



**KENDİNDEN UYARLAMALI
BİR DİFERANSİYEL GELİŞİM ALGORİTMASI
TASARIMI VE GERÇEKLENMESİ**

Yüksek Lisans Tezi

Hatem DURLU

Kütahya - 2023

T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

**KENDİNDEN UYARLAMALI BİR DİFERANSİYEL GELİŞİM
ALGORİTMASI TASARIMI VE GERÇEKLENMESİ**

Danışman:
Doç. Dr. Gürcan YAVUZ

Hazırlayan:
Hatem DUMLU

Kütahya – 2023

Kabul Onay

KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında, 202085151072 öğrenci numaralı, Hatem DUMLU'nun hazırlamış olduğu "Kendinden Uyarlamalı Bir Diferansiyel Gelişim Algoritması Tasarımı ve Gerçeklenmesi" başlıklı yüksek lisans tez çalışması ile ilgili tez savunma sınavı jüri tarafından yapılmış ve adayın tezinin OY BİRLİĞİ ile kabul edilmesine karar verilmiştir.

06/07/2023

Tez Jürisi	İmza	
	Kabul	Red
Doç. Dr. Gürcan YAVUZ (Danışman)		
Dr. Öğr. Üyesi Pınar ÖZEN KAVAS		
Dr. Öğr. Üyesi Sevcan EMEK		

Onay

Doç. Dr. Arif KOLAY
Enstitü Müdürü

Bilimsel Etik Bildirimi

Yüksek Lisans tezi olarak hazırladığım “*Kendinden Uyarlamalı Bir Diferansiyel Gelişim Algoritması Tasarımı ve Gerçeklenmesi*” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

06/07/2023

Hatem DUMLU

Özgeçmiş

Hatem DURLU, ilköğretim ve ortaöğretimini İzmir’de tamamladı. 2012 yılında Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Bölümünde lisans eğitimine başladı. 2017 yılında lisans eğitimini tamamladı. Ankara ve İzmir’de iki farklı firmada stajını tamamladıktan sonra İzmir’de bir yıla yakın e-ticaret alanında yazılım geliştiren bir firmada çalıştı. 2018 yılından itibaren Eskişehir’de kalite ve risk yönetimi, ortam izleme sistemleri alanlarında farklı firmalarda iki yıl yazılım mühendisi olarak çalıştı. 2020 yılı eylül ayında Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Anabilim dalında yüksek lisans öğrenimine başladı. Aynı tarihte Trabzon Üniversitesi’nin kurumsal iletişim departmanında, yazılım geliştirme alanında göreve başladı. 2021 yılının şubat ayında Trabzon Üniversitesi’nden ayrılarak Kütahya Dumlupınar Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak göreve başladı ve yüksek lisans öğrenimine Kütahya Dumlupınar Üniversitesi Bilgisayar Mühendisliği Bölümünde devam etti. Halen Kütahya Dumlupınar Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak devam etmektedir. İngilizce bilmektedir. Optimizasyon algoritmaları, büyük veri gibi alanlarda yayınları bulunmakta ve bu alanlarda çalışmalarına devam etmektedir.

ÖZET

KENDİNDEN UYARLAMALI BİR DİFERANSİYEL GELİŞİM ALGORİTMASI TASARIMI VE GERÇEKLENMESİ

DUMLU, Hatem

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Gürcan YAVUZ

Haziran, 2023, 47 sayfa

Günümüz dünyasında insan nüfusunun artması üretim sektöründen ulaşım sektörüne, sağlık sektöründen kimya endüstrisine, yenilenebilir enerji kaynaklarından ağır sanayiye kadar insanoğlunun yaşamını ilgilendiren neredeyse her alana duyulan ihtiyacı artırmıştır. Bu çeşitlilikte talep edilen ihtiyaçlar, bazı problemleri de beraberinde getirmiştir. Bu problemler optimizasyon algoritmaları ile maliyeti azaltma, verimi artırma ya da bunları optimuma getirme şeklinde çözüme kavuşturulmaktadır. Bu problemleri çözmek için birçok optimizasyon algoritması yaklaşımı vardır. Bu optimizasyon yaklaşımlarının çoğunu meta sezgisel algoritmalar oluşturur. Diferansiyel Gelişim (DG) Algoritması da bir meta sezgisel algoritmadır. DG algoritması kolay kullanılabilirliği ve hızlı yakınsama özelliği ile bilinir. Fakat DG algoritması evrimsel süreç boyunca sabit tutulan çok az parametreye sahip olduğundan bu parametreleri doğru bir şekilde ayarlamak algoritmanın performansı açısından ayrı bir problemdir. Bu sebeple araştırmacılar kendinden uyarlanabilen kontrol parametrelili DG algoritmaları üzerine çalışmalar yürütmektedir. Bu çalışmada kendinden uyarlamalı bir diferansiyel gelişim algoritması tasarımı yapılmış ve gerçekleştirilmiştir. İlk adımda diferansiyel gelişim algoritmasının mutasyon evresine farklı bir mutasyon stratejisi uygulanmış ve CEC 2019 ölçüt seti fonksiyonlarında test edilmiş ve literatürdeki Genişletilmiş Balina Optimizasyon Algoritması-GBOA (Whale Optimization Algorithm Modified Mutualism-WOAmM), Balina Optimizasyon Algoritması-BOA (Whale Optimization Algorithm-WOA), Güve Alevi Optimizasyon Algoritması-GAOA (Moth Flame Optimization-MFO), Kelebek Optimizasyon Algoritması-KOA (Butterfly Optimization Algorithm-BOA), Sinüs Kosinüs Algoritması-SKA (A Sine Cosine Algorithm-SCA) ve JAYA algoritması ile karşılaştırılmıştır. CEC 2019 ölçüt seti 50 boyutta test edilmiştir. İkinci adımda ise bu DG algoritmasına rekabetçi yerel arama stratejisi de eklenerek CEC 2014 ölçüt seti fonksiyonlarında test edilmiş ve literatürdeki SHADE (Success-History based Adaptive Differential Evolution), L-SHADE (SHADE with Linear Population Size Reduction), PADE (Parameter Adaptive Differential Evolution) ve LPalmDE

(Parameters with Adaptive Learning Mechanism Differential Evolution) algoritmaları ile karşılaştırılmıştır. CEC 2014 ölçüt seti ise 30 ve 50 boyutta test edilmiştir. Elde edilen sonuçlara göre önerilen kendinden uyarlamalı DG algoritması “ortalama” ve “en iyi” sonuçlarda kıyaslanan algoritmalara karşı ilk sırada yer almıştır.

Anahtar Kelimeler: CEC 2014, CEC 2019, Diferansiyel Gelişim Algoritması, Kendinden Uyarlamalı Diferansiyel Gelişim Algoritması, Optimizasyon Algoritmaları



ABSTRACT**A SELF-ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM DESIGN
AND IMPLEMENTATION****DUMLU, Hatem****Master Thesis, Department of Computer Engineering****Supervisor : Assoc. Prof. Gürcan YAVUZ****July, 2023, 47 pages**

The increasing of human population has increased the demand for almost every area of human life in the production sector, transportation sector, health sector, chemical industry, renewable energy sources, heavy industry etc. These demands have led to some problems. These problems are solved by optimization algorithms by reducing cost, increasing efficiency or making them optimum values. There are many optimization algorithm approaches to solve these problems. Most of these optimization approaches are meta heuristic algorithms. Differential Evolution (DE) Algorithm is also a meta heuristic algorithm. The DE algorithm is known for its ease of use and fast convergence. However, since the DE algorithm has very few parameters that are kept constant throughout the evolutionary process, accurately tuning these parameters is a particular problem for the performance of the algorithm. For this reason, researchers have been working on DE algorithms with self-adaptive control parameters. In this study, a self-adaptive differential evolution algorithm is designed and implemented. In the first step, a different mutation strategy was applied to the mutation phase of the differential evolution algorithm and tested on CEC 2019 benchmark set functions and compared with Whale Optimization Algorithm Modified Mutualism-WOAmM, Whale Optimization Algorithm-WOA, Moth Flame Optimization Algorithm-MFO, Butterfly Optimization Algorithm-BOA, A Sine Cosine Algorithm-SCA and JAYA algorithms in the literature. CEC 2019 benchmark functions tested on 50 dimensions. In the second step, a competitive local search strategy was added to this DE algorithm and tested on the CEC 2014 benchmark functions and compared with SHADE (Success-History based Adaptive Differential Evolution), L-SHADE (SHADE with Linear Population Size Reduction), PADE (Parameter Adaptive Differential Evolution) and LPALMDE (Parameters with Adaptive Learning Mechanism Differential Evolution) algorithms in the literature. The CEC 2014 benchmark functions was tested in 30 and 50 dimensions. According to the results, the proposed self-adaptive DE algorithm ranked first in the "average" and "best" results against the compared algorithms.

Keywords: CEC 2014, CEC 2019, Differential Evolution Algorithm, Self-Adaptive Differential Evolution Algorithm, Optimization Algorithms

TEŐEKKÜR

İlk olarak Yüksek Lisans Tez Çalışması süresince geçirdiğim zorlu süreçte kendimi daha iyi geliştirmemi sağlayan, özverisini ve desteğini benden hiç esirgemeyen değerli danışman Hocam Sayın Doç. Dr. Gürcan YAVUZ'a teşekkürlerimi sunarım.

Bu zorlu süreçte sabrı, anlayışı ve manevi desteği ile daima yanımda olan eşim Tuğba DUMLU'ya teşekkürü bir borç bilirim.



İÇİNDEKİLER

Sayfa

ÖZET.....	v
TABLolar LİSTESİ	xi
KISALTMALAR.....	xii
GİRİŞ	1

BİRİNCİ BÖLÜM

ÖNCEKİ ÇALIŞMALAR

1.1. LİTERATÜR TARAMASI.....	4
1.1.1. İlkendirme Adımına Yapılan İyileştirmeler	6
1.1.2. Algoritma için Yeni Mutasyon Operatörünün Önerilmesi	7
1.1.3. Çaprazlama Katsayısı ve Kesirsel Ölçek Parametrelerinin Değerlerinin Belirlenme Yöntemi	9
1.1.4. Seçme Adımında Yapılan Değişiklikler	9
1.1.5. DG'nin Farklı Optimize Teknikleri ile Birleştirilmesi	9
1.1.6. DG için Mutasyon Havuz Yönteminin Önerilmesi	10

İKİNCİ BÖLÜM

MATERYAL VE METOT

2.1. OPTİMİZASYON ALGORİTMALARI	12
2.2. META SEZGİSEL ALGORİTMALAR.....	13
2.3. DİFERANSİYEL GELİŞİM ALGORİTMASI	14
2.3.1. İlkendirme	15
2.3.2. Mutasyon.....	15
2.3.3. Çaprazlama.....	16
2.3.4. Seçme.....	16
2.4. CEC ÖLÇÜT SETLERİ.....	17
2.4.1. CEC 2014 Ölçüt Seti	18
2.4.2. CEC 2019 Ölçüt Seti	19
2.5. PARAMETRE BELİRLEME KÜTÜPHANESİ	19
2.5.1. İrace	19

ÜÇÜNCÜ BÖLÜM

KENDİNDEN UYARLAMALI BİR DİFERANSİYEL GELİŞİM ALGORİTMASI

3.1. DENEYSEL ÇALIŞMA 1	22
3.1.1. Kendinden Uyarlamalı Mutasyon Operatörü Stratejisi	22
3.2. DENEYSEL ÇALIŞMA 2	24
3.2.1. Rekabetçi Yerel Arama Stratejisi.....	24
3.2.2. IPOPCMAES	26
3.2.3. POWELL Metodu.....	27

DÖRDÜNCÜ BÖLÜM

BULGULAR VE TARTIŞMA

4.1. PERFORMANS DEĞERLENDİRME METRİKLERİ	29
4.1.1. Aritmetik Ortalama.....	29
4.1.2. Standart Sapma.....	29
4.2. ÖNERİLEN ALGORİTMANIN PERFORMANS SONUÇLARI	29
4.2.1. Deneysel Çalışma 1' in CEC 2019 Sonuçları ve Literatür Karşılaştırması	30
4.2.2. Deneysel Çalışma 2'nin CEC 2014 Sonuçları ve Literatür Karşılaştırması	32

BEŞİNCİ BÖLÜM

SONUÇ VE ÖNERİLER

5.1. SONUÇ	39
KAYNAKÇA	41
DİZİN	47

TABLOLAR LİSTESİ

	<u>Sayfa</u>
Tablo 2.1: Diferansiyel Gelişim Algoritması Sözde Kodu.....	15
Tablo 2.2: CEC 2014 Ölçüt Seti Fonksiyonları	18
Tablo 2.3: CEC 2019 Ölçüt Seti Fonksiyonları	19
Tablo 2.4: Deneysel Çalışmalarda Önerilen Algoritmanın İrace Parametre Aracı ile Belirlenmiş Olan Kontrol Parametreleri	20
Tablo 3.1: Deneysel Çalışma 1'in Sözde Kodu	22
Tablo 3.2: Mutasyon Operatörü Üretme Şablonu	23
Tablo 3.3: Algoritmanın Genelleştirilmiş Arama Denklemindeki Her Bir Bileşen için Alternatif Seçenekler	23
Tablo 3.4: Tablo 3.3'deki Kısaltmaların Açıklamaları	23
Tablo 3.5: Önerilen Algoritmanın Mutasyon Stratejisi Sözde Kodu	24
Tablo 3.6: KUYA-DGA Sözde Kodu	25
Tablo 3.7: Rekabetçi Yerel Arama Stratejisi Sözde Kodu	26
Tablo 4.1: Önerilen Algoritmanın CEC 2019 Sonuçları	30
Tablo 4.2: Deneysel Çalışma 1 Sonuçları ve Literatürdeki Çalışmalar ile Karşılaştırılması.....	31
Tablo 4.3: Deneysel Çalışma 2'nin 30 Boyutta CEC 2014 Sonuçları ve DG Varyantları ile Karşılaştırılması.....	33
Tablo 4.4: Deneysel Çalışma 2'nin 50 Boyutta CEC 2014 Sonuçları ve DG Varyantları ile Karşılaştırılması.....	35

KISALTMALAR

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
BigOpt	Big Optimization
BOA	Balina Optimizasyon Algoritması
BOA	Butterfly Optimization Algorithm
CCODE	Chaotic Circle Oppositional Differential Evolution
CEC 2005	2005 Congress on Evolutionary Computation
CEC 2013	2013 Congress on Evolutionary Computation
CEC 2014	2014 Congress on Evolutionary Computation
CEC 2017	2017 Congress on Evolutionary Computation
CEC 2019	2019 Congress on Evolutionary Computation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CODE	Chaotic Oppositional Differential Evolution
CoDE	Composite Differential Evolution
CR	Crossover Rate
DE	Differential Evolution
DG	Diferansiyel Gelişim
EEG	Elektroensefalografi
FLDE	Fitness Landscape Differential Evolution
GAO	Güve Alevi Optimizasyonu
GBOA	Genişletilmiş Balina Optimizasyonu Algoritması
İrace	Iterative Racing
KKO	Karınca Kolonisi Optimizasyonu
KOA	Kelebek Optimizasyon Algoritması
KUYA-DGA	Kendinden Uyarlamalı Yerel Aramalı Diferansiyel Gelişim Algoritması
PALM-DE	Parameters with Adaptive Learning Mechanism Differential Evolution
LPalmDE	PALM-DE with Linear Population Size Reduction
L-SHADE	SHADE with Linear Population Size Reduction
MAXFES	Maximum Function Evaluations

MFO	Moth Flame Optimization
MTSLS1	Multiple Trajectory Search
NBOLDE	Neighborhood Mutation and Opposition-Based Learning Differential Evolution
OLS	Objective Oriented Local Segmenting
PADE	Parameter Adaptive Differential Evolution
PSO	Parçacık Sürü Optimizasyonu-Particle Swarm Optimization
PTA	Purified Terephthalic Acid
SaDE	Self Adaptive Differential Evolution
SCA	Sine Cosine Algorithm
SHADE	Success-History Adaptive Differential Evolution
SinDE	Sinusoidal Differential Evolution-SinDE
SKA	Sinüs Kosinüs Algoritması
SOA	Simbiyotik Organizmalar Araması
SOS	Symbiotic Organisms Search
TDE	Two-Stage Differential Evolution
WOA	Whale Optimization Algorithm
WOAmM	Whale Optimization Algorithm Modified Mutualism
YAK	Yapay Arı Kolonisi
YSA	Yapay Sinir Ağı



TEZ METNİ

GİRİŞ

Son yıllarda hızla gelişen teknoloji ve teknolojinin bize sunduğu imkanlar beraberinde çözülmeyi bekleyen yeni problemleri doğurmuştur. Bu problemler insan nüfusunun hızla artması ve beraberinde gelen ihtiyaç talepleri doğrultusunda problemlere çabuk uyum sağlayabilen, hızlı ve verimli çözümlere duyulan ihtiyacı artırmıştır. Optimizasyonun problemlere karşı olan çözüm yaklaşımı ve sunduğu verimli çözümlerle son yıllarda popülerleşmiş ve sunduğu bu çözümlerle araştırmacıların ve çoğu sektörün iştahını kabartmıştır (Abualigah vd., 2022; Pham ve Castellani, 2014). Kısaca tanımlamak gerekirse, optimizasyon kısıtlı şartlar altında verilen sınırlı kaynakların, en verimli ve etkili bir şekilde kullanılmasıyla ilgilenen bir araştırma alanıdır (Talbi, 2009). Dolayısıyla çoğu alanda kullanılabilir. Bunlara sağlık (Vommi ve Battula, 2023), ulaşım (Balakrishnan, 2016), mühendislik (Houssein vd., 2022), ekonomi-finans (Vasant, 2012) gibi alanlar örnek verilebilir.

