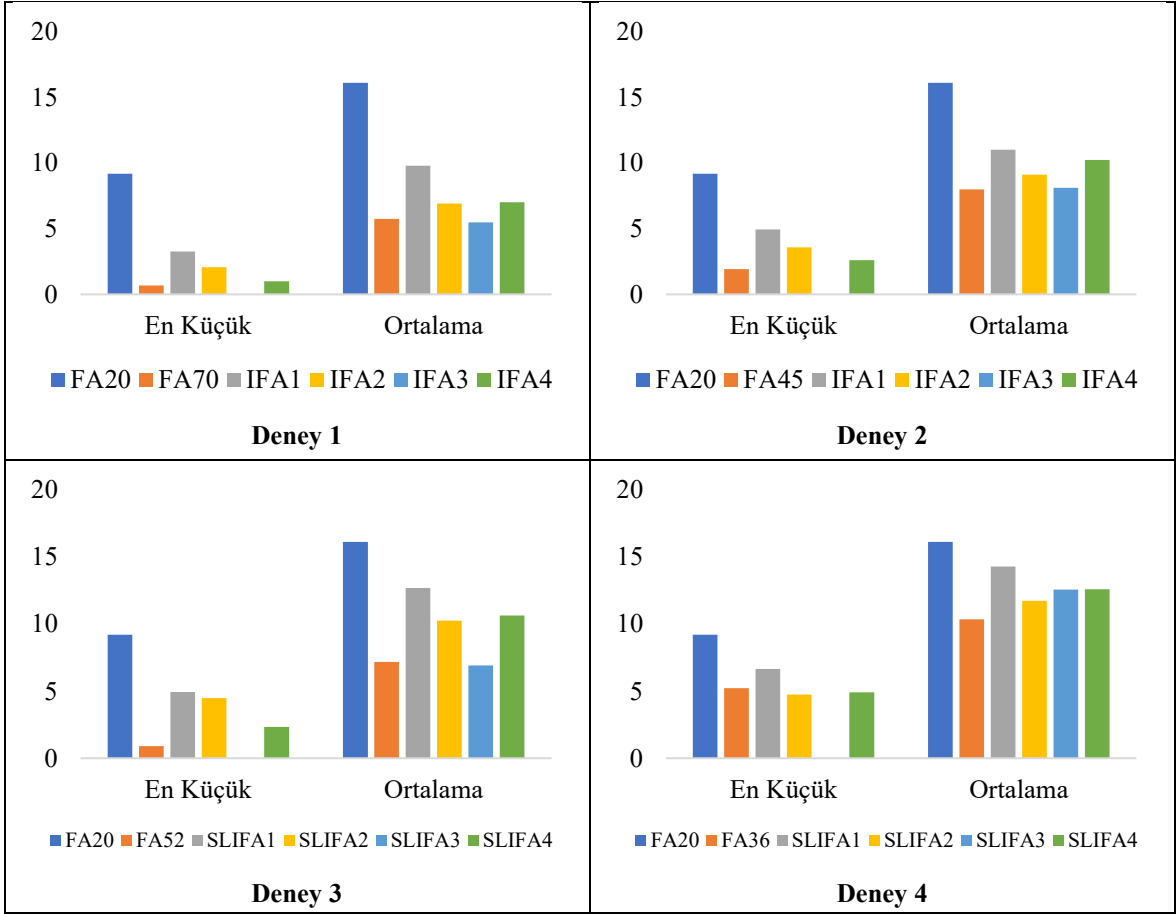
Şekil 3.23: Dixon Price fonksiyonunun Deney 1,2,3,4 değerleri.

Bent Cigar test fonksiyonu üzerinde uygulanan IFA'nın başarısı, farklı yeni birey ekleme durumları ve sosyal öğrenme yaklaşımları detaylı biçimde incelenmiş ve sonuçlar Şekil 3.24'te verilmiştir. Dört farklı stratejiyle üretilen yeni ateşböcekleri arasında, mevcut en iyi birey ile rastgele birey arasındaki vektörel farkı kullanan Durum-4 Deney 1, 2 ve 3 için en başarılı sonuçları üretmiştir. 5 iterasyonda bir birey eklenen Deney 1'de, 10 iterasyonluk arttırmalara göre daha başarılı sonuçlar elde edilmiştir. Ayrıca arttırımın 100 iterasyondan sonra başlatıldığı SLIFA durumlarından SLIFA1 ve SLIFA4 stratejileri ön plana çıkmaktadır.



**Şekil 3.24:** Bent Cigar fonksiyonunun Deney 1,2,3,4 değerleri.

Şekil 3.25'te Weierstrass fonksiyonunun en küçük ve ortalama sonuçlarına göre IFA için en iyi sonuçlar IFA3 durumunda elde edilmiştir. SLIFA için ise SLIFA3 durumu Deney 1, 2 ve 3'te hem en küçük hem de ortalama değerlerde en başarılı sonuçları üretmiştir. SLIFA ve IFA'nın sonuçları kıyaslandığında IFA daha başarılı sonuçlar vermiştir. SLIFA'nın daha az başarılı olmasının temel nedeni, erken iterasyonlarda birey sayısının sabit kalması ve keşif gücünün sınırlı kalmasıdır.



Şekil 3.25: Weierstrass fonksiyonunun Deneysel 1,2,3,4 değerleri.

#### 4. SONUÇ VE ÖNERİLER

Bu çalışmada, IFA ve SLIFA'nın performansı, sabit popülasyonlu klasik FA algoritmasıyla karşılaştırılmıştır. Deneysel bulgular, artımlı yapıların özellikle çözüm kalitesi, en küçük değer bulma başarısı ve ortalama performans açısından sabit yapıya göre çoğunlukla üstün olduğunu göstermiştir. Bu üstünlük, artan birey sayısının çözüm uzayını daha iyi taraması ve algoritmanın hem keşif hem de sömürü yeteneklerini geliştirmesiyle ilişkilidir. Özellikle en iyi birey ile rastgele birey arasındaki vektörel farkla yeni bireylerin oluşturulduğu arttırım stratejisi (IFA4 ve SLIFA4) çok sayıda test fonksiyonunda en başarılı sonuçları vermiştir.

Elde edilen sonuçlar artım aralığının performans üzerinde etkili olduğunu göstermiştir. Popülasyona her 5 iterasyonda bir birey eklenen yapılandırmalar, 10 iterasyonda bir birey eklenenlere kıyasla daha yüksek çözüm kalitesine ulaşmıştır. Bu da daha sık arttırımın, özellikle karmaşık ve çok modlu fonksiyonlarda algoritmanın daha detaylı bir keşif gerçekleştirmesine olanak tanıdığını göstermektedir. Öte yandan, algoritmaların başarıları test edilen fonksiyonların doğasına göre de değişiklik göstermektedir. Unimodal fonksiyonlarda sömürü odaklı stratejiler etkiliyken çok modlu fonksiyonlarda çeşitliliği koruyan ve keşif yeteneğini artıran stratejiler daha başarılı sonuçlar vermiştir.

Artımlı popülasyon yapısı, klasik FA algoritmasına kıyasla daha güçlü ve esnek bir optimizasyon yeteneği sunmaktadır. Hem IFA hem de SLIFA, yüksek boyutlu ve karmaşık problemlerde etkin birer çözüm aracı olarak değerlendirilebilir. Artımlı popülasyon, bireyler arası bilgi paylaşımı sayesinde algoritmanın verimli bir arama gerçekleştirmesini desteklemektedir. Bu nedenle, bireyler arası bilgi paylaşımının performansa olan katkısı, sadece sayısal sonuçlarla değil, algoritmanın genel arama dinamikleri üzerindeki olumlu etkisiyle birlikte değerlendirilmelidir.

Gelecek çalışmalarda, dinamik arttırım stratejileri kullanılarak algoritmanın uyarlanabilir hale getirilmesi, bireylerin ekleneceği zamanların problem özelliklerine göre belirlenmesi önerilebilir. Gerçek mühendislik problemleri ve çok hedefli optimizasyon senaryolarında algoritmanın uygulanması, elde edilen bulguların daha geniş bir bağlamda test edilmesini sağlayacaktır. Artan popülasyon boyutunun işlem yüküne etkisini azaltmak amacıyla paralel işlemeye uygun versiyonların geliştirilmesi algoritmanın kullanım alanını genişletecektir.