Beyaz ve Sümer 65 hastanın pelvik bölgesinden alınan X-ray görüntülerinden konvolüsyonel sinir ağı (Convolutional Neural Network-CNN) mimarisi ile femur boyun kırığı tespiti gerçekleştirmişlerdir (Beyaz vd., 2020). CNN mimarisinin hiper parametrelerini optimize etmek için ise Genetik Algoritma'yı kullanmışlardır.

Siddiqui vd. petrol yüklü gemilerin kazalarında yaşanan sosyoekonomik ve çevresel problemlere çözüm olabilecek koşullu risk değeri (conditional value at risk-CVaR) tabanlı bir optimizasyon yaklaşımı önermişlerdir (Siddiqui vd., 2023). Bu yaklaşım uygun büyüklükteki ham petrol tankerlerinin seçilmesi ve ardından bunların kıtalararası ağ üzerinde kaza riskini minimuma çekecek şekilde rotalarının belirlenmesini öngörür.

Optimizasyon problemlerinin artışı ve daha geniş alanlara uygulanabilirliği ihtiyacı, yeni optimizasyon algoritmalarının üretilmesinin yanında, araştırmacıları mevcut optimizasyon algoritmalarında da değişiklik yaparak daha verimli algoritmalar bulmalarına kapı aralamaktadır. Yavuz vd. Yapay Arı Kolonisi-YAK (Artificial Bee Colony-ABC) algoritmasının arama denkleminde yaptıkları değişiklik ile Yapay Arı Kolonisi algoritmasının yeni bir varyantını önermişlerdir (Yavuz vd., 2016).

Bu çalışmada Kendinden Uyarlamalı Bir Diferansiyel Gelişim Algoritması önerilmiştir. Önerilen algoritma deneysel çalışma 1 ve deneysel çalışma 2 olarak iki aşamada tasarlanmıştır. Her aşama için farklı ölçüt setleri ile test edilmiş ve literatürdeki

çalışmalarla kıyaslanmıştır. Deneysel çalışma 1’de, genetik bir algoritma olan diferansiyel gelişim algoritmasının mutasyon evresine kendinden uyarlamalı terim tabanlı bir mutasyon stratejisi yaklaşımı uygulanmış ve CEC 2019 ölçüt seti fonksiyonlarında test edilmiştir. Ortalama, standart sapma ve en iyi metrikleri kullanılarak literatürdeki çalışmalar ile kıyaslanmıştır. Deneysel çalışma 2’de ise önerilen algoritmaya Powell ve IPOPCMEAS yerel arama yaklaşımlarının yarıştırdığı rekabetçi yerel arama stratejisi eklenmiştir. Ortalama, standart sapma ve sıralama metrikleri kullanılarak literatürdeki çalışmalar ile kıyaslanmıştır.

Bu çalışmada birinci bölümde Diferansiyel Gelişim Algoritması ile ilgili literatür çalışmalarına yer verilmiştir.

İkinci bölümde optimizasyon algoritmalarına genel bir bakış ve ardından meta sezgisel algoritmalara değinildikten sonra Diferansiyel Gelişim Algoritması hakkında detaylı bilgi verilmiştir. Önerilen algoritmanın testlerinin yapıldığı CEC ölçüt setleri hakkında bilgi verildikten sonra CEC 2014, CEC 2019 ölçüt setleri anlatılmış ve fonksiyonları verilmiştir. Ayrıca parametre belirlemede kullanılan irace paketi anlatılmıştır.

Üçüncü bölümde önerilen algoritmanın, deneysel çalışma 1 ve deneysel çalışma 2 başlıkları altında iki aşamada yapılan tasarımı detaylı şekilde anlatılmıştır. Deneysel çalışma 1’de kullanılan mutasyon stratejisi ve deneysel çalışma 2’de kullanılan rekabetçi yerel arama stratejisi anlatılmıştır.

Dördüncü bölümde performans değerlendirmesi ve literatür kıyaslaması için kullanılan değerlendirme metriklerine yer verilmiştir. Önerilen algoritmanın performans sonuçları ve literatürdeki diğer çalışmalar ile olan kıyaslanması incelenmiştir.

Beşinci bölümde elde edilen sonuçlar ve çıkarımlar verilmiştir. İlerideki çalışmalar için öneriler sunulmuştur.



BİRİNCİ BÖLÜM
ÖNCEKİ ÇALIŞMALAR

1.1. LİTERATÜR TARAMASI

Son yıllarda optimizasyon algoritmaları büyük gelişme göstermiştir. Bu optimizasyon algoritmaların birçoğunu popülasyon tabanlı meta sezgisel algoritmalar oluşturur. Bu meta sezgisel algoritmalar evrimsel, sürü zekası, biyolojiden esinlenen veya fizik tabanlı olabilir (Ahmed vd., 2020; Dhiman, 2021). Literatüre ait meta sezgisel algoritmalarla ait bazı çalışmalar aşağıdaki gibidir.

Meta sezgisel ve sezgisel yaklaşımların en temel sorunlarından biri yerel optimuma takılma problemi (Chakraborty vd., 2021). Erken yakınsama olarak da adlandırılan bu durum yerel olarak uygun olan bir çözümün ilgili optimizasyon problemi için en iyi çözüm zannedilmesi durumudur. Yerel iyi bir çözüm bulunmuştur fakat bulunan çözüm global en iyi çözüm değildir. Chakraborty vd. Simbiyotik Organizmalar Aramasındaki-SOA (Symbiotic Organisms Search-SOS) mutualizm aşamasını erken yakınsamadan kaçınmak için değiştirmiş ve Balina Optimizasyon Algoritmasına eklemiştir. Böylece arama uzayının daha kapsamlı keşfini ve kaynakların boşa harcanmamasını hedeflemiştir. WOAmM adını verdikleri bu algoritmayı gerçek mühendislik problemlerinin çözümünde kullanmış ve CEC 2019 ölçüt setinde test etmişlerdir (Chakraborty vd., 2021).

Mirjalili ve Lewis kumbur balinaların avlanma davranışını taklit eden yeni bir meta sezgisel optimizasyon algoritması önermişlerdir (Mirjalili ve Lewis, 2016). Balina Optimizasyon Algoritması-BOA (Whale Optimization Algorithm-WOA) adını verdikleri bu algorithmada kumbur balinaların kabarcık ağı saldırı mekanizmasından esinlenilmiştir. Önerdikleri algoritmayı analiz etmek için 29 matematiksel kıyaslama fonksiyonu ile test etmişlerdir. Literatürdeki son teknoloji meta sezgisel yöntemlerle kıyaslandığında BOA'nın oldukça rekabetçi olduğunu ortaya koymuşlardır.

Mirjalili, güvelerin geceleri gerçekleştirdiği hareket stratejilerinden esinlenerek Güve Alevi Optimizasyonu-GAO (Moth Flame Optimization - MFO) algoritmasını önermiştir (Mirjalili, 2015). Güveler geceleri aya göre sabit bir açıyla uçarak düz bir çizgide uzun yollar katederken yapay bir ışıkla karşılaştıkları zaman etrafında spiral çizerek bir yörüngeye hapsolmelerini matematiksel olarak modellemiştir. GAO algoritmasını 29 kıyas fonksiyonu ve 7 gerçek mühendislik probleminde test ederek doğadan ilham alan diğer meta sezgisel algoritmalar ile kıyaslamıştır. İstatistiksel sonuçlara göre GAO algoritmasının oldukça umut verici ve rekabetçi olduğunu ortaya

koymuştur.

Arora ve Singh global optimizasyon problemlerini çözmek için kelebeklerin yiyecek arama ve çiftleşme davranışlarını taklit eden Kelebek Optimizasyon Algoritması-KOA (Butterfly Optimization Algorithm - BOA) adını verdikleri yeni bir algoritma önermişlerdir (Arora ve Singh, 2019). Algoritma temel olarak kelebeklerin koku duyularını kullanarak için eş ve çiçek nektarlarının yerini belirleme stratejisine dayanmaktadır. KOA algoritmasını 30 test fonksiyonu üzerinde diğer meta sezgisel algoritmalar ile kıyaslamışlardır. Ayrıca klasik mühendislik problemlerinde çözmek için kullanmışlardır. Sonuçlara göre KOA algoritmasının diğer meta sezgisel yaklaşımlara kıyasla daha verimli olduğunu göstermişlerdir.

Mirjalili optimizasyon problemlerini çözmek amacıyla yeni bir optimizasyon algoritması olan Sinüs Kosinüs Algoritmasını-SKA (A Sine Cosine Algorithm – SCA) önermiştir (Mirjalili, 2016). Bu algoritma, başlangıç için birden fazla rastgele aday çözümü oluşturur ve sinus ve kosinüs fonksiyonlarını kullanarak en iyi çözüme ulaşmayı amaçlar. Ayrıca arama uzayının keşfi için uyarlanabilir bazı değişkenler entegre edilmiştir. SKA'nın yerel optimumdan kaçınma, yakınsama gibi özellikleri test edilmiştir. Ayrıca bir uçak kanadının kesiti SKA ile optimize edilmiştir. Mirjalili, SKA'nın kısıtlı ve bilinmeyen arama uzaylarına sahip gerçek mühendislik problemlerinde oldukça başarılı olabileceğini ortaya koymuştur.

Rao, kısıtlı ve kısıtsız optimizasyon problemlerini çözmek için sadece ortak kontrol parametre değerlerine ihtiyaç duyan, algoritmaya özgü herhangi bir kontrol parametresi gerektirmeyen, basit ama güçlü JAYA algoritmasını tanıtmıştır (Rao, 2016). Bu algoritmada amaç verilen problemde en iyi çözüme ulaşmak ve en kötü çözümden kaçınmaktır. Rao, JAYA algoritmasının performansını CEC 2006 ölçüt setinde test ederek literatürdeki iyi bilinen optimizasyon algoritmaları ile kıyaslamıştır. Rao, JAYA algoritmasını Friedman'ın sıralama testinde 24 tane kısıtlı kıyaslama probleminin hepsinde “en iyi” ve “ortalama” çözümlerde birinci sırayı aldığını göstermiştir.

Meta sezgisel algoritmaların bir kısmını da Evrimsel Algoritmalar temsil eder. Doğadan ilham alan mekanizmaları kullanan ve canlı organizmaların davranışlarını taklit eden süreçleri kapsayarak sorunları çözen Evrimsel Algoritmalar, meta sezgisel optimizasyon yöntemlerin arasında büyük öneme sahiptir.

Evrimsel algoritmalarından birisi de Diferansiyel Gelişim-DG (Differential Evolution-DE) algoritmasıdır. DG algoritması güçlü bir meta sezgisel algoritma olmasına rağmen algoritma yerel optimuma takılma gibi çeşitli problemleri de mevcuttur. Bu problemlerin üstesinden gelebilmek için araştırmacılar, DG algoritmasının adımlarına, çeşitli iyileştirmeler önermişlerdir. Bu bölümde DG algoritması üzerinde yapılan iyileştirmeler alt başlıklar halinde sunulmuştur.

1.1.1. İlkendirme Adımına Yapılan İyileştirmeler

DG algoritmasının yakınsama hızını artırmak için araştırmacılar orijinal DG'nin ilkendirme adımını, çeşitli tekniklerle değiştirmişlerdir. Literatürde DG algoritmasına yapılan iyileştirmelerin çoğunun mutasyon ve çaprazlama operatörü üzerine olduğunu belirten Kumar vd., DG ilkendirme adımı için uyguladıkları ortogonal dizi tabanlı başlatma (orthogonal array-based initialization) ve komşuluk arama (neighborhood search) stratejisini kullanmışlardır (Kumar vd., 2022). Ayrıca algoritmanın performansını artırmak için bir seçim şeması sunmuşlardır. Önerdikleri ilkendirme ve seçim şemalarını DG'nin çeşitli varyantları üzerinde test etmişlerdir. Yapılan değişiklik DG varyantlarının performansını önemli ölçüde artırmıştır.

Rahnamayan vd. ile Deng vd. ise ilkendirme adımı için aykırılık tabanlı öğrenme (opposition-based learning) yöntemini tercih etmişlerdir (Rahnamayan vd., 2007; Deng vd., 2021). Çünkü popülasyon ilkendirme aşaması yakınsama hızını ve nihai çözüm kalitesini etkiler (Rahnamayan vd., 2007).Yapılan çalışmada rastgele ilkendirme yönteminin, aykırılık tabanlı ilkendirme ile değiştirilmesinin algoritmanın yakınsama hızını artırdığını göstermişlerdir.

Ahmad vd. DG algoritmasının ilkendirme adımı için kaotik haritalar ve aykırılık tabanlı öğrenme (oppositional-based learning) stratejisi kullanarak başlangıç popülasyon çeşitliliğini artırmayı hedeflemişler ve kaotik aykırılıklı DG (Chaotic Oppositional Differential Evolution - CODE) adında bir varyant önermişlerdir (Ahmad vd., 2022). CODE algoritmasını 7 farklı kaotik harita ile birleştirerek, bu varyantları CEC 2014 ölçüt setinde test etmişlerdir. En performanslı DG varyantını, kaotik daire aykırılıklı DG (Chaotic Circle Oppositional Differential Evolution- CCODE) olarak belirlemişlerdir. CCODE diğer varyantlarına göre daha hızlı yakınsama gerçekleştirmiştir. Aynı zamanda popülasyon çeşitliliğini koruma konusunda diğer varyantlarına oranla daha iyi performans göstermiştir.

1.1.2. Algoritma için Yeni Mutasyon Operatörünün Önerilmesi

Diferansiyel Gelişim algoritması üzerine yapılan ikinci iyileştirme kategorisi yeni bir mutasyon operatörünün önerilmesidir. Çünkü DG algoritması için mutasyon stratejisinin seçimi, keşif yeteneği, yakınsama doğruluğu ve yakınsama hızı gibi optimizasyon performansında önemli bir rol oynar (Deng vd., 2021).

Meng vd. İki-Aşamalı Diferansiyel Gelişim Algoritması (Two-stage Differential Evolution -TDE) adını verdikleri varyantlarında DG algoritmasının evrimsel sürecini iki safhaya ayırmışlardır (Meng ve Yang, 2022). Bu iki safha birbirlerinden farklı mutasyon stratejilerine sahiptir. Bunlara ek olarak algoritmalarında F ve CR parametrelerini belirlemek için kontrol stratejileri önermişlerdir. Algoritmalarının performansını CEC 2013, CEC 2014 ve CEC 2017 ölçüt setlerinde test etmişlerdir. Önerilen TDE algoritmasını CEC 2014 yarışmasının kazananı L-SHADE (Tanabe ve Fukunaga, 2014) algoritması ile karşılaştırmışlar ve 30 fonksiyonun 20'sinde benzer veya daha iyi performans sergilediğini ortaya koymuşlardır. Ayrıca TDE algoritmasının CEC 2017 yarışmasının kazanan DG varyantı jSO (Brest vd., 2017) algoritması ile karşılaştırıldığında 30 fonksiyonun 19'unda benzer veya daha iyi performans sergilediği görülmüştür.

Zhang vd. ise "DE/current-to-pbest" mutasyon stratejisini kullanan ve F ve CR parametrelerini Cauchy ve Gaussian dağılım ile belirleyen JADE algoritmasını geliştirmişlerdir ve bir çok algoritmanın temelini oluşturmuşlardır (Zhang ve Sanderson, 2009). Bu çalışmaya göre JADE nispeten büyük boyutlu optimizasyon problemleri için umut verici sonuçlar göstermiştir.

Tanabe ve Fukunaga JADE algoritmasını temel alan ve geçmişe dayalı bellek depolama mekanizmasını parametre belirleme yöntemi olarak kullanan DG varyantı olan SHADE algoritmasını sunmuşlardır (Tanabe ve Fukunaga, 2013). Daha sonra, SHADE algoritmasının popülasyon boyutunu zamana göre azaltılmasını sağlayan stratejiyi ekleyerek L-SHADE algoritmasını önermişlerdir (Tanabe ve Fukunaga, 2014).

Houssein vd. popülasyonu çeşitlendirilmesi için üç tane mutasyon mekanizması ile uyarlamalı parametre yöntemini DG algoritmasına entegre ederek Uyarlanabilir Güdümlü Diferansiyel Gelişim (Adaptive Guided Differential Evolution) Algoritmasını mühendislik problemlerine uygulamıştır (Houssein vd., 2022).

Deng vd. komşuluk mutasyonu (neighborhood mutation) mekanizmasını DG algoritmasına ekleyerek, komşuluk mutasyon operatörlü ve aykırılık tabanlı öğrenme stratejisine sahip NBOLDE adında yeni bir DG varyantı önermişlerdir (Deng vd., 2021). NBOLDE'nin 12 test fonksiyonunda test edilerek popüler DG varyantları ile karşılaştırıldığında daha hızlı yakınsama hızına ve yakınsama doğruluğuna sahip olduğu ortaya konmuştur. Ayrıca yüksek boyutlu karmaşık fonksiyonların çözümünde daha iyi sonuç vermiştir.

Tan vd. uygunluk manzarası (fitness landscape) yaklaşımını temel alan uyarlamalı mutasyon stratejisini DG algoritmasına dahil etmişlerdir (Tan vd., 2021). Önerdikleri algoritmaya Uygunluk Manzarası Diferansiyel Gelişim (Fitness Landscape Differential Evolution-FLDE) adını vermişlerdir. FLDE, CEC 2017 ölçüt setinde test edilmiş popüler 5 DG varyantı ile karşılaştırıldığında oldukça rekabetçi sonuçlar vermiştir.

Meng vd. DG algoritmasının mutasyon stratejisinde değişiklik yaparak Adaptif Öğrenme Mekanizmalı Parametrelili DG (Parameters with Adaptive Learning Mechanism Differential Evolution-PALM-DE) algoritmasını önermişlerdir (Meng vd., 2018). PALM-DE kontrol parametresi seçimindeki uygunsuzlukları gidermek ve eski bir mutasyon stratejisini geliştirmek için önerilmiş olup CEC 2013 ve CEC 2014 ölçüt setlerindeki 44 fonksiyonda denenmiş, en bilinen DG varyantları ile karşılaştırıldığında oldukça rekabetçi olduğu saptanmıştır. PALM-DE'nin performansını artırmak için doğrusal popülasyon boyutu azaltımı (linear population size reduction) yöntemi uygulanmış ve hala rekabetçi olduğu görülmüştür. Buna ise LPalmDE adını vermişlerdir.

Meng vd. uygun olmayan kontrol parametresi adaptasyon şemaları ve verilen mutasyon stratejisindeki kusurların sebep olduğu yavaş yakınsama gibi zayıflıkların üstesinden gelebilecek bir algoritma önermişlerdir (Meng vd., 2019). Parametre Uyarlamalı Diferansiyel Gelişim Algoritması (Parameter Adaptive Differential Evolution –PADE) adını verdikleri bu algoritma CEC 2013 ve CEC 2017 ölçüt setleri ile test edilmiştir. Sonuçlara göre hem optimizasyon doğruluğu hem de yakınsama hızı açısından PADE algoritması popüler DG varyantları ile rekabet edebilmektedir.

1.1.3. Çaprazlama Katsayısı ve Kesirsel Ölçek Parametrelerinin Değerlerinin Belirlenme Yöntemi

Orijinal DG algoritmasında kesirsel ölçek F (Scale factor-F) ve çaprazlama oranı (Crossover Rate-CR) parametre değerleri algoritma başında belirlenir. Algoritma çalışması tamamlanıncaya kadar da bu parametreler değişmez. Parametre değerlerin sabit olarak belirlenmesi yaklaşımının aksine araştırmacılar bu iki parametrenin değerlerini dinamik olarak belirleyen yaklaşımlar önermişlerdir.

Brest vd. Cr ve F parametrelerini meta sezgisel kurallar kullanarak belirleyen bir kendinden uyarlamalı (self-adaptive) bir DG varyantı önermişlerdir (Janez Brest vd., 2006). Sonuçlara göre kendinden uyarlamalı kontrol parametresine sahip önerilen algoritma klasik DG ve literatürdeki evrimsel algoritmalarından daha iyi sonuçlar vermiştir.

Draa vd. kesirsel ölçek F değerini ve çaprazlama oranı CR parametre değerlerinin otomatik olarak belirlemek için sinüsoidal formülleri kullanan bir DG varyantı önermişlerdir (Draa vd., 2015). Sinüsoidal Diferansiyel Gelişim (Sinusoidal Differential Evolution-SinDE) adını verdikleri bu algoritmada amaç arama uzayındaki daha az ziyaret edilen bölgelerin keşfedilmesi ile iyi çözümlerden daha çok faydalanma arasındaki dengeyi kurabilmektir. SinDE, CEC 2013 ölçüt setinde test edilerek klasik DG ve diğer popüler meta sezgisel algoritmalar ile karşılaştırıldığında daha iyi performans verdiği ortaya konmuştur.

1.1.4. Seçme Adımında Yapılan Değişiklikler

Bu kategoride Zeng vd. DG algoritmasının seçme adımını değiştirerek, algoritmanın yerel optimum değerden kaçınma yeteneğini artıran farklı bir yaklaşım önermişler ve 58 adet test fonksiyonu ile önerdikleri algoritmayı test etmişlerdir (Zeng vd., 2021). Yapılan çalışmaya göre seçme adımında yapılan değişikliğin DG algoritmasının performansını önemli ölçüde artırdığı saptanmıştır.

1.1.5. DG'nin Farklı Optimize Teknikleri ile Birleştirilmesi

Bir diğer iyileştirme kategorisi DG algoritmasının başka teknikler ile birleştirilmesidir. Bunlara örnek; Lin vd. DG algoritması ile PSO algoritmalarını hibritleştirerek orijinal DG'nin performansını artırmaya çalışmışlardır (Lin vd., 2022). Daha sonra DG için yerel arama olarak hedef tabanlı yerel segmentasyon (objective-

oriented local segmenting-OLS) stratejisini önermişler ve tek görevli (single objective) ile çok görevli (multi objective) optimizasyon problemlerinde kullanmışlardır (Lin vd., 2022).

Elyased vd. büyük veri BigOpt (Big Optimization) problemleri için önerdikleri DG çatısı algoritmaya, iç nokta (interior point) metodunu yerel arama algoritması olarak eklemiştirler (Elsayed ve Sarker, 2016). Sonuçlar önerilen algoritmanın çeşitli popüler algoritmalara göre üstünlüğünü göstermiştir.

1.1.6. DG için Mutasyon Havuz Yönteminin Önerilmesi

Son iyileştirme kategorisi ise mutasyon denklem havuzu yaklaşımını araştırmacılar önermişlerdir. Örneğin; Qin vd. Kendinden Uyarlamalı Diferansiyel Gelişim (Self Adaptive Differential Evolution-SaDE) adını verdikleri "DE/rand/1", "DE/rand-to-best/2", "DE/rand/2" ve "DE/current-to-rand/1" mutasyon stratejilerini önceki başarılarına göre bir havuzdan seçen bir DG varyantı önermişlerdir (Qin vd., 2008). SaDE algoritması 26 adet kısıtlı sayısal optimizasyon problemi üzerinde test edilmiş, klasik DG ve birkaç popüler kendinden uyarlamalı DG ile karşılaştırılmıştır.

Wang vd. büyük ölçekli özellik seçimi (large scale feature selection) problemi için çoklu popülasyon stratejisini, F ve CR parametrelerini dinamik olarak belirlemek için kendinden uyarlamalı bir yöntemini, mutasyon denklemi için sekiz adet mutasyon denklemi ile doldurulmuş olan havuz stratejisini ve son olarak özellik belirlemek için ağırlıklandırılmış model (weighted model) yöntemini DG'ye uyarlayarak bir varyant önermişlerdir (Wang vd., 2022). SaWDE adını verdikleri bu algoritma test sonuçlarına göre oldukça verimli sonuçlar üretmiştir.

Ayrıca Wang vd. üç parametre ayarı ve üç mutasyon stratejisini içeren aday havuzuna sahip bir DG varyantı olan Bileşik Diferansiyel Gelişim (Composite Differential Evolution - CoDE) algoritmasını önermişlerdir (Wang vd., 2011). Bu algoritmaya göre, üç parametre ayarı ve mutasyon stratejileri rastgele seçilerek belirlenmektedir. CoDE, CEC 2005 ölçüt seti ile test edilmiş, rekabetçi sonuçlar ortaya koymuştur.



İKİNCİ BÖLÜM
MATERYAL VE METOT

2.1. OPTİMİZASYON ALGORİTMALARI

Optimizasyon algoritmaları, belirli bir hedefi en iyi şekilde gerçekleştirmek için en iyi çözümü bulmayı amaçlayan matematiksel ve hesaplama tabanlı yöntemlerdir (Levy, 2022). Bu algoritmalar, sınırlı kaynakların verimli kullanılmasını sağlamak, maliyetleri azaltmak, performansı artırmak veya başka bir hedefi optimize etmek için kullanılır. Bu hedef, genellikle belirli bir kısıtlar kümesi altında maksimum veya minimum değeri elde etmek olabilir. Optimizasyon algoritmaları, mühendislik, ekonomi, sağlık gibi çeşitli alanlarda kullanılır (Abualigah vd., 2022; Beyaz vd., 2020; Dhiman, 2021; Houssein vd., 2022; Vasant, 2012). Bu alanlara ait bazı örnekler aşağıda verilmiştir.

Li vd. petrokimya ve kimyasal elyaf endüstrilerindeki en önemli ara maddelerinden biri olan saflaştırılmış tereftalik asit (PTA) maddesinin üretim aşamalarında oluşabilecek hataları azaltmak için diferansiyel gelişim algoritmasını kullanarak ekonomik kontrol şeması tasarım yöntemini geliştirmişlerdir. (Liv d., 2017). DG algoritması kullanılarak ekonomik kontrol şeması tasarımını bir optimizasyon problemine dönüştürülmüş ve PTA'nın üretim sürecinde meydana gelebilecek hatalar baz alınarak kar kaybını minimuma düşürmek hedeflenmiştir.

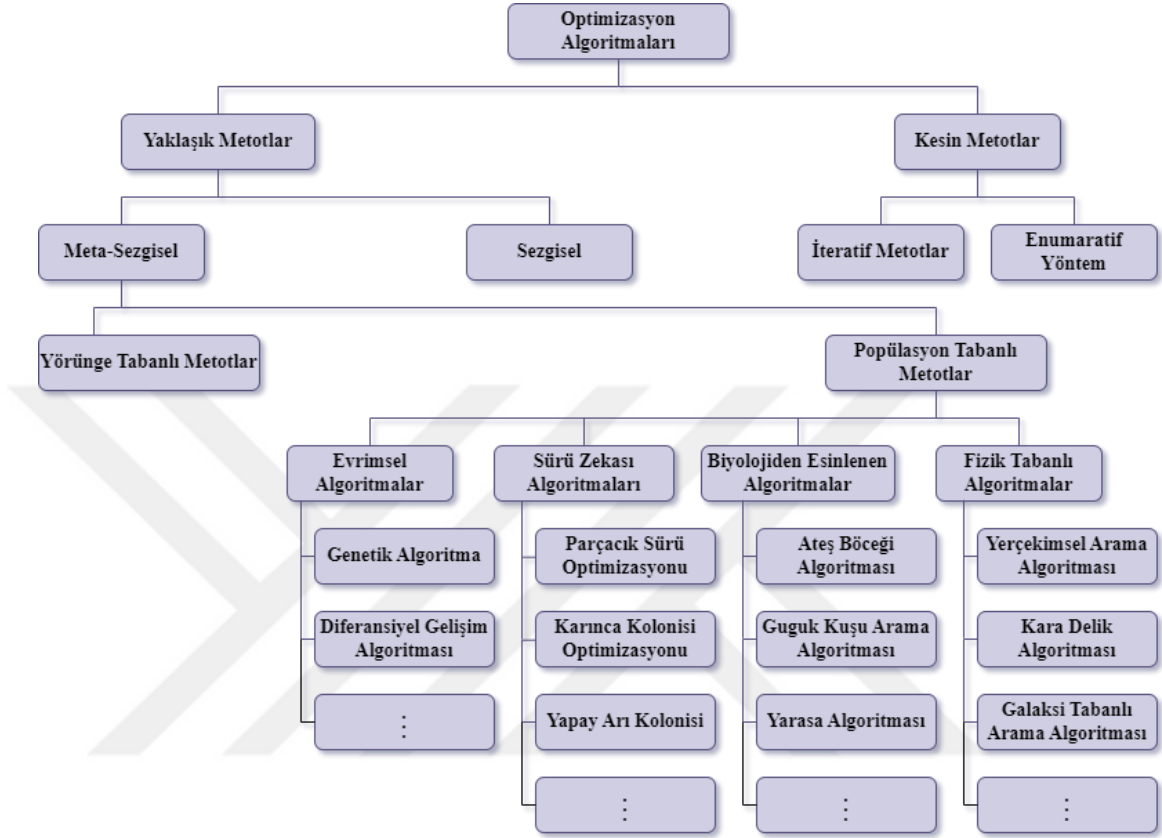
Sharma vd. borsa tahmini için yaptıkları çalışmalarında, hisse senetlerindeki verilerin doğrusal olmaması sebebiyle, geleneksel veya tek bir teknik yaklaşımla sonuç tahmininde yetersiz kalabileceğini savunmuşlardır. Bu nedenle borsa tahmini için Yapay Sinir Ağları (YSA) ve Genetik Algoritma'nın melezlenmesi ile oluşan yeni bir yöntem önermişlerdir (Sharma vd., 2022). Bu yöntemde, tahmin için ABD borsa endeksi olan DOW30 ve NASDAQ100 kullanılmıştır. Önerdikleri hibrit yöntemin sadece tek YSA kullanılarak çıkan sonuçlara oranla daha başarılı olduğunu ortaya koymuşlardır.