## 5. KAYNAKLAR (IEEE)

- [1] B. Durmuş, A. Gün, ve H. Temurtaş, “Artımsal Popülasyonlu Karga Arama Algoritması”, *Eur. J. Sci. Technol.*, Haz. 2021, doi: 10.31590/ejosat.932389.
- [2] S. Temiz, “Betonarme sürekli kirişlerin yapay arı koloni, öğretme-öğrenmeye dayalı optimizasyon ve öğretme-öğrenmeye dayalı yapay arı koloni algoritmalarıyla optimum tasarımı”, Yüksek Lisans, Karadeniz Teknik Üniversitesi, Trabzon, 2022.
- [3] A. Tülek, “Göçmen kuşlar optimizasyon algoritmasının paralel bilgisayarlarda uygulanması”, Yüksek Lisans, Balıkesir Üniversitesi, Balıkesir, 2019.
- [4] Z. Beheshti ve S. M. H. Shamsuddin, “A Review of Population-based Meta-Heuristic Algorithm”, *International Journal of Advances in Soft Computing and Its Applications*, Mar. 2013.
- [5] G. Dalkılıç ve F. Türkmen, “Karınca Kolonisi Optimizasyonu”, Yüksek Performanslı Bilişim Sempozyumu, 2002.
- [6] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, 30. print. Boston: Addison-Wesley, 2012.
- [7] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm”, *Comput. Struct.*, c. 169, ss. 1-12, Haz. 2016, doi: 10.1016/j.compstruc.2016.03.001.
- [8] H. Wang, S. Rahnamayan, ve Z. Wu, “Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems”, *J. Parallel Distrib. Comput.*, c. 73, sy 1, ss. 62-73, Oca. 2013, doi: 10.1016/j.jpdc.2012.02.019.
- [9] J. Kennedy ve R. Eberhart, “Particle swarm optimization”, *Proceedings of ICNN’95 - International Conference on Neural Networks*, Perth, WA, Australia: IEEE, 1995, ss. 1942-1948. doi: 10.1109/ICNN.1995.488968.
- [10] R. Rajabioun, “Cuckoo Optimization Algorithm”, *Appl. Soft Comput.*, c. 11, sy 8, ss. 5508-5518, Ara. 2011, doi: 10.1016/j.asoc.2011.05.008.
- [11] S. Mirjalili, S. M. Mirjalili, ve A. Lewis, “Grey Wolf Optimizer”, *Adv. Eng. Softw.*, c. 69, ss. 46-61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [12] X.-S. Yang, “Firefly Algorithms for Multimodal Optimization”, *Stochastic Algorithms: Foundations and Applications*, c. 5792, O. Watanabe ve T. Zeugmann, Ed., Lecture