Sumathi vd. beyin tümörü tespitinde radyologlara yardımcı olmak amacı ile otomatik bir tümör tespit yöntemi önermişlerdir (Sumathi vd., 2022). Bu yöntemde Kernel-Fuzzy C Means (Kernel FCM) algoritmasına Parçacık Sürü Optimizasyonu-PSO (Particle Swarm Optimization-PSO) algoritmasını entegre ederek, beyin MR görüntülerinden tümörlü bölge otomatik tespit edilmiştir. Önerilen hibrit yöntemin beyin tümörü tespitinde %97,6 kesinlik elde ederek mevcut yaklaşımlardan daha üstün olduğu kanıtlanmıştır.

Yukarıdaki çalışmalara bakıldığında optimizasyon algoritmalarının geniş bir alana uygulandığı, insanoğlu ve yaşamını kapsayan neredeyse her alanda oluşan

problemlere çözüm olduğu göze çarpmaktadır. Optimizasyon algoritmalarının genel sınıflandırılma şeması Şekil 2.1’de (Ahmed vd., 2020; Dhiman, 2021) verilmiştir.

Şekil 2.1: Optimizasyon Algoritmalarının Sınıflandırılması



2.2. META SEZGİSEL ALGORİTMALAR

Meta sezgisel algoritmalar, çeşitli sezgisel yöntemleri birleştiren ve genel bir çözüm sunan bir optimizasyon yaklaşımıdır (Hussain vd., 2019). Doğal sistemlerin prensiplerini taklit ederek, en iyi çözümleri bulmak için arama alanını etkili bir şekilde keşfetme yetenekleri vardır. Bu algoritmalarda, doğal sistemlerin evrim, sürü davranışı veya diğer benzer süreçlerinden ilham alınarak problem çözümü gerçekleştirilir. Bu tür algoritmalar, karmaşık ve genellikle çok boyutlu optimizasyon problemlerini etkili bir şekilde çalışabilen sezgisel optimizasyon yöntemleridir. Bu nedenle, çeşitli uygulama alanlarında (ekonomi, sağlık, mühendislik vb.) kullanılmaktadırlar (Sumathi vd., 2022; Li vd., 2017; Sharma vd., 2022).

Meta sezgisel algoritmaların temel amacı, en iyi çözümü bulmak için arama alanını etkili bir şekilde keşfetmektir. Bu algoritmalarda, çözüm adayları (bireyler,

parçacıklar vb.) bir popülasyon şeklinde temsil edilir ve iteratif olarak geliştirilir. Her iterasyonda, mevcut popülasyon değerlendirilir, uygunluk fonksiyonları kullanılarak her bir çözüm adayı değerlendirilir ve bir sonraki nesil için yeni çözüm adayları üretilir. Meta sezgisel algoritmalarla örnek olarak, Genetik Algoritma-GA (Genetic Algorithm -GA), Parçacık Sürü Optimizasyonu-PSO (Particle Swarm Optimization-PSO), Karınca Kolonisi Optimizasyonu-KKO (Ant Colony Optimization-ACO) algoritmaları örnek verilebilir.

- *Genetik Algoritma*: Genetik algoritma, biyolojik evrimin temel prensiplerini taklit eden bir meta sezgisel yaklaşımdır. Bir başlangıç popülasyonu üzerinde çalışır ve genetik operatörler olan çaprazlama ve mutasyonu kullanarak yeni nesiller üretir. Popülasyon, uygunluk fonksiyonlarına göre değerlendirilir ve en iyi çözüm bulunmaya çalışılır (Srinivas ve Patnaik, 1994).

- *Parçacık Sürü Optimizasyonu*: Parçacık sürü optimizasyonu, bir sürü davranışını taklit eden bir meta sezgisel yaklaşımdır. Her parçacık, konum ve hız bilgileriyle temsil edilir. Parçacıklar, en iyi buldukları konumları ve sürü lideri tarafından sağlanan rehberlikle hareket ederek arama alanında gezinir. Bu şekilde, en iyi çözüm bulunmaya çalışılır (Wang vd., 2018).

- *Karınca Kolonisi Optimizasyonu*: Karınca kolonisi optimizasyonu, gerçek karınca kolonilerinin izlediği iz ve feromonlarla ilgili davranışlarını taklit eden bir meta sezgisel yaklaşımdır. Karıncalar, izledikleri yollar üzerinde feromon bırakır ve diğer karıncalar bu feromonları takip ederek en iyi yolu bulur. Bu algoritma, optimizasyon problemlerinde yol bulma ve en iyi çözümü bulma konularında etkilidir (Blum, 2005).

Meta sezgisel algoritmaların avantajları arasında genel uygulanabilirlik, problem bağımsızlığı, paralelleştirme yeteneği ve karmaşık arama alanlarında etkili performans gösterme potansiyeli bulunur. Ancak, bu algoritmaların da bazı zorlukları vardır, örneğin parametre ayarlaması, yakınsama hızı ve yerel optimuma takılma gibi konular dikkate alınmalıdır (Meng vd., 2019).

2.3. DİFERANSİYEL GELİŞİM ALGORİTMASI

Diferansiyel Gelişim (DG) Algoritması, R. Storn ve K. Price tarafından optimizasyon problemlerinin çözülmesi için önerilmiş olan popülasyon tabanlı bir meta sezgisel algoritmadır (Storn ve Price, 1997). DG'nin sözde kodu Tablo 2.1'deki gibidir.

Orijinal DG algoritması, ikklendirme, mutasyon, çaprazlama ve seçme adımlarından oluşmaktadır (Das ve Suganthan, 2010).

Tablo 2.1: Diferansiyel Gelişim Algoritması Sözde Kodu

```

NP boyutunda ilk popülasyonu oluştur
Do while
  For popülasyondaki  $\forall j$  bireyi
     $r_1, r_2, r_3 \in (1, NP)$  olmak üzere üç rastgele tamsayı oluştur ( $r_1 \neq r_2 \neq r_3 \neq j$ )
     $i_{rand} \in (1, n)$  olmak üzere rastgele bir tamsayı oluştur
    For  $\forall i$  parametresi
       $X'_{i,j} = \begin{cases} X_{i,r_3} + F \times (x_{i,r_1} - x_{i,r_2}) & \text{if } rand(0,1) < CR \text{ or } i = i_{rand} \\ X_{i,j} & \text{diğer durumda} \end{cases}$ 
    End For
    Eğer yeni oluşan birey ( $X'_j$ ),  $X_j$  'den daha iyi ise  $X_j$ 'yi  $X'_j$  ile yer değiştir.
  End For
Sonlandırma koşulu sağlanana kadar

```

2.3.1. İkklendirme

DG algoritması optimizasyon işlemine, D boyutlu , çözümleri rastgele üreterek NP boyutlu popülasyon ile başlar. Bu popülasyona ikklendirme popülasyonu adı verilir ve aşağıda yer alan Denklem 2.1 ile üretilmektedir:

$$X_{ij} = X_L + rand(X_U - X_L) \quad (2.1)$$

Burada, X_U ve X_L problemin en büyük ve en küçük sınır değerlerini göstermektedir. rand, uniform dağılım kullanılarak [0,1] aralığından üretilmiş olan rastgele bir sayıdır.

2.3.2. Mutasyon

Her iterasyonda t 'de, her birey X_i 'ye -hedef birey olarak tanımlanır- mutasyon operatörü adı verilen bir işlem uygulanır. Bu işlemin sonucunda hedef bireye ait olan bir mutant birey adı verilen bir birey V_i üretilir. Bunun için literatürde araştırmacıların önerdiği çok sayıda mutasyon operatörü mevcuttur.

Bunlardan bazıları şunlardır (Aydın vd., 2017; Gürcan vd., 2019);

DE\rand\1

$$V_i = X_{r_1} + F(X_{r_2} - X_{r_3}) \quad (2.2)$$

DE\rand\2

$$V_i = X_{r1} + F(X_{r2} - X_{r3} + X_{r4} - X_{r5}) \quad (2.3)$$

2.3.3. Çaprazlama

Çaprazlama, evrimsel algoritmalarda iki bireyin karşılıklı olarak genlerinin değiştirilmesini ifade eder. Bu işlemin temel amacı oluşan yeni bireylerin genetik çeşitliliğini artırarak daha geniş bir arama alanında keşif imkanı sunmaktır. CR-çaprazlama oranı popülasyon içindeki her bir çift bireyin çaprazlama işlemine tabi tutulma olasılığını belirler. Dolayısıyla yüksek CR değeri çaprazlama işleminin daha sık gerçekleşmesi ve akabinde popülasyonun hızlı bir şekilde çözüm alanına yayılmasını sağlar. Fakat bu durumda mevcut iyi çözümlerin kaybedilme olasılığı ve algoritmanın lokal optimuma takılma problemi artar. Aksine düşük bir CR değeri çaprazlamanın daha az gerçekleşmesine ve bu durumda da popülasyon daha temkinli bir büyüme gerçekleştirmesini sağlar. Bu durum da algoritmanın daha iyi çözümleri koruma yeteneğini artırırken daha az genetik çeşitlilik olması sebebiyle algoritmanın keşif yeteneğini törpüler. Bu sebeplerden dolayı CR değeri problemin özelliklerine ve aranan çözüme göre seçilmelidir çünkü CR algoritmanın çalışmasının başında kullanıcı tanımlı olarak belirlenen bir parametredir. CR, Denklem 2.4'teki geçici çözümde kullanılır.

$$u_{ij} = \begin{cases} V_{ij}, & \text{if } rand_j < CR \text{ or } j == j_{rand} \\ X_{ij}, & \text{diğer durumda} \end{cases} \quad (2.4)$$

u_{ij} , i. geçici ,çözümü temsil etmektedir. Bu ,çözüm, X_i ve V_i 'nin çaprazlanması sonucunda oluşmaktadır. j_{rand} , D boyutu göstermek üzere $[1, D]$ aralığından rastgele seçilen bir sayıdır.

2.3.4. Seçme

Diferansiyel Gelişim algoritmasındaki son operatördür. Bu adımda üretilmiş olan geçici çözüm ile mevcut çözüm arasında karar verilmektedir. Hangi çözüm daha iyi ise bir sonraki iterasyondaki popülasyona dahil olur.

$$X_i^{t+1} = \begin{cases} X_i^t, & \text{if } f(X_i^t) < f(u_i^t) \\ u_i^t, & \text{otherwise} \end{cases} \quad (2.5)$$

2.4. CEC ÖLÇÜT SETLERİ

CEC bir dizi test fonksiyonu içeren, optimizasyon alanında yaygın olarak kullanılan ölçüt setleridir. CEC, "IEEE Congress on Evolutionary Computation (CEC)" konferansının bir parçası olarak düzenlenen bir yarışma olan "CEC2005 Special Session on Real-Parameter Optimization"da tanıtılmıştır ve her yıl tekrarlanarak, ölçüt seti fonksiyonları güncellenmekte ve CEC 2013, CEC 2014, CEC 2019 gibi yıl bazlı isimlendirilmektedir. CEC ölçüt fonksiyonları, optimizasyon algoritmalarının performansını değerlendirmek ve karşılaştırmak için kullanılır. Bu fonksiyonlar, gerçek değerli optimizasyon problemlerini temsil etmek üzere tasarlanmıştır ve genellikle global optimuma ulaşılması zor olan çok boyutlu ve karmaşık fonksiyonlardır. Bunun nedeni, optimizasyon algoritmalarının performansını gerçek hayattaki zorlu problemlere daha iyi uyarlamak ve test etmek için tasarlanmış olmalarıdır. Optimizasyon algoritmalarının yerel optimuma takılma problemleri mevcuttur. Bu yüzden bu ölçüt setleri optimizasyon algoritmalarının performanslarını test etmek ve optimizasyon algoritmalarını birbiri ile kıyaslamak için önemli bir yere sahiptir.

Optimizasyon algoritmaları, CEC ölçüt fonksiyonlarını kullanarak global optimuma mümkün olduğunca yakın bir çözüm bulmayı hedefler. Bu fonksiyonlar, algoritmaların konumlarının ve çözüm kalitelerinin ölçülmesi için kullanılan uygunluk değeri veya hata değerini hesaplamak için kullanılır. Bu değerler, algoritmaların performansını karşılaştırmak, yeni algoritmaları değerlendirmek ve geliştirmek için kullanışlı bir ölçüt sağlar.

Son yıllarda yapılan çalışmalar, CEC ölçüt fonksiyonlarının iyileştirilmesi, yeni fonksiyonların eklenmesi ve farklı optimizasyon stratejilerinin değerlendirilmesi gibi konuları ele almaktadır. Bu çalışmalarda, fonksiyonların özelliklerinin ve zorluk düzeylerinin analizi, optimizasyon algoritmalarının performansının artırılması ve CEC ölçüt fonksiyonlarının genel kullanımının geliştirilmesi gibi konular üzerinde çalışılmaktadır. Sonuç olarak CEC ölçüt fonksiyonları, evrimsel hesaplama alanında standartlaşma ve karşılaştırılabilirlik sağlar. Araştırmacılar, bu fonksiyonları kullanarak yöntemlerini değerlendirebilir, geliştirebilir ve diğer çalışmalarla karşılaştırabilirler.

2.4.1. CEC 2014 Ölçüt Seti

Liang vd. gerçek parametrelili optimizasyon problemlerini çözmek ve yeni üretilen optimizasyon algoritmaların da test edilebilmek için CEC 2013 fonksiyonlarına yapılan yorumlamaları da baz alarak yeni bir ölçüt seti tanıtmışlardır. Bu, CEC 2014 ölçüt setidir ve 3 adet tek-modlu, 13 adet çok-modlu, 6 adet hibrit ve 8 adet bileşik olmak üzere 30 adet test fonksiyonu vardır. Bu fonksiyonlar Tablo 2.2'deki gibidir (Liang vd., 2013).

Tablo 2.2: CEC 2014 Ölçüt Seti Fonksiyonları

	No	Fonksiyonlar	$F_i^*=F_i(x^*)$
Tek-Modlu Fonksiyonlar	F1	Döndürülmüş Yüksek Koşullu Eliptik Fonksiyon	100
	F2	Döndürülmüş Bent Cigar Fonksiyon	200
	F3	Döndürülmüş Discus Fonksiyon	300
Basit Çok-Modlu Fonksiyonlar	F4	Kaydırılmış ve Döndürülmüş Rosenbrock Fonksiyonu	400
	F5	Kaydırılmış ve Döndürülmüş Ackley Fonksiyonu	500
	F6	Kaydırılmış ve Döndürülmüş Weierstrass Fonksiyonu	600
	F7	Kaydırılmış ve Döndürülmüş Griewank Fonksiyonu	700
	F8	Kaydırılmış Rastrigin Fonksiyonu	800
	F9	Kaydırılmış ve Döndürülmüş Rastrigin Fonksiyonu	900
	F10	Kaydırılmış Schwefel Fonksiyonu	1000
	F11	Kaydırılmış ve Döndürülmüş Schwefel Fonksiyonu	1100
	F12	Kaydırılmış ve Döndürülmüş Katsuura Fonksiyon	1200
	F13	Kaydırılmış ve Döndürülmüş HappyCat Fonksiyon	1300
	F14	Kaydırılmış ve Döndürülmüş HGBat Fonksiyon	1400
	F15	Kaydırılmış ve Döndürülmüş Expanded Griewank + Rosenbrock Fonksiyonu	1500
	F16	Kaydırılmış ve Döndürülmüş Expanded Scaffer F6 Fonksiyon	1600
Hibrit Fonksiyonlar	F17	Hibrit Fonksiyon 1 ($N=3$)	1700
	F18	Hibrit Fonksiyon 2 ($N=3$)	1800
	F19	Hibrit Fonksiyon 3 ($N=4$)	1900
	F20	Hibrit Fonksiyon 4 ($N=4$)	2000
	F21	Hibrit Fonksiyon 5 ($N=5$)	2100
	F22	Hibrit Fonksiyon 6 ($N=5$)	2200
Bileşik Fonksiyonlar	F23	Kompozit Fonksiyon 1 ($N=5$)	2300
	F24	Kompozit Fonksiyon 2 ($N=3$)	2400
	F25	Kompozit Fonksiyon 3 ($N=3$)	2500
	F26	Kompozit Fonksiyon 4 ($N=5$)	2600
	F27	Kompozit Fonksiyon 5 ($N=5$)	2700
	F28	Kompozit Fonksiyon 6 ($N=5$)	2800
	F29	Kompozit Fonksiyon 7 ($N=3$)	2900
	F30	Kompozit Fonksiyon 8 ($N=3$)	3000

Arama Menzili : [-100,100]^D

2.4.2. CEC 2019 Ölçüt Seti

CEC 2019, 2019 yılında düzenlenen bir CEC yarışmasının bir parçası olarak oluşturulan, 100 basamak problemi olarak da bilinen bir ölçüt setidir (Price vd., 2018). Bu ölçüt seti, karmaşık optimizasyon problemlerini temsil eden 10 adet test fonksiyonundan oluşur. Tablo 2.3'te CEC 2019 fonksiyonları listelenmiştir (Yavuz, 2021).

Tablo 2.3: CEC 2019 Ölçüt Seti Fonksiyonları

No	Fonksiyonlar	Boyut	Aralık
1	Storn's Chebyshev Polinom Uydurma Problemi	9	[-8192, 8192]
2	Ters Hilbert Matris Problemi	16	[-16384, 16384]
3	Lennard-Jones Minimum Enerji Kümesi	18	[-4, 4]
4	Rastrigin Fonksiyonu	10	[-100, 100]
5	Griewangk Fonksiyonu	10	[-100, 100]
6	Weierstrass Fonksiyon	10	[-100, 100]
7	Modifiye Schwefel Fonksiyonu	10	[-100, 100]
8	Genişletilmiş Schaffer F6 Fonksiyonu	10	[-100, 100]
9	Mutlu Kedi Fonksiyonu	10	[-100, 100]
10	Ackley Fonksiyon	10	[-100, 100]

2.5. PARAMETRE BELİRLEME KÜTÜPHANESİ

2.5.1. İrace

Optimizasyon algoritmaları kontrol parametrelerine sahiptir. Bu parametreler algoritma performansını etkilemektedir (Yang vd., 2013). Önerilen algoritma da kontrol parametrelerine sahiptir ve bu parametreler algoritmanın performansını doğrudan etkilemektedir. Dolayısıyla bu parametre değerlerinin doğru seçilmesi uygulamanın performansını gerçekçi bir şekilde yansıtması için önemlidir (Kumar vd., 2022). Bu sebeple deneyde kontrol parametrelerini belirlemek için bir otomatik yapılandırma aracı olan irace (iterative racing) kullanılmıştır (López-Ibáñez vd., 2016). İrace, bir R dili paketidir. İrace, parametre ayarlaması için F-Race adında bir yarışma tabanlı bir yaklaşım kullanır. Bu yaklaşım, bir dizi aday parametre konfigürasyonunu birbiriyle karşılaştırarak en iyi konfigürasyonu belirlemeyi amaçlar. İrace, iteratif bir şekilde aday parametre konfigürasyonlarını oluşturur, bu konfigürasyonları değerlendirir ve en iyi sonuçları veren konfigürasyonları bir sonraki iterasyona taşır. Bu işlem, birçok iterasyon sonunda en iyi parametre ayarlarını elde etmek için tekrarlanır.

F-Race, temel olarak Friedman'ın parametrik olmayan iki yönlü varyans analiz testini kullanan bir yöntemdir (López-Ibáñez vd., 2016). İteratif şekilde çalıştığı için ilk tur için, kendisine verilen test veri seti üzerinde aday yapılandırmalar üretir. Bunu kendisine verilmiş olan algoritma üzerinde çalıştırır. İstatistiksel testler ile bu aday yapılandırmaların en iyisini seçer. Kötü olan yapılandırmaları ise deneme kümesinden çıkartır. Kötü olanların belirlenmesi ve deneme kümesinden çıkarmak için Friedman'ın parametrik olmayan iki yönlü varyans analiz testi kullanılır. Böylece ilk tur bitmiş olur. İkinci turda ise ilk turda birinci gelen yapılandırmaları kullanarak yeni yapılandırmalar üretir ve yine en iyilerini alarak kötü olanları deneme kümesinden çıkartır. Bu işleyiş F-Race çalıştırılma bütçesi bitene kadar devam eder. Böylece irace tüm bunlar bittiğinde algoritmaya ait en iyi yapılandırma değerlerini kullanıcılarına sunar.

Deneysel çalışma 1 ve deneysel çalışma 2'de kullanılan algoritma parametreleri Tablo 2.4'te verilmiştir.

Tablo 2.4: Deneysel Çalışmalarda Önerilen Algoritmanın İrace Parametre Aracı ile Belirlenmiş Olan Kontrol Parametreleri

Parametre	Değer	Değer Aralığı
-asize	200	(25, 50, 100, 200, 300)
-CR	0,4	(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
-ttunea	9,7906	(1, 10)
-ttuneb	1,975	(1, 5)
-ttunec	0,6167	(0.1, 1)
-ttuned	3,1501	(1, 4)
-ttunee	-17,4133	(-20, -6)
-ttunef	-16,9353	(-20, -6)
-ttuneg	-16,9383	(-20, -6)
-lsItr	12	(3, 100)
-tr(trainRatio)	0,05	(0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3)
-cmar	0,4	(0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5)
-ftol	-4,6961	(-14, 0)



ÜÇÜNCÜ BÖLÜM
KENDİNDEN UYARLAMALI BİR DİFERANSİYEL GELİŞİM ALGORİTMASI

3.1. DENEYSEL ÇALIŞMA 1

Bu kısımda, önerilen algoritmanın sadece mutasyon evresine uygulanan mutasyon stratejisi değişikliği detaylandırılarak anlatılacaktır. Bu kısımda önerilen algoritmanın sözde kodu Tablo 3.1’de verilmiştir.

3.1.1. Kendinden Uyarlamalı Mutasyon Operatörü Stratejisi

DG’nin performansı, sahip olduğu mutasyon, çaprazlama ve seçme adımlarına bağlıdır. Bu adımlar birbirini takip eden adımlar olduğu için özellikle mutasyon adımının başarısı diğerlerini doğrudan etkilemektedir. Bu nedenle, burada seçilen mutasyon operatörü algoritmanın en önemli safhasını oluşturmaktadır. Bu çalışmada, literatürde yer alan yaklaşımlar tek mutasyon operatöründen veya birkaç mutasyon operatöründen oluşan yaklaşımın aksine terim bazlı oluşturulan kendinden uyarlamalı bir yöntem tercih edilmiştir.

Bu stratejide, terim bazlı olarak operatörler oluşturularak mutasyon havuzuna denklem doldurulmaktadır. Bu operatörler, Tablo 3.2 (Aydın vd., 2017; Yavuz ve Aydın, 2019), şablonu kullanılarak algoritma başlangıcında rastgele üretilmektedir. Bu şablon kullanılarak üretilen denklemler bir ile dört terimli olabilmektedir. Her bir terim Tablo 3.3’ten (Aydın vd., 2017) ilgili terim sütunundaki seçeneklerden bir tanesi rastgele olarak seçilmektedir. Bu denklem üretilme işlemi, algoritma başlangıcında belirtilen denklem sayısı kadar yapılır. Üretilen her bir denkleme karşılık, denklemin kalitesini gösteren bir başarı sayısı atanır. Tablo 3.3’teki kısaltmaların açıklamaları Tablo 3.4’te verilmiştir. Mutasyon stratejisi sözde kodu Tablo 3.5’te gösterilmiştir.

Tablo 3.1: Deneysel Çalışma 1’in Sözde Kodu

```

1: Algoritma Parametrelerini Belirle (NP, maxFes, MPS)
2: Denklem havuzunu rastgele denklemler ile doldur
3: OldIndividualArchive ← 1
4: equationCounter ← 1
5: while Sonlandırma Kriteri Karşılanmaz İse do
6:   for i = 1 to NP do
7:     KendindenUyarlamalıMutasyonStratejisiniUygula(CR, i, equationCounter)
8:     EquCount ← equationCounter + 1
9:     if equationCounter == MPS then
10:    Başarı oranlarına göre denklemleri sırala
11:    Denklem havuzunun boyutunu küçült.
12:    equationCounter ← 1
13:   if OldIndividualArchive > NP then
14:     En iyi eski bireydeki çözümleri amaç fonksiyon değerlerine göre sırala
15:     İlk NP kadarını sakla. Geri kalan bireyleri havuzdan çıkar
16:   iter ← iter + 1

```

Tablo 3.1 sözde kodundaki “equationCounter” havuzdaki denklemlerin numaralarına karşılık gelmektedir.

Tablo 3.2: Mutasyon Operatörü Üretme Şablonu

1: $X_{b,j} = \text{terim}_1 + \text{terim}_2 + \text{terim}_3 + \text{terim}_4$

Tablo 3.3: Algoritmanın Genelleştirilmiş Arama Denklemindeki Her Bir Bileşen için Alternatif Seçenekler

terim ₁	terim ₂	terim ₃	terim ₄
$X_{i,j}$	$\text{rand}(-1, 1)(X_{best,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{best,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{best,j} - X_{i,j})$
$X_{best,j}$	$\text{rand}(-1, 1)(X_{best,j} - X_{r1,j})$	$\text{rand}(-1, 1)(X_{best,j} - X_{r1,j})$	$\text{rand}(-1, 1)(X_{best,j} - X_{r1,j})$
$X_{r1,j}$	$1)(X_{best,j} - X_{wo,j}) \text{rand}(-1, 1)(X_{r1,j} - X_{r2,j})$	$1)(X_{best,j} - X_{wo,j}) \text{rand}(-1, 1)(X_{r1,j} - X_{r2,j})$	$\text{rand}(-1, 1)(X_{best,j} - X_{wo,j})$
	$\text{rand}(-1, 1)(X_{r1,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{r1,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{r1,j} - X_{i,j})$
	$\text{rand}(-1, 1)(X_{pbestTop,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{pbestTop,j} - X_{i,j})$	$\text{rand}(-1, 1)(X_{pbestTop,j} - X_{i,j})$
0	0	0	0

Tablo 3.4: Tablo 3.3'teki Kısaltmaların Açıklamaları

Kısaltma	Tanım
i	Popülasyondaki i. çözümü
j	Problemin j. boyutu
X_i	Mevcut birey
X_{best}	Popülasyonun en iyi bireyi
X_{r1}, X_{r2}	Popülasyondan rastgele seçilmiş olan bireyler ($r1 \neq r2$)
X_{wo}	Popülasyonun en kötü bireyi
$X_{pbestTop}$	En iyi p kadar birey içinden rastgele seçilmiş birey
0	Terim seçilmedi

Evrimsel süreç boyunca her iterasyonda denklem havuzundan sırası ile rastgele üretilmiş olan bir denklem alınır ve mutasyon operatörü olarak kullanılır. Eğer ki bu denklem mevcut bireyi iyileştirmede başarılı olursa denklemin başarı değeri artırılır. Denklem havuzundaki bütün denklemler kullanıldıktan sonra, havuz içindeki denklemler başarı değerine göre sıralanır ve başarısız olan denklemler havuzdan çıkarılır. Böylelikle denklem havuzunun boyutu MPS küçültülür. Çalıştırma bütçesinin sonuna doğru denklem havuzunda çok az sayıda başarılı denklem kalır. Mutasyon denkleminin başarılı, başarılı olan güncellemesi ile hesaplanmaktadır.

$$MPS = \frac{MPS^2}{itr_{MAX}} \quad (2.6)$$

$$MAXFES = itr_{MAX} \times NP \quad (2.7)$$

itr_{MAX} maksimum iterasyon sayısını göstermektedir. $MAXFES$ maksimum

fonksiyon çağrım sayısıdır.

Tablo 3.5: Önerilen Algoritmanın Mutasyon Stratejisi Sözde Kodu

```

1: procedure KendindenUyarlamalıMutasyonStratejisiniUygula(CR,i,EquCount)
2: for d=1 to Dim do
    $u_{i,j}^t = \begin{cases} MP(EquCount) & \text{if } (rand \leq CR \parallel d = rand[1, D]) \\ 0 & \text{diğer durumda} \end{cases}$ 
3: if  $f(u^t) \leq f(x^t)$  then
4:    $X_i^{t+1} = u_i^t$ 
5:    $X_i^t \rightarrow OldIndividualArchive$ 
6:   MP (EquCount) başarı artırma
7: else
8:    $X_i^{t+1} = X_i^t$ 

```

Tablo 3.5 sözde kodunda MP denklem havuzuna karşılık gelmektedir.

3.2. DENEYSEL ÇALIŞMA 2

Deneysel çalışma 1’de DG algoritmasının mutasyon evresine yapılan değişiklik gösterilerek önerilen algoritma anlatılmıştı. Deneysel Çalışma 2’de ise buna ek olarak önerilen algoritmaya yeni bir yaklaşım daha eklenmektedir. Bu yaklaşım rekabetçi yerel arama stratejisi yaklaşımıdır. Deneysel çalışma 2’nin sözde kodu Tablo 3.6’da verilmiştir. Deneysel çalışma 2 ile önerilen algoritma nihai haline ulaşmıştır. Nihai haline ulaşan algoritmaya Kendinden Uyarlamalı Yerel Aramalı Diferansiyel Gelişim Algoritması (KUYA-DGA) adı verilmiştir.

3.2.1. Rekabetçi Yerel Arama Stratejisi

Önerilen algoritmanın faydalanma yeteneklerini güçlendirmek için yerel arama ile hibritleştirilmesi düşünülmüştür. Farklı problem türlerine göre yerel aramalar yetersiz gelebilmektedir. Bunun için tek bir tane yerel arama yöntemi yerine birden fazla yerel aramanın yarıştırdığı ve kazananın galip geldiği Rekabetçi Yerel Arama Stratejisi (Competitive Local Search Strategy) algoritmaya dahil edilmiştir. Bu yaklaşım daha önce ABC algoritmasına MTSL1 ve IPOPCMAES (Hansen vd., 2003) tekniklerinin yarıştırdılması ile dahil edilmiştir (Durmuş vd., 2019; Yavuz vd., 2016; Yavuz ve Aydın, 2019). Bu çalışmada önerilen algoritmaya Brent’in tekniğini kullanan (Brent, 2013) Powel’in Eşlenik Talimatları (Powell’s Conjugate Directions) methodu (Michael J. D. Powell, 1964) ile Kovaryans Matrisi Uyarlamalı Gelişim Stratejisi (Evolution Strategy With Covariance Matrix Adaptation (IPOPCMAES) yerel aramalarının yarıştırdığı Rekabetçi Yerel Arama Stratejisi eklenmiştir. Bu yaklaşıma ait sözde kod Tablo 3.7’de

verilmiştir.

Bu stratejiye göre algoritma ilk olarak yerel arama olmadan çalıştırılmaktadır. Daha sonra algoritma yarışma safhasına geçer. O andaki popülasyonun en iyi bireyi alınır ve yerel arama yöntemleri en başta belirlenen bütçe miktarı kadar çalıştırılırlar. Eğer ki yerel arama yöntemleri kendisine verilen bireyi iyileştirirler ise en iyi iyileştirmeyi yapan yerel arama yöntemi ile algoritma yoluna devam eder. Eğer ki iyileştirme meydana gelmezse algoritma bir yerel arama yöntemi olmadan evrimsel surecine devam eder. Bu işlem her iterasyonda gerçekleşmemektedir. Algoritma kendisini durgunluğa girdiğini düşündüğü anda yani ona verilen bireyin değerlerini iyileştiremez duruma geldiği anda bu safhaya girmektedir.