- Notes in Computer Science, vol. 5792. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ss. 169-178. doi: 10.1007/978-3-642-04944-6\_14.
- [13] X. S. Yang, “Firefly algorithm, stochastic test functions and design optimisation”, *Int. J. Bio-Inspired Comput.*, c. 2, sy 2, s. 78, 2010, doi: 10.1504/IJBIC.2010.032124.
- [14] M. N. Seyman, “Determination of the cyclic prefix length in MC-CDMA using firefly algorithm”, *Pamukkale Univ. J. Eng. Sci.*, c. 23, sy 5, ss. 486-491, 2017, doi: 10.5505/pajes.2016.04764.
- [15] R. E. Neapolitan ve K. Naimipour, *Foundations of algorithms using C++ pseudocode*, 3rd ed. Sudbury, Mass: Jones and Bartlett, 2004.
- [16] J. Nocedal ve S. J. Wright, *Numerical optimization*, 2nd ed. Springer series in operations research. New York: Springer, 2006.
- [17] X.-S. Yang ve X. He, “Firefly Algorithm: Recent Advances and Applications”, 2013, doi: 10.48550/ARXIV.1308.3898.
- [18] I. Fister, I. Fister, X.-S. Yang, ve J. Brest, “A comprehensive review of firefly algorithms”, *Swarm Evol. Comput.*, c. 13, ss. 34-46, Ara. 2013, doi: 10.1016/j.swevo.2013.06.001.
- [19] X.-S. Yang ve S. Deb, “Cuckoo Search via Levy Flights”, 2010, doi: 10.48550/ARXIV.1003.1594.
- [20] X-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press 2nd Edition. Haziran 2010.
- [21] T. VonRueden, “Captivating Firefly Behavior: Exploring Their Mating Rituals”, 2019, Erişim: 7 Mart 2025 [Çevrimiçi]. Erişim adresi: <https://lampyridjournal.com>
- [22] V. R. Viviani, T. L. Oehlmeyer, F. G. C. Arnoldi, ve M. R. Brochetto-Braga, “A New Firefly Luciferase with Bimodal Spectrum: Identification of Structural Determinants of Spectral pH-Sensitivity in Firefly Luciferases”, *Photochem. Photobiol.*, c. 81, sy 4, ss. 843-848, Tem. 2005, doi: 10.1111/j.1751-1097.2005.tb01452.x.
- [23] K. Foo ve M. M. Dawood, “Diversity of Fireflies (Coleoptera: Lampyridae) of Sungai Teratak, Sabah, Malaysia”, *J. Trop. Biol. Conserv.*, 2015.
- [24] İ. B. Aydilek, “Değiştirilmiş ateşböceği optimizasyon algoritması ile kural tabanlı çoklu sınıflama yapılması”, *Gazi Üniversitesi Mühendis.-Mimar. Fakültesi Derg.*, c. 32, sy 4, ss. 1097-1108, Ara. 2017, doi: 10.17341/gazimmfd.369406.
- [25] G. Pekdemir, “Çoklu İmge Eşikleme Problemlerinde Metasezgisel Algoritmaların Performans Analizi”, Yüksek Lisans, Selçuk Üniversitesi, Konya, 2012.

- [26] Sh. M. Farahani, A. A. Abshouri, B. Nasiri, ve M. R. Meybodi, “A Gaussian Firefly Algorithm”, *Int. J. Mach. Learn. Comput.*, ss. 448-453, 2011, doi: 10.7763/IJMLC.2011.V1.67.
- [27] S. Palit, S. N. Sinha, M. A. Molla, A. Khanra, ve M. Kule, “A cryptanalytic attack on the knapsack cryptosystem using binary Firefly algorithm”, 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), Allahabad, India: IEEE, Eyl. 2011, ss. 428-432. doi: 10.1109/ICCCT.2011.6075143.
- [28] “A Teaching-Learning-Based Optimization Algorithm for the Weighted Set-Covering Problem”, *Teh. Vjesn. - Tech. Gaz.*, c. 27, sy 5, Eki. 2020, doi: 10.17559/TV-20180501230511.
- [29] M. A. M. De Oca, T. Stutzle, K. Van Den Eenden, ve M. Dorigo, “Incremental Social Learning in Particle Swarms”, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, c. 41, sy 2, ss. 368-384, Nis. 2011, doi: 10.1109/TSMCB.2010.2055848.
- [30] S. Özyön, “Differential Evolution Algorithm with Incremental Social Learning”, *Bilecik Şeyh Edebali Üniversitesi Fen Bilim. Derg.*, c. 7, sy 100. Yıl Özel Sayı, ss. 133-162, Mar. 2020, doi: 10.35193/bseufbd.666626.
- [31] Y. Liu, C. H. Mu, W. D. Kou, ve J. Liu, “Adaptive Particle Swarm Optimization with Dynamic Population and its Application to Constrained Engineering Design Optimization”, *Adv. Mater. Res.*, c. 538-541, ss. 3074-3078, Haz. 2012, doi: 10.4028/www.scientific.net/AMR.538-541.3074.
- [32] S. Özyön, C. Yaşar, ve H. Temurtaş, “Incremental gravitational search algorithm for high-dimensional benchmark functions”, *Neural Comput. Appl.*, c. 31, sy 8, ss. 3779-3803, Ağu. 2019, doi: 10.1007/s00521-017-3334-8.
- [33] T. Liao, M. A. Montes De Oca, D. Aydın, T. Stützle, ve M. Dorigo, “An incremental ant colony algorithm with local search for continuous optimization”, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, Dublin Ireland: ACM, Tem. 2011, ss. 125-132. doi: 10.1145/2001576.2001594.
- [34] D. Aydın, T. Liao, M. A. Montes De Oca, ve T. Stützle, “Improving Performance via Population Growth and Local Search: The Case of the Artificial Bee Colony Algorithm”, *Artificial Evolution*, c. 7401, J.-K. Hao, P. Legrand, P. Collet, N. Monmarché, E. Lutton, ve M. Schoenauer, Ed., *Lecture Notes in Computer Science*, vol. 7401. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ss. 85-96. doi: 10.1007/978-3-642-35533-2\_8.