Ayrıca KUYA-DGA algoritmasının çalışması başlangıcında, eski birey arşivi adında (Old Individual Archive) bir arşiv oluşturulur. Bu arşivin oluşturulma amacı eski bireylerin bilgilerini burada saklamaktır. KUYA-DGA algoritması mutasyon havuzundan seçilmiş mutasyon operatörü ile yeni bireyi (mutant birey) üretir. Eğer elde edilen mutant birey, eski bireye göre daha iyi bir amaç fonksiyon değerine sahip ise eski birey ile yer değiştirilir. Bu adımda eski bireyin bilgileri atılmaz ve Tablo 3.6'daki "OldIndividualArchive" arşivine kaydedilir (Yavuz, 2022; Zhang ve Sanderson, 2009). Buradaki amaç eski birey bilgilerinin kaybedilmesinin önüne geçmektir. Eski bireylerin bilgileri, kullanılacak terim Tablo 3.3'ten seçilmiş ise kullanılırlar. Arşiv doldukça amaç fonksiyon değeri kötü olanlar arşivden çıkarılarak arşivin kötü bireyler ile doldurulmasının önüne geçilir.

Tablo 3.6: KUYA-DGA Sözde Kodu

```

1: Algoritma Parametrelerini Belirle (NP, trainRatio, maxFes, MPS)
2: Denklem havuzunu rastgele denklemler ile doldur
3: OldIndividualArchive (eski birey arşivi) ← 1
4: equationCounter ← 1
5: isCompetitionNeeded = True
6: while Sonlandırma kriteri karşılanmaz ise do
7:   for i = 1 to NP do
8:     KendindenUyarlamaliMutasyonStratejisiniUygula(CR, i, equationCounter)
9:   EquCount ← equationCounter + 1
10:  RekabetçiYerelAramaUygula(Xbest)
11:  if equationCounter == MPS then
12:    Havuzdaki denklemleri başarı oranlarına göre sırala
13:    MP havuzunun boyutunu küçült.
14:    equationCounter ← 1
15:  if OldIndividualArchive > NP then
16:    Eski birey arşivindeki çözümleri amaç fonksiyon değerlerine göre sırala
17:    İlk NP kadarını sakla. Geri kalan bireyleri havuzdan çıkar
18:  iter ← iter + 1

```

Tablo 3.7: Rekabetçi Yerel Arama Stratejisi Sözde Kodu

```

1: procedure RekabetçiYerelAramaUygula ( $X_{best}$ )
2:   if isCompetitionNeeded == True & currentFES  $\geq$  trainRatio * MAXFES then
3:     selectedLS = NULL
4:      $X_{imp}^{IPOPMAES}$  = applyIPOPMAES(X, trainRatio * 0.5 * MAXFES)
5:      $X_{imp}^{POWELL}$  = applyPOWELL(X, trainRatio * 0.5 * MAXFES)
6:      $X_{imp}^{NoLocal}$  =  $X_{best}$ 
7:      $ls = argmin(X_{imp}^{IPOPMAES}, X_{imp}^{POWELL}, X_{imp}^{NoLocal})$ 
8:     if  $ls == 1$  then
9:       selectedLS = IPOPMAES
10:    else if  $ls == 2$  then
11:      selectedLS = POWELL
12:    else if  $ls == 3$  then
13:      selectedLS = NULL
14:    if selectedLS  $\neq$  NULL then
15:      isCompetitionNeeded = False
16:    if isCompetitionNeeded = False & selectedLS  $\neq$  NULL then
17:      selectedLS yerel aramasını  $X_{best}$  'e uygula
18:    if  $X_{best}$  iyileştirilmez ise then
19:      isCompetitionNeeded = True

```

3.2.2. IPOPMAES

Hansen ve Ostermeier, evrimsel stratejiye derandomizasyon ve kümülasyon adında iki yararlı yöntem ile kendinden uyarlamalı bir mutasyon dağılımı elde etmişlerdir (Hansen ve Ostermeier, 2001). Kovaryans Matris Adaptasyonu kullanarak önerdikleri bu yaklaşım CMA-ES'tir. Fakat CMA-ES ancak küçük popülasyonlarda kovaryans matrisini güvenilir bir şekilde uyarlayabildiğinden büyük popülasyonlarda verimli bir şekilde kullanılamaz. Problem boyutu belirli bir eşiği geçtiğinde adaptasyon süresi CMA-ES'in performansında sınırlayıcı faktör olmaya başlar (Hansen vd., 2003). Bu da CMA-ES'in sadece mükemmel ölçeklendirilmiş fonksiyonlarda kayda değer bir performans artışı sağlaması demektir (Hansen ve Ostermeier, 2001).

Hansen vd. optimuma yakınsama için gereken nesil sayısını azaltmayı amaçlayan kovaryans matris adaptasyonlu (CMA-ES) yeni bir evrimsel strateji ortaya atmışlardır (Hansen vd., 2003). Çünkü nesil sayısının azaltılması diğer bir deyişle algoritmanın zaman karmaşıklığının azaltılması büyük bir popülasyon boyutu söz konusu olduğunda çok önemlidir. Yeni bu evrimsel stratejinin adı ise IPOPMAES'tir.

IPOPMAES'teki amaç daha büyük popülasyonlarda bulunan daha fazla bilgiden yararlanarak kovaryans matrisini uyarlamaktır. Bu sebeple Hansen vd. kovaryans matrisini birinci dereceden bilgilerle güncellemek yerine daha yüksek dereceden bilgileri de dahil ederek kovaryans matrisinin değişimini (önerilen mutasyon

dağılım şeklinin değişimini) sağlamışlardır. Sonuç olarak orijinal CMA-ES yaklaşımından daha az nesil sayısı ile yapılabilen bir optimizasyon tekniği sunulmuştur. Ayrıca bu teknik paralel makinelerde de kullanıma uygun olacak şekilde tasarlanmıştır.

3.2.3. POWELL Metodu

Powell'in metodu bir fonksiyonun yerel minimumunu bulmak için önerilen bir yöntemdir. Bu yöntemi farklı kılan şey ise fonksiyonun türevlenebilir olmadan da minimumunun bulunabilmesidir. Dolayısıyla bu yöntemde fonksiyonun türevi de alınmaz (Powell, 1964; Powell, 1977).

Bu yöntemde fonksiyon üzerinde çift yönlü bir arama söz konusudur. Her bir arama vektörü boyunca çift yönlü bir arama yapılarak fonksiyon minimize edilir. Algoritma, önemli ve kayda değer bir iyileştirme yapılmayana kadar isteğe bağlı sayıda yinelenir (Gao vd., 2013).



DÖRDÜNCÜ BÖLÜM
BULGULAR VE TARTIŞMA

4.1. PERFORMANS DEĞERLENDİRME METRİKLERİ

Bir optimizasyon algoritması tasarlanıp test edildikten sonra bazı metrikler kullanılarak algoritma performansının değerlendirilmesi gerekmektedir. Ancak bu şekilde tasarlanan algoritmanın test sonuçları anlam kazanır ve diğer algoritmalar ile karşılaştırılabilir hale gelir. Sayısal veri söz konusu olduğunda verinin yeterince anlamlı ve özetlenebilir olması için iki özelliğinin belirtilmiş olması gerekmektedir. Bunlardan bir tanesi merkezi dağılım diğeri ise yayılma kıstasıdır. Merkezi dağılım başlığı altında aritmetik ortalama, yayılma kıstası başlığı altında da standart sapma örnek verilebilir.

4.1.1. Aritmetik Ortalama

Aritmetik ortalama merkezi dağılım ölçütü için sıkça kullanılmaktadır. Aritmetik ortalama bir veri kümesindeki bütün verilerin veri sayısına bölünmesi ile bulunur. Aritmetik ortalama formülü Denklem 4.1 de verilmiştir.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.1)$$

x_i veri kümesindeki i . veriyi(sayıyı) ifade ederken n ise veri sayısını ifade eder. \bar{x} ise aritmetik ortalamadır.

4.1.2. Standart Sapma

Standart sapma veri kümesinde olan sayıların birbiri ile olan uyumluluğunu ölçer. Standart sapma bir yaygınlık kıstası metriğidir. Veri gurubundaki sayıların, aritmetik ortalamaya göre olan yayılımını verir ve birbirleriyle olan tutarlılığını gösterir. Standart sapmanın formülü Denklem 4.2’de verilmiştir.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4.2)$$

x_i veri kümesindeki i . veriyi(sayıyı) ifade ederken n ise veri sayısını ifade eder. \bar{x} aritmetik ortalama değeri , σ ise standart sapmadır.

4.2. ÖNERİLEN ALGORİTMANIN PERFORMANS SONUÇLARI

Bu çalışmada kendinden uyarlamalı bir diferansiyel gelişim algoritması önerilmiştir. Önerilen bu algoritma bir DG algoritmasının aşamalarına yapılan değişiklikler ile bir DG varyantıdır. Yapılan değişiklikler Deneysel Çalışma 1 ve Deneysel Çalışma 2

olarak materyal ve metot bölümünde anlatılmıştır. Bu kısımda ise hem Deneysel Çalışma 1 kapsamında yapılan değişikliklerin CEC 2019 ölçüt seti sonuçları ve literatürde yer alan diğer algoritmalar ile olan karşılaştırma sonuçları hem de Deneysel Çalışma 2 kapsamında yapılan değişikliklerin CEC 2014 ölçüt seti sonuçları ve literatürde yer alan diğer algoritmalar ile karşılaştırma sonuçları verilmiştir.

4.2.1. Deneysel Çalışma 1' in CEC 2019 Sonuçları ve Literatür Karşılaştırması

Deneysel Çalışma 1 kapsamında DG algoritmasının mutasyon evresinde değişiklik yapılmıştır. Yapılan bu değişiklik CEC 2019 ölçüt setinde çalıştırılmış ve sonuçları elde edilmiştir. Bu sonuçlar Chakraborty vd. yapmış olduğu çalışma sonuçları ile karşılaştırılmış ve deneylere dahil edilmiştir. Chakraborty vd. önerdikleri Genişletilmiş Balina Optimizasyon Algoritması'nı (WOAmM) çeşitli meta sezgisel algoritmalar ile karşılaştırmıştır (Chakraborty vd., 2021). Bunlar Balina Optimizasyon Algoritması (WOA), Güve Alevi Optimizasyonu (MFO), Kelebek Optimizasyon Algoritması (BOA), Sinüs Kosinüs Algoritması (SCA) ve JAYA algoritmasıdır. Deneysel Çalışma 1 kapsamında yapılan değişiklik ile Chakraborty vd. yapmış olduğu çalışma karşılaştırılırken adil bir karşılaştırma olması için Chakraborty vd. kullanmış olduğu parametreler kullanılmıştır. Bu parametreler popülasyon boyutu 50 ve iterasyonu 1000 olacak şekilde CEC 2019 problemlerinde çalıştırılmıştır. Önerilen algoritmanın parametreleri Çaprazlama Oranı 0.4 ve Denklem Havuzu Boyutu 200 olacak şekilde seçilerek, CEC 2019 ölçüt setinin her fonksiyonu için 30 defa çözülmüştür. Ortalama, standart sapma ve en iyi değerleri hesaplanmıştır. Bu hesaplamalar Intel Xeon E5 2670 işlemci ve 64 GB hafızaya sahip bilgisayarda yapılmıştır. DG algoritmasına deneysel çalışma 1 kapsamında yapılan değişikliklerin CEC 2019 ölçüt seti üzerinde alınan sonuçlara Tablo 4.1'de, Chakraborty vd. yapmış olduğu çalışma sonuçları ile karşılaştırılması ise Tablo 4.2'de verilmiştir.

Tablo 4.1: Önerilen Algoritmanın CEC 2019 Sonuçları

No	Ortalama	Standart Sapma	En İyi
F1	1.77×10^6	1.93×10^6	1.50×10^{-1}
F2	4.17×10^3	9.59×10^2	2.12×10^3
F3	4.01	1.6	1.49
F4	2.57×10^1	4.05	1.77×10^1
F5	7.16×10^{-1}	2.06×10^{-1}	3.26×10^{-2}
F6	3.44	8.64×10^{-1}	6.50×10^{-1}
F7	7.78×10^2	1.62×10^2	3.69×10^2
F8	2.99	2.33×10^{-1}	2.49
F9	3.26×10^{-1}	6.29×10^{-2}	1.31×10^{-1}
F10	2.01×10^1	4.24×10^{-2}	2.00×10^1

Tablo 4.1’de görüldüğü üzere Deneysel Çalışma 1 kapsamında önerilen algoritma, CEC 2019 ölçüt setindeki F1, F2, F3, F4, F5, F6, F7, F8, F9, F10 fonksiyonlarının her biri için 30 defa çalıştırılmış. Bunun sonucunda alınan ortalama, standart sapma ve en iyi metrik sonuçları bu tabloda gösterilmiştir. En iyi değeri bir fonksiyonun 30 defa çalıştırılma sonucunda elde edilen 30 sonuçtan en iyisi seçilerek yazılmıştır.

Tablo 4.2: Deneysel Çalışma 1 Sonuçları ve Literatürdeki Çalışmalar ile Karşılaştırılması

No	Önerilen	WOAmM	WOA	MFO	BOA	SCA	JAYA
F1	Ortalama	1.77x10 ⁶	1	3.35x10 ⁶	1	1	8.39x10 ⁶
	Std. Sapma	1.93x10 ⁶	≈0	3.94x10 ⁶	0	4.13x10 ⁻⁸	3.12x10 ⁶
	En İyi	1.50x10⁻¹	1	4.54x10 ²	1	1	3.87x10 ⁶
F2	Ortalama	4.17x10 ³	4.31	7.71x10 ³	5	5.01	5.35x10 ³
	Std. Sapma	9.59x10 ²	9.04x10 ⁻²	2.73x10 ³	0	3.47x10 ⁻³	9.02x10 ²
	En İyi	2.12x10 ³	4.23	3.56x10 ³	5	5	3.71x10 ³
F3	Ortalama	4.01	2.88	4.11	7.43	4.77	9.24
	Std. Sapma	1.6	1.11	1.82	5.02x10 ⁻²	9.38x10 ⁻¹	1.45
	En İyi	1.49	1.41	1.43	7.33	3.28	4.41
F4	Ortalama	2.57x10¹	3.42x10 ¹	5.04x10 ¹	7.80x10 ¹	9.88x10 ¹	7.52x10 ¹
	Std. Sapma	4.05	8.42	2.09x10 ¹	7.84	1.17x10 ¹	1.52x10 ¹
	En İyi	1.77x10 ¹	1.89x10 ¹	1.30x10 ¹	7.11x10 ¹	7.85x10 ¹	4.69x10 ¹
F5	Ortalama	7.16x10⁻¹	1.60	2.05	5.48x10 ¹	1.18x10 ²	1.49x10 ¹
	Std. Sapma	2.06x10⁻¹	2.48x10 ⁻¹	4.55x10 ⁻¹	1.09x10 ¹	2.19x10 ¹	9.61
	En İyi	3.26x10⁻²	1.25	1.45	3.79x10 ¹	7.42x10 ¹	4.33
F6	Ortalama	3.44	5.55	8.13	1.05x10 ¹	1.12x10 ¹	8.62
	Std. Sapma	8.64x10 ⁻¹	1.56	1.83	3.87x10 ⁻¹	1.22	1.54
	En İyi	6.50x10⁻¹	3.33	3.85	1.04x10 ¹	6.66	6.54
F7	Ortalama	7.78x10²	1.14x10 ³	1.30x10 ³	1.69x10 ³	2.81x10 ³	2.6x10 ³
	Std. Sapma	1.62x10²	2.62x10 ²	3.5x10 ²	1.81x10 ²	2.07x10 ²	3.88x10 ²
	En İyi	3.69x10²	4.9x10 ²	5.42x10 ²	1.59x10 ³	1.65x10 ³	1.44x10 ³
F8	Ortalama	2.99	4.19	4.53	4.49	4.96	5.16
	Std. Sapma	2.33x10 ⁻¹	3.67x10 ⁻¹	3.29x10 ⁻¹	4.66x10 ⁻⁴	2.08x10 ⁻¹	1.57x10 ⁻¹
	En İyi	2.49	3.23	3.93	4.49	4.42	4.74
F9	Ortalama	3.26x10⁻¹	1.34	1.36	2.32	4.52	1.93
	Std. Sapma	6.29x10⁻²	1.2x10 ⁻¹	1.78x10 ⁻¹	9.3x10 ⁻¹	4.62x10 ⁻¹	2.15x10 ⁻¹
	En İyi	1.31x10⁻¹	1.06	1.11	1.52	3.23	1.48
F10	Ortalama	2.01x10¹	2.11x10 ¹	2.12x10 ¹	2.13x10 ¹	2.15x10 ¹	2.20x10 ¹
	Std. Sapma	4.24x10⁻²	7.36x10 ²	1.05x10 ⁻¹	8.55x10 ⁻²	8.73x10 ⁻²	1.78x10 ⁻¹
	En İyi	2.00x10¹	2.1x10 ¹	2.1x10 ¹	2.13x10 ¹	2.13x10 ¹	2.16x10 ¹

Önerilen algoritmanın rakiplerinden başarılı olan değerleri koyu olarak yazılmıştır.

Tablo 4.2’de de görüldüğü üzere Deneysel Çalışma 1 kapsamında önerilen algoritma, CEC 2019 ölçüt setinin F4, F5, F6, F7, F8, F9, F10 fonksiyonlarında, ortalama sonuçlar baz alındığında WOAmM, WOA, MFO, BOA, SCA, JAYA algoritmalarını geride bırakarak birinci sırada yer almıştır. En iyi sonuçlarda ise F1, F5, F6, F7, F8, F9, F10 fonksiyonlarında rakiplerini geride bırakarak birinci sırada yer almıştır. Standart sapma değerlerine bakıldığında sonuçların tutarlı olduğu görülmüştür. Sonuç olarak en iyi ve ortalama sonuçlarda toplamda CEC 2019 on test fonksiyonunun yedisinde, önerilen algoritma birinci gelmiştir.

4.2.2. Deneysel Çalışma 2'nin CEC 2014 Sonuçları ve Literatür Karşılaştırması

Deneysel Çalışma 1'de yapılan değişikliklere artı olarak önerilen algoritmaya Deneysel Çalışma 2'de rekabetçi yerel arama stratejisi eklenmiştir. Algoritmanın faydalanma yeteneklerini güçlendirmek için eklenen bu yerel arama stratejisi birden fazla yerel arama stratejisininin yarıştırılması mantığına dayanmaktadır. Deneysel Çalışma 2'de yapılan bu değişiklikle nihai haline ulaşan önerilen algoritmaya KUYA-DGA adı verilmiştir. KUYA-DGA, CEC 2014 ölçüt seti ile çalıştırılmış ve sonuçları elde edilmiştir. Bu sonuçlar literatürdeki DG varyantları olan SHADE, L-SHADE, PADE, LPalmDE algoritmaları ile karşılaştırılmış ve deneylere dahil edilmiştir. Deneyler 30 ve 50 boyut için gerçekleştirilmiştir. Önerilen algoritmanın 30 ve 50 boyutta sonuçları literatürdeki DG varyantlarının 30 ve 50 boyuttaki sonuçları ile birleştirilip sırasıyla Tablo 4.3 ve Tablo 4.4 'de verilmiştir. Deneylerde algoritma karşılaştırmanın adil yapılabilmesi için popülasyon boyut değeri her algoritma için aynı olarak seçilmiştir. Önerilen algoritmanın geri kalan kontrol parametreleri, irace parametre yapılandırma aracı kullanılarak belirlenmiştir.

Tablo 4.3: Deneysel Çalışma 2'nin 30 Boyutta CEC 2014 Sonuçları ve DG Varyantları ile Karşılaştırılması

No	KUYA-DGA		SHADE		L-SHADE		PADE		LPalmDE	
	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.
F1	1,000E-08	1,090E-16	4,309E+02	1,529E+03	4,592E+02	9,091E+02	6,695E+02	1,271E+03	3,609E+02	5,289E+02
F2	1,000E-08	1,146E-16	1,000E-08	0,000E+00	1,000E-08	0,000E+00	1,000E-08	0,000E+00	1,000E-08	0,000E+00
F3	1,000E-08	1,063E-16	1,000E-08	2,189E-27	1,000E-08	3,925E-27	1,000E-08	1,260E-27	1,000E-08	1,919E-27
F4	1,000E-08	1,015E-16	1,000E-08	2,424E-29	1,243E+00	8,790E+00	1,243E+00	8,790E+00	2,486E+00	1,231E+01
F5	2,007E+01	5,997E-02	2,010E+01	1,823E-02	2,001E+01	3,409E-03	2,003E+01	1,541E-02	2,000E+01	2,667E-03
F6	1,000E-08	0,000E+00	5,691E-01	1,324E+00	2,641E+00	2,520E+00	2,020E+00	1,578E+00	1,282E+00	1,097E+00
F7	1,000E-08	1,554E-16	1,000E-08	0,000E+00	2,317E-03	5,688E-03	2,366E-03	5,315E-03	1,063E-03	3,057E-03
F8	3,726E+00	1,693E+00	1,000E-08	0,000E+00	1,000E-08	0,000E+00	1,000E-08	2,463E-16	1,000E-08	4,926E-16
F9	3,317E+00	1,656E+00	1,506E+01	3,214E+00	1,311E+01	2,826E+00	1,560E+01	3,248E+00	1,822E+01	4,896E+00
F10	1,753E+02	3,027E+02	2,449E-03	6,708E-03	6,940E-03	1,142E-02	2,449E-03	6,708E-03	8,573E-03	1,552E-02
F11	5,537E+01	7,050E+01	1,494E+03	2,017E+02	1,347E+03	2,451E+02	1,323E+03	3,188E+02	1,824E+03	4,822E+02
F12	2,532E-02	7,539E-02	1,653E-01	2,598E-02	9,032E-02	1,490E-02	1,058E-01	3,531E-02	7,083E-02	2,896E-02
F13	3,258E-02	7,104E-03	2,124E-01	3,383E-02	1,860E-01	3,505E-02	1,793E-01	3,434E-02	1,550E-01	3,677E-02
F14	3,690E-01	5,191E-02	2,639E-01	8,785E-02	2,656E-01	4,884E-02	2,257E-01	4,071E-02	2,400E-01	3,111E-02
F15	2,888E+00	2,948E+00	2,508E+00	3,522E-01	1,983E+00	2,732E-01	2,059E+00	3,858E-01	2,815E+00	6,985E-01
F16	8,405E+00	1,709E+00	9,181E+00	3,212E-01	8,622E+00	4,938E-01	8,718E+00	7,583E-01	9,079E+00	1,037E+00
F17	9,618E+02	2,967E+02	1,175E+03	4,121E+02	1,066E+03	3,659E+02	8,735E+02	3,272E+02	9,642E+02	3,209E+02
F18	6,258E+01	2,682E+01	6,399E+01	3,239E+01	7,299E+01	2,968E+01	5,487E+01	2,485E+01	5,825E+01	2,498E+01
F19	4,370E+00	1,038E+00	4,402E+00	7,783E-01	4,503E+00	8,960E-01	4,195E+00	7,976E-01	3,784E+00	8,428E-01
F20	4,374E+01	3,070E+01	1,386E+01	7,190E+00	1,737E+01	9,520E+00	1,669E+01	8,689E+00	1,327E+01	7,069E+00
F21	3,988E+02	1,571E+02	2,583E+02	9,718E+01	2,827E+02	1,227E+02	2,367E+02	9,899E+01	2,159E+02	8,280E+01
F22	7,881E+01	6,022E+01	1,130E+02	6,578E+01	8,575E+01	7,029E+01	1,215E+02	6,763E+01	1,373E+02	5,074E+01
F23	3,152E+02	6,367E-14	3,152E+02	4,059E-14	3,152E+02	4,059E-14	3,152E+02	4,971E-14	3,152E+02	4,286E-14
F24	2,050E+02	9,688E+00	2,251E+02	1,515E+00	2,306E+02	5,736E+00	2,294E+02	5,654E+00	2,299E+02	5,991E+00
F25	2,026E+02	7,544E-02	2,032E+02	5,749E-01	2,033E+02	5,684E-01	2,036E+02	7,062E-01	2,036E+02	9,801E-01
F26	1,000E+02	8,063E-03	1,002E+02	3,056E-02	1,002E+02	4,346E-02	1,002E+02	3,194E-02	1,021E+02	1,384E+01
F27	3,000E+02	4,135E-14	3,246E+02	4,209E+01	3,373E+02	2,731E+01	3,287E+02	2,693E+01	3,206E+02	2,320E+01
F28	8,525E+02	2,521E+01	8,340E+02	3,285E+01	8,565E+02	3,309E+01	8,795E+02	3,230E+01	8,659E+02	5,672E+01
F29	7,030E+02	8,528E+01	7,257E+02	1,341E+01	7,249E+02	4,066E+01	6,209E+02	2,168E+02	6,232E+02	2,062E+02
F30	2,199E+03	6,943E+02	2,014E+03	8,394E+02	1,781E+03	6,825E+02	1,116E+03	5,047E+02	1,059E+03	4,426E+02

Tablo 4.3: (Devamı) Deneysel Çalışma 2'nin 30 Boyutta CEC 2014 Sonuçları ve DG Varyantları ile Karşılaştırılması

	KUYA-DGA	SHADE	L-SHADE	PADE	LPalmDE
Sıralama	2,30	2,90	3,13	2,67	2,70
Galibiyet		17	19	15	16
Mağlubiyet		8	8	12	11
Berabere		5	3	3	3

Her fonksiyon için en başarılı değerler koyu olarak yazılmıştır. Son üç satırdaki değerler önerilen algoritmanın ilgili sütundaki algoritmaya karşı fonksiyon sayılarında aldığı galibiyet, mağlubiyet ve berabere sayılarıdır.

Tablo 4.4: Deneysel Çalışma 2'nin 50 Boyutta CEC 2014 Sonuçları ve DG Varyantları ile Karşılaştırılması

No	KUYA-DGA		SHADE		L-SHADE		PADE		LPalmDE	
	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.	Ortalama	Std. Sap.
F1	1,000E-08	5,932E-17	1,818E+04	1,263E+04	7,260E+03	5,073E+03	1,023E+04	7,413E+03	1,247E+04	1,682E+04
F2	1,000E-08	8,079E-17	1,000E-08	1,299E-22	1,000E-08	3,439E-17	1,000E-08	2,975E-21	1,000E-08	2,835E-18
F3	1,000E-08	5,170E-17	1,000E-08	1,174E-13	2,919E-08	1,575E-07	4,415E+02	1,605E+03	1,000E-08	4,362E-08
F4	3,763E+01	4,418E+01	3,488E+01	4,673E+01	2,272E+01	4,079E+01	5,459E+00	2,149E+01	3,937E+00	1,903E+01
F5	2,007E+01	5,058E-02	2,014E+01	1,668E-02	2,002E+01	6,085E-03	2,004E+01	1,884E-02	2,000E+01	3,511E-03
F6	1,000E-08	0,000E+00	3,659E+00	1,753E+00	1,248E+01	2,441E+00	1,125E+01	2,942E+00	1,006E+01	2,382E+00
F7	1,000E-08	9,785E-17	9,182E-04	2,844E-03	5,119E-03	6,678E-03	5,841E-03	8,003E-03	1,788E-03	3,782E-03
F8	8,231E+00	2,513E+00	1,000E-08	2,463E-16	1,000E-08	0,000E+00	1,000E-08	0,000E+00	1,000E-08	1,770E-15
F9	9,751E+00	2,586E+00	3,256E+01	4,404E+00	3,042E+01	4,411E+00	3,839E+01	8,175E+00	4,468E+01	8,117E+00
F10	4,447E+02	2,242E+02	1,959E-03	4,543E-03	4,164E-03	6,387E-03	2,449E-03	5,542E-03	1,984E-02	1,528E-02
F11	2,160E+02	1,627E+02	3,371E+03	3,451E+02	3,348E+03	3,320E+02	3,344E+03	5,975E+02	3,954E+03	7,434E+02
F12	1,398E-03	8,415E-04	1,609E-01	2,280E-02	9,195E-02	1,402E-02	9,779E-02	1,863E-02	9,307E-02	5,062E-02
F13	4,930E-02	8,527E-03	3,078E-01	3,537E-02	2,895E-01	3,733E-02	3,096E-01	5,147E-02	2,949E-01	4,949E-02
F14	3,567E-01	4,756E-02	3,403E-01	1,284E-01	3,236E-01	7,384E-02	2,958E-01	7,610E-02	2,969E-01	7,160E-02
F15	4,398E+03	1,034E+03	5,695E+00	6,194E-01	5,332E+00	1,165E+00	5,924E+00	1,261E+00	5,807E+00	1,101E+00
F16	1,616E+01	3,168E+00	1,748E+01	5,291E-01	1,700E+01	5,029E-01	1,656E+01	8,516E-01	1,713E+01	1,457E+00
F17	2,316E+03	4,778E+02	2,432E+03	5,359E+02	2,379E+03	7,093E+02	1,986E+03	1,020E+03	1,988E+03	7,461E+02
F18	1,230E+02	2,048E+01	1,673E+02	4,434E+01	1,726E+02	4,187E+01	1,297E+02	3,174E+01	1,360E+02	2,723E+01
F19	9,218E+00	1,485E+00	9,120E+00	2,257E+00	1,051E+01	2,358E+00	1,170E+01	2,629E+00	9,638E+00	2,107E+00
F20	2,273E+02	9,099E+01	1,839E+02	6,912E+01	1,949E+02	7,189E+01	1,531E+02	5,173E+01	1,500E+02	5,031E+01
F21	1,248E+03	2,232E+02	1,279E+03	3,658E+02	1,219E+03	3,576E+02	9,554E+02	3,839E+02	1,057E+03	3,396E+02
F22	1,290E+02	5,234E+01	3,722E+02	1,087E+02	3,687E+02	1,480E+02	3,944E+02	1,425E+02	4,625E+02	2,037E+02
F23	3,440E+02	5,684E-14	3,440E+02	7,758E-14	3,440E+02	7,839E-14	3,440E+02	4,907E-14	3,440E+02	5,399E-14
F24	2,709E+02	3,796E+00	2,743E+02	1,739E+00	2,797E+02	3,077E+00	2,793E+02	3,469E+00	2,795E+02	3,137E+00
F25	2,051E+02	1,765E-01	2,077E+02	4,716E+00	2,083E+02	5,106E+00	2,131E+02	7,468E+00	2,131E+02	7,683E+00
F26	1,001E+02	1,327E-02	1,023E+02	1,381E+01	1,004E+02	1,139E-01	1,043E+02	1,934E+01	1,101E+02	2,963E+01
F27	3,099E+02	1,631E+01	4,223E+02	4,499E+01	6,182E+02	5,627E+01	5,972E+02	7,484E+01	5,830E+02	5,567E+01
F28	1,205E+03	2,379E+01	1,133E+03	4,500E+01	1,214E+03	5,896E+01	1,457E+03	1,071E+02	1,425E+03	1,004E+02
F29	7,920E+02	3,893E+00	8,840E+02	7,484E+01	8,782E+02	8,772E+01	8,207E+02	1,497E+02	7,938E+02	1,507E+02
F30	6,585E+03	6,226E+02	9,547E+03	6,547E+02	1,006E+04	9,656E+02	1,132E+04	8,222E+02	1,113E+04	6,948E+02

Tablo 4.4: (Devamı) Deneysel Çalışma 2'nin 50 boyutta CEC 2014 sonuçları ve DG varyantları ile karşılaştırılması

	KUYA-DGA	SHADE	L-SHADE	PADE	LPalmDE
Sıralama	2,13	2,93	2,97	3,10	2,90
Galibiyet		19	20	19	18
Mağlubiyet		8	8	9	9
Berabere		3	2	2	3

Her fonksiyon için en başarılı değerler koyu olarak yazılmıştır. Son üç satırdaki değerler önerilen algoritmanın ilgili sütundaki algoritmaya karşı fonksiyon sayılarında aldığı galibiyet, mağlubiyet ve berabere sayılarıdır.