- [35] M. A. Montes De Oca ve T. Stützle, “Towards incremental social learning in optimization and multiagent systems”, Proceedings of the 10th annual conference companion on Genetic and evolutionary computation, Atlanta GA USA: ACM, Tem. 2008, ss. 1939-1944. doi: 10.1145/1388969.1389004.
- [36] S. Surjanovic ve D. Bingham, “Rastrigin Function”, Simulation Optimization Test Functions and Datasets. Erişim: 03 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/rastr.html>
- [37] “Sphere Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/spheref.html>
- [38] “Ackley Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/ackley.html>
- [39] “Rosenbrock Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/rosen.html>
- [40] “Griewank Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/griewank.html>
- [41] “Schwefel Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/schwef.html>
- [42] “Zakharov Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/zakharov.html>
- [43] “Levy Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/levy.html>
- [44] “Sum Squares Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/sumsqu.html>
- [45] “Dixon-Price Function”. Erişim: 10 Mart 2025. [Çevrimiçi]. Erişim adresi: <https://www.sfu.ca/~ssurjano/dixonpr.html>
- [46] J. Zhao ve Z.-M. Gao, “The hybridized Harris hawk optimization and slime mould algorithm”, J. Phys. Conf. Ser., c. 1682, sy 1, s. 012029, Kas. 2020, doi: 10.1088/1742-6596/1682/1/012029.
- [47] “Weierstrass Function - an overview | ScienceDirect Topics”. Erişim: 11 Mart 2025. [Çevrimiçi]. Erişim adresi: [https://www.sciencedirect.com/topics/mathematics/weierstrass-function?utm\\_source](https://www.sciencedirect.com/topics/mathematics/weierstrass-function?utm_source)
- [48] H. H. Rosenbrock, “An Automatic Method for Finding the Greatest or Least Value of a Function”, Comput. J., c. 3, sy 3, ss. 175-184, Mar. 1960, doi: 10.1093/comjnl/3.3.175.

- [49] “Schwefel Function”. Eriřim: 04 Mart 2025. [Çevrimiçi]. Eriřim adresi: <https://www.sfu.ca/~ssurjano/schwef.html>
- [50] R. Storn ve Kenneth Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces”, *J. Glob. Optim.*, c. 11, sy 4, ss. 341-359, 1997, doi: 10.1023/A:1008202821328.
- [51] I. Boussaïd, J. Lepagnot, ve P. Siarry, “A survey on optimization metaheuristics”, *Inf. Sci.*, c. 237, ss. 82-117, Tem. 2013, doi: 10.1016/j.ins.2013.02.041.
- [52] A. E. Eiben ve J. E. Smith, *Introduction to evolutionary computing*, 1. ed. corr. 2. printing, Softcover version of original hardcover ed. 2003. Natural computing series. Berlin: Springer, 2010.

# ÖZGEÇMİŞ

## Kişisel Bilgiler

Adı Soyadı :Murat İŞMAN

Doğum tarihi ve yeri :15.06.1988

e-posta :ismanmurat@gmail.com

## Öğrenim Bilgileri

Derece	Okul/Program	Yıl
Lisans	Süleyman Demirel Üniversitesi	2006-2011
Lise	Balıkesir Lisesi	2001-2005