Tablo 4.3'te de görüldüğü üzere Deneysel Çalışma 2 kapsamında önerilen nihai varyant DG algoritması KUYA-DGA, CEC 2014 ölçüt seti ile 30 boyutta çalıştırıldığında rakipleri SHADE, L-SHADE, PADE, LPalmDE algoritmalarına göre 2.30'luk sıralama değeri ile ilk sırada yer almıştır. 2.67'lik ortalama sıralama değeri ile PADE ikinci, 2.70'lik sıralama değeri ile LPalmDE üçüncü, 2.90'lık sıralama değeri ile SHADE dördüncü, 3.13'lük sıralama değeri ile L-SHADE sonuncu sırada yer almıştır. Önerilen algoritma 30 boyutta CEC 2014 ölçüt setinin 17 test fonksiyonunda ilk sırada yer almıştır. Bu test fonksiyonları F1, F2, F3, F4, F6, F7, F9, F11, F12, F13, F16, F22, F23, F24, F25, F26, F27'dir.

Tablo 4.4'te ise 50 boyut sonuçları verilmiştir. Burada da önerilen algoritma 2.13'lük sıralama değeri ile rakiplerini geçerek ilk sırada yer almıştır. 2.90'lık sıralama değeri ile LPalmDE ikinci, 2.93'lük sıralama değeri ile SHADE üçüncü, 2.97'lik sıralama değeri ile L-SHADE dördüncü, 3.10'luk sıralama değeri ile PADE sonuncu sırada yer almıştır. Önerilen algoritma 50 boyutta CEC 2014 ölçüt setinin 19 test fonksiyonunda ilk sırada yer almıştır. Bu test fonksiyonları F1, F2, F3, F6, F7, F9, F11, F12, F13, F16, F18, F22, F23, F24, F25, F26, F27, F29, F30'dur.



BEŞİNCİ BÖLÜM
SONUÇ VE ÖNERİLER

5.1. SONUÇ

Bu tez çalışmasında kendinden uyarlamalı bir diferansiyel gelişim algoritması önerilmiştir. Önerilen DG varyantı bu algoritma deneysel çalışma 1 ve deneysel çalışma 2 olarak iki aşamada gerçekleştirilmiştir. Deneysel çalışma 1’de diferansiyel gelişim algoritmasının mutasyon adımına, mutasyon operatörü için kendinden uyarlamalı terim tabanlı bir yaklaşım uygulanmıştır. Mutasyon adımına yapılan bu değişikliği test etmek için CEC 2019 ölçüt seti fonksiyonları kullanılmıştır. Sonuçlar ortalama, standart sapma ve en iyi metrikleri kullanılarak literatürde yer alan Balina Optimizasyon Algoritması, Genişletilmiş Balina Optimizasyon Algoritması, Güve Alevi Optimizasyon Algoritması, Kelebek Optimizasyon Algoritması, Sinüs Kosinüs Algoritması ve JAYA Algoritması ile karşılaştırılmıştır. Sonuçlara göre önerilen algoritma CEC 2019 ölçüt setinin on test fonksiyonundan yedisinde diğer algoritmalarla karşı en iyi performansı göstermiştir. Deneysel çalışma 2’de ise algoritmanın çözümden faydalanma yeteneğini de güçlendirmek için algoritmaya rekabetçi yerel arama stratejisi de eklenmiştir. Eklenen bu strateji literatürde yer alan güçlü yerel arama stratejilerinden olan Powell metodunun ve IPOPC-MAES’in yarıştırmalarına dayanmaktadır. Nihai haline ulaşan önerilen algoritma Kendinden Uyarlamalı Yerel Aramalı Diferansiyel Gelişim Algoritması (KUYA-DGA) adı verilmiştir. KUYA-DGA, CEC 2014 ölçüt seti fonksiyonlarında önce 30 boyutta sonra ise 50 boyutta test edilmiştir. Sonuçlar ortalama ve standart sapma metrikleri kullanılarak literatürde yer alan kendinden uyarlamalı diferansiyel gelişim algoritmaları ile karşılaştırılmıştır. Bu algoritmalar SHADE, LSHADE, PADE ve LPalmDE ‘dir. Sonuçlara göre önerilen algoritma 30 boyutta CEC 2014 ölçüt setinin otuz test fonksiyonundan on yedisinde diğer algoritmalarla karşı en iyi performansı göstermiştir. 50 boyutta ise CEC 2014 ölçüt setinin otuz test fonksiyonundan on dokuzunda diğer algoritmalarla karşı en iyi performansı göstermiştir. Elde edilen sonuçlara bakıldığında önerilen algoritma diğer kendinden uyarlamalı diferansiyel gelişim algoritmalarına göre en iyi performansı göstererek ilk sırada yer almıştır.

Ayrıca önerilen algoritma, güncel bir mühendislik problemine çözüm sunması açısından EEG sinyallerinde bulunan gürültüleri temizlemede kullanılacak ve test edilecektir. Büyük optimizasyon problemlerinin temelini oluşturan EEG, beyin elektriksel sinyallerini gözlemlemede ve hastalık teşhislerinde kullanılan önemli bir araçtır.

İlerideki çalışmalarda önerilen algoritma farklı ölçüt setleri ve farklı boyutlarda deneyerek, bu kombinasyonların performans sonuçlarına da bakılabilir. Ayrıca önerilen algoritma enerji alanındaki optimizasyon problemlerine uygulanarak performans araştırması yapılabilir.



KAYNAKÇA

- Abualigah, L., Elaziz, M. A., Khasawneh, A. M., Alshinwan, M., Ibrahim, R. A., Alqaness, M. A. A., ... Gandomi, A. H. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. *Neural Computing and Applications*, 34(6), 4081-4110.
- Ahmad, M. F., Isa, N. A. M., Lim, W. H. ve Ang, K. M. (2022). Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy. *Alexandria Engineering Journal*, 61(12), 11835-11858.
- Ahmed, Z. E., Hasan, M. K., Saeed, R. A., Hassan, R., Islam, S., Mokhtar, R. A., ... Akhtaruzzaman, M. (2020). Optimizing energy consumption for cloud internet of things. *Frontiers in Physics*, 8, 1-10.
- Sumathi, R., Alagarsamy, S., Nagaraj, P., & Venkatesh, R. (2022). Detection of Tumor Region in MRI Images Using Kernel Fuzzy C Means with PSO. *2022 International Conference on Computer Communication and Informatics (ICCCI), India*, 1-6
- Arora, S. ve Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3), 715-734.
- Aydın, D., Yavuz, G. ve Stützle, T. (2017). ABC-X: a generalized, automatically configurable artificial bee colony framework. *Swarm Intelligence*, 11(1), 1-38.
- Balakrishnan, H. (2016). Control and optimization algorithms for air transportation systems. *Annual Reviews in Control*, 41, 39-46.
- Beyaz, S., Açııcı, K. ve Sümer, E. (2020). Femoral neck fracture detection in X-ray images using deep learning and genetic algorithm approaches. *Joint Diseases and Related Surgery*, 31(2), 175-183.
- Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4), 353-373.
- Brent, R. P. (2013). *Algorithms for minimization without derivatives*. New York:Dover Publications.
- Brest, J, Maučec, M. S. ve Bošković, B. (2017). Single objective real-parameter

optimization: Algorithm jSO. *2017 IEEE Congress on Evolutionary Computation (CEC), Spain*, 1311-1318.

Brest, Janez, Greiner, S., Boskovic, B., Mernik, M. ve Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6), 646-657.

Chakraborty, S., Saha, A. K., Sharma, S., Mirjalili, S. ve Chakraborty, R. (2021). A novel enhanced whale optimization algorithm for global optimization. *Computers & Industrial Engineering*, 153, 107086.

Das, S. ve Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4-31.

Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y. ve Xu, J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*, 25(7), 5277-5298.

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

Draa, A., Bouzoubia, S. ve Boukhalifa, I. (2015). A sinusoidal differential evolution algorithm for numerical optimisation. *Applied Soft Computing*, 27, 99-126.

Durmuş, B., Yavuz, G. ve Aydın, D. (2019). Adaptive iir filter design using self-adaptive search equation based artificial bee colony algorithm. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(6), 4797-4817.

Elsayed, S. ve Sarker, R. (2016). Differential evolution framework for big data optimization. *Memetic Computing*, 8, 17-33.

Gao, W., Liu, S. ve Huang, L. (2013). A novel artificial bee colony algorithm with powell's method. *Applied Soft Computing*, 13(9), 3763-3775.

Hansen, N. ve Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2), 159-195.

Hansen, Nikolaus, Müller, S. D. ve Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1), 1-18.

- Houssein, E. H., Rezk, H., Fathy, A., Mahdy, M. A. ve Nassef, A. M. (2022). A modified adaptive guided differential evolution algorithm applied to engineering applications. *Engineering Applications of Artificial Intelligence*, 113, 104920.
- Hussain, K., Mohd Salleh, M. N., Cheng, S. ve Shi, Y. (2019). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4), 2191-2233.
- Kumar, A., Biswas, P. P. ve Suganthan, P. N. (2022). Differential evolution with orthogonal array-based initialization and a novel selection strategy. *Swarm and Evolutionary Computation*, 68, 101010.
- Levy, A. B. (2022). *The basics of practical optimization*. Philadelphia:Society for Industrial and Applied Mathematics .
- Li, Z., Qian, F., Du, W. ve Zhong, W. (2017). DE based economic control chart design and application for a typical petrochemical process. *Frontiers of Engineering Management*, 4(3), 348-356.
- Liang, J. J., Qu, B. Y. ve Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635(2), 1-33.
- Lin, J., Zhang, S. X., Zheng, S. Y. ve Pan, Y. M. (2022). Differential evolution with fusion of local and global search strategies. *Journal of Computational Science*, 63, 101746.
- Lin, M., Wang, Z., Chen, D. ve Zheng, W. (2022). Particle swarm-differential evolution algorithm with multiple random mutation. *Applied Soft Computing*, 120, 108640.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M. ve Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43-58.
- Meng, Z., Pan, J.-S. ve Kong, L. (2018). Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution. *Knowledge-Based Systems*, 141, 92-112.

- Meng, Z., Pan, J.-S. ve Tseng, K.-K. (2019). PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. *Knowledge-Based Systems*, 168, 80-99.
- Meng, Z. ve Yang, C. (2022). Two-stage differential evolution with novel parameter control. *Information Sciences*, 596, 321-342.
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228-249.
- Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133.
- Mirjalili, S. ve Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51-67.
- Pham, D. T. ve Castellani, M. (2014). Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms. *Soft Computing*, 18(5), 871-903.
- Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2), 155-162.
- Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12, 241-254.
- Price, K. V., Awad, N. H., Ali, M. Z. ve Suganthan, P. N. (2018). *Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization*. Nanyang Technological University Singapore. https://www3.ntu.edu.sg/home/epnsugan/index_files/GECCO2019/GECCO2019.htm
- Qin, A. K., Huang, V. L. ve Suganthan, P. N. (2008). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), 398-417.
- Rahnamayan, S., Tizhoosh, H. R. ve Salama, M. M. A. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, 53(10), 1605-1614.

- Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7(1), 19-34.
- Sharma, D. K., Hota, H. S., Brown, K. ve Handa, R. (2022). Integration of genetic algorithm with artificial neural network for stock market forecasting. *International Journal of System Assurance Engineering and Management*, 13(Suppl 2), 828-841.
- Siddiqui, A. W., Sarhadi, H. ve Verma, M. (2023). A robust optimization approach to risk-averse routing of marine crude oil tankers. *Computers & Industrial Engineering*, 175, 108878.
- Srinivas, M. ve Patnaik, L. M. (1994). Genetic algorithms: a survey. *Computer*, 27(6), 17-26.
- Storn, R. ve Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341-359.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Tan, Z., Li, K. ve Wang, Y. (2021). Differential evolution with adaptive mutation strategy based on fitness landscape analysis. *Information Sciences*, 549, 142-163.
- Tanabe, R. ve Fukunaga, A. (2013). Evaluating the performance of SHADE on CEC 2013 benchmark problems. *IEEE Congress on Evolutionary Computation*, Mexico, 1952-1959.
- Tanabe, R. ve Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. *2014 IEEE Congress on Evolutionary Computation (CEC), China*, 1658-1665.
- Vasant, P. M. (2012). *Meta-heuristics optimization algorithms in engineering, business, economics, and finance*. USA:IGI Global.
- Vommi, A. M. ve Battula, T. K. (2023). A binary Bi-phase mutation-based hybrid equilibrium optimizer for feature selection in medical datasets classification. *Computers and Electrical Engineering*, 105, 108553.
- Wang, D., Tan, D. ve Liu, L. (2018). Particle swarm optimization algorithm: an overview.

Soft Computing, 22, 387-408.

- Wang, X., Wang, Y., Wong, K.-C. ve Li, X. (2022). A self-adaptive weighted differential evolution approach for large-scale feature selection. *Knowledge-Based Systems*, 235, 107633.
- Wang, Y., Cai, Z. ve Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 15(1), 55-66.
- Yang, X.-S., Deb, S., Loomes, M. ve Karamanoglu, M. (2013). A framework for self-tuning optimization algorithm. *Neural Computing and Applications*, 23(7), 2051-2057.
- Yavuz, G. (2022). Senior learning JAYA with powell's method and incremental population strategy. *IEEE Access*, 10, 103765-103780.
- Yavuz, G. (2021). 100 Basamak probleminin JADE algoritması ile çözülmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (21), 493-500.
- Yavuz, G., Aydın, D. ve Stützle, T. (2016). Self-adaptive search equation-based artificial bee colony algorithm on the CEC 2014 benchmark functions. *2016 IEEE Congress on Evolutionary Computation (CEC)*, 1173-1180.
- Yavuz, G. ve Aydın, D. (2019). Improved self-adaptive search equation-based artificial bee colony algorithm with competitive local search strategy. *Swarm and Evolutionary Computation*, 51, 100582.
- Zeng, Z., Zhang, M., Chen, T. ve Hong, Z. (2021). A new selection operator for differential evolution algorithm. *Knowledge-Based Systems*, 226, 107150.
- Zhang, J. ve Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945-958.
- Zhang, J. ve Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5), 945-958.

DİZİN

A	
A Sine Cosine Algorithm	5, 44
Algoritma	5, 7, 9, 14, 15, 22, 23, 24, 25, 26, 27
Aritmetik ortalama	29
B	
Balina Optimizasyon Algoritması, 4, 30, 39	
BOA	4, 5, 30, 31
Butterfly Optimization Algorithm	5
C	
CEC 2014 Ölçüt Seti	18
CEC 2019 ölçüt seti	2, 4, 30, 31, 39
CNN	1
CNN mimarisi	1
Convolutional Neural Network	1
CVaR	1
D	
Diferansiyel Gelişim Algoritması ., 1, 2, 7, 8, 24, 39	
E	
EEG sinyalleri	39
G	
Genetik Algoritma	1, 12, 14
Genişletilmiş Balina Optimizasyon Algoritması	30, 39
Güve Alevi Optimizasyon Algoritması.. 39	
I	
IPOPCMAES	24, 26, 27
İ	
İklendirme	6, 15
irace	2, 19, 20, 32, 43
J	
JAYA algoritması	5, 30
K	
Kelebek Optimizasyon Algoritması... 5, 30, 39	
Kendinden Uyarlamalı, 1, 10, 22, 24, 39	
KOA	5
konvolüsyonel sinir ağı	1
KUYA-DGA..24, 25, 32, 33, 34, 35, 36, 37, 39	
M	
MFO	4, 30, 31
Moth Flame Optimization	4
Mutasyon	7, 10, 15, 22, 23, 39
O	
optimizasyon algoritması	4, 5, 29
P	
POWELL Metodu	27
S	
SCA	5, 30, 31, 44
Sinüs Kosinüs Algoritması	5, 30, 39
SKA	5
Standart sapma	29, 31
W	
Whale Optimization Algorithm	4
WOA	4, 30, 31
WOAmM	4, 30, 31
X	
X-ray	1, 